

В.И. ДЖИГАН

Адаптивная фильтрация сигналов: теория и алгоритмы

$$\begin{aligned} \mathbf{t}_N(k) &= \mathbf{E}_{NM}(k-1) \mathbf{c}_M(k), \\ \mathbf{q}_N(k) &= \mathbf{t}_N(k) - \mathbf{H}_{NM}^*(k-1) \mathbf{b}_{M,\min}^*(k-1) \mathbf{c}_M(k) = \\ &= \mathbf{t}_N(k) - \mathbf{H}_{NM}^*(k-1) \tilde{\mathbf{q}}_M(k), \\ \mathbf{t}_N(k) &= \mathbf{t}_N(k-1) + (k-1) \mathbf{q}_N(k), \\ \mathbf{t}_N(k) &= \tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^*(k-1) \tilde{\mathbf{q}}_M(k). \end{aligned}$$



ТЕХНОСФЕРА



М И Р

цифровой обработки

В.И. ДЖИГАН

Адаптивная
фильтрация сигналов:
теория и алгоритмы

ТЕХНОСФЕРА
Москва
2013

УДК 621.372

ББК 32.811.3

Д41

Д41 Джиган В.И.

Адаптивная фильтрация сигналов: теория и алгоритмы

Москва: Техносфера, 2013. – 528 с., ISBN 978-5-94836-342-4

В книге рассматриваются основные разновидности адаптивных фильтров и их применение в радиотехнических системах и системах связи. Дается представление о математических объектах и методах, используемых в теории адаптивной фильтрации сигналов. Рассматриваются приемы получения вычислительных процедур, сами процедуры и свойства таких алгоритмов адаптивной фильтрации, как алгоритмы Ньютона и наискорейшего спуска, алгоритмы по критерию наименьшего квадрата, рекурсивные алгоритмы по критерию наименьших квадратов и их быстрые (вычислительно эффективные) версии; рекурсивные алгоритмы по критерию наименьших квадратов для многоканальных фильтров и их версии для обработки нестационарных сигналов, а также многоканальные алгоритмы аффинных проекций. Дано описание стандартных и нестандартных приложений для моделирования адаптивных фильтров на современных языках программирования MATLAB, LabVIEW и SystemVue, а также реализаций адаптивных фильтров на современных цифровых сигнальных процессорах отечественного и зарубежного производства.

Особенностью книги является изложения теоретических материалов для наиболее общего случая – адаптивных фильтров с комплексными весовыми коэффициентами, наличие разделов по многоканальным адаптивным фильтрам и алгоритмам адаптивной фильтрации нестационарных сигналов.

Книга является первым систематическим изложением теории адаптивной фильтрации на русском языке.

Она предназначена для научных работников, инженеров, аспирантов и студентов радиотехнических и связных специальностей, изучающих и использующих на практике цифровую обработку сигналов и, в частности, адаптивную фильтрацию сигналов.

УДК 621.372

ББК 32.811.3

© 2013, В.И. Джиган

© 2013, ЗАО «РИЦ «Техносфера», оригинал-макет, оформление.

ISBN 978-5-94836-342-4

СОДЕРЖАНИЕ

Список сокращений	8
Список основных обозначений	10
Предисловие	15
Глава 1. Введение в адаптивную обработку сигналов	20
1.1. Введение	20
1.2. Требования к адаптивным фильтрам	24
1.3. Критерии функционирования адаптивных фильтров	31
1.4. Идентификация неизвестной линейной системы	33
1.5. Компенсация эхо-сигналов	37
1.6. Выравнивание характеристик электрических каналов связи	44
1.7. Адаптивные антенные решетки	51
1.8. Шумоочистка сигналов	53
1.9. Линейное предсказание сигналов	57
1.10. Выводы по главе	58
Глава 2. Оператор комплексного градиента и его применение в теории адаптивной фильтрации сигналов	59
2.1. Введение	59
2.2. Несколько простых примеров дифференцирования функций действительных и комплексных переменных	60
2.3. Дифференцирование комплексных и действительных функций комплексного переменного	63
2.4. Свойства оператора комплексного градиента	69
2.5. Выводы по главе	71
Глава 3. Основы адаптивной фильтрации сигналов	72
3.1. Введение	72
3.2. Корреляционная матрица	73
3.3. Собственные числа и собственные векторы корреляционной матрицы	78
3.4. Винеровская фильтрация	86
3.5. Поверхность среднеквадратической ошибки	89
3.6. Пример расчета собственных чисел, собственных векторов корреляционной матрицы и поверхности среднеквадратической ошибки	99
3.7. Линейно-ограниченная винеровская фильтрация	102
3.8. Выводы по главе	109



Глава 4. Поиск винеровского решения	110
4.1. Введение	110
4.2. Алгоритм Ньютона	111
4.3. Постоянные времена алгоритма Ньютона	114
4.4. Алгоритм наискорейшего спуска	118
4.5. Постоянные времена алгоритма наискорейшего спуска	120
4.6. Другая интерпретация алгоритма Ньютона	131
4.7. Выводы по главе	134
Глава 5. LMS-алгоритм и его свойства	135
5.1. Введение	135
5.2. LMS-алгоритм	135
5.3. Переходные процессы в LMS-алгоритме	139
5.4. Качество адаптивной фильтрации сигналов с помощью LMS-алгоритма	146
5.5. NLMS-алгоритм	166
5.6. Градиентные алгоритмы с переменным шагом сходимости	171
5.7. Линейно-ограниченные LMS- и NLMS-алгоритмы	173
5.8. Выводы по главе	181
Глава 6. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов	182
6.1. Введение	182
6.2. Задача наименьших квадратов и ее решение	182
6.3. Основные свойства LS-решения	186
6.4. Геометрическая интерпретация метода наименьших квадратов	190
6.5. Рекурсивная задача наименьших квадратов	194
6.6. Решение рекурсивной задачи наименьших квадратов	196
6.7. Качество адаптивной фильтрации с помощью RLS-алгоритма	205
6.8. Выводы по главе	217
Глава 7. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов на основе обратного QR-разложения	218
7.1. Введение	218
7.2. RLS-фильтрация с помощью прямого QR-разложения	219
7.3. Рекурсивное вычисление вектора весовых коэффициентов	222
7.4. Рекурсивное вычисление некоторых переменных	225
7.5. Обратное QR-разложение	228
7.6. Вычисление обратного разложения Холецкого	236
7.7. Параллельная реализация IQRD RLS-алгоритма	239
7.8. Выводы по главе	243



Глава 8. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов на базе прямого QR-разложения	244
8.1. Введение	244
8.2. Рекурсивная форма прямого QR-разложения	245
8.3. Рекурсивное вычисление ошибок	250
8.4. Параллельная реализация QRD RLS-алгоритма	257
8.5. Выводы по главе	259
Глава 9. Уменьшение вычислительной сложности рекурсивных алгоритмов адаптивной фильтрации по критерию наименьших квадратов	260
9.1. Введение	260
9.2. Преобразование Хаусхолдера	261
9.3. RLS-алгоритм, использующий преобразование Хаусхолдера	265
9.4. Исключение операций извлечения квадратного корня в алгоритмах на основе обратного QR-разложения	270
9.5. Исключение операций извлечения квадратного корня в алгоритмах на основе прямого QR-разложения	277
9.6. Выводы по главе	283
Глава 10. Линейно-ограниченная рекурсивная адаптивная фильтрация по критерию наименьших квадратов	284
10.1. Введение	284
10.2. Постановка и решение линейно-ограниченной RLS-задачи адаптивной фильтрации	284
10.3. Линейно-ограниченный RLS-алгоритм	287
10.4. Двойное ограничение в RLS-алгоритме	291
10.5. RLS-алгоритм с двойным ограничением и линейной вычислительной сложностью	293
10.6. Выводы по главе	297
Глава 11. Быстрые рекурсивные адаптивные алгоритмы по критерию наименьших квадратов	298
11.1. Введение	298
11.2. Линейное предсказание вперед	300
11.3. Линейное предсказание назад	304
11.4. Быстрое вычисление вектора коэффициентов Калмана	307
11.5. Соотношение между апостериорными и априорными ошибками	312
11.6. Рекурсивное вычисление отношения ошибок	313
11.7. Быстрый алгоритм Калмана	315
11.8. FTF-алгоритм	317



11.9. FAEST-алгоритм.....	320
11.10. Выводы по главе.....	323
Глава 12. Быстрые лестничные алгоритмы.....	324
12.1. Введение.....	324
12.2. Рекурсивное вычисление энергий ошибок линейного предсказания.....	325
12.3. Рекурсивное вычисление ошибок линейного предсказания и коэффициентов отражения	330
12.4. Рекурсивное вычисление ошибок моделирования требуемого сигнала адаптивного фильтра.....	334
12.5. Рекурсивные адаптивные алгоритмы на основе вычисления априорных и апостериорных ошибок.....	338
12.6. Рекурсивные адаптивные алгоритмы с обратными связями на основе вычисления априорных и апостериорных ошибок.....	343
12.7. Нормализованный лестничный алгоритм.....	350
12.8. Алгоритм на основе QR-разложения с операциями извлечения квадратного корня.....	359
12.9. Алгоритм на основе QR-разложения без операций извлечения квадратного корня.....	370
12.10. Алгоритм на основе QR-разложения с операциями извлечения квадратного корня в пространстве состояний.....	373
12.11. Выводы по главе.....	376
Глава 13. Многоканальные RLS-алгоритмы.....	377
13.1. Введение.....	377
13.2. B-SUSD линейное предсказание.....	378
13.3. B-SUSD-вычисление векторов коэффициентов Калмана и отношений ошибок линейного предсказания.....	385
13.4. B-SUSD быстрые RLS-алгоритмы.....	392
13.5. 1-SUSD линейное предсказание.....	398
13.6. 1-SUSD вычисление векторов коэффициентов Калмана и отношений ошибок линейного предсказания.....	402
13.7. 1-SUSD быстрые RLS-алгоритмы.....	411
13.8. Выводы по главе.....	418
Глава 14. Другие разновидности RLS-алгоритмов.....	419
14.1. Введение.....	419
14.2. Быстрые алгоритмы на основе обратного QR-разложения.....	420
14.3. Регуляризация RLS-алгоритмов.....	425
14.4. RLS-алгоритмы со скользящим окном	428

14.5. Одновременное использование скользящего окна и регуляризации в RLS-алгоритмах.....	432
14.6. Параллельные RLS-алгоритмы, допускающие вычисления с помощью двух или четырех процессоров.....	438
14.7. Особенности построения быстрых многоканальных алгоритмов аффинных проекций.....	443
14.8. Инициализация RLS-алгоритмов.....	458
14.9. Выводы по главе.....	463
 Глава 15. Применение адаптивных алгоритмов.....	 464
15.1. Введение.....	464
15.2. Моделирование адаптивных фильтров с помощью приложений DSP System Toolbox языка MATLAB.....	466
15.3. Нестандартные программные средства для моделирования адаптивных фильтров на языке MATLAB.....	471
15.4. Моделирование приложений адаптивной фильтрации с помощью Adaptive Filter Toolkit среды разработки LabVIEW.....	472
15.5. Нестандартные программные средства для моделирования адаптивных фильтров в среде разработки LabVIEW.....	484
15.6. Моделирование приложений адаптивной фильтрации с помощью Adaptive Equalization Library среды разработки SystemVue.....	489
15.7. Библиотека адаптивной фильтрации для ЦСП отечественного семейства «Мультикор».....	494
15.8. Приложения адаптивной фильтрации для цифровых сигнальных процессоров компаний Texas Instruments, Analog Devices и Freescale Semiconductor.....	498
15.9. Выводы по главе.....	501
 Заключение.....	 502
 Список литературы.....	 505
 Предметный указатель.....	 521

СПИСОК СОКРАЩЕНИЙ

ААР	— адаптивная антенная решетка.
АЦП	— аналого-цифровой преобразователь.
АЧХ	— амплитудно-частотная характеристика.
БИХ	— бесконечная импульсная характеристика.
БПФ	— быстрое преобразование Фурье.
ВК	— весовой коэффициент.
ГИП	— графический интерфейс пользователя.
ДН	— диаграмма направленности.
КИХ	— конечная импульсная характеристика.
МНК	— метод наименьшего квадрата.
ОСШ	— отношение сигнал—шум.
ПЛИС	— программируемая логическая интегральная схема.
РМНК	— рекурсивный метод наименьших квадратов.
ФНЧ	— фильтр низких частот.
ФЧХ	— фазочастотная характеристика.
СБИС	— сверхбольшая интегральная схема.
ЦАП	— цифро-аналоговый преобразователь.
ЦОС	— цифровая обработка сигналов.
ЦСП	— цифровой сигнальный процессор.
1-SUSD	— 1 Step Up Step Down.
B-SUSD	— Block Step Up Step Down.
CM	— Constant Modulus, критерий постоянства модуля (огибающей) информационных символов.
DPSK	— Differential PSK, дифференциальная квадратурная фазовая модуляция.
ERLE	— Echo Return Loss Enhancement, коэффициент ослабления эхо-сигнала.
FAEST	— Fast a Posteriori Error Sequential Technique.
FAP	— Fast Affine Projections, быстрый алгоритм аффинных проекций.
FK	— Fast Kalman.
FPGA	— Field-Programmable Gate Array, программируемая логическая матрица.
FTF	— Fast Transversal Filter.
IQRD	— Inverse QRD, обратное QR-разложение.
LC	— Linearly Constrained, линейно-ограниченный.
LC MV	— Linearly Constrained Minimum Variance.
LC MVDR	— LC MV Distortionless Response.
LMS	— Least Mean Square, алгоритм по критерию наименьшего среднего квадрата.
LS	— Least Squares, критерий наименьших квадратов.

- MIL – Matrix Inversion Lemma, лемма об обращении матрицы.
- MIMO – Multi Input – Multi Output, система со многими входами и многими выходами.
- MSE – Mean Square Error, среднеквадратическая ошибка.
- MV – Minimum Variance.
- NLMS – Normalized Least Mean Square, нормализованный алгоритм по критерию наименьшего среднего квадрата.
- PSK – Phase Shift Keying, квадратурная фазовая модуляция.
- QAM – Quadrature Amplitude Modulation, квадратурная амплитудная модуляция.
- QRD – QR Decomposition, QR-разложение.
- RLS – Recursive Least Squares, рекурсивные алгоритмы по критерию наименьших квадратов.
- SISD – Single Instruction Single Data, скалярный режим ЦСП.
- UL – UL-разложение.
- VI – Virtual Instrument, виртуальный прибор.
- WLS – Weighted Least Squares, взвешенный критерий средних квадратов.

СПИСОК ОСНОВНЫХ ОБОЗНАЧЕНИЙ

$x(k)$ — входной сигнал адаптивного фильтра.

$d(k)$ — требуемый сигнал адаптивного фильтра.

$y(k), y_N(k)$ — выходной сигнал адаптивного фильтра.

$\alpha(k), \alpha_N(k)$ — априорная ошибка моделирования требуемого сигнала.

$e(k), e_N(k)$ — апостериорная ошибка моделирования требуемого сигнала.

$\alpha_N^f(k)$ — априорная ошибка линейного предсказания вперед входного сигнала.

$e_N^f(k)$ — апостериорная ошибка линейного предсказания вперед входного сигнала.

$\alpha_N^b(k)$ — априорная ошибка линейного предсказания назад входного сигнала.

$e_N^b(k)$ — апостериорная ошибка линейного предсказания назад входного сигнала.

$\varphi_N(k)$ — отношение апостериорной и априорной ошибок.

$e_o(k), z(k)$ — сигнал аддитивного шума.

σ_x^2 — дисперсия входного сигнала.

σ_d^2 — дисперсия требуемого сигнала.

σ_o^2 — дисперсия аддитивного шума.

$\mathbf{x}_N(k)$ — вектор (с числом элементов $N \times 1$) сигналов адаптивного фильтра.

$\mathbf{x}_{N_m}(k)$ — вектор (с числом элементов $N_m \times 1$) сигналов в m -м канале многоканального адаптивного фильтра.

$\xi \mathbf{p}_N(k)$ — вектор (с числом элементов $N \times 1$) искусственных сигналов динамической регуляризации корреляционной матрицы адаптивного фильтра.

$h_n(k)$ — n -й комплексный весовой коэффициент адаптивного фильтра.

$\mathbf{h}_N(k)$ — вектор (с числом элементов $N \times 1$) комплексных весовых коэффициентов адаптивного фильтра.

$\Delta h_n(k)$ — n -й комплексный весовой коэффициент адаптивного фильтра в смещенной системе координат.

$\Delta \mathbf{h}_N(k)$ — вектор (с числом элементов $N \times 1$) комплексных весовых коэффициентов адаптивного фильтра в смещенной системе координат.

$v_N(k)$ — n -й комплексный весовой коэффициент адаптивного фильтра в смещенной и повернутой системе координат.

$\mathbf{v}_N(k)$ — вектор (с числом элементов $N \times 1$) комплексных весовых коэффициентов адаптивного фильтра в смещенной и повернутой системе координат.

$\mathbf{h}_{N,o}$ — вектор (с числом элементов $N \times 1$) оптимальных комплексных весовых коэффициентов адаптивного фильтра (винеровское решение, координаты минимума функции среднеквадратической ошибки).



$\mathbf{h}_{N,o}$ — вектор (с числом элементов $N \times 1$) оптимальных комплексных весовых коэффициентов адаптивного фильтра (винеровское решение, координаты минимума функции среднеквадратической ошибки) в задаче линейно-ограниченной адаптивной фильтрации.

$\tilde{h}_n(k)$ — n -й действительный весовой коэффициент адаптивного фильтра.

$\tilde{\mathbf{h}}_N(k)$ — вектор (с числом элементов $N \times 1$) действительных весовых коэффициентов адаптивного фильтра.

$\Delta\tilde{h}_n(k)$ — n -й действительный весовой коэффициент адаптивного фильтра в смещенной системе координат.

$\Delta\tilde{\mathbf{h}}_N(k)$ — вектор (с числом элементов $N \times 1$) действительных весовых коэффициентов адаптивного фильтра в смещенной системе координат.

$\tilde{v}_n(k)$ — n -й действительный весовой коэффициент адаптивного фильтра в смещенной и повернутой системе координат.

$\tilde{\mathbf{v}}_N(k)$ — вектор (с числом элементов $N \times 1$) действительных весовых коэффициентов адаптивного фильтра в смещенной и повернутой системе координат.

$\tilde{\mathbf{h}}_{N,o}$ — вектор (с числом элементов $N \times 1$) действительных оптимальных весовых коэффициентов адаптивного фильтра (винеровское решение, координаты минимума функции среднеквадратической ошибки).

$\mathbf{h}_{N_m}(k)$ — вектор (с числом элементов $N_m \times 1$) комплексных весовых коэффициентов в m -м канале многоканального адаптивного фильтра.

$\mathbf{h}_N^f(k)$ — вектор (с числом элементов $N \times 1$) комплексных весовых коэффициентов фильтра линейного предсказания вперед.

$\mathbf{h}_N^b(k)$ — вектор (с числом элементов $N \times 1$) комплексных весовых коэффициентов фильтра линейного предсказания назад.

N — число весовых коэффициентов адаптивного фильтра.

N_m — число весовых коэффициентов в m -м канале многоканального адаптивного фильтра.

M — число каналов в многоканальном адаптивном фильтре, относительное значение дополнительной среднеквадратической ошибки.

k — индекс дискретного времени (номер отсчета обрабатываемого сигнала и итерации алгоритма).

K — число итераций алгоритма.

K_E — число итераций блочного алгоритма, на длительности которых оценивается значение градиента целевой функции.

$\mathbf{R}_N, \mathbf{R}_N(k)$ — корреляционная матрица (с числом элементов $N \times N$).

R_{mn} — mn -й элемент матрицы \mathbf{R}_N .

\mathbf{R}_N^{-1} — обратная корреляционная матрица (с числом элементов $N \times N$).

$\tilde{\mathbf{R}}_N(k)$ — верхняя треугольная матрица (с числом элементов $N \times N$), результат RQ-разложения.

δ^2 — параметр начальной и динамической регуляризации корреляционной матрицы.

λ_n — n -е собственное число корреляционной матрицы.

χ — отношение максимального λ_{\max} и минимального λ_{\min} собственных чисел корреляционной матрицы.

$\mathbf{q}_{N,n}$ — n -й собственный вектор (с числом элементов $N \times 1$) корреляционной матрицы.

\mathbf{Q}_N — матрица (с числом элементов $N \times N$) из собственных векторов корреляционной матрицы.

Λ_N — диагональная матрица собственных чисел (с числом элементов $N \times N$).

\mathbf{S}_N — матрица Вандермонда (с числом элементов $N \times N$).

\mathbf{I}_N — единичная квадратная матрица (с числом элементов $N \times N$).

\mathbf{O}_N — нулевая квадратная матрица (с числом элементов $N \times N$).

\mathbf{O}_{NM} — нулевая прямоугольная матрица (с числом элементов $N \times M$).

$\text{diag}\{\bullet\}$ — диагональная квадратная матрица.

$\text{cov}\{\bullet\}$ — ковариационная матрица.

$\det\{\bullet\}$ — определитель квадратной матрицы.

$\text{tr}\{\bullet\}$ — след квадратной матрицы.

\mathbf{r}_N — вектор (с числом элементов $N \times 1$) взаимной корреляции между входными сигналами линейного сумматора и требуемым сигналом.

\mathbf{i}_N — единичный вектор (с числом элементов $N \times 1$).

$\mathbf{0}_N$ — нулевой вектор (с числом элементов $N \times 1$).

\mathbf{e}_N — направляющий вектор (с числом элементов $N \times 1$).

\mathbf{T} — знак операции транспонирования вектора или матрицы.

\mathbf{H} — знак операции эрмитова сопряжения вектора или матрицы.

$E\{\bullet\}$ — операция усреднения по ансамблю реализаций.

F — целевая функция.

$\mathbf{w}_N, \mathbf{w}_\infty$ — векторы (с числом элементов $N \times 1$ и $\infty \times 1$) весовых коэффициентов импульсного оклика идентифицируемой линейной системы или канала связи.

∞ — символ «бесконечность».

$|\bullet|$ — операция взятия модуля от комплексного числа.

$\|\bullet\|_2$ — евклидова норма вектора.

$\delta(\bullet)$ — делта-функция.

$\text{Re}\{a\}$ — действительная часть комплексного числа a .

$\text{Im}\{a\}$ — мнимая часть комплексного числа a .

$j = \sqrt{-1}$ — мнимая единица.

$*$ — операция комплексного сопряжения комплексной переменой.

∇ — оператор градиента.

$\hat{\nabla}$ — оценка вектора градиента целевой функции.

$\bar{\nabla}$ — оценка вектора градиента целевой функции в блочных адаптивных алгоритмах.

Δ — приращение переменной.

$\xi, \xi(k)$ — среднеквадратическая ошибка.

ξ_{\min} — минимальное значение среднеквадратической ошибки.

ξ_c — среднеквадратическая ошибка в задаче линейно-ограниченной адаптивной фильтрации.

ξ_{exc} — дополнительная среднеквадратическая ошибка.

\mathbf{C}_{NJ} — матрица (с числом элементов $N \times J$) J ограничений.

$\mathbf{c}_{N,j}$ — j -й вектор (с числом элементов $J \times 1$) матрицы \mathbf{C}_{NJ} .

\mathbf{f}_j — вектор (с числом элементов $J \times 1$) значений ограничивающего параметра.

λ_j — вектор (с числом элементов $J \times 1$) множителей Лагранжа.

$\Lambda_j(k)$ — матрица множителей Лагранжа (с числом элементов $J \times J$).

\mathbf{P}_N — проекционная матрица (с числом элементов $N \times N$).

μ — постоянный шаг сходимости адаптивного алгоритма.

$\mu(k)$ — переменный шаг сходимости адаптивного алгоритма.

μ_{\max} — максимально допустимое значение шага сходимости адаптивного алгоритма.

$\hat{\mu}$ — масштабирующий множитель динамического шага сходимости.

k_{τ_h} — постоянная времени переходного процесса весовых коэффициентов адаптивного фильтра.

k_{τ_ξ} — постоянная времени переходного процесса обучающей кривой адаптивного фильтра.

L — число статистически независимых испытаний при усреднении обучающих кривых, длина скользящего окна (число отсчетов) для оценки корреляционной матрицы при обработке нестационарных сигналов, размер (число отсчетов) в алгоритме аффинных проекций.

λ — параметр экспоненциального взвешивания сигналов в рекурсивных алгоритмах по критерию наименьших квадратов.

$E_N(k)$ — целевая функция в рекурсивных алгоритмах по критерию наименьших квадратов.

$E_{N,\min}(k)$ — минимальное значение целевой функции в рекурсивных алгоритмах по критерию наименьших квадратов.

$E_N^f(k)$ — энергия ошибок линейного предсказания вперед.

$E_{N,\min}^f(k)$ — минимальное значение энергии ошибок линейного предсказания вперед.

$E_N^b(k)$ — энергия ошибок линейного предсказания назад.

$E_{N,\min}^b(k)$ — минимальное значение энергии ошибок линейного предсказания назад.

$\mathbf{g}_N(k)$ — апостериорный вектор коэффициентов Калмана.

$\mathbf{t}_N(k)$ — априорный вектор коэффициентов Калмана.

$\Gamma_n^f(k)$, $\Gamma_n^b(k)$, $\gamma_n(k)$ — коэффициенты отражения в лестничных адаптивных алгоритмах.

\mathbf{T}_{N+M} , \mathbf{S}_{N+M} — перестановочные матрицы (с числом элементов $[N+M] \times [N+M]$) в многоканальных B-SUSD адаптивных алгоритмах.

$\mathbf{T}_{N+1}^{(m)}$, $\mathbf{S}_{N+1}^{(m)}$ — перестановочные матрицы (с числом элементов $[N+1] \times [N+1]$) m -го канала в многоканальных 1-SUSD адаптивных алгоритмах.

\mathbf{S}_F — матрица (с числом элементов $F \times F$) знаков в параллельных адаптивных алгоритмах.

$\mathbf{d}_F(k)$ — вектор (с числом элементов $1 \times F$) отсчетов требуемых сигналов адаптивного фильтра в параллельных адаптивных алгоритмах.

$\mathbf{a}_F(k)$ — вектор (с числом элементов $1 \times F$) отсчетов априорных ошибок адаптивного фильтра в параллельных адаптивных алгоритмах.

$\mathbf{G}_{NF}(k)$ — матрица (с числом элементов $N \times F$) апостериорных коэффициентов Калмана адаптивного фильтра в параллельных адаптивных алгоритмах.

$\mathbf{X}_{NF}(k)$ — матрица (с числом элементов $N \times F$) входных сигналов адаптивного фильтра в параллельных адаптивных алгоритмах.

$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \begin{bmatrix} c & s^* \\ -s & c \end{bmatrix}, \begin{bmatrix} c & -s^* \\ s & c \end{bmatrix}$ — матрицы элементарных вращений Гивенса.

$O(N)$ — линейная вычислительная сложность.

$O(N^2)$ — квадратичная вычислительная сложность.

$O(N^3)$ — кубическая вычислительная сложность.

T_s — период дискретизации обрабатываемых сигналов.

F_s — частота дискретизации обрабатываемых сигналов.

$\rho(k)$ — рассогласование, т.е. евклидова мера расстояния между вектором весовых коэффициентов адаптивного фильтра и вектором весовых коэффициентов идентифицируемого импульсного отклика.

$ERLE(k)$ — коэффициент ослабления эхо-сигнала.

Посвящается памяти моих родителей
Марии Васильевны и Ивана Андреевича

Виктор Джиган

ПРЕДИСЛОВИЕ

Цифровая обработка сигналов (ЦОС) сегодня широко используется в радиолокационных и радионавигационных системах, системах радио, проводной и оптической связи; в вычислительной и бытовой технике. Это обусловлено огромными теоретическими достижениями за последние шесть десятилетий в ЦОС, а также технологическими достижениями электронной техники, позволившими создавать одновременно компактные и высокопроизводительные цифровые устройства для реализации алгоритмов ЦОС в реальном масштабе времени. Столь длительное развитие теории и практики ЦОС привело к появлению в этой области знаний ряда самостоятельных научно-технических направлений. Адаптивная обработка сигналов — это одно из таких направлений.

Адаптивная обработка сигналов зарождалась в середине прошлого века. Однако корни используемых в ней математических методов уходят в более遠ое прошлое. Затем эти методы были использованы Винером, Колмогоровым, Крейном, Левинсоном и рядом других ученых для формирования основных принципов адаптивной обработки сигналов. Использованию же адаптивной обработки сигналов на практике человечество обязано Бэрнарду Уидроу (Bernard Widrow), который в конце 50-х годов прошлого века разработал и реализовал первый адаптивный фильтр на базе Least Mean Square (LMS) алгоритма, также известного как алгоритм по методу наименьшего квадрата (МНК).

Когда же требуется использовать адаптивную обработку сигналов, под которой обычно понимается адаптивная фильтрация? Тогда, когда с поставленной задачей не могут справиться фильтры с фиксированными параметрами. Как правило, это случаи когда условия фильтрации меняются, а потому требования к параметрам фильтра не могут быть сформулированы заранее.

Адаптивный фильтр — это фильтр с перестраиваемыми весовыми коэффициентами (ВК). В зависимости от решаемой задачи он может быть одноканальным или многоканальным, иметь действительные или комплексные ВК. Алгоритмы вычисления ВК могут быть простыми, как, например, LMS-алгоритм, или сложными, как, например, Recursive Least Squares (RLS) алгоритм, также известный как рекурсивный метод наименьших квадратов (РМНК).

Сегодня существует огромное число радиоэлектронных изделий, показатели работы которых немыслимы без использования адаптивных фильтров.

Примерами таких изделий являются адаптивные антенные решетки, адаптивные компенсаторы сигналов электрического и акустического эха, адаптивные эквалайзеры (выравниватели) электрических характеристик каналов связи, адаптивные компенсаторы шумов и ряд других устройств. Адаптивные фильтры, входящие в состав этих устройств, в зависимости от используемого алгоритма вычисления ВК демонстрируют различную функциональную эффективность. Для ответа на вопросы, почему это так и от чего это зависит, необходимо понимать, что представляет собой адаптивная фильтрация сигналов с математической точки зрения.

Ответы на эти вопросы содержатся в многочисленных публикациях в периодических изданиях, трудах научно-технических конференций и технических книгах, посвященных адаптивной обработке сигналов, в основном на английском языке и труднодоступных большинству российских читателей. Переводные книги [1–3] не содержат в себе многих важных аспектов адаптивной обработки сигналов, а также уже являются библиографической редкостью. Книги российских авторов [4, 5] являются узкоспециализированными, так как отражают лишь аспекты адаптивной обработки сигналов в антенных решетках.

Настоящая книга является первым систематическим изложением теории (математических основ) и алгоритмов (практики) адаптивной фильтрации на русском языке. В книге рассматривается большое число разнообразных алгоритмов адаптивной фильтрации: от простейших алгоритмов на базе стратегий градиентного спуска до сложных алгоритмов по критерию наименьших квадратов. Изложение материала дается для общего случая фильтрации комплексных сигналов, в то время как в большинстве других книг аналогичные материалы излагаются, как правило, лишь для случая фильтрации действительных сигналов. Рассматриваются алгоритмы для одноканальных и многоканальных адаптивных фильтров. Книга иллюстрирована большим числом рисунков и графиков, отражающих результаты численных экспериментов. И хотя в книге не преследуется цель описания приложений адаптивных фильтров, в ней приводятся общие сведения о таких наиболее известных приложениях, как адаптивные антенные решетки, эхо-компенсаторы, эквалайзеры каналов связи и ряд других.

Книга структурирована следующим образом.

В главе 1 приводятся общие сведения об адаптивной обработке сигналов, о структурах, целевых функциях и критериях работы адаптивных фильтров, а также показателях их качества. Рассматриваются примеры применения адаптивных фильтров при решении практических задач, встречающихся в радиотехнике и связи.

В главе 2 дается определение оператора комплексного градиента и рассматриваются его свойства. Анализируются результаты применения этого опера-

тора к действительным и комплексным целевым функциям. Эти результаты также сравниваются с результатами применения оператора действительного градиента к действительным целевым функциям.

Глава 3 посвящена фундаментальным понятиям адаптивной обработки сигналов, таким как корреляционная матрица, собственные числа и собственные векторы этой матрицы, поверхность среднеквадратической ошибки, безусловная и условная (линейно-ограниченная) винеровская фильтрация сигналов.

В главе 4 описываются два способа поиска решения задачи винеровской фильтрации: алгоритм Ньютона и алгоритм наискорейшего спуска. Анализируются свойства этих алгоритмов и приводятся результаты их численного моделирования.

В главе 5 рассматривается LMS-алгоритм адаптивной фильтрации. Анализируются свойства этого алгоритма и приводятся результаты его численного моделирования. Также рассматриваются нормализованная версия LMS-алгоритма (NLMS-алгоритм), LMS- и NLMS-алгоритмы с переменным шагом сходимости, изменяемым по градиентному закону, и линейно-ограниченные версии LMS- и NLMS-алгоритмов.

В главе 6 рассматриваются постановка и решение задачи идентификации линейных объектов по критерию наименьших квадратов, а также RLS-алгоритм адаптивной фильтрации, вытекающий из этой задачи и полученный с помощью леммы об обращении матриц для рекурсивного обращения корреляционной матрицы входных сигналов адаптивного фильтра. Анализируются свойства этого алгоритма и приводятся результаты его численного моделирования.

В главе 7 рассматривается RLS-алгоритм адаптивной фильтрации сигналов на основе обратного QR-разложения с использованием вращений Гивенса. Представлены основные шаги получения алгоритма, его вычислительная процедура, а также вопросы аппаратной реализации, ориентированной на параллельные вычисления.

В главе 8 рассматривается RLS-алгоритм адаптивной фильтрации сигналов, основанный на прямом QR-разложении с использованием вращений Гивенса.

В главе 9 рассматривается RLS-алгоритм адаптивной фильтрации сигналов, основанный на обратном QR-разложении с использованием преобразования Хаусхолдера. Также описывается способ исключения операций извлечения квадратного корня в RLS-алгоритмах на основе прямого и обратного QR-разложений с использованием вращений Гивенса.

В главе 10 описаны три алгоритма линейно-ограниченной рекурсивной адаптивной фильтрации сигналов по критерию наименьших квадратов, основанные на использовании леммы об обращении матриц.



В главе 11 рассматриваются быстрые, т.е. вычислительно эффективные, RLS-алгоритмы адаптивной фильтрации, основанные на использовании лемм об обращении клеточных матриц.

Глава 12 посвящена вопросам построения быстрых лестничных RLS-алгоритмов. Приводятся вычислительные процедуры десяти разновидностей таких алгоритмов.

В главе 13 рассматриваются многоканальные версии быстрых RLS-алгоритмов, рассмотренных в главах 6–11, для адаптивных фильтров с неодинаковым числом весовых коэффициентов в каналах.

В главе 14 рассматривается ряд RLS-алгоритмов адаптивной фильтрации, которые могут быть построены на основе приемов, описанных в главах 6–13. Это быстрые алгоритмы на основе QR-разложения, а также алгоритмы обработки нестационарных сигналов и их параллельные версии, которые могут использоваться как самостоятельно, так и в качестве части вычислительных процедур в быстрых алгоритмах аффинных проекций. Кроме того, в главе освещены вопросы инициализации (задания начальных значений ВК) RLS-алгоритмов.

В главе 15 приводятся сведения о стандартных и нестандартных средствах моделирования адаптивных фильтров и их приложений на языке программирования MATLAB, LabVIEW и SystemVue. Кроме того, приводятся сведения о прикладных адаптивных фильтрах, реализованных на современных цифровых сигнальных процессорах отечественного и зарубежного производства.

Таким образом, в книге изложены основные теоретические вопросы адаптивной обработки комплексных сигналов, рассмотрены алгоритмы вычисления ВК для одно- и многоканальных адаптивных фильтров, включая фильтры, предназначенные для обработки нестационарных сигналов. В ней обобщены и представлены с единой точки зрения как известные, так и оригинальные сведения, которые отражают результаты моих многолетних исследований в области адаптивной обработки сигналов.

Я надеюсь, что у читателей сложится систематическое представление об адаптивной обработке сигналов, а полученные знания вызовут мотивацию для ее дальнейшего изучения, развития и использования на практике.

Книга будет полезна как зрелым, так и начинающим научным работникам, инженерам, а также аспирантам и студентам. Книга получилась достаточно объемной, хотя многие вопросы адаптивной обработки сигналов остались за ее рамками. Подробное изложение делает эту книгу самодостаточной, а потому простой и понятной.

Насколько это удалось, судить читателям. Я буду признателен за любые замечания и пожелания, касающиеся этой книги. Ваши отзывы можно направлять по адресу электронной почты djigan@yandex.ru.

Пользуясь случаем, я хотел бы выразить благодарность моим учителям Незлину Д. В., Бецу В. П., Колюскину В. А. и Мингазину А. Т. за полученные под их руководством знания за время учебы в Московском институте электронной техники (сейчас Национальный исследовательский университет «МИЭТ»), приведшие меня в конечном итоге в адаптивную обработку сигналов, а также коллегам и друзьям Александрову Ю. Н., Архипкину В. Я., Акопяну А. Г., Быстрову А. Н., Брюханову Ю. А., Вечтомову В. А., Витязеву В. В., Витязеву С. В., Головенко Ю. Н., Гринкевичу В. Э., Гудковой Н. В., Гурееву А. В., Егорову С. И., Засову В. А., Захарову Ю. В., Земляку А. М., Лесникову В. А., Лупину С. А., Петричковичу Я. Я., Петрову А. С., Петрову В. Г., Плетневой И. Д., Приорову А. Л., Карпенко А. Л., Комарову В. Т., Коплович Е. А., Светлову А. Ю., Семушкину И. В., Сергиенко А. Б., Солохиной Т. В., Ханяну Г. С., Хрящеву В. В., Чистюхину В. В., Чобану М. К. и Щербакову С. В. за постоянную поддержку и оценку моей работы.

Отдельная благодарность сотрудникам российских представительств компаний Agilent Technologies и National Instruments Бардашовой О. Н., Бурматову А. В., Жигунову Б. С., Иванисову А. В., Кондратьеву Д. Е., Петренко Б. П., Подольскому А. С., Руданову Г. С., Сепояну П. Р. и Чутко О. В. за поддержку моих работ и предоставление информационных материалов по программным продуктам LabVIEW и SystemVue, вошедших в главу 15 настоящей книги.

Я также выражаю благодарность своим зарубежным коллегам Джону М. Чиоффи (John M. Cioffi), Стивену Л. Гранту (Steven L. Grant) и Эдварду Джонсу (Edward Jones), внесшим свой вклад в формирование моего профессионального интереса к адаптивной обработке сигналов.

Я благодарю сестру Александру и брата Василия за их постоянную поддержку.

Особая благодарность моей жене Галине и дочерям Марии и Ольге за их понимание и поддержку на протяжении написания этой книги и в повседневной жизни.

ГЛАВА I

ВВЕДЕНИЕ В АДАПТИВНУЮ ОБРАБОТКУ СИГНАЛОВ

Настоящая глава является введением в теорию адаптивной обработки сигналов. В ней приводятся общие сведения об адаптивных фильтрах, их структурах, целевых функциях, критериях работы, а также показателях качества. Рассматриваются примеры применения адаптивных фильтров при решении практических задач, встречающихся в радиотехнике и связи: адаптивные эхо-компенсаторы, эквалайзеры, антенные решетки, адаптивная шумоочистка и адаптивная фильтрация узкополосных сигналов.

I.I. Введение

Согласно разным оценкам на сегодняшний день в научно-технических журналах и трудах научно-технических конференций насчитывается более 10 тысяч публикаций, освещающих самые разные вопросы теории и практики адаптивной обработки сигналов. Эти вопросы также рассматриваются в ряде книг [1—33]. Такое число публикаций свидетельствует об огромных теоретических достижениях в этой области, нашедших свое отражение в адаптивных устройствах, применяемых на протяжении уже многих десятилетий в различных радиоэлектронных изделиях. Примерами адаптивных устройств являются компенсаторы сигналов электрического и акустического эха, эквалайзеры (выравниватели) характеристик электрических и акустических каналов связи, адаптивные антенные и акустические решетки [34—36]. Сегодня можно констатировать тот факт, что адаптивные устройства — это неотъемлемые элементы оборудования современных радиотехнических систем и систем связи, тактико-технические характеристики которых порой недостижимы без использования адаптивной обработки сигналов.

Действительно, сигналы ближнего и дальнего эха являются источниками помех для модемов, работающих в дуплексном режиме, а межсимвольная интерференция является источником помех в каналах связи с ограниченной по-



лосой пропускания. Для достижения модемами высоких скоростей передачи и приема цифровых данных эти помехи должны быть подавлены, что осуществляется с помощью адаптивных эхо-компенсаторов и эквалайзеров.

Положительная обратная акустическая связь, возникающая в оборудовании озвучивания помещений (концертные залы, залы для совещаний) и в устройствах голосовой связи («hands-free»), устраняется с помощью адаптивных компенсаторов сигналов акустического эха. Выравнивание акустических характеристик помещений при использовании оборудования высококачественного воспроизведения звука осуществляется с помощью адаптивных эквалайзеров.

Подавление активных помех, частотный диапазон которых совпадает с частотным диапазоном полезного сигнала, в радиолокации, гидролокации и связи осуществляется с помощью адаптивных антенных и акустических решеток. Эти решетки являются пространственными фильтрами. Они обеспечивают подавление сигналов помех, направления на источники которых отличаются от направления на источник полезного сигнала.

Практические результаты адаптивной обработки сигналов во многом сегодня обязаны технологическим достижениям в электронике и микроэлектронике, благодаря которым появились одновременно высокопроизводительные и компактные цифровые устройства на базе программируемых логических интегральных схем (ПЛИС) и сверхбольших интегральных схем (СБИС) цифровых сигнальных процессоров (ЦСП). Эти устройства обеспечивают реализацию сложных алгоритмов ЦОС в реальном масштабе времени, т. е. на требуемой частоте дискретизации обрабатываемых сигналов.

Таким образом, многолетние теоретические и практические достижения в адаптивной обработке сигналов позволяют считать ее сложившимся научно-техническим направлением в современных радиотехнике и связи.

Но что такое адаптивная обработка сигналов, или, в более широком смысле, адаптация? Вот лишь некоторые определения этого термина [3]. «Адаптация — это:

- действие, процесс приспособления;
- состояние приспособляемого, приспособление;
- любое изменение в структуре или функции организма или любой из его частей в результате естественного отбора, с помощью которого организм становится более приспособленным для выживания и размножения в окружающей его среде; видоизменение формы или структуры в соответствии с изменением окружающей среды;
- ослабление отклика сенсорных рецепторов, таких как зрение, осязание, температура, обоняние, слух и боль, на изменяющиеся, постоянно воздействующие условия окружающей среды;

- регулирование зрачком количества света, поступающего в глаз;
- медленное, обычно не осознанное изменение индивидуальной и социальной деятельности в процессе приспособления к культурной среде».

Перечисленные определения термина «адаптация» характеризуют в основном живой мир. Однако этот термин применим и к искусственным устройствам и системам, которые в процессе своего функционирования могут самообучаться. В основе большинства искусственных адаптивных устройств лежит **адаптивный фильтр**.

Фильтрация сигналов — это такая их обработка, целью которой является извлечение интересуемой информации, например сигналов, обладающих определенными характеристиками. Обработка сигналов может выполняться как аналоговым, так и цифровым способом. В книге мы будем рассматривать только цифровую фильтрацию, предполагая, что обрабатываемые сигналы дискретизованы с частотой, регламентируемой теоремой Котельникова, известной в зарубежной литературе как теорема отсчетов, и оцифрованы с числом разрядов, определяемым динамическим диапазоном этих сигналов.

Известно, что свойства цифрового фильтра с фиксированными параметрами обычно определяются его передаточной функцией. В свою очередь, передаточная функция определяет структуру фильтра и его вычислительную сложность. Если требования к передаточной функции невозможно сформулировать заранее или эти требования могут меняться в процессе работы фильтра, то вместо фильтра с фиксированными параметрами необходимо использовать фильтр с изменяемыми параметрами, например адаптивный.

Так как параметры адаптивного фильтра меняются в процессе его работы, то очевидно, что такой фильтр является нелинейным устройством. Однако при каждом фиксированном значении параметров адаптивный фильтр представляет собой линейное устройство, так как между его входными и выходными сигналами обычно существует линейная зависимость, определяемая текущим набором ВК, подобно линейным фильтрам с фиксированными параметрами.

Таким образом, адаптивный фильтр — это фильтр с изменяемыми в процессе работы параметрами, набор которых во многом зависит от критерия работы адаптивного фильтра. Этим критерием часто является достижение минимума некоторой целевой функции, как правило, квадратичной функции ошибки между так называемым требуемым и выходным сигналами адаптивного фильтра. Достижение минимума целевой функции означает, что выходной сигнал адаптивного фильтра «близок» к требуемому сигналу, т. е. повторяет по форме этот сигнал.

Выходной сигнал адаптивного фильтра приближается к требуемому сигналу за счет изменения ВК, рассчитываемых на основе обработки требуемого и вход-



ного (входных в многоканальном случае) сигналов. В установившемся режиме значения ВК соответствуют минимуму целевой функции или находятся в небольшой окрестности этого минимума. Так как состояние адаптивного фильтра меняется в процессе настройки ВК, то считается, что он приспосабливается или адаптируется к существующим условиям функционирования. Поэтому и такой фильтр, и алгоритм вычисления его ВК называют адаптивными.

Адаптивные алгоритмы являются основным предметом рассмотрения в настоящей книге. Это, собственно, алгоритмы фильтрации, т.е. вычисления выходного сигнала адаптивного фильтра, и алгоритмы вычисления его ВК. Сочетание данных алгоритмов будем равнозначно называть или адаптивным фильтром, или адаптивным алгоритмом.

Анализ адаптивных фильтров как нелинейных устройств является более сложным, чем анализ фильтров с фиксированными параметрами. Но адаптивные фильтры являются самонастраивающимися и с этой точки зрения они более простые, так как не требуют использования сложных и ресурсоемких методов расчета, применяемых при синтезе фильтров с фиксированными ВК.

Если условия функционирования меняются, то адаптивный фильтр может эти изменения в некоторой степени отслеживать. При этом следует понимать, что адаптивный фильтр и следящая система — это не совсем одно и то же. Адаптивный фильтр, подобно фильтрам с фиксированными ВК, характеризуется длительностью переходного процесса. Если изменения в системе медленные, т.е. происходят за время, существенно превышающее длительность переходного процесса адаптивного фильтра, то этот фильтр такие изменения, как правило, отслеживает. По мере же увеличения скорости изменений в системе эффективность адаптивного фильтра падает, так как за время изменения он не успевает «полностью настроиться», т.е. перейти в установившийся режим, когда его переходный процесс считается законченным.

В общем случае одноканальный адаптивный фильтр представляет собой устройство, показанное на рис. 1.1. Здесь $x(k)$ — входной сигнал, $d(k)$ — требуемый сигнал (используемый в обозначении символ «d» происходит от английского слова «desired», т.е. требуемый), $y(k)$ — выходной сигнал, $\alpha(k) = d(k) - y(k)$ — сигнал ошибки, k — индекс дискретного времени или номер отсчета обрабатываемого сигнала. Эти отсчеты обычно равномерно распределены на оси времени как $t = kT_s = k/F_s$, где T_s — период дискретизации, а F_s — частота дискретизации обрабатываемых сигналов. Длительность одной итерации адаптивного алгоритма, т.е. время, в течение которого осуществляется расчет ВК, используемых для вычисления выходного сигнала адаптивного фильтра на следующей итерации, в большинстве случаев равна одному периоду дискретизации. Если расчет ВК ведется медленно, т.е. в течение нескольких периодов дискретизации, то формирование сигнала $y(k)$ все равно необходимо

осуществлять на каждом периоде дискретизации для удовлетворения требованиям теоремы Котельникова.

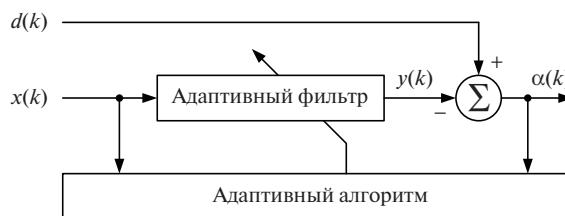


Рис. 1.1. Адаптивный фильтр

Для работы адаптивного фильтра, как правило, недостаточно информации, получаемой только из входного сигнала. Источником недостающей информации обычно служит требуемый сигнал $d(k)$, физическая природа которого определяется конкретным приложением адаптивного фильтра. Приложение является одним из факторов, определяющих требования, предъявляемые к адаптивному фильтру на этапе его проектирования.

1.2. Требования к адаптивным фильтрам

Согласно [29] основные требования к адаптивному фильтру определяются такими составляющими, как приложение, структура и алгоритм.

1. Приложения. Тип приложения определяется сигналами, поступающими из окружающей среды в качестве входного $x(k)$ и требуемого $d(k)$ сигналов. Наиболее известными и широко используемыми приложениями адаптивных фильтров являются уже упоминавшиеся эхо-компенсаторы, эквалайзеры, адаптивные решетки и ряд других устройств. Данная книга не преследует своей целью рассмотрение приложений адаптивных фильтров. Однако некоторые из них будут кратко рассмотрены в настоящей главе с целью демонстрации областей использования адаптивных фильтров при решении задач радиотехники и связи.

2. Структуры адаптивных фильтров. Подобно фильтрам с фиксированными ВК, существуют две основные структуры адаптивных фильтров. Это фильтры с конечной импульсной характеристикой (КИХ), или трансверсальные, и фильтры с бесконечной импульсной характеристикой (БИХ), или рекурсивные. Структура одноканального адаптивного КИХ-фильтра приведена на рис. 1.2.

Выходной сигнал $y(k)$ адаптивного фильтра формируется как линейная комбинация задержанных отсчетов входного сигнала $x(k-n+1)$, взятых с весами $h_n(k-1)$, вычисляемыми на предыдущих итерациях $(k-1)$ по отношению к текущим итерациям k , т. е. как

$$y(k) = \sum_{n=1}^N h_n^*(k-1) x(k-n+1) = \mathbf{h}_N^H(k-1) \mathbf{x}_N(k), \quad (1.1)$$

где N — число ВК фильтра, а $\mathbf{h}_N(k-1) = [h_1(k-1), h_2(k-1), \dots, h_n(k-1), \dots, h_{N-1}(k-1), h_N(k-1)]^T$ — вектор ВК и $\mathbf{x}_N(k) = [x(k), x(k-1), \dots, x(k-n+1), \dots, x(k-N+2), x(k-N+1)]^T$ — вектор сигналов в фильтре. Порядок фильтра определяется как $N-1$, т. е. по числу линий задержки. Векторы $\mathbf{h}_N(k)$ и $\mathbf{x}_N(k)$ в общем случае являются комплексными.

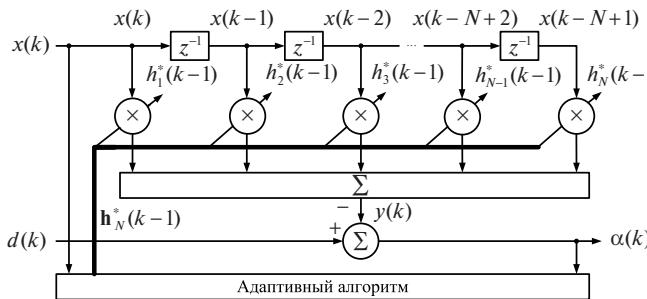


Рис. 1.2. Одноканальный адаптивный КИХ-фильтр

Здесь и далее верхний индекс Т будет использоваться для обозначения операции транспонирования вектора или матрицы, а верхний индекс Н — для обозначения операции эрмитова сопряжения, т. е. транспонирования и комплексного сопряжения, обозначаемого символом *, элементов вектора или матрицы. Первый нижний индекс N в обозначениях векторов и матриц будет указывать на число элементов в векторе или число элементов в квадратной матрице, определяемое как $N \times N$. Число элементов в прямоугольной матрице будет обозначаться двумя первыми нижними индексами. Например, запись \mathbf{A}_{NM}^T будет обозначать операцию транспонирования матрицы \mathbf{A}_{NM} , т. е. матрицу $\mathbf{B}_{MN} = \mathbf{A}_{NM}^T$. Таким образом, нижние индексы в обозначении матриц будут указывать на число элементов нетранспонированных матриц. В ряде случаев нижний индекс N также будет использоваться в обозначениях скалярных переменных, указывая на то, что эти переменные являются функциями от N -мерных векторов для различения одноименных скалярных переменных, являющихся функциями от $(N+1)$ -, N - или $(N-1)$ -мерных векторов в быстрых алгоритмах адаптивной фильтрации.

Векторы везде будут обозначаться полужирными строчными символами, а матрицы — полужирными заглавными. Элементы матриц и векторов будут обозначаться соответственно обычными строчными и заглавными символами. Нижние индексы в элементах матриц и векторов будут обозначать номера этих элементов в матрицах и векторах согласно общепринятой системе их нумерации. В качестве других признаков в обозначениях векторов, матриц или их элементов могут быть использованы различные символы в нижнем индексе,

следующие через запятую после обозначения размера вектора или матрицы или номеров их элементов, а также символы в верхнем индексе. Единичная матрица (квадратная диагональная матрица с единицами на главной диагонали и нулевыми остальными элементами) будет обозначаться символом \mathbf{I}_N , нулевые матрицы (матрицы, содержащие только нулевые элементы) — символами $\mathbf{0}_N$ и $\mathbf{0}_{NM}$, единичный вектор (вектор, содержащий все единицы) — символом \mathbf{i}_N , а нулевой вектор (вектор, содержащий только нулевые элементы) — символом $\mathbf{0}_N$. Под вектором обычно будет пониматься вектор-столбец.

Используя выходной сигнал адаптивного фильтра (1.1), можно вычислить сигнал априорной ошибки моделирования требуемого сигнала $d(k)$ как

$$\alpha(k) = d(k) - y(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k). \quad (1.2)$$

Сигнал апостериорной ошибки определяется при ВК, равных $\mathbf{h}_N(k)$, как

$$e(k) = d(k) - \hat{y}(k) = d(k) - \mathbf{h}_N^H(k)\mathbf{x}_N(k). \quad (1.3)$$

Как видно, термины «априорный» и «апостериорный» связаны со значениями ВК адаптивного фильтра, вычисляемыми соответственно на предыдущей и текущей итерациях алгоритма адаптивной фильтрации.

На практике при работе адаптивного фильтра на его выходе наблюдаются априорные ошибки (1.2), так как текущее значение выходного сигнала фильтра $y(k)$ формируется по значениям ВК, вычисленным на предыдущей итерации. Апостериорные ошибки (1.3) обычно используются в алгоритмах вычисления ВК и при формировании целевой функции работы адаптивного фильтра, например среднеквадратической ошибки (Mean Square Error, MSE) $F = E\{e(k)e^*(k)\} = E\{|e(k)|^2\}$, где $E\{\bullet\}$ — операция усреднения по ансамблю реализаций. В случае КИХ-фильтров такая функция, как будет показано в главе 3, является действительной унимодальной квадратичной функцией в пространстве действительных или комплексных ВК, т. е. характеризуется единственным минимумом.

В книге будут рассматриваться в основном адаптивные фильтры с комплексными ВК. Это обусловлено некоторыми математическими тонкостями, используемыми при получении алгоритмов вычисления ВК для таких фильтров, а также тем, что переход от адаптивных алгоритмов для фильтров с действительными ВК к алгоритмам для фильтров с комплексными ВК часто не очевиден, несмотря на то, что он сводится в основном к правильной расстановке операций комплексного сопряжения некоторых переменных, используемых в алгоритмах.

Поэтому в настоящей книге теория адаптивной фильтрации будет излагаться применительно к фильтрам с комплексными ВК, так как в большинстве литературных источников адаптивные алгоритмы рассматриваются в основном для фильтров с действительными ВК. Переход от описания фильтров с комплекс-

ными ВК к описанию фильтров с действительными ВК является тривиальным и обычно сводится к исключению всех операций комплексного сопряжения в матрицах, векторах и скалярных переменных. Этот переход, как будет показано в главе 2, сопровождается появлением в некоторых математических выражениях фиксированного множителя, равного 2, который отсутствует в математических выражениях алгоритмов для адаптивных фильтров с комплексными ВК.

В адаптивных антенных решетках (AAP) или эхо-компенсаторах и эквализерах оборудования цифровых систем связи с квадратурной модуляцией обрабатываются комплексные сигналы. Это естественным образом приводит к необходимости использования адаптивных фильтров с комплексными ВК.

В общем случае адаптивные КИХ-фильтры могут быть многоканальными и при этом содержать неодинаковое число ВК в каналах (рис. 1.3). Вектор ВК такого M -канального адаптивного фильтра

$$\mathbf{h}_N(k-1) = [\mathbf{h}_{N_1}^T(k-1), \mathbf{h}_{N_2}^T(k-1), \dots, \mathbf{h}_{N_m}^T(k-1), \dots, \mathbf{h}_{N_{M-1}}^T(k-1), \mathbf{h}_{N_M}^T(k-1)]^T$$

образуется из последовательности векторов ВК каналов

$$\mathbf{h}_{N_m}(k) = [h_{1,m}(k-1), h_{2,m}(k-1), \dots, h_{n,m}(k-1), \dots, h_{N_m-1,m}(k-1), h_{N_m,m}(k-1)]^T,$$

а вектор сигналов $\mathbf{x}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_{M-1}}^T(k), \mathbf{x}_{N_M}^T(k)]^T$ – из последовательности векторов сигналов каналов

$$\mathbf{x}_{N_m}(k) = [x_m(k), x_m(k-1), \dots, x_m(k-n_m+1), \dots, x_m(k-N_m+2), x_m(k-N_m+1)]^T.$$

Суммарное число ВК многоканального фильтра определяется как $N = \sum_{m=1}^M N_m$.

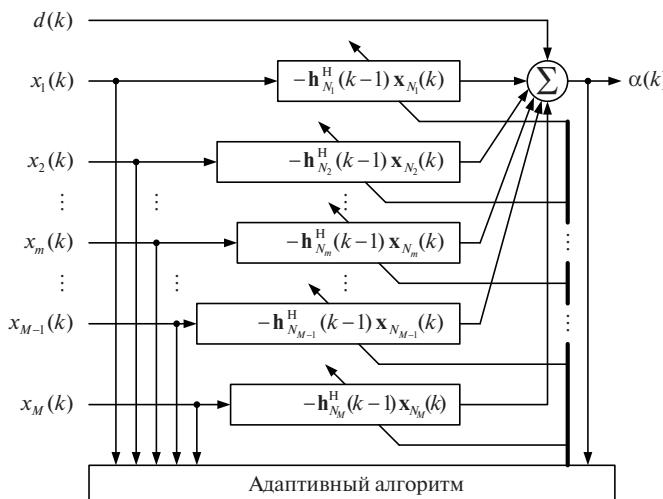


Рис. 1.3. Многоканальный адаптивный КИХ-фильтр

Неодинаковое число ВК в каналах многоканального адаптивного фильтра, с одной стороны, часто диктуется физической природой задачи, а с другой — обусловлено всегда существующими ограничениями на вычислительную сложность реализации фильтров. Так как в адаптивных фильтрах вычислительная сложность является функцией полного числа ВК N , то это число не следует увеличивать, если на то нет каких-либо причин, несмотря даже на то, что ряд алгоритмов, например для многоканальных адаптивных фильтров с одинаковым числом ВК в каналах, математически (алгоритмически) «много проще» алгоритмов для фильтров с неодинаковым числом ВК в каналах.

Под вычислительной сложностью алгоритмов ЦОС, к которым принадлежат и алгоритмы адаптивной фильтрации сигналов, понимается число арифметических операций (обычно сложений, вычитаний, умножений и делений, а иногда и более сложных операций, например извлечения квадратных корней), требуемых для выполнения одной итерации алгоритма.

Следует отметить две частные структуры адаптивного фильтра (см. рис. 1.3). Это многоканальный фильтр с одним ВК ($N_m=1$, $m=1, 2, \dots, M$) в каждом из каналов, который используется в узкополосных ААР, и многоканальный фильтр с одинаковым числом ВК в каналах $N_1=N_2=\dots=N_M>1$, который используется в широкополосных акустических решетках [1, 4–6, 23] или многоканальных компенсаторах сигналов акустического эха [37]. Структура многоканального фильтра общего вида (см. рис. 1.3) также используется при реализации нелинейных полиномиальных адаптивных фильтров, в которых нелинейные ядра представляют собой наборы из многоканальных фильтров с числом ВК в каналах, изменяемым от $N_1=1$ до $N_M=N$ с шагом, равным одному ВК [28]. Кроме того, компенсатор сигналов ближнего и дальнего эха в модемах для проводных каналов связи [38] или эквалайзер с обратной связью [39, 40] можно также рассматривать как двухканальный адаптивный фильтр с неодинаковым числом ВК в каналах.

В настоящей книге в основном будут рассматриваться одноканальные адаптивные фильтры. В главе 13 будут рассмотрены многоканальные адаптивные фильтры общего вида (см. рис. 1.3).

Адаптивные БИХ-фильтры [16] на сегодняшний день пока не нашли широкого применения на практике, так как, помимо проблем с устойчивостью, им свойственна проблема многоэкстремальности (т. е. неунимодальности) целевой функции, что не позволяет в общем случае гарантировать сходимость процесса вычисления ВК к глобальному (т. е. наилучшему) решению.

3. Алгоритмы. Адаптивный алгоритм — это процедура вычисления ВК, обеспечивающих минимизацию целевой функции адаптивного фильтра, т. е. выполнения критерия его работы. Алгоритм характеризуется видом используемой целевой функции, методом поиска оптимального решения и природой

сигналов ошибок. В адаптивных алгоритмах, как правило, используются арифметические операции как над априорными, так и над апостериорными ошибками. Процедуры вычисления выходных сигналов адаптивного фильтра $y(k) = \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$ и $\hat{y}(k) = \mathbf{h}_N^H(k)\mathbf{x}_N(k)$ в уравнениях (1.2) и (1.3), т. е. собственно фильтрация сигналов, также часто считаются составными частями адаптивного алгоритма.

Большинство вычислений в адаптивных алгоритмах представляет собой итерационные процедуры. Эти итерации, как правило, совпадают по длительности с периодом дискретизации обрабатываемых сигналов. При этом вычисляемые переменные обновляются на каждой итерации в результате прибавления некоторых добавок к предыдущим значениям или вычитания этих добавок из предыдущих значений.

Например, вычисление ВК в адаптивном алгоритме по критерию наименьшего среднего квадрата (Least Mean Square, LMS) выполняется как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k), \quad (1.4)$$

т. е.

$$\left\{ \begin{array}{l} \text{текущее} \\ \text{значение} \\ \text{вектора ВК} \end{array} \right\} = \left\{ \begin{array}{l} \text{предыдущее} \\ \text{значение} \\ \text{вектора ВК} \end{array} \right\} + \left\{ \begin{array}{l} \text{шаг} \\ \text{сходимости} \end{array} \right\} \left\{ \begin{array}{l} \text{вектор} \\ \text{входных} \\ \text{сигналов} \end{array} \right\} \left\{ \begin{array}{l} \text{ошибка} \end{array} \right\}.$$

В рекурсивных алгоритмах по критерию наименьших квадратов (Recursive Least Squares, RLS) вычисление ВК выполняется как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha^*(k), \quad (1.5)$$

т. е.

$$\left\{ \begin{array}{l} \text{текущее} \\ \text{значение} \\ \text{вектора ВК} \end{array} \right\} = \left\{ \begin{array}{l} \text{предыдущее} \\ \text{значение} \\ \text{вектора ВК} \end{array} \right\} + \left\{ \begin{array}{l} \text{вектор} \\ \text{коэффициентов} \\ \text{Калмана} \end{array} \right\} \left\{ \begin{array}{l} \text{ошибка} \end{array} \right\}.$$

Здесь текущие и предыдущие значения векторов ВК связаны с номерами итераций алгоритма k и $k-1$. В ряде других алгоритмов, например в алгоритмах на основе QR-разложения и в лестничных алгоритмах, также используются рекурсивные вычисления, связанные с изменением порядка фильтра на единицу на каждой итерации по k .

Выбор адаптивного алгоритма обычно обусловлен рядом таких требований, как длительность переходного процесса, характеризующая скорость сходимости и следящие свойства адаптивного фильтра; значение остаточных ошибок в установившемся режиме, характеризующее точность нахождения оптимального решения; вычислительная сложность алгоритма, характеризующая сложность его аппаратной или программной реализации.

К этим требованиям следует добавить понятие алгоритмической сложности, под которой понимается сложность математического представления алгоритмов, обусловленная числом и разнообразием математических выражений (формул), являющихся, собственно, алгоритмом. Алгоритмическая сложность и арифметическая сложность в конечном итоге влияют на сложность реализации алгоритмов.

С точки зрения сложности алгоритмы адаптивной обработки сигналов можно условно разделить на несколько групп. Существуют вычислительно-простые алгоритмы, т. е. с малым числом арифметических операций на одну итерацию. Это алгоритмы с линейной вычислительной сложностью $O(N)$ и малой алгоритмической сложностью. Такие алгоритмы обычно описываются всего лишь несколькими простыми математическими выражениями. Примерами простых алгоритмов являются LMS-алгоритм и другие адаптивные алгоритмы на основе стратегий градиентного поиска ВК. Алгоритм по критерию наименьших квадратов (Least Squares, LS) характеризуется большой вычислительной сложностью $O(N^3)$ и малой алгоритмической сложностью. RLS-алгоритмы характеризуются средней вычислительной сложностью $O(N^2)$ и средней алгоритмической сложностью. Быстрые, т. е. вычислительно-эффективные или с малым числом арифметических операций на одну итерацию, RLS-алгоритмы характеризуются низкой вычислительной сложностью $O(N)$ и большой алгоритмической сложностью, так как представляются достаточно большим числом разнообразных математических формул.

При реализации адаптивных фильтров формально воспроизвести сложные алгоритмы адаптивной обработки сигналов не всегда удается с первого раза. Это происходит от того, что авторы книг и статей обычно не придерживаются единого обозначения переменных в адаптивных алгоритмах, опускают важные вопросы их инициализации, что не способствует однозначному пониманию этих алгоритмов.

В результате, на практике до сих пор в основном используются адаптивные фильтры на базе простых алгоритмов. Известный специалист в области адаптивной фильтрации Джон Чиофи (John Cioffi) в 1996 году сказал: «Эксперты на протяжении уже многих лет разрабатывают сложные и эффективные RLS-алгоритмы, однако инженеры их не любят, так как не понимают, а потому предпочитают использовать в основном простые градиентные алгоритмы» [41].

Такое отношение к сложным алгоритмам адаптивной фильтрации в прошлом было во многом обусловлено также и низкой производительностью цифровой элементной базы. Чтобы при реализации сложных алгоритмов обеспечивалась требуемая частота дискретизации сигналов, необходимо было располагать определенными вычислительными ресурсами, которые на то время были часто

недоступны. Однако сегодня такие алгоритмы [42] уже реализуемы на современной элементной базе [43, 44].

Поэтому цель настоящей книги — показать исследователям и разработчикам адаптивных устройств, как синтезируются простые и сложные алгоритмы адаптивной обработки сигналов, а также сравнить преимущества и недостатки таких алгоритмов по отношению друг к другу. Это должно способствовать использованию в современных приложениях не только простых, но и сложных алгоритмов адаптивной фильтрации.

1.3. Критерии функционирования адаптивных фильтров

В процессе работы адаптивного алгоритма осуществляется поиск таких параметров адаптивного фильтра, как правило, вектора ВК $\mathbf{h}_N(k)$, при которых значение заданной целевой функции является минимальным. Это обычно функция входного, выходного и требуемого сигналов, т. е. $F = f\{x(k), y(k), d(k)\}$. Целевую функцию можно рассматривать и как функцию ошибок, т. е. $F = f\{e(k)\}$. Она должна быть действительной и неотрицательной, т. е. $F \geq 0$ для всех $x(k)$, $y(k)$ и $d(k)$. При работе адаптивного фильтра $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$, где $\mathbf{h}_{N,o}$ — вектор координат минимума многомерной целевой функции в пространстве ВК. Достижение минимума этой функции является критерием работы адаптивного фильтра.

Существуют разные способы определения целевой функции адаптивного фильтра, влияющие в конечном итоге на сложность алгоритмов ее минимизации. Целевыми функциями, наиболее часто используемыми в адаптивной фильтрации, являются:

- 1) $F = f\{e(k)\} = E\{|e(k)|^2\}$ — среднеквадратическая ошибка (MSE);
- 2) $F = f\{e(k)\} = \sum_{i=1}^k |e(i)|^2$ — наименьшие квадраты ошибок (Least Squares, LS);
- 3) $F = f\{e(k)\} = \sum_{i=1}^k \lambda^{k-i} |e(i)|^2$ — взвешенные наименьшие квадраты ошибок (Weighted Least Squares, WLS);
- 4) $F = f\{e(k)\} = |e(k)|^2$ — квадрат мгновенной ошибки.

На практике используются и другие критерии, например критерий постоянства модуля информационных символов (Constant Modulus, CM) [45], часто применяемый при адаптивной обработке сигналов в системах цифровой связи. В основе этого критерия лежит минимизация функции ошибок между значением модуля (комплексной огибающей) выходного сигнала адаптивного фильтра и известным значением модуля (комплексной огибающей) информа-

ционных символов, возведенных в некоторую степень. Эта минимизация обеспечивает приближение комплексной огибающей выходного сигнала адаптивного фильтра к известному постоянному значению комплексной огибающей информационных символов. Поэтому данный критерий получил свое название не от названия целевой функции, а от названия требуемого конечного результата, т. е. значения огибающей выходного сигнала адаптивного фильтра, позволяющей корректно распознавать (обнаруживать) в этом сигнале принимаемые информационные символы.

Выбор квадратичных целевых функций в адаптивных фильтрах во многом обусловлен тем, что в результате их использования можно получать алгоритмы адаптивной фильтрации в виде рекуррентных вычислительных процедур, не содержащих логических операций, что отличает такие алгоритмы от вычислительных процедур оптимизации общего вида. Кроме того, эти функции являются унимодальными, что гарантирует сходимость алгоритмов адаптивной фильтрации на их основе к единственному оптимальному решению.

Строго говоря, первая MSE-функция является лишь удобным математическим понятием, так как для ее вычисления требуется наличие бесконечного объема данных, поскольку получение этой функции подразумевает усреднение по ансамблю реализаций наблюдаемых сигналов. На практике она применяется редко. MSE-функция используется, например, при анализе адаптивных фильтров или в оптимальной винеровской фильтрации, представляющей в основном теоретический интерес.

По словам Бернарда Уидроу (Bernard Widrow), автора LMS-алгоритма, винеровская фильтрация является лишь красивой теоретической задачей: «Я был знаком с теорией винеровской фильтрации в непрерывной и дискретной формах. Для построения фильтра Винера требуется знать автокорреляционную функцию входного сигнала и взаимную корреляционную функцию между входным и требуемым сигналами. Это замечательно, когда вы делаете домашнее упражнение, но что делать на практике, когда статистику входных сигналов никто вам не может предоставить? Все, что у вас есть, — это сами входные сигналы...» [46].

Целевые функции со второй по четвертую отличаются как сложностью реализации алгоритмов на их основе, так и характеристиками сходимости и остаточными ошибками этих алгоритмов в установленном режиме. Так, квадрат мгновенной ошибки является самой простой с точки зрения реализации алгоритмов целевой функцией. Однако алгоритмы на основе этой функции характеризуются медленной сходимостью, поскольку это сильно упрощенная целевая функция. Целевая LS-функция обычно используется при обработке стационарных сигналов, а WLS — при обработке медленно меняющихся сигналов.

В большей части настоящей книги основное внимание будет сосредоточено на алгоритмах, использующих в качестве целевых функций квадрат мгновенной ошибки, LS- и WLS-функции. Для простоты алгоритмы, использующие две последние функции, будут называться алгоритмами на основе критерия наименьших квадратов.

1.4. Идентификация неизвестной линейной системы

Существует ряд известных приложений адаптивных фильтров. Наиболее простым и легко понимаемым из них является прямая идентификация неизвестной линейной системы (рис. 1.4), базирующаяся на определении импульсного отклика этой системы. При решении задачи идентификации сигнал $x(k)$ является входным и для идентифицируемой системы, и для адаптивного фильтра. Поэтому минимизация некоторой функции ошибок $\alpha(k)$ между выходными сигналами неизвестной системы $d(k)$ и адаптивного фильтра $y(k)$ означает, что адаптивный фильтр моделирует неизвестную систему, т. е. $\mathbf{h}_N(k) \rightarrow \mathbf{w}_N$, так как фильтр «воспроизводит» выходной сигнал системы $y(k)$.

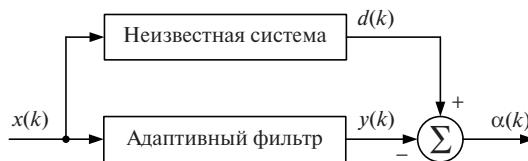


Рис. 1.4. Адаптивная идентификация неизвестной системы

На рис. 1.5 приведен пример моделирования работы адаптивного фильтра, характеризуемого вектором ВК $\mathbf{h}_N(k)$ (обозначен как «Фильтр», темная кривая), при решении задачи идентификации импульсного отклика \mathbf{w}_N (обозначен как «Система», светлая кривая) при $N=64$. На этом рисунке показаны значения ВК \mathbf{w}_N и $\mathbf{h}_N(k)$ на разных итерациях k работы адаптивного фильтра. Видно, что со временем, т. е. с увеличением номера итерации k , $\mathbf{h}_N(k) \rightarrow \mathbf{w}_N$.

На рис. 1.6 показаны сигналы, используемые при решении задачи идентификации (см. рис. 1.5). В качестве сигнала $x(k)$ использовался белый шум. Из рис. 1.6а, рис. 1.6в—з видно, что по мере приближения ВК адаптивного фильтра к ВК идентифицируемой линейной системы выходной сигнал фильтра $y(k)$ (обозначен как «Фильтр», темная кривая) приближается к сигналу $d(k)$ (обозначен как «Система», светлая кривая), а сигнал ошибки (см. рис. 1.6б) уменьшается, т. е. $|\alpha(k)| \rightarrow 0$.

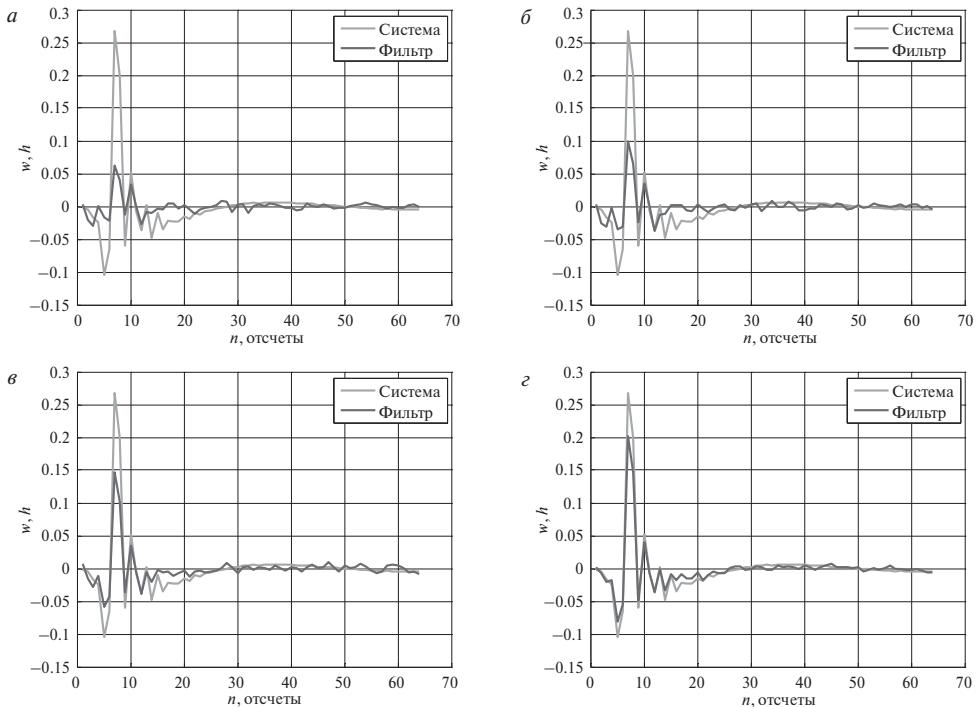


Рис. 1.5. Идентификация неизвестной линейной системы: а) $k=100$; б) $k=200$; в) $k=400$; г) $k=800$

Компенсация сигналов электрического или акустического эха является примером решения задачи идентификации неизвестной системы. На качество решения этой задачи, независимо от вида используемого алгоритма адаптивной фильтрации, влияет наличие аддитивного шума $e_o(k)$ на входе требуемого сигнала (рис. 1.7).

Шум $e_o(k)$ может быть как тепловым шумом в каналах проводной связи или акустическим шумом в помещении, так и шумом, обусловленным не полной идентификацией неизвестной системы из-за того, что точное число ВК идентифицируемого импульсного отклика обычно неизвестно. Поэтому на практике число ВК адаптивного фильтра N_h , как правило, меньше предполагаемого числа ВК N_w идентифицируемого импульсного отклика, что часто также обусловлено и ограничениями на вычислительную сложность алгоритмов адаптивной фильтрации, которая пропорциональна N_h .

Нескомпенсированная часть импульсного отклика образует шум, который оценивается как

$$\begin{aligned}
 e_o(k) &= d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \sum_{n=1}^{\infty} w_n^* x(k-n+1) - \sum_{n=1}^N h_n^*(k-1) x(k-n+1) = \\
 &= \sum_{n=1}^N w_n^* x(k-n+1) + \sum_{n=N+1}^{\infty} w_n^* x(k-n+1) - \sum_{n=1}^N h_n^*(k-1) x(k-n+1) = \sum_{n=N+1}^{\infty} w_n^* x(k-n+1)
 \end{aligned} \tag{1.6}$$

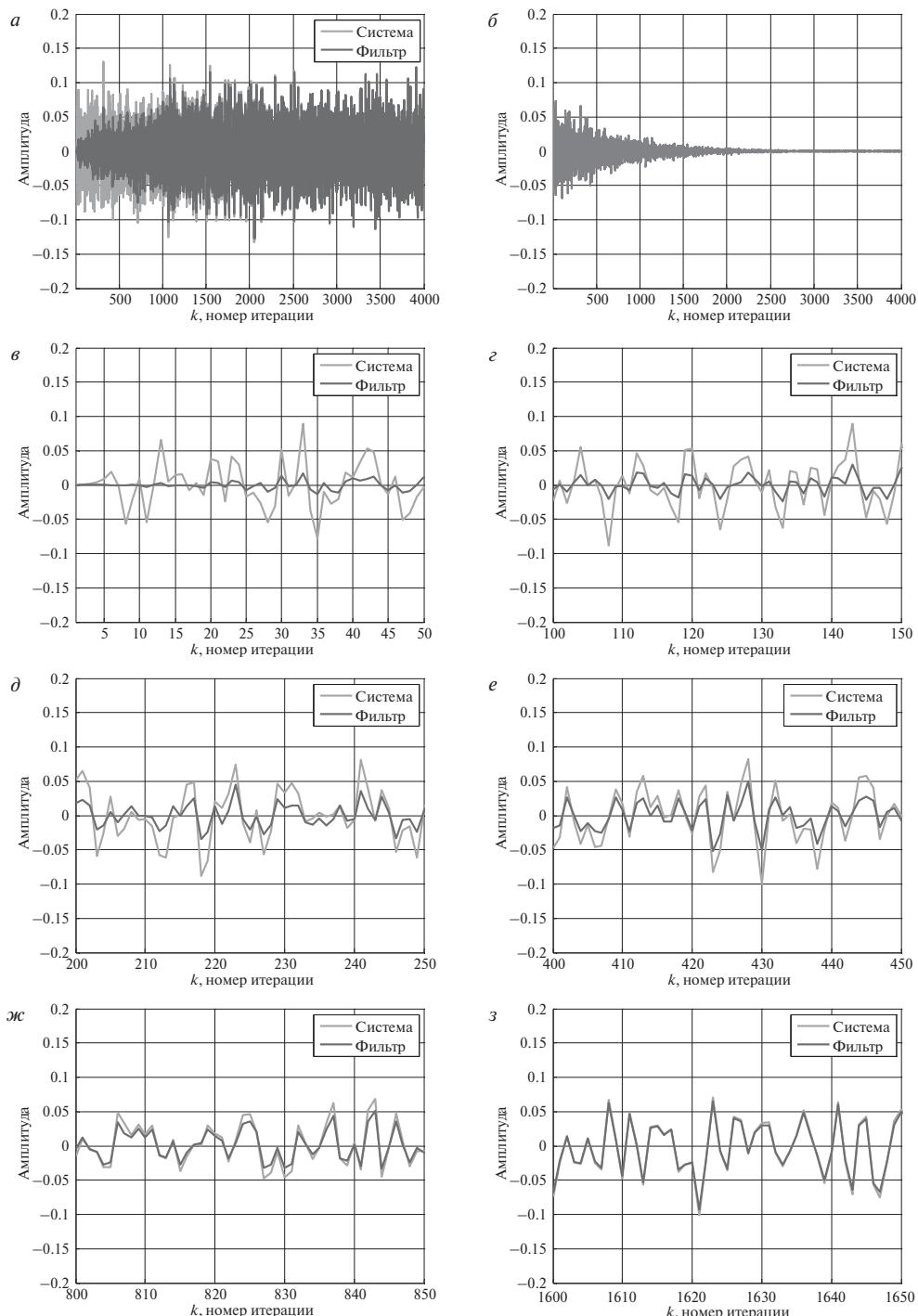


Рис. 1.6. Идентификация неизвестной линейной системы: а) — сигналы $d(k)$ и $y(k)$; б) — сигнал $\alpha(k)$; в) — з) сигналы $d(k)$ и $y(k)$ на разных интервалах в 50 отсчетов (итераций)

в предположении, что

$$\sum_{n=1}^N w_n^* x(k-n+1) = \sum_{n=1}^N h_n^*(k-1) x(k-n+1), \quad (1.7)$$

т. е. некоторая (как правило, начальная) часть ВК неизвестной линейной системы \mathbf{w}_∞ в точности идентифицируется адаптивным фильтром, или, другими словами, $\mathbf{h}_N(k-1) = \mathbf{w}_N$.



Рис. 1.7. Адаптивная идентификация неизвестной системы при наличии аддитивного шума

На рис. 1.8 представлены графики рассогласования («misalignment») — одного из показателей качества адаптивного фильтра, используемого в задачах идентификации, при $N_h \leq N_w$, где $N_w = 64$. Рассогласование — это евклидова мера расстояния между вектором ВК адаптивного фильтра $\mathbf{h}_{N_h}(k)$ и вектором коэффициентов $\mathbf{w}_{N_w} = [w_1, w_2, \dots, w_N]^T$, т. е.

$$\rho(k) = 20 \lg \frac{\|\mathbf{w}_{N_w} - \mathbf{h}_{N_h}(k)\|_2}{\|\mathbf{w}_{N_w}\|_2} = 20 \lg \frac{\sqrt{\sum_{n=1}^N [w_n - h_n(k)]^2}}{\sqrt{\sum_{n=1}^N w_n^2}}, \quad (1.8)$$

где символ $\|\bullet\|$ обозначает норму вектора. При $N_h < N_w$ вектор $\mathbf{h}_{N_h}(k)$ в уравнении (1.8) дополняется нулями от $N_h + 1$ до N_w . Из рис. 1.8а следует, что при $N_h < N_w$ длительность переходного процесса адаптивного фильтра в терминах $\rho(k)$ уменьшается, так как уменьшается «длина» фильтра (т. е. длина его импульсного отклика или число ВК). Однако при этом возникает шум (1.6), влияющий на качество алгоритма адаптивной фильтрации. В частности, этот шум уменьшает точность настройки ВК, характеризуемую рассогласованием (1.8).

Длительность переходного процесса адаптивного фильтра — это время, в течение которого наблюдаемый показатель качества претерпевает «заметные» изменения. За переходным процессом обычно следует установившийся режим, в течение которого наблюдаемый показатель качества не претерпевает «заметных» изменений.

В адаптивной фильтрации качество решения задачи идентификации также принято характеризовать коэффициентом ослабления эхо-сигнала (Echo Return Loss Enhancement, ERLE), который определяется как отношение энергии

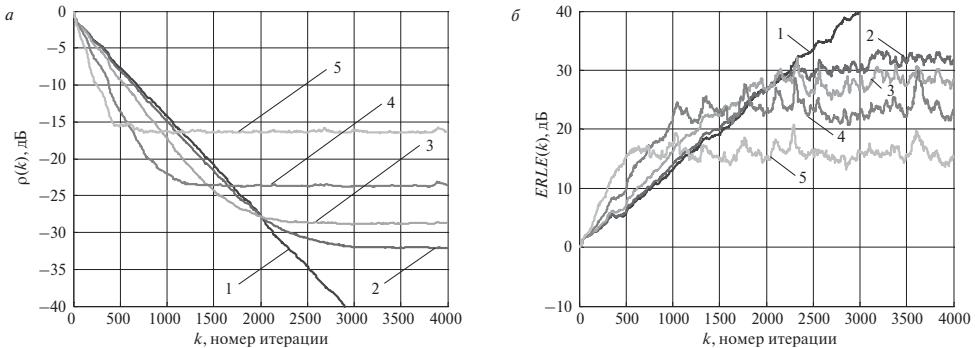


Рис. 1.8. Идентификация импульсного отклика: а) — рассогласование; б) — ERLE; 1 — $N_h=64$; 2 — $N_h=60$; 3 — $N_h=50$; 4 — $N_h=32$; 5 — $N_h=16$

требуемого сигнала $d(k)$ к энергии сигнала остаточных ошибок $\alpha(k)$, именуемого эхо-сигналом, т. е.

$$ERLE(k)=10 \lg \frac{E\{d^2(k)\}}{E\{\alpha^2(k)\}}. \quad (1.9)$$

Переходные процессы, аналогичные графикам на рис. 1.8а, в терминах ERLE представлены на рис. 1.8б. Этот рисунок также подтверждает ухудшение качества решения задачи идентификации при $N_h \leq N_w$, т. е. возрастание энергии ошибок $\alpha(k)$, что обусловлено наличием шума $e_o(k)$, влияющего на качество работы адаптивного фильтра. При решении задачи идентификации число ВК N_h по отношению к числу N_w выбирается, исходя из компромисса между требуемым качеством идентификации и вычислительными ресурсами, требуемыми для реализации адаптивного фильтра.

1.5. Компенсация эхо-сигналов

Ниже рассматривается пример идентификации неизвестного импульсного отклика в задачах эхо-компенсации. Необходимость решения этой задачи обусловлена архитектурой существующих систем проводной телефонной связи. Так исторически сложилось, что из-за экономии проводников двунаправленная связь между телефонным аппаратом абонента и коммутационным оборудованием телефонной станции осуществляется с помощью двухпроводного канала связи без усиления сигналов (рис. 1.9), так как этот участок телефонной сети во всем мире обычно не превышает нескольких километров и потери (ослабление) сигналов из-за сопротивления используемых проводников являются несущественными или допустимыми. Связь же между телефонными станциями осуществляется на значительно большие расстояния, а значит, сигналы подвер-

жены заметному ослаблению. Поэтому передаваемые между станциями сигналы необходимо усиливать, отчего связь на этом участке сети может быть только односторонней. Она осуществляется по двум односторонним двухпроводным каналам, т. е. по четырехпроводному каналу. Переход между двух- и четырехпроводными каналами осуществляется с помощью так называемых гибридных или дифференциальных схем («hybrid circuits»). Один из возможных вариантов такой схемы показан на рис. 1.10.

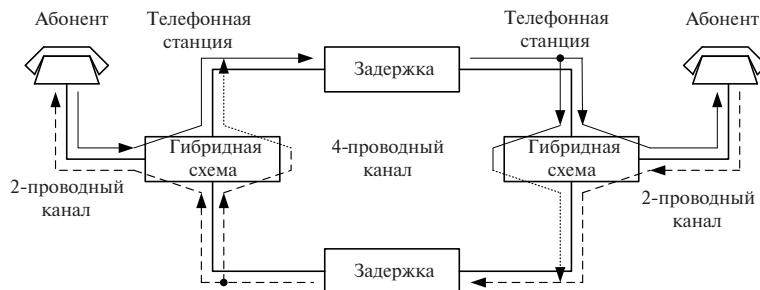


Рис. 1.9. Устройство телефонной связи между двумя абонентами

Гибридная схема — это в идеале сбалансированный мост. Однако этот мост невозможно сбалансировать во всем диапазоне частот обрабатываемых сигналов, так как проводные линии связи характеризуются частотнозависимым входным импедансом, в результате чего через гибридную схему (см. рис. 1.10) происходит утечка части сигнала от удаленного абонента $u_{AB}(t)$ обратно к удаленному абоненту, т. е. образуется эхо-сигнал $|u_{CD}(t)| < |u_{AB}(t)|$ [47]. Действительно, условие баланса рассматриваемой гибридной схемы определяется как $Z_i/(R_1+Z_i)=Z_b/(R_2+Z_b)$. Если $R_1=Z_i$, то $Z_b=R_2$. Но $R_1 \neq Z_i$, так как Z_i — это комплексный импеданс двухпроводной линии связи, нагруженной на развязывающий трансформатор. Импеданс Z_i зависит от топологии канала связи [48, 49]. Балансный импеданс Z_b — это в большинстве случаев просто резистор. Поэтому согласовать схему (см. рис. 1.10) во всей интересуемой полосе частот не представляется возможным. Если условие баланса не соблюдается, то образуется эхо-сигнал $u_{CD}(t) \neq 0$.

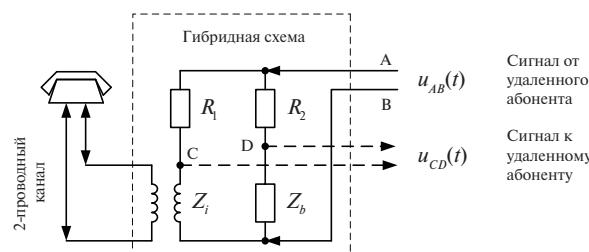


Рис. 1.10. Гибридная схема

Таким образом, передаваемый сигнал, будучи задержанным в силу конечной скорости распространения в проводниках, пройдя через гибридную схему, возвращается (также с некоторой задержкой) обратно к удаленному абоненту. Принятый удаленным абонентом собственный задержанный и ослабленный сигнал называется сигналом электрического эха. Для обеспечения комфортного разговора при связи эхо-сигнал необходимо подавлять, чем больше подавлять, чем больше задержка (рис. 1.11) [50].

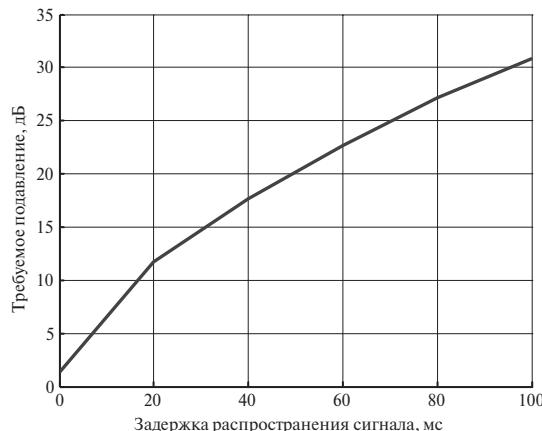


Рис. 1.11. Требуемое подавление эхо-сигнала

Известно, что гибридная схема моделируется линейным КИХ-фильтром [48, 49]. Поэтому задача подавления сигналов электрического эха формализуется как задача идентификации импульсного отклика этого фильтра w_∞ (рис. 1.12). Здесь на рисунке аналого-цифровые и цифро-аналоговые преобразователи (АЦП и ЦАП) для простоты не показаны.

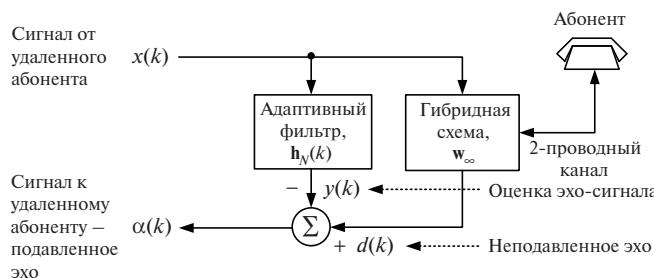


Рис. 1.12. Подавление сигналов электрического эха в телефонных сетях

Аналогично решается задача подавления сигналов электрического эха в модемах проводных систем связи, где также требуется разделять сигналы между

двух- и четырехпроводными каналами, образующими каналы приема и передачи данных (рис. 1.13).

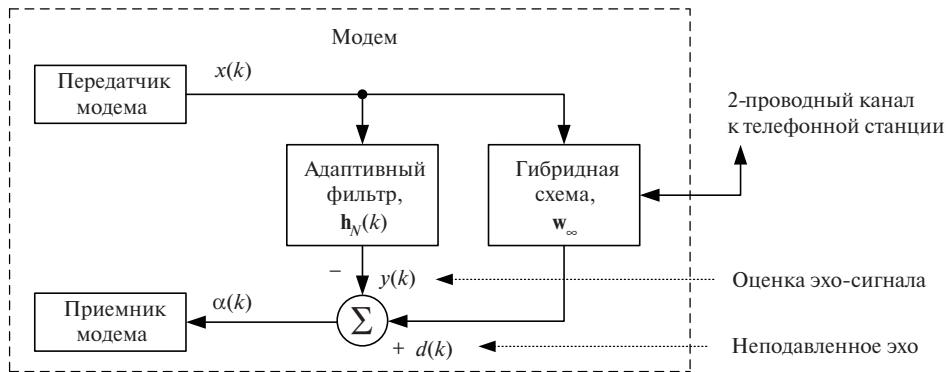


Рис. 1.13. Подавление сигналов электрического эха в модемах

Акустическое эхо имеет несколько иную природу. Оно образуется за счет отражения акустических сигналов, излучаемых, например, громкоговорителем, от стен помещения и приема этих отражений микрофоном. Громкоговоритель и микрофон могут быть элементами устройств озвучивания помещений или устройств громкоговорящей связи. Акустическая среда характеризуется акустическим импульсным откликом [51], поэтому принцип подавления сигналов акустического эха (рис. 1.14) такой же, как и принцип подавления сигналов электрического эха. Идентифицируемый импульсный отклик в данном случае

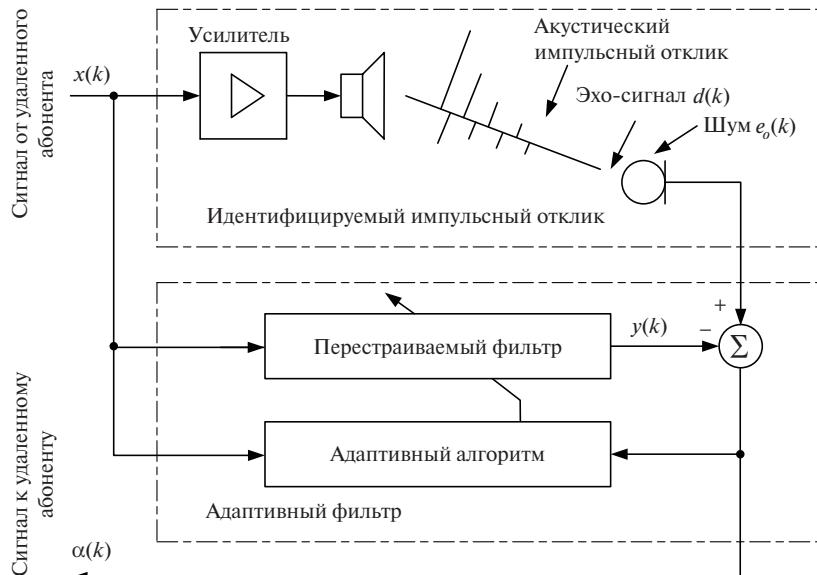


Рис. 1.14. Подавление сигналов акустического эха

образуется как свертка импульсных откликов усилителя, громкоговорителя, микрофона и акустического импульсного отклика помещения.

Рассмотрим эффективность применения адаптивной эхо-компенсации на примере компенсации сигналов электрического эха [47]. Стандарт [52] регламентирует численные модели ряда импульсных откликов, порождающих сигналы электрического эха. На рис. 1.15а показан один из таких откликов (такой же, как на рис. 1.5). Гибридная схема, характеризуемая этим импульсным откликом, подавляет сигнал $x(k)$ примерно на 7.6 dB. Это подавление грубо можно оценить по амплитудно-частотной характеристике (АЧХ) гибридной схемы (см. рис. 1.15б), которая «примерно одинаковая» (-7...-8 dB) в большей части полосы частот телефонного канала 0...4 кГц.

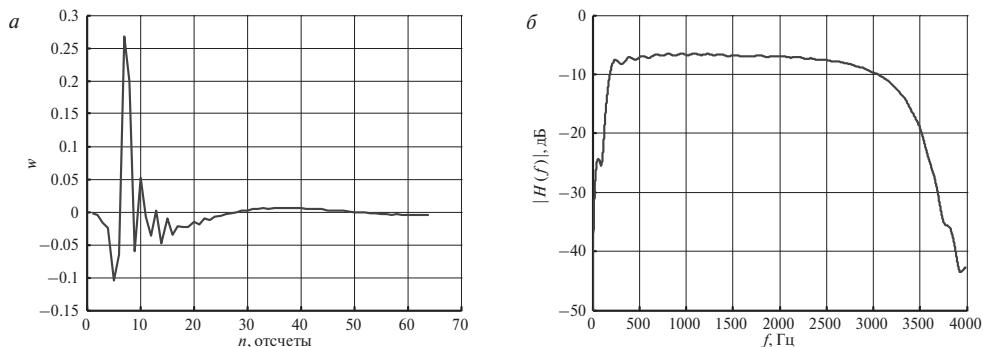


Рис. 1.15. $ERLE \approx 7.6$ dB: а) импульсный отклик гибридной схемы; б) АЧХ гибридной схемы

Ослабление эхо-сигналов в терминах $ERLE$ для гибридной схемы представляет собой отношение энергий входного сигнала и эхо-сигнала, т. е.

$$ERLE(k) = 10 \lg \frac{E\{x^2(k)\}}{E\{d^2(k)\}}. \quad (1.10)$$

На рис. 1.16 приведены результаты моделирования подавления эхо-сигнала, образованного гибридной схемой с характеристиками, такими же, как на рис. 1.15, с помощью адаптивного фильтра на базе простого нормализованного алгоритма по критерию наименьшего среднего квадрата (Normalized LMS, NLMS). Из рис. 1.16а видно, что эхо-сигнал имеет достаточно большую амплитуду (средняя кривая на рисунке). Это означает, что данный сигнал будет хорошо восприниматься человеческим ухом, создавая дискомфорт в разговоре, а значит, должен быть подавлен. Видно, что адаптивный фильтр подавляет эхо-сигнал (нижняя кривая). Аддитивный шум $e_o(k)$ в экспериментах составляет примерно -30 dB относительно среднего уровня эхо-сигналов.

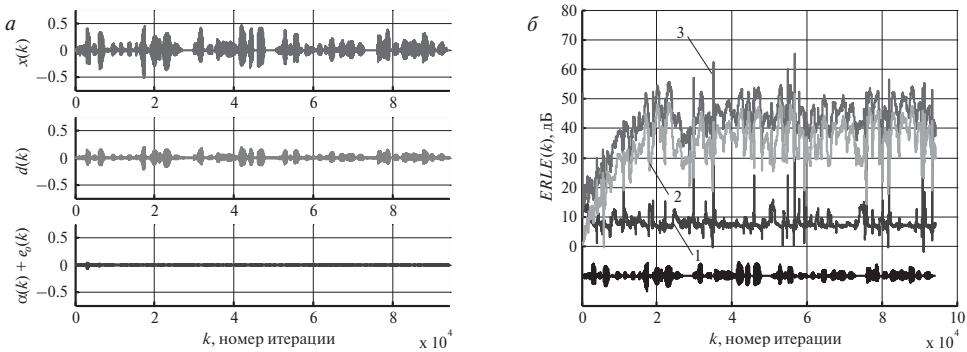


Рис. 1.16. NLMS-алгоритм: 1) ERLE гибридной схемы; 2) ERLE адаптивного фильтра; 3) ERLE — гибридной схемы и адаптивного фильтра

Параметр ERLE адаптивного фильтра для графиков (см. рис. 1.16б) определяется как

$$ERLE(k)=10 \lg \frac{E\{d^2(k)\}}{E\{\alpha^2(k)\}}, \quad (1.11)$$

а для гибридной схемы и адаптивного фильтра — как

$$ERLE(k)=10 \lg \frac{E\{x^2(k)\}}{E\{\alpha^2(k)\}}. \quad (1.12)$$

Энергия сигналов в расчетах (1.10)–(1.12) для данного эксперимента определялась на скользящем окне в 256 отсчетов, что при частоте дискретизации 8 кГц, принятой при обработке сигналов в телефонных каналах, равно интервалу усреднения $256/(8 \cdot 10^3)=32 \cdot 10^{-3}=32$ мс, т. е. условному интервалу стационарности речевого сигнала [53].

Результаты для RLS-алгоритма, аналогичные приведенным на рис. 1.16, представлены на рис. 1.17. Из сравнения рис. 1.16 и 1.17 видно, что RLS-алгоритм

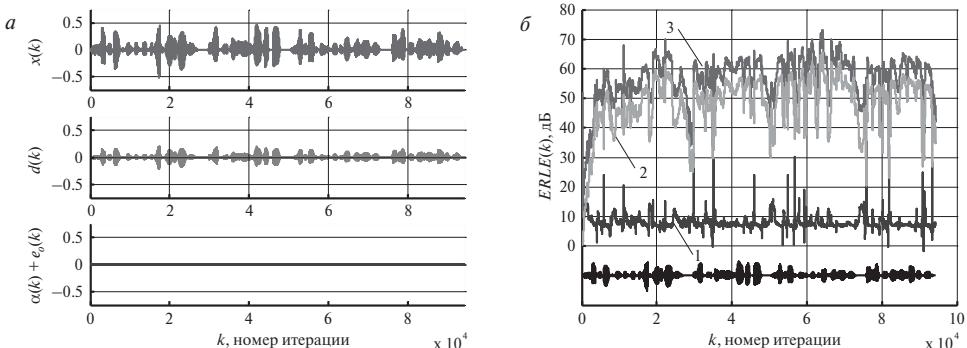


Рис. 1.17. RLS-алгоритм: 1) ERLE гибридной схемы; 2) ERLE адаптивного фильтра; 3) ERLE гибридной схемы и адаптивного фильтра

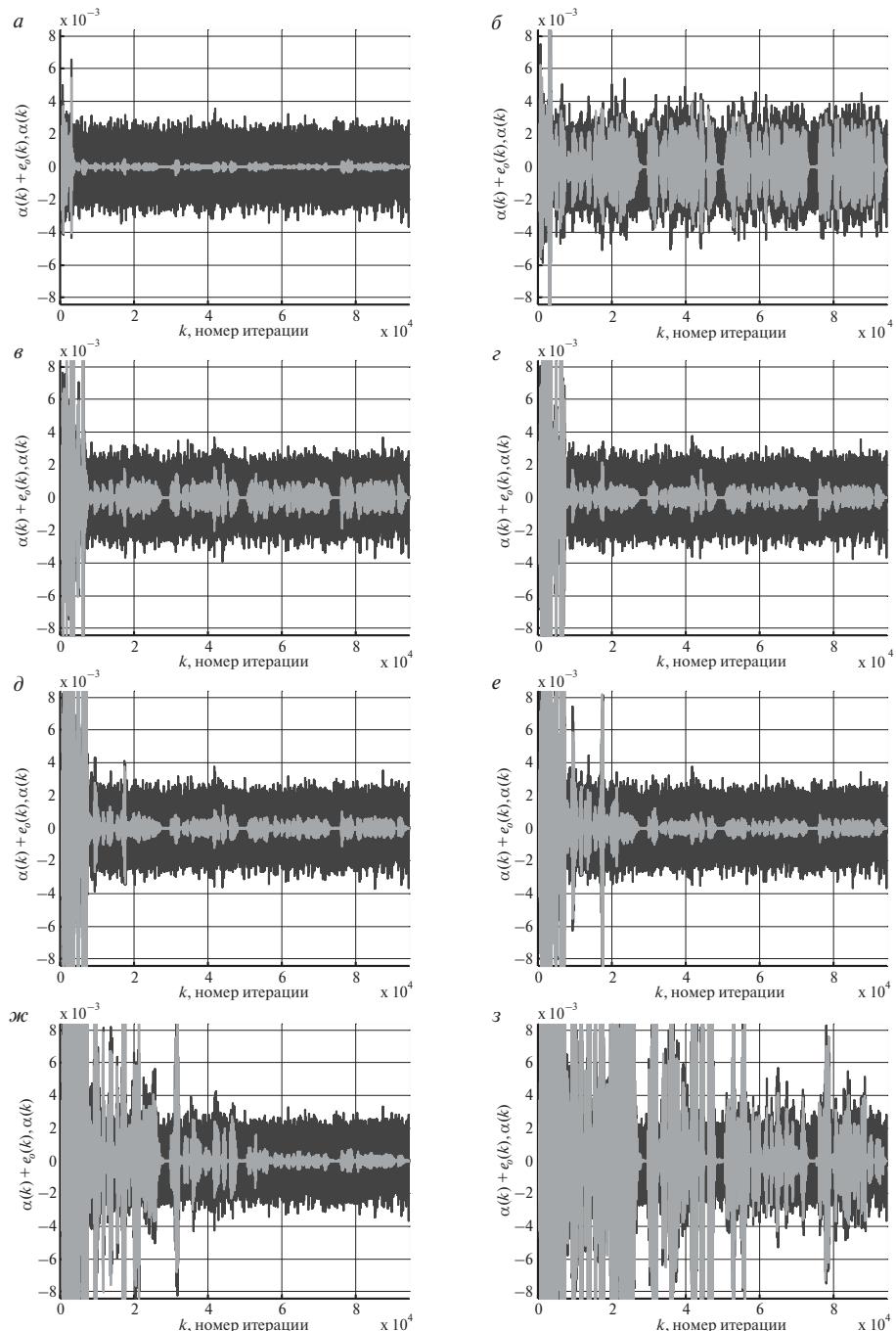


Рис. 1.18. Зашумленный остаточный эхо-сигнал $\alpha(k) + e_o(k)$ (темная кривая) и незашумленный остаточный эхо-сигнал $\alpha(k)$ (светлая кривая):
 а) RLS-алгоритм; б) NLMS-алгоритм, $\hat{\mu} = 0.1$; в) NLMS-алгоритм, $\hat{\mu} = 0.2$; г) NLMS-алгоритм, $\hat{\mu} = 0.1$; д) NLMS-алгоритм, $\hat{\mu} = 0.075$;
 е) NLMS-алгоритм, $\hat{\mu} = 0.05$; ж) NLMS-алгоритм, $\hat{\mu} = 0.025$; з) NLMS-алгоритм, $\hat{\mu} = 0.01$

более эффективен как в терминах достижимого ERLE, так и в терминах длительности переходных процессов с учетом того, что результаты для NLMS-алгоритма получены при значении масштабирующего множителя шага сходимости $\hat{\mu}$, обеспечивающего наибольшее значение параметра ERLE.

Из-за представления всех сигналов в одинаковом масштабе на рис. 1.16 и 1.17 визуально оценить уровни подавленных эхо-сигналов сложно. Графики подавленных эхо-сигналов в увеличенном масштабе приведены на рис. 1.18. Темным цветом показан сигнал $\alpha(k) + e_o(k)$, т. е. наблюдаемый сигнал на выходе ошибки адаптивного фильтра, а светлым — подавленное эхо $\alpha(k)$. Так как подавление эхо-сигнала таково, что уровень остаточного эха ниже уровня шума, то визуально оценить качество эхо-компенсации не представляется возможным. Однако при моделировании можно выделить только сигнал подавленного эха $\alpha(k)$, так как реализация шумового сигнала является известной.

Из рис. 1.18 видно, что вычислительно более сложный RLS-алгоритм является и более эффективным, чем менее сложный NLMS-алгоритм, т. е. несмотря на то что в обоих случаях уровень подавленного эхо-сигнала сравним или даже ниже уровня шума, человеческое ухо различает подавленный эхо-сигнал в случае NLMS-алгоритма и не различает — в случае RLS-алгоритма. Из рис. 1.18 также видно, что RLS-алгоритм обеспечивает демонстрируемое качество (подавление эхо-сигнала и длительность переходного процесса) без подбора параметров, в то время как качество NLMS-алгоритма зависит от параметра $\hat{\mu}$. При увеличении $\hat{\mu}$ длительность переходного процесса уменьшается, но увеличивается остаточная ошибка в установленном режиме, и наоборот.

Таким образом, рис. 1.4—1.18 иллюстрируют принципы решения прямой задачи адаптивной идентификации неизвестной системы и результаты ее использования для эхо-компенсации. Они также показывают большую функциональную эффективность сложных RLS-алгоритмов адаптивной фильтрации по сравнению с простыми градиентными алгоритмами.

1.6. Выравнивание характеристик электрических каналов связи

Другим примером использования адаптивных фильтров является задача обратной идентификации объекта. Решение этой задачи широко используется для выравнивания характеристик электрических каналов связи (рис. 1.19) с помощью адаптивных эквалайзеров [39, 40].

Необходимость выравнивания канала связи возникает из-за того, что при передаче цифровых данных по каналам с ограниченной полосой пропускания возникает явление, называемое межсимвольной интерференцией. Межсимволь-

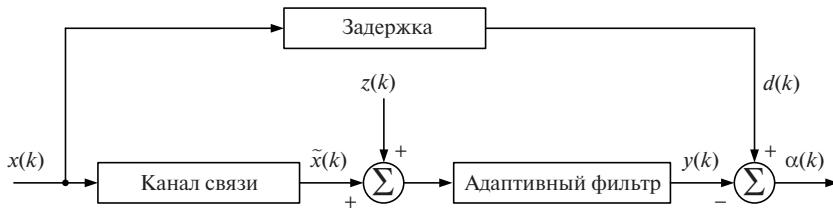


Рис. 1.19. Выравнивание АЧХ электрического канала связи

вольная интерференция, наряду с шумом канала $z(k)$ (рис. 1.19), оказывается на правильности решения о соответствии принятого информационного символа $\tilde{x}(k)$ символу $x(k)$, переданному по каналу связи. Если бы этот канал представлял собой просто линию задержки на время t_0 , то принимаемый сигнал (информационные символы) можно было бы легко оценивать (различать). Однако большинство каналов связи с математической точки зрения представляют собой КИХ-фильтр с импульсным откликом w (рис. 1.20а).

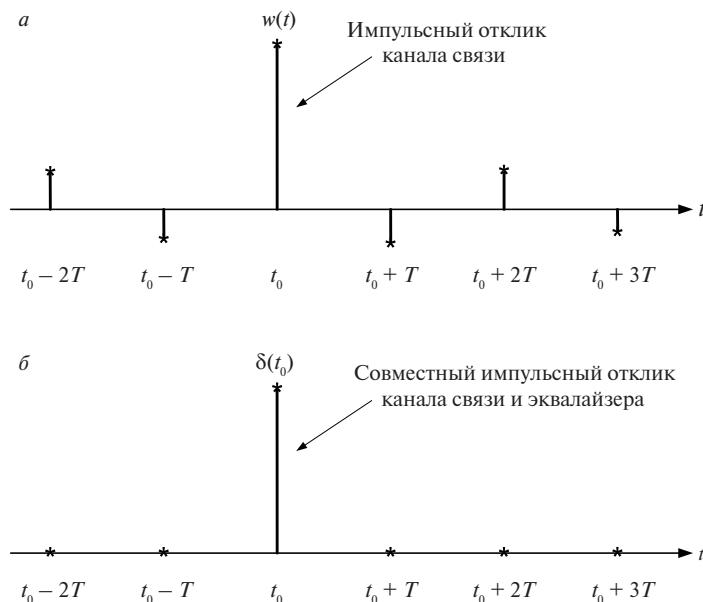


Рис. 1.20. Выравнивание электрического канала связи

Согласно рис. 1.20а в каждый отсчет времени k , совпадающий с моментом смены информационных символов, следующих с периодом T , на выходе канала образуется взвешенная сумма передаваемых подряд символов, т. е.

$$\tilde{x}(t_0 + kT) = w(t_0)x(t_0 + kT) + \sum_{n \neq k} w(t_0 + kT - nT)x(t_0 - nT). \quad (1.13)$$

В этой сумме только один символ несет текущую информацию, а остальные являются шумом (межсимвольной интерференцией), который не позволяет правильно различать символы на приемной стороне. Чтобы повысить вероятность правильного решения относительно принимаемых символов, используются эквалайзеры — устройства, которые уменьшают межсимвольную интерференцию.

Эквалайзер функционирует таким образом, что его АЧХ становится близкой к обратной АЧХ канала связи. Другими словами, совместная АЧХ канала связи и эквалайзера становится близкой к равномерной. Это означает, что свертка импульсных откликов канала связи w и эквалайзера h в моменты времени kT близка к дельта-функции (см. рис. 1.20б), в результате чего межсимвольная интерференция уменьшается. В этом случае выходной сигнал эквалайзера определяется как

$$y(t_0 + kT) = \delta(t_0)x(t_0 + kT) + 0, \quad (1.14)$$

т. е. информационный символ может быть правильно обнаружен (оценен и декодирован).

Если сравнить схемы на рис. 1.4 и 1.19, то можно заметить, что в последнем случае также решается задача идентификации линейной системы, определяющей задержку приема передаваемого сигнала. Известно, что импульсная характеристика линии задержки равна дельта-функции (т. е. одному ненулевому ВК), а АЧХ, как следствие, является равномерной. Идентификация импульсной характеристики линии задержки осуществляется с помощью каскадного соединения КИХ-фильтра с фиксированными ВК (т. е. канала связи) и КИХ-фильтра с изменямыми ВК (т. е. адаптивного фильтра). Это означает, что свертка фиксированного импульсного отклика канала связи с изменяемым импульсным откликом адаптивного фильтра в процессе адаптации приближается к дельта-функции, а АЧХ становится равномерной. Для обеспечения этого условия АЧХ адаптивного фильтра при адаптации приближается к обратной АЧХ канала связи.

Рис. 1.19 демонстрирует лишь принцип работы эквалайзера. На практике требуемый сигнал «не поступает» поциальному каналу связи. Он формируется на приемной стороне в течение протокола установления модемом связи, когда известная (тренировочная) последовательность символов передается модемом источника информации на противоположном конце канала связи. После настройки адаптивного фильтра эквалайзера, т. е. когда канал связи уже выровнен, вместо известной тренировочной последовательности передаются собственно неизвестные данные, а в качестве сигнала $d(k)$ используется сигнал $\hat{y}(k)$, т. е. выходной сигнал адаптивного фильтра $y(k)$, «квантованный» по критерию минимума расстояния до элементов созвездия алфавита передаваемых данных.

Существует два типа эквалайзеров (рис. 1.21): без обратной связи и с обратной связью [39, 40]. Адаптивный фильтр эквалайзера с обратной связью

можно рассматривать как двухканальный адаптивный фильтр. Эквалайзеры с обратной связью обеспечивают более качественное выравнивание характеристик канала, так как выходной сигнал такого эквалайзера после обучения частично формируется из уже ранее принятых «правильных» символов. Результаты моделирования, подтверждающие это высказывание, представлены на рис. 1.22—1.24.

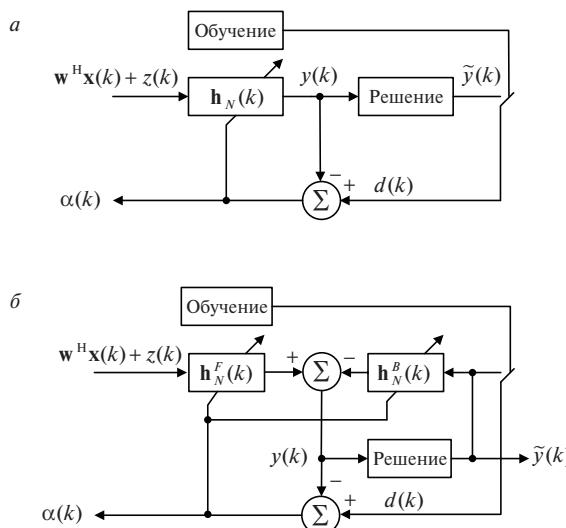


Рис. 1.21. Адаптивные эквалайзеры: а) без обратной связи; б) с обратной связью

Из рис. 1.22 и 1.23 следует, что эквалайзер без обратной связи выравнивает характеристики канала несколько хуже, чем эквалайзер с обратной связью. Это проявляется в том, что после адаптации свертка импульсных откликов канала связи и эквалайзера (см. рис. 1.23) в меньшей степени напоминает дельта-функцию, чем в случае, приведенном на рис. 1.22. Как следствие, провал в АЧХ канала связи (-60 дБ в рассматриваемом случае) выравнивается по-разному. Поэтому передаваемые информационные символы, в данном случае с квадратурной фазовой модуляцией (Phase Shift Keying, PSK-8), после выравнивания канала различаются также по-разному (см. рис. 1.24).

На рис. 1.24 показаны результаты выравнивания канала связи в терминах со-звездий наблюдаемых информационных символов. Здесь символы «Данные» — это созвездие символов источника информации, т. е. данных, передаваемых по каналу связи. Символы «Канал» — созвездие на выходе канала связи. Видно, что без принятия каких-то дополнительных мер эти данные невозможно различить. Такой «дополнительной мерой» является выравнивание канала связи. Из рис. 1.24 (см. созвездие, обозначенное как «Эквалайзер») следует очевидное превосходство эквалайзера с обратной связью перед эквалайзером без обратной связи.

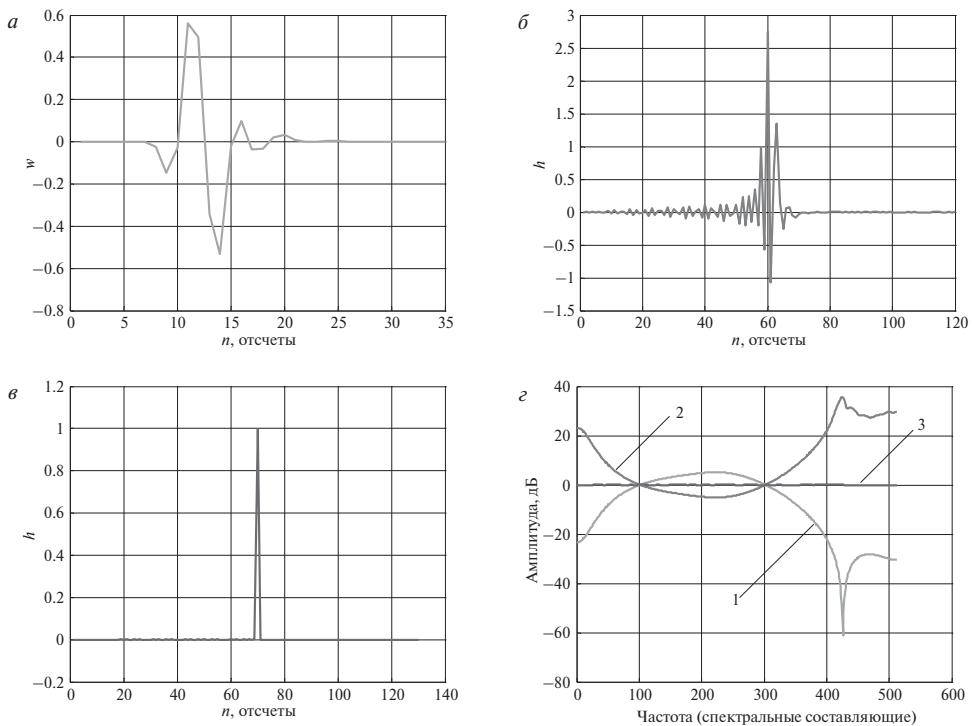


Рис. 1.22. Выравнивание канала, RLS-алгоритм, эквалайзер с обратной связью:
 а) импульсный отклик канала связи; б) импульсный отклик эквалайзера;
 в) свертка импульсных откликов канала связи и эквалайзера;
 в) АЧХ: 1) канала связи, 2) эквалайзера, 3) канала связи и эквалайзера

Если же сравнить поведение рассмотренных эквалайзеров с эквалайзерами на базе NLMS-алгоритма, то в обоих случаях результаты будут худшими. Эквалайзер на базе этого алгоритма требует в десятки раз превышающую по длительности тренировочную последовательность для обеспечения сходимости, а также некоторого искусства при выборе оптимального значения параметра μ . Это также свидетельствует о превосходстве адаптивных фильтров на базе вычислительно-сложных RLS-алгоритмов перед фильтрами на базе простых градиентных алгоритмов.

Следует также отметить, что для выравнивания каналов связи могут применяться и так называемые слепые эквалайзеры, т.е. не требующие тренировочных последовательностей. Один из возможных способов построения таких устройств основывается на использовании СМ-критерия [45] в случае, если передаваемые информационные символы удовлетворяют этому критерию. Адаптивные алгоритмы на основе СМ-критерия применяются для обработки PSK-сигналов (рис. 1.25), а также других сигналов, характеризуемых постоянным значением модуля информационных символов. Каждый СМ-символ a_i , представляющий собой комплексное число $\text{Re}\{a_i\} + j\text{Im}\{a_i\}$, обладает свойством $|a_i| = \sqrt{a_i^* a_i} = s = \text{const}$, где $i = 1, 2, \dots, I$, I — число символов алфавита (используется

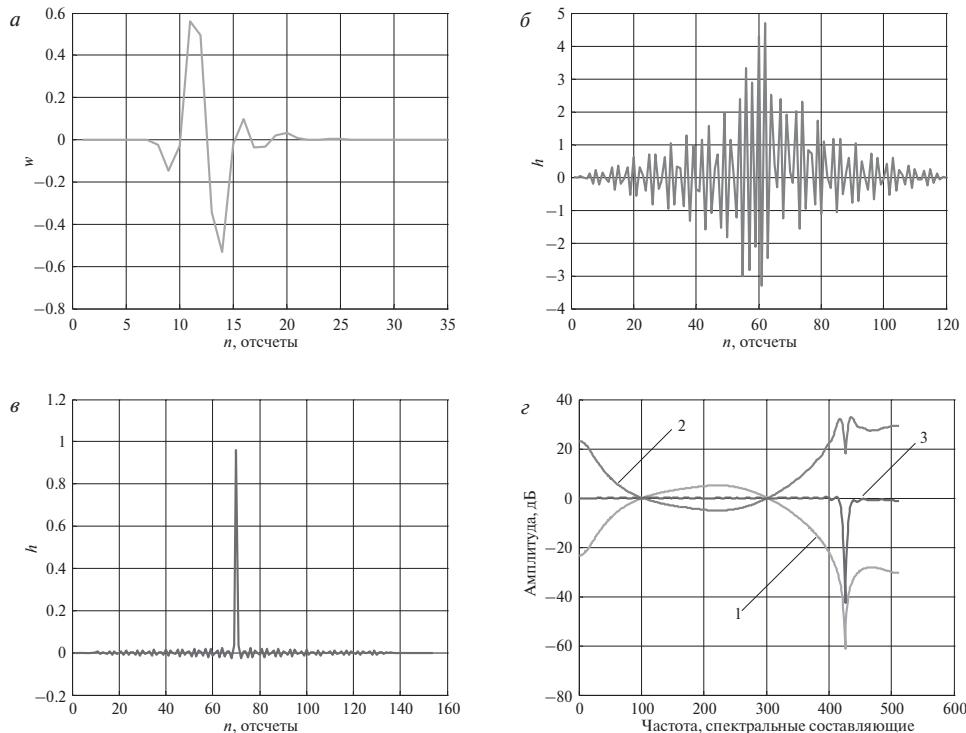


Рис. 1.23. Выравнивание канала, RLS-алгоритм, эквалайзер без обратной связи: а) импульсный отклик канала связи; б) импульсный отклик эквалайзера; в) свертка импульсных откликов канала связи и эквалайзера, в) АЧХ: 1) — канала связи, 2) — эквалайзера, 3) — канала связи и эквалайзера

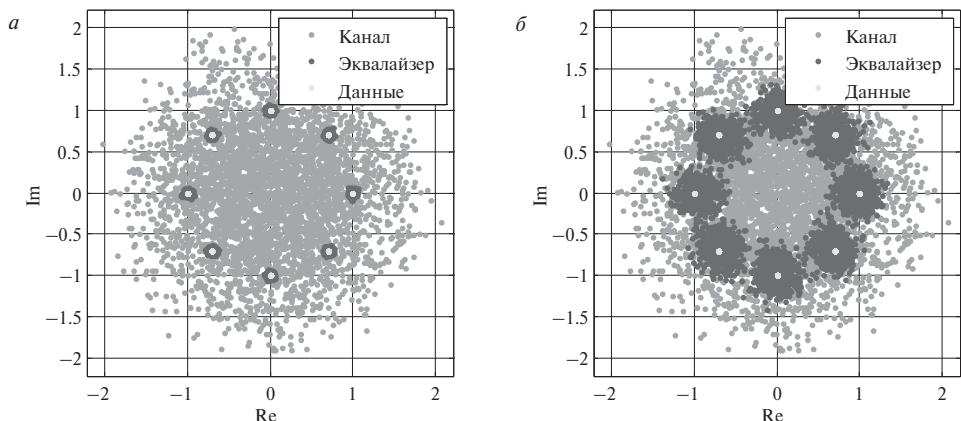


Рис. 1.24. Выравнивание канала, RLS-алгоритм: а) эквалайзер с обратной связью; б) эквалайзер без обратной связи

в обозначении PSK- I), а $j = \sqrt{-1}$ — мнимая единица. Значение модуля информационных символов s на приемной стороне является известным.

СМ-критерий для адаптивного фильтра формулируется как

$$E\left\{ \left| s^p - |y(k)|^q \right|^2 \right\} \rightarrow \min, \quad (1.15)$$

а соответствующие адаптивные алгоритмы обозначаются как СМ(p, q).

Целевая функция в уравнении (1.15) является многоэкстремальной, а потому использование простых градиентных алгоритмов для ее минимизации часто приводит к локальным решениям. Алгоритмы по критерию наименьших квадратов, например RLS-алгоритмы, напрямую не могут быть использованы в этом случае. В то же время согласно [54] при $p=q=2$ целевая функция в уравнении (1.15) может быть сведена к квадратичной функции, что позволяет для ее минимизации использовать незначительно модифицированные алгоритмы оптимизации квадратичных целевых функций.

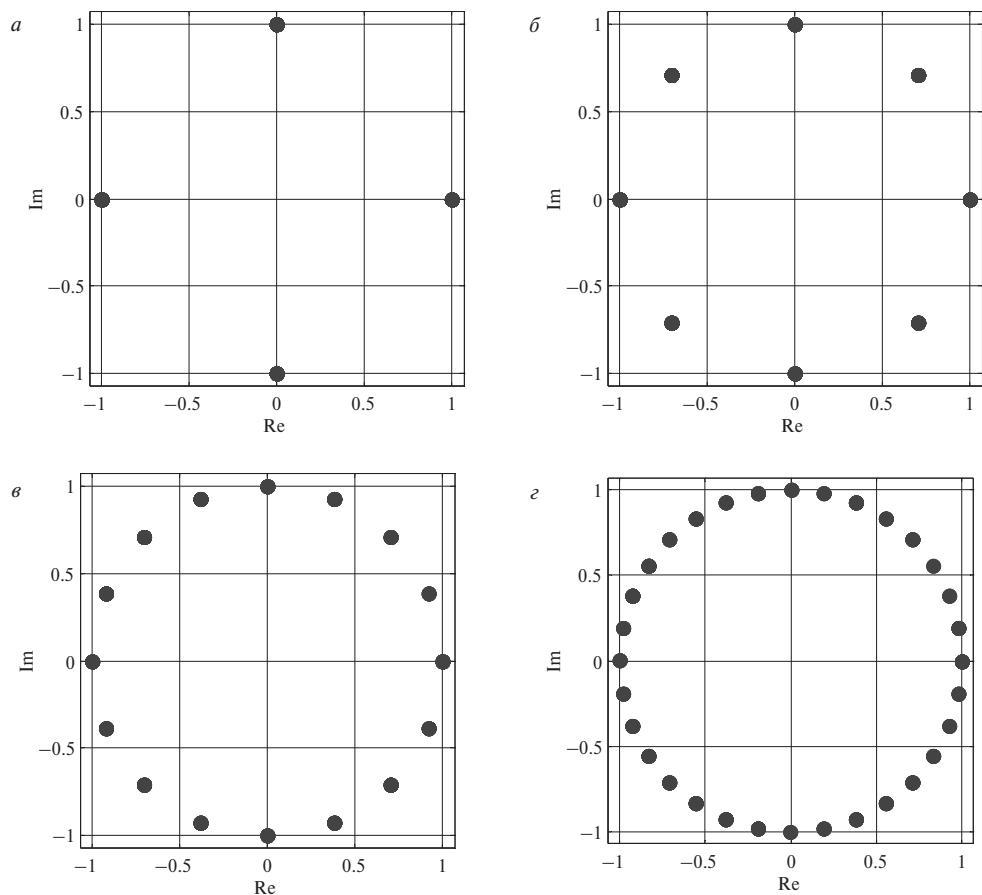


Рис. 1.25. Созвездия информационных символов: а) PSK-4; б) PSK-8; в) PSK-16; г) PSK-32

1.7. Адаптивные антенные решетки

Еще одним широко используемым приложением адаптивных фильтров являются ААР [1, 4–6, 23]. ААР (рис. 1.26) представляет собой многоканальный адаптивный фильтр (см. рис. 1.3) с одним ВК в канале. В общем случае в алгоритмах вычисления ВК для ААР используются входные сигналы $x_1(k), x_2(k), \dots, x_m(k), \dots, x_M(k)$, выходной $y(k)$ и требуемый $d(k)$ сигналы. В то же время в ААР могут также применяться алгоритмы, не требующие сигнала $d(k)$, если полезным принимаемым сигналом является СМ-сигнал или направление приема полезного сигнала известно. В последнем случае это направление может быть использовано в качестве характеристики полезного сигнала путем «введения в алгоритм» линейного ограничения, обеспечивающего необходимый уровень полезного сигнала на выходе ААР.

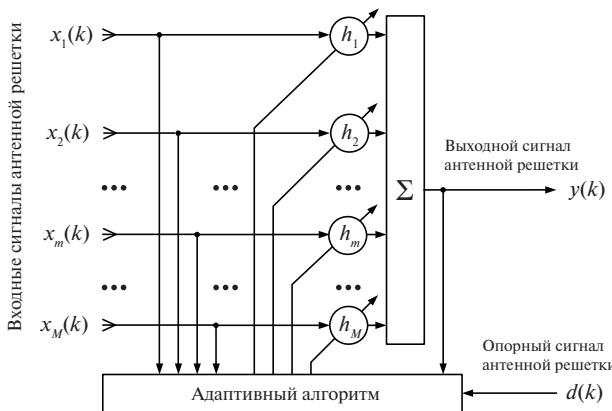


Рис. 1.26. Адаптивная антенная решетка

Для обеспечения цифровой обработки необходимо усиливать входные сигналы ААР до уровня, обеспечивающего преобразование аналогового сигнала в цифровой код, и понизить частоту этих сигналов до нулевой несущей частоты (до информационной полосы частот), с целью уменьшения требований к быстродействию устройств, обеспечивающих ЦОС. В [55] показано (см. также раздел 3.7), что обработка сигналов в информационной полосе частот эквивалентна формированию выходного сигнала на несущей частоте путем традиционного аналогового умножения и сложения.

Антеннная решетка характеризуется диаграммой направленности (ДН), т. е. амплитудно-угловой характеристикой $|F(\theta)|$, которая, подобно АЧХ обычного цифрового фильтра, с помощью ВК может менять свою форму, обеспечивая требуемые значения в направлениях на источники принимаемых сигналов, например, единичное в направлении на источник полезного сигнала и нулевое или

«очень маленькое» в направлениях на источники помех. ААР с комплексными ВК характеризуется наличием M степеней свободы. Она может принимать, например, один полезный сигнал и полностью подавлять $M - 1$ сигналов источников пространстворазнесенных помех.

На рис. 1.27 приведены примеры моделирования подавления помех с помощью 8-элементной эквидистантной линейной ААР ($M = 8$). Направления на источники принимаемых сигналов обозначены вертикальными стрелками в верхней части рис. 1.27а. Источник полезного сигнала располагался в направлении максимума основного лепестка ДН, а источники помех — в симметрич-

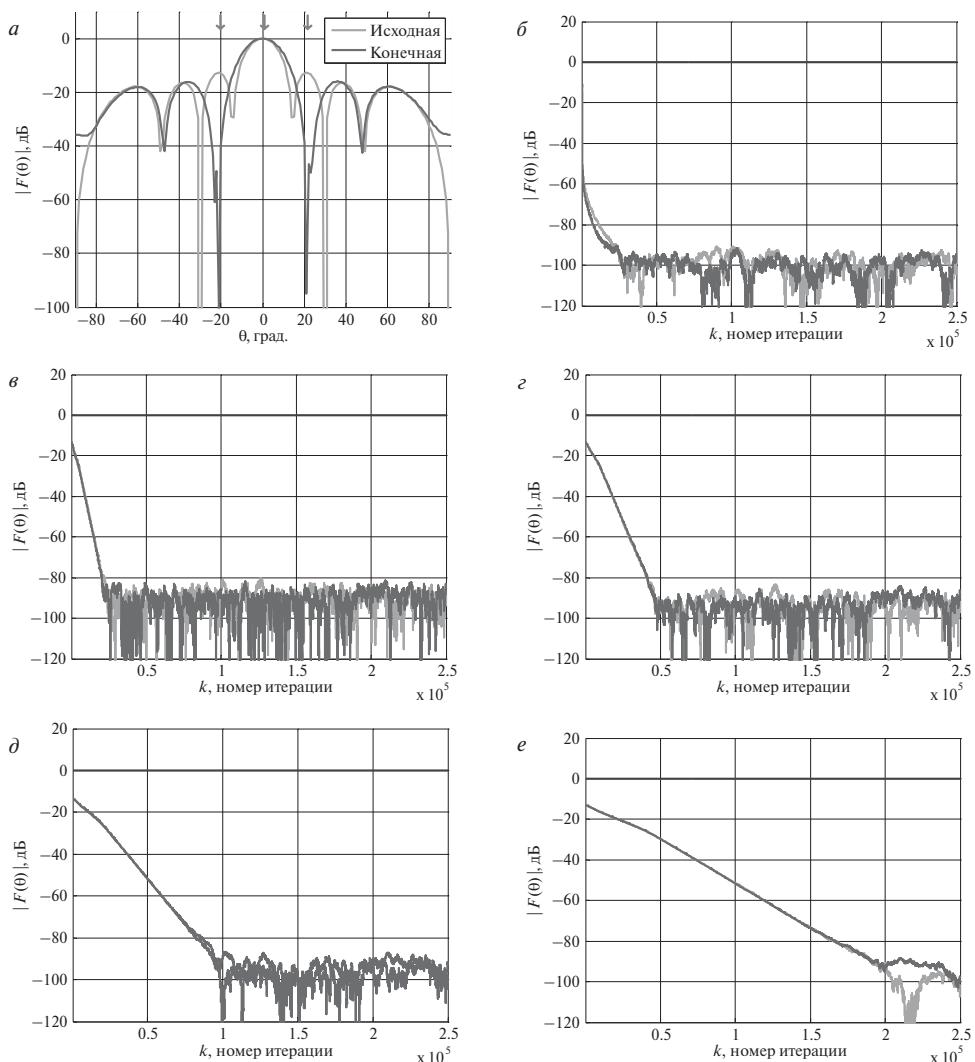


Рис. 1.27. Адаптивная антенная решетка: а) ДН; б) RLS-алгоритм; в) NLMS-алгоритм, $\hat{\mu} = 0.0008$; г) NLMS-алгоритм, $\hat{\mu} = 0.0004$; д) NLMS-алгоритм, $\hat{\mu} = 0.0002$; ж) NLMS-алгоритм, $\hat{\mu} = 0.0001$



ных направлениях максимумов двух первых боковых лепестков. Уровни этих лепестков примерно равны -13 дБ. Поэтому если любая из помех на 13 дБ превышает уровень полезного сигнала, то ее вклад в выходной сигнал ААР будет соизмеримым с полезным сигналом.

Из рис. 1.27а следует, что ААР в результате адаптации так меняет свои ВК, что в направлении источника полезного сигнала сохраняется заданный уровень основного лепестка ДН, равный 0 дБ в рассматриваемом случае, а в направлениях на источники помех в ДН образуются глубокие провалы. Это означает, что принимаемые из этих направлений помехи будут значительно ослаблены. В данном случае дополнительное ослабление помех за счет адаптивной пространственной фильтрации составляет примерно $-13 - (-100) = 87$ дБ. При этом изменения ДН во всех других направлениях не имеют принципиального значения, так как источники каких-либо сигналов в этих направлениях отсутствуют.

На остальных графиках (см. рис. 1.27) показано, как в процессе адаптации в направлениях на источники принимаемых сигналов меняется ДН, т. е. эти рисунки представляют собой переходной процесс ААР в терминах значений ее ДН. Из рис. 1.27б—рис. 1.27ж видно, что заданный уровень ДН (0 дБ) в направлении на источник полезного сигнала выдерживается на всех итерациях алгоритма (прямая линия в верхней части графиков), а длительность переходного процесса (в итерациях алгоритма) и достижимые уровни ДН в установившемся режиме в направлениях на источники помех (показаны темной и серой кривыми) меняются в зависимости от используемого алгоритма и его параметров. При $\hat{\mu} = 0.0001$ NLMS-алгоритм в установившемся режиме обеспечивает примерно такие же уровни ДН в направлениях на источники помех, как и RLS-алгоритм, но при этом длительность переходного процесса NLMS-алгоритма примерно в 10 раз превышает длительность переходного процесса RLS-алгоритма. При увеличении $\hat{\mu}$ длительность переходного процесса в NLMS-алгоритме уменьшается. Однако при этом также уменьшается и достигаемая глубина провалов ДН в направления на источники помех. Эти результаты в очередной раз свидетельствуют о превосходстве вычислительно-сложных адаптивных RLS-алгоритмов над простыми градиентными алгоритмами.

1.8. Шумоочистка сигналов

С помощью адаптивных фильтров может решаться и задача шумоочистки сигналов (компенсация шумов). В задаче шумоочистки (рис. 1.28) в качестве сигнала $d(k)$ используется зашумленный полезный сигнал $x(k) + z_1(k)$, а в качестве входного сигнала — сигнал $z_2(k)$, коррелированный с сигналом $z_1(k)$ и некоррелированный с $x(k)$. Адаптивный фильтр из сигнала $z_2(k)$ выделяет сигнал $z_1(k)$, т. е. $y(k) \approx z_1(k)$, поэтому сигнал ошибки $\alpha(k) = d(k) - y(k) \approx x(k)$.

Примером устройства (см. рис. 1.28) является двухмикрофонная система шумоочистки (рис. 1.29) [3], когда микрофон, формирующий сигнал $d(k)$, располагается рядом с говорящим (A), а микрофон, формирующий сигнал $z_2(k)$ (D), — рядом с источником шума, который нельзя физически устраниить, например, рядом с вентилятором (C) или двигателями автомобиля, самолета, вертолета. В этом случае адаптивный фильтр (E) моделирует акустическую среду от источника шума $z_2(k)$ до источника полезного сигнала $x(k)$ (рис. 1.30).

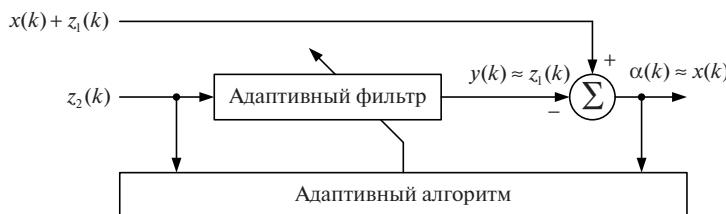


Рис. 1.28. Адаптивная шумоочистка сигналов: принцип

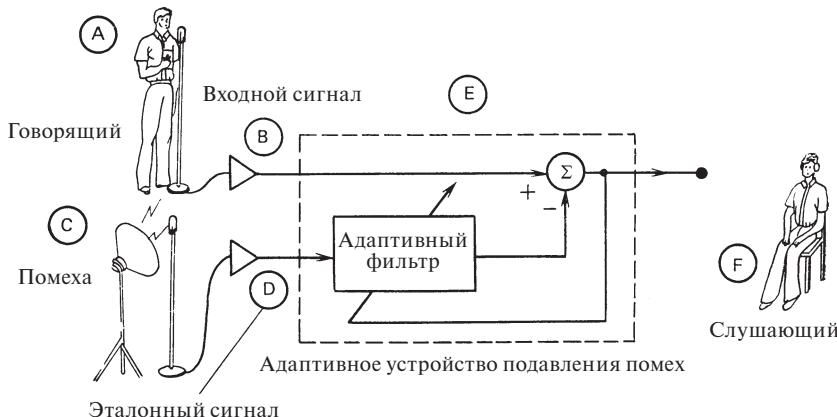


Рис. 1.29. Двухмикрофонная система шумоочистки

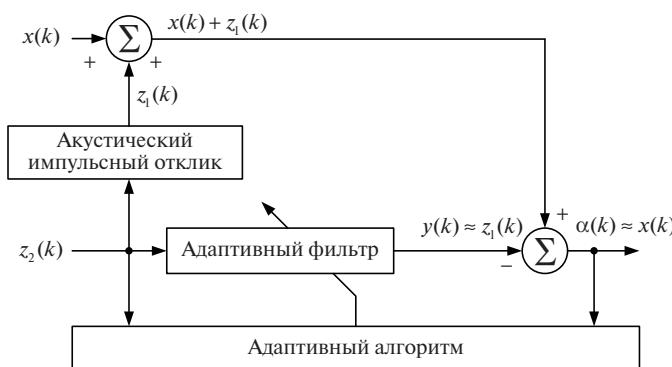


Рис. 1.30. Адаптивная шумоочистка сигналов: природа сигналов

Результаты моделирования шумоочистки (см. рис. 1.30) на базе адаптивного фильтра, использующего RLS-алгоритм, при $N_b=100$ и $N_w=100$ приведены на рис. 1.31. Рассматривалась очистка сигнала речи (см. рис. 1.31а) от шума двигателя вертолета (см. рис. 1.31б). Из сравнения рис. 1.31а и рис. 1.31в видно, что результат очистки зашумленной речи (см. рис. 1.31г) является достаточно хорошим, о чем свидетельствует также рис. 1.31д. Этот результат является следствием хорошей идентификации акустического импульсного отклика, формирующего

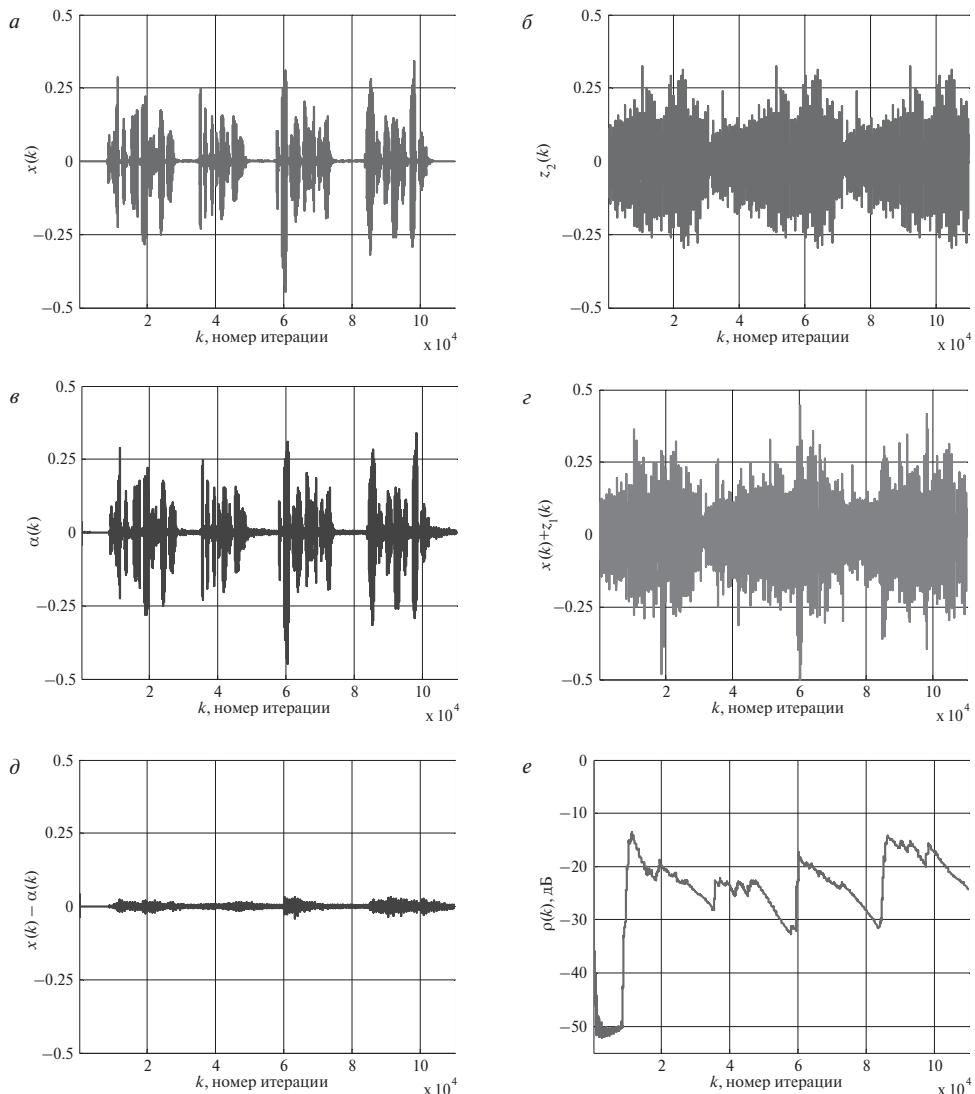


Рис. 1.31. Адаптивная шумоочистка, RLS-алгоритм: а) сигнал речи $x(k)$; б) сигнал источника шума $z_2(k)$; в) сигнал очищенной речи $\alpha(k)$; г) сигнал $d(k)=x(k)+z_1(k)$; д) сигнал $x(k)-\alpha(k)$; е) рассогласование

шум $z_1(k)$, что также подтверждается значениями параметра рассогласования (1.8), представленными на рис. 1.31е. В аналогичных условиях NLMS-алгоритм демонстрирует несколько худшие результаты (рис. 1.32). Похожие результаты также получаются при небольших изменениях N_h (в пределах $\pm 20\ldots 30$ ВК) относительно N_w , что свидетельствует о достаточно высокой эффективности адаптивной шумоочистки.

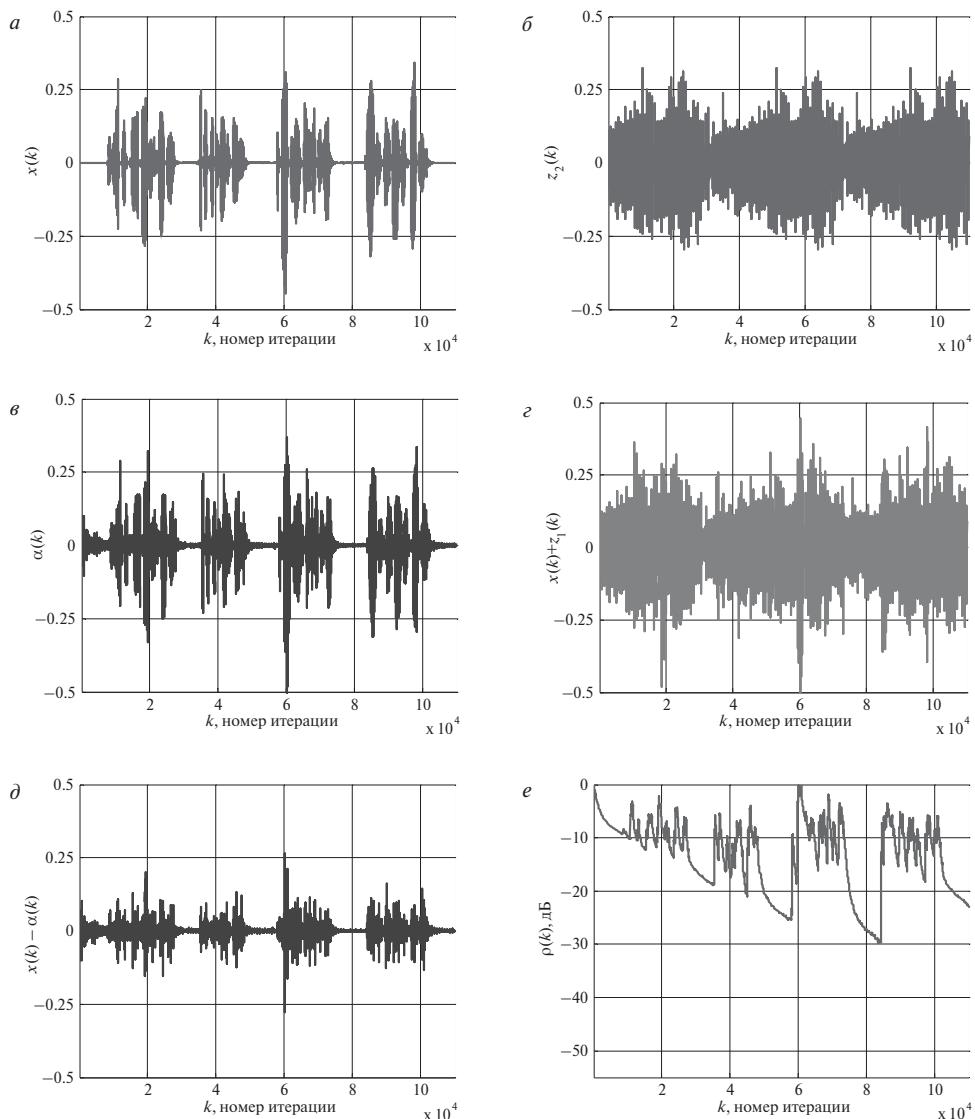


Рис. 1.32. Адаптивная шумоочистка, NLMS-алгоритм (при $\hat{\mu} = 0.01$): а) сигнал речи $x(k)$; б) сигнал источника шума $z_2(k)$; в) сигнал очищенной речи $\alpha(k)$; г) сигнал $d(k) = x(k) + z_1(k)$; е) рассогласование

1.9. Линейное предсказание сигналов

Еще одним приложением адаптивных фильтров является линейное предсказание наблюдаемых сигналов. При линейном предсказании в качестве требуемого сигнала адаптивного фильтра используется наблюдаемый сигнал, а в качестве входного сигнала — его задержанная копия. По окончании переходного процесса адаптивный фильтр представляет собой модель источника сигнала, а потому может быть использован для предсказания этого сигнала. Линейное предсказание, в частности, используется при кодировании речи (в вокодерах) [53], при построении быстрых адаптивных фильтров, рассматриваемых далее в главе 11, а также для выделения узкополосных сигналов $x(k)$ (спектральных линий, «line enhancement») на фоне широкополосных помех $z(k)$ (рис. 1.33).

В схеме (см. рис. 1.33) величина задержки D в отсчетах обрабатываемых сигналов выбирается такой, что $k_z < D < k_x$, где k_z — ширина основного лепестка автокорреляционной функции широкополосного сигнала $z(k)$ и k_x — ширина основного лепестка автокорреляционной функции узкополосного сигнала $x(k)$. Будучи задержанным на D отсчетов, сигнал $x(k - D)$ остается коррелированным с сигналом $x(k)$, потому в процессе адаптации на выходе адаптивного фильтра формируется сигнал, близкий к узкополосному сигналу $x(k)$, а на выходе сигнала ошибки — сигнал, близкий к широкополосному сигналу $z(k)$. В зависимости от того, какой из обрабатываемых сигналов является «полезным» — узкополосный или широкополосный, в качестве «полезного» выходного сигнала используется выходной сигнал адаптивного фильтра или сигнал ошибки.

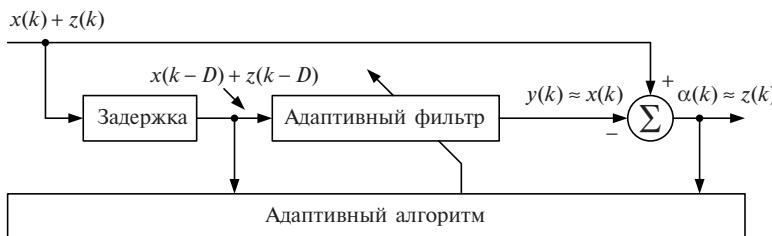


Рис. 1.33. Адаптивная фильтрация узкополосных сигналов

Среди других известных применений адаптивных фильтров можно отметить широкополосные акустические решетки, компенсаторы боковых лепестков направленных антенн, активные компенсаторы шумов, многоканальные эквалайзеры акустических каналов в аппаратуре высококачественного воспроизведения звука, одноканальные и многоканальные компенсаторы сигналов акустического эха в аппаратуре озвучивания помещений, полиномиальные (нелинейные) адаптивные фильтры и ряд других устройств. Их описание можно найти в современной научно-технической литературе.

1.10. Выводы по главе

В главе 1 дано общее представление об адаптивной обработке сигналов, о структурах адаптивных фильтров, целевых функциях и критериях, лежащих в основе построения адаптивных фильтров, а также о некоторых показателях качества, используемых в адаптивной фильтрации. Рассмотренные примеры известных приложений адаптивных фильтров демонстрируют зависимость качества функционирования адаптивных фильтров от вида используемого алгоритма вычисления ВК и его параметров. Причины разной эффективности адаптивных фильтров на базе разных алгоритмов объясняются в следующих главах настоящей книги.

ГЛАВА 2

ОПЕРАТОР КОМПЛЕКСНОГО ГРАДИЕНТА И ЕГО ПРИМЕНЕНИЕ В ТЕОРИИ АДАПТИВНОЙ ФИЛЬТРАЦИИ СИГНАЛОВ

В настоящей главе приводится определение оператора комплексного градиента и рассматриваются его свойства. Анализируются результаты применения этого оператора к действительным и комплексным целевым функциям. Эти результаты также сравниваются с результатами применения оператора действительного градиента к действительным целевым функциям.

2.1. Введение

В главе 1 было дано объяснение целесообразности изложения теории адаптивной фильтрации сигналов применительно к фильтрам не с действительными ВК, как в большинстве литературных источников, а с комплексными ВК. Одним из важных моментов этой теории является определение комплексного градиента, используемого при получении математических соотношений для адаптивных алгоритмов и анализа адаптивных фильтров, обрабатывающих комплексные сигналы. Поэтому в настоящей главе рассматриваются вопросы, относящиеся к определению и свойствам оператора комплексного градиента. Эти вопросы носят фундаментальный характер и являются полезными с точки зрения дальнейшего понимания материалов, представленных в настоящей книге.

Исторически развитие теории адаптивных фильтров с комплексными ВК происходило вслед за развитием теории адаптивных фильтров с действительными ВК. Так LMS-алгоритм [56] изначально был разработан для адаптивных фильтров с действительными ВК. Однако вскоре была получена версия этого алгоритма для адаптивных фильтров с комплексными ВК [57]. Аналогичным образом развивался и линейно-ограниченный (Linearly Constrained, LC) LMS-алгоритм адаптивной фильтрации сигналов [58, 59]. Необходимость разработки комплексных алгоритмов была связана с их использованием в ААР [1, 4–6, 23, 60, 61] и в адаптивных эквалайзерах [39, 40, 62, 63]. Комплексные алгоритмы

также применяются при реализации адаптивных фильтров в частотной области [64, 65]. Сегодня существуют комплексные версии не только простейших LMS-алгоритмов, но и более сложных RLS-алгоритмов [66–68].

К сожалению, в большинстве литературных источников алгоритмы адаптивной фильтрации рассматриваются в основном применительно к фильтрам с действительными ВК. В алгоритмах вычисления ВК этих фильтров все переменные являются действительными. Однако в алгоритмах адаптивной фильтрации комплексных сигналов содержатся переменные, которые могут быть действительными или комплексными, а также в ряде формул могут принимать комплексно-сопряженные значения по отношению к одноименным переменным в других формулах. По этой причине формальный переход от математического описания алгоритмов для адаптивных фильтров с действительными ВК к описанию алгоритмов для фильтров с комплексными ВК не всегда является очевидным. Данное обстоятельство подчеркивает важность умения разрабатывать и анализировать адаптивные фильтры с комплексными ВК. Это важно и потому, что используемое в ряде литературных источников математическое описание адаптивных фильтров с N комплексными ВК в виде фильтров с $2N$ действительными ВК является не только громоздким, но и часто содержит систематическую погрешность в виде множителя, равного 2, который хоть и не влияет на характеристики полученных алгоритмов, но все же не вполне корректен с математической точки зрения.

В связи с этим в настоящей главе рассматриваются основы дифференцирования функции комплексного переменного, дается определение комплексного градиента, а также приводятся некоторые примеры его применения к целевым функциям, используемым в теории адаптивной фильтрации [22, 30, 69, 70].

2.2. Несколько простых примеров дифференцирования функций действительных и комплексных переменных

Поле комплексных чисел является естественной средой описания сигналов, содержащих информацию не только в амплитуде колебания несущей, но и в фазе. Такие сигналы используются, например, при передаче данных в современных цифровых системах связи. Это сигналы с квадратурной амплитудной, фазовой и дифференциальной фазовой модуляциями: Quadrature Amplitude Modulation (QAM), PSK и Differential PSK (DPSK), где каждый информационный символ представляется в виде двух составляющих, которые можно интерпретировать как комплексное число [21]. В случае QAM эти числа характеризуются различными значениями амплитуды и фазы, а в случае PSK (см. рис. 1.25) и DPSK —

постоянными значениями амплитуды и различными значениями фазы, равномерно распределенными в пределах углов $0\dots 2\pi$.

Одной из задач, решаемых в теории адаптивной фильтрации, является минимизация целевой функции — действительной неотрицательной многомерной функции, аргументы которой (ВК адаптивного фильтра) могут быть действительными или комплексными. В комплексных адаптивных фильтрах каждый ВК представляется парой действительных чисел как $h=h_R+jh_I$, где h_R и h_I — действительная и мнимая составляющие ВК. Обрабатываемые сигналы представляются аналогичным образом как $x=x_R+jx_I$. При таком представлении ВК и сигналов анализ адаптивного фильтра может проводиться в поле действительных чисел. Однако этот анализ достаточно громоздкий. Поэтому поле комплексных чисел является более удобным для такого анализа, что иллюстрируется на примере ниже.

Определим целевую функцию как квадрат модуля ошибки между требуемым d и выходным y сигналами комплексного адаптивного фильтра в дискретный момент времени как

$$|\alpha|^2 = \alpha\alpha^* = (d - y)(d^* - y^*) = (d - \mathbf{h}_N^H \mathbf{x}_N)(d^* - \mathbf{x}_N^H \mathbf{h}_N) = (d - \mathbf{h}_{N,R}^T \mathbf{x}_N + j \mathbf{h}_{N,I}^T \mathbf{x}_N)(d^* - \mathbf{x}_N^H \mathbf{h}_{N,R} - j \mathbf{x}_N^H \mathbf{h}_{N,I}), \quad (2.1)$$

где $\mathbf{h}_N = [h_1, h_2, \dots, h_N]^T = \mathbf{h}_{N,R} + j \mathbf{h}_{N,I} = [h_{1,R}, h_{2,R}, \dots, h_{N,R}]^T + j [h_{1,I}, h_{2,I}, \dots, h_{N,I}]^T$ — вектор комплексных ВК, $\mathbf{x}_N = [x_1, x_2, \dots, x_N]^T$ — вектор комплексных отсчетов входных сигналов адаптивного фильтра. Здесь в обозначениях сигналов и ВК индекс дискретного времени k для простоты записи опущен.

Определим оператор комплексного градиента как

$$\nabla_{\mathbf{h}_N} = \left[\frac{\partial}{\partial h_1}, \frac{\partial}{\partial h_2}, \dots, \frac{\partial}{\partial h_N} \right]^T \quad (2.2)$$

или

$$\nabla_{\mathbf{h}_N^*} = \left[\frac{\partial}{\partial h_1^*}, \frac{\partial}{\partial h_2^*}, \dots, \frac{\partial}{\partial h_N^*} \right]^T. \quad (2.3)$$

Тогда

$$\nabla_{\mathbf{h}_{N,R}} \{ \alpha\alpha^* \} = \alpha \left(\nabla_{\mathbf{h}_{N,R}} \alpha^* \right) + \left(\nabla_{\mathbf{h}_{N,R}} \alpha \right) \alpha^* = \alpha (-\mathbf{x}_N^*) + (-\mathbf{x}_N) \alpha^*, \quad (2.4)$$

$$\nabla_{\mathbf{h}_{N,I}} \{ \alpha\alpha^* \} = \alpha \left(\nabla_{\mathbf{h}_{N,I}} \alpha^* \right) + \left(\nabla_{\mathbf{h}_{N,I}} \alpha \right) \alpha^* = \alpha (-j\mathbf{x}_N^*) + (j\mathbf{x}_N) \alpha^*. \quad (2.5)$$

Если предположить, что комплексный градиент определяется как это показано в работе [57]

$$\nabla_{\mathbf{h}_N^*} = \nabla_{\mathbf{h}_{N,R}} + j \nabla_{\mathbf{h}_{N,I}} = \left[\frac{\partial}{\partial h_{1,R}}, \frac{\partial}{\partial h_{2,R}}, \dots, \frac{\partial}{\partial h_{N,R}} \right]^T + j \left[\frac{\partial}{\partial h_{1,I}}, \frac{\partial}{\partial h_{2,I}}, \dots, \frac{\partial}{\partial h_{N,I}} \right]^T, \quad (2.6)$$

то

$$\begin{aligned}\nabla_{\mathbf{h}_N^*} \{ \alpha \alpha^* \} &= \nabla_{\mathbf{h}_{N,R}} \{ \alpha \alpha^* \} + j \nabla_{\mathbf{h}_{N,I}} \{ \alpha \alpha^* \} = \\ &= \alpha (-\mathbf{x}_N^*) + (-\mathbf{x}_N) \alpha^* + j [\alpha (-j \mathbf{x}_N^*) + (j \mathbf{x}_N) \alpha^*] = -2 \mathbf{x}_N \alpha^*.\end{aligned}\quad (2.7)$$

Выражение (2.7) без учета операции комплексного сопряжения переменной α совпадает с выражением градиента для фильтров с действительными ВК $\tilde{\mathbf{h}}_N = [\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_N]^T$, т. е. содержит множитель 2, см. ниже

$$\nabla_{\tilde{\mathbf{h}}_N} \{ \alpha^2 \} = 2\alpha \nabla_{\tilde{\mathbf{h}}_N} \alpha = 2\alpha \nabla_{\tilde{\mathbf{h}}_N} \{ d - \tilde{\mathbf{h}}_N^T \mathbf{x}_N \} = -2 \mathbf{x}_N \alpha. \quad (2.8)$$

Здесь и далее для различия действительных и комплексных ВК адаптивного фильтра будем использовать соответственно обозначения \tilde{h} и h , т. е. действительные ВК будут отмечаться символом «~» («тильда»).

Однако, как будет показано далее, **правильным** выражением для комплексного градиента является не выражение (2.7), а выражение

$$\nabla_{\mathbf{h}_N^*} \{ \alpha \alpha^* \} = -\mathbf{x}_N \alpha^*. \quad (2.9)$$

Аналогично, градиент функции $\tilde{\mathbf{h}}_N^T \mathbf{R}_N \tilde{\mathbf{h}}_N$ определяется как

$$\nabla_{\tilde{\mathbf{h}}_N} \{ \tilde{\mathbf{h}}_N^T \mathbf{R}_N \tilde{\mathbf{h}}_N \} = 2 \mathbf{R}_N \tilde{\mathbf{h}}_N \quad (2.10)$$

для действительных ВК. В случае же комплексных ВК согласно [57] градиент функции $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$ определяется как

$$\nabla_{\mathbf{h}_N^*} \{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \} = \nabla_{\mathbf{h}_{N,R}} \{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \} + j \nabla_{\mathbf{h}_{N,I}} \{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \} = 2 \mathbf{R}_N \mathbf{h}_N. \quad (2.11)$$

Однако и этот результат **неправильный**, так как правильным является выражение

$$\nabla_{\mathbf{h}_N^*} \{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \} = \mathbf{R}_N \mathbf{h}_N. \quad (2.12)$$

При дифференцировании рассмотренных функций необходимо иметь в виду, что используемые в них произведения векторов и матриц выражаются как

$$\tilde{\mathbf{h}}_N^T \mathbf{x}_N = \mathbf{x}_N^T \tilde{\mathbf{h}}_N = \sum_{n=1}^N \tilde{h}_n x_n, \quad (2.13)$$

$$\mathbf{h}_N^H \mathbf{x}_N = \sum_{n=1}^N h_n^* x_n, \quad (2.14)$$

$$\mathbf{x}_N^H \mathbf{h}_N = \sum_{n=1}^N x_n^* h_n, \quad (2.15)$$

$$\tilde{\mathbf{h}}_N^T \mathbf{R}_N \tilde{\mathbf{h}}_N = \sum_{m=1}^N \sum_{n=1}^N \tilde{h}_m R_{mn} \tilde{h}_n, \quad (2.16)$$

$$\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N = \sum_{m=1}^N \sum_{n=1}^N h_m^* R_{mn} h_n, \quad (2.17)$$

где \tilde{h}_n — элементы вектора $\tilde{\mathbf{h}}_N$, h_n — элементы вектора \mathbf{h}_N , x_n — элементы вектора \mathbf{x}_N , а R_{mn} — элементы матрицы \mathbf{R}_N .

Также необходимо учитывать следующие правила дифференцирования скалярных переменных:

$$\frac{\partial h_m}{\partial h_n} = \delta_{mn}, \quad (2.18)$$

$$\frac{\partial h_m}{\partial h_n^*} = 0, \quad (2.19)$$

$$\frac{\partial h_m^*}{\partial h_n} = \delta_{mn}, \quad (2.20)$$

$$\frac{\partial h_m^*}{\partial h_n} = 0, \quad (2.21)$$

$$\frac{\partial h_{m,R}}{\partial h_{n,R}} = \delta_{mn}, \quad (2.22)$$

$$\frac{\partial h_{m,R}}{\partial h_{n,I}} = 0, \quad (2.23)$$

$$\frac{\partial h_{m,I}}{\partial h_{n,I}} = \delta_{mn}, \quad (2.24)$$

$$\frac{\partial h_{m,I}}{\partial h_{n,R}} = 0, \quad (2.25)$$

где $\delta_{nm} = \begin{cases} 1, & \text{при } m=n \\ 0, & \text{при } m \neq n \end{cases}$ — дельта-функция.

Возникают закономерные вопросы. Какие из результатов (2.7)–(2.12) верны на самом деле? Почему исчезает множитель 2 в уравнениях (2.9) и (2.12)? То есть насколько верны обозначенные выше правила дифференцирования (2.18)–(2.25) и как оператор $\partial/\partial h$ воздействует на h^* ?

Ответы на эти вопросы находятся в теории функции комплексного переменного. Некоторые элементы этой теории [69, 70] рассматриваются ниже.

2.3. Дифференцирование комплексных и действительных функций комплексного переменного

Рассмотрим комплексную функцию $g(h_1, h_2)$ двух комплексных переменных h_1 и h_2 , каждая из которых, в свою очередь, является функцией двух действительных переменных h_R и h_I

$$\begin{cases} h_1 = h_R + kh_1 \\ h_2 = h_R - kh_1, \end{cases} \quad (2.26)$$

где k — ненулевое комплексное число (не путать с индексом дискретного времени). То есть

$$g(h_1, h_2) = g\{h_1(h_R, h_1), h_2(h_R, h_1)\} = f(h_R, h_1). \quad (2.27)$$

Если функция g является независимо аналитической по отношению к каждой из переменных h_1 и h_2 , то частные производные по h_R и h_1 можно определить как

$$\begin{cases} \frac{\partial g}{\partial h_1} \frac{\partial h_1}{\partial h_R} + \frac{\partial g}{\partial h_2} \frac{\partial h_2}{\partial h_R} = \frac{\partial f}{\partial h_R} \\ \frac{\partial g}{\partial h_1} \frac{\partial h_1}{\partial h_1} + \frac{\partial g}{\partial h_2} \frac{\partial h_2}{\partial h_1} = \frac{\partial f}{\partial h_1}. \end{cases} \quad (2.28)$$

Так как из системы (2.26) следует, что $\frac{\partial h_1}{\partial h_R} = \frac{\partial h_2}{\partial h_R} = 1$ и $\frac{\partial h_1}{\partial h_1} = -\frac{\partial h_2}{\partial h_1} = k$, то уравнение (2.28) преобразуется к виду

$$\begin{cases} \frac{\partial g}{\partial h_1} + \frac{\partial g}{\partial h_2} = \frac{\partial f}{\partial h_R} \\ k \left(\frac{\partial g}{\partial h_1} - \frac{\partial g}{\partial h_2} \right) = \frac{\partial f}{\partial h_1}, \end{cases} \quad (2.29)$$

откуда следует, что

$$\begin{cases} \frac{\partial g}{\partial h_1} = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} + \frac{1}{k} \frac{\partial f}{\partial h_1} \right) \\ \frac{\partial g}{\partial h_2} = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} - \frac{1}{k} \frac{\partial f}{\partial h_1} \right). \end{cases} \quad (2.30)$$

Если $k=j$, где $j^2=-1$, то, учитывая, что $h_2=h_1^*$, уравнение (2.30) можно представить как

$$\begin{cases} \frac{\partial g}{\partial h} = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} - j \frac{\partial f}{\partial h_1} \right) \\ \frac{\partial g}{\partial h^*} = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} + j \frac{\partial f}{\partial h_1} \right). \end{cases} \quad (2.31)$$

Таким образом, уравнение (2.31) показывает, что воздействие оператора $\partial/\partial h$ на функцию $g(h, h^*)$ эквивалентно воздействию оператора $(\partial/\partial h_R - j\partial/\partial h_1)/2$ на функцию $f(h_R, h_1) = g(h, h^*)$, а воздействие $\partial/\partial h^*$ на эту же функцию эквивалентно воздействию оператора $(\partial/\partial h_R + j\partial/\partial h_1)/2$.

Уравнение (2.31) можно также получить, исходя из общих принципов дифференцирования функции комплексного переменного и его сопряженного значения [69].

Пусть имеется аналитическая функция комплексного переменного

$$p(h) = g(h, h^*) = u(h_R, h_I) + jv(h_R, h_I) = f(h_R, h_I), \quad (2.32)$$

где $u(h_R, h_I)$ и $v(h_R, h_I)$ — действительные функции действительных переменных.

По определению производная dp/dh в точке $h_0 = h_{0,R} + jh_{0,I}$ вычисляется как

$$\frac{dp}{dh} = \lim_{\Delta h \rightarrow 0} \frac{p(h_0 + \Delta h) - p(h_0)}{\Delta h} = \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I \rightarrow 0}} \frac{p(h_{0,R} + \Delta h_R, h_{0,I} + \Delta h_I) - p(h_{0,R}, h_{0,I})}{\Delta h_R + j\Delta h_I}. \quad (2.33)$$

Уравнение (2.33) должно выполняться независимо от того, как $\Delta h \rightarrow 0$. Поэтому если $\Delta h_I = 0$, то

$$\begin{aligned} \frac{dp}{dh} &= \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I = 0}} \frac{u(h_{0,R} + \Delta h_R, h_{0,I}) - u(h_{0,R}, h_{0,I})}{\Delta h_R} + \\ &+ j \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I = 0}} \frac{v(h_{0,R} + \Delta h_R, h_{0,I}) - v(h_{0,R}, h_{0,I})}{\Delta h_R} = \frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} = \frac{\partial g}{\partial h}. \end{aligned} \quad (2.34)$$

Аналогично, если $\Delta h_R = 0$, то

$$\begin{aligned} \frac{dp}{dh} &= \lim_{\substack{\Delta h_R = 0 \\ \Delta h_I \rightarrow 0}} \frac{u(h_{0,R}, h_{0,I} + \Delta h_I) - u(h_{0,R}, h_{0,I})}{j\Delta h_I} + \\ &+ j \lim_{\substack{\Delta h_R = 0 \\ \Delta h_I \rightarrow 0}} \frac{v(h_{0,R}, h_{0,I} + \Delta h_I) - v(h_{0,R}, h_{0,I})}{j\Delta h_I} = \frac{\partial v}{\partial h_I} - j \frac{\partial u}{\partial h_I} = \frac{\partial g}{\partial h}. \end{aligned} \quad (2.35)$$

Складывая оба результата, получим

$$\begin{aligned} \frac{\partial g}{\partial h} &= \frac{1}{2} \left(\frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} + \frac{\partial v}{\partial h_I} - j \frac{\partial u}{\partial h_I} \right) = \\ &= \frac{1}{2} \left(\left[\frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} \right] - j \left[\frac{\partial u}{\partial h_I} + j \frac{\partial v}{\partial h_I} \right] \right) = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} - j \frac{\partial f}{\partial h_I} \right). \end{aligned} \quad (2.36)$$

Это уравнение совпадает с верхним уравнением в выражении (2.31).

Приравнивая оба результата $\frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} = \frac{\partial v}{\partial h_I} - j \frac{\partial u}{\partial h_I}$, получим, что

$$\begin{cases} \frac{\partial u}{\partial h_R} = \frac{\partial v}{\partial h_I} \\ \frac{\partial v}{\partial h_R} = -\frac{\partial u}{\partial h_I}. \end{cases} \quad (2.37)$$

Уравнения (2.37) называются условиями Коши — Римана дифференцируемости функции комплексного переменного. Эти уравнения устанавливают связь между частными производными этой функции по действительным и мнимым составляющими комплексного переменного.

Производная dp/dh^* определяется аналогично уравнению (2.33) как

$$\frac{dp}{dh^*} = \lim_{\Delta h \rightarrow 0} \frac{p(h_0 + \Delta h^*) - p(h_0)}{\Delta h^*} = \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I \rightarrow 0}} \frac{p(h_{0,R} + \Delta h_R, h_{0,I} + \Delta h_I) - p(h_{0,R}, h_{0,I})}{\Delta h_R - j\Delta h_I}. \quad (2.38)$$

Уравнение (2.38) должно выполняться независимо от того, как $\Delta h \rightarrow 0$. Поэтому если $\Delta h_I = 0$, то

$$\begin{aligned} \frac{dp}{dh^*} &= \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I = 0}} \frac{u(h_{0,R} + \Delta h_R, h_{0,I}) - u(h_{0,R}, h_{0,I})}{\Delta h_R} + \\ &+ j \lim_{\substack{\Delta h_R \rightarrow 0 \\ \Delta h_I = 0}} \frac{v(h_{0,R} + \Delta h_R, h_{0,I}) - v(h_{0,R}, h_{0,I})}{\Delta h_R} = \frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} = \frac{\partial g}{\partial h^*}. \end{aligned} \quad (2.39)$$

Аналогично, если $\Delta h_R = 0$, то

$$\begin{aligned} \frac{dp}{dh^*} &= \lim_{\substack{\Delta h_R = 0 \\ \Delta h_I \rightarrow 0}} \frac{u(h_{0,R}, h_{0,I} + \Delta h_I) - u(h_{0,R}, h_{0,I})}{-j \Delta h_I} + \\ &+ j \lim_{\substack{\Delta h_R = 0 \\ \Delta h_I \rightarrow 0}} \frac{v(h_{0,R}, h_{0,I} + \Delta h_I) - v(h_{0,R}, h_{0,I})}{-j \Delta h_I} = -\frac{\partial v}{\partial h_I} + j \frac{\partial u}{\partial h_I} = \frac{\partial g}{\partial h^*}. \end{aligned} \quad (2.40)$$

Складывая оба результата, получим, что

$$\begin{aligned} \frac{\partial g}{\partial h^*} &= \frac{1}{2} \left(\frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} - \frac{\partial v}{\partial h_I} + j \frac{\partial u}{\partial h_I} \right) = \\ &= \frac{1}{2} \left(\left[\frac{\partial u}{\partial h_R} + j \frac{\partial v}{\partial h_R} \right] + j \left[\frac{\partial u}{\partial h_I} + j \frac{\partial v}{\partial h_I} \right] \right) = \frac{1}{2} \left(\frac{\partial f}{\partial h_R} + j \frac{\partial f}{\partial h_I} \right). \end{aligned} \quad (2.41)$$

Это уравнение совпадает с нижним уравнением в выражении (2.31).

Приведем пример, демонстрирующий рассмотренные правила дифференцирования. Определим функцию $g(h, h^*)$ как

$$g(h, h^*) = h^2 h^* = f(h_R, h_I) = (h_R + j h_I)^2 (h_R - j h_I) = (h_R^3 + h_R h_I^2) + j(h_R^2 h_I + h_I^3).$$

Тогда $\frac{\partial g}{\partial h} = 2 h h^* = 2(h_R + j h_I)(h_R - j h_I) = 2(h_R^2 + h_I^2)$. С другой стороны, согласно системе (2.31)

$$\frac{\partial g}{\partial h} = \frac{1}{2} \left[(3h_R^2 + h_I^2 + j2h_R h_I) - j(2h_R h_I + jh_R^2 + j3h_I^2) \right] = 2(h_R^2 + h_I^2).$$

Аналогично

$$\frac{\partial g}{\partial h^*} = h^2 = h_R^2 + j2h_R h_I - h_I^2$$

$$\text{и } \frac{\partial g}{\partial h^*} = \frac{1}{2} \left[(3h_R^2 + h_I^2 + j2h_R h_I) + j(2h_R h_I + jh_R^2 + j3h_I^2) \right] = h_R^2 + j2h_R h_I - h_I^2.$$

Таким образом, рассмотренный пример подтверждает справедливость уравнений (2.31).

Если определить функцию $g(h, h^*)$ как $g_1(h, h^*) = h = f(h_R, h_I) = h_R + j h_I$ и $g_2(h, h^*) = h^* = f(h_R, h_I) = h_R - j h_I$, то из системы (2.31) следует, что

$$\frac{\partial h^*}{\partial h} = \frac{1}{2} \left(\frac{\partial(h_R - j h_I)}{\partial h_R} - j \frac{\partial(h_R - j h_I)}{\partial h_I} \right) = \frac{1}{2}(1 - 1) = 0, \quad (2.42)$$

$$\frac{\partial h}{\partial h^*} = \frac{1}{2} \left(\frac{\partial (h_R + jh_I)}{\partial h_R} + j \frac{\partial (h_R + jh_I)}{\partial h_I} \right) = \frac{1}{2} (1 - 1) = 0, \quad (2.43)$$

$$\frac{\partial h^*}{\partial h^*} = \frac{1}{2} \left(\frac{\partial (h_R - jh_I)}{\partial h_R} + j \frac{\partial (h_R - jh_I)}{\partial h_I} \right) = \frac{1}{2} (1 + 1) = 1, \quad (2.44)$$

$$\frac{\partial h}{\partial h} = \frac{1}{2} \left(\frac{\partial (h_R + jh_I)}{\partial h_R} - j \frac{\partial (h_R + jh_I)}{\partial h_I} \right) = \frac{1}{2} (1 + 1) = 1, \quad (2.45)$$

т. е. $\partial h^* / \partial h = \partial h / \partial h^* = 0$ и $\partial h^* / \partial h^* = \partial h / \partial h = 1$. Этот результат подтверждает справедливость правил дифференцирования (2.18)–(2.21).

Пусть теперь p — это действительная функция комплексного переменного h . Она также является функцией действительных переменных h_R и h_I , так как $h = h_R + jh_I$. Однако в теории адаптивной фильтрации удобнее выражать эту действительную функцию в терминах комплексных переменных h и h^* вместо переменных h_R и h_I .

Чтобы быть действительной, функция p комплексного переменного h должна содержать и переменную h^* . Такая функция не является аналитической, так как $h \rightarrow h^*$ — не аналитическая функция.

Но мы предполагаем, что p **такова**, что функция $g(h, h^*) = p(h)$ — независимо аналитическая по h и h^* в смысле существования частных производных по этим переменным. Например, пусть имеется функция $p(h) = |h|^2 = hh^*$. Функция $p(h) = |h|^2$ — это не аналитическая функция от h , так как dp/dh не существует. Однако $g(h, h^*) = hh^*$ — это аналитическая функция независимо по h и по h^* , так как существуют частные производные $\partial g / \partial h$ и $\partial g / \partial h^*$.

Чтобы функция $g(h, h^*) = u(h_R, h_I) + jv(h_R, h_I)$ была действительной, необходимо, чтобы $v(h_R, h_I) = 0$. В этом случае уравнения (2.31) преобразуются к виду

$$\begin{cases} \frac{\partial g}{\partial h} = \frac{1}{2} \left(\frac{\partial u}{\partial h_R} - j \frac{\partial u}{\partial h_I} \right) \\ \frac{\partial g}{\partial h^*} = \frac{1}{2} \left(\frac{\partial u}{\partial h_R} + j \frac{\partial u}{\partial h_I} \right). \end{cases} \quad (2.46)$$

Согласно системе (2.46) справедливы следующие соотношения:

$$\begin{cases} \frac{\partial u}{\partial h_R} = \frac{\partial u}{\partial h_I} = 0 \Rightarrow \frac{\partial g}{\partial h} = 0 \\ \frac{\partial g}{\partial h^*} = 0 \Rightarrow \frac{\partial u}{\partial h_R} = \frac{\partial u}{\partial h_I} = 0. \end{cases} \quad (2.47)$$

Аналогично

$$\begin{cases} \frac{\partial u}{\partial h_R} = \frac{\partial u}{\partial h_I} = 0 \Rightarrow \frac{\partial g}{\partial h^*} = 0 \\ \frac{\partial g}{\partial h^*} = 0 \Rightarrow \frac{\partial u}{\partial h_R} = \frac{\partial u}{\partial h_I} = 0. \end{cases} \quad (2.48)$$

Поскольку $\partial u / \partial h_R = \partial u / \partial h_I = 0$ — это условие того, что функция u , а значит и функция g имеют стационарную точку (минимум или максимум), то, как следует из систем (2.47) и (2.48), равенство $\partial g / \partial h = 0$ или равенство $\partial g / \partial h^* = 0$ — это также необходимое и достаточное условие того, что функция g , а значит и функция p , имеет стационарную точку.

Рассмотренные выше результаты можно распространить на многомерный (N -мерный) случай. Если имеется комплексный вектор

$$\mathbf{h}_N = [h_1, h_2, \dots, h_N]^T = \mathbf{h}_{N,R} + j \mathbf{h}_{N,I} = [h_{1,R}, h_{2,R}, \dots, h_{N,R}]^T + j [h_{1,I}, h_{2,I}, \dots, h_{N,I}]^T, \quad (2.49)$$

то оператор комплексного градиента $\nabla_{\mathbf{h}_N}$ определяется уравнением (2.2), а его компоненты — уравнением (2.31) как

$$\frac{\partial}{\partial h_n} = \frac{1}{2} \left(\frac{\partial}{\partial h_{n,R}} - j \frac{\partial}{\partial h_{n,I}} \right), \quad (2.50)$$

где $n = 1, 2, \dots, N$. Аналогично, оператор комплексного градиента $\nabla_{\mathbf{h}_N^*}$ определяется уравнением (2.3), а его компоненты — уравнением (2.31) как

$$\frac{\partial}{\partial h_n^*} = \frac{1}{2} \left(\frac{\partial}{\partial h_{n,R}} + j \frac{\partial}{\partial h_{n,I}} \right), \quad (2.51)$$

где $n = 1, 2, \dots, N$.

Если действительная функция p комплексного вектора \mathbf{h}_N такова, что $p(\mathbf{h}_N) = g(\mathbf{h}_N, \mathbf{h}_N^*)$ — независимо аналитическая (т. е. дифференцируемая) по \mathbf{h}_N и по \mathbf{h}_N^* функция, то любое из условий $\nabla_{\mathbf{h}_N} g = \mathbf{0}_N$ или $\nabla_{\mathbf{h}_N^*} g = \mathbf{0}_N$ является необходимым и достаточным условием наличия стационарной точки в функции p .

Действительно, если выразить функцию p как функцию u от $2N$ переменных $h_{n,R}$ и $h_{n,I}$, то функция p , а значит, и функция u стационарны тогда и только тогда, когда $\partial u / \partial h_{n,R} = \partial u / \partial h_{n,I} = 0$ для всех n . Тогда из системы (2.47) следует, что и $\partial g / \partial h_n = 0$ для всех n , т. е. $\nabla_{\mathbf{h}_N} g = \mathbf{0}_N$. Аналогично, из системы (2.48) следует, что если $\partial u / \partial h_{n,R} = \partial u / \partial h_{n,I} = 0$ для всех n , то и $\nabla_{\mathbf{h}_N^*} g = \mathbf{0}_N$, т. е. $\partial g / \partial h_n^* = 0$ для всех n .

В одномерном случае дифференциал (т. е. приращение) функции $g(h, h^*)$ определяется как

$$\Delta g = \frac{\partial g}{\partial h} \Delta h + \frac{\partial g}{\partial h^*} \Delta h^* = 2 \operatorname{Re} \left\{ \frac{\partial g}{\partial h} \Delta h \right\}, \quad (2.52)$$

так как $\frac{\partial g}{\partial h^*} \Delta h^* = \left(\frac{\partial g}{\partial h} \right)^* \Delta h^* = \left(\frac{\partial g}{\partial h} \Delta h \right)^*$, см. систему (2.46).

В многомерном случае дифференциал функции $g(\mathbf{h}_N, \mathbf{h}_N^*)$ определяется как

$$\begin{aligned} \Delta g &= \sum_{n=1}^N \left(\frac{\partial g}{\partial h_n} \Delta h_n + \frac{\partial g}{\partial h_n^*} \Delta h_n^* \right) = \sum_{n=1}^N \frac{\partial g}{\partial h_n} \Delta h_n + \sum_{n=1}^N \frac{\partial g}{\partial h_n^*} \Delta h_n^* = \\ &= \left(\nabla_{\mathbf{h}_N} g \right)^T \Delta \mathbf{h}_N + \left(\nabla_{\mathbf{h}_N^*} g \right)^T \Delta \mathbf{h}_N^* = 2 \operatorname{Re} \left\{ \left(\nabla_{\mathbf{h}_N} g \right)^T \Delta \mathbf{h}_N \right\} = 2 \operatorname{Re} \left\{ \left(\nabla_{\mathbf{h}_N^*} g \right)^H \Delta \mathbf{h}_N \right\}, \end{aligned} \quad (2.53)$$

так как $\nabla_{\mathbf{h}_N} g = (\nabla_{\mathbf{h}_N} g)^*$, см. систему (2.46).

Известно, что скалярное произведение векторов \mathbf{a}_N и \mathbf{b}_N удовлетворяет неравенству Шварца $|\langle \mathbf{a}_N, \mathbf{b}_N \rangle| \leq \|\mathbf{a}_N\|_2 \|\mathbf{b}_N\|_2$, где $\|\mathbf{a}_N\|_2 = \sqrt{\sum_{n=1}^N a_n^2}$ — норма (т. е. длина) вектора \mathbf{a}_N , $\|\mathbf{b}_N\|_2 = \sqrt{\sum_{n=1}^N b_n^2}$ — норма вектора \mathbf{b}_N . Равенство в этом выражении выполняется, если векторы \mathbf{a}_N и \mathbf{b}_N коллинеарные, т. е. $\mathbf{b}_N = k \mathbf{a}_N$, где k — скаляр (константа, которая может быть комплексной; не путать с индексом дискретного времени). В этом случае $\langle \mathbf{a}_N, \mathbf{b}_N \rangle = \|\mathbf{a}_N\|_2 k \|\mathbf{a}_N\|_2 = k \|\mathbf{a}_N\|_2^2$. Поэтому $\operatorname{Re}\{\langle \mathbf{a}_N, \mathbf{b}_N \rangle\}$ максимально, если k — действительное положительное число, т. е. когда \mathbf{a}_N и \mathbf{b}_N направлены в одну сторону. Применяя вышесказанное к уравнению (2.53), можно сделать вывод о том, что для заданного значения $\Delta \mathbf{h}_N$ максимальное изменение функции g , а значит, и функции p , так как $p(\mathbf{h}_N) = g(\mathbf{h}_N, \mathbf{h}_N^*)$, имеет место, когда направления векторов $\Delta \mathbf{h}_N$ и $\nabla_{\mathbf{h}_N} g$ совпадают.

Таким образом, для функции $p(\mathbf{h}_N) = g(\mathbf{h}_N, \mathbf{h}_N^*)$ градиент $\nabla_{\mathbf{h}_N^*} g$ определяет максимальную скорость ее изменения при изменении вектора \mathbf{h}_N . Для бесконечно малого $\Delta \mathbf{h}_N$ это изменение согласно уравнению (2.53) определяется как

$$\Delta p = 2 \operatorname{Re} \left\{ \left(\nabla_{\mathbf{h}_N^*} g \right)^H \Delta \mathbf{h}_N \right\}. \quad (2.54)$$

2.4. Свойства оператора комплексного градиента

Оператор комплексного градиента, определяемый уравнением (2.2) или (2.3), формирует комплексный вектор из скалярной функции от \mathbf{h}_N . Скалярная функция в общем случае может быть комплексной. Однако в приложениях адаптивной фильтрации эта функция обычно действительная, в частности, когда она является величиной, по физическому смыслу соответствующей мощности, или равна сумме комплексной и комплексно-сопряженной величин $a + a^* = 2\operatorname{Re}(a)$. В теории адаптивной фильтрации типичными являются скалярные функции типа $\mathbf{a}_N^H \mathbf{h}_N$ и $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$, где компоненты вектора \mathbf{a}_N и матрицы \mathbf{R}_N являются константами по отношению к компонентам вектора \mathbf{h}_N . Функция $\mathbf{a}_N^H \mathbf{h}_N$ в общем случае является комплексной, а функция $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$ — действительной. Поскольку матрица \mathbf{R}_N является эрмитовой матрицей, то $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$ — это действительная величина, так как $\mathbf{R}_N = \mathbf{b}_N \mathbf{b}_N^H$ и $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N = \mathbf{h}_N^H \mathbf{b}_N \mathbf{b}_N^H \mathbf{h}_N = c c^* = |c|^2$.

Учитывая соотношения (2.18)–(2.21), можно показать, что

$$\nabla_{\mathbf{h}_N^*} \left\{ \mathbf{a}_N^H \mathbf{h}_N \right\} = \nabla_{\mathbf{h}_N^*} \left\{ \sum_{n=1}^N a_n^* h_n \right\} = [0, 0, \dots, 0]^T = \mathbf{0}_N, \quad (2.55)$$

$$\nabla_{\mathbf{h}_N} \left\{ \mathbf{a}_N^H \mathbf{h}_N \right\} = \nabla_{\mathbf{h}_N} \left\{ \sum_{n=1}^N a_n^* h_n \right\} = [a_1^*, a_2^*, \dots, a_N^*]^T = \mathbf{a}_N^*, \quad (2.56)$$

$$\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{a}_N \right\} = \nabla_{\mathbf{h}_N} \left\{ \sum_{n=1}^N h_n^* a_n \right\} = \left[a_1, a_2, \dots, a_N \right]^T = \mathbf{a}_N, \quad (2.57)$$

$$\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{a}_N \right\} = \nabla_{\mathbf{h}_N} \left\{ \sum_{n=1}^N h_n^* a_n \right\} = \left[0, 0, \dots, 0 \right]^T = \mathbf{0}_N, \quad (2.58)$$

$$\nabla_{\mathbf{h}_N^*} \left\{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \right\} = \nabla_{\mathbf{h}_N^*} \left\{ \sum_{m=1}^N h_m^* \sum_{n=1}^N R_{mn} h_n \right\} = \left[\sum_{n=1}^N R_{1n} h_n, \sum_{n=1}^N R_{2n} h_n, \dots, \sum_{n=1}^N R_{Nn} h_n \right]^T = \mathbf{R}_N \mathbf{h}_N, \quad (2.59)$$

$$\begin{aligned} \nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \right\} &= \nabla_{\mathbf{h}_N} \left\{ \sum_{m=1}^N h_m^* \sum_{n=1}^N R_{mn} h_n \right\} = \\ &= \left[\sum_{m=1}^N h_m^* R_{m1}, \sum_{m=1}^N h_m^* R_{m2}, \dots, \sum_{m=1}^N h_m^* R_{mN} \right]^T = \mathbf{h}_N^H \left[\mathbf{r}_{N,1}, \mathbf{r}_{N,2}, \dots, \mathbf{r}_{N,N} \right] = \mathbf{h}_N^H \mathbf{R}_N. \end{aligned} \quad (2.60)$$

Аналогичным образом могут быть получены результаты для других функций $p(\mathbf{h}_N) = g(\mathbf{h}_N, \mathbf{h}_N^*)$.

Поскольку любой из двух градиентов $\nabla_{\mathbf{h}_N^*}$ и $\nabla_{\mathbf{h}_N}$ позволяет определить условия наличия стационарной точки функции, то представляется **более удобным** использовать градиент $\nabla_{\mathbf{h}_N^*}$, учитывая и то, что он определяет максимальную скорость изменения функции.

Различие между операторами градиента функций действительного и комплексного переменных имеет важное значение для понимания адаптивных алгоритмов. Оно, в частности, позволяет понять, почему математические выражения в комплексных адаптивных алгоритмах нельзя получить простой заменой действительных переменных в действительных алгоритмах на комплексные переменные.

Если, например, в уравнении (2.60) $\mathbf{R}_N = \mathbf{I}_N$, то градиент определяется как

$$\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{h}_N \right\} = \mathbf{h}_N^*, \quad (2.61)$$

тогда как в случае действительного вектора $\tilde{\mathbf{h}}_N$ градиент определяется как

$$\nabla_{\tilde{\mathbf{h}}_N} \left\{ \tilde{\mathbf{h}}_N^T \tilde{\mathbf{h}}_N \right\} = 2\tilde{\mathbf{h}}_N. \quad (2.62)$$

Уравнение (2.61) отличается от уравнения (2.62) двумя моментами. В нем отсутствует множитель 2 и дифференцирование по \mathbf{h}_N не такое же самое, как дифференцирование по \mathbf{h}_N^* . Эквивалентом уравнения (2.62) в случае комплексных переменных является уравнение $\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^T \mathbf{h}_N \right\} = 2\mathbf{h}_N$. Однако комплексная функция $\mathbf{h}_N^T \mathbf{h}_N$, т. е. функция только \mathbf{h}_N , а не \mathbf{h}_N и \mathbf{h}_N^* , в теории адаптивной фильтрации не встречается.

Таким образом, справедливы следующие соотношения: $\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{h}_N^* \right\} = \mathbf{0}_N$, $\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^H \mathbf{h}_N \right\} = \mathbf{h}_N^*$ и $\nabla_{\mathbf{h}_N} \left\{ \mathbf{h}_N^T \mathbf{h}_N \right\} = 2\mathbf{h}_N$. Видно, что градиент теряет множитель 2, когда комплексная функция переменной \mathbf{h}_N заменяется действительной функцией переменных \mathbf{h}_N и \mathbf{h}_N^* .

Таким образом, вполне вероятно, что ошибочное мнение, будто комплексный эквивалент действительного градиента $\nabla_{\tilde{\mathbf{h}}_N} \{ \tilde{\mathbf{h}}_N^T \mathbf{R}_N \tilde{\mathbf{h}}_N \} = 2\mathbf{R}_N \tilde{\mathbf{h}}_N$ должен равняться $\nabla_{\mathbf{h}_N^*} \{ \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \} = 2\mathbf{R}_N \mathbf{h}_N$, послужило причиной того, что встречающееся в литературе определение комплексного градиента эквивалентно значению $2\nabla_{\mathbf{h}_N^*}$. Хотя множитель 2 и не приводит к неправильным конечным результатам, так как производные вектора градиента просто масштабируются, однако более естественным является то, что комплексная производная $\partial h / \partial \mathbf{h} = 1$, а не $\partial h / \partial \mathbf{h} = 2$.

2.5. Выводы по главе

В главе 2 дано определение оператора комплексного градиента, позволяющего достаточно просто получать математические соотношения для комплексных алгоритмов адаптивной фильтрации. Показано, что оператор комплексного градиента и оператор комплексно-сопряженного градиента позволяют определять стационарную точку (точку минимума, в которой компоненты градиента равны нулю) действительной функции от вектора комплексных переменных. Оператор комплексно-сопряженного градиента определяет направление максимальной скорости изменения этой функции, поэтому его целесообразно использовать при получении и анализе вычислительных процедур алгоритмов для комплексных адаптивных фильтров.

ГЛАВА 3

ОСНОВЫ АДАПТИВНОЙ ФИЛЬТРАЦИИ СИГНАЛОВ

В настоящей главе рассматриваются фундаментальные понятия адаптивной обработки сигналов, такие как корреляционная матрица, собственные числа и собственные векторы этой матрицы, поверхность среднеквадратической ошибки, безусловная и условная (линейно-ограниченная) винеровская фильтрация.

3.1. Введение

Адаптивная фильтрация сигналов базируется на ряде принципов, вытекающих из оптимальной винеровской фильтрации. Поэтому изложение теории адаптивной фильтрации в литературе обычно начинается с основных элементов теории винеровской фильтрации. Такие элементы рассматриваются и в настоящей главе, которая основана на материалах, представленных в [3, 20, 29].

Решение задачи оптимальной винеровской фильтрации представляет собой решение задачи минимизации MSE-функции дискретных или цифровых фильтров, реализуемых с помощью линейного сумматора. Это решение зависит от корреляционной матрицы входных сигналов сумматора и вектора взаимной корреляции входных и требуемого сигналов. Многомерная поверхность MSE-функции является унимодальной в пространстве ВК линейного сумматора, что гарантирует единственное решение задачи минимизации этой функции.

На практике корреляционная матрица и вектор взаимной корреляции, как правило, неизвестны. Однако их можно оценивать с помощью обрабатываемых сигналов, что позволяет получать алгоритмы, использующие эти оценки для расчета ВК адаптивного фильтра, которые в некотором смысле сходятся к оптимальному винеровскому решению. Поэтому винеровское решение можно рассматривать в качестве эталона, с которым обычно сравниваются решения, получаемые с помощью адаптивных алгоритмов.

3.2. Корреляционная матрица

Адаптивные алгоритмы используют в своих уравнениях входные сигналы линейного сумматора (рис. 3.1), измеренные в дискретные моменты времени k . Результаты этих измерений образуют вектор отсчетов входных сигналов

$$\mathbf{x}_N(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T. \quad (3.1)$$

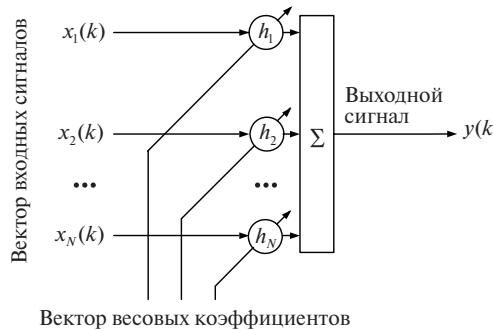


Рис. 3.1. Линейный сумматор

Если адаптивный фильтр по своей структуре является одноканальным КИХ-фильтром (см. рис. 1.2), то вектор

$$\mathbf{x}_N(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T = [x(k), x(k-1), \dots, x(k-N+1)]^T \quad (3.2)$$

представляет собой задержанные отсчеты входного сигнала этого фильтра, т. е. дискретный стохастический процесс.

Если адаптивный фильтр — это многоканальный фильтр с одним ВК в каждом канале, например антенная решетка (см. рис. 1.26), то

$$\mathbf{x}_N(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T = [x_1(k), x_2(k), \dots, x_M(k)]^T, \quad (3.3)$$

где $N = M$. В антенной решетке сигналы $x_m(k)$, $m = 1, 2, \dots, M$ являются дискретными стохастическими процессами с одинаковыми статистическими характеристиками.

Если адаптивный фильтр является M -канальным КИХ-фильтром с неодинаковым числом ВК N_m в каналах (см. рис. 1.3), то

$$\mathbf{x}_N(k) = [\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_M}^T(k)]^T, \quad (3.4)$$

где $\mathbf{x}_{N_m}(k) = [x_m(k), x_m(k-1), \dots, x_m(k-N_m+1)]^T$, $m = 1, 2, \dots, M$, а $N = \sum_{m=1}^M N_m$.

При этом сигналы $x_m(k)$, $m = 1, 2, \dots, M$ могут иметь одинаковые или разные (одинаковые в пределах каналов) статистические характеристики.

Таким образом, в общем случае сигналы $x_n(k)$ (см. рис. 3.1) могут иметь разные статистические характеристики. Корреляционная матрица этих сигналов определяется как

$$\begin{aligned}
 \mathbf{R}_N &= E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} = E\left[\begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} \begin{bmatrix} x_1^*(k), x_2^*(k), \dots, x_N^*(k) \end{bmatrix}\right] = \\
 &= E\left[\begin{bmatrix} x_1(k)x_1^*(k) & x_1(k)x_2^*(k) & \dots & x_1(k)x_N^*(k) \\ x_2(k)x_1^*(k) & x_2(k)x_2^*(k) & \dots & x_2(k)x_N^*(k) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(k)x_1^*(k) & x_N(k)x_2^*(k) & \dots & x_N(k)x_N^*(k) \end{bmatrix}\right] = \\
 &= \begin{bmatrix} E\{|x_1(k)|^2\} & E\{x_1(k)x_2^*(k)\} & \dots & E\{x_1(k)x_N^*(k)\} \\ E\{x_2(k)x_1^*(k)\} & E\{|x_2(k)|^2\} & \dots & E\{x_2(k)x_N^*(k)\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{x_N(k)x_1^*(k)\} & E\{x_N(k)x_2^*(k)\} & \dots & E\{|x_N(k)|^2\} \end{bmatrix}, \tag{3.5}
 \end{aligned}$$

где $E\{\cdot\}$ — операция усреднения по ансамблю реализаций.

Если сигналы $x_n(k)$, $n=1, 2, \dots, N$ обладают одинаковыми статистическими характеристиками, то корреляционная матрица (3.5) определяется как

$$\mathbf{R}_N = \begin{bmatrix} R(0) & R(1) & R(2) & R(3) & \dots & R(N-1) \\ R(-1) & R(0) & R(1) & R(2) & \dots & R(N-2) \\ R(-2) & R(-1) & R(0) & R(1) & \dots & R(N-3) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ R(-N+2) & R(-N+3) & R(-N+4) & R(-N+5) & \dots & R(1) \\ R(-N+1) & R(-N+2) & R(-N+3) & R(-N+4) & \dots & R(0) \end{bmatrix}. \tag{3.6}$$

В матрице (3.6) $R(n-m) = E\{x_n(k)x_m^*(k)\}$ — это отсчеты взаимокорреляционной функции, если вектор $\mathbf{x}_N(k)$ определяется уравнением (3.3), и $R(n-m) = E\{x(k-n+1)x^*(k-m+1)\}$ — отсчеты автокорреляционной функции, если вектор $\mathbf{x}_N(k)$ определяется уравнением (3.2). Диагональные элементы $R(0)$ матрицы (3.6) являются одинаковыми. Они равны среднеквадратическим значениям входных сигналов линейного сумматора, т.е. представляют собой действительные положительные числа. Остальные элементы $R(1), \dots, R(N-1), R(-1), \dots, R(-N+1)$ действительные или комплексные в зависимости от того, какие сигналы линейного сумматора: действительные или комплексные. Для этих элементов всегда справедливо соотношение $|R(n-m)| \leq R(0)$, так как они пред-

ставляют собой значения боковых лепестков авто- или взаимокорреляционной функции, которые по абсолютной величине не превышают значения основного лепестка $R(0)$ этой функции.

Корреляционная матрица обладает рядом свойств. Некоторые из них рассматриваются ниже.

Свойство 1. Корреляционная матрица является эрмитовой, т. е.

$$\mathbf{R}_N = \mathbf{R}_N^H. \quad (3.7)$$

Свойство (3.7) следует из уравнения

$$\begin{aligned} \mathbf{R}_N^H &= \left[E \left\{ \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right\} \right]^H = E \left\{ \left[\mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^H \right\} = \\ &= E \left\{ \left[\mathbf{x}_N^H(k) \right]^H \left[\mathbf{x}_N(k) \right]^H \right\} = E \left\{ \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right\} = \mathbf{R}_N, \end{aligned} \quad (3.8)$$

так как согласно [71]

$$\left[\mathbf{A}_{NM} \mathbf{B}_{MN} \right]^H = \mathbf{B}_{MN}^H \mathbf{A}_{NM}^H. \quad (3.9)$$

Из уравнения (3.7) следует, что корреляционная матрица действительных сигналов является симметричной, т. е. ее элементы, расположенные симметрично относительно главной диагонали, являются одинаковыми. Кроме того, из свойства (3.7) следует, что $\mathbf{R}_N^{-1} = \mathbf{R}_N^{-H}$, т. е. обратная корреляционная матрица также является эрмитовой.

Действительно, допустим, что обратная матрица \mathbf{R}_N^{-1} существует. Тогда, согласно (3.7), $\mathbf{R}_N^{-1} \mathbf{R}_N = \mathbf{R}_N^{-1} \mathbf{R}_N^H = \mathbf{I}_N$. Выполнив эрмитово сопряжение обеих частей этого равенства, т. е. $(\mathbf{R}_N^{-1} \mathbf{R}_N^H)^H = (\mathbf{I}_N)^H$, получим, что $\mathbf{R}_N \mathbf{R}_N^{-H} = \mathbf{I}_N$. Здесь использовано соотношение (3.9). Следовательно, $\mathbf{R}_N^{-H} = \mathbf{R}_N^{-1}$, а значит, $\mathbf{R}_N^{-1} = (\mathbf{R}_N^{-1})^H = \mathbf{R}_N^{-H}$ — это эрмитова матрица.

Свойство 2. Корреляционная матрица стационарного дискретного стохастического процесса является теплицевой.

Теплицевой называется матрица, у которой одинаковые между собой элементы на главной диагонали и элементы на каждой из диагоналей, параллельных главной, также одинаковы между собой. Из уравнения (3.6) следует, что все элементы на главной диагонали матрицы \mathbf{R}_N стационарного дискретного стохастического процесса равны $R(0)$. На каждой из диагоналей над основной диагональю располагаются соответственно элементы $R(1), R(2), \dots, R(N-1)$, а на каждой из диагоналей под основной диагональю согласно равенству (3.7) — элементы $R(-1) = R^*(1), R(-2) = R^*(2), \dots, R(-N+1) = R^*(N-1)$. Таким образом, можно сделать вывод, что матрица (3.6) является теплицевой.

Свойство 3. Корреляционная матрица является положительно полуопределенной.

Для доказательства этого утверждения определим для произвольных комплексных векторов $\mathbf{x}_N(k)$ и $\mathbf{h}_N \neq \mathbf{0}_N$ переменную $y(k)$ как скалярное произведение этих векторов, т. е.

$$y(k) = \mathbf{h}_N^H \mathbf{x}_N(k) = \sum_{n=1}^N h_n^* x_n(k). \quad (3.10)$$

Так как произведение $y(k)y^*(k)$ является неотрицательной величиной

$$y(k)y^*(k) = |y(k)|^2 = \mathbf{h}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{h}_N \geq 0, \quad (3.11)$$

то среднеквадратическое значение переменной $y(k)$ — это тоже неотрицательная величина

$$\begin{aligned} E\{|y(k)|^2\} &= E\{\mathbf{h}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{h}_N\} = \\ &= \mathbf{h}_N^H E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} \mathbf{h}_N = \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N \geq 0. \end{aligned} \quad (3.12)$$

Неравенство (3.12) означает, что матрица \mathbf{R}_N является положительно полуопределенной [71]. Если $\mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N > 0$, то матрица \mathbf{R}_N — положительно определенная.

На практике матрица \mathbf{R}_N является положительно определенной за исключением редко встречающихся случаев, когда сигналы в векторе (3.1) линейно-зависимы [20]. Положительная определенность означает, что матрица невырожденная (неособенная), т. е. ее определитель $\det\{\mathbf{R}_N\} > 0$, а значит, существует обратная матрица \mathbf{R}_N^{-1} , так как

$$\mathbf{R}_N^{-1} = \frac{1}{\det\{\mathbf{R}_N\}} \hat{\mathbf{R}}_N = \frac{1}{\det\{\mathbf{R}_N\}} \begin{bmatrix} \hat{R}_{11} & \hat{R}_{21} & \dots & \hat{R}_{N1} \\ \hat{R}_{12} & \hat{R}_{22} & \dots & \hat{R}_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{R}_{1N} & \hat{R}_{2N} & \dots & \hat{R}_{NN} \end{bmatrix}, \quad (3.13)$$

где $\hat{\mathbf{R}}_N$ — присоединенная матрица, составленная из алгебраических дополнений матрицы \mathbf{R}_N^T [72]. Для получения присоединенной матрицы нужно транспонировать исходную матрицу \mathbf{R}_N , а затем все элементы транспонированной матрицы \mathbf{R}_N^T заменить алгебраическими дополнениями. Аналогичный результат получается и в случае, если исходную матрицу заменить алгебраическими дополнениями, а затем транспонировать. Алгебраическим дополнением элемента a_{nm} квадратной матрицы \mathbf{A}_N называется число $A_{nm} = (-1)^{n+m} M_{nm}$, где M_{nm} — минор, т. е. определитель матрицы порядка $N-1$, получающейся путем вычеркивания n -й строки и m -го столбца из матрицы \mathbf{A}_N .

Свойство $\det\{\mathbf{R}_N\} > 0$ можно продемонстрировать для случая $N=2$, т. е. на примере матрицы $\mathbf{R}_N = \begin{bmatrix} R(0) & R(1) \\ R^*(1) & R(0) \end{bmatrix}$. Для этой матрицы $\det\{\mathbf{R}_N\} = R(0)R(0) - R(1)R^*(1) > 0$, так как $R(0)R(0) = R^2(0)$, $R(1)R^*(1) = |R(1)|^2$.

и $R^2(0) > |R(1)|^2$, поскольку $R^2(0)$ — это квадрат центрального отсчета взаимокорреляционной функции, $|R(1)|^2$ — это квадрат модуля первого бокового отсчета этой функции, а боковые отсчеты по модулю всегда меньше ее центрального отсчета.

Рассмотренный пример объясняет плохую обусловленность (т. е. плохую обратимость) матрицы \mathbf{R}_N коррелированных сигналов, для которых $\det\{\mathbf{R}_N\} \rightarrow 0$, а значит, операция деления в уравнении (3.13) может быть сопряжена с вычислительными трудностями. Действительно, для коррелированных процессов значения $R^2(0)$ и $|R(1)|^2$ являются близкими по величине, а потому разность этих величин является малой положительной величиной, обратное значение которой возрастает по мере уменьшения разности. Аналогично можно показать, что определитель $\det\{\mathbf{R}_N\} > 0$ для любых $N > 2$.

Свойство 4. Корреляционная матрица сигналов многоканального фильтра может быть представлена в виде клеточной матрицы.

Из уравнений (3.4) и (3.5) следует, что корреляционная матрица сигналов многоканального фильтра (см. рис. 1.3) определяется как

$$\begin{aligned}
 \mathbf{R}_N &= E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} = E\left\{\begin{bmatrix} \mathbf{x}_{N_1}(k) \\ \mathbf{x}_{N_2}(k) \\ \vdots \\ \mathbf{x}_{N_M}(k) \end{bmatrix} \left[\mathbf{x}_{N_1}^H(k), \mathbf{x}_{N_2}^H(k), \dots, \mathbf{x}_{N_M}^H(k) \right] \right\} = \\
 &= E\left\{\begin{bmatrix} \mathbf{x}_{N_1}(k)\mathbf{x}_{N_1}^H(k) & \mathbf{x}_{N_1}(k)\mathbf{x}_{N_2}^H(k) & \dots & \mathbf{x}_{N_1}(k)\mathbf{x}_{N_M}^H(k) \\ \mathbf{x}_{N_2}(k)\mathbf{x}_{N_1}^H(k) & \mathbf{x}_{N_2}(k)\mathbf{x}_{N_2}^H(k) & \dots & \mathbf{x}_{N_2}(k)\mathbf{x}_{N_M}^H(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{N_M}(k)\mathbf{x}_{N_1}^H(k) & \mathbf{x}_{N_M}(k)\mathbf{x}_{N_2}^H(k) & \dots & \mathbf{x}_{N_M}(k)\mathbf{x}_{N_M}^H(k) \end{bmatrix}\right\} = \\
 &= \begin{bmatrix} E\{\mathbf{x}_{N_1}(k)\mathbf{x}_{N_1}^H(k)\} & E\{\mathbf{x}_{N_1}(k)\mathbf{x}_{N_2}^H(k)\} & \dots & E\{\mathbf{x}_{N_1}(k)\mathbf{x}_{N_M}^H(k)\} \\ E\{\mathbf{x}_{N_2}(k)\mathbf{x}_{N_1}^H(k)\} & E\{\mathbf{x}_{N_2}(k)\mathbf{x}_{N_2}^H(k)\} & \dots & E\{\mathbf{x}_{N_2}(k)\mathbf{x}_{N_M}^H(k)\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{\mathbf{x}_{N_M}(k)\mathbf{x}_{N_1}^H(k)\} & E\{\mathbf{x}_{N_M}(k)\mathbf{x}_{N_2}^H(k)\} & \dots & E\{\mathbf{x}_{N_M}(k)\mathbf{x}_{N_M}^H(k)\} \end{bmatrix} = \\
 &= \begin{bmatrix} \mathbf{R}_{N_1} & \mathbf{R}_{N_1N_2} & \dots & \mathbf{R}_{N_1N_M} \\ \mathbf{R}_{N_1N_2}^H & \mathbf{R}_{N_2} & \dots & \mathbf{R}_{N_2N_M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{N_1N_M}^H & \mathbf{R}_{N_2N_{M-1}}^H & \dots & \mathbf{R}_{N_M} \end{bmatrix}. \tag{3.14}
 \end{aligned}$$

Из (3.14) видно, что на главной диагонали корреляционной матрицы многоканального фильтра располагаются квадратные корреляционные матрицы входных сигналов отдельных каналов, а остальные элементы являются матрицами взаимной корреляции сигналов между каналами. Структура

матрицы (3.14) свидетельствует о том, что при разработке алгоритмов адаптивной фильтрации следует учитывать взаимосвязь между сигналами каналов фильтра.

3.3. Собственные числа и собственные векторы корреляционной матрицы

Одно из важных свойств корреляционной матрицы состоит в том, что эта матрица может быть разложена по собственным числам и соответствующим им собственным векторам.

Число λ называется собственным числом матрицы \mathbf{R}_N , если выполняется условие

$$\mathbf{R}_N \mathbf{q}_N = \lambda \mathbf{q}_N, \quad (3.15)$$

где $\mathbf{q}_N \neq \mathbf{0}_N$ — собственный вектор, соответствующий этому собственному числу. Уравнение (3.15) означает, что эрмитова матрица \mathbf{R}_N осуществляет линейное преобразование вектора \mathbf{q}_N в вектор $\lambda \mathbf{q}_N$, т. е. в вектор, совпадающий по направлению с исходным непреобразованным вектором, так как далее будет показано, что собственные числа матрицы являются действительными и неотрицательными.

Уравнение (3.15) можно представить как

$$\mathbf{R}_N \mathbf{q}_N - \lambda \mathbf{q}_N = (\mathbf{R}_N - \lambda \mathbf{I}_N) \mathbf{q}_N = \mathbf{0}_N. \quad (3.16)$$

Уравнение (3.16) является однородным линейным уравнением с неизвестным \mathbf{q}_N . Оно имеет нетривиальное решение $\mathbf{q}_N \neq \mathbf{0}_N$, только если определитель

$$\det\{\mathbf{R}_N - \lambda \mathbf{I}_N\} = 0. \quad (3.17)$$

Уравнение (3.17) называется характеристическим уравнением матрицы \mathbf{R}_N и представляет собой полином N -й степени, т. е.

$$p(\lambda) = \lambda^N + a_{N-1} \lambda^{N-1} + \dots + a_1 \lambda + a_0 = 0. \quad (3.18)$$

Например, при $N=2$

$$\mathbf{R}_N - \lambda \mathbf{I}_N = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{11} - \lambda & R_{12} \\ R_{21} & R_{22} - \lambda \end{bmatrix},$$

а уравнение (3.17) имеет вид

$$\begin{aligned} p(\lambda) &= \det\{\mathbf{R}_N - \lambda \mathbf{I}_N\} = (R_{11} - \lambda)(R_{22} - \lambda) - R_{12}R_{21} = \\ &= \lambda^2 - \lambda(R_{11} + R_{22}) + (R_{11}R_{22} - R_{12}R_{21}) = \lambda^2 + a_1 \lambda + a_0 = 0, \end{aligned}$$

т. е. является полиномом 2-й степени, где $a_1 = -(R_{11} + R_{22})$, $a_2 = R_{11}R_{22} - R_{12}R_{21}$.

Уравнение (3.18) имеет N корней (не обязательно разных) $\lambda_1, \lambda_2, \dots, \lambda_N$, называемых собственными числами матрицы \mathbf{R}_N .

Каждый вектор $\mathbf{q}_{N,n} \neq \mathbf{0}_N$, где $n = 1, 2, \dots, N$, удовлетворяющий уравнению

$$\mathbf{R}_N \mathbf{q}_{N,n} = \lambda_n \mathbf{q}_{N,n}, \quad (3.19)$$

называется собственным вектором, соответствующим собственному числу λ_n .

Каждый собственный вектор соответствует только одному собственному числу, в то время как одному собственному числу соответствует бесконечное множество собственных векторов. Действительно, если $\mathbf{q}_{N,n}$ — собственный вектор, соответствующий собственному числу λ_n , то вектор $a \mathbf{q}_{N,n}$ при $a \neq 0$ также является собственным вектором, так как уравнение $\mathbf{R}_N a \mathbf{q}_{N,n} = \lambda_n a \mathbf{q}_{N,n}$ после сокращения параметра a совпадает с уравнением (3.19).

Собственные числа λ_n и собственные векторы $\mathbf{q}_{N,n}$ корреляционной матрицы \mathbf{R}_N стационарного дискретного стохастического процесса характеризуются рядом свойств, основные из которых приводятся ниже.

Свойство 1. Если $\lambda_1, \lambda_2, \dots, \lambda_N$ — собственные числа корреляционной матрицы \mathbf{R}_N , то для любого целого числа $k > 0$ собственные числа матрицы $\mathbf{R}_N^k = \underbrace{\mathbf{R}_N \mathbf{R}_N \cdots \mathbf{R}_N}_k$ равны $\lambda_1^k, \lambda_2^k, \dots, \lambda_N^k$.

Доказательство этого утверждения строится на доказательстве справедливости выражения

$$\mathbf{R}_N^k \mathbf{q}_{N,n} = \lambda_n^k \mathbf{q}_{N,n}. \quad (3.20)$$

Это доказательство осуществляется следующим образом. Умножим обе части уравнения (3.19) слева на \mathbf{R}_N^{k-1} как

$$\mathbf{R}_N^{k-1} (\mathbf{R}_N \mathbf{q}_{N,n}) = \mathbf{R}_N^{k-1} (\lambda_n \mathbf{q}_{N,n}). \quad (3.21)$$

Далее, используя уравнения (3.21) и (3.19), можно выполнить последовательность преобразований левой части уравнения (3.20), приводящих к правой части этого уравнения. Действительно,

$$\begin{aligned} \mathbf{R}_N^k \mathbf{q}_{N,n} &= \mathbf{R}_N^{k-1} (\mathbf{R}_N \mathbf{q}_{N,n}) = \mathbf{R}_N^{k-1} (\lambda_n \mathbf{q}_{N,n}) = \lambda_n \mathbf{R}_N^{k-1} \mathbf{q}_{N,n} = \lambda_n \mathbf{R}_N^{k-2} (\mathbf{R}_N \mathbf{q}_{N,n}) = \\ &= \lambda_n \mathbf{R}_N^{k-2} (\lambda_n \mathbf{q}_{N,n}) = \lambda_n^2 \mathbf{R}_N^{k-2} \mathbf{q}_{N,n} = \lambda_n^2 \mathbf{R}_N^{k-3} (\mathbf{R}_N \mathbf{q}_{N,n}) = \lambda_n^2 \mathbf{R}_N^{k-3} (\lambda_n \mathbf{q}_{N,n}) = \\ &= \lambda_n^3 \mathbf{R}_N^{k-3} \mathbf{q}_{N,n} = \dots = \lambda_n^{k-1} \mathbf{R}_N^1 \mathbf{q}_{N,n} = \lambda_n^{k-1} \mathbf{R}_N^0 (\mathbf{R}_N \mathbf{q}_{N,n}) = \lambda_n^{k-1} \mathbf{R}_N^0 (\lambda_n \mathbf{q}_{N,n}) = \lambda_n^k \mathbf{q}_{N,n}. \end{aligned} \quad (3.22)$$

Свойство 2. Ненулевые собственные векторы $\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}$ корреляционной матрицы \mathbf{R}_N , соответствующие различным собственным числам $\lambda_1, \lambda_2, \dots, \lambda_N$, являются линейно-независимыми.

Векторы $\mathbf{q}_{N,n}$ являются линейно-зависимыми, если существуют такие $a_n \neq 0$, что

$$a_1 \mathbf{q}_{N,1} + a_2 \mathbf{q}_{N,2} + \cdots + a_N \mathbf{q}_{N,N} = \sum_{n=1}^N a_n \mathbf{q}_{N,n} = \mathbf{0}_N. \quad (3.23)$$

Если равенство (3.23) выполняется только при всех $a_n = 0$, то векторы $\mathbf{q}_{N,n}$ являются линейно-независимыми.

Для доказательства свойства 2 представим уравнение (3.23) как $\sum_{n=1}^N a_n \mathbf{R}_N^0 \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^0 \mathbf{q}_{N,n} = \mathbf{0}_N$. Затем, умножая $N-1$ раз правую часть этого уравнения на \mathbf{R}_N и используя уравнение (3.20), получим систему N уравнений

$$\left\{ \begin{array}{l} \sum_{n=1}^N a_n \mathbf{R}_N^0 \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^0 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \sum_{n=1}^N a_n \mathbf{R}_N \mathbf{R}_N^0 \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \mathbf{R}_N \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^1 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \sum_{n=1}^N a_n \mathbf{R}_N \mathbf{R}_N \mathbf{R}_N^0 \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \mathbf{R}_N^2 \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^2 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \vdots \\ \sum_{n=1}^N a_n \mathbf{R}_N \mathbf{R}_N^{k-2} \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \mathbf{R}_N^{k-1} \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^{k-1} \mathbf{q}_{N,n} = \mathbf{0}_N \\ \vdots \\ \sum_{n=1}^N a_n \mathbf{R}_N^{N-2} \mathbf{R}_N \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \mathbf{R}_N^{N-1} \mathbf{q}_{N,n} = \sum_{n=1}^N a_n \lambda_n^{N-1} \mathbf{q}_{N,n} = \mathbf{0}_N, \end{array} \right. \quad (3.24)$$

или, используя крайнюю правую часть этих уравнений, получим систему из N уравнений

$$\left\{ \begin{array}{l} \sum_{n=1}^N a_n \lambda_n^0 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \sum_{n=1}^N a_n \lambda_n^1 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \sum_{n=1}^N a_n \lambda_n^2 \mathbf{q}_{N,n} = \mathbf{0}_N \\ \vdots \\ \sum_{n=1}^N a_n \lambda_n^{k-1} \mathbf{q}_{N,n} = \mathbf{0}_N \\ \vdots \\ \sum_{n=1}^N a_n \lambda_n^{N-1} \mathbf{q}_{N,n} = \mathbf{0}_N, \end{array} \right. \quad (3.25)$$

которая может быть представлена как

$$\left[a_1 \mathbf{q}_{N,1}, a_2 \mathbf{q}_{N,2}, \dots, a_N \mathbf{q}_{N,N} \right] \mathbf{S}_N = \mathbf{O}_N, \quad (3.26)$$

где

$$\mathbf{S}_N = \begin{bmatrix} \lambda_1^0 & \lambda_1^1 & \lambda_1^2 & \cdots & \lambda_1^{N-1} \\ \lambda_2^0 & \lambda_2^1 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\ \lambda_3^0 & \lambda_3^1 & \lambda_3^2 & \cdots & \lambda_3^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_N^0 & \lambda_N^1 & \lambda_N^2 & \cdots & \lambda_N^{N-1} \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\ 1 & \lambda_3 & \lambda_3^2 & \cdots & \lambda_3^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{N-1} \end{bmatrix} \quad (3.27)$$

— матрица Вандермонда.

Так как согласно определению все собственные числа различны и не-нулевые (поскольку собственные векторы ненулевые), то $\det\{\mathbf{S}_N\} \neq 0$, а значит, существует обратная матрица \mathbf{S}_N^{-1} . Тогда из уравнения (3.26) следует, что $[\mathbf{a}_1 \mathbf{q}_{N,1}, \mathbf{a}_2 \mathbf{q}_{N,2}, \dots, \mathbf{a}_N \mathbf{q}_{N,N}] \mathbf{S}_N \mathbf{S}_N^{-1} = \mathbf{O}_N \mathbf{S}_N^{-1} = \mathbf{O}_N$. Поскольку $\mathbf{S}_N \mathbf{S}_N^{-1} = \mathbf{I}_N$, то

$$[\mathbf{a}_1 \mathbf{q}_{N,1}, \mathbf{a}_2 \mathbf{q}_{N,2}, \dots, \mathbf{a}_N \mathbf{q}_{N,N}] \mathbf{S}_N \mathbf{S}_N^{-1} = [\mathbf{a}_1 \mathbf{q}_{N,1}, \mathbf{a}_2 \mathbf{q}_{N,2}, \dots, \mathbf{a}_N \mathbf{q}_{N,N}] = \mathbf{O}_N. \quad (3.28)$$

Так как каждый столбец матрицы (3.28) $\mathbf{a}_n \mathbf{q}_{N,n} = \mathbf{0}_N$ и $\mathbf{q}_{N,n} \neq \mathbf{0}_N$, то равенство (3.28) выполняется, только если $a_n = 0$. Это согласно уравнению (3.23) свидетельствует о линейной независимости собственных векторов корреляционной матрицы \mathbf{R}_N .

Свойство 2 может быть использовано для представления произвольных векторов в виде линейной комбинации собственных векторов. Например,

$$\mathbf{h}_N = \sum_{n=1}^N a_n \mathbf{q}_{N,n} \quad (3.29)$$

означает, что собственные векторы $\mathbf{q}_{N,n}$ размера $N \times 1$ могут быть использованы в виде базиса для представления произвольного вектора \mathbf{h}_N с таким же размером $N \times 1$.

Если выражение (3.29) умножить слева и справа на матрицу \mathbf{R}_N как

$$\mathbf{R}_N \mathbf{h}_N = \sum_{n=1}^N a_n \mathbf{R}_N \mathbf{q}_{N,n}, \quad (3.30)$$

то, используя уравнение (3.19), получим, что

$$\mathbf{R}_N \mathbf{h}_N = \sum_{n=1}^N a_n (\mathbf{R}_N \mathbf{q}_{N,n}) = \sum_{n=1}^N a_n (\lambda_n \mathbf{q}_{N,n}). \quad (3.31)$$

Уравнение (3.31) показывает, что линейное преобразование (3.30) над вектором \mathbf{h}_N сводится лишь к масштабированию линейно независимых собственных векторов в уравнении (3.29) собственными числами этих векторов, так как линейная независимость собственных векторов при этом не нарушается.

Свойство 3. Собственные числа $\lambda_1, \lambda_2, \dots, \lambda_N$ корреляционной матрицы \mathbf{R}_N являются неотрицательными и действительными.

Для доказательства этого утверждения умножим уравнение (3.19) слева на вектор $\mathbf{q}_{N,n}^H$, т. е.

$$\mathbf{q}_{N,n}^H \mathbf{R}_N \mathbf{q}_{N,n} = \lambda_n \mathbf{q}_{N,n}^H \mathbf{q}_{N,n}. \quad (3.32)$$

Тогда согласно свойству 3 корреляционной матрицы для положительно полуопределенной корреляционной матрицы \mathbf{R}_N левая часть уравнения (3.32) $\mathbf{q}_{N,n}^H \mathbf{R}_N \mathbf{q}_{N,n} \geq 0$. В правой части уравнения (3.32) произведение $\mathbf{q}_{N,n}^H \mathbf{q}_{N,n} > 0$, так как $\mathbf{q}_{N,n} \neq \mathbf{0}_N$ и

$$\mathbf{q}_{N,n}^H \mathbf{q}_{N,n} = \begin{bmatrix} q_{1,n}^*, q_{2,n}^*, \dots, q_{N,n}^* \end{bmatrix} \begin{bmatrix} q_{1,n} \\ q_{2,n} \\ \vdots \\ q_{N,n} \end{bmatrix} = \sum_{m=1}^N q_{m,n}^* q_{m,n} = \sum_{m=1}^N |q_{m,n}|^2 > 0.$$

Поэтому из уравнения (3.32) следует, что

$$\lambda_n = \frac{\mathbf{q}_{N,n}^H \mathbf{R}_N \mathbf{q}_{N,n}}{\mathbf{q}_{N,n}^H \mathbf{q}_{N,n}} \geq 0. \quad (3.33)$$

Отношение (3.33) называется отношением Релея. Таким образом, собственные числа корреляционной матрицы всегда положительные и равны соотношениям Релея для ее собственных векторов, соответствующих этим числам.

Свойство 4. Собственные векторы $\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}$ корреляционной матрицы \mathbf{R}_N , соответствующие различным собственным числам $\lambda_1, \lambda_2, \dots, \lambda_N$, являются ортогональными друг другу.

Ортогональность означает, что

$$\mathbf{q}_{N,n}^H \mathbf{q}_{N,m} = 0, \quad (3.34)$$

если $n \neq m$.

Запишем уравнение (3.19) для двух значений индексов $n \neq m$, т. е.

$$\mathbf{R}_N \mathbf{q}_{N,n} = \lambda_n \mathbf{q}_{N,n}, \quad (3.35)$$

$$\mathbf{R}_N \mathbf{q}_{N,m} = \lambda_m \mathbf{q}_{N,m}. \quad (3.36)$$

Поскольку согласно свойству 1 корреляционной матрицы $\mathbf{R}_N = \mathbf{R}_N^H$, а собственные числа являются действительными согласно их свойству 3, то эрмитово сопряжение уравнения (3.35) $(\mathbf{R}_N \mathbf{q}_{N,n})^H = (\lambda_n \mathbf{q}_{N,n})^H$ приводит к уравнению

$$\mathbf{q}_{N,n}^H \mathbf{R}_N = \lambda_n \mathbf{q}_{N,n}^H. \quad (3.37)$$

Здесь использовано свойство произведения матриц (3.9).

Умножая уравнение (3.37) на вектор $\mathbf{q}_{N,m}$ справа и умножая уравнение (3.36) на вектор $\mathbf{q}_{N,n}^H$ слева, получим

$$\mathbf{q}_{N,n}^H \mathbf{R}_N \mathbf{q}_{N,m} = \lambda_n \mathbf{q}_{N,n}^H \mathbf{q}_{N,m}, \quad (3.38)$$

$$\mathbf{q}_{N,n}^H \mathbf{R}_N \mathbf{q}_{N,m} = \lambda_m \mathbf{q}_{N,n}^H \mathbf{q}_{N,m}, \quad (3.39)$$

откуда следует, что

$$\lambda_n \mathbf{q}_{N,n}^H \mathbf{q}_{N,m} = \lambda_m \mathbf{q}_{N,n}^H \mathbf{q}_{N,m}. \quad (3.40)$$

Так как $\lambda_n \neq \lambda_m$, то равенство (3.40) выполняется, если только $\mathbf{q}_{N,n}^H \mathbf{q}_{N,m} = 0$, что согласно равенству (3.34) означает ортогональность собственных векторов $\mathbf{q}_{N,n}$ и $\mathbf{q}_{N,m}$.

Как уже отмечалось, каждому собственному числу λ_n соответствует бесконечное множество собственных векторов $a \mathbf{q}_{N,n}$ при $a \neq 0$. Поэтому без потери общности в дальнейшем будем рассматривать только нормированные собственные векторы, т. е. такие, для которых выполняется условие

$$\mathbf{q}_{N,n}^H \mathbf{q}_{N,m} = \left[q_{1,n}^*, q_{2,n}^*, \dots, q_{N,n}^* \right] \begin{bmatrix} q_{1,m} \\ q_{2,m} \\ \vdots \\ q_{N,m} \end{bmatrix} = \sum_{i=1}^N q_{i,n}^* q_{i,m} = \begin{cases} 1, & \text{при } n = m \\ 0, & \text{при } n \neq m \end{cases} \quad (3.41)$$

для всех $n = 1, \dots, N$ и $m = 1, \dots, N$. Векторы, удовлетворяющие условию (3.41), называются ортонормированными.

Свойство 5. Если собственные векторы $\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}$ корреляционной матрицы \mathbf{R}_N , соответствующие различным собственным числам $\lambda_1, \lambda_2, \dots, \lambda_N$, представить матрицей $\mathbf{Q}_N = [\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}]$ ортонормированных векторов (3.41), а собственные числа — диагональной матрицей

$$\Lambda_N = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_N \end{bmatrix}, \quad (3.42)$$

то матрица \mathbf{R}_N может быть приведена к диагональному виду как $\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \Lambda_N$, что доказывается следующим образом.

Условие (3.41) означает, что векторы $\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}$ образуют ортонормированное множество.

Используя определение матрицы \mathbf{Q}_N , систему уравнений (3.19) можно представить в виде матричного уравнения

$$\mathbf{R}_N \mathbf{Q}_N = \mathbf{Q}_N \Lambda_N. \quad (3.43)$$

Согласно уравнению (3.41)

$$\begin{aligned}
 \mathbf{Q}_N^H \mathbf{Q}_N &= \begin{bmatrix} \mathbf{q}_{N,1}^H \\ \mathbf{q}_{N,2}^H \\ \vdots \\ \mathbf{q}_{N,N}^H \end{bmatrix} \begin{bmatrix} \mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N} \end{bmatrix} = \\
 &= \begin{bmatrix} \mathbf{q}_{N,1}^H \mathbf{q}_{N,1} & \mathbf{q}_{N,1}^H \mathbf{q}_{N,2} & \cdots & \mathbf{q}_{N,1}^H \mathbf{q}_{N,N} \\ \mathbf{q}_{N,2}^H \mathbf{q}_{N,1} & \mathbf{q}_{N,2}^H \mathbf{q}_{N,2} & \cdots & \mathbf{q}_{N,2}^H \mathbf{q}_{N,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_{N,N}^H \mathbf{q}_{N,1} & \mathbf{q}_{N,N}^H \mathbf{q}_{N,2} & \cdots & \mathbf{q}_{N,N}^H \mathbf{q}_{N,N} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}_N,
 \end{aligned} \tag{3.44}$$

откуда

$$\mathbf{Q}_N^H = \mathbf{Q}_N^{-1}. \tag{3.45}$$

Из равенства (3.45) следует, что $\mathbf{Q}_N \mathbf{Q}_N^H = \mathbf{Q}_N \mathbf{Q}_N^{-1} = \mathbf{I}_N$. Комплексная матрица, обладающая свойством (3.45), называется унитарной, а аналогичная действительная матрица — ортонормированной [71].

Если уравнение (3.43) умножить слева на матрицу \mathbf{Q}_N^H , т.е. $\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{Q}_N^H \mathbf{Q}_N \Lambda_N$, то, используя уравнение (3.44), получим, что

$$\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \Lambda_N. \tag{3.46}$$

Если уравнение (3.43) умножить справа на матрицу \mathbf{Q}_N^H , т.е. $\mathbf{R}_N \mathbf{Q}_N \mathbf{Q}_N^H = \mathbf{Q}_N \Lambda_N \mathbf{Q}_N^H$, то, используя уравнение (3.44), получим, что

$$\mathbf{R}_N = \mathbf{Q}_N \Lambda_N \mathbf{Q}_N^H = \begin{bmatrix} \mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_N \end{bmatrix} \begin{bmatrix} \mathbf{q}_{N,1}^H \\ \mathbf{q}_{N,2}^H \\ \vdots \\ \mathbf{q}_{N,N}^H \end{bmatrix} = \sum_{n=1}^N \lambda_n \mathbf{q}_{N,n} \mathbf{q}_{N,n}^H. \tag{3.47}$$

Уравнение (3.47) является спектральным разложением матрицы \mathbf{R}_N или разложением по собственным числам и собственным векторам.

Аналогично, используя равенство (3.20) как $\mathbf{R}_N^k \mathbf{Q}_N = \mathbf{Q}_N \Lambda_N^k$, можно показать, что

$$\mathbf{R}_N^k \mathbf{Q}_N \mathbf{Q}_N^H = \mathbf{Q}_N \Lambda_N^k \mathbf{Q}_N^H = \sum_{n=1}^N \lambda_n^k \mathbf{q}_{N,n} \mathbf{q}_{N,n}^H = \mathbf{R}_N^k. \tag{3.48}$$

Из (3.47) следует, что среднее значение квадрата модуля выходного сигнала адаптивного фильтра определяется как

$$\begin{aligned}
 E\{|y(k)|^2\} &= E\{y(k)y^*(k)\} = E\{\mathbf{h}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H \mathbf{h}_N\} = \mathbf{h}_N^H E\{\mathbf{x}_N(k) \mathbf{x}_N^H\} \mathbf{h}_N = \\
 &= \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N = \mathbf{h}_N^H \left[\sum_{n=1}^N \lambda_n \mathbf{q}_{N,n} \mathbf{q}_{N,n}^H \right] \mathbf{h}_N = \sum_{n=1}^N \lambda_n \mathbf{h}_N^H \mathbf{q}_{N,n} \mathbf{q}_{N,n}^H \mathbf{h}_N = \sum_{n=1}^N \lambda_n |\mathbf{h}_N^H \mathbf{q}_{N,n}|^2,
 \end{aligned} \tag{3.49}$$

т. е. оно равно взвешенной с помощью собственных чисел сумме квадратов модулей скалярных произведений вектора ВК фильтра и собственных векторов корреляционной матрицы его сигналов.

Используя равенство (3.45) и свойство обратимых квадратных матриц [71]

$$[\mathbf{A}_N \mathbf{B}_N]^{-1} = \mathbf{B}_N^{-1} \mathbf{A}_N^{-1}, \quad (3.50)$$

из (3.47) можно также определить обратную матрицу \mathbf{R}_N^{-1} как

$$\begin{aligned} \mathbf{R}_N^{-1} &= (\mathbf{Q}_N \mathbf{\Lambda}_N \mathbf{Q}_N^H)^{-1} = \mathbf{Q}_N \mathbf{\Lambda}_N^{-1} \mathbf{Q}_N^H = \\ &= [\mathbf{q}_{N,1}, \mathbf{q}_{N,2}, \dots, \mathbf{q}_{N,N}] \begin{bmatrix} \lambda_1^{-1} & 0 & \dots & 0 \\ 0 & \lambda_2^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_N^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{q}_{N,1}^H \\ \mathbf{q}_{N,2}^H \\ \vdots \\ \mathbf{q}_{N,N}^H \end{bmatrix} = \sum_{n=1}^N \lambda_n^{-1} \mathbf{q}_{N,n} \mathbf{q}_{N,n}^H. \end{aligned} \quad (3.51)$$

Свойство 6. Сумма собственных чисел $\lambda_1, \lambda_2, \dots, \lambda_N$ корреляционной матрицы \mathbf{R}_N равна следу этой матрицы.

Следом квадратной матрицы \mathbf{A}_N называется сумма ее диагональных элементов $\text{tr}\{\mathbf{A}_N\} = \sum_{n=1}^N a_{nn}$. Из линейной алгебры известно [71], что

$$\text{tr}\{\mathbf{A}_N \mathbf{B}_N\} = \text{tr}\{\mathbf{B}_N \mathbf{A}_N\}. \quad (3.52)$$

Тогда, используя уравнения (3.46) и (3.44), получим, что

$$\text{tr}\{\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N\} = \text{tr}\{\mathbf{R}_N \mathbf{Q}_N \mathbf{Q}_N^H\} = \text{tr}\{\mathbf{R}_N\} = \text{tr}\{\mathbf{\Lambda}_N\} = \sum_{n=1}^N \lambda_n. \quad (3.53)$$

Свойство 7. Произведение собственных чисел $\lambda_1, \lambda_2, \dots, \lambda_N$ корреляционной матрицы \mathbf{R}_N равно определителю этой матрицы.

Из линейной алгебры также известно, что

$$\det\{\mathbf{A}_N \mathbf{B}_N\} = \det\{\mathbf{A}_N\} \det\{\mathbf{B}_N\}. \quad (3.54)$$

Тогда используя (3.46) и (3.44), получим, что

$$\begin{aligned} \det\{\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N\} &= \det\{\mathbf{Q}_N^H\} \det\{\mathbf{R}_N\} \det\{\mathbf{Q}_N\} = \\ &= \det\{\mathbf{R}_N\} \det\{\mathbf{Q}_N^H\} \det\{\mathbf{Q}_N\} = \det\{\mathbf{R}_N\} \det\{\mathbf{Q}_N^H \mathbf{Q}_N\} = \\ &= \det\{\mathbf{R}_N\} \det\{\mathbf{I}_N\} = \det\{\mathbf{R}_N\} = \det\{\mathbf{\Lambda}_N\} = \prod_{n=1}^N \lambda_n. \end{aligned} \quad (3.55)$$

Рассмотренные свойства корреляционной матрицы, ее собственных чисел и собственных векторов используются при анализе процессов адаптивной фильтрации сигналов.

3.4. Винеровская фильтрация

В главе 1 отмечалось, что одной из целевых функций, широко используемых в теории адаптивной фильтрации, является MSE-функция, т.е. функция среднеквадратической ошибки, определяемая как $\xi = E\{ |e(k)|^2 \}$.

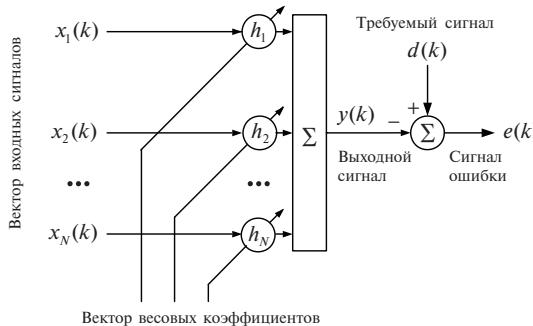


Рис. 3.2. Линейный сумматор: формирование сигнала ошибки

Для линейного сумматора (рис. 3.2) MSE-функция определяется как

$$\begin{aligned} \xi &= E\{ |e(k)|^2 \} = E\{ e(k)e^*(k) \} = E\{ [d(k) - y(k)][d(k) - y(k)]^* \} = \\ &= E\{ d(k)d^*(k) - y(k)d^*(k) - d(k)y^*(k) + y(k)y^*(k) \}, \end{aligned} \quad (3.56)$$

где $e(k) = d(k) - y(k)$ и $y(k) = \mathbf{h}_N^H \mathbf{x}_N(k)$. Уравнение (3.56) может быть преобразовано к виду

$$\begin{aligned} \xi &= E\{ d(k)d^*(k) \} - E\{ y(k)d^*(k) \} - E\{ d(k)y^*(k) \} + E\{ y(k)y^*(k) \} = \\ &= E\{ d(k)d^*(k) \} - \mathbf{h}_N^H E\{ \mathbf{x}_N(k)d^*(k) \} - E\{ d(k)\mathbf{x}_N^H(k) \} \mathbf{h}_N + \mathbf{h}_N^H E\{ \mathbf{x}_N(k)\mathbf{x}_N^H(k) \} \mathbf{h}_N. \end{aligned} \quad (3.57)$$

Используя обозначения $\sigma_d^2 = E\{ d(k)d^*(k) \}$, $\mathbf{r}_N = E\{ \mathbf{x}_N(k)d^*(k) \}$ и $\mathbf{R}_N = E\{ \mathbf{x}_N(k)\mathbf{x}_N^H(k) \}$, MSE-функция (3.57) линейного сумматора с постоянными ВК может быть представлена как

$$\xi = E\{ |e(k)|^2 \} = \sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N, \quad (3.58)$$

где σ_d^2 — дисперсия (средняя мощность) требуемого сигнала, \mathbf{r}_N — вектор взаимной корреляции между входными и требуемым сигналами, \mathbf{R}_N — корреляционная матрица входных сигналов (3.5). Вектор \mathbf{r}_N определяется как

$$\mathbf{r}_N = E\left\{ \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} d^*(k) \right\} = \begin{bmatrix} E\{ x_1(k)d^*(k) \} \\ E\{ x_2(k)d^*(k) \} \\ \vdots \\ E\{ x_N(k)d^*(k) \} \end{bmatrix}. \quad (3.59)$$

Вектор градиента (см. главу 2) MSE-функции (3.58) определяется как

$$\nabla_{\mathbf{h}_N^*} \xi = \frac{\partial \xi}{\partial \mathbf{h}_N^*} = \left[\frac{\partial \xi}{\partial h_1^*}, \frac{\partial \xi}{\partial h_2^*}, \dots, \frac{\partial \xi}{\partial h_N^*} \right]^T = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N. \quad (3.60)$$

В случае действительных ВК вектор градиента определяется как

$$\nabla_{\tilde{\mathbf{h}}_N} \xi = -2\mathbf{r}_N + 2\mathbf{R}_N \tilde{\mathbf{h}}_N. \quad (3.61)$$

Полагая, что матрица \mathbf{R}_N является обратимой, условие

$$\nabla_{\tilde{\mathbf{h}}_N} \xi = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N = \mathbf{0}_N \quad (3.62)$$

позволяет найти оптимальный вектор ВК

$$\mathbf{h}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N, \quad (3.63)$$

соответствующий минимуму целевой функции (3.58). Уравнение (3.62) называется уравнением Винера—Хопфа, вектор ВК, определяемый уравнением 3.63, — оптимальным винеровским решением, а фильтр (см. рис. 3.2) с этим ВК — оптимальным винеровским фильтром.

На практике точные значения \mathbf{r}_N и \mathbf{R}_N обычно неизвестны. Однако если обрабатываемые сигналы являются эргодическими случайными процессами (т. е. процессами, у которых средние по времени значения равны их средним статистическим значениям), то, используя усреднение по времени, можно оценить значения \mathbf{r}_N и \mathbf{R}_N , что и осуществляется в большинстве адаптивных алгоритмов.

Используя уравнение (3.63) в (3.58), можно определить минимальное значение MSE-функции

$$\begin{aligned} \xi_{\min} &= E \left\{ |e(k)|^2 \right\} = \sigma_d^2 - \mathbf{h}_{N,o}^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_{N,o} = \\ &= \sigma_d^2 - \mathbf{h}_{N,o}^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{R}_N^{-1} \mathbf{r}_N = \\ &= \sigma_d^2 - \mathbf{h}_{N,o}^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{r}_N = \sigma_d^2 - \mathbf{r}_N^H \mathbf{h}_{N,o} = \sigma_d^2 - \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{r}_N. \end{aligned} \quad (3.64)$$

Вектор градиента (3.60) можно выразить и как

$$\begin{aligned} \nabla_{\mathbf{h}_N^*} \xi &= \frac{\partial \xi}{\partial \mathbf{h}_N^*} = \frac{\partial E \left\{ |e(k)|^2 \right\}}{\partial \mathbf{h}_N^*} = \frac{\partial E \left\{ e(k) e^*(k) \right\}}{\partial \mathbf{h}_N^*} = E \left\{ \frac{\partial [e(k) e^*(k)]}{\partial \mathbf{h}_N^*} \right\} = \\ &= E \left\{ e(k) \frac{\partial e^*(k)}{\partial \mathbf{h}_N^*} \right\} + E \left\{ \frac{\partial e(k)}{\partial \mathbf{h}_N^*} e^*(k) \right\} = E \left\{ \frac{\partial [d(k) - y(k)]}{\partial \mathbf{h}_N^*} e^*(k) \right\} = \\ &= E \left\{ \frac{\partial [d(k) - \mathbf{h}_N^H(k) \mathbf{x}_N(k)]}{\partial \mathbf{h}_N^*} e^*(k) \right\} = -E \left\{ \mathbf{x}_N(k) e^*(k) \right\}, \end{aligned} \quad (3.65)$$

так как

$$E \left\{ e(k) \frac{\partial e^*(k)}{\partial \mathbf{h}_N^*} \right\} = E \left\{ e(k) \frac{\partial [d^*(k) - y^*(k)]}{\partial \mathbf{h}_N^*} \right\} = E \left\{ e(k) \frac{\partial [d^*(k) - \mathbf{x}_N^H(k) \mathbf{h}_N(k)]}{\partial \mathbf{h}_N^*} \right\} = \mathbf{0}_N.$$

В случае действительных ВК вектор градиента определяется как

$$\nabla_{\mathbf{h}_N} \xi = -2E\{\mathbf{x}_N(k)e(k)\}. \quad (3.66)$$

Если требуемый сигнал определяется как

$$d(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k), \quad (3.67)$$

т. е. в нем присутствует аддитивный шум $e_o(k)$, часто называемый измерительным, и число ВК линейной системы $\mathbf{h}_{N,o}$, порождающей сигнал $d(k)$, совпадает с числом ВК линейного сумматора \mathbf{h}_N (см. рис. 3.2), то при $\mathbf{h}_N = \mathbf{h}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N$ ошибка $e(k)$ определяется как $e(k) = d(k) - y(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k) - \mathbf{h}_{N,o}^H \mathbf{x}_N(k) = e_o(k)$, а согласно уравнениям (3.65) и (3.60) градиент равен нуль-вектору, так как

$$\nabla_{\mathbf{h}_N} \xi = -E\{\mathbf{x}_N(k)e_o^*(k)\} = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N = -\mathbf{r}_N + \mathbf{R}_N \mathbf{R}_N^{-1} \mathbf{r}_N = -\mathbf{r}_N + \mathbf{r}_N = \mathbf{0}_N. \quad (3.68)$$

Уравнение (3.68) означает, что при $\mathbf{h}_N = \mathbf{h}_{N,o}$ сигнал ошибки ортогонален компонентам вектора входного сигнала, т. е. $E\{\mathbf{x}_N(k-n+1)e_o^*(k)\} = 0$, $n=1, 2, \dots, N$, а также ортогонален выходному сигналу фильтра, так как

$$\mathbf{h}_{N,o}^H E\{\mathbf{x}_N(k)e_o^*(k)\} = E\{\mathbf{h}_{N,o}^H \mathbf{x}_N(k)e_o^*(k)\} = E\{y_o(k)e_o^*(k)\} = 0. \quad (3.69)$$

Уравнения (3.68) и (3.69) известны как принцип ортогональности, означающий, что при $\mathbf{h}_N = \mathbf{h}_{N,o}$ сигнал ошибки фильтра ортогонален к его входным и выходным сигналам, т. е. сигнал ошибки и эти сигналы некоррелированы.

С учетом уравнений (3.67) и (3.68) дисперсия требуемого сигнала σ_d^2 определяется как

$$\begin{aligned} \sigma_d^2 &= E\{d(k)d^*(k)\} = \\ &= \mathbf{h}_{N,o}^H E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} \mathbf{h}_{N,o} + E\{e_o(k)\mathbf{x}_N^H(k)\} \mathbf{h}_N + \\ &+ \mathbf{h}_{N,o}^H E\{\mathbf{x}_N(k)e_o^*(k)\} + E\{e_o(k)e_o^*(k)\} = \\ &= \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_{N,o} + \sigma_o^2 = \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{R}_N \mathbf{R}_N^{-1} \mathbf{r}_N + \sigma_o^2 = \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{r}_N + \sigma_o^2, \end{aligned} \quad (3.70)$$

где $\sigma_o^2 = E\{e_o(k)e_o^*(k)\}$ — дисперсия (средняя мощность) измерительного шума. В этом случае согласно уравнению (3.64)

$$\xi_{\min} = \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{r}_N + \sigma_o^2 - \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{r}_N = \sigma_o^2, \quad (3.71)$$

а при $\sigma_o^2 = 0$ значение $\xi_{\min} = 0$.

Природа шума $e_o(k)$ определяется приложением адаптивного фильтра. Например, в компенсаторах сигналов акустического эха — это шум окружающей среды, в компенсаторах электрического эха — шум канала связи. Шум $e_o(k)$ также может порождаться, если число ВК N_h адаптивного фильтра, т. е. линейного сумматора (см. рис. 3.2), не совпадает с числом ВК N_w идентифицируемой линейной системы, см. уравнение (1.6).

3.5. Поверхность среднеквадратической ошибки

MSE-функция (3.58), т. е. среднеквадратическая ошибка, является квадратичной функцией ВК \mathbf{h}_N . Она образует в N -мерном пространстве ВК эллиптический параболоид. Его поверхность вогнута и содержит только положительные значения. Пример MSE-функции при $N = 2$ показан на рис. 3.3.

При исследовании свойств поверхности среднеквадратической ошибки MSE-функцию удобно выражать в терминах смещенных векторов ВК $\Delta\mathbf{h}_N = \mathbf{h}_N - \mathbf{h}_{N,o}$, т. е. в системе координат, началом которой является точка $\mathbf{h}_{N,o}$. Для этого уравнение (3.58) необходимо преобразовать как

$$\begin{aligned}
 \xi &= \sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{r}_N^H \mathbf{h}_{N,o} = \\
 &= \sigma_d^2 - \mathbf{h}_N^H \mathbf{R}_N \mathbf{R}_N^{-1} \mathbf{r}_N - \mathbf{r}_N^H \mathbf{R}_N^{-H} \mathbf{R}_N \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{R}_N^H \mathbf{h}_{N,o} = \\
 &= \sigma_d^2 - \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_{N,o} - \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_{N,o} = \\
 &= \sigma_d^2 - \mathbf{r}_N^H \mathbf{h}_{N,o} + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N - \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_N - \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_{N,o} + \mathbf{h}_{N,o}^H \mathbf{R}_N \mathbf{h}_{N,o} = \\
 &= \xi_{\min} + (\mathbf{h}_N - \mathbf{h}_{N,o})^H \mathbf{R}_N (\mathbf{h}_N - \mathbf{h}_{N,o}) = \xi_{\min} + \Delta\mathbf{h}_N^H \mathbf{R}_N \Delta\mathbf{h}_N.
 \end{aligned} \tag{3.72}$$

Здесь использованы свойства корреляционной матрицы $\mathbf{R}_N^H = \mathbf{R}_N$, $\mathbf{R}_N^{-H} = \mathbf{R}_N^{-1}$, $\mathbf{R}_N^{-H} \mathbf{R}_N = \mathbf{R}_N^{-1} \mathbf{R}_N = \mathbf{I}_N$, $\mathbf{R}_N \mathbf{R}_N^{-1} = \mathbf{I}_N$, уравнение (3.63) $\mathbf{h}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N$ и оно же в виде $\mathbf{r}_N^H = \mathbf{h}_{N,o}^H \mathbf{R}_N^H = \mathbf{h}_{N,o}^H \mathbf{R}_N$.

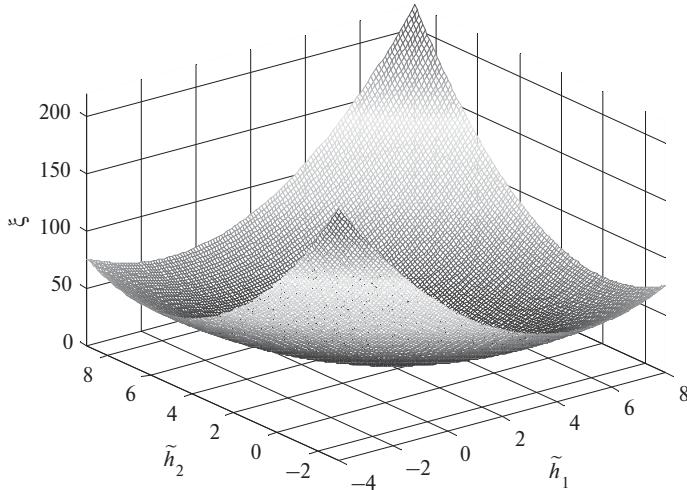


Рис. 3.3. Поверхность среднеквадратической ошибки

Смещенная система координат приведена на рис. 3.4. Для наглядности на рисунке показаны только две компоненты N -мерного векторного пространства. Здесь $\mathbf{e}_{N,h_n} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ — направляющие векторы исходной системы координат и $\mathbf{e}_{N,\Delta h_n} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ — направляющие векторы смещенной

системы координат, ненулевые единичные элементы которых находятся в n -й позиции, где $n=1, 2, \dots, N$. Из рис. 3.4 видно, что смещенная система координат образуется из исходной (несмещенной) системы координат путем ее параллельного переноса в точку многомерного пространства с координатами $\mathbf{h}_{N,o}$.

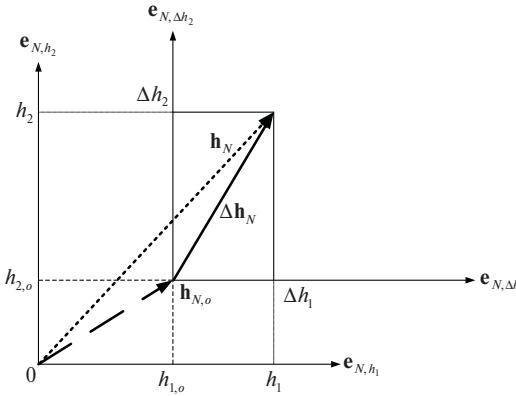


Рис. 3.4. Преобразование вектора \mathbf{h}_N в вектор $\Delta\mathbf{h}_N$

Используя (3.47), уравнение (3.72) можно также представить как

$$\begin{aligned}\xi &= \xi_{\min} + \Delta\mathbf{h}_N^H \mathbf{R}_N \Delta\mathbf{h}_N = \xi_{\min} + \Delta\mathbf{h}_N^H \mathbf{Q}_N \Lambda_N \mathbf{Q}_N^H \Delta\mathbf{h}_N = \\ &= \xi_{\min} + \mathbf{v}_N^H \Lambda_N \mathbf{v}_N = \xi_{\min} + \sum_{n=1}^N \lambda_n v_n^* v_n = \xi_{\min} + \sum_{n=1}^N \lambda_n |v_n|^2,\end{aligned}\quad (3.73)$$

где

$$\mathbf{v}_N = \mathbf{Q}_N^H \Delta\mathbf{h}_N = \begin{bmatrix} \mathbf{q}_{N,1}^H \\ \mathbf{q}_{N,2}^H \\ \vdots \\ \mathbf{q}_{N,N}^H \end{bmatrix} \Delta\mathbf{h}_N = \begin{bmatrix} \mathbf{q}_{N,1}^H \Delta\mathbf{h}_N \\ \mathbf{q}_{N,2}^H \Delta\mathbf{h}_N \\ \vdots \\ \mathbf{q}_{N,N}^H \Delta\mathbf{h}_N \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (3.74)$$

— вектор ВК в повернутой и смещенной системе координат (рис. 3.5).

Матрица \mathbf{Q}_N согласно уравнению (3.74) осуществляет поворот вектора $\Delta\mathbf{h}_N$, т. е. меняет лишь направление этого вектора, не меняя его нормы (модуля). Действительно, вектор \mathbf{v}_N отличается от вектора $\Delta\mathbf{h}_N$ только направлением, так как нормы обоих векторов равны между собой, т. е.

$$\|\mathbf{v}_N\|_2 = \sqrt{\mathbf{v}_N^H \mathbf{v}_N} = \sqrt{\Delta\mathbf{h}_N^H \mathbf{Q}_N \mathbf{Q}_N^H \Delta\mathbf{h}_N} = \sqrt{\Delta\mathbf{h}_N^H \Delta\mathbf{h}_N} = \|\Delta\mathbf{h}_N\|_2. \quad (3.75)$$

Так как поворот вектора эквивалентен повороту его системы координат, то в повернутой и смещенной системе координат вектор \mathbf{v}_N можно отображать в виде вектора, совпадающего с вектором $\Delta\mathbf{h}_N$ в смещенной системе координат.

Повернутая и смещенная система координат совпадает со смещенной в точку $\mathbf{h}_{N,o}$ системой координат, образованной ортонормированными столбцами матрицы \mathbf{Q}_N , являющимися собственными векторами $\mathbf{q}_{N,n}$ корреляционной матрицы

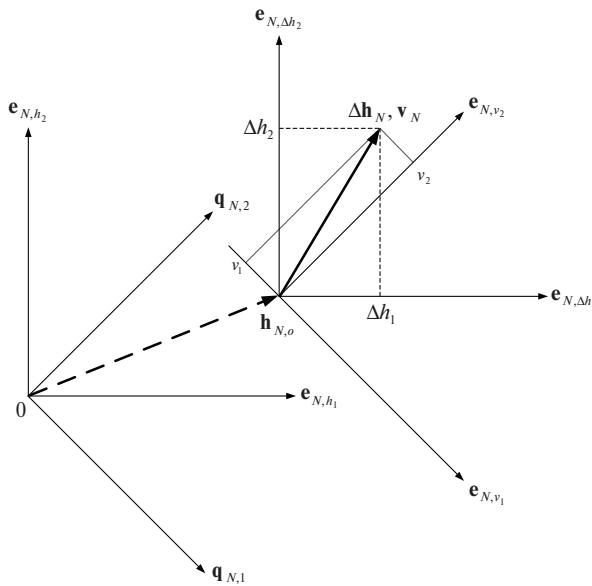


Рис. 3.5. Преобразование вектора Δh_N в вектор v_N

R_N , см. уравнение (3.15). Действительно, в повернутой системе координат векторы $q_{N,n}$ являются направляющими, так как согласно уравнению (3.41) эти векторы единичные, т. е. их нормы равны единице, поскольку $\|q_{N,n}\|_2 = \sqrt{q_{N,n}^H q_{N,n}} = 1$.

Таким образом, повернутая и смещенная система координат с направляющими векторами $e_{N,v_n} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ образуется путем поворота и смещения исходной системы координат с направляющими векторами $e_{N,h_n} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$, т. е. путем получения повернутой системы координат с направляющими векторами $q_{N,n} = [q_{1,n}, q_{2,n}, \dots, q_{N,n}]^T$, и ее последующего смещения в точку многомерного пространства с координатами $h_{N,o}$. Поворот исходной системы координат в направлении, противоположном направлению вращения вектора Δh_N , осуществляется с помощью преобразования (3.74).

При исследовании свойств адаптивных алгоритмов, базирующихся на минимизации MSE-функции, полезно использовать графики линий постоянного уровня $\xi = \text{const}$. Эти графики представляют собой гиперэллипсы, так как уравнение (3.73) $\xi = \xi_{\min} + \sum_{n=1}^N \lambda_n |v_n|^2 = \xi_{\min} + \lambda_1 |v_1|^2 + \lambda_2 |v_2|^2 + \dots + \lambda_N |v_N|^2$ при $\xi = \text{const}$ — это уравнение гиперэллипса

$$\frac{|v_1|^2}{a_1^2} + \frac{|v_2|^2}{a_2^2} + \dots + \frac{|v_N|^2}{a_N^2} = 1, \quad (3.76)$$

N главных осей которого определяются как $2a_1, 2a_2, \dots, 2a_N$, где $a_1 = \sqrt{(\xi - \xi_{\min})/\lambda_1}$, $a_2 = \sqrt{(\xi - \xi_{\min})/\lambda_2}$, ..., $a_N = \sqrt{(\xi - \xi_{\min})/\lambda_N}$. Главными называются ортогональные

друг к другу и пересекающиеся в центре гиперэллипса оси, нормальные к касательным линиям кривой гиперэллипса.

Таким образом, проекция сечения MSE-функции гиперплоскостью, параллельной координатной гиперплоскости, на эту гиперплоскость согласно уравнению (3.76) представляет собой гиперэллипс. Это сечение сохраняет свою форму независимо от выбора системы координат в уравнениях (3.58), (3.72) и (3.73). Поэтому сечения MSE-функции для разных значений ξ представляют собой одни и те же концентрические гиперэллипсы в пространствах ВК \mathbf{h}_N , $\Delta\mathbf{h}_N$ и \mathbf{v}_N (см. рис. 3.6 для случая $N=2$).

На рис. 3.6 показаны сечения MSE-функции, представленной на рис. 3.3. Цифры на поле рис. 3.6 показывают значения MSE-функции ξ на линиях постоянного уровня. В исходной системе координат центры концентрических гиперэллипсов находятся в точке $\mathbf{h}_{N,o}$. В смещенной, а также в повернутой и смещенной системах координат центры гиперэллипсов расположены в начале координат этих систем.

Из рис. 3.4 и 3.5 видно, что направляющие векторы смещенной системы координат совпадают по направлению с направляющими векторами несмещенной системы координат, а направляющие векторы повернутой и смещенной системы координат совпадают по направлению с собственными векторами корреляционной матрицы. Можно также показать, что главные оси гиперэллипсов $\xi=\text{const}$ совпадают с направляющими векторами смещенной и повернутой систем координат.

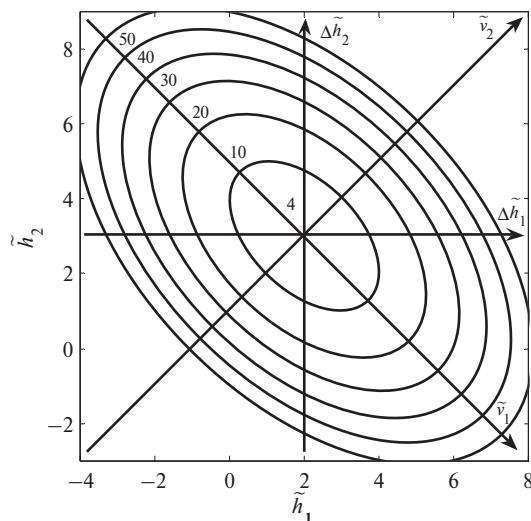


Рис. 3.6. Проекции сечений поверхности среднеквадратической ошибки

Действительно, в повернутой и смещенной системе координат компоненты v_n вектора \mathbf{v}_N согласно уравнению (3.74) представляют собой проекции вектора $\Delta\mathbf{h}_N$ на ортогональные смещенные направляющие векторы $\mathbf{q}_{N,n}$, т. е. $v_n = \mathbf{q}_{N,n}^H \Delta\mathbf{h}_N$ (см. рис. 3.5 для случая $N=2$).

Вектор градиента MSE-функции (3.73) в повернутой и смещенной системе координат определяется как

$$\begin{aligned}\nabla_{\mathbf{v}_N^*} \xi &= \frac{\partial \xi}{\partial \mathbf{v}_N^*} = \left[\frac{\partial \xi}{\partial v_1^*}, \frac{\partial \xi}{\partial v_2^*}, \dots, \frac{\partial \xi}{\partial v_N^*} \right]^T = \frac{\partial [\xi_{\min} + \mathbf{v}_N^H \mathbf{\Lambda}_N \mathbf{v}_N]}{\partial \mathbf{v}_N^*} = \\ &= \mathbf{\Lambda}_N \mathbf{v}_N = [\lambda_1 v_1, \lambda_2 v_2, \dots, \lambda_N v_N]^T.\end{aligned}\quad (3.77)$$

В случае действительных ВК этот градиент определяется как

$$\nabla_{\tilde{\mathbf{v}}_N} \xi = 2\mathbf{\Lambda}_N \tilde{\mathbf{v}}_N. \quad (3.78)$$

Градиент является вектором, нормальным к гиперплоскости, касающейся линии уровня целевой функции в точке его определения. Этот вектор направлен в сторону возрастания целевой функции (рис. 3.7). Поэтому на графиках сечений среднеквадратической ошибки вектор градиента $\nabla_{\mathbf{v}_N^*} \xi \{ \mathbf{v}_N(k) \}$, $\nabla_{\Delta\mathbf{h}_N^*} \xi \{ \Delta\mathbf{h}_N(k) \}$ или $\nabla_{\mathbf{h}_N^*} \xi \{ \mathbf{h}_N(k) \}$ (его проекция на гиперплоскость, образованную системой координат) отображается в виде вектора, нормального к касательной линии кривой $\xi = \text{const}$ в точке его определения (см. рис. 3.7).

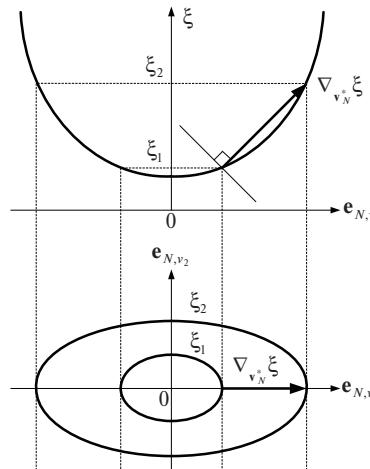


Рис. 3.7. Визуализация градиента в пространстве и на плоскости

С другой стороны, в повернутой и смещенной системе координат любая линия, проходящая через начало системы координат, имеет вид $\alpha \mathbf{v}_N$, где α — некоторая скалярная величина, а \mathbf{v}_N — вектор, совпадающий с этой линией. Чтобы на графике сечений среднеквадратической ошибки линия, проходящая через начало координат, и градиент (3.77) $\nabla_{\mathbf{v}_N^*} \xi = \mathbf{\Lambda}_N \mathbf{v}_N$ совпадали, необходимо, чтобы

выполнялось условие $\alpha \mathbf{v}_N = \mathbf{\Lambda}_N \mathbf{v}_N$. Согласно уравнению (3.15) это условие выполняется только в случае, когда α — собственное число матрицы $\mathbf{\Lambda}_N$, а \mathbf{v}_N — собственный вектор матрицы $\mathbf{\Lambda}_N$, соответствующий собственному числу α .

Так как матрица $\mathbf{\Lambda}_N$ диагональная, то собственные числа этой матрицы равны ее диагональным элементам. Действительно, согласно уравнению (3.17) характеристическое уравнение матрицы $\mathbf{\Lambda}_N$ определяется как

$$\det[\mathbf{\Lambda}_N - \alpha \mathbf{I}_N] = \det \begin{bmatrix} \lambda_1 - \alpha & 0 & \cdots & 0 \\ 0 & \lambda_2 - \alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_N - \alpha \end{bmatrix} = \prod_{n=1}^N (\lambda_n - \alpha) = 0. \quad (3.79)$$

Корни этого уравнения можно определить следующим образом. При $N = 1$ характеристическое уравнение имеется один корень $\alpha_1 = \lambda_1$. При $N = 2$ характеристическое уравнение определяется как

$$(\lambda_1 - \alpha)(\lambda_2 - \alpha) = \alpha^2 - \alpha(\lambda_1 + \lambda_2) + \lambda_1 \lambda_2 = 0,$$

а его два корня определяются как

$$\alpha_{1,2} = \frac{\lambda_1 + \lambda_2}{2} \pm \sqrt{\left(\frac{\lambda_1 + \lambda_2}{2}\right)^2 - \lambda_1 \lambda_2} = \frac{\lambda_1 + \lambda_2}{2} \pm \frac{\lambda_1 - \lambda_2}{2},$$

откуда $\alpha_1 = \lambda_1$ и $\alpha_2 = \lambda_2$. Получив и решив аналогичные уравнения при $N = 3$, $N = 4$ и т. д., можно убедиться, что $\alpha_n = \lambda_n$ для всех $n = 1, 2, \dots, N$.

Согласно уравнению (3.77) вектор градиента $\nabla_{\mathbf{v}_N} \xi = \mathbf{\Lambda}_N \mathbf{v}_N = [\lambda_1 v_1, \lambda_2 v_2, \dots, \lambda_N v_N]^T$ в общем случае имеет N различных компонент, т. е. $\alpha \mathbf{v}_N \neq \mathbf{\Lambda}_N \mathbf{v}_N$. Соотношение $\alpha \mathbf{v}_N = \mathbf{\Lambda}_N \mathbf{v}_N$ выполняется только когда вектор \mathbf{v}_N содержит одну ненулевую компоненту, т. е. вектор $\mathbf{v}_N = [0, 0, \dots, 0, v_n, 0, \dots, 0, 0]^T$ совпадает с направляющим вектором $\mathbf{e}_{N, v_n} = [0, 0, \dots, 0, 1, 0, \dots, 0, 0]^T$. Тогда

$$\alpha \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_{n-1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \lambda_n & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \lambda_{n+1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \lambda_{N-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \lambda_N \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ v_n \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \lambda_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ v_n \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (3.80)$$

при $\alpha = \lambda_n$.

Таким образом, из этого рассуждения следует, что главная ось эллипса (линия, проходящая через начало повернутой и смещенной системы коор-

динат, нормальная к касательной линии кривой $\xi = \text{const}$) и вектор градиента $\nabla_{\mathbf{v}_N^*} \xi = \mathbf{\Lambda}_N \mathbf{v}_N$ (также проходящий через линию, нормальную к касательной линии кривой $\xi = \text{const}$) совпадают только в случае, когда этот градиент совпадает с направляющим вектором $\mathbf{e}_{N, \mathbf{v}_n} = [0, 0, \dots, 0, 1, 0, \dots, 0, 0]^T$ повернутой и смещенной системы координат, который, в свою очередь, совпадает по направлению с собственным вектором $\mathbf{q}_{N,n} = [q_{1,n}, q_{2,n}, \dots, q_{N,n}]^T$ корреляционной матрицы \mathbf{R}_N .

Согласно уравнению (3.77) каждая компонента градиента $\nabla_{\mathbf{v}_N^*} \xi = \mathbf{\Lambda}_N \mathbf{v}_N$ зависит только от одной компоненты вектора \mathbf{v}_N , т. е. $\frac{\partial \xi}{\partial v_n^*} = \lambda_n v_n$, в то время как каждая компонента градиента (3.60) $\nabla_{\mathbf{h}_N^*} \xi = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N$ зависит от всех ВК как

$$\frac{\partial \xi}{\partial h_n^*} = -r_n + \sum_{m=1}^N R_{nm} h_m, \quad (3.81)$$

и аналогично, см. уравнение (3.72), каждая компонента градиента

$$\nabla_{\Delta \mathbf{h}_N^*} \xi = \frac{\partial \xi}{\partial \Delta \mathbf{h}_N^*} = \left[\frac{\partial \xi}{\partial \Delta h_1^*}, \frac{\partial \xi}{\partial \Delta h_2^*}, \dots, \frac{\partial \xi}{\partial \Delta h_N^*} \right]^T = \frac{\partial [\xi_{\min} + \Delta \mathbf{h}_N^* \mathbf{R}_N \Delta \mathbf{h}_N]}{\partial \Delta \mathbf{h}_N^*} = \mathbf{R}_N \Delta \mathbf{h}_N \quad (3.82)$$

зависит от всех ВК в смещенной системе координат как

$$\frac{\partial \xi}{\partial \Delta h_n^*} = \sum_{m=1}^N R_{nm} \Delta h_m, \quad (3.83)$$

что отличает градиенты (3.60), (3.77) и (3.82) друг от друга.

Продифференцировав уравнение (3.73) дважды и используя уравнение (3.77), также получим, что

$$\begin{aligned} \nabla_{\mathbf{v}_N}^2 \xi &= \nabla_{\mathbf{v}_N} \left\{ \nabla_{\mathbf{v}_N^*} \xi \right\} = \frac{\partial^2 \xi}{\partial \mathbf{v}_N \partial \mathbf{v}_N^*} = \left[\frac{\partial^2 \xi}{\partial v_1 \partial v_1^*}, \frac{\partial^2 \xi}{\partial v_2 \partial v_2^*}, \dots, \frac{\partial^2 \xi}{\partial v_N \partial v_N^*} \right]^T = \\ &= \frac{\partial [\mathbf{\Lambda}_N \mathbf{v}_N]}{\partial \mathbf{v}_N} = \frac{\partial [\lambda_1 v_1, \lambda_2 v_2, \dots, \lambda_N v_N]^T}{\partial \mathbf{v}_N} = [\lambda_1, \lambda_2, \dots, \lambda_N]^T, \end{aligned} \quad (3.84)$$

т. е. компоненты вектора вторых производных MSE-функции (3.73) равны собственным числам корреляционной матрицы \mathbf{R}_N . В случае действительных ВК $\nabla_{\mathbf{v}_N}^2 \xi = 2[\lambda_1, \lambda_2, \dots, \lambda_N]^T$.

Графическая интерпретация градиентов MSE-функции в различных системах координат показана на рис. 3.8 и рис. 3.9, из которых следует, что если нормаль (градиент) к касательной линии кривой $\xi = \text{const}$ параллельна направляющему вектору системы координат, то вектор градиента MSE-функции параллелен указанному направляющему вектору, т. е. содержит только одну не-нулевую компоненту.

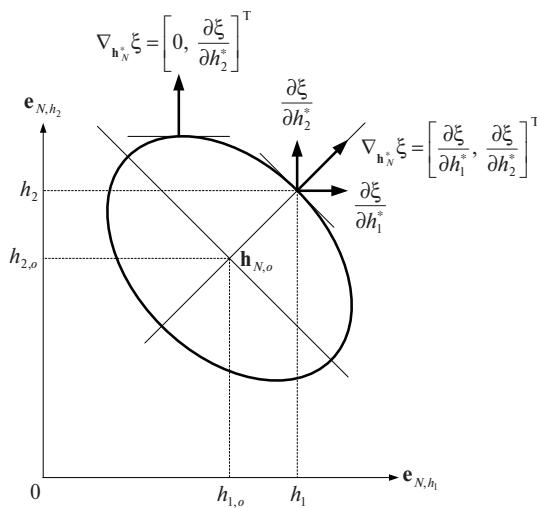


Рис. 3.8. Варианты градиента в исходной системе координат

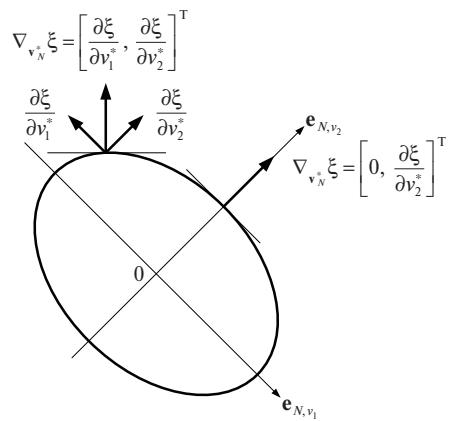


Рис. 3.9. Варианты градиента в смещенной и повернутой системе координат

Таким образом, согласно уравнению (3.77) компоненты градиента $\nabla_{\mathbf{v}_N^*} \xi$ MSE-функции располагаются вдоль главных осей концентрических гиперэллипсов $\xi = \text{const}$, или, что то же самое, вдоль направляющих векторов смещенной и повернутой системы координат, или вдоль собственных векторов корреляционной матрицы.

Из уравнения (3.77) также следует, что чем больше собственное число λ_n , тем больше градиент (т. е. изменение MSE-функции) вдоль оси v_n , или, другими словами, тем уже расстояния между линиями концентрических гиперэллипсов в направлении $\mathbf{q}_{N,n}$. То есть, чем больше λ_n , тем уже (более сжатые) гиперэллипсы вдоль оси v_n .

Чем меньше λ_n , тем меньше градиент (изменение MSE-функции) вдоль оси v_n , или, другими словами, тем шире расстояния между линиями концентрических гиперэллипсов в направлении $\mathbf{q}_{N,n}$. То есть, чем меньше λ_n , тем шире (более растянутые) гиперэллипсы вдоль оси v_n .

При $\lambda_n = \lambda_m$ для всех n и m гиперэллипсы преобразуются в гиперокружности, так как

$$\frac{|v_1|^2}{a^2} + \frac{|v_2|^2}{a^2} + \dots + \frac{|v_N|^2}{a^2} = 1 \quad (3.85)$$

или

$$|v_1|^2 + |v_2|^2 + \dots + |v_N|^2 = a^2. \quad (3.86)$$

Таким образом, разброс собственных чисел корреляционной матрицы \mathbf{R}_N , характеризуемый параметром $\chi = \lambda_{\max}/\lambda_{\min}$, определяет форму гиперэллипсов $\xi = \text{const}$.

Сказанное подтверждается ниже. На рис. 3.10 приведена схема адаптивного устройства линейного предсказания сигнала авторегрессионного процесса $x(k) + a_1x(k-1) + a_2x(k-2) = e_o(k)$, где $e_o(k)$ — белый шум с нулевым средним значением $E\{e_o(k)\} = 0$ и дисперсией σ_o^2 [20]. Собственные числа корреляционной матрицы \mathbf{R}_N , где $N = 2$, определяются соотношением ВК a_1 и a_2 .

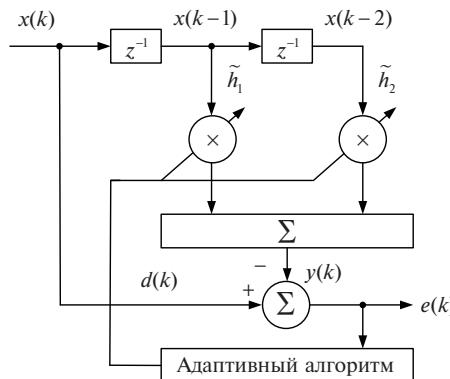


Рис. 3.10. Линейное предсказание

На рис. 3.11 приведены сечения поверхности MSE-функции фильтра (см. рис. 3.10) при $\sigma_o^2 = 1$, $a_1 = 0$, $a_2 = 0.5$, $\lambda_1 = 1.33$, $\lambda_2 = 1.33$, $\chi = 1$, $\tilde{\mathbf{h}}_{N,o} = [0, -0.5]^T$, $\sigma_x^2 = 1.33$, $\mathbf{r}_N = [0, -0.67]^T$, $\mathbf{R}_N = \begin{bmatrix} 1.33 & 0 \\ 0 & 1.33 \end{bmatrix}$ и $\xi_{\min} = 1$.

На рис. 3.12 приведены сечения поверхности этой же MSE-функции при $\sigma_o^2 = 1$, $a_1 = -1$, $a_2 = 0.5$, $\lambda_1 = 0.8$, $\lambda_2 = 4$, $\chi = 5$, $\tilde{\mathbf{h}}_{N,o} = [1, -0.5]^T$, $\sigma_x^2 = 2.4$, $\mathbf{r}_N = [1.6, 0.4]^T$, $\mathbf{R}_N = \begin{bmatrix} 2.4 & 1.6 \\ 1.6 & 2.4 \end{bmatrix}$ и $\xi_{\min} = 1$.

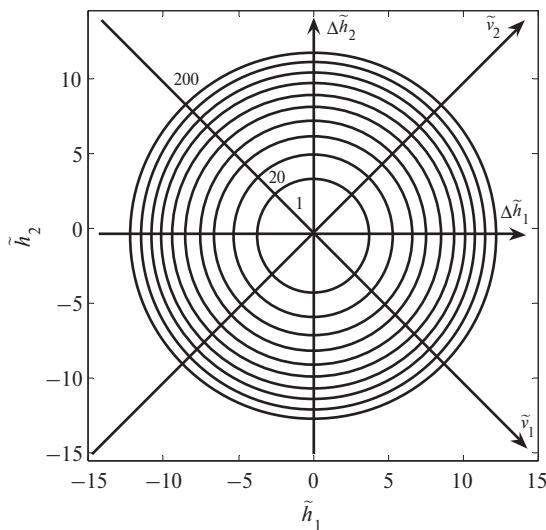


Рис. 3.11. Проекции сечений поверхности среднеквадратической ошибки: $\chi=1$

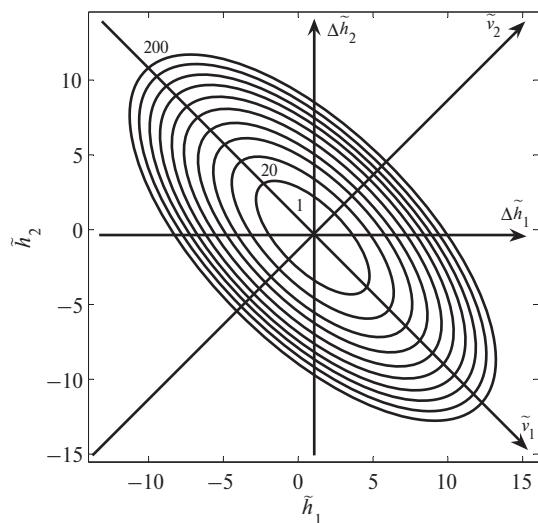


Рис. 3.12. Проекции сечений поверхности среднеквадратической ошибки: $\chi=5$

И на рис. 3.13 приведены сечения MSE-поверхности при $\sigma_o^2=1$, $a_1=-1.23$, $a_2=0.5$, $\lambda_1=0.578$, $\lambda_2=5.78$, $\chi=10$, $\tilde{\mathbf{h}}_{N,o}=[1.23, -0.5]^T$, $\sigma_x^2=4.3$, $\mathbf{r}_N=[3.30, 2.03]^T$, $\mathbf{R}_N=\begin{bmatrix} 4.03 & 3.30 \\ 3.30 & 4.03 \end{bmatrix}$ и $\xi_{\min}=1$.

Из рис. 3.11—3.13 видно, что с увеличением разброса собственных чисел χ поверхность MSE-функции становится более вытянутой. Она вытягивается вдоль

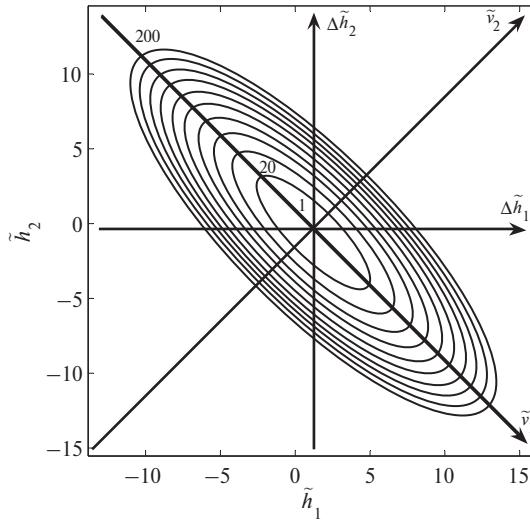


Рис. 3.13. Проекции сечений поверхности среднеквадратической ошибки: $\chi=10$

оси повернутой и смещенной системы координат, определяемой меньшим значением собственного числа λ , и сжимается вдоль оси, определяемой большим значением собственного числа λ .

3.6. Пример расчета собственных чисел, собственных векторов корреляционной матрицы и поверхности среднеквадратической ошибки

Рассмотрим пример анализа MSE-функции фильтра Винера с двумя действительными ВК [3]. Изображения на рис. 3.3 и рис. 3.6 получены на основе этого примера.

Пусть для $N = 2$ даны $\mathbf{R}_N = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $\mathbf{r}_N = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$, $\sigma_d^2 = 42$. Подставив эти значения в (3.58), можно получить уравнение для расчета MSE-функции ξ при изменении действительных ВК

$$\xi = \sigma_d^2 - 2\tilde{\mathbf{h}}_N^T \mathbf{r}_N + \tilde{\mathbf{h}}_N^T \mathbf{R}_N \tilde{\mathbf{h}}_N = 42 - 2[\tilde{h}_1, \tilde{h}_2] \begin{bmatrix} 7 \\ 8 \end{bmatrix} + [\tilde{h}_1, \tilde{h}_2] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \end{bmatrix}.$$

При построении трехмерного графика MSE-функции или графиков ее линий постоянного уровня ВК необходимо изменять в некоторых пределах относительно координат оптимального винеровского решения (3.63), которое в данном случае определяется как

$$\tilde{\mathbf{h}}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 0.6667 & -0.3333 \\ -0.3333 & 0.6667 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Далее с помощью уравнения (3.64) можно вычислить значение ξ_{\min} как

$$\xi_{\min} = \sigma_d^2 - \mathbf{r}_N^T \tilde{\mathbf{h}}_{N,o} = 42 - [7, 8] \begin{bmatrix} 2 \\ 3 \end{bmatrix} = 4.$$

Можно также убедиться, что согласно уравнению (3.72) среднеквадратическая ошибка определяется и как

$$\xi = \xi_{\min} + \Delta \tilde{\mathbf{h}}_N^T \mathbf{R}_N \Delta \tilde{\mathbf{h}}_N = 4 + [\tilde{h}_1 - 2, \tilde{h}_2 - 3] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \tilde{h}_1 - 2 \\ \tilde{h}_2 - 3 \end{bmatrix}.$$

Используя уравнение (3.17), можно составить характеристическое уравнение

$$\det \{ \mathbf{R}_N - \lambda \mathbf{I}_N \} = \det \begin{bmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{bmatrix} = \lambda^2 - 4\lambda + 3 = 0,$$

решение которого дает два собственных числа $\lambda_{1,2} = 2 \pm \sqrt{4 - 3} = 2 \pm 1$, т. е. $\lambda_1 = 1$ и $\lambda_2 = 3$.

Затем, используя уравнение (3.16), можно составить уравнения

$$(\mathbf{R}_N - \lambda_1 \mathbf{I}_N) \mathbf{q}_N = \begin{bmatrix} 2 - \lambda_1 & 1 \\ 1 & 2 - \lambda_1 \end{bmatrix} \begin{bmatrix} q_{1,1} \\ q_{2,1} \end{bmatrix} = \begin{bmatrix} 2 - 1 & 1 \\ 1 & 2 - 1 \end{bmatrix} \begin{bmatrix} q_{1,1} \\ q_{2,1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} q_{1,1} \\ q_{2,1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

и

$$(\mathbf{R}_N - \lambda_2 \mathbf{I}_N) \mathbf{q}_N = \begin{bmatrix} 2 - \lambda_2 & 1 \\ 1 & 2 - \lambda_2 \end{bmatrix} \begin{bmatrix} q_{1,2} \\ q_{2,2} \end{bmatrix} = \begin{bmatrix} 2 - 3 & 1 \\ 1 & 2 - 3 \end{bmatrix} \begin{bmatrix} q_{1,2} \\ q_{2,2} \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} q_{1,2} \\ q_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Из первого уравнения следует, что $q_{1,1} + q_{2,1} = 0$, а значит, $q_{1,1} = -q_{2,1} = c_1$. Аналогично из второго уравнения следует, что $-q_{1,2} + q_{2,2} = 0$, а значит, $q_{1,2} = q_{2,2} = c_2$. Таким образом, решение этих уравнений может быть найдено лишь в виде произвольных констант, так как определитель участвующих в вычислениях матриц равен нулю.

При $N=2$ справедливость рассмотренных двух уравнений и следующих из них решений можно доказать и для произвольных корреляционных матриц

$$\mathbf{R}_N = \begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix}.$$

Действительно, из (3.18) следует, что характеристическое уравнение определяется полиномом $\lambda^2 - 2R(0)\lambda + R^2(0) - R^2(1) = 0$. Решениями этого уравнения являются два разных собственных числа $\lambda_{1,2} = R(0) \pm \sqrt{R^2(0) - R^2(0) + R^2(1)} = R(0) \pm R(1)$, которые для определенности можно обозначить как $\lambda_1 = R(0) - R(1)$ и $\lambda_2 = R(0) + R(1)$, т. е. $\lambda_1 < \lambda_2$, так как $R(0) > R(1)$.

Затем, используя уравнение (3.16)

$$(\mathbf{R}_N - \lambda_n \mathbf{I}_N) \mathbf{q}_N = \begin{bmatrix} R(0) - \lambda_n & R(1) \\ R(1) & R(0) - \lambda_n \end{bmatrix} \begin{bmatrix} q_{1,n} \\ q_{2,n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

где n принимает значения, равные 1 и 2, можно составить систему однородных линейных уравнений

$$\begin{cases} [R(0) - \lambda_n]q_{1,n} + R(1)q_{2,n} = 0 \\ R(1)q_{1,n} + [R(0) - \lambda_n]q_{2,n} = 0, \end{cases}$$

решением которой является

$$q_{2,n} = -\frac{R(0) - \lambda_n}{R(1)} q_{1,n}.$$

Используя значения $\lambda_1 = R(0) - R(1)$ и $\lambda_2 = R(0) + R(1)$, получим, что

$$\mathbf{Q}_N = [\mathbf{q}_{N,1}, \mathbf{q}_{N,2}] = \begin{bmatrix} q_{1,1} & q_{1,2} \\ q_{2,1} & q_{2,2} \end{bmatrix} = \begin{bmatrix} q_{1,1} & q_{1,2} \\ -q_{1,1} & q_{1,2} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ -c_1 & c_2 \end{bmatrix}.$$

То есть, решение однородного уравнения (3.16) может быть найдено лишь с точностью до констант c_1 и c_2 . Эти константы можно, например, выбирать так, чтобы матрица \mathbf{Q}_N была ортонормированной, т. е.

$$\mathbf{Q}_N \mathbf{Q}_N^T = \begin{bmatrix} c_1 & c_2 \\ -c_1 & c_2 \end{bmatrix} \begin{bmatrix} c_1 & -c_1 \\ c_2 & c_2 \end{bmatrix} = \begin{bmatrix} c_1^2 + c_2^2 & -c_1^2 + c_2^2 \\ -c_1^2 + c_2^2 & c_1^2 + c_2^2 \end{bmatrix} = \mathbf{I}_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

откуда следует, что

$$\begin{cases} c_1^2 + c_2^2 = 1 \\ -c_1^2 + c_2^2 = 0. \end{cases}$$

Решая эту систему уравнений, получим, что $c_1^2 = c_2^2$ и $c_2^2 + c_1^2 = 2c_2^2 = 1$, откуда $c_2 = \pm 1/\sqrt{2}$ и $c_1 = \pm 1/\sqrt{2}$. Используя значения c_1 и c_2 , матрицу \mathbf{Q}_N можно определить четырьмя различными способами как

$$\mathbf{Q}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{Q}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}, \mathbf{Q}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \text{ или } \mathbf{Q}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Рассматривая различные корреляционные матрицы \mathbf{R}_N при $N=2$, легко убедиться, что во всех случаях матрицы \mathbf{Q}_N будут иметь такой же вид, как это показано выше.

Также можно убедиться, что $\mathbf{Q}_N \mathbf{Q}_N^T = \mathbf{I}_N$ и что согласно уравнению (3.47)

$$\mathbf{R}_N = \mathbf{Q}_N \Lambda_N \mathbf{Q}_N^T = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

для всех разновидностей обозначенных выше матриц \mathbf{Q}_N . Соотношение (3.74) определяет ВК в повернутой и смещенной системе координат также для всех четырех значений \mathbf{Q}_N .

Например, для матрицы

$$\mathbf{Q}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

вектор $\tilde{\mathbf{v}}_N$ определяется как

$$\tilde{\mathbf{v}}_N = \mathbf{Q}_N^T \Delta \tilde{\mathbf{h}}_N = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \tilde{h}_1 - 2 \\ \tilde{h}_2 - 3 \end{bmatrix}.$$

Векторы \mathbf{v}_N для разных матриц \mathbf{Q}_N отличаются положительным направлением осей смещенной и повернутой системы координат. Графики на рис. 3.6 и рис. 3.11—3.13 соответствуют рассмотренному выше случаю.

Таким образом, используя приведенные вычисления, можно строить трехмерные графики MSE-функции (рис. 3.3) и графики линий постоянного уровня (рис. 3.6).

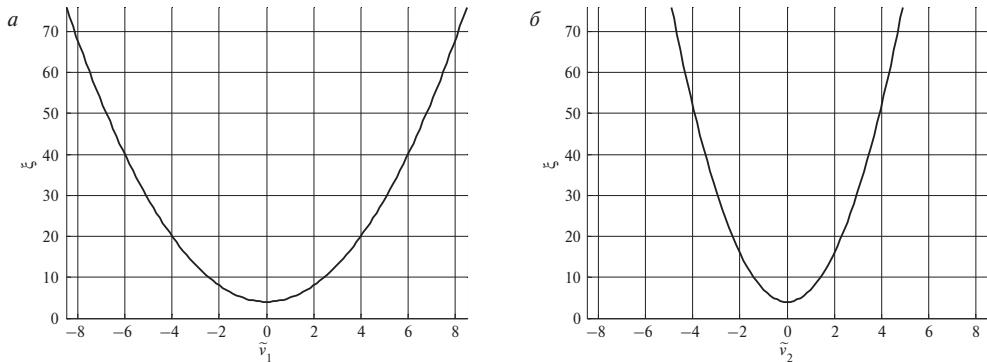


Рис. 3.14. Сечения поверхности среднеквадратической ошибки (см. рис. 3.3)

При $N > 2$ такие вычисления удобнее делать, используя доступные языки программирования. Строить многомерную поверхность и графики линий постоянного уровня при этом не представляется возможным. Однако, задавая векторы \mathbf{v}_N как $\mathbf{v}_N = [0, 0, \dots, v_n, \dots, 0, 0]^T$, можно строить и анализировать сечения MSE-функции вдоль направлений главных осей гиперэллипсов. Такие сечения для рассмотренного примера (см. рис. 3.6) приведены на рис. 3.14.

3.7. Линейно-ограниченная винеровская фильтрация

В разделе 3.4 было получено решение задачи оптимальной винеровской фильтрации, основанное на безусловной минимизации целевой MSE-функции. Однако существуют задачи, в которых требуется накладывать ограничения на ВК адаптивного фильтра в процессе поиска минимума этой функции. Наиболее распространенными (в силу простоты реализации) являются линейные огра-

ничения. Таким способом решаются, например, задачи идентификации при наложении ограничений на значения АЧХ или фазо-частотной характеристики (ФЧХ) адаптивного фильтра на выбранных частотах; задачи, в которых необходимо обеспечить фиксированное значение нормы ВК или требуемое значение уровня полезного сигнала на выходе ААР, и ряд других задач.

Формулировка и решение задачи линейно-ограниченной (Linearly Constrained, LC) минимизации MSE-функции во многом аналогичны такой задаче без ограничений (3.56)–(3.63).

Переход от условной оптимизации к безусловной осуществляется с помощью метода множителей Лагранжа. Идея LC адаптивной фильтрации состоит в том, что поиск минимального значения функций (3.58) осуществляется при наличии линейных ограничений вида

$$\mathbf{C}_{NJ}^H \mathbf{h}_N = \mathbf{f}_J. \quad (3.87)$$

Здесь \mathbf{C}_{NJ} — матрица J ограничений, \mathbf{f}_J — вектор J значений ограничиваемого параметра (например уровня выходного сигнала адаптивного фильтра).

Суть линейно-ограниченной оптимизации основана на том, что минимизируемая функция (3.58) с помощью множителей Лагранжа модифицируется как

$$\begin{aligned} \xi_c = E \left\{ |e(k)|^2 \right\} &= \sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N + \\ &+ \lambda_J^H \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N \right] + \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N \right]^H \lambda_J, \end{aligned} \quad (3.88)$$

где $\lambda_J = [\lambda_1, \lambda_2, \dots, \lambda_J]^T$ — вектор множителей Лагранжа (не путать с собственными числами).

Значение функции (3.88) минимально при

$$\nabla_{\mathbf{h}_N^*} \xi_c = \frac{\partial \xi_c}{\partial \mathbf{h}_N^*} = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N - \mathbf{C}_{NJ} \lambda_J = \mathbf{0}_N,$$

откуда

$$\mathbf{h}_{N,o_c} = \mathbf{R}_N^{-1} [\mathbf{r}_N + \mathbf{C}_{NJ} \lambda_J]. \quad (3.89)$$

В случае действительных ВК

$$\nabla_{\tilde{\mathbf{h}}_N} \xi_c = -2\mathbf{r}_N + 2\mathbf{R}_N \mathbf{h}_N - 2\mathbf{C}_{NJ} \lambda_J. \quad (3.90)$$

Вектор λ_J можно найти, подставив (3.89) в уравнение (3.87), т. е.

$$\mathbf{C}_{NJ}^H \mathbf{h}_{N,o_c} = \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{r}_N + \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{C}_{NJ} \lambda_J = \mathbf{f}_J. \quad (3.91)$$

Из уравнения (3.91) следует, что

$$\lambda_J = \left[\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{r}_N \right]. \quad (3.92)$$

Подставив (3.92) в уравнение (3.89), получим решение LC оптимальной винеровской фильтрации

$$\mathbf{h}_{N,o_c} = \mathbf{R}_N^{-1} \mathbf{r}_N + \mathbf{R}_N^{-1} \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{r}_N \right]. \quad (3.93)$$

Геометрическая интерпретация метода множителей Лагранжа [25] приведена на рис. 3.15. Из рисунка видно, что плоскость, проходящая через линию ограничений $g(\mathbf{h}_N) = 0$, отсекает часть гиперпараболоида, а на пересечении гиперпараболоида и этой плоскости образуется гиперпарабола (показана штриховой линией).

При отсутствии линейных ограничений допустимые значения вектора \mathbf{h}_N ничем не ограничиваются, а минимум целевой функции ξ_{\min} находится в точке с координатами \mathbf{h}_{N,o_c} . В случае линейных ограничений область допустимых значений вектора \mathbf{h}_N находится на линии ограничений $g(\mathbf{h}_N) = 0$, а минимум LC целевой функции $\xi_{\min,c}$ — в точке минимума гиперпараболы с координатами \mathbf{h}_{N,o_c} .

Минимум функции $\xi(\mathbf{h}_N)$ находится в точке, где ее градиент равен нуль-вектору, т.е. $\nabla \xi_{\mathbf{h}_N^*} = \mathbf{0}_N$. Минимум гиперпараболы находится в точке \mathbf{h}_{N,o_c} , где компонента градиента $\nabla \xi_{\mathbf{h}_N^*}$, параллельная линии ограничений $g(\mathbf{h}_N) = 0$, также равна нуль-вектору, т.е. $\nabla \xi_{\mathbf{h}_{N,c}^*} = \mathbf{0}_N$. Чтобы найти $\nabla \xi_{\mathbf{h}_{N,c}^*}$, рассмотрим функцию $g(\mathbf{h}_N + \delta \mathbf{h}_N)$, где вектор $\delta \mathbf{h}_N$ такой, что точка $\mathbf{h}_N + \delta \mathbf{h}_N$ находится на линии ограничения, т.е. $g(\mathbf{h}_N + \delta \mathbf{h}_N) = g(\mathbf{h}_N)$, а значит, $\delta \mathbf{h}_N$ лежит на линии ограничения. Используя разложение в ряд Тейлора, можем записать, что $g(\mathbf{h}_N + \delta \mathbf{h}_N) = g(\mathbf{h}_N) + \delta \mathbf{h}_N^H \nabla g_{\mathbf{h}_N^*} = g(\mathbf{h}_N)$. Это выражение справедливо, если $\delta \mathbf{h}_N^H \nabla g_{\mathbf{h}_N^*} = 0$, т.е. вектор $\nabla g_{\mathbf{h}_N^*}$ ортогонален вектору $\delta \mathbf{h}_N$ и ортогонален вектору $\nabla \xi_{\mathbf{h}_{N,c}^*}$, поскольку векторы $\delta \mathbf{h}_N$ и $\nabla \xi_{\mathbf{h}_{N,c}^*}$ коллинеарные ($\delta \mathbf{h}_N$ лежит на линии ограничений, а $\nabla \xi_{\mathbf{h}_{N,c}^*}$ параллелен этой линии).

Таким образом, из этого рассуждения следует (см. рис. 3.15), что вектор $\nabla \xi_{\mathbf{h}_{N,c}^*}$ можно представить в виде суммы двух векторов как $\nabla \xi_{\mathbf{h}_{N,c}^*} = \nabla \xi_{\mathbf{h}_N^*} + \lambda \nabla g_{\mathbf{h}_N^*}$, где λ — масштабирующий множитель (множитель Лагранжа). В правом нижнем углу рис. 3.15 векторы градиентов показаны в увеличенном масштабе.

Из свойства линейности оператора градиента следует, что LC целевая функция также определяется как сумма двух слагаемых $\xi_c(\mathbf{h}_N) = \xi(\mathbf{h}_N) + \lambda g(\mathbf{h}_N)$. При наличии J ограничений LC целевая функция определяется как $\xi_c(\mathbf{h}_N) = \xi(\mathbf{h}_N) + \sum_{j=1}^J \lambda_j g_j(\mathbf{h}_N)$. Следовательно, в уравнении (3.88) функция $\xi(\mathbf{h}_N)$ определяется как $\sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$, а функция $\sum_{j=1}^J \lambda_j g_j(\mathbf{h}_N)$ — как $\lambda_J^H \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N \right] + \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N \right]^H \lambda_J$.

Структура матрицы \mathbf{C}_{NJ} и вектора \mathbf{f}_J зависит от вида задачи, решаемой с помощью LC адаптивного фильтра.

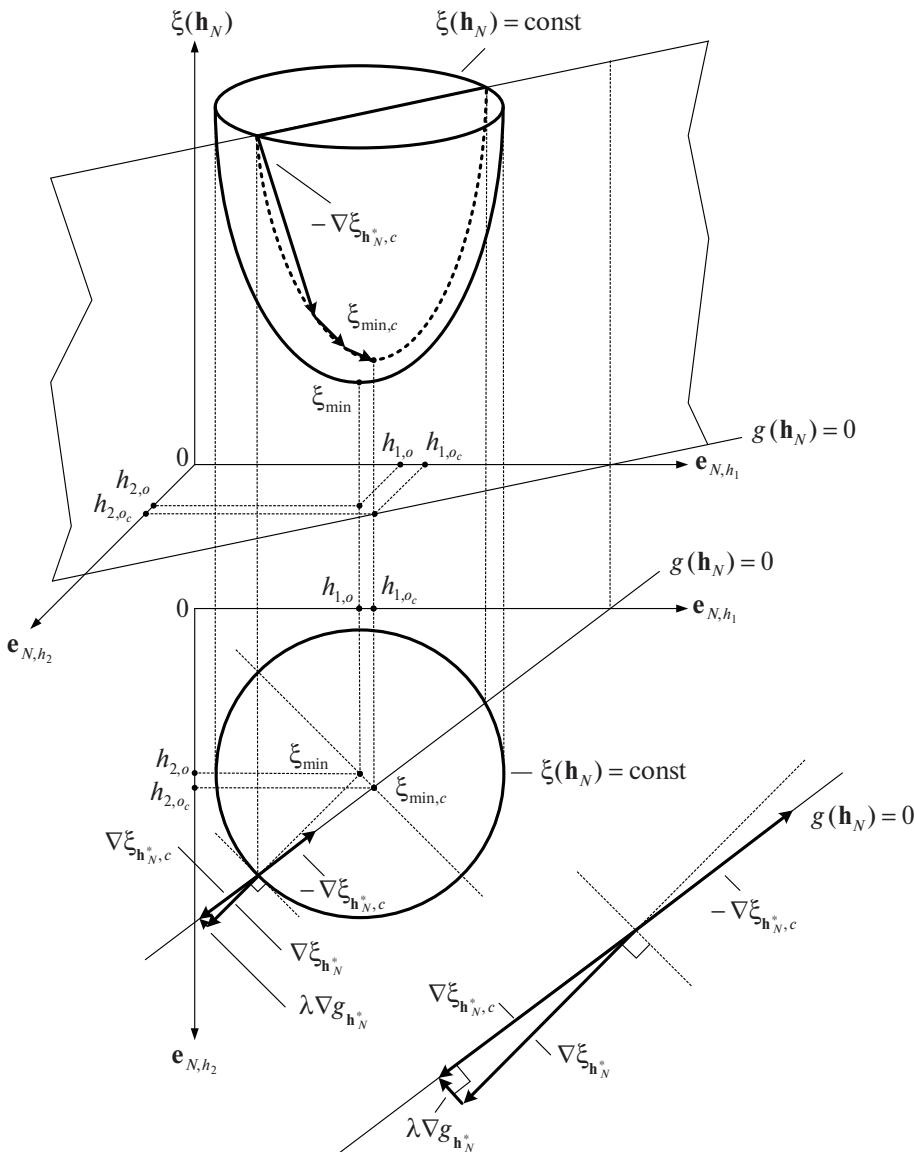


Рис. 3.15. Геометрическая интерпретация линейных ограничений

Например, пусть требуется, чтобы на частотах ω_j АЧХ адаптивного КИХ-фильтра (см. рис. 1.2) всегда имела фиксированные значения, определяемые элементами вектора \mathbf{f}_j . Тогда согласно уравнению (3.87) столбцы матрицы $\mathbf{C}_{Nj} = [\mathbf{c}_{N,1}, \mathbf{c}_{N,2}, \dots, \mathbf{c}_{N,J}]$ можно определить векторами

$$\mathbf{c}_{N,j} = [1, e^{j(\omega_j/F_S)}, e^{j(\omega_j/F_S)2}, \dots, e^{j(\omega_j/F_S)(N-2)}, e^{j(\omega_j/F_S)(N-1)}]^T \quad (3.94)$$

для всех $j = 1, \dots, J$. Здесь ω_j — значение круговой частоты гармонического сигнала, на которой задается j -е ограничение; F_s — частота дискретизации обрабатываемых сигналов.

В случае ААР вид матрицы \mathbf{C}_{Nj} определяется следующим образом [55]. На рис. 3.16 показана конфигурация элементов N -элементной эквидистантной узкополосной линейной антенной решетки (с одним ВК в каждом канале, см. рис. 1.26, где $M = N$). Расстояние между элементами равно d_0 . Оно, как правило, выбирается равным половине длины волны λ_0 несущего сигнала (на частоте f_0), т. е. $d_0 = 0.5\lambda_0 = 0.5v/f_0$, где v — скорость распространения электромагнитной волны в свободном пространстве, равная скорости света.

Плоская волна, принимаемая антенной решеткой, без учета ее временно-го запаздывания и амплитудного ослабления в силу эффекта распространения от источника излучения до антенных элементов описывается уравнением $s(t) = A_a(t)e^{j\omega_0 t}$, где $A_a(t) = A_{a,I}(t) + jA_{a,Q}(t)$ — непрерывный во времени модулирующий сигнал (комплексный в общем случае), $A_{a,I}(t)$ — действительная часть модулирующего сигнала, $A_{a,Q}(t)$ — мнимая часть модулирующего сигнала, $\omega_0 = 2\pi f_0$ и t — текущее время. Здесь и далее нижний индекс a указывает на то, что переменная меняется непрерывно во времени. Если принять любой из элементов решетки за опорный, например элемент с номером 1 (см. рис. 3.16), то сигнал на выходе этого элемента будет $s_1(t) = \operatorname{Re}\{s(t)\}$.

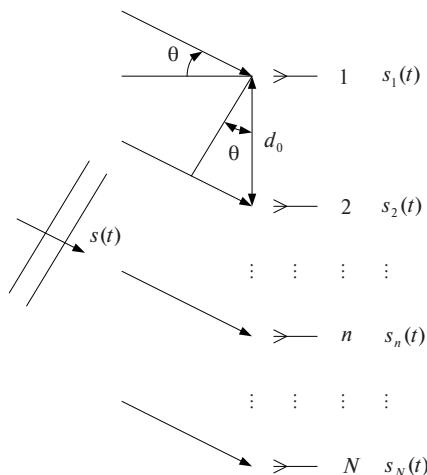


Рис. 3.16. Сигналы в линейной антенной решетке

Подразумевается, что каждая антенна решетки является всенаправленной. На практике антенны обладают некоторой направленностью, что учитывается дополнительным множителем, зависящим от угла θ в выражении для $s_1(t)$. Этот множитель является масштабирующим, и его наличие не влияет на принципы обработки сигналов.

Сигнал на выходе n -го элемента антенной решетки можно представить как

$$\begin{aligned}
 s_n(t) &= \operatorname{Re}\left\{A_a(t+\tau_n)e^{j\omega_0(t+\tau_n)}\right\} = \\
 &= \operatorname{Re}\left\{\left[A_{a,I}(t+\tau_n)+jA_{a,Q}(t+\tau_n)\right]\left(\cos[\omega_0(t+\tau_n)]+j\sin[\omega_0(t+\tau_n)]\right)\right\} = \\
 &= A_{a,I}(t+\tau_n)\cos[\omega_0(t+\tau_n)]-A_{a,Q}(t+\tau_n)\sin[\omega_0(t+\tau_n)] = \\
 &= |A_a(t+\tau_n)|\cos[\omega_0(t+\tau_n)+\theta_a],
 \end{aligned} \quad (3.95)$$

где τ_n — абсолютная задержка между сигналами на входах опорного и n -го антенного элементов. Здесь использованы следующие переменные, определяющие комплексную огибающую $A_a(t+\tau_n)$ (т.е. модулирующий сигнал):

$|A_a(t+\tau_n)|=\sqrt{\left[A_{a,I}(t+\tau_n)\right]^2+\left[A_{a,Q}(t+\tau_n)\right]^2}$ — амплитуда комплексной огибающей, $A_{a,I}(t+\tau_n)=|A_a(t+\tau_n)|\cos(\theta_a)$ — синфазная и $A_{a,Q}(t+\tau_n)=|A_a(t+\tau_n)|\sin(\theta_a)$ — квадратурная составляющие комплексной огибающей, $\theta_a=\tan^{-1}\left[\frac{A_{a,Q}(t+\tau_n)}{A_{a,I}(t+\tau_n)}\right]$ — фаза комплексной огибающей. В представленном выше уравнении использована формула $\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$.

В случае узкополосных сигналов (когда $f_0 \gg \Delta f$, где Δf — ширина полосы сигнала) $A_a(t+\tau_n) \approx A_a(t)$ и

$$s_n(t)=A_{a,I}(t)\cos[\omega_0(t+\tau_n)]-A_{a,Q}(t)\sin[\omega_0(t+\tau_n)]. \quad (3.96)$$

Это означает, что сигналы, принимаемые антеннами решетки, различаются только фазовым сдвигом, равным $\omega_0\tau_n$.

Чтобы выполнять ЦОС принимаемых сигналов необходимо выполнить перенос информации, содержащейся в сигнале $s_n(t)$, $n=1, 2, \dots, N$, что осуществляется путем преобразования частоты (рис. 3.17).

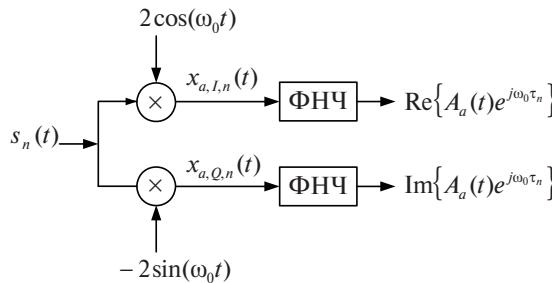


Рис. 3.17. Преобразование частоты сигнала в n -м канале антенной решетки

Для этого сигнал, описываемый выражением (3.96), умножается на сигналы $2\cos(\omega_0 t)$ и $-2\sin(\omega_0 t)$, т.е.

$$\begin{aligned}
 x_{a,I,n}(t) &= 2\cos(\omega_0 t)s_n(t)=A_{a,I}(t)\cos(\omega_0\tau_n)-A_{a,Q}(t)\sin(\omega_0\tau_n)+ \\
 &+ A_{a,I}(t)\cos(2\omega_0 t+\omega_0\tau_n)-A_{a,Q}(t)\sin(2\omega_0 t+\omega_0\tau_n)
 \end{aligned} \quad (3.97)$$

и

$$x_{a,Q,n}(t) = -2\sin(\omega_0 t)s_n(t) = A_{a,Q}(t)\cos(\omega_0\tau_n) + A_{a,I}(t)\sin(\omega_0\tau_n) - A_{a,Q}(t)\cos(2\omega_0 t + \omega_0\tau_n) - A_{a,I}(t)\sin(2\omega_0 t + \omega_0\tau_n). \quad (3.98)$$

Выражения (3.97) и (3.98) можно переписать как

$$x_{a,I,n}(t) = A_{a,I}(t)\cos(\omega_0\tau_n) - A_{a,Q}(t)\sin(\omega_0\tau_n) + z_{a,I,n}(t), \quad (3.99)$$

$$x_{a,Q,n}(t) = A_{a,Q}(t)\cos(\omega_0\tau_n) + A_{a,I}(t)\sin(\omega_0\tau_n) + z_{a,Q,n}(t). \quad (3.100)$$

Здесь $z_{a,I,n}(t)$ и $z_{a,Q,n}(t)$ — аддитивный шум, обусловленный конечным подавлением составляющих сигналов с частотой $2\omega_0$ фильтрами низкой частоты (ФНЧ), обычно следующими за гетеродином (преобразователем частоты).

Выражения (3.99) и (3.100) (без учета аддитивного шума) можно записать и как

$$x_{a,n}(t) = A_a(t)e^{j\omega_0\tau_n} = [A_{a,I}(t)\cos(\omega_0\tau_n) - A_{a,Q}(t)\sin(\omega_0\tau_n)] + j[A_{a,Q}(t)\cos(\omega_0\tau_n) + A_{a,I}(t)\sin(\omega_0\tau_n)]. \quad (3.101)$$

Из (3.101) следует, что правые части уравнений (3.99) и (3.100) можно представить как действительную и мнимую части комплексной переменной, т. е.

$$x_{a,n}(t) = x_{a,I,n}(t) + jx_{a,Q,n}(t) = A_a(t)e^{j\omega_0\tau_n} + z_{a,n}(t) = A_a(t)e^{j\omega_0\tau_n} + z_n(t). \quad (3.102)$$

При ЦОС отсчеты сигнала (3.102) производятся на дискретных интервалах времени k , кратных интервалу дискретизации $T_S = 1/F_S$. В этом случае значения (3.102) в моменты отсчетов представляются как

$$x_n(k) = x_{a,n}(kT)|_{t=kT} = x_{I,n}(k) + jx_{Q,n}(k) = A(k)e^{j\omega_0\tau_n} + z_{I,n}(k) + jz_{Q,n}(k) = A(k)e^{j\omega_0\tau_n} + z_n(k), \quad (3.103)$$

где $z_n(k)$ — шум, включающий в себя также эффекты квантования сигналов при аналого-цифровом преобразовании. Выражение (3.103) можно использовать при моделировании алгоритмов функционирования узкополосных ААР.

Если антенная решетка принимает J_s сигналов, то по принципу суперпозиции сигналы $x_n(k)$ определяются как

$$x_n(k) = \sum_{j=1}^{J_s} x_n^{(j)}(k) = \sum_{j=1}^{J_s} A^{(j)}(k)e^{j\omega_0\tau_n^{(j)}} + z_n(k), \quad (3.104)$$

где верхний индекс (j) указывает на параметры j -го сигнала.

Таким образом, преобразование частоты сохраняет амплитудно-фазовые соотношения в сигналах, принимаемых антеннами ААР. Из рис. 3.16 следует, что $\tau_n = d_0(n-1)\sin(\theta)/v$ и

$$\omega_0\tau_n = 2\pi f_0 d_0(n-1)\sin(\theta)/v = 2\pi d_0(n-1)\sin(\theta)/\lambda_0 = \psi_n. \quad (3.105)$$

При LC обработке сигналов в узкополосной ААР, как правило, используется одно линейное ограничение

$$\mathbf{c}_N^H \mathbf{h}_N = f, \quad (3.106)$$

где

$$\mathbf{c}_N(\theta) = [c_1(\theta), \dots, c_n(\theta), \dots, c_N(\theta)]^T = [e^{j\psi_1}, \dots, e^{j\psi_n}, \dots, e^{j\psi_N}]^T. \quad (3.107)$$

Выражение (3.106) представляет собой значение комплексной ДН ААР в направлении θ , а вектор (3.107) — это вектор искусственных комплексных сигналов единичной амплитуды на входах ААР.

Ограничение (3.106) означает, что при приеме сигналов плоской волны $s(t)$ с направления θ выходной сигнал ААР должен быть пропорционален значению f , которое в общем случае является комплексной величиной.

Если $d(k) = 0$ для всех k , т. е. требуемый сигнал отсутствует, то уравнение (3.93) упрощается как

$$\mathbf{h}_{N,o_c} = \mathbf{R}_N^{-1} \mathbf{c}_N \left(\mathbf{c}_N^H \mathbf{R}_N^{-1} \mathbf{c}_N \right)^{-1} f. \quad (3.108)$$

Это уравнение для вычисления оптимальных ВК узкополосной ААР, известной как LC Minimum Variance (MV) beam-former. Если в уравнении (3.108) в качестве ограничения используется $|f|$, т. е. действительная величина, то такая ААР называется LC MV Distortionless Response (LC MVDR) beam-former [20].

Линейное ограничение (3.106) позволяет минимизировать выходной сигнал ААР, подавляя сигналы, источники которых по угловому положению не совпадают с полезным сигналом, а значение полезного сигнала, благодаря этому ограничению, поддерживается на заданном уровне.

Таким образом, ААР, ВК которой определяются уравнениями (3.106) и (3.108) или в более общем случае уравнениями (3.87) и (3.93), представляет собой пространственный адаптивный фильтр. Модуль ДН ААР, определяемый как

$$|F(\theta)| = \left| \mathbf{c}_N^H(\theta) \mathbf{h}_N \right|, \quad (3.109)$$

является аналогом АЧХ-фильтра.

3.8. Выводы по главе

В главе 3 рассмотрены основные свойства корреляционной матрицы входных сигналов линейного сумматора, свойства собственных чисел и собственных векторов этой матрицы, введено понятие оптимального фильтра Винера, рассмотрены свойства этого фильтра, а также свойства поверхности его MSE-функции. Также рассмотрен LC оптимальный фильтр Винера и его применение в узкополосных ААР.

ГЛАВА 4

ПОИСК ВИНЕРОВСКОГО РЕШЕНИЯ

В настоящей главе рассматриваются два способа поиска решения задачи винеровской фильтрации: алгоритм Ньютона и алгоритм наискорейшего спуска. Анализируются свойства этих алгоритмов и приводятся результаты их численного моделирования.

4.1. Введение

В главе 3 было показано, что при стационарных входных и требуемом сигналах линейного сумматора его MSE-функция является квадратичной, что позволяет найти координаты ее минимума, соответствующие винеровскому решению. К сожалению, в большинстве встречающихся на практике случаев параметры этой функции неизвестны. Однако, усредняя квадрат ошибки между требуемым $d(k)$ и выходным $y(k)$ сигналами адаптивного фильтра на некотором интервале времени, можно получать оценки MSE-функции. Разработка алгоритмов адаптивной фильтрации заключается в получении математических зависимостей, которые, используя эти оценки, позволяют находить ВК адаптивного линейного сумматора, соответствующие винеровскому решению $\mathbf{h}_{N,o}$ или близкие к нему в некотором смысле.

В настоящей главе рассматриваются два алгоритма поиска винеровского решения в случае, когда параметры \mathbf{R}_N , \mathbf{r}_N и σ_d^2 MSE-функции известны. Это алгоритм Ньютона и алгоритм наискорейшего спуска. Алгоритмы являются итерационными. На каждой итерации алгоритма Ньютона изменения ВК направлены непосредственно к минимуму MSE-функции, а на каждой итерации алгоритма наискорейшего спуска — в направлении, противоположном градиенту MSE-функции, которое совпадает с направлением к минимуму этой функции только в случае, если точка многомерного пространства ВК, в которой оценивается градиент, расположена на одной из главных осей гиперэллипсов сечений $\xi=\text{const}$ (см. главу 3).

Данная глава основана на материалах, представленных в [3, 18, 20, 29].

4.2. Алгоритм Ньютона

Алгоритм Ньютона для поиска минимума MSE-функции может быть получен следующим образом. Если обе части уравнения градиента (3.60) умножить на обратную корреляционную матрицу $\mathbf{R}_N^{-1}[-\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N] = \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi$, то, используя уравнение оптимального винеровского решения (3.63), можно получить уравнение $-\mathbf{R}_N^{-1} \mathbf{r}_N + \mathbf{R}_N^{-1} \mathbf{R}_N \mathbf{h}_N = -\mathbf{h}_{N,o} + \mathbf{h}_N = \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi$. Из него следует, что вектор ВК, соответствующий винеровскому решению, можно вычислить как

$$\mathbf{h}_{N,o} = \mathbf{h}_N - \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi \quad (4.1)$$

при условии, что корреляционная матрица \mathbf{R}_N^{-1} и градиент $\nabla_{\mathbf{h}_N^*} \xi$ точно известны. Здесь \mathbf{h}_N — вектор произвольных ВК.

Однако, как уже отмечалось, на практике \mathbf{R}_N^{-1} и $\nabla_{\mathbf{h}_N^*} \xi$ известны редко. Поэтому уравнение (4.1) обычно модифицируется как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) - \mu \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi(k-1), \quad (4.2)$$

где μ — параметр, называемый шагом сходимости алгоритма. Уравнение (4.2) представляет собой алгоритм Ньютона, с помощью которого находятся координаты минимума MSE-функции, соответствующие винеровскому решению $\mathbf{h}_{N,o}$.

Используя уравнения (3.60) и (3.63), уравнение (4.2) можно преобразовать к виду

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) - \mu \mathbf{R}_N^{-1} [-\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N(k-1)] = \\ &= \mathbf{h}_N(k-1) + \mu \mathbf{R}_N^{-1} \mathbf{r}_N - \mu \mathbf{R}_N^{-1} \mathbf{R}_N \mathbf{h}_N(k-1) = \\ &= \mathbf{h}_N(k-1) + \mu \mathbf{h}_{N,o} - \mu \mathbf{h}_N(k-1) = (1-\mu) \mathbf{h}_N(k-1) + \mu \mathbf{h}_{N,o}. \end{aligned} \quad (4.3)$$

Из (4.3) следует, что при $\mu=1$ вектор $\mathbf{h}_{N,o}(k)$ можно определить за один шаг. Для случая действительных ВК это значение $\mu=0.5$, так как $\nabla_{\mathbf{h}_N^*} \xi(k) = -2\mathbf{r}_N + 2\mathbf{R}_N \tilde{\mathbf{h}}_N(k)$, см. уравнение (3.61).

На рис. 4.1 показаны три случая поиска действительных ВК с помощью уравнения (4.2) при $\mu=0.5$ и трех начальных значениях действительных ВК $\tilde{\mathbf{h}}_N(0) = [0, 0]^T$, $\tilde{\mathbf{h}}_N(0) = [-6, 2]^T$ и $\tilde{\mathbf{h}}_N(0) = [1, 8]^T$. Параметры моделирования этого примера аналогичны параметрам, использованным при получении изображения на рис. 3.6. При $\mu < 0.5$ движение к оптимальному винеровскому решению $\mathbf{h}_{N,o} = [2, 3]^T$ осуществляется в тех же направлениях, что и на рис. 4.1, но с более мелкими шагами, равными $\mu \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi(k-1)$.

Функция $\xi(k)$, вычисляемая с помощью уравнения (3.58) как $\xi(k) = \sigma_d^2 - \mathbf{h}_N^H(k) \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N(k) + \mathbf{h}_N^H(k) \mathbf{R}_N \mathbf{h}_N(k)$, называется обучающей кривой. Такая кривая для алгоритма Ньютона с шагом сходимости $\mu=0.1$ представлена на рис. 4.2. Видно, что эта кривая имеет участок, именуемый переходным процессом, где функция $\xi(k)$ от итерации к итерации меняется «заметно», и участок,

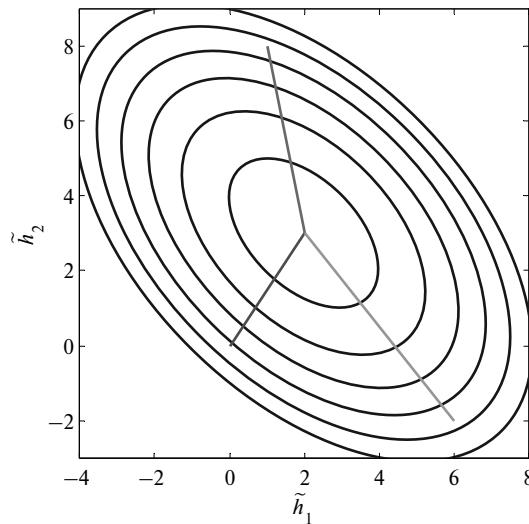


Рис. 4.1. Нахождение винеровского решения за один шаг с помощью алгоритма Ньютона

именуемый установившимся режимом, где значения $\xi(k)$ от итерации к итерации остаются постоянными или «практически постоянными». В рассматриваемом случае в установившемся режиме $\xi(k) = \xi_{\min} = 4$, см. раздел 3.6.

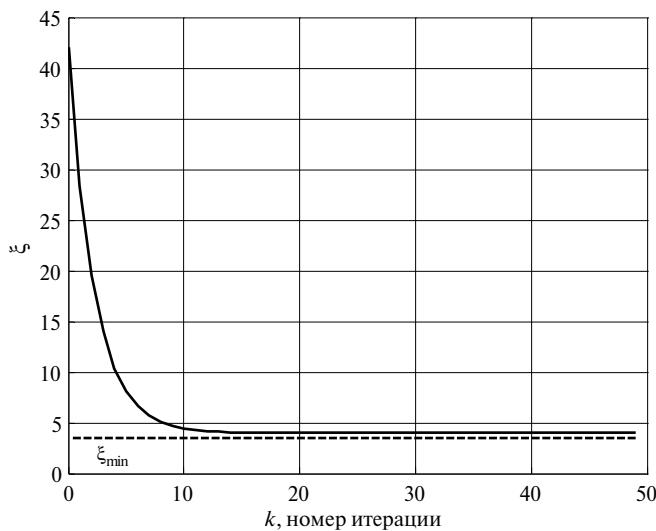


Рис. 4.2. Обучающая кривая алгоритма Ньютона: $\mu = 0.1$, $\tilde{\mathbf{h}}_N(0) = [0, 0]^T$

Переходные процессы часто отображают и в терминах ВК. Графики таких процессов для рассмотренного случая приведены на рис. 4.3. Из рисунка видно,

что движение вектора ВК осуществляется из начальной точки $\tilde{\mathbf{h}}_N(0) = [0, 0]^T$ в направлении вектора оптимального винеровского решения $\tilde{\mathbf{h}}_{N,o} = [2, 3]^T$.

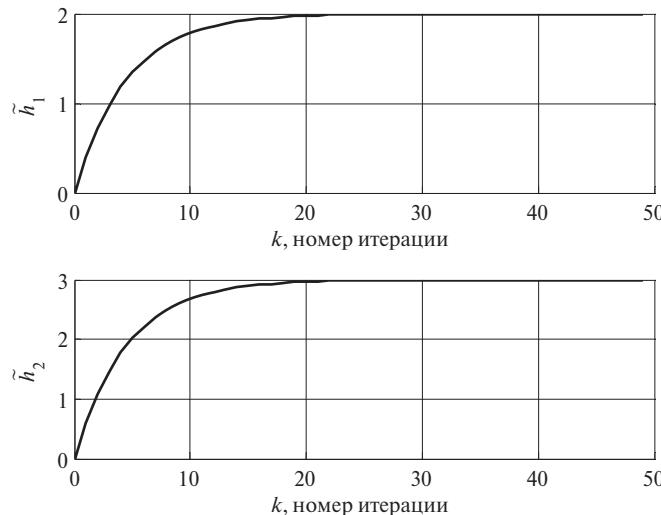


Рис. 4.3. Переходные процессы ВК в алгоритме Ньютона: $\mu = 0.1$, $\tilde{\mathbf{h}}_N(0) = [0, 0]^T$

Допустимый диапазон значений μ , при которых алгоритм Ньютона является устойчивым, т. е. сходится к винеровскому решению, можно определить, если представить уравнение (4.3) в немного измененном виде. Для этого запишем уравнение (4.3) при нескольких значениях $k=1, 2, 3$:

$$\mathbf{h}_N(1) = (1-\mu) \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o},$$

$$\begin{aligned} \mathbf{h}_N(2) &= (1-\mu) \mathbf{h}_N(1) + \mu \mathbf{h}_{N,o} = (1-\mu) \left[(1-\mu) \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o} \right] + \mu \mathbf{h}_{N,o} = \\ &= (1-\mu)^2 \mathbf{h}_N(0) + (1-\mu) \mu \mathbf{h}_{N,o} + \mu \mathbf{h}_{N,o} = (1-\mu)^2 \mathbf{h}_N(0) + [(1-\mu) + 1] \mu \mathbf{h}_{N,o}, \end{aligned}$$

$$\begin{aligned} \mathbf{h}_N(3) &= (1-\mu) \mathbf{h}_N(2) + \mu \mathbf{h}_{N,o} = (1-\mu) \left\{ (1-\mu)^2 \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o} [(1-\mu) + 1] \right\} + \mu \mathbf{h}_{N,o} = \\ &= (1-\mu)^3 \mathbf{h}_N(0) + \left[(1-\mu)^2 + (1-\mu) + 1 \right] \mu \mathbf{h}_{N,o}. \end{aligned}$$

Видно, что в общем случае уравнение (4.3) на k -й итерации можно представить как

$$\begin{aligned} \mathbf{h}_N(k) &= (1-\mu)^k \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o} \sum_{n=0}^{k-1} (1-\mu)^n = \\ &= (1-\mu)^k \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o} \frac{1 - (1-\mu)^{k-1} (1-\mu)}{1 - (1-\mu)} = (1-\mu)^k \mathbf{h}_N(0) + \mu \mathbf{h}_{N,o} \frac{1 - (1-\mu)^k}{1 - (1-\mu)}. \end{aligned} \quad (4.4)$$

В уравнении (4.4) использована формула суммы k членов геометрической прогрессии

$$s_k = \frac{a_1 - a_k q}{1 - q}, \quad (4.5)$$

где $a_1 = \mu^0 = 1$ — первый член, $a_k = (1 - \mu)^k$ — последний член и

$$q = \frac{a_k}{a_{k-1}} = \frac{(1 - \mu)^k}{(1 - \mu)^{k-1}} = 1 - \mu \quad (4.6)$$

— знаменатель геометрической прогрессии.

Уравнение (4.4) можно преобразовать как

$$\begin{aligned} \mathbf{h}_N(k) &= (1 - \mu)^k \mathbf{h}_N(0) + \mathbf{h}_{N,o} \left[1 - (1 - \mu)^k \right] = \\ &= \mathbf{h}_{N,o} + (1 - \mu)^k \mathbf{h}_N(0) - \mathbf{h}_{N,o} (1 - \mu)^k = \mathbf{h}_{N,o} + (1 - \mu)^k \left[\mathbf{h}_N(0) - \mathbf{h}_{N,o} \right]. \end{aligned} \quad (4.7)$$

Из уравнения (4.7) следует, что для того, чтобы при $k \rightarrow \infty$ значение $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$, необходимо, чтобы $(1 - \mu)^k \rightarrow 0$, для чего должно выполняться условие $|1 - \mu| < 1$. Это условие можно записать в виде двойного неравенства $-1 < 1 - \mu < 1$, откуда допустимые границы параметра μ определяются как

$$0 < \mu < 2. \quad (4.8)$$

Соответственно, в случае действительных ВК

$$0 < \mu < 1, \quad (4.9)$$

так как согласно уравнению (3.61) $\nabla_{\tilde{\mathbf{h}}_N} \xi = -2\mathbf{r}_N + 2\mathbf{R}_N \tilde{\mathbf{h}}_N$.

Таким образом, уравнения (4.8), (4.9) являются условием, гарантирующим устойчивость алгоритма Ньютона, т. е. сходимость итерационной процедуры (4.2) к винеровскому решению.

4.3. Постоянные времени алгоритма Ньютона

Скорость движения алгоритма Ньютона в направлении $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$ можно оценить с помощью постоянной времени переходного процесса вычисления элементов вектора $\mathbf{h}_N(k)$. Из уравнения (4.7) следует, что этот переходный процесс определяется множителем $(1 - \mu)^k$ во втором слагаемом. При малых значениях μ его можно трактовать как экспоненциально затухающий, т. е. $(1 - \mu)^k \left[\mathbf{h}_N(0) - \mathbf{h}_{N,o} \right] = \Delta \mathbf{h}_N(0) e^{-\frac{k}{k_{t_h}}}$, где k_{t_h} — постоянная времени переходного процесса в терминах числа отсчетов обрабатываемых сигналов, а $\Delta \mathbf{h}_N(0) = \left[\mathbf{h}_N(0) - \mathbf{h}_{N,o} \right] = \text{const.}$

Постоянную времени k_{t_h} можно определить как число отсчетов относительно текущего отсчета k , в течение которых значение второго слагаемого $(1 - \mu)^{k+k_{t_h}} \left[\mathbf{h}_N(0) - \mathbf{h}_{N,o} \right]$ в уравнении (4.7) уменьшится в $e \approx 2.7183$ раза по сравнению со значением $(1 - \mu)^k \left[\mathbf{h}_N(0) - \mathbf{h}_{N,o} \right]$, т. е.

$$\frac{\Delta \mathbf{h}_N(0) e^{-\frac{k}{k_{\tau_h}}}}{\Delta \mathbf{h}_N(0) e^{-\frac{k+k_{\tau_h}}{k_{\tau_h}}}} = \frac{(1-\mu)^k [\mathbf{h}_N(0) - \mathbf{h}_{N,o}]}{(1-\mu)^{k+k_{\tau_h}} [\mathbf{h}_N(0) - \mathbf{h}_{N,o}]} = \frac{(1-\mu)^k}{(1-\mu)^{k+k_{\tau_h}}} = (1-\mu)^{-k_{\tau_h}} = e. \quad (4.10)$$

Из выражения (4.10) можно определить k_{τ_h} как

$$k_{\tau_h} = -\log_{(1-\mu)}(1-\mu)^{-k_{\tau_h}} = -\log_{(1-\mu)} e = -\frac{\ln e}{\ln(1-\mu)} = -\frac{1}{\ln(1-\mu)} \approx -\frac{1}{-\mu} = \frac{1}{\mu} \quad (4.11)$$

при $\mu \ll 1$.

Здесь использована формула перехода от одного основания логарифма a к другому основанию c

$$\log_a b = \log_c b / \log_c a \quad (4.12)$$

и первый член из формулы разложения логарифма

$$\ln(1-x) = -\left[x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots \right], \quad (4.13)$$

которая справедлива при $-1 \leq x < 1$, где $x = 1 - \mu$.

Справедливость уравнения (4.11) можно продемонстрировать и другим способом. Известно, что

$$a^x = e^{x \ln a}. \quad (4.14)$$

Тогда из уравнения

$$\frac{a^{x_1}}{a^{x_2}} = \frac{e^{x_1 \ln a}}{e^{x_2 \ln a}} = e^{(x_1 - x_2) \ln a} = e \quad (4.15)$$

можно определить, что $(x_1 - x_2) \ln a = 1$, откуда $x_1 - x_2 = 1 / \ln a$. При $x_1 = k$, $x_2 = k + k_{\tau_h}$ и $a = 1 - \mu$ видно, что

$$x_2 - x_1 = k + k_{\tau_h} - k = k_{\tau_h} = -\frac{1}{\ln(1-\mu)} \approx -\frac{1}{-\mu} = \frac{1}{\mu}, \quad (4.16)$$

т. е. полученное выражение совпадает с выражением (4.11).

В случае действительных ВК постоянная времени k_{τ_h} определяется как

$$k_{\tau_h} = \frac{1}{2\mu}. \quad (4.17)$$

Постоянную времени переходного процесса обучающей кривой $\xi(k)$ алгоритма Ньютона можно оценить следующим образом. Из выражения (4.7) определим

$$\Delta \mathbf{h}_N(k) = \mathbf{h}_N(k) - \mathbf{h}_{N,o} = (1-\mu)^k [\mathbf{h}_N(0) - \mathbf{h}_{N,o}] = (1-\mu)^k \Delta \mathbf{h}_N(0). \quad (4.18)$$

Затем используем в уравнении MSE-функции (3.72). Тогда

$$\begin{aligned}\xi(k) &= \xi_{\min} + \Delta \mathbf{h}_N^H(k) \mathbf{R}_N \Delta \mathbf{h}_N(k) = \xi_{\min} + \left[(1-\mu)^k \Delta \mathbf{h}_N(0) \right]^H \mathbf{R}_N \left[(1-\mu)^k \Delta \mathbf{h}_N(0) \right] = \\ &= \xi_{\min} + (1-\mu)^{2k} \Delta \mathbf{h}_N^H(0) \mathbf{R}_N \Delta \mathbf{h}_N(0).\end{aligned}\quad (4.19)$$

Из уравнения (4.19) видно, что переходный процесс обучающей кривой $\xi(k)$ определяется множителем $(1-\mu)^{2k}$ подобно тому, как переходный процесс вектора ВК $\mathbf{h}_N(k)$ согласно выражению (4.7) определяется множителем $(1-\mu)^k$. Поэтому, выполнив преобразования, аналогичные (4.10)–(4.16), получим, что в случае комплексных ВК постоянная времени обучающей кривой $k_{\tau_{\xi}}$ определяется как

$$k_{\tau_{\xi}} = \frac{1}{2\mu}, \quad (4.20)$$

а в случае действительных ВК — как

$$k_{\tau_{\xi}} = \frac{1}{4\mu}. \quad (4.21)$$

Таким образом, из уравнений (4.8) и (4.9) следует, что в алгоритме Ньютона значение шага сходимости, гарантирующее его устойчивость, не зависит от собственных чисел корреляционной матрицы обрабатываемых сигналов. Из (4.11), (4.17), (4.20) и (4.21) также следует, что постоянная времени переходного процесса движения вектора ВК $\mathbf{h}_N(k)$ к вектору $\mathbf{h}_{N,o}$ в два раза больше постоянной времени обучающей кривой $\xi(k)$. При этом все компоненты вектора ВК $\mathbf{h}_N(k)$ имеют одну и ту же постоянную времени, что обуславливает одинаковую скорость сходимости этих компонент к вектору $\mathbf{h}_{N,o}$ (см. рис. 4.3). Постоянные вре-

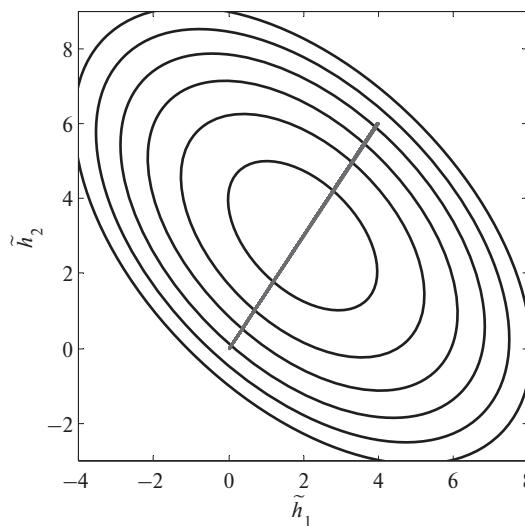


Рис. 4.4. Алгоритм Ньютона: $\mu=1$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

мени (4.11), (4.17), (4.20) и (4.21) также не зависят от собственных чисел корреляционной матрицы.

Поведение алгоритма Ньютона при разных значениях шага сходимости μ демонстрируется на рис. 4.4—4.6. Параметры моделирования в этих примерах также аналогичны параметрам, использованным при получении изображения на рис. 3.6.

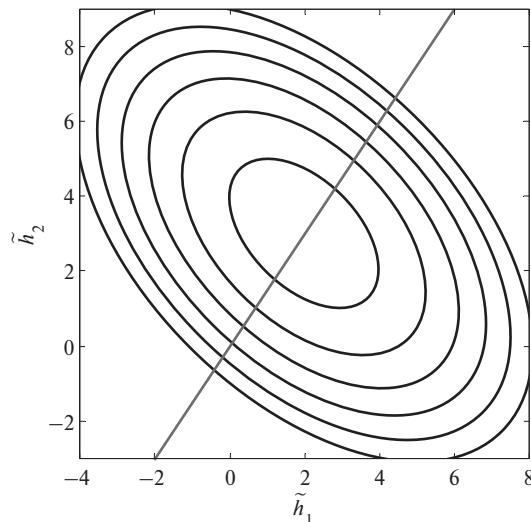


Рис. 4.5. Алгоритм Ньютона: $\mu=1.05$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

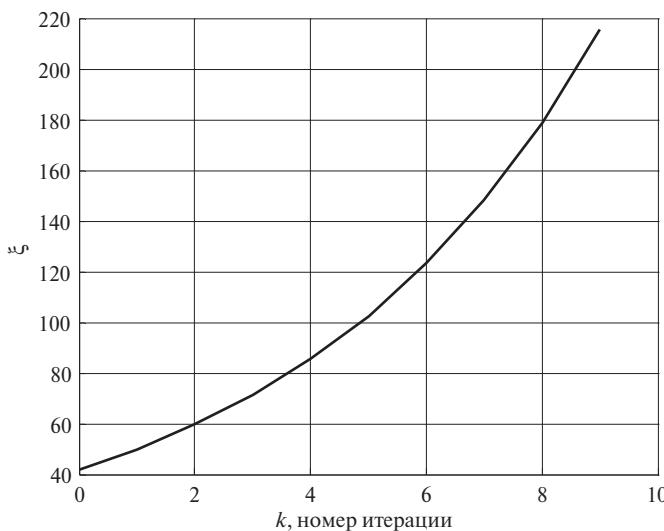


Рис. 4.6. Обучающая кривая алгоритма Ньютона: $\mu=1.05$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$



Из уравнения (4.7) следует, что при $\mu=2$ в случае комплексных ВК и при $\mu=1$ в случае действительных ВК алгоритм совершает колебательные движения между точками ВК $\mathbf{h}_N(k)=\mathbf{h}_N(0)$ и $\mathbf{h}_N(k)=2\mathbf{h}_{N,o}-\mathbf{h}_N(0)$, минуя точку $\mathbf{h}_N(k)=\mathbf{h}_{N,o}$. Такое движение при действительных ВК показано на рис. 4.4. При $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$ и при $\tilde{\mathbf{h}}_{N,o}=[2, 3]^T$ вычисляемые по этим формулам значения действительных ВК соответственно равны $\tilde{\mathbf{h}}_N(k)=[0, 0]^T$ и $\tilde{\mathbf{h}}_N(k)=[4, 6]^T$.

При $\mu>1$ для действительных ВК итерационный процесс поиска винеровского решения с помощью алгоритма Ньютона расходится как это показано на рис. 4.5. Обучающая кривая для этого случая показана на рис. 4.6, где $\xi(0)=42$ (см. разд. 3.6).

4.4. Алгоритм наискорейшего спуска

Другим алгоритмом поиска ВК адаптивного линейного сумматора, соответствующих минимуму MSE-функции, является алгоритм наискорейшего спуска. Этот алгоритм также представляет собой итерационную процедуру. В отличие от алгоритма Ньютона, алгоритм наискорейшего спуска на каждой итерации изменяет вектор ВК не в непосредственном направлении минимума MSE-функции, а в направлении, противоположном градиенту этой функции в точке многомерного пространства, определяемой текущими значениями ВК. То есть, движение к минимуму MSE-функции осуществляется в направлении, обратном максимальной скорости изменения целевой функции, что и определило название «алгоритм наискорейшего спуска».

В алгоритме наискорейшего спуска используется точное аналитическое значение градиента (3.60), а также не требуется обращать корреляционную матрицу.

С одной стороны, получение значений матрицы \mathbf{R}_N и вектора \mathbf{r}_N , используемых в аналитическом выражении градиента, является сложной задачей. Поэтому, подобно алгоритму Ньютона, применять алгоритм наискорейшего спуска на практике неудобно. С другой стороны, в других градиентных алгоритмах, рассматриваемых в главе 5, используются лишь оценки градиента, а значит, их поведение может только приближаться к поведению алгоритма наискорейшего спуска. Поэтому, несмотря на сложность практической реализации, алгоритм наискорейшего спуска можно рассматривать как образцовый алгоритм, с поведением которого можно сравнивать другие градиентные алгоритмы.

Вычисление ВК в алгоритме наискорейшего спуска осуществляется в соответствие с уравнением

$$\mathbf{h}_N(k)=\mathbf{h}_N(k-1)-\mu \nabla_{\mathbf{h}_N} \xi(k-1), \quad (4.22)$$

из которого видно, что изменения ВК направлены в сторону, противоположную градиенту, т. е. в сторону, противоположную максимальной скорости изменения MSE-функции в точке $\mathbf{h}_N(k-1)$.

Используя уравнение градиента MSE-функции (3.60), уравнение (4.22) можно представить как

$$\begin{aligned}\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) - \mu [-\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N(k-1)] = \\ &= \mathbf{h}_N(k-1) + \mu \mathbf{r}_N - \mu \mathbf{R}_N \mathbf{h}_N(k-1).\end{aligned}\quad (4.23)$$

Для анализа условий устойчивости алгоритма наискорейшего спуска уравнение (4.23) необходимо привести к виду, подобному уравнению (4.7). Однако это не представляется возможным, так как матрица \mathbf{R}_N не является диагональной. В то же время этот анализ можно выполнить, если уравнение (4.23) выразить в терминах ВК (3.74) $\mathbf{v}_N(k) = \mathbf{Q}_N^H \Delta \mathbf{h}_N(k)$ в повернутой и смещенной системе координат.

Для этого в (4.23) используем выражение $\mathbf{r}_N = \mathbf{R}_N \mathbf{h}_{N,o}$, следующее из уравнения (3.63), и вычтем вектор $\mathbf{h}_{N,o}$ из левой и правой частей уравнения (4.23), как

$$\mathbf{h}_N(k) - \mathbf{h}_{N,o} = \mathbf{h}_N(k-1) - \mathbf{h}_{N,o} + \mu \mathbf{R}_N \mathbf{h}_{N,o} - \mu \mathbf{R}_N \mathbf{h}_N(k-1). \quad (4.24)$$

Тогда из уравнения (4.24) следует, что

$$\Delta \mathbf{h}_N(k) = \mathbf{h}_N(k-1) - \mu \mathbf{R}_N \Delta \mathbf{h}_N(k-1) = (\mathbf{I}_N - \mu \mathbf{R}_N) \Delta \mathbf{h}_N(k-1), \quad (4.25)$$

а из уравнения (3.74) — что

$$\mathbf{Q}_N \mathbf{v}_N(k) = \mathbf{Q}_N \mathbf{Q}_N^H \Delta \mathbf{h}_N(k) = \Delta \mathbf{h}_N(k). \quad (4.26)$$

С учетом соотношения (4.26) уравнение (4.25) преобразуется к виду

$$\mathbf{Q}_N \mathbf{v}_N(k) = (\mathbf{I}_N - \mu \mathbf{R}_N) \mathbf{Q}_N \mathbf{v}_N(k-1). \quad (4.27)$$

Умножив уравнение (4.27) слева на матрицу \mathbf{Q}_N^H как

$$\begin{aligned}\mathbf{Q}_N^H \mathbf{Q}_N \mathbf{v}_N(k) &= \mathbf{Q}_N^H (\mathbf{I}_N - \mu \mathbf{R}_N) \mathbf{Q}_N \mathbf{v}_N(k-1) = \\ &= (\mathbf{Q}_N^H \mathbf{I}_N \mathbf{Q}_N - \mu \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N) \mathbf{v}_N(k-1)\end{aligned}\quad (4.28)$$

и используя уравнение (3.46) $\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{\Lambda}_N$, получим, что

$$\mathbf{v}_N(k) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(k-1). \quad (4.29)$$

Для уравнения (4.29) выполним преобразования, аналогичные преобразованиям при получении уравнения (4.4), т. е.

$$\mathbf{v}_N(1) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(0),$$

$$\mathbf{v}_N(2) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(1) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N)(\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(0),$$

$$\mathbf{v}_N(3) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(2) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N)(\mathbf{I}_N - \mu \mathbf{\Lambda}_N)(\mathbf{I}_N - \mu \mathbf{\Lambda}_N) \mathbf{v}_N(0).$$

Из этих уравнений следует, что

$$\mathbf{v}_N(k) = (\mathbf{I}_N - \mu \mathbf{\Lambda}_N)^k \mathbf{v}_N(0). \quad (4.30)$$

Выражение (4.30) в развернутом виде представляется как

$$\mathbf{v}_N(k) = \begin{bmatrix} (1 - \mu \lambda_1)^k & 0 & \dots & 0 \\ 0 & (1 - \mu \lambda_2)^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (1 - \mu \lambda_N)^k \end{bmatrix} \mathbf{v}_N(0). \quad (4.31)$$

Из (4.31) следует, что для того, чтобы при $k \rightarrow \infty$ значение $\mathbf{v}_N(k) \rightarrow \mathbf{0}_N$ при $\mathbf{v}_N(0) \neq \mathbf{0}_N$, или, что то же самое, что $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$, необходимо, чтобы $(1 - \mu \lambda_n)^k \rightarrow 0$ для всех $n = 1, 2, \dots, N$. Для этого должно выполняться условие $|1 - \mu \lambda_n| < 1$. Это условие можно записать в виде двойного неравенства $-1 < 1 - \mu \lambda_n < 1$, откуда следует, что $0 < \mu < 2/\lambda_n$. Общее условие для всех значений λ_n в случае комплексных ВК можно сформулировать как

$$0 < \mu < 2/\lambda_{\max}, \quad (4.32)$$

а в случае действительных ВК — как

$$0 < \mu < 1/\lambda_{\max}. \quad (4.33)$$

Таким образом, уравнения (4.32) и (4.33) являются условием, гарантирующим устойчивость алгоритма наискорейшего спуска, т. е. сходимость итерационной процедуры (4.31), или то же самое, что (4.23) или (4.25), к винеровскому решению $\mathbf{h}_{N,o}$.

4.5. Постоянные времени алгоритма наискорейшего спуска

Скорость движения алгоритма наискорейшего спуска в направлении $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$ можно оценить с помощью постоянных времени переходных процессов вычисления элементов вектора $\mathbf{v}_N(k)$ в уравнении (4.31). Эта оценка осуществляется аналогично уравнениям (4.10)–(4.17).

Переходные процессы каждой из компонент вектора $\mathbf{v}_N(k)$ в алгоритме наискорейшего спуска, подобно алгоритму Ньютона, можно трактовать как экспоненциально затухающие, т. е. $(1 - \mu \lambda_n)^k v_n(0) = C_n e^{-\frac{k}{k_{\tau_{n,v}}}}$, где $k_{\tau_{n,v}}$ — постоянная времени переходного процесса весового коэффициента $v_n(k)$ в терминах числа отсчетов обрабатываемых сигналов, а $C_n = v_n(0) = \text{const}_n$.

Постоянную времени $k_{\tau_{n,v}}$ можно определить как число отсчетов относительно текущего отсчета k , в течение которых значение $(1-\mu\lambda_n)^{k+k_{\tau_{n,v}}} v_n(0)$ уменьшится в $e \approx 2.7183$ раза по сравнению со значением $(1-\mu\lambda_n)^k v_n(0)$, т. е.

$$\frac{C_n e^{-\frac{k}{k_{\tau_{n,v}}}}}{C_n e^{-\frac{k+k_{\tau_{n,v}}}{k_{\tau_{n,v}}}}} = \frac{(1-\mu\lambda_n)^k v_n(0)}{(1-\mu\lambda_n)^{k+k_{\tau_{n,v}}} v_n(0)} = \frac{(1-\mu\lambda_n)^k}{(1-\mu\lambda_n)^{k+k_{\tau_{n,v}}}} = (1-\mu\lambda_n)^{-k_{\tau_{n,v}}} = e. \quad (4.34)$$

Из условия (4.34), аналогично (4.11), можно определить $k_{\tau_{n,v}}$ как

$$\begin{aligned} k_{\tau_{n,v}} &= -\log_{(1-\mu\lambda_n)} (1-\mu\lambda_n)^{-k_{\tau_{n,v}}} = -\log_{(1-\mu\lambda_n)} e = \\ &= -\frac{\ln e}{\ln(1-\mu\lambda_n)} = -\frac{1}{\ln(1-\mu\lambda_n)} \approx -\frac{1}{-\mu\lambda_n} = \frac{1}{\mu\lambda_n} \end{aligned} \quad (4.35)$$

при $\mu \ll 1$.

Переход к исходной системе координат можно осуществить следующим образом. Из выражения (4.26) следует, что $\mathbf{Q}_N \mathbf{v}_N(k) = \Delta \mathbf{h}_N(k) = \mathbf{h}_N(k) - \mathbf{h}_{N,o}$, откуда

$$\mathbf{h}_N(k) = \mathbf{h}_{N,o} + \mathbf{Q}_N \mathbf{v}_N(k) = \mathbf{h}_{N,o} + [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N] \begin{bmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_N(k) \end{bmatrix} = \mathbf{h}_{N,o} + \sum_{m=1}^N \mathbf{q}_m v_m(k). \quad (4.36)$$

Так как согласно уравнению (4.31) $v_m(k) = (1-\mu\lambda_m)^k v_m(0)$, то

$$h_n(k) = h_{o,n} + \sum_{m=1}^N q_{m,n} (1-\mu\lambda_m)^k v_m(0), \quad (4.37)$$

где $q_{m,n}$ — это n -й элемент m -го собственного вектора \mathbf{q}_m корреляционной матрицы \mathbf{R}_N . Из уравнения (4.37) следует, что каждый ВК $h_n(k)$ в алгоритме наискорейшего спуска сходится к соответствующему ВК винеровского решения $h_{o,n}(k)$

пропорционально $\sum_{m=1}^N (1-\mu\lambda_m)^k$. Определить из-за этой суммы постоянную времени $k_{\tau_{n,h}}$ для ВК $h_n(k)$, подобно уравнению (4.35), не представляется возможным. Тем не менее, для корреляционной матрицы \mathbf{R}_N с разными собственными числами и $\mu = \text{const}$ сходимость уравнения (4.37) будет определяться значением λ_{\min} , так как слагаемое $(1-\mu\lambda_{\min})^k \rightarrow 0$ медленнее всех других аналогичных слагаемых, содержащих $\lambda_n \neq \lambda_{\min}$, в этом уравнении. Другими словами, согласно уравнениям (4.35) и (4.37) каждая из постоянных времени $k_{\tau_{n,h}}$ находится в пределах

$$\frac{1}{\mu\lambda_{\max}} \leq k_{\tau_{n,h}} \leq \frac{1}{\mu\lambda_{\min}}. \quad (4.38)$$

В случае действительных ВК постоянная времени $k_{\tau_{n,h}}$ находится в пределах

$$\frac{1}{2\mu\lambda_{\max}} \leq k_{\tau_{n,h}} \leq \frac{1}{2\mu\lambda_{\min}}. \quad (4.39)$$

Таким образом, если корреляционная матрица \mathbf{R}_N имеет сильно различающиеся собственные числа, т. е. $\chi = \lambda_{\max} / \lambda_{\min} \gg 1$, то длительность переходного процесса вычисления ВК аддитивного линейного сумматора, использующего алгоритм наискорейшего спуска, будет определяться наименьшим собственным числом.

Постоянную времени переходного процесса обучающей кривой $\xi(k)$ в алгоритме наискорейшего спуска можно оценить следующим образом. Согласно уравнению (3.73) MSE-функция определяется как $\xi(k) = \xi_{\min} + \mathbf{v}_N^H(k) \mathbf{\Lambda}_N \mathbf{v}_N(k)$. Если в этом уравнении использовать выражение (4.30), то

$$\begin{aligned}
 \xi(k) &= \xi_{\min} + \left[(\mathbf{I}_N - \mu \mathbf{\Lambda}_N)^k \mathbf{v}_N(0) \right]^H \mathbf{\Lambda}_N \left[(\mathbf{I}_N - \mu \mathbf{\Lambda}_N)^k \mathbf{v}_N(0) \right] = \\
 &= \xi_{\min} + \mathbf{v}_N^H(0) (\mathbf{I}_N - \mu \mathbf{\Lambda}_N)^k \mathbf{\Lambda}_N (\mathbf{I}_N - \mu \mathbf{\Lambda}_N)^k \mathbf{v}_N(0) = \\
 &= \xi_{\min} + \left[v_1^*(0), v_2^*(0), \dots, v_N^*(0) \right] \begin{bmatrix} (1-\lambda_1)^k & 0 & \dots & 0 \\ 0 & (1-\lambda_2)^k & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & (1-\lambda_N)^k \end{bmatrix} \times \quad (4.40) \\
 &\quad \times \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \begin{bmatrix} (1-\lambda_1)^k & 0 & \dots & 0 \\ 0 & (1-\lambda_2)^k & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & (1-\lambda_N)^k \end{bmatrix} \begin{bmatrix} v_1(0) \\ v_2(0) \\ \vdots \\ v_N(0) \end{bmatrix} = \\
 &= \xi_{\min} + \sum_{n=1}^N (1-\mu \lambda_n)^{2k} \lambda_n v_n^*(0) v_n(0) = \xi_{\min} + \sum_{n=1}^N (1-\mu \lambda_n)^{2k} \lambda_n |v_n(0)|^2.
 \end{aligned}$$

Здесь учтено, что матрицы \mathbf{I}_N и $\mathbf{\Lambda}_N$ являются действительными и диагональными.

Следовательно, переходный процесс обучающей кривой (4.40) также определяется суммой затухающих экспонент $\sum_{n=1}^N (1-\mu \lambda_n)^{2k}$. Выполнив преобразования, аналогичные (4.34) и (4.35), легко заметить, что $2k_{\tau_{n,\xi}} \leq \frac{1}{\mu \lambda_n}$, или

$$k_{\tau_{n,\xi}} \leq \frac{1}{2\mu \lambda_n}. \quad (4.41)$$

Это означает, что значение каждой из постоянных времени $k_{\tau_{n,\xi}}$ находится в пределах

$$\frac{1}{2\mu \lambda_{\max}} \leq k_{\tau_{n,\xi}} \leq \frac{1}{2\mu \lambda_{\min}}. \quad (4.42)$$

В случае действительных ВК условие (4.42) формулируется как

$$\frac{1}{4\mu\lambda_{\max}} \leq k_{\tau_{n,\xi}} \leq \frac{1}{4\mu\lambda_{\min}}. \quad (4.43)$$

Таким образом, согласно уравнениям (4.37) и (4.40) сходимость ВК и MSE-функции к оптимальным значениям определяется минимальным собственным числом корреляционной матрицы входных сигналов адаптивного линейного сумматора, а также зависит от шага сходимости μ . Чем меньше μ , тем больше постоянные времена переходных процессов и тем медленнее происходит движение вектора ВК в направлении оптимального винеровского решения.

Результаты моделирования алгоритма наискорейшего спуска в таких же условиях, как и моделирование алгоритма Ньютона (см. раздел 4.3), представлены на рис. 4.7–4.20.

На рис. 4.7, рис. 4.10, рис. 4.13 и рис. 4.16 показаны переходные процессы алгоритма наискорейшего спуска при $\mu=0.05$, $\mu=0.1$, $\mu=0.3$, $\mu=0.4$ и трех начальных значениях ВК.

Обучающие кривые для случая $\mathbf{h}_N(0)=[0, 0]^T$ приведены на рис. 4.8, рис. 4.11, рис. 4.14 и рис. 4.17, а графики переходных процессов при вычислении ВК — на рис. 4.9, рис. 4.12, рис. 4.15 и рис. 4.18. Графики переходных процессов для случаев $\tilde{\mathbf{h}}_N(0)=[-6, 2]^T$ и $\tilde{\mathbf{h}}_N(0)=[1, 8]^T$ и $\mu=0.05$ приведены на рис. 4.19 и рис. 4.20.

Из этих рисунков следует, что чем меньше значение μ , тем более плавные переходные процессы. Для действительных ВК согласно условию (4.33) допустимое значение шага сходимости определяется как $0 < \mu < 1/3$, так как в рассматриваемых случаях $\lambda_{\max} = \lambda_2 = 3$ (см. раздел 3.6).

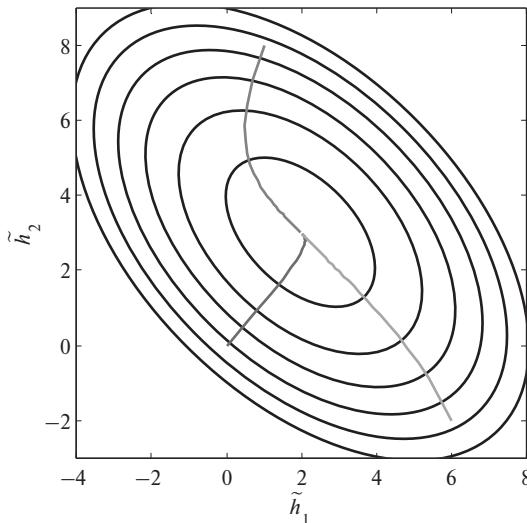


Рис. 4.7. Нахождение винеровского решения с помощью алгоритма наискорейшего спуска: $\mu=0.05$

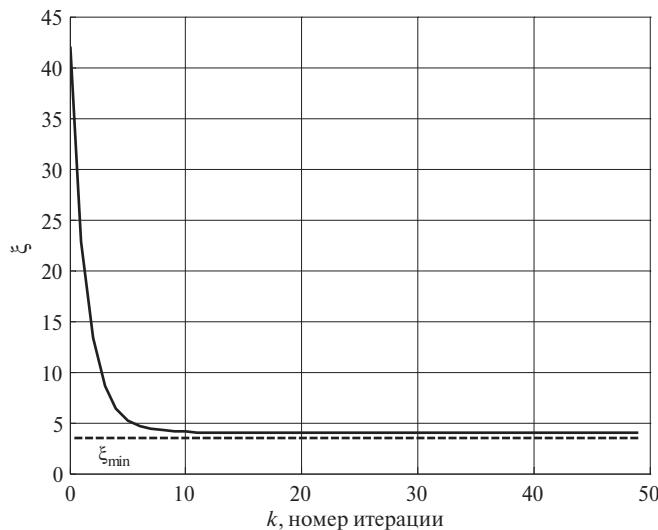


Рис. 4.8. Обучающая кривая алгоритма наискорейшего спуска: $\mu=0.05$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

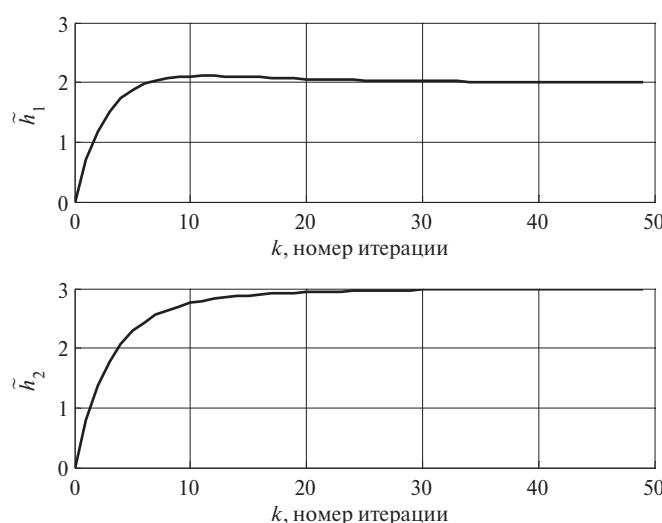


Рис. 4.9. Переходные процессы ВК в алгоритме наискорейшего спуска: $\mu=0.05$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

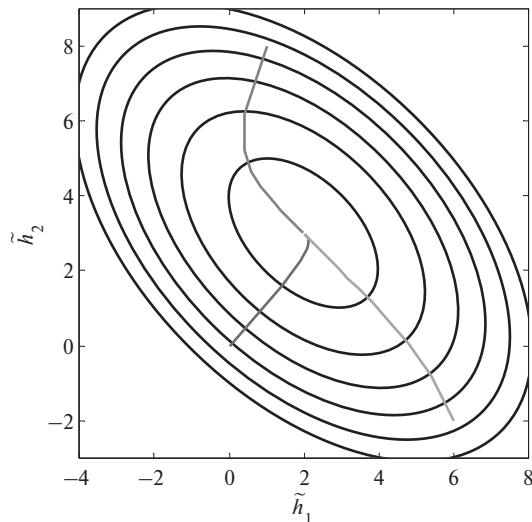


Рис. 4.10. Нахождение винеровского решения с помощью алгоритма наискорейшего спуска: $\mu=0.1$

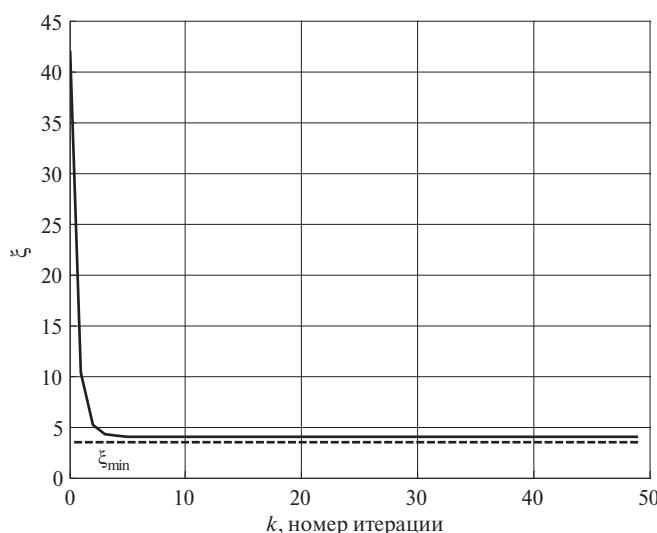


Рис. 4.11. Обучающая кривая алгоритма наискорейшего спуска: $\mu=0.1$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

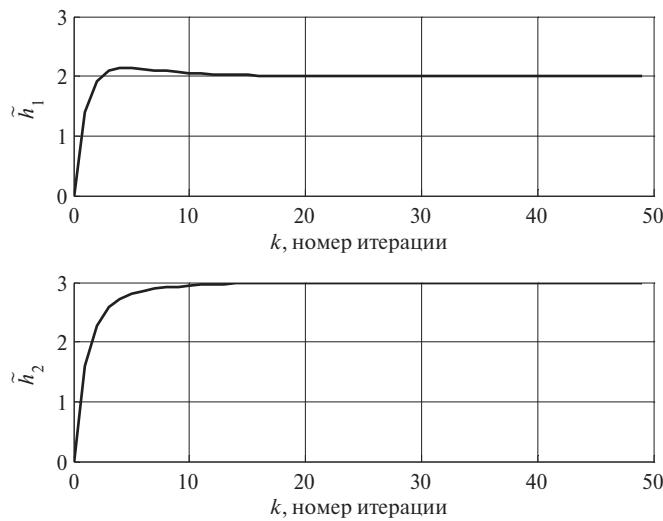


Рис. 4.12. Переходные процессы ВК в алгоритме наискорейшего спуска: $\mu=0.1$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

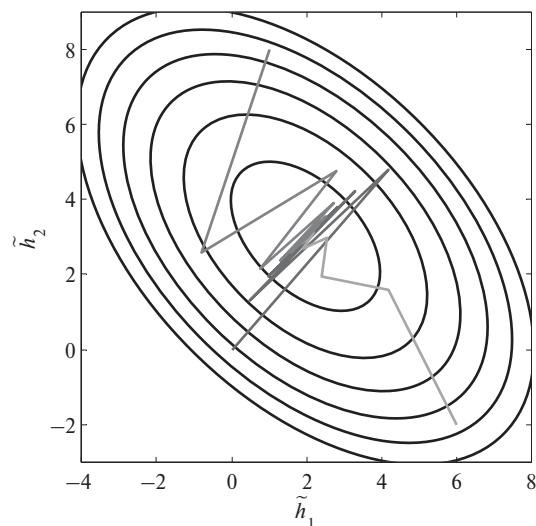


Рис. 4.13. Нахождение винеровского решения с помощью алгоритма наискорейшего спуска: $\mu=0.3$

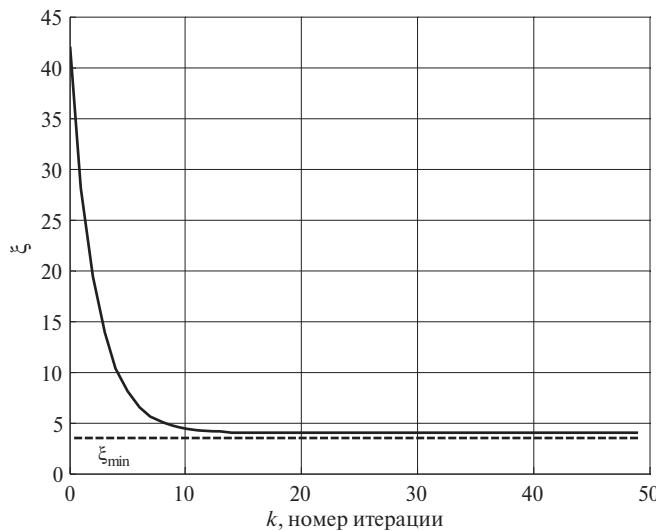


Рис. 4.14. Обучающая кривая алгоритма наискорейшего спуска: $\mu=0.3$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

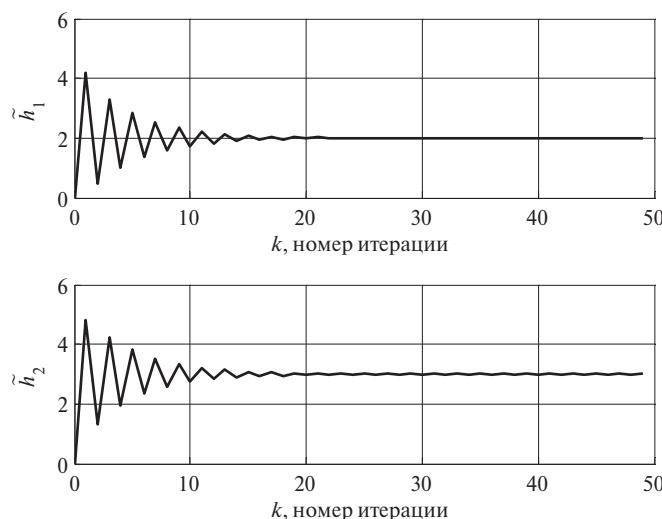


Рис. 4.15. Переходные процессы BK в алгоритме наискорейшего спуска: $\mu=0.3$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

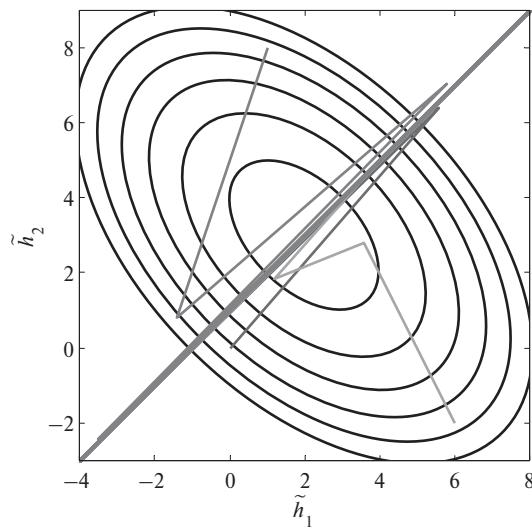


Рис. 4.16. Нахождение винеровского решения с помощью алгоритма наискорейшего спуска: $\mu=0.4$

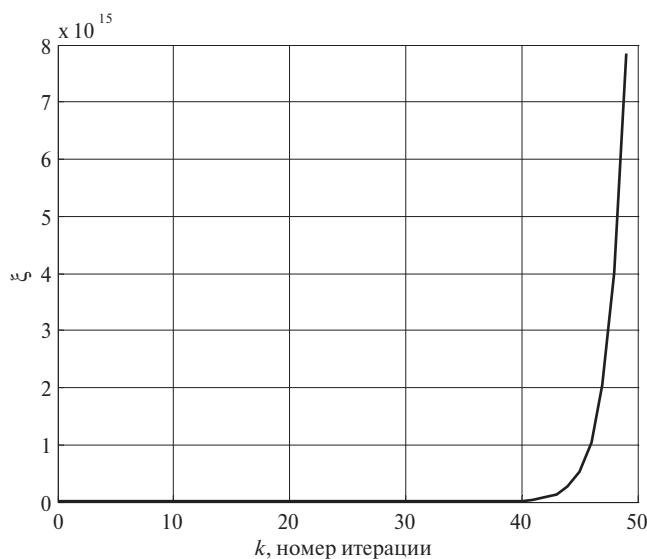


Рис. 4.17. Обучающая кривая алгоритма наискорейшего спуска: $\mu=0.4$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

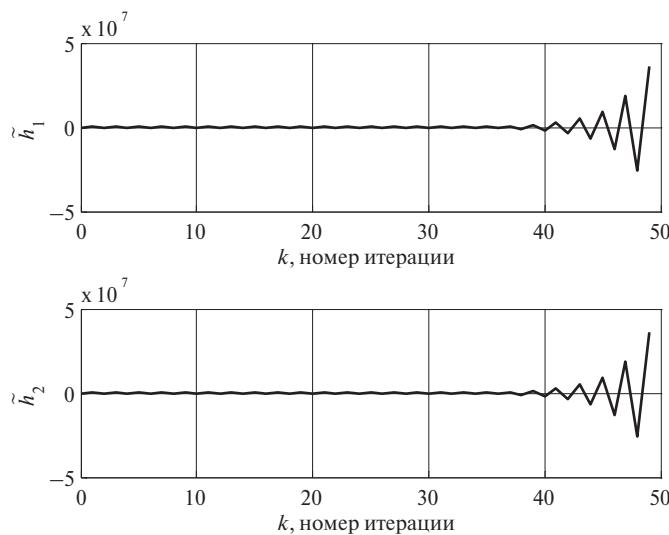


Рис. 4.18. Переходные процессы BK в алгоритме наискорейшего спуска: $\mu=0.4$, $\tilde{\mathbf{h}}_N(0)=[0, 0]^T$

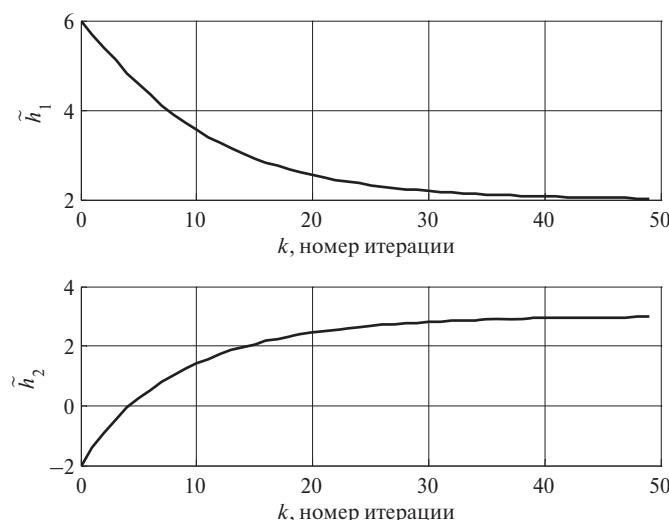


Рис. 4.19. Переходные процессы BK в алгоритме наискорейшего спуска: $\mu=0.05$, $\tilde{\mathbf{h}}_N(0)=[6, -2]^T$

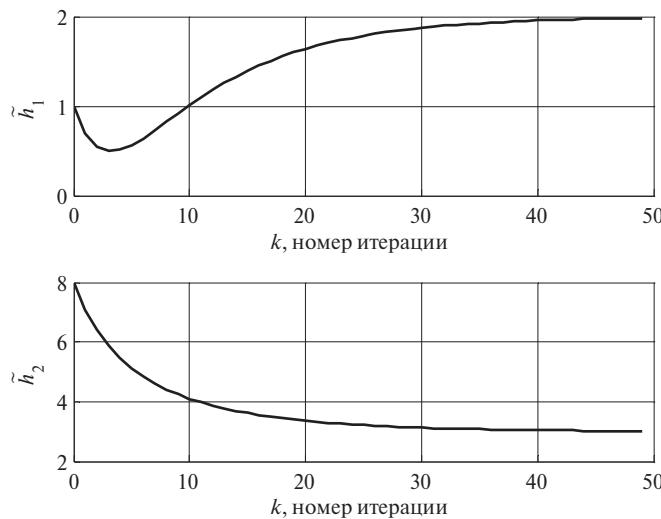


Рис. 4.20. Переходные процессы ВК в алгоритма наискорейшего спуска: $\mu=0.05$, $\tilde{h}_N(0)=[1, 8]^T$

Приведенные результаты моделирования позволяют сделать следующий вывод. При $\mu \rightarrow \mu_{\max}$ (рис. 4.13—4.15) длительность переходного процесса вычисления ВК увеличивается, так как изменения значений ВК на каждой итерации становятся большими, и точка на поверхности целевой функции «перепрыгивает» в другую точку с несколько меньшим значением MSE, но большим, чем в случае $\mu \ll \mu_{\max}$, когда движения к точке $\tilde{h}_{N,o}$ осуществляется не зигзагообразно, а плавно.

При превышении допустимого значения $\mu \approx 0.33$, как и следовало ожидать, алгоритм наискорейшего спуска становится неустойчивым (см. рис. 4.16—4.18).

Из сравнения рис. 4.8, рис. 4.11, рис. 4.14 и рис. 4.9, рис. 4.12, рис. 4.15 видно, что переходные процессы ВК длиннее, чем переходные процессы обучающих кривых, что подтверждает справедливость уравнений (4.35) и (4.41).

Рис. 4.9, рис. 4.19 и рис. 4.20 демонстрируют тот факт, что длительность переходных процессов зависит от начальных условий, т. е. от того, насколько далеки или близки векторы $\tilde{h}_N(0)$ и $\tilde{h}_{N,o}$. На практике, однако, в большинстве случаев начальные значения ВК выбираются как $\tilde{h}_N(0)=[0, 0]^T$, так как нет никаких оснований выбирать любые иные начальные условия в силу того, что вектор $\tilde{h}_{N,o}$ является неизвестным.

Иключение, пожалуй, составляют эквалайзеры, у которых один ВК выбирается, как правило, не равным нулю, и ААР, у которых начальные значения ВК выбираются такими, чтобы максимум исходной ДН был направлен на источник полезного сигнала.

4.6. Другая интерпретация алгоритма Ньютона

Из разделов 4.2—4.5 следует, что алгоритм Ньютона, несмотря на свою большую вычислительную сложность из-за необходимости обращения корреляционной матрицы входных сигналов линейного сумматора, является более эффективным с функциональной точки зрения по сравнению с алгоритмом наискорейшего спуска. Эффективность демонстрируется тем, что допустимое значение шага сходимости (4.8), (4.9), постоянная времени переходного процесса ВК (4.16), (4.17) и постоянная времени обучающей кривой (4.20), (4.21) в алгоритме Ньютона не зависят от значений собственных чисел корреляционной матрицы входных сигналов линейного сумматора, в то время как в алгоритме наискорейшего спуска такая зависимость существует, что следует из уравнений (4.32), (4.33), (4.38), (4.39), (4.42) и (4.43).

Поэтому компоненты вектора ВК в алгоритме Ньютона сходятся к винеровскому решению с одинаковой скоростью, независимо от отношения собственных чисел корреляционной матрицы $\chi = \lambda_{\max} / \lambda_{\min}$, что подтверждается графиками на рис. 4.3. В алгоритме же наискорейшего спуска согласно уравнениям (4.38) и (4.39) при фиксированном значении шага сходимости переходные процессы компонент вектора ВК зависят от собственных чисел корреляционной матрицы (см. рис. 4.9, рис. 4.12 и рис. 4.15).

Объяснение указанных свойств алгоритма Ньютона приводится в настоящем разделе.

Согласно уравнению (3.46) справедливо соотношение $\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{\Lambda}_N$ которое можно интерпретировать как

$$\begin{aligned} \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N &= \mathbf{Q}_N^H E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} \mathbf{Q}_N = E\{\mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N\} = \\ &= E\{\mathbf{x}'_N(k) \mathbf{x}'_N^H(k)\} = \mathbf{\Lambda}_N, \end{aligned} \quad (4.44)$$

где $\mathbf{x}'_N(k) = \mathbf{Q}_N^H \mathbf{x}_N(k)$ — преобразованный вектор входных сигналов.

Матрица (4.44) является диагональной. Это означает, что элементы вектора $\mathbf{x}'_N(k)$ не коррелированы друг с другом. Другими словами, элементы вектора $\mathbf{x}'_N(k)$ являются взаимоортогональными случайными переменными. Кроме того, из уравнения (4.44) следует, что

$$E\{x'_n(k) x'^*_n(k)\} = \lambda_n \quad (4.45)$$

для всех $n = 1, 2, \dots, N$, где $x'_n(k)$ — элементы вектора $\mathbf{x}'_N(k)$.

Если элементы $x'_n(k)$ пронормировать как

$$x''_n(k) = \lambda_n^{-0.5} x'_n(k), \quad (4.46)$$

то вектор из таких элементов можно представить как

$$\mathbf{x}''_N(k) = \mathbf{\Lambda}_N^{-0.5} \mathbf{x}'_N(k). \quad (4.47)$$

Из уравнения (4.47) следует, что

$$\begin{aligned}\mathbf{R}_N'' &= E\{\mathbf{x}_N''(k)\mathbf{x}_N''^H(k)\} = E\{\Lambda_N^{-0.5}\mathbf{x}_N'(k)\mathbf{x}_N'^H(k)\Lambda_N^{-0.5}\} = \\ &= \Lambda_N^{-0.5}E\{\mathbf{x}_N'(k)\mathbf{x}_N'^H(k)\}\Lambda_N^{-0.5} = \Lambda_N^{-0.5}\Lambda\Lambda_N^{-0.5} = \mathbf{I}_N.\end{aligned}\quad (4.48)$$

Если определить вектор ВК линейного сумматора, нормированных как

$$\mathbf{h}_N'' = \Lambda_N^{0.5}\mathbf{Q}_N^H\mathbf{h}_N, \quad (4.49)$$

а в качестве его входных сигналов использовать элементы преобразованного и нормированного вектора $\mathbf{x}_N''(k)$, то выходной сигнал этого сумматора будет равным выходному сигналу линейного сумматора с ВК $\mathbf{h}_N(k)$ и входными сигналами $\mathbf{x}_N(k)$. Действительно,

$$\begin{aligned}\mathbf{h}_N''^H\mathbf{x}_N''(k) &= \left[\Lambda_N^{0.5}\mathbf{Q}_N^H\mathbf{h}_N\right]^H\Lambda_N^{-0.5}\mathbf{x}_N'(k) = \left[\mathbf{Q}_N^H\mathbf{h}_N\right]^H\Lambda_N^{0.5}\Lambda_N^{-0.5}\mathbf{Q}_N^H\mathbf{x}_N(k) = \\ &= \mathbf{h}_N^H\mathbf{Q}_N\mathbf{Q}_N^H\mathbf{x}_N(k) = \mathbf{h}_N^H\mathbf{x}_N(k),\end{aligned}\quad (4.50)$$

так как $\mathbf{Q}_N\mathbf{Q}_N^H = \mathbf{I}_N$, см. уравнения (3.44), (3.45).

Таким образом, из уравнения (4.50) следует, что оба фильтра являются эквивалентными друг другу, поскольку выходные сигналы таких фильтров равны между собой.

На основании этого вывода можно записать уравнение вычисления вектора ВК \mathbf{h}_N'' с помощью алгоритма наискорейшего спуска, аналогичное уравнению (4.23),

$$\begin{aligned}\mathbf{h}_N''(k) &= \mathbf{h}_N''(k-1) - \mu[-\mathbf{r}_N' + \mathbf{R}_N''\mathbf{h}_N''(k-1)] = \\ &= \mathbf{h}_N''(k-1) + \mu\mathbf{r}_N'' - \mu\mathbf{R}_N''\mathbf{h}_N''(k-1) = (\mathbf{I}_N - \mu\mathbf{R}_N'')\mathbf{h}_N''(k-1) + \mu\mathbf{r}'',\end{aligned}\quad (4.51)$$

где вектор \mathbf{r}'' определяется как

$$\mathbf{r}_N'' = E\{\mathbf{x}_N''(k)d^*(k)\} = E\left\{\begin{bmatrix} x_1''(k) \\ x_2''(k) \\ \vdots \\ x_N''(k) \end{bmatrix}d^*(k)\right\} = \begin{bmatrix} E\{x_1''(k)d^*(k)\} \\ E\{x_2''(k)d^*(k)\} \\ \vdots \\ E\{x_N''(k)d^*(k)\} \end{bmatrix}. \quad (4.52)$$

Поскольку $\mathbf{R}_N'' = \mathbf{I}_N$, см. выражение (4.48), то уравнение (4.51) можно представить как

$$\mathbf{h}_N''(k) = (\mathbf{I}_N - \mu\mathbf{I}_N)\mathbf{h}_N''(k-1) + \mu\mathbf{r}'' = (1-\mu)\mathbf{h}_N''(k-1) + \mu\mathbf{h}_{N,o}'', \quad (4.53)$$

так как $\mathbf{h}_{N,o}'' = (\mathbf{R}_N'')^{-1}\mathbf{r}'' = \mathbf{I}_N^{-1}\mathbf{r}''$.

Таким образом, алгоритм наискорейшего спуска (4.51) для ВК $\mathbf{h}_N''(k)$ эквивалентен уравнению (4.53), которое является алгоритмом Ньютона для ВК $\mathbf{h}_N''(k)$, аналогичным уравнению (4.3).

Можно показать, что уравнение (4.53) и уравнение алгоритма Ньютона (4.3) также являются эквивалентными. Для этого в уравнении (4.53) представим ВК с помощью уравнения (4.49) как $\Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N$, т. е.

$$\Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N(k) = (1-\mu) \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N(k-1) + \mu \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_{N,o}. \quad (4.54)$$

Далее, умножим обе части уравнения (4.54) слева на

$$(\Lambda_N^{0.5} \mathbf{Q}_N^H)^{-1} = (\mathbf{Q}_N^H)^{-1} \Lambda_N^{-0.5} = (\mathbf{Q}_N^{-1})^{-1} \Lambda_N^{-0.5} = \mathbf{Q}_N \Lambda_N^{-0.5}. \quad (4.55)$$

В уравнении (4.55) использованы соотношения (3.50) и (3.45). Тогда

$$\begin{aligned} & \mathbf{Q}_N \Lambda_N^{-0.5} \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N(k) = \\ & = \mathbf{Q}_N \Lambda_N^{-0.5} \left[(1-\mu) \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N(k-1) + \mu \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_{N,o} \right] = \\ & = (1-\mu) \mathbf{Q}_N \Lambda_N^{-0.5} \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_N(k-1) + \mu \mathbf{Q}_N \Lambda_N^{-0.5} \Lambda_N^{0.5} \mathbf{Q}_N^H \mathbf{h}_{N,o} \end{aligned} \quad (4.56)$$

или

$$\mathbf{h}_N(k) = (1-\mu) \mathbf{h}_N(k-1) + \mu \mathbf{h}_{N,o}, \quad (4.57)$$

так как $\Lambda_N^{-0.5} \Lambda_N^{0.5} = \mathbf{I}_N$ и $\mathbf{Q}_N \mathbf{Q}_N^H = \mathbf{I}_N$.

Таким образом, уравнение (4.51) эквивалентно уравнению (4.53), а уравнение (4.53) эквивалентно уравнению (4.54) и эквивалентно уравнению (4.57), т. е. алгоритму Ньютона (4.3).

Поэтому, с учетом (4.50), можно сделать вывод, что алгоритм Ньютона (4.3) для линейного сумматора с вектором ВК $\mathbf{h}_N(k)$ и вектором входных сигналов $\mathbf{x}_N(k)$ эквивалентен алгоритму наискорейшего спуска для линейного сумматора с вектором ВК $\mathbf{h}_N''(k)$ и вектором входных сигналов $\mathbf{x}_N''(k)$. Поскольку сходимость процедуры вычисления ВК в алгоритме Ньютона не зависит от собственных чисел корреляционной матрицы входных сигналов линейного сумматора, то такие же свойства с точки зрения сходимости демонстрирует и алгоритм наискорейшего спуска для эквивалентного линейного сумматора с вектором ВК $\mathbf{h}_N''(k)$ и вектором входных сигналов $\mathbf{x}_N''(k)$.

Это обусловлено тем обстоятельством, что преобразования (4.44) и (4.47) в соответствии с (4.48) обеспечивают $\mathbf{R}_N'' = \mathbf{I}_N$, т. е. корреляционная матрица сигналов $\mathbf{x}_N''(k)$ является единичной, а значит, согласно выражению (3.79) собственные числа λ_n'' матрицы \mathbf{R}_N'' равны между собой и равны единице, что согласно уравнениям (4.38) и (4.39) обеспечивает одинаковые условия сходимости всех компонент вектора $\mathbf{h}_N''(k)$ к вектору $\mathbf{h}_{N,o}''$.

Таким образом, зависимость сходимости алгоритма наискорейшего спуска (4.23) от значений собственных чисел корреляционной матрицы \mathbf{R}_N исчезает в алгоритме (4.53) за счет декорреляции элементов векторов $\mathbf{x}_N(k)$ с помощью преобразования $\mathbf{x}_N'(k) = \mathbf{Q}_N^H \mathbf{x}_N(k)$, известного как преобразование Карунена — Лоева, и нормализации $\mathbf{x}_N''(k) = \Lambda_N^{-0.5} \mathbf{x}_N'(k)$.

Это свойство объясняет лучшую сходимость градиентных алгоритмов, полученных в результате упрощения алгоритма Ньютона, по сравнению с градиентными алгоритмами, полученными в результате упрощения алгоритма наискорейшего спуска.

4.7. Выводы по главе

В главе 4 рассмотрены алгоритм Ньютона и алгоритм наискорейшего спуска для поиска ВК адаптивного линейного сумматора, соответствующих минимуму MSE-функции, т. е. винеровскому решению. Характеристики этих алгоритмов будут использоваться для сравнения с характеристиками рассматриваемых в главе 5 других градиентных алгоритмов.

ГЛАВА 5

LMS-АЛГОРИТМ И ЕГО СВОЙСТВА

В настоящей главе рассматривается LMS-алгоритм адаптивной фильтрации. Анализируются свойства этого алгоритма и приводятся результаты его численного моделирования. Также рассматриваются нормализованная версия LMS-алгоритма (NLMS-алгоритм), LMS- и NLMS-алгоритмы с переменным шагом сходимости, изменяемым по градиентному закону, и линейно-ограниченные версии LMS- и NLMS-алгоритмов.

5.1. Введение

LMS-алгоритм адаптивной фильтрации [46, 56, 57, 73, 74] принадлежит к классу алгоритмов стохастического градиентного поиска. Авторами этого алгоритма являются Уидроу (B. Widrow) и Хофф (M. E. Hoff). Термин «стохастический» применяется для того, чтобы отличать LMS-алгоритм, использующий упрощенный градиент, от алгоритма наискорейшего спуска, который использует точное значение градиента целевой MSE-функции. Упрощение градиента является следствием упрощения целевой функции. Поскольку упрощенный градиент неточный, то он подвержен случайным флуктуациям. Поэтому движение вектора ВК к оптимальному решению носит в некотором смысле случайный характер, отчего и применяется термин «стохастический», т. е. случайный. Тем не менее, при обработке стационарных сигналов и соответствующем выборе шага сходимости LMS-алгоритм сходится к винеровскому решению.

Сегодня LMS-алгоритм является наиболее широко используемым на практике благодаря тому, что его вычислительная сложность наименьшая среди всех известных алгоритмов адаптивной фильтрации.

5.2. LMS-алгоритм

В главе 4 были рассмотрены алгоритмы поиска оптимального винеровского решения (3.63) $\mathbf{h}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N$, соответствующего минимуму целевой MSE-функции (см. рис. 3.3) линейного сумматора (см. рис. 3.2). Это решение соответствует минимальной среднеквадратической ошибке оценивания требуемого сигнала $d(k)$

в предположении, что входные сигналы $x_n(k)$ и сигнал $d(k)$ линейного сумматора стационарны, а матрица \mathbf{R}_N и вектор \mathbf{r}_N определяются как $\mathbf{R}_N = E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\}$, см. уравнение (3.5), и $\mathbf{r}_N = E\{\mathbf{x}_N(k)d^*(k)\}$, см. уравнение (3.59).

Если на k -м временном отсчете имеются оценки матрицы \mathbf{R}_N и вектора \mathbf{r}_N , обозначенные как $\hat{\mathbf{R}}_N(k)$ и $\hat{\mathbf{r}}_N(k)$, то для поиска винеровского решения можно использовать алгоритм наискорейшего спуска (4.22), (4.23), в котором используются эти оценки:

$$\begin{aligned}\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) - \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = \mathbf{h}_N(k-1) - \mu \left[-\hat{\mathbf{r}}_N(k) + \hat{\mathbf{R}}_N(k) \mathbf{h}_N(k-1) \right] = \\ &= \mathbf{h}_N(k-1) + \mu \hat{\mathbf{r}}_N(k) - \mu \hat{\mathbf{R}}_N(k) \mathbf{h}_N(k-1),\end{aligned}\quad (5.1)$$

где $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -\hat{\mathbf{r}}_N(k) + \hat{\mathbf{R}}_N(k) \mathbf{h}_N(k-1)$ — оценка вектора градиента целевой функции.

Если производить оценку матрицы \mathbf{R}_N и вектора \mathbf{r}_N на окне всего в один отсчет обрабатываемых сигналов, т. е. как $\hat{\mathbf{R}}_N(k) = \mathbf{x}_N(k)\mathbf{x}_N^H(k)$ и $\hat{\mathbf{r}}_N = \mathbf{x}_N(k)d^*(k)$, то градиент $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1)$ можно выразить как

$$\begin{aligned}\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) &= -\mathbf{x}_N(k)d^*(k) + \mathbf{x}_N(k)\mathbf{x}_N^H(k)\mathbf{h}_N(k-1) = \\ &= -\mathbf{x}_N(k) \left[d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k-1) \right] = -\mathbf{x}_N(k) \left[d^*(k) - y^*(k) \right] = -\mathbf{x}_N(k)\alpha^*(k),\end{aligned}\quad (5.2)$$

где $\alpha(k)$ — априорная ошибка моделирования (оценивания) сигнала $d(k)$, см. уравнение (1.2).

Выражение (5.2) можно получить и другим способом. Для этого MSE-функцию следует заменить квадратом модуля мгновенной ошибки $F = f\{\alpha(k)\} = |\alpha(k)|^2 = \alpha(k)\alpha^*(k)$. Тогда

$$\begin{aligned}\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) &= \hat{\nabla}_{\mathbf{h}_N^*} \{\alpha(k)\alpha^*(k)\} = \frac{\partial [\alpha(k)\alpha^*(k)]}{\partial \mathbf{h}_N^*(k-1)} = \frac{\partial \alpha(k)}{\partial \mathbf{h}_N^*(k-1)} \alpha^*(k) + \alpha(k) \frac{\partial \alpha^*(k)}{\partial \mathbf{h}_N^*(k-1)} = \\ &= \frac{\partial [d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)]}{\partial \mathbf{h}_N^*(k-1)} \alpha^*(k) = -\mathbf{x}_N(k)\alpha^*(k),\end{aligned}\quad (5.3)$$

так как

$$\alpha(k) \frac{\partial \alpha^*(k)}{\partial \mathbf{h}_N^*(k-1)} = \alpha(k) \frac{\partial [d^*(k) - y^*(k)]}{\partial \mathbf{h}_N^*(k-1)} = \alpha(k) \frac{\partial [d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k-1)]}{\partial \mathbf{h}_N^*(k-1)} = \mathbf{0}_N.$$

В случае действительных ВК градиент определяется как

$$\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -2\mathbf{x}_N(k)\alpha(k). \quad (5.4)$$

Таким образом, используя (5.2) или (5.3), уравнение (5.1) можно представить как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) - \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k)\alpha^*(k). \quad (5.5)$$

Уравнение (5.5) — это LMS-алгоритм поиска ВК адаптивного фильтра. Название Least Mean Square происходит от того, что алгоритм минимизирует квадрат мгновенной ошибки, т. е. ошибки, определяемой по одному отсчету обрабатываемых сигналов.

Поскольку в уравнении (5.5) используются неточные оценки градиента, то в адаптивном процессе возникает шум, в результате чего траектория движения вектора $\mathbf{h}_N(k)$ к вектору $\mathbf{h}_{N,o}$ не совпадает с траекторией движения в алгоритме наискорейшего спуска (4.23). Этот шум уменьшается по мере движения к стационарной точке $\mathbf{h}_{N,o}$, так как приращения вектора ВК, равные $-\mu \hat{\nabla}_{\mathbf{h}_N} \xi(k-1)$, вблизи точки $\mathbf{h}_{N,o}$ уменьшаются по сравнению с приращениями в начальной точке $\mathbf{h}_N(0) \neq \mathbf{h}_{N,o}$. Это происходит из-за того, что значения градиента MSE-функции вблизи стационарной точки $\mathbf{h}_{N,o}$ стремятся к нулю, т. е. $\nabla_{\mathbf{h}_N} \xi(k-1) \rightarrow 0$, а значит, и оценки упрощенного градиента также стремятся к нулю, т. е. $\hat{\nabla}_{\mathbf{h}_N} \xi(k-1) \rightarrow 0$.

Алгоритм (5.5) привлекателен своей простотой, так как для вычисления градиента не требуется получать оценки MSE-функции (т. е. возводить ошибки в квадрат и усреднять) и вычислять по этим оценкам производные (например по приращениям), для того чтобы сформировать вектор градиента $\nabla_{\mathbf{h}_N} \xi(k-1)$.

Вычислительная процедура LMS-алгоритма приведена в табл. 5.1.

Таблица 5.1. LMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	(5.1.0)
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(5.1.1)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k)$	(5.1.2)
End for k	

Инициализация LMS-алгоритма, как правило, осуществляется в соответствии с уравнением (5.1.0), если нет каких-либо априорных сведений о векторе $\mathbf{h}_{N,o}$, которые при $\mathbf{h}_N(0) \neq \mathbf{0}_N$ могут способствовать уменьшению длительности переходного процесса, так как в зависимости от $\mathbf{h}_N(0)$ обычно требуется разное число шагов адаптивного алгоритма для преодоления расстояния $|\mathbf{h}_N(0) - \mathbf{h}_{N,o}|$ между точками многомерного пространства $\mathbf{h}_N(0)$ и $\mathbf{h}_{N,o}$.

Как уже отмечалось, LMS-алгоритм является самым простым среди алгоритмов адаптивной фильтрации с точки зрения вычислительной и алгоритмической сложности. При оценке вычислительной сложности алгоритмов ЦОС число операций умножения и сложения, как правило, подсчитывается раздельно. Это позволяет характеризовать сложность алгоритмов независимо от вычислительной платформы, реализующей эти алгоритмы. В оценках вычислительной сложности алгоритмов обычно учитываются только векторные операции

сложения и умножения, т.е. операции, число которых пропорционально размерности используемых векторов (в данном случае N).

Более сложные операции (деления, извлечения квадратных корней, если такие присутствуют в рассматриваемом алгоритме), число которых часто не пропорционально N , в оценках учитывается точно, так как в современных вычислительных платформах, как правило, не существует аппаратной поддержки таких операций. Эти операции реализуются с помощью различных функциональных разложений, и их вычислительная сложность в терминах простых арифметических операций умножения и сложения определяется используемым алгоритмом разложения. В ряде случаев число простых арифметических операций в одной сложной операции может быть соизмеримым с N . Это обуславливает необходимость точного подсчета числа сложных арифметических операций при оценке сложности адаптивных алгоритмов.

Оценка вычислительной сложности алгоритма (см. табл. 5.1), равная $2N$ умножений и $2N$ сложений на одну итерацию, определяется следующим образом.

Уравнение (5.1.1) при вычислении скалярного произведения векторов $\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$ требует N умножений и N сложений на одну итерацию алгоритма. Если быть более точным, то число сложений равно $N - 1$. Однако при программной реализации операция скалярного произведения обычно выполняется как накапливающая сумма, т.е. $\mathbf{h}_N^H(k-1)\mathbf{x}_N(k) = \sum_{n=1}^N h_n^*(k-1)x(k-n+1)$, см. выражение (1.1).

Начальное значение этой суммы задается равным нулю. Поэтому выражение

$$\begin{aligned} \sum_{n=1}^N h_n^*(k-1)x(k-n+1) &= \\ &= \underbrace{0 + h_1^*(k-1)x(k) + h_2^*(k-1)x(k-1) + \dots + h_N^*(k-1)x(k-N+1)}_{N \text{ умножений, } N \text{ сложений}} \end{aligned} \quad (5.6)$$

содержит N умножений и N сложений. В случае аппаратной реализации уравнения (5.6), как, например, показано на рис. 1.2, «лишняя» операция сложения с нулем обычно не выполняется, так как ее выполнение требует наличия дополнительного сумматора. Но поскольку адаптивные фильтры используются в приложениях, где, как правило, число ВК $N \gg 1$, то оценка числа сложений в операции $\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$, равная N , также является правомочной.

Уравнение (5.1.2) при выполнении операции умножения вектора на скаляр с накоплением

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) = \mathbf{h}_N(k-1) + \gamma(k) \mathbf{x}_N(k) = \\ &= \begin{bmatrix} h_1(k-1) \\ h_2(k-1) \\ \vdots \\ h_N(k-1) \end{bmatrix} + \gamma(k) \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} = \begin{bmatrix} h_1(k-1) + \gamma(k)x_1(k) \\ h_2(k-1) + \gamma(k)x_2(k) \\ \vdots \\ h_N(k-1) + \gamma(k)x_N(k) \end{bmatrix}, \end{aligned} \quad (5.7)$$

где

$$\gamma(k) = \mu \alpha^*(k), \quad (5.8)$$

также требует N умножений и N сложений на одну итерацию, т. е. по одному умножению и одному сложению при вычислении нового значения каждого из N ВК, образующих вектор $\mathbf{h}_N(k)$.

В дальнейшем аналогичным образом будет оцениваться вычислительная сложность и других алгоритмов адаптивной фильтрации, содержащих операции скалярного произведения двух векторов и операции векторного умножения с накоплением. Отдельные операции сложения и умножения, число которых не пропорционально числу присутствующих в алгоритмах векторов и матриц, определяющих размерность задачи, при оценке вычислительной сложности учитываться не будут. Пример такой операции — это операция умножения (5.8) в LMS-алгоритме. Число таких операций, как правило, много меньше N , а потому на оценку вычислительной сложности алгоритмов они не влияют.

5.3. Переходные процессы в LMS-алгоритме

Основной характеристикой всех алгоритмов адаптивной фильтрации является сходимость вектора ВК к оптимальному значению $\mathbf{h}_{N,o}$, при котором MSE-функция достигает своего минимума. ВК LMS-алгоритма, несмотря на его простоту, также сходятся к $\mathbf{h}_{N,o}$. Это доказывается следующим образом.

Для начала покажем, что в LMS-алгоритме при постоянном значении вектора ВК оценка градиента $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1)$ является несмешенной.

Действительно, согласно выражению (3.60) точное значение градиента MSE-функции определяется как $\nabla_{\mathbf{h}_N^*} \xi = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N$. Согласно же (5.2) оценка градиента определяется как $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -\mathbf{x}_N(k) d^*(k) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{h}_N(k-1)$. Поскольку градиент и его оценка равны друг другу лишь приблизительно, т. е.

$$\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) \approx \nabla_{\mathbf{h}_N^*} \xi, \quad (5.9)$$

то очевидно, что на каждой итерации направления движения вектора ВК, определяемые градиентами (3.60) и (5.2), отличаются друг от друга. Следовательно, на каждой итерации сходимость LMS-алгоритма не может быть точно такой же, как сходимость алгоритма наискорейшего спуска. В то же время в среднем оценка градиента LMS-алгоритма (5.2) равна точному значению градиента (3.60), так как при фиксированном значении вектора ВК

$$\begin{aligned} E\left\{\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1)\right\} &= E\left\{-\mathbf{x}_N(k) d^*(k) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{h}_N\right\} = \\ &= E\left\{-\mathbf{x}_N(k) d^*(k)\right\} + E\left\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\right\} \mathbf{h}_N = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N = \nabla_{\mathbf{h}_N^*} \xi, \end{aligned} \quad (5.10)$$

т. е. среднее значение оценки градиента (5.2) согласно уравнению (5.10) является несмещенным. Это свойство используется в так называемых блочных градиентных алгоритмах, где градиент (5.3) оценивается на нескольких отчетах K_E при фиксированном значении \mathbf{h}_N и усредняется как

$$\bar{\nabla}_{\mathbf{h}_N} \xi(K_E - 1) = E\left\{\hat{\nabla}_{\mathbf{h}_N} \xi(k-1)\right\} = -\frac{1}{K_E} \sum_{k_E=0}^{K_E-1} \mathbf{x}_N(k+k_E) \alpha^*(k+k_E), \quad (5.11)$$

а вычисление вектора ВК осуществляется не на каждом k -м отсчете, а через каждые K_E отсчетов обрабатываемых сигналов

$$\mathbf{h}_N(K_E) = \mathbf{h}_N(K_E - 1) - \mu \bar{\nabla}_{\mathbf{h}_N} \xi(K_E - 1). \quad (5.12)$$

Выходной сигнал адаптивного фильтра формируется на каждом k -м отсчете как $y(k) = \mathbf{h}_N^H(K_E - 1) \mathbf{x}_N(k)$, для чего в течение каждого блока из K_E отсчетов используются текущие значения векторов сигналов $\mathbf{x}_N(k)$ и вектора ВК $\mathbf{h}_N(K_E - 1)$, вычисляемого один раз за K_E предыдущих отсчетов.

Согласно (4.23) уравнение вычисления вектора ВК с помощью алгоритма наискорейшего спуска можно представить как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \mathbf{r}_N - \mu \mathbf{R}_N \mathbf{h}_N(k-1) = (\mathbf{I}_N - \mu \mathbf{R}_N) \mathbf{h}_N(k-1) + \mu \mathbf{r}_N. \quad (5.13)$$

Можно показать, что среднее значение вектора ВК (5.5) равно вектору (5.13). Действительно, используя выражение (5.1), получим, что

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mu \hat{\mathbf{r}}_N(k) - \mu \hat{\mathbf{R}}_N(k) \mathbf{h}_N(k-1) = \\ &= \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) d^*(k) - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{h}_N(k-1). \end{aligned} \quad (5.14)$$

Среднее значение вектора ВК (5.14) определяется как

$$\begin{aligned} E\{\mathbf{h}_N(k)\} &= E\{\mathbf{h}_N(k-1)\} + \mu E\{\mathbf{x}_N(k) d^*(k)\} - \\ &\quad - \mu E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} E\{\mathbf{h}_N(k-1)\} = \\ &= E\{\mathbf{h}_N(k-1)\} + \mu \mathbf{r}_N - \mu \mathbf{R}_N E\{\mathbf{h}_N(k-1)\} = \\ &= (\mathbf{I}_N - \mu \mathbf{R}_N) E\{\mathbf{h}_N(k-1)\} + \mu \mathbf{r}_N. \end{aligned} \quad (5.15)$$

Таким образом, средние значения вектора ВК (5.15) в LMS-алгоритме (5.5) совпадают со значениями вектора ВК (4.23) в алгоритме наискорейшего спуска. В уравнении (5.15) использованы положения теории независимости [74], согласно которой $\mathbf{x}_N(k)$ и $\mathbf{h}_N(k)$ предполагаются статистически независимыми.

Средние значения вектора ВК в LMS-алгоритме в смещенной системе координат также совпадают со значениями значениями вектора ВК в смещенной системе координат (4.25) в алгоритме наискорейшего спуска. Действительно, согласно уравнению (3.67) требуемый сигнал может быть представлен как $d(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k)$. Используя это уравнение и уравнение (5.5), можно определить вектор ВК $\Delta \mathbf{h}_N(k)$ в смещенной системе координат как

$$\begin{aligned}
\Delta \mathbf{h}_N(k) &= \mathbf{h}_N(k) - \mathbf{h}_{N,o} = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) - \mathbf{h}_{N,o} = \\
&= \mathbf{h}_N(k-1) - \mathbf{h}_{N,o} + \mu \mathbf{x}_N(k) \left[d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \right]^* = \\
&= \mathbf{h}_N(k-1) - \mathbf{h}_{N,o} + \mu \mathbf{x}_N(k) \left[e_o(k) + \mathbf{h}_{N,o}^H \mathbf{x}_N(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \right]^* = \\
&= \Delta \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) e_o^*(k) - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1) = \\
&= \left[\mathbf{I}_N - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right] \Delta \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) e_o^*(k),
\end{aligned} \tag{5.16}$$

а также определить среднее значение вектора ВК $E\{\Delta \mathbf{h}_N(k)\}$ в смещенной системе координат как

$$E\{\Delta \mathbf{h}_N(k)\} = E\left\{ \left[\mathbf{I}_N - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right] \Delta \mathbf{h}_N(k-1) \right\} + E\{\mu \mathbf{x}_N(k) e_o^*(k)\}. \tag{5.17}$$

Предполагая, что $\mathbf{x}_N(k)$, $\Delta \mathbf{h}_N(k-1)$ и $e_o(k)$ являются статистически независимыми, получим, что

$$\begin{aligned}
E\{\Delta \mathbf{h}_N(k)\} &= \left[\mathbf{I}_N - \mu E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} \right] E\{\Delta \mathbf{h}_N(k-1)\} + \\
&+ \mu E\{\mathbf{x}_N(k) e_o^*(k)\} = \left[\mathbf{I}_N - \mu \mathbf{R}_N(k) \right] E\{\Delta \mathbf{h}_N(k-1)\},
\end{aligned} \tag{5.18}$$

так как согласно принципу ортогональности (3.68) значение $E\{\mathbf{x}_N(k) e_o^*(k)\} = \mathbf{0}_N$.

Таким образом, среднее значение вектора ВК $E\{\Delta \mathbf{h}_N(k)\}$ LMS-алгоритма в смещенной системе координат совпадает со значением ВК $\Delta \mathbf{h}_N(k)$ алгоритма наискорейшего спуска (4.25), которые согласно анализу уравнений (4.26)–(4.33) сходятся к винеровскому решению $\mathbf{h}_{N,o}$ при $0 < \mu < 2/\lambda_{\max}$ в случае комплексных ВК и при $0 < \mu < 1/\lambda_{\max}$ в случае действительных ВК.

Выполнив такой же анализ для вектора $E\{\Delta \mathbf{h}_N(k)\}$, можно получить аналогичное условие сходимости в среднем ВК LMS-алгоритма к винеровскому решению. То есть, в среднем эти ВК сходятся к винеровскому решению, если выполняется условие

$$0 < \mu < 2/\lambda_{\max} \tag{5.19}$$

в случае комплексных ВК или выполняется условие

$$0 < \mu < 1/\lambda_{\max} \tag{5.20}$$

в случае действительных ВК.

На практике при использовании адаптивного фильтра определять собственные числа корреляционной матрицы обычно не представляется возможным. Однако, используя (3.53) (свойство 6 собственных чисел корреляционной матрицы) согласно которому $\text{tr}\{\mathbf{R}_N\} = \sum_{n=1}^N R_{nn} = \sum_{n=1}^N \lambda_n$ можно заключить, что $\lambda_{\max} \leq \text{tr}\{\mathbf{R}_N\} = \sum_{n=1}^N R_{nn}$. С учетом этого соотношения условие (5.19) можно представить в виде несколько более жесткого условия устойчивости LMS-алгоритма

$$0 < \mu < \frac{2}{\sum_{n=1}^N R_{nn}}, \quad (5.21)$$

так как

$$\frac{2}{\sum_{n=1}^N R_{nn}} \leq \frac{2}{\lambda_{\max}}. \quad (5.22)$$

В общем случае диагональные элементы корреляционной матрицы определяются как $R_{nn} = \sigma_{x_n}^2$, где $\sigma_{x_n}^2$ — дисперсия n -го входного сигнала многоканального линейного сумматора. В случае трансверсального адаптивного фильтра все значения $\sigma_{x_n}^2$ равны между собой и равны σ_x^2 . Поэтому условие (5.21) можно сформулировать как

$$0 < \mu < \frac{2}{N\sigma_x^2}. \quad (5.23)$$

Если ВК являются действительными, то уравнения (5.21) и (5.23) записываются как

$$0 < \mu < \frac{1}{\sum_{n=1}^N R_{nn}} \quad (5.24)$$

и

$$0 < \mu < \frac{1}{N\sigma_x^2}. \quad (5.25)$$

Используя преобразования, аналогичные (4.36) и (4.37), можно показать (см. раздел 4.5), что постоянная времени $k_{\tau_{n,h}}$ среднего значения комплексных ВК LMS-алгоритма находится в пределах

$$\frac{1}{\mu\lambda_{\max}} \leq k_{\tau_{n,h}} \leq \frac{1}{\mu\lambda_{\min}}, \quad (5.26)$$

а постоянная времени среднего значения действительных ВК — в пределах

$$\frac{1}{2\mu\lambda_{\max}} \leq k_{\tau_{n,h}} \leq \frac{1}{2\mu\lambda_{\min}}. \quad (5.27)$$

Таким образом, постоянные времени переходных процессов среднего значения ВК в LMS-алгоритме совпадают с соответствующими постоянными временем алгоритма наискорейшего спуска. Другими словами, графики переходных процессов усредненных значений ВК в LMS-алгоритме совпадают с одноименными графиками алгоритма наискорейшего спуска.

На рис. 5.1 и рис. 5.2 показаны результаты моделирования алгоритма наискорейшего спуска и LMS-алгоритма при решении задачи линейного предсказания (см. рис. 3.10) авторегрессионного процесса при $\chi = 5$. Темной линией показан переходный процесс в терминах действительных ВК алгоритма наи-

скорейшего спуска, а светлой — LMS-алгоритма при двух значениях параметра μ , удовлетворяющих условиям (5.20) и (5.25).

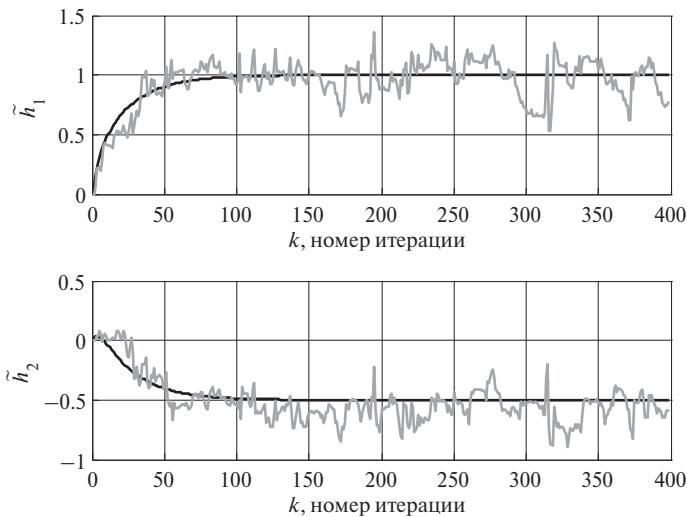


Рис. 5.1. Переходные процессы ВК при $\mu = 0.025$: одна реализация

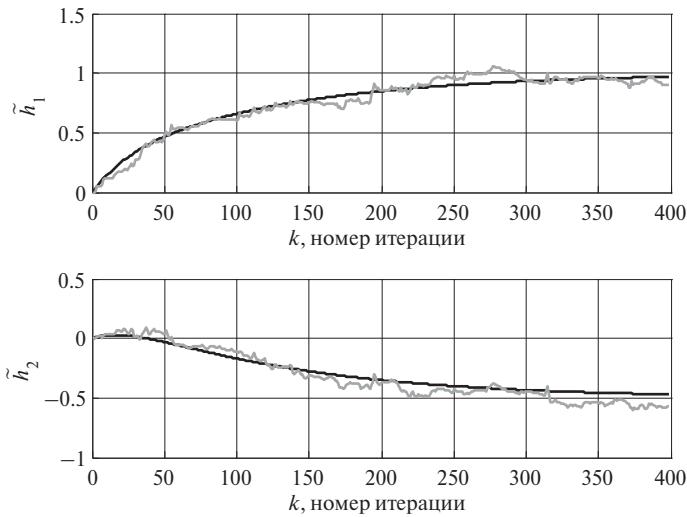


Рис. 5.2. Переходные процессы ВК при $\mu = 0.005$: одна реализация

Из рис. 5.1 и рис. 5.2 следует, что переходный процесс LMS-алгоритма примерно совпадает с переходным процессом алгоритма наискорейшего спуска, так как в LMS-алгоритме используется лишь приближенное значение градиента. При уменьшении значения μ длительность переходного процесса

LMS-алгоритма увеличивается, но уменьшаются его флуктуации. Таким образом, значение параметра μ может выбираться, исходя из компромисса между длительностью переходного процесса и точностью настройки адаптивного фильтра.

Результат усреднения переходных процессов (см. рис. 5.2) по 100 реализациям показан на рис. 5.3. Из рис. 5.3 следует, что действительно среднее значение ВК LMS-алгоритма (5.15) приближается к значениям ВК алгоритма наискорейшего спуска (4.23), т. е. в одинаковых условиях поведение LMS-алгоритма в терминах ВК адаптивного фильтра в среднем приближается к поведению алгоритма наискорейшего спуска.

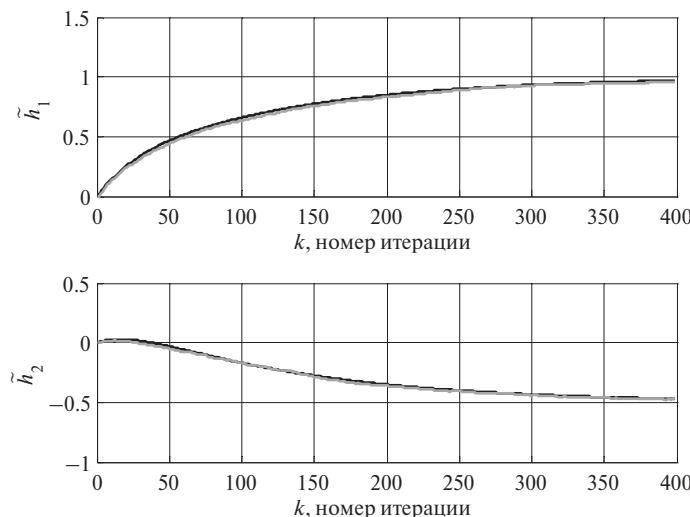


Рис. 5.3. Переходные процессы ВК при $\mu = 0.005$: усреднение по 100 реализациям

ВК, усредненные по ансамблю L реализаций, определяются как

$$E\{\mathbf{h}_N(k)\} = \frac{1}{L} \sum_{l=1}^L \mathbf{h}_{N,l}(k), \quad (5.28)$$

где $\mathbf{h}_{N,l}(k)$ — l -я реализация вектора $\mathbf{h}_N(k)$, полученная при l -й реализации входного процесса $x(k)$.

Определение постоянной времени переходного процесса в терминах среднеквадратической ошибки, т. е. обучающей кривой, осуществляется следующим образом. Для этого, используя уравнение (3.67) $d(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k)$, выразим ошибку $\alpha(k)$ как

$$\begin{aligned} \alpha(k) &= d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \\ &= e_o(k) + [\mathbf{h}_{N,o}^H - \mathbf{h}_N^H(k-1)] \mathbf{x}_N(k) = e_o(k) - \Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k), \end{aligned} \quad (5.29)$$

а ее квадрат модуля $|\alpha(k)|^2$ — как

$$|\alpha(k)|^2 = \alpha(k)\alpha^*(k) = e_o(k)e_o^*(k) - e_o(k)\mathbf{x}_N^H(k)\Delta\mathbf{h}_N(k-1) - \Delta\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)e_o^*(k) + \Delta\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)\mathbf{x}_N^H(k)\Delta\mathbf{h}_N(k-1). \quad (5.30)$$

Полагая, что $\mathbf{x}_N(k)$ и $\Delta\mathbf{h}_N(k)$ являются статистически независимыми, определим среднеквадратическую ошибку моделирования требуемого сигнала в LMS-алгоритме, т. е. обучающую кривую, как

$$\begin{aligned} \xi(k) &= E\{|\alpha(k)|^2\} = E\{e_o(k)^2\} - E\{e_o(k)\mathbf{x}_N^H(k)\}E\{\Delta\mathbf{h}_N(k-1)\} - \\ &- E\{\Delta\mathbf{h}_N^H(k-1)\}E\{\mathbf{x}_N(k)e_o^*(k)\} + E\{\Delta\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)\mathbf{x}_N^H(k)\Delta\mathbf{h}_N(k-1)\} = \\ &= \xi_{\min} + E\{\Delta\mathbf{h}_N^H(k-1)\}E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\}E\{\Delta\mathbf{h}_N(k-1)\} = \\ &= \xi_{\min} + E\{\Delta\mathbf{h}_N^H(k-1)\}\mathbf{R}_N E\{\Delta\mathbf{h}_N(k-1)\}, \end{aligned} \quad (5.31)$$

где

$$\xi_{\min} = E\{e_o(k)^2\} = \sigma_o^2 \quad (5.32)$$

— дисперсия (средняя мощность) процесса ошибок $e_o(k)$. В уравнении (5.31) учтено, что согласно выражению (3.68) $E\{\mathbf{x}_N(k)e_o^*(k)\} = \mathbf{0}_N$, а потому

$$E\{e_o(k)\mathbf{x}_N^H(k)\}E\{\Delta\mathbf{h}_N(k-1)\} + E\{\Delta\mathbf{h}_N^H(k-1)\}E\{\mathbf{x}_N(k)e_o^*(k)\} = 0. \quad (5.33)$$

Кроме того, учтено что согласно выражению (3.5) $\mathbf{R}_N = E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\}$.

Выражение для среднеквадратической ошибки $\xi(k)$ (5.31) совпадает с выражением (3.72) для MSE-функции с той лишь разницей, что вместо $\Delta\mathbf{h}_N$ используются значения $E\{\Delta\mathbf{h}_N(k-1)\}$. Поэтому, выполнив преобразования, аналогичные (4.40)–(4.43), получим следующие значения постоянных времени обучающих кривых LMS-алгоритма. В случае комплексных ВК

$$\frac{1}{2\mu\lambda_{\max}} \leq k_{\tau_{n,\xi}} \leq \frac{1}{2\mu\lambda_{\min}} \quad (5.34)$$

и в случае действительных ВК

$$\frac{1}{4\mu\lambda_{\max}} \leq k_{\tau_{n,\xi}} \leq \frac{1}{4\mu\lambda_{\min}}. \quad (5.35)$$

Таким образом, в среднем ВК LMS-алгоритма сходятся к винеровскому решению $\mathbf{h}_{N,o}$. Однако, даже находясь вблизи $\mathbf{h}_{N,o}$, ВК на каждой итерации отличаются от этого решения как

$$\Delta\mathbf{h}_N(k) = \mathbf{h}_N(k) - \mathbf{h}_{N,o} \neq \mathbf{0}_N, \quad (5.36)$$

что обусловлено грубой оценкой вектора градиента (5.2), содержащей шумовую составляющую. Это, в свою очередь, обуславливает наличие в установившемся состоянии дополнительной среднеквадратической ошибки ξ_{exc} , т. е.

$$\xi(\infty) = \xi_{\min} + \xi_{\text{exc}}. \quad (5.37)$$

5.4. Качество адаптивной фильтрации сигналов с помощью LMS-алгоритма

Одним из параметров, используемых для оценки качества адаптивной фильтрации сигналов с помощью LMS-алгоритма, является относительное значение дополнительной среднеквадратической ошибки (misadjustment)

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}}. \quad (5.38)$$

Для получения уравнения (5.38) используются приемы, представленные в [18, 25, 75, 76]. Результаты использования этих приемов несколько отличаются для адаптивных фильтров с комплексными и действительными ВК. Так как этот анализ является довольно громоздким, то представляется целесообразным рассмотреть его одновременно для адаптивных фильтров с ВК обоих типов.

В этом анализе используется корреляционная матрица ВК адаптивного фильтра, выраженных в терминах смещенной и повернутой системы координат (см. рис. 3.5). Для получения аналитического выражения этой матрицы используем следующие приемы.

Согласно уравнению (5.16) вектор комплексных ВК адаптивного фильтра на базе LMS-алгоритма в системе смещенных координат вектор определяется как

$$\Delta \mathbf{h}_N(k) = [\mathbf{I}_N - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k)] \Delta \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) e_o^*(k), \quad (5.39)$$

а вектор действительных ВК — как

$$\Delta \mathbf{h}_N(k) = [\mathbf{I}_N - 2\mu \mathbf{x}_N(k) \mathbf{x}_N^T(k)] \Delta \mathbf{h}_N(k-1) + 2\mu \mathbf{x}_N(k) e_o(k). \quad (5.40)$$

Чтобы перейти от вектора ВК (5.39) к вектору ВК в системе повернутых и смещенных координат (3.74) $\mathbf{v}_N(k) = \mathbf{Q}_N^H \Delta \mathbf{h}_N(k)$, умножим уравнение (5.39) на матрицу \mathbf{Q}_N^H слева как

$$\begin{aligned} \mathbf{Q}_N^H \Delta \mathbf{h}_N(k) &= \mathbf{Q}_N^H \{ [\mathbf{I}_N - \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k)] \Delta \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) e_o^*(k) \} = \\ &= \mathbf{Q}_N^H \mathbf{I}_N \Delta \mathbf{h}_N(k-1) - \mathbf{Q}_N^H \mu \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1) + \mathbf{Q}_N^H \mu \mathbf{x}_N(k) e_o^*(k) = \\ &= \mathbf{Q}_N^H \Delta \mathbf{h}_N(k-1) - \mu \mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N^H \Delta \mathbf{h}_N(k-1) + \mu \mathbf{Q}_N^H \mathbf{x}_N(k) e_o^*(k). \end{aligned} \quad (5.41)$$

В уравнении (5.41) использовано дополнительное умножение на $\mathbf{Q}_N \mathbf{Q}_N^H = \mathbf{I}_N$, не меняющее результат. Используя вектор трансформированных входных сигналов

$$\mathbf{x}'_N(k) = \mathbf{Q}_N^H \mathbf{x}_N(k), \quad (5.42)$$

уравнение (5.41) можно преобразовать к виду

$$\begin{aligned}\mathbf{v}_N(k) &= \mathbf{v}_N(k-1) - \mu \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) \mathbf{v}_N(k-1) + \mu \mathbf{x}'_N(k) e_o^*(k) = \\ &= \left[\mathbf{I}_N - \mu \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) \right] + \mu \mathbf{x}'_N(k) e_o^*(k).\end{aligned}\quad (5.43)$$

Аналогично вектор действительных ВК LMS-алгоритма в смещенной и повернутой системе координат можно преобразовать к виду

$$\begin{aligned}\mathbf{v}_N(k) &= \mathbf{v}_N(k-1) - 2\mu \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) \mathbf{v}_N(k-1) + 2\mu \mathbf{x}'_N(k) e_o(k) = \\ &= \left[\mathbf{I}_N - 2\mu \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) \right] + \mu 2 \mathbf{x}'_N(k) e_o(k).\end{aligned}\quad (5.44)$$

Корреляционная матрица векторов (5.43) и (5.44) определяется как среднее значение произведения векторов $\mathbf{v}_N(k) \mathbf{v}_N^H(k)$ и $\mathbf{v}_N(k) \mathbf{v}_N^T(k)$. Это произведение в случае комплексных ВК определяется как

$$\begin{aligned}\mathbf{v}_N(k) \mathbf{v}_N^H(k) &= \mathbf{v}_N(k-1) \mathbf{v}_N^H(k-1) - \\ &- \mu \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) \mathbf{v}_N(k-1) \mathbf{v}_N^H(k-1) + \\ &+ \mu \mathbf{x}'_N(k) e_o^*(k) \mathbf{v}_N^H(k-1) - \\ &- \mathbf{v}_N(k-1) \mu \mathbf{v}_N^H(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) + \\ &+ \mu \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) \mathbf{v}_N(k-1) \mu \mathbf{v}_N^H(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) - \\ &- \mu \mathbf{x}'_N(k) e_o^*(k) \mu \mathbf{v}_N^H(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) + \\ &+ \mathbf{v}_N(k-1) \mu e_o(k) \mathbf{x}'_N^H(k) - \\ &- \mu \mathbf{x}'_N(k) \mathbf{x}'_N^H(k) \mathbf{v}_N(k-1) \mu e_o(k) \mathbf{x}'_N^H(k) + \\ &+ \mu \mathbf{x}'_N(k) e_o^*(k) \mu e_o(k) \mathbf{x}'_N^H(k),\end{aligned}\quad (5.45)$$

а в случае действительных ВК — как

$$\begin{aligned}\mathbf{v}_N(k) \mathbf{v}_N^T(k) &= \mathbf{v}_N(k-1) \mathbf{v}_N^T(k-1) - \\ &- 2\mu \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) \mathbf{v}_N(k-1) \mathbf{v}_N^T(k-1) + \\ &+ 2\mu \mathbf{x}'_N(k) e_o(k) \mathbf{v}_N^T(k-1) - \\ &- \mathbf{v}_N(k-1) 2\mu \mathbf{v}_N^T(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) + \\ &+ 2\mu \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) \mathbf{v}_N(k-1) 2\mu \mathbf{v}_N^T(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) - \\ &- 2\mu \mathbf{x}'_N(k) e_o(k) 2\mu \mathbf{v}_N^T(k-1) \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) + \\ &+ \mathbf{v}_N(k-1) 2\mu e_o(k) \mathbf{x}'_N^T(k) - \\ &- 2\mu \mathbf{x}'_N(k) \mathbf{x}'_N^T(k) \mathbf{v}_N(k-1) 2\mu e_o(k) \mathbf{x}'_N^T(k) + \\ &+ 2\mu \mathbf{x}'_N(k) e_o(k) 2\mu e_o(k) \mathbf{x}'_N^T(k).\end{aligned}\quad (5.46)$$

Далее, используя уравнения (5.45) и (5.46), получим, что корреляционные матрицы векторов (5.43) и (5.44) определяются как

$$\begin{aligned}
\mathbf{K}_N(k) &= E\{\mathbf{v}_N(k)\mathbf{v}_N^H(k)\} = \\
&= E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} - \\
&\quad - \mu E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} - \\
&\quad - \mu E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} + \\
&\quad + \mu E\{e_o^*(k)\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\} + \\
&\quad + \mu E\{e_o(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} + \\
&\quad + \mu^2 E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} - \\
&\quad - \mu^2 E\{e_o^*(k)\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} - \\
&\quad - \mu^2 E\{e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} + \\
&\quad + \mu^2 E\{e_o^*(k)e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\}
\end{aligned} \tag{5.47}$$

И

$$\begin{aligned}
\mathbf{K}_N(k) &= E\{\mathbf{v}_N(k)\mathbf{v}_N^T(k)\} = \\
&= E\{\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\} - \\
&\quad - 2\mu E\{\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\} - \\
&\quad - 2\mu E\{\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\} + \\
&\quad + 2\mu E\{e_o(k)\mathbf{x}'_N(k)\mathbf{v}_N^T(k-1)\} + \\
&\quad + 2\mu E\{e_o(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^T(k)\} + \\
&\quad + 4\mu^2 E\{\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\} - \\
&\quad - 4\mu^2 E\{e_o(k)\mathbf{x}'_N(k)\mathbf{v}_N^T(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\} - \\
&\quad - 4\mu^2 E\{e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^T(k)\} + \\
&\quad + 4\mu^2 E\{e_o(k)e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\}.
\end{aligned} \tag{5.48}$$

Для упрощения выражений (5.47) и (5.48) используются следующие соотношения между независимыми случайными переменными

$$E\{xy\} = E\{x\}E\{y\} = E\{xE[y]\} \tag{5.49}$$

И

$$E\{x^2y^2\} = E\{x^2\}E\{y^2\} = E\{x^2E[y^2]\} = E\{xE[y^2]x\}. \tag{5.50}$$

С учетом этих соотношений, а также уравнения (3.46) $\mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{\Lambda}_N$, можно показать, что

$$\begin{aligned}
& E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} = \\
& = E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} = \\
& = E\{\mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N\} E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} = \\
& = \mathbf{Q}_N^H E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} \mathbf{Q}_N E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} = \\
& = \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} = \mathbf{\Lambda}_N \mathbf{K}_N(k-1)
\end{aligned} \tag{5.51}$$

и аналогично

$$\begin{aligned}
& E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \\
& = E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \\
& = E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} E\{\mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N\} = \\
& = E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} \mathbf{Q}_N^H E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} \mathbf{Q}_N = \\
& = E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\} \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{K}_N(k-1) \mathbf{\Lambda}_N.
\end{aligned} \tag{5.52}$$

Кроме того,

$$E\{e_o^*(k)\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\} = E\{e_o^*(k)\} E\{\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\} = \mathbf{0}_N \tag{5.53}$$

так как $E\{e_o^*(k)\} = 0$, поскольку $e_o^*(k)$ имеет нулевое среднее значение, и

$$E\{e_o(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} = E\{e_o(k)\} E\{\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} = \mathbf{0}_N, \tag{5.54}$$

так как $E\{e_o(k)\} = 0$, поскольку $e_o(k)$ имеет нулевое среднее значение.

Аналогично

$$\begin{aligned}
& E\{e_o^*(k)\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \\
& = E\{e_o^*(k)\} E\{\mathbf{x}'_N(k)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \mathbf{0}_N
\end{aligned} \tag{5.55}$$

и

$$\begin{aligned}
& E\{e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} = \\
& = E\{e_o(k)\} E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{x}'_N^H(k)\} = \mathbf{0}_N.
\end{aligned} \tag{5.56}$$

Также с учетом выражения (3.46) соотношение

$$\begin{aligned}
& E\{e_o^*(k)e_o(k)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = E\{e_o^*(k)e_o(k)\} E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \\
& = E\{|e_o(k)|^2\} E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = E\{|e_o(k)|^2\} E\{\mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N\} = \\
& = E\{|e_o(k)|^2\} \mathbf{Q}_N^H E\{\mathbf{x}_N(k)\mathbf{x}_N^H(k)\} \mathbf{Q}_N = \xi_{\min} \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \xi_{\min} \mathbf{\Lambda}_N.
\end{aligned} \tag{5.57}$$

Следовательно, используя выражения (5.51)–(5.57), уравнение (5.47) можно преобразовать к виду

$$\mathbf{K}_N(k) = \mathbf{K}_N(k-1) - \mu [\mathbf{\Lambda}_N \mathbf{K}_N(k-1) - \mathbf{K}_N(k-1) \mathbf{\Lambda}_N] + \mu^2 \mathbf{A}_N(k-1) + \mu^2 \xi_{\min} \mathbf{\Lambda}_N, \tag{5.58}$$

где

$$\mathbf{A}_N(k-1) = E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\}. \quad (5.59)$$

Выполнив преобразования (5.51)–(5.57) применительно к уравнению (5.48), получим, что в случае действительных ВК

$$\mathbf{K}_N(k) = \mathbf{K}_N(k-1) - 2\mu[\mathbf{A}_N\mathbf{K}_N(k-1) - \mathbf{K}_N(k-1)\mathbf{A}_N] + 4\mu^2\mathbf{A}_N(k-1) + 4\mu^2\xi_{\min}\mathbf{A}_N, \quad (5.60)$$

где

$$\mathbf{A}_N(k-1) = E\{\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\}. \quad (5.61)$$

Вычисление матрицы $\mathbf{A}_N(k-1)$ является довольно сложной задачей. Эта задача упрощается в предположении, что векторы $\mathbf{x}'_N(k)$ и $\mathbf{v}_N(k)$ являются независимыми. При вычислении матрицы $\mathbf{A}_N(k-1)$ также используются следующие соотношения для выражения моментов четвертого порядка нормально распределенных случайных переменных в терминах моментов второго порядка. Согласно [77], если x_1, x_2, x_3, x_4 — комплексные нормально распределенные переменные с нулевым средним, то

$$E\{x_1x_2^*x_3x_4^*\} = E\{x_1x_2^*\}E\{x_3x_4^*\} + E\{x_1x_4^*\}E\{x_2^*x_3\}, \quad (5.62)$$

а если x_1, x_2, x_3, x_4 — действительные нормально распределенные переменные с нулевым средним, то

$$E\{x_1x_2x_3x_4\} = E\{x_1x_2\}E\{x_3x_4\} + E\{x_1x_3\}E\{x_2x_4\} + E\{x_1x_4\}E\{x_2x_3\}. \quad (5.63)$$

Тогда с учетом выражения (5.50) уравнение (5.59) принимает вид

$$\begin{aligned} \mathbf{A}_N(k-1) &= E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)E\{\mathbf{v}_N(k-1)\mathbf{v}_N^H(k-1)\}\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\} = \\ &= E\{\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\}, \end{aligned} \quad (5.64)$$

а уравнение (5.61) — вид

$$\begin{aligned} \mathbf{A}_N(k-1) &= E\{\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)E\{\mathbf{v}_N(k-1)\mathbf{v}_N^T(k-1)\}\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\} = \\ &= E\{\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\}. \end{aligned} \quad (5.65)$$

Для упрощения выражений (5.64) и (5.65) определим скалярные переменные

$$\mathbf{x}'_N^H(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k) = \sum_{i=1}^N \sum_{j=1}^N x_i''(k)x_j'(k)k_{ij}(k-1), \quad (5.66)$$

$$\mathbf{x}'_N^T(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k) = \sum_{i=1}^N \sum_{j=1}^N x_i'(k)x_j'(k)k_{ij}(k-1) \quad (5.67)$$

и матрицы

$$\mathbf{C}_N(k-1) = \mathbf{x}'_N(k)\mathbf{x}'_N^H(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^H(k), \quad (5.68)$$

$$\mathbf{C}_N(k-1) = \mathbf{x}'_N(k)\mathbf{x}'_N^T(k)\mathbf{K}_N(k-1)\mathbf{x}'_N(k)\mathbf{x}'_N^T(k). \quad (5.69)$$

Элементы матриц (5.68) и (5.69) определяются соответственно как

$$\begin{aligned} c_{lm}(k-1) &= x'_l(k) \left[\sum_{i=1}^N \sum_{j=1}^N x_i'^*(k) x_j'(k) k_{ij}(k-1) \right] x_m'^*(k) = \\ &= \sum_{i=1}^N \sum_{j=1}^N x_i'(k) x_m'^*(k) x_i'^*(k) x_j'(k) k_{ij}(k-1) \end{aligned} \quad (5.70)$$

и

$$\begin{aligned} c_{lm}(k-1) &= x'_l(k) \left[\sum_{i=1}^N \sum_{j=1}^N x_i'(k) x_j'(k) k_{ij}(k-1) \right] x_m'(k) = \\ &= \sum_{i=1}^N \sum_{j=1}^N x_i'(k) x_m'(k) x_i'(k) x_j'(k) k_{ij}(k-1), \end{aligned} \quad (5.71)$$

а их средние значения — как

$$E\{c_{lm}(k-1)\} = \sum_{i=1}^N \sum_{j=1}^N E\{x_i'(k) x_m'^*(k) x_i'^*(k) x_j'(k)\} k_{ij}(k-1) \quad (5.72)$$

и

$$E\{c_{lm}(k-1)\} = \sum_{i=1}^N \sum_{j=1}^N E\{x_i'(k) x_m'(k) x_i'(k) x_j'(k)\} k_{ij}(k-1). \quad (5.73)$$

Так как согласно уравнению (5.42) $\mathbf{x}'_N(k) = \mathbf{Q}_N^H \mathbf{x}_N(k)$, то согласно уравнению (3.46)

$$\begin{aligned} E\{\mathbf{x}'_N(k) \mathbf{x}'_N^H(k)\} &= E\{\mathbf{Q}_N^H \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{Q}_N\} = \\ &= \mathbf{Q}_N^H E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} \mathbf{Q}_N = \mathbf{Q}_N^H \mathbf{R}_N \mathbf{Q}_N = \mathbf{\Lambda}_N. \end{aligned} \quad (5.74)$$

Это означает, что

$$E\{x_i'(k) x_j'^*(k)\} = \lambda_i \delta(i-j) \quad (5.75)$$

и

$$E\{x_i'(k) x_j'(k)\} = \lambda_i \delta(i-j). \quad (5.76)$$

С учетом выражений (5.75), (5.76) и (5.62), (5.63) средние значения произведения четырех независимых переменных в уравнениях (5.72) и (5.73) определяются как

$$\begin{aligned} E\{x_i'(k) x_m'^*(k) x_i'^*(k) x_j'(k)\} &= \\ &= E\{x_i'(k) x_m'^*(k)\} E\{x_i'^*(k) x_j'(k)\} + E\{x_i'(k) x_i'^*(k)\} E\{x_m'^*(k) x_j'(k)\} = \\ &= \lambda_i \delta(l-m) \lambda_i \delta(i-j) + \lambda_i \delta(l-i) \lambda_m \delta(m-j) = \lambda_i \lambda_i \delta(l-m) \delta(i-j) + \lambda_i \lambda_m \delta(l-i) \delta(m-j) \end{aligned} \quad (5.77)$$

и

$$\begin{aligned} E\{x_i'(k) x_m'(k) x_i'(k) x_j'(k)\} &= E\{x_i'(k) x_m'(k)\} E\{x_i'(k) x_j'(k)\} + \\ &+ E\{x_i'(k) x_i'(k)\} E\{x_m'(k) x_j'(k)\} + E\{x_i'(k) x_j'(k)\} E\{x_m'(k) x_i'(k)\} = \\ &= \lambda_i \delta(l-m) \lambda_i \delta(i-j) + \lambda_i \delta(l-i) \lambda_m \delta(m-j) + \lambda_i \delta(l-j) \lambda_m \delta(m-i) = \\ &= \lambda_i \lambda_i \delta(l-m) \delta(i-j) + \lambda_i \lambda_m \delta(l-i) \delta(m-j) + \lambda_i \lambda_m \delta(l-j) \delta(m-i). \end{aligned} \quad (5.78)$$

Поэтому уравнение (5.72) принимает вид

$$\begin{aligned}
 E\{c_{lm}(k-1)\} &= \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j \delta(l-m) \delta(i-j) k_{ij}(k-1) + \\
 &+ \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_m \delta(l-i) \delta(m-j) k_{ij}(k-1) = \\
 &= \lambda_l \delta(l-m) \sum_{i=1}^N \lambda_i k'_{ii}(k-1) + \lambda_l \lambda_m k_{lm}(k-1) = \\
 &= \lambda_l \delta(l-m) \sum_{i=1}^N \lambda_i k'_{ii}(k-1) + \lambda_l \lambda_m k_{lm}(k-1),
 \end{aligned} \tag{5.79}$$

а уравнение (5.64) — вид

$$\mathbf{A}_N(k-1) = E\{\mathbf{C}_N(k-1)\} = \mathbf{\Lambda}_N \mathbf{K}_N(k-1) \mathbf{\Lambda}_N + \text{tr}\{\mathbf{\Lambda}_N \mathbf{K}_N(k-1)\} \mathbf{\Lambda}_N. \tag{5.80}$$

Аналогично уравнение (5.73) принимает вид

$$\begin{aligned}
 E\{c_{lm}(k-1)\} &= \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j \delta(l-m) \delta(i-j) k'_{ij}(k-1) + \\
 &+ \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_m \delta(l-i) \delta(m-j) k'_{ij}(k-1) + \\
 &+ \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_m \delta(l-j) \delta(m-i) k'_{ij}(k-1) = \\
 &= \lambda_l \delta(l-m) \sum_{i=1}^N \lambda_i k'_{ii}(k-1) + \lambda_l \lambda_m k'_{lm}(k-1) + \lambda_l \lambda_m k'_{ml}(k-1) = \\
 &= \lambda_l \delta(l-m) \sum_{i=1}^N \lambda_i k'_{ii}(k-1) + 2\lambda_l \lambda_m k'_{lm}(k-1),
 \end{aligned} \tag{5.81}$$

так как $k'_{lm}(k-1) = k'_{ml}(k-1)$, а уравнение (5.65) — вид

$$\mathbf{A}_N(k-1) = E\{\mathbf{C}_N(k-1)\} = 2\mathbf{\Lambda}_N \mathbf{K}_N(k-1) \mathbf{\Lambda}_N + \text{tr}\{\mathbf{\Lambda}_N \mathbf{K}_N(k-1)\} \mathbf{\Lambda}_N. \tag{5.82}$$

Следовательно, уравнения (5.58) и (5.60) приобретают окончательный вид

$$\begin{aligned}
 \mathbf{K}_N(k) &= \mathbf{K}_N(k-1) - \mu [\mathbf{\Lambda}_N \mathbf{K}_N(k-1) + \mathbf{K}_N(k-1) \mathbf{\Lambda}_N] + \\
 &+ \mu^2 [\mathbf{\Lambda}_N \mathbf{K}_N(k-1) \mathbf{\Lambda}_N + \text{tr}\{\mathbf{\Lambda}_N \mathbf{K}_N(k-1)\} \mathbf{\Lambda}_N] + \mu^2 \xi_{\min} \mathbf{\Lambda}_N
 \end{aligned} \tag{5.83}$$

и

$$\begin{aligned}
 \mathbf{K}_N(k) &= \mathbf{K}_N(k-1) - 2\mu [\mathbf{\Lambda}_N \mathbf{K}_N(k-1) + \mathbf{K}_N(k-1) \mathbf{\Lambda}_N] + \\
 &+ 4\mu^2 [2\mathbf{\Lambda}_N \mathbf{K}_N(k-1) \mathbf{\Lambda}_N + \text{tr}\{\mathbf{\Lambda}_N \mathbf{K}_N(k-1)\} \mathbf{\Lambda}_N] + 4\mu^2 \xi_{\min} \mathbf{\Lambda}_N,
 \end{aligned} \tag{5.84}$$

где

$$\text{tr}\{\mathbf{\Lambda}_N \mathbf{K}_N(k-1)\} = \sum_{j=1}^N \lambda_j k_{jj}(k-1), \tag{5.85}$$

так как $\mathbf{\Lambda}_N$ — диагональная матрица, см. уравнение (3.42).

Анализ уравнений (5.83) и (5.84) можно упростить, рассмотрев только диагональные элементы матрицы $\mathbf{K}_N(k)$.

С учетом (5.85) диагональные элементы матрицы $\mathbf{K}_N(k)$ в уравнении (5.83) определяются как

$$\begin{aligned} k_{ii}(k) &= k_{ii}(k-1) - \mu \left[\lambda_i k_{ii}(k-1) + k_{ii}(k-1) \lambda_i \right] + \\ &+ \mu^2 \left\{ \lambda_i k_{ii}(k-1) \lambda_i + \left[\sum_{j=1}^N \lambda_j k_{jj}(k-1) \right] \lambda_i \right\} + \mu^2 \xi_{\min} \lambda_i = \\ &= k_{ii}(k-1) - 2\mu \lambda_i k_{ii}(k-1) + \mu^2 \lambda_i^2 k_{ii}(k-1) + \mu^2 \lambda_i \sum_{j=1}^N \lambda_j k_{jj}(k-1) + \\ &+ \mu^2 \xi_{\min} \lambda_i = \rho_i k_{ii}(k-1) + \mu^2 \lambda_i \sum_{j=1}^N \lambda_j k_{jj}(k-1) + \mu^2 \xi_{\min} \lambda_i, \end{aligned} \quad (5.86)$$

где

$$\rho_i = 1 - 2\mu \lambda_i + \mu^2 \lambda_i^2, \quad (5.87)$$

а диагональные элементы матрицы $\mathbf{K}_N(k)$ уравнения (5.84) — как

$$\begin{aligned} k_{ii}(k) &= k_{ii}(k-1) - 2\mu \left[\lambda_i k_{ii}(k-1) + k_{ii}(k-1) \lambda_i \right] + \\ &+ 4\mu^2 \left\{ 2\lambda_i k_{ii}(k-1) \lambda_i + \left[\sum_{j=1}^N \lambda_j k_{jj}(k-1) \right] \lambda_i \right\} + 4\mu^2 \xi_{\min} \lambda_i = \\ &= k_{ii}(k-1) - 4\mu \lambda_i k_{ii}(k-1) + 8\mu^2 \lambda_i^2 k_{ii}(k-1) + 4\mu^2 \lambda_i \sum_{j=1}^N \lambda_j k_{jj}(k-1) + \\ &+ 4\mu^2 \xi_{\min} \lambda_i = \rho_i k_{ii}(k-1) + 4\mu^2 \lambda_i \sum_{j=1}^N \lambda_j k_{jj}(k-1) + 4\mu^2 \xi_{\min} \lambda_i, \end{aligned} \quad (5.88)$$

где

$$\rho_i = 1 - 4\mu \lambda_i + 8\mu^2 \lambda_i^2. \quad (5.89)$$

Так как матрица $\mathbf{K}_N(k)$ представляет собой корреляционную матрицу ВК $\mathbf{v}_N(k)$, то для ее элементов справедливо соотношение $k_{ii}(k) \leq |k_{ij}(k)|$ для всех i и j . Поэтому можно заключить, что сходимость диагональных элементов матрицы $\mathbf{K}_N(k)$, т. е. сохранение ими ограниченных значений при увеличении номера итерации k , является достаточным условием для сходимости и всех остальных элементов этой матрицы, что, собственно, и требуется для обеспечения устойчивой работы LMS-алгоритма. Поэтому при анализе уравнений (5.83) и (5.84) достаточно сосредоточиться только на диагональных элементах матрицы $\mathbf{K}_N(k)$.

Для этого определим вектор диагональных элементов матрицы $\mathbf{K}_N(k)$

$$\mathbf{k}_N(k) = [k_{11}(k), k_{22}(k), \dots, k_{NN}(k)]^T, \quad (5.90)$$

вектор собственных чисел

$$\boldsymbol{\lambda}_N = [\lambda_1, \lambda_2, \dots, \lambda_N]^T, \quad (5.91)$$

матрицу

$$\mathbf{F}_N = \text{diag}\{\rho_1, \rho_2, \dots, \rho_N\} + \mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \quad (5.92)$$

в случае уравнения (5.83) и матрицу

$$\mathbf{F}_N = \text{diag}\{\rho_1, \rho_2, \dots, \rho_N\} + 4\mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \quad (5.93)$$

в случае уравнения (5.84).

С учетом выражений (5.90)–(5.93) уравнения для диагональных элементов матриц (5.86) и (5.88) принимают вид

$$\mathbf{k}_N(k) = \mathbf{F}_N \mathbf{k}_N(k-1) + \mu^2 \xi_{\min} \boldsymbol{\lambda}_N, \quad (5.94)$$

$$\mathbf{k}_N(k) = \mathbf{F}_N \mathbf{k}_N(k-1) + 4\mu^2 \xi_{\min} \boldsymbol{\lambda}_N. \quad (5.95)$$

Уравнения (5.94) и (5.95) могут быть использованы для анализа устойчивости работы LMS-алгоритма. Как уже отмечалось, LMS-алгоритм устойчив, если элементы матрицы $\mathbf{K}_N(k)$, или, что то же самое, элементы вектора $\mathbf{k}_N(k)$, остаются ограниченными при увеличении числа итераций k . В [76] было показано, что устойчивая работа LMS-алгоритм обеспечивается, если все собственные числа матрицы \mathbf{F}_N меньше единицы.

Однако в настоящей книге условие устойчивости LMS-алгоритма будет получено в терминах допустимого значения шага сходимости μ [18], более строгого, чем определяемого уравнениями (5.19)–(5.21) и (5.23)–(5.25), поскольку эти уравнения гарантируют сходимость вектора ВК адаптивного фильтра $\mathbf{h}_N(k)$ к винеровскому решению $\mathbf{h}_{N,o}$ лишь в среднем. Для проведения данного анализа необходимо получить аналитическое выражение относительной дополнительной среднеквадратической ошибки (5.38).

Для этого сначала, используя уравнение (5.31), выразим значение среднеквадратической ошибки моделирования требуемого сигнала в LMS-алгоритме

$$\xi(k) = \xi_{\min} + E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1)\} \quad (5.96)$$

в терминах элементов матрицы $\mathbf{K}_N(k)$.

Так как оба слагаемые в уравнение (5.96) являются скалярными, то можно формально записать, что

$$E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1)\} = \text{tr}\{E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1)\}\}, \quad (5.97)$$

поскольку след скалярной величины (матрицы с числом элементов 1×1) равен этой величине.

Тогда с учетом выражений (5.50), (3.52) и (5.97) уравнение (5.96) преобразуется как

$$\begin{aligned}
 \xi(k) &= \xi_{\min} + \text{tr} \left\{ E \left\{ \Delta \mathbf{h}_N^H(k-1) E \left\{ \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right\} \Delta \mathbf{h}_N(k-1) \right\} \right\} = \\
 &= \xi_{\min} + \text{tr} \left\{ E \left\{ \Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1) \right\} \right\} = \\
 &= \xi_{\min} + \text{tr} \left\{ E \left\{ \Delta \mathbf{h}_N(k-1) \Delta \mathbf{h}_N^H(k-1) \right\} \mathbf{R}_N \right\} = \\
 &= \xi_{\min} + \text{tr} \left\{ E \left\{ \Delta \mathbf{h}_N(k-1) \Delta \mathbf{h}_N^H(k-1) \right\} \mathbf{Q}_N \mathbf{\Lambda}_N \mathbf{Q}_N^H \right\} = \\
 &= \xi_{\min} + \text{tr} \left\{ E \left\{ \mathbf{Q}_N^H \Delta \mathbf{h}_N(k-1) \Delta \mathbf{h}_N^H(k-1) \mathbf{Q}_N \right\} \mathbf{\Lambda}_N \right\} = \\
 &= \xi_{\min} + \text{tr} \left\{ \mathbf{K}_N(k-1) \mathbf{\Lambda}_N \right\} = \xi_{\min} + \sum_{n=1}^N \lambda_n k_{nn}(k-1),
 \end{aligned} \tag{5.98}$$

так как $\mathbf{\Lambda}_N$ — диагональная матрица, см. выражение (3.42).

При $\mathbf{h}_N(k) \rightarrow \mathbf{h}_{N,o}$ градиент функции среднеквадратической ошибки (5.31), или, что то же самое, градиент функции среднеквадратической ошибки (5.98), стремится к нулю, хотя мгновенное значение этого градиента, см. уравнение (5.2),

$$\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -\mathbf{x}_N(k) \alpha^*(k) \tag{5.99}$$

может быть и ненулевым. Следствием этого является то, что вектор ВК $\mathbf{h}_N(k)$ совершает зигзагообразные движения в районе точки $\mathbf{h}_{N,o}$ и $\xi(k) > \xi_{\min}$. То есть

$$\xi(k) = \xi_{\min} + \xi_{\text{exc}}. \tag{5.100}$$

Значение дополнительной среднеквадратической ошибки ξ_{exc} можно оценить следующим образом. При $k \rightarrow \infty$, т. е. в установившемся режиме, справедливо соотношение $\mathbf{k}_N(k) \approx \mathbf{k}_N(k-1)$. Тогда из уравнения (5.98) следует, что

$$\xi(\infty) = \xi_{\min} + \text{tr} \left\{ \mathbf{K}_N(\infty) \mathbf{\Lambda}_N \right\} = \xi_{\min} + \sum_{n=1}^N \lambda_n k'_{nn}(\infty) = \xi_{\min} + \mathbf{\lambda}_N^T \mathbf{k}_N(\infty) = \xi_{\min} + \xi_{\text{exc}}, \tag{5.101}$$

где

$$\xi_{\text{exc}} = \text{tr} \left\{ \mathbf{K}_N(\infty) \mathbf{\Lambda}_N \right\} = \sum_{n=1}^N \lambda_n k_{nn}(\infty) = \mathbf{\lambda}_N^T \mathbf{k}_N(\infty). \tag{5.102}$$

Согласно выражению (5.94)

$$\mathbf{k}_N(\infty) = \mathbf{F}_N \mathbf{k}_N(\infty) + \mu^2 \xi_{\min} \mathbf{\lambda}_N \tag{5.103}$$

или

$$\mathbf{k}_N(\infty) - \mathbf{F}_N \mathbf{k}_N(\infty) = [\mathbf{I}_N - \mathbf{F}_N] \mathbf{k}_N(\infty) = \mu^2 \xi_{\min} \mathbf{\lambda}_N, \tag{5.104}$$

откуда

$$\mathbf{k}_N(\infty) = \mu^2 \xi_{\min} [\mathbf{I}_N - \mathbf{F}_N]^{-1} \mathbf{\lambda}_N. \tag{5.105}$$

Аналогично для уравнения (5.95)

$$\mathbf{k}_N(\infty) = \mathbf{F}_N \mathbf{k}_N(\infty) + 4\mu^2 \xi_{\min} \boldsymbol{\lambda}_N \quad (5.106)$$

или

$$\mathbf{k}_N(\infty) - \mathbf{F}_N \mathbf{k}_N(\infty) = [\mathbf{I}_N - \mathbf{F}_N] \mathbf{k}_N(\infty) = 4\mu^2 \xi_{\min} \boldsymbol{\lambda}_N, \quad (5.107)$$

откуда

$$\mathbf{k}_N(\infty) = 4\mu^2 \xi_{\min} [\mathbf{I}_N - \mathbf{F}_N]^{-1} \boldsymbol{\lambda}_N. \quad (5.108)$$

Тогда, используя уравнения (5.102), (5.105) и (5.108), для адаптивного фильтра с комплексными ВК значение дополнительной среднеквадратической ошибки можно определить как

$$\xi_{\text{exc}} = \mu^2 \xi_{\min} \boldsymbol{\lambda}_N^T [\mathbf{I}_N - \mathbf{F}_N]^{-1} \boldsymbol{\lambda}_N, \quad (5.109)$$

а для адаптивного фильтра с действительными ВК — как

$$\xi_{\text{exc}} = 4\mu^2 \xi_{\min} \boldsymbol{\lambda}_N^T [\mathbf{I}_N - \mathbf{F}_N]^{-1} \boldsymbol{\lambda}_N. \quad (5.110)$$

Соответственно, относительное значение дополнительной среднеквадратической ошибки определяется как

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} = \mu^2 \boldsymbol{\lambda}_N^T [\mathbf{I}_N - \mathbf{F}_N]^{-1} \boldsymbol{\lambda}_N \quad (5.111)$$

в случае LMS-алгоритма для адаптивного фильтра с комплексными ВК и как

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} = 4\mu^2 \boldsymbol{\lambda}_N^T [\mathbf{I}_N - \mathbf{F}_N]^{-1} \boldsymbol{\lambda}_N \quad (5.112)$$

в случае LMS-алгоритма для адаптивного фильтра с действительными ВК.

Для обращения матрицы в уравнениях (5.111) и (5.112) необходимо воспользоваться леммой об обращении матриц (Matrix Inversion Lemma, MIL) [8], которая для матриц вида

$$\begin{aligned} \mathbf{I}_N - \mathbf{F}_N &= \mathbf{I}_N - \text{diag}\{\rho_1, \rho_2, \dots, \rho_N\} - \mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T = \\ &= \text{diag}\{1-\rho_1, 1-\rho_2, \dots, 1-\rho_N\} - \mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \end{aligned} \quad (5.113)$$

и

$$\begin{aligned} \mathbf{I}_N - \mathbf{F}_N &= \mathbf{I}_N - \text{diag}\{\rho_1, \rho_2, \dots, \rho_N\} - 4\mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T = \\ &= \text{diag}\{1-\rho_1, 1-\rho_2, \dots, 1-\rho_N\} - 4\mu^2 \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T, \end{aligned} \quad (5.114)$$

т. е. $\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D}$, представляется как

$$(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}, \quad (5.115)$$

где $\mathbf{C} = \text{diag}\{1-\rho_1, 1-\rho_2, \dots, 1-\rho_N\}$, $\mathbf{C}^{-1} = \text{diag}\{1-\rho_1, 1-\rho_2, \dots, 1-\rho_N\}^{-1}$, $\mathbf{B}^H = \mu \boldsymbol{\lambda}_N$, $\mathbf{D} = \mu \boldsymbol{\lambda}_N^T$ в уравнении (5.113) и $\mathbf{B}^H = 2\mu \boldsymbol{\lambda}_N$, $\mathbf{D} = 2\mu \boldsymbol{\lambda}_N^T$ в уравнении (5.114), а $\mathbf{A}^{-1} = 1$.

Используя выражение (5.115), можно получить обратное значение матрицы (5.113) в виде

$$\mathbf{C}^{-1} - \frac{\mathbf{C}^{-1} \mu \boldsymbol{\lambda}_N \mu \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{\mu \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \mu \boldsymbol{\lambda}_N - 1} = \mathbf{C}^{-1} - \frac{\mu^2 \mathbf{C}^{-1} \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{\mu^2 \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N - 1} \quad (5.116)$$

и обратное значение матрицы (5.114) в виде

$$\mathbf{C}^{-1} - \frac{\mathbf{C}^{-1} 2\mu \boldsymbol{\lambda}_N 2\mu \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{2\mu \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} 2\mu \boldsymbol{\lambda}_N - 1} = \mathbf{C}^{-1} - \frac{4\mu^2 \mathbf{C}^{-1} \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{4\mu^2 \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N - 1}. \quad (5.117)$$

Тогда согласно уравнениям (5.111), (5.116) и (5.87)

$$\begin{aligned} M &= \frac{\xi_{\text{exc}}}{\xi_{\min}} = \mu^2 \boldsymbol{\lambda}_N^T \left[\mathbf{C}^{-1} - \frac{\mu^2 \mathbf{C}^{-1} \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{\mu^2 \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N - 1} \right] \boldsymbol{\lambda}_N = \frac{\mu^2 [\boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N]}{1 - \mu^2 [\boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N]} = \\ &= \frac{\mu^2 \sum_{n=1}^N \frac{\lambda_n^2}{1 - \rho_n}}{1 - \mu^2 \sum_{n=1}^N \frac{\lambda_n^2}{1 - \rho_n}} = \frac{\sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n}}{1 - \sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n}} \end{aligned} \quad (5.118)$$

и согласно уравнениям (5.112), (5.117) и (5.89)

$$\begin{aligned} M &= \frac{\xi_{\text{exc}}}{\xi_{\min}} = 4\mu^2 \boldsymbol{\lambda}_N^T \left[\mathbf{C}^{-1} - \frac{4\mu^2 \mathbf{C}^{-1} \boldsymbol{\lambda}_N \boldsymbol{\lambda}_N^T \mathbf{C}^{-1}}{4\mu^2 \boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N - 1} \right] \boldsymbol{\lambda}_N = \frac{4\mu^2 [\boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N]}{1 - 4\mu^2 [\boldsymbol{\lambda}_N^T \mathbf{C}^{-1} \boldsymbol{\lambda}_N]} = \\ &= \frac{4\mu^2 \sum_{n=1}^N \frac{\lambda_n^2}{1 - \rho_n}}{1 - 4\mu^2 \sum_{n=1}^N \frac{\lambda_n^2}{1 - \rho_n}} = \frac{\sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n}}{1 - \sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n}}. \end{aligned} \quad (5.119)$$

Результаты (5.118) и (5.119) полезно упростить с целью их использования для определения предельного значения шага сходимости μ . На практике μ выбирается так, чтобы значение относительной дополнительной среднеквадратической ошибки не превышало 10%, т. е. $M \leq 0.1$. Это достигается, если $\mu \lambda_n \ll 1$ в случае (5.118) или $2\mu \lambda_n \ll 1$ в случае (5.119). Тогда уравнения (5.118) и (5.119) с учетом (3.53) можно аппроксимировать как

$$M = \frac{\sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n}}{1 - \sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n}} \approx \frac{\sum_{n=1}^N \frac{\mu \lambda_n}{2}}{1 - \sum_{n=1}^N \frac{\mu \lambda_n}{2}} = \frac{\mu \sum_{n=1}^N \lambda_n}{2 - \mu \sum_{n=1}^N \lambda_n} = \frac{\mu \text{tr}\{\mathbf{R}_N\}}{2 - \mu \text{tr}\{\mathbf{R}_N\}} \quad (5.120)$$

и как

$$M = \frac{\sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n}}{1 - \sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n}} \approx \frac{\sum_{n=1}^N \mu \lambda_n}{1 - \sum_{n=1}^N \mu \lambda_n} = \frac{\mu \sum_{n=1}^N \lambda_n}{1 - \mu \sum_{n=1}^N \lambda_n} = \frac{\mu \text{tr}\{\mathbf{R}_N\}}{1 - \mu \text{tr}\{\mathbf{R}_N\}}. \quad (5.121)$$

В настоящем разделе уже отмечалось, что условие сходимости в среднем вектора ВК $\mathbf{h}_N(k)$ адаптивного фильтра на базе LMS-алгоритма к винеровскому решению $\mathbf{h}_{N,o}$ определяется уравнениями (5.19)–(5.21) и (5.23)–(5.25). Однако вследствие того, что оценка вектора градиента (5.2) $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -\mathbf{x}_N(k)\alpha^*(k)$ подвержена случайным флуктуациям, верхняя граница шага сходимости μ , при которой гарантируется устойчивое поведение LMS-алгоритма, должна быть несколько меньше, чем значение, определяемое уравнениями (5.19)–(5.21), (5.23)–(5.25).

Значение шага сходимости LMS-алгоритма, гарантирующее его устойчивость, можно получить, используя уравнения (5.118) и (5.119).

Для этого определим функции

$$J(\mu) = \sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n} \quad (5.122)$$

и

$$J(\mu) = \sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n}. \quad (5.123)$$

Тогда с учетом выражений (5.122) и (5.123) уравнения (5.118) и (5.119) примут вид

$$M = \frac{J(\mu)}{1 - J(\mu)}. \quad (5.124)$$

Согласно уравнению (5.122)

$$\frac{dJ(\mu)}{d\mu} = \sum_{n=1}^N \frac{(2 - \mu \lambda_n) \lambda_n - \mu \lambda_n (-\mu \lambda_n)}{(2 - \mu \lambda_n)^2} = \sum_{n=1}^N \frac{2\lambda_n}{(2 - \mu \lambda_n)^2} \quad (5.125)$$

и согласно уравнению (5.123)

$$\frac{dJ(\mu)}{d\mu} = \sum_{n=1}^N \frac{(1 - 2\mu \lambda_n) \lambda_n - \mu \lambda_n (-2\mu \lambda_n)}{(1 - 2\mu \lambda_n)^2} = \sum_{n=1}^N \frac{\lambda_n}{(1 - 2\mu \lambda_n)^2}. \quad (5.126)$$

Из выражений (5.125) и (5.126) следует, что функция $J(\mu)$ является возрастающей, т. е. $J(\mu_2) > J(\mu_1)$, если $\mu_2 > \mu_1$, и $\mu \lambda_{\max} < 2$ в случае (5.125) или $2\mu \lambda_{\max} < 1$ в случае (5.126).

Тогда, начиная с $\mu=0$, функция (5.124) возрастает по мере увеличения μ , а $M \rightarrow \infty$ при $J(\mu) \rightarrow 1$. Таким образом, условие $J(\mu) < 1$ позволяет определить предельное значение шага сходимости μ_{\max} , меньшее которого LMS-алгоритм остается устойчивым.

Для нахождения параметра μ_{\max} необходимо решить уравнения

$$J(\mu) = \sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n} = 1 \quad (5.127)$$

и

$$J(\mu) = \sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n} = 1. \quad (5.128)$$

Решение этих уравнений относительно μ является сложной задачей. Тем не менее, можно оценить предельное значение μ , гарантирующее выполнения равенств (5.127) и (5.128).

Для этого рассмотрим уравнения

$$c = \frac{a}{1-a} + \frac{b}{1-b} \quad (5.129)$$

и

$$d = \frac{a+b}{1-(a+b)} = \frac{a}{1-(a+b)} + \frac{b}{1-(a+b)}. \quad (5.130)$$

Если $a+b < 1$, то очевидно, что $c < d$.

Представим уравнения (5.122) и (5.123) как

$$\begin{aligned} \sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n} &= \frac{\mu \lambda_1}{2 - \mu \lambda_1} + \frac{\mu \lambda_2}{2 - \mu \lambda_2} + \dots + \frac{\mu \lambda_N}{2 - \mu \lambda_N} = \\ &= \frac{2}{2} \left[\frac{0.5 \mu \lambda_1}{1 - 0.5 \mu \lambda_1} + \frac{0.5 \mu \lambda_2}{1 - 0.5 \mu \lambda_2} + \dots + \frac{0.5 \mu \lambda_N}{1 - 0.5 \mu \lambda_N} \right] = \frac{a}{1-a} + \frac{b}{1-b} + \dots + \frac{z}{1-z} \end{aligned} \quad (5.131)$$

и как

$$\begin{aligned} \sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n} &= \frac{\mu \lambda_1}{1 - 2\mu \lambda_1} + \frac{\mu \lambda_2}{1 - 2\mu \lambda_2} + \dots + \frac{\mu \lambda_N}{1 - 2\mu \lambda_N} = \\ &= \frac{1}{2} \left[\frac{2\mu \lambda_1}{1 - 2\mu \lambda_1} + \frac{2\mu \lambda_2}{1 - 2\mu \lambda_2} + \dots + \frac{2\mu \lambda_N}{1 - 2\mu \lambda_N} \right] = \frac{1}{2} \left[\frac{a}{1-a} + \frac{b}{1-b} + \dots + \frac{z}{1-z} \right]. \end{aligned} \quad (5.132)$$

Из соотношения между выражениями (5.129) и (5.130) следует, что

$$\sum_{n=1}^N \frac{\mu \lambda_n}{2 - \mu \lambda_n} < \frac{2}{2} \frac{\sum_{n=1}^N 0.5 \mu \lambda_n}{1 - \sum_{n=1}^N 0.5 \mu \lambda_n} = \frac{\mu \sum_{n=1}^N \lambda_n}{2 - \mu \sum_{n=1}^N \lambda_n} \quad (5.133)$$

и

$$\sum_{n=1}^N \frac{\mu \lambda_n}{1 - 2\mu \lambda_n} < \frac{1}{2} \frac{\sum_{n=1}^N 2\mu \lambda_n}{1 - \sum_{n=1}^N 2\mu \lambda_n} = \frac{\mu \sum_{n=1}^N \lambda_n}{1 - 2\mu \sum_{n=1}^N \lambda_n}. \quad (5.134)$$

Это позволяет вместо уравнений (5.127) и (5.128) использовать для нахождения μ_{\max} следующие из выражений (5.133) и (5.134) уравнения

$$\frac{\mu_{\max} \sum_{n=1}^N \lambda_n}{2 - \mu_{\max} \sum_{n=1}^N \lambda_n} = 1 \quad (5.135)$$

и

$$\frac{\mu_{\max} \sum_{n=1}^N \lambda_n}{1 - 2\mu_{\max} \sum_{n=1}^N \lambda_n} = 1. \quad (5.136)$$

Уравнения (5.135) и (5.136) можно представить как

$$\mu_{\max} \sum_{n=1}^N \lambda_n = 2 - \mu_{\max} \sum_{n=1}^N \lambda_n \quad (5.137)$$

и

$$\mu_{\max} \sum_{n=1}^N \lambda_n = 1 - 2\mu_{\max} \sum_{n=1}^N \lambda_n, \quad (5.138)$$

откуда следует, что с учетом свойства (3.53) для LMS-алгоритма адаптивного фильтра с комплексными ВК значение μ_{\max} определяется как

$$\mu_{\max} < \frac{1}{\sum_{n=1}^N \lambda_n} = \frac{1}{\text{tr}\{\mathbf{R}_N\}}, \quad (5.139)$$

а для LMS-алгоритма адаптивного фильтра с действительными ВК как

$$\mu_{\max} < \frac{1}{3 \sum_{n=1}^N \lambda_n} = \frac{1}{3 \text{tr}\{\mathbf{R}_N\}}. \quad (5.140)$$

Следовательно, аналогично выражениям (5.21), (5.23)–(5.25), абсолютные (т. е. достаточные) условия, гарантирующие устойчивую работу LMS-алгоритма для адаптивного фильтра с комплексными ВК, можно определить как

$$0 < \mu < \frac{1}{\text{tr}\{\mathbf{R}_N\}} \quad (5.141)$$

или как

$$0 < \mu < \frac{1}{N \sigma_x^2} \quad (5.142)$$

в случае трансверсального адаптивного фильтра.

Аналогично, условия, гарантирующие устойчивую работу LMS-алгоритма для адаптивного фильтра с действительными ВК, можно определить как

$$0 < \mu < \frac{1}{3 \text{tr}\{\mathbf{R}_N\}} \quad (5.143)$$

или как

$$0 < \mu < \frac{1}{3N \sigma_x^2} \quad (5.144)$$

в случае трансверсального адаптивного фильтра.

Сравнивая уравнения (5.21), (5.23)–(5.25) и (5.141)–(5.144), можно заметить, что предельное значение шага сходимости в LMS-алгоритме меньше значения, определяемого по аналогии с алгоритмом наискорейшего спуска. Это обуслов-

лено использованием в LMS-алгоритме не точного значения градиента целевой функции, а его оценок (5.3) или (5.4).

Необходимые условия устойчивости (т. е. еще более точные) можно найти из точного решения уравнений (5.127) и (5.128), если определить первые положительные корни этих уравнений. Однако, как уже отмечалось, это решение является сложным, что не способствует его использованию на практике.

Из уравнения (5.38) следует, что чем меньше значение параметра M , тем качественнее LMS-алгоритм осуществляет адаптивную фильтрацию. Данные выводы иллюстрируются ниже. На рис. 5.4 представлены обучающие кривые алгоритма наискорейшего спуска (темная кривая) и одной реализации $|\alpha(k)|^2$ (светлая кривая) LMS-алгоритма в условиях, аналогичных рис. 5.2.

Обучающая кривая $\xi(k) = E\{\alpha(k)\alpha^*(k)\}$, усредненная по 100 реализациям, приведена на рис. 5.5. Из рисунка видно, что $E\{|\alpha(k)|^2\}$ в среднем повторяет обучающую кривую алгоритма наискорейшего спуска.

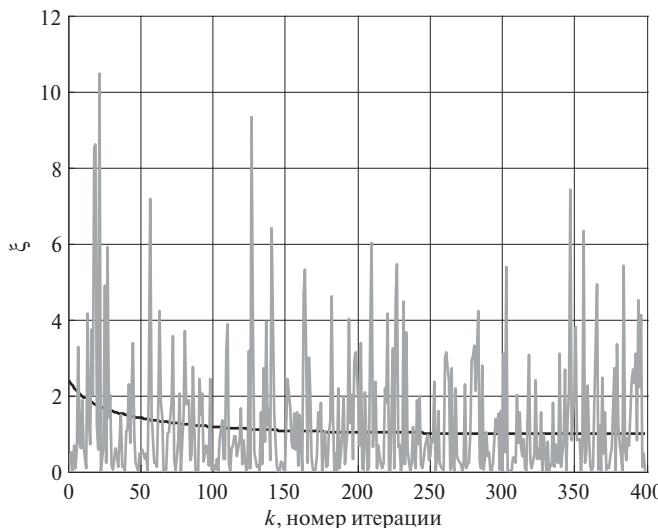


Рис. 5.4. Обучающие кривые при $\mu=0.005$: одна реализация

Усредненная по ансамблю L реализаций обучающая кривая определяется как

$$E\{|\alpha(k)|^2\} = \frac{1}{L} \sum_{l=1}^L |\alpha_l(k)|^2, \quad (5.145)$$

где $\alpha_l(k)$ — l -я реализация процесса $\alpha(k)$, полученная при l -й реализации входного процесса $x(k)$. Усредненная таким образом обучающая кривая является аппроксимацией действительной обучающей кривой $\xi(k)$ [30].

Кривая $E\{|\alpha(k)|^2\}$, усредненная по 100000 реализаций, приведена на рис. 5.6. Из этого рисунка следует, что $E\{|\alpha(k)|^2\}$ отличается от обучающей кривой ал-

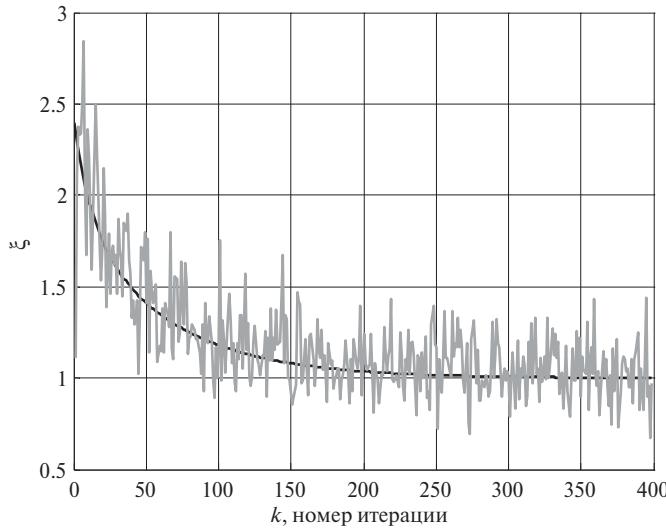


Рис. 5.5. Обучающие кривые при $\mu=0.005$: усреднение по 100 реализациям

горитма наискорейшего спуска на некоторую величину ξ_{exc} , которую согласно рис. 5.7 можно примерно оценить как 0.025. Согласно данным, приведенным в разделе 3.5, рис. 5.1—5.7 соответствуют эксперименту при $\lambda_1=0.8$, $\lambda_2=4$, $\xi_{\min}=1$. Так как в случае действительных ВК уравнение (5.121) имеет вид

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} \approx \frac{\mu \sum_{n=1}^N \lambda_n}{1 - \mu \sum_{n=1}^N \lambda_n}, \quad (5.146)$$

то, определив из этого уравнения ξ_{exc} как

$$\xi_{\text{exc}} = \xi_{\min} \frac{\mu \sum_{n=1}^N \lambda_n}{1 - \mu \sum_{n=1}^N \lambda_n}, \quad (5.147)$$

можно посчитать, что при $\mu=0.005$

$$\xi_{\text{exc}} = 1 \cdot \frac{0.005 \cdot (0.8 + 4)}{1 - 0.005 \cdot (0.8 + 4)} \approx 0.025,$$

а при $\mu=0.01$ значение $\xi_{\text{exc}} \approx 0.050$, что подтверждается путем моделирования (см. рис. 5.7 и рис. 5.9, т. е. участки рис. 5.6 и рис. 5.8 в увеличенном масштабе).

Совместные графики обучающих кривых LMS-алгоритма (рис. 5.6 и рис. 5.8) приведены на рис. 5.10. Из этого рисунка следует, что при увеличении μ уменьшается длительность переходного процесса, но увеличивается ξ_{exc} . Согласно выбранному соотношению между значениями параметра μ и приведенным ранее расчетам увеличение параметра μ в два раза увеличивает примерно в два раза

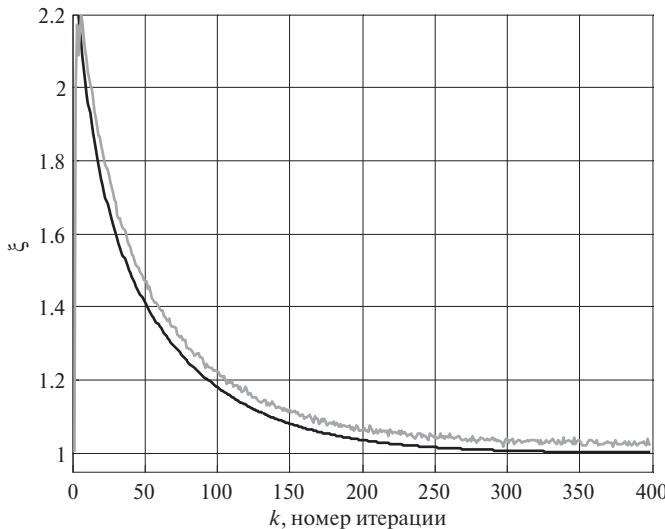


Рис. 5.6. Обучающие кривые при $\mu=0.005$: усреднение по 100000 реализациям

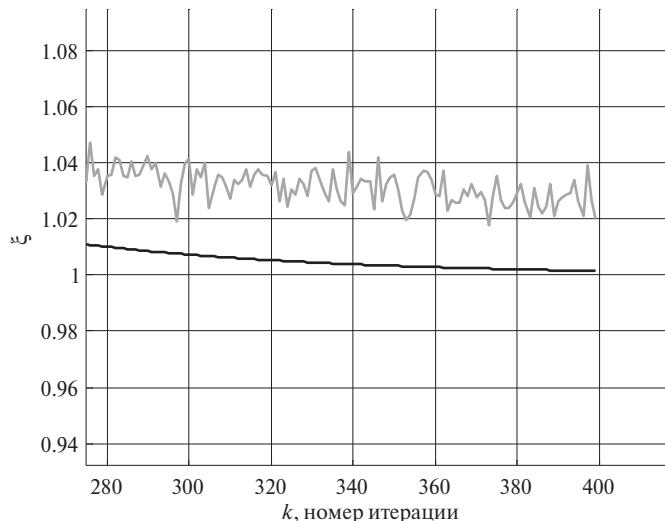


Рис. 5.7. Обучающие кривые (рис. 5.6) в увеличенном масштабе при $\mu=0.005$: усреднение по 100000 реализациям

и ξ_{exc} . Из рис. 5.10 видно, что в данном эксперименте при большем μ переходный процесс завершается примерно на 200-й итерации, а при меньшем — на 400-й. Этот результат также согласуется с соотношениями для постоянных времени переходного процесса обучающей кривой.

На рис. 5.11 показаны примеры траекторий движения по поверхности среднеквадратической ошибки вектора ВК $\tilde{\mathbf{h}}_N(k)$ в условиях, аналогичных рис. 5.4.

Как видно, сходимость алгоритма к $\tilde{\mathbf{h}}_{N,o}$ (обозначено светлым крестиком) обеспечивается при любых начальных значениях ВК $\tilde{\mathbf{h}}_N(0)$ в случае, если μ удовлетворяет условиям (5.143)–(5.144).

Предположение стационарности обрабатываемых сигналов $x(k)$ и $d(k)$, их статистической независимости между собой и от ВК является упрощением, позволяющим выполнять приведенный в настоящем разделе анализ. Несмотря

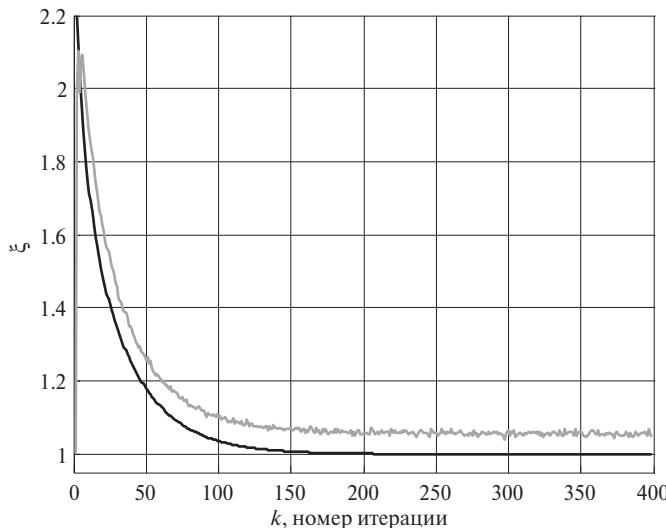


Рис. 5.8. Обучающие кривые при $\mu=0.01$: усреднение по 100000 реализациям

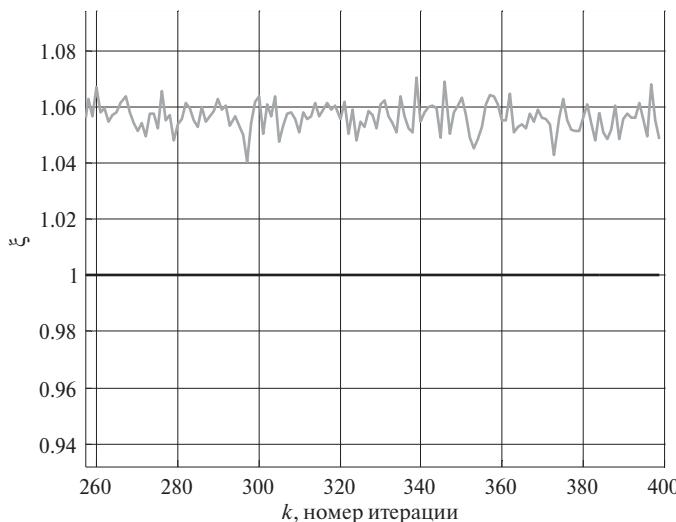


Рис. 5.9. Обучающие кривые (рис. 5.8) в увеличенном масштабе при $\mu=0.01$: усреднение по 100000 реализациям

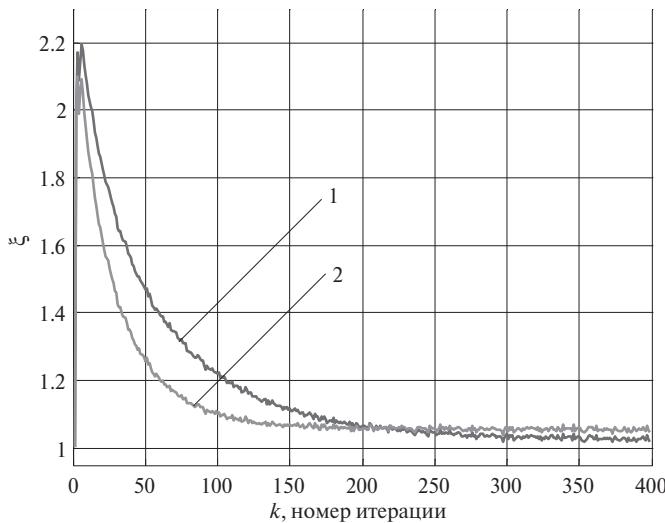


Рис. 5.10. LMS-алгоритм, обучающие кривые при усреднении по 100000 реализациям: 1 — $\mu=0.005$; 2 — $\mu=0.01$

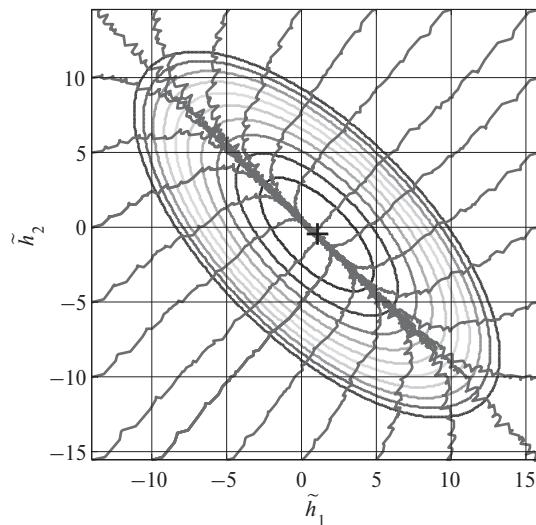


Рис. 5.11. LMS-алгоритм, поиск $\tilde{h}_{N,o}=[1, -0.5]^T$ (обозначено крестиком в центре рисунка) при $\mu=0.005$ и различных начальных значениях ВК $\tilde{h}_N(0)$

на это упрощение, теоретические результаты имеют достаточно хорошее совпадение на практике.

Более строгий анализ условий сходимости LMS-алгоритма в нестационарных условиях и при нарушении условия статистической независимости может быть найден в работах [3, 17, 20, 29].

5.5. NLMS-алгоритм

LMS-алгоритм используется в приложениях, где априори можно оценить предельно допустимое значение шага сходимости μ , гарнирующее устойчивость алгоритма. Как следует из выражений (5.142) и (5.144), такая оценка возможна, если известны статистические характеристики входного сигнала $x(k)$.

В приложениях, где эти характеристики неизвестны или меняются со временем (например, при обработке нестационарных сигналов), часто используется нормализованный LMS-алгоритм (Normalized LMS, NLMS) [78, 79], являющийся одной из разновидностей LMS-алгоритма.

В [20] NLMS-алгоритм получен в результате решения задачи линейно-ограниченной оптимизации. В [29] NLMS-алгоритм рассматривается как LMS-алгоритм с переменным шагом сходимости $\mu(k)$, в котором этот шаг определяется, исходя из условия минимизации $|\alpha(k)|^2$ с целью улучшения скорости сходимости алгоритма.

Подобно тому как LMS-алгоритм является результатом упрощения алгоритма наискорейшего спуска, NLMS-алгоритм можно рассматривать и как результат упрощения алгоритма Ньютона (4.2) [22, 30], в котором градиент определяется аналогично LMS-алгоритму, а в качестве матрицы \mathbf{R}_N используется ее регуляризованная оценка

$$\hat{\mathbf{R}}_N(k) = \delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k), \quad (5.148)$$

где $\delta^2 \geq 0.01\sigma_x^2$ [80] — параметр регуляризации, а σ_x^2 — дисперсия (средняя мощность) входного сигнала адаптивного фильтра. Регуляризация улучшает обращение матрицы $\mathbf{x}_N(k) \mathbf{x}_N^H(k)$, которая может быть плохо обусловленной из-за того, что в ее формировании используется только один отсчет вектора входных сигналов линейного сумматора адаптивного фильтра.

С учетом выражений (5.148) и (5.2) уравнение (4.2) преобразуется как

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) - \mu \hat{\mathbf{R}}_N^{-1} \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = \\ &= \mathbf{h}_N(k-1) + \hat{\mu} \left[\delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \mathbf{x}_N(k) \alpha^*(k). \end{aligned} \quad (5.149)$$

Для обращения матрицы в уравнении (5.149), т.е. матрицы (5.148) вида $\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D}$, необходимо воспользоваться MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где $\mathbf{C} = \delta^2 \mathbf{I}_N$, $\mathbf{C}^{-1} = \delta^{-2} \mathbf{I}_N$, $\mathbf{B}^H = \mathbf{x}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k)$ и $\mathbf{A}^{-1} = -1$.

Тогда

$$\begin{aligned} \left[\delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^{-1} &= \delta^{-2} \mathbf{I}_N - \frac{\delta^{-2} \mathbf{I}_N \mathbf{x}_N(k) \mathbf{x}_N^H(k) \delta^{-2} \mathbf{I}_N}{\mathbf{x}_N^H(k) \delta^{-2} \mathbf{I}_N \mathbf{x}_N(k) + 1} = \\ &= \delta^{-2} \mathbf{I}_N - \frac{\delta^{-2} \mathbf{x}_N(k) \mathbf{x}_N^H(k)}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2}. \end{aligned} \quad (5.150)$$

Произведение $\left[\delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \mathbf{x}_N(k)$ в уравнении (5.149) определяется как

$$\begin{aligned} \left[\delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \mathbf{x}_N(k) &= \left[\delta^{-2} \mathbf{I}_N - \frac{\delta^{-2} \mathbf{x}_N(k) \mathbf{x}_N^H(k)}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \right] \mathbf{x}_N(k) = \\ &= \frac{\delta^{-2} \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^{-2} \mathbf{x}_N(k) \delta^2 - \delta^{-2} \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{x}_N(k)}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} = \\ &= \frac{\mathbf{x}_N(k)}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2}. \end{aligned} \quad (5.151)$$

С учетом выражения (5.151) уравнение (5.149) принимает вид

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k), \quad (5.152)$$

где согласно уравнению (4.8) $0 < \hat{\mu} < 2$, см. также [81], — масштабирующий множитель динамического шага сходимости $\left[\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2 \right]^{-1}$.

В случае действительных ВК уравнение (5.152) принимает вид

$$\tilde{\mathbf{h}}_N(k) = \tilde{\mathbf{h}}_N + \frac{2\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k), \quad (5.153)$$

а $0 < \hat{\mu} < 1$, см. уравнение (4.9).

Таким образом, из уравнения (5.152) следует, что градиент в приращении вектора ВК в NLMS-алгоритме нормализуется к значению $\mathbf{x}_N^H(k) \mathbf{x}_N(k)$. Это послужило источником слова «нормализованный» в названии данного алгоритма.

Если в случае комплексных ВК $\hat{\mu} = 1$, то согласно [82] NLMS-алгоритм обеспечивает наибольшую скорость сходимости, хотя относительное среднее значение ошибки M несколько больше, чем при $0 < \hat{\mu} < 1$. И хотя максимально допустимое значение $\hat{\mu}$ ограничено величиной, равной 2, при $1 < \hat{\mu} < 2$ сходимость алгоритма также меньше, чем при $\hat{\mu} = 1$, а M имеет большее значение.

На рис. 5.12 показаны теоретические зависимости относительного среднего значения дополнительной ошибки M от $\hat{\mu}$ [83] в NLMS-алгоритме

$$M(\hat{\mu}) \approx \frac{\hat{\mu}(N+2)}{(2-\hat{\mu})(N-1)} \quad (5.154)$$

для случая, когда входной сигнал адаптивного фильтра на базе NLMS-алгоритма является белым гауссовым шумом, решается задача идентификации объекта, а порядок линейной модели которого известен и равен порядку адаптивного фильтра.

Из рис. 5.12 видно, что при увеличении $\hat{\mu}$ значение M также увеличивается. Кроме того, при больших N значение M практически не зависит от N , так как $(N+2)/(N-1) \rightarrow 1$.

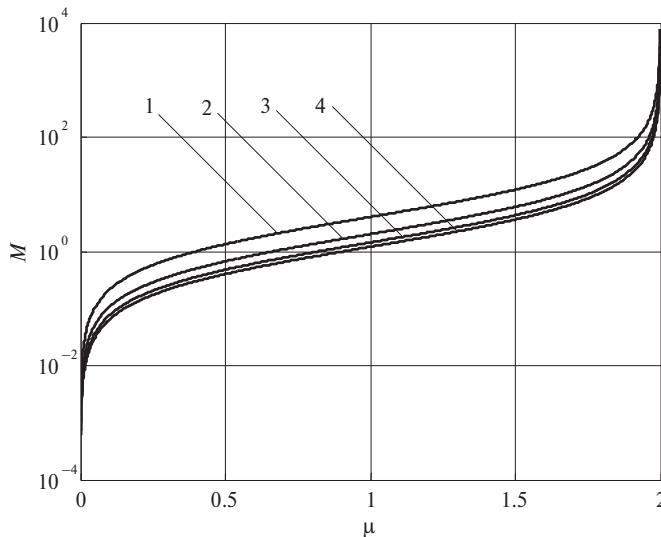


Рис. 5.12. Среднее значение ошибки M в NLMS-алгоритме: 1 — $N=2$; 2 — $N=4$; 3 — $N=8$; 4 — $N=16$; входной сигнал — белый гауссовой шум

Вычислительная процедура NLMS-алгоритма приведена в табл. 5.2.

Таблица 5.2. NLMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0)=\mathbf{0}_N$, $\mathbf{h}_N(0)=\mathbf{0}_N$	(5.2.0)
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$	(5.2.1)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\mathbf{x}_N^H(k)\mathbf{x}_N(k) + \delta} \mathbf{x}_N(k) \alpha^*(k)$	(5.2.2)
End for k	

Оценка вычислительной сложности алгоритма (см. табл. 5.2) равна $3N$ умножениям, $3N$ сложениям и одному делению на одну итерацию.

Если адаптивный фильтр представляет собой КИХ-фильтр (см. рис. 1.2), то вычислительную сложность NLMS-алгоритма можно уменьшить до $2N$ арифметических операций за счет рекурсивного вычисления произведения $\mathbf{x}_N^H(k)\mathbf{x}_N(k)$. Действительно, так как вектор $\mathbf{x}_N(k)$ представляет собой сдвинутые отсчеты входных сигналов, т. е.

$$\mathbf{x}_N(k-1) = [x(k-1), x(k-2), \dots, x(k-N)]^T \quad (5.155)$$

и

$$\mathbf{x}_N(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T, \quad (5.156)$$

то очевидно, что

$$\begin{aligned}
 \mathbf{x}_N^H(k) \mathbf{x}_N(k) &= \sum_{n=1}^N x^*(k-n+1) x(k-n+1) = \\
 &= x^*(k) x(k) + \sum_{n=1}^N x^*(k-n) x(k-n) - x^*(k-N) x(k-N) = \\
 &= x^*(k) x(k) + \mathbf{x}_N^H(k-1) \mathbf{x}_N(k-1) - x^*(k-N) x(k-N).
 \end{aligned} \tag{5.157}$$

Если в NLMS-алгоритме для КИХ-фильтра произвести масштабирование [30], то получится NLMS-алгоритм, в котором осуществляется нормализация к средней мощности

$$p(k) = \left[\mathbf{x}_N^H(k) \mathbf{x}_N(k) \right] / N \approx \sigma_x^2 \tag{5.158}$$

входного сигнала адаптивного фильтра. Уравнение для вычисления вектора ВК этого фильтра определяется как

$$\begin{aligned}
 \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \frac{\hat{\mu}/N}{\left[\mathbf{x}_N^H(k) \mathbf{x}_N(k) \right] / N + \tilde{\delta}^2 / N} \mathbf{x}_N(k) \alpha^*(k) = \\
 &= \mathbf{h}_N(k-1) + \frac{\tilde{\mu}}{p(k) + \tilde{\delta}^2} \mathbf{x}_N(k) \alpha^*(k),
 \end{aligned} \tag{5.159}$$

где $\tilde{\mu} = \hat{\mu}/N$, $\tilde{\delta}^2 = \delta^2/N$. Вычислительная сложность такого NLMS-алгоритма также равна $2N$ арифметических операций.

Значение средней мощности $p(k)$ входного сигнала адаптивного фильтра можно оценивать как

$$p(k) = \beta p(k-1) + (1-\beta) x(k) x^*(k) \tag{5.160}$$

при $p(0) = 0$, $0 < \beta < 1$ и $\beta \approx 1$.

Докажем, что при $k \rightarrow \infty$ значение $E\{p(k)\} = \sigma_x^2$. Действительно, запишем выражение $p(k)$ для нескольких значений k , т. е.

$$\begin{aligned}
 p(1) &= (1-\beta) x(1) x^*(1), \\
 p(2) &= \beta p(1) + (1-\beta) x(2) x^*(2) = \beta [(1-\beta) x(1) x^*(1)] + (1-\beta) x(2) x^*(2) = \\
 &= \beta (1-\beta) x(1) x^*(1) + (1-\beta) x(2) x^*(2), \\
 p(3) &= \beta p(2) + (1-\beta) x(3) x^*(3) = \\
 &= \beta [\beta (1-\beta) x(1) x^*(1) + (1-\beta) x(2) x^*(2)] + (1-\beta) x(3) x^*(3) = \\
 &= \beta^2 (1-\beta) x(1) x^*(1) + \beta (1-\beta) x(2) x^*(2) + (1-\beta) x(3) x^*(3).
 \end{aligned}$$

Из этих уравнений следует, что

$$p(k) = (1-\beta) \sum_{i=1}^k \beta^{k-i} x^*(k) x(k), \tag{5.161}$$

а $E\{p(k)\}$ определяется как

$$\begin{aligned}
E\{p(k)\} &= E\left[\left(1-\beta\right) \sum_{i=1}^k \beta^{k-i} x^*(k) x(k)\right] = \\
&= \left(1-\beta\right) \sum_{i=1}^k \beta^{k-i} \sigma_x^2 = \left(1-\beta\right) \sigma_x^2 \sum_{i=1}^k \beta^{k-i} = \left(1-\beta\right) \sigma_x^2 \frac{1-\beta^{k-1} \beta}{1-\beta} = \sigma_x^2 \left(1-\beta^k\right) \approx \sigma_x^2.
\end{aligned} \tag{5.162}$$

То есть, среднее значение $p(k)$ равно дисперсии или средней мощности входного сигнала адаптивного фильтра. Здесь использована формула суммы членов геометрической прогрессии (4.5) $s_k = (a_1 - a_k q) / (1 - q)$, где первый член геометрической прогрессии $a_1 = \beta^0 = 1$, а ее k -й член определяется как $a_k = \beta^{k-1}$, $q = \beta$ — знаменатель геометрической прогрессии и $a_k \rightarrow \infty$ при $k \rightarrow \infty$, так как $\beta < 1$.

При обработке стационарных сигналов LMS- и NLMS-алгоритмы демонстрируют примерно одинаковое поведение, если параметры μ и $\hat{\mu}$ выбраны так, что, с одной стороны, $0 < \mu < \alpha / \sum_{n=1}^N R_{nn}$, см. уравнение (5.21), (5.141), где $\sum_{n=1}^N R_{nn} \approx \mathbf{x}_N^H(k) \mathbf{x}_N(k)$ и, с другой стороны, $\mu \approx \hat{\mu} [\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2]^{-1}$, см. уравнение (5.152), т. е. $\alpha \approx \hat{\mu}$.

В случае обработки нестационарных сигналов вектор $\mathbf{x}_N(k)$ может сильно меняться от итерации к итерации, т. е. может меняться норма этого вектора

$$\|\mathbf{x}_N(k)\|_2 = \sqrt{\sum_{n=1}^N x^*(k-n+1) x(k-n+1)}, \tag{5.163}$$

а значит, может меняться градиент $\mathbf{x}_N(k) \alpha^*(k)$ и приращение вектора ВК

$$\Delta \mathbf{h}_N(k) = \mu \mathbf{x}_N(k) \alpha^*(k) \tag{5.164}$$

в LMS-алгоритме (5.5). В NLMS-алгоритме это приращение определяется как

$$\Delta \mathbf{h}_N(k) = \frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k) = \frac{\hat{\mu}}{\|\mathbf{x}_N(k)\|_2^2 + \delta^2} \mathbf{x}_N(k) \alpha^*(k), \tag{5.165}$$

т. е. оно обратно пропорционально $\|\mathbf{x}_N(k)\|_2^2$. Нормирование градиента к $\|\mathbf{x}_N(k)\|_2^2$ «сглаживает» резкие его изменения из-за изменения $\mathbf{x}_N(k)$. Поэтому при фиксированных значениях параметров μ и $\hat{\mu}$ поведение NLMS-алгоритма при обработке нестационарных сигналов является более устойчивым, чем поведение LMS-алгоритма.

Таким образом, множитель

$$\mu(k) = \frac{\hat{\mu}}{\|\mathbf{x}_N(k)\|_2^2 + \delta^2} \tag{5.166}$$

в NLMS-алгоритме (5.152) представляет собой разновидность переменного шага сходимости, величина которого зависит от времени и значений вектора вход-

ных сигналов адаптивного фильтра $\mathbf{x}_N(k)$, так как определяется нормой этого вектора $\|\mathbf{x}_N(k)\|_2$.

5.6. Градиентные алгоритмы с переменным шагом сходимости

Идея переменного шага сходимости широко используется для дополнительного улучшения быстродействия градиентных алгоритмов адаптивной фильтрации [84–88]. В настоящем разделе рассматривается применение градиентного шага сходимости [86] в LMS- и NLMS-алгоритмах.

LMS-алгоритм с переменным шагом сходимости имеет вид

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu(k) \mathbf{x}_N(k) \alpha^*(k). \quad (5.167)$$

Используя уравнение (5.167), выразим ошибку $\alpha(k)$ как

$$\begin{aligned} \alpha(k) &= d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \\ &= d(k) - [\mathbf{h}_N^H(k-2) + \mu(k-1) \alpha(k-1) \mathbf{x}_N^H(k-1)] \mathbf{x}_N(k) = \\ &= d(k) - \mathbf{h}_N^H(k-2) \mathbf{x}_N(k) - \mu(k-1) \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k). \end{aligned} \quad (5.168)$$

Чтобы для выражения (5.167) получить уравнение поиска параметра $\mu(k)$, минимизирующего $|\alpha(k)|^2 = \alpha(k) \alpha^*(k)$, требуется определить одномерный градиент (производную)

$$\begin{aligned} \nabla_\mu(k) &= \frac{\partial [\alpha(k) \alpha^*(k)]}{\partial \mu(k-1)} = \frac{\partial \alpha(k)}{\partial \mu(k-1)} \alpha^*(k) + \alpha(k) \frac{\partial \alpha^*(k)}{\partial \mu(k-1)} = \\ &= \frac{\partial [\dots - \mu(k-1) \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k)]}{\partial \mu(k-1)} \alpha^*(k) + \\ &\quad + \alpha(k) \frac{\partial [\dots - \mu(k-1) \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k)]^*}{\partial \mu(k-1)} = \\ &= -\alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \alpha^*(k) - \alpha(k) [\mathbf{x}_N^H(k-1) \mathbf{x}_N(k)]^* \alpha^*(k-1) = \\ &= -2 \operatorname{Re} \{ \alpha^*(k) \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \}. \end{aligned} \quad (5.169)$$

В уравнении (5.169) троеточиями обозначены компоненты, не зависящие от μ .

С учетом выражения (5.169) уравнение переменного шага сходимости, изменяемого по градиентному закону, для LMS-алгоритма можно записать как

$$\mu(k) = \mu(k-1) - \rho \nabla_\mu(k) = \mu(k-1) + 2\rho \operatorname{Re} \{ \alpha^*(k) \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \}, \quad (5.170)$$

где $\rho \leq 1$ — шаг сходимости уравнения (5.170).

Уравнение NLMS-алгоритма (5.152) также может быть модифицировано для случая переменного параметра $\hat{\mu}(k)$ как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \frac{\hat{\mu}(k)}{\mathbf{x}_N^H(k)\mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k). \quad (5.171)$$

Выполнив преобразования, аналогичные (5.168) и (5.169), получим, что

$$\begin{aligned} \alpha(k) &= d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = d(k) - \\ &- \left[\mathbf{h}_N^H(k-2) + \frac{\hat{\mu}(k-1)}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} \alpha(k-1) \mathbf{x}_N^H(k-1) \right] \mathbf{x}_N(k) = \\ &= d(k) - \mathbf{h}_N^H(k-2) \mathbf{x}_N(k) - \frac{\hat{\mu}(k-1)}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \end{aligned} \quad (5.172)$$

и

$$\begin{aligned} \nabla_{\hat{\mu}}(k) &= \frac{\partial [\alpha(k) \alpha^*(k)]}{\partial \hat{\mu}(k-1)} = \frac{\partial \alpha(k)}{\partial \hat{\mu}(k-1)} \alpha(k)^* + \alpha(k) \frac{\partial \alpha^*(k)}{\partial \hat{\mu}(k-1)} = \\ &= \frac{\partial \left[\dots - \frac{\hat{\mu}(k-1)}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \right]}{\partial \hat{\mu}(k-1)} \alpha^*(k) + \\ &+ \alpha(k) \frac{\partial \left[\dots - \frac{\hat{\mu}(k-1)}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \right]^*}{\partial \hat{\mu}(k-1)} = \\ &= -\frac{1}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} \alpha(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \alpha^*(k) - \\ &- \alpha(k) \frac{1}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2} [\mathbf{x}_N^H(k-1) \mathbf{x}_N(k)]^* \alpha^*(k-1) = \\ &= -2 \operatorname{Re} \{ \alpha^*(k) \tilde{\alpha}(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \}, \end{aligned} \quad (5.173)$$

где

$$\tilde{\alpha}(k-1) = \frac{\alpha(k-1)}{\mathbf{x}_N^H(k-1)\mathbf{x}_N(k-1) + \delta^2}. \quad (5.174)$$

В уравнении (5.173) троеточиями обозначены компоненты, не зависящие от $\hat{\mu}$.

Тогда уравнение вычисления переменного шага сходимости, изменяемого по градиентному закону, для NLMS-алгоритма можно записать как

$$\hat{\mu}(k) = \hat{\mu}(k-1) - \rho \nabla_{\hat{\mu}}(k) = \hat{\mu}(k-1) + 2\rho \operatorname{Re} \{ \alpha^*(k) \tilde{\alpha}(k-1) \mathbf{x}_N^H(k-1) \mathbf{x}_N(k) \}, \quad (5.175)$$

где $\rho \leq 1$ — шаг сходимости.

Выбор параметра ρ в выражениях (5.170) и (5.175) является неоднозначной задачей, поскольку не существует критерия для получения его предельных зна-

чений. Однако известно, что для LMS-алгоритма этот параметр должен быть таким, чтобы $\mu(k)$ удовлетворяло соотношению (5.141) или (5.143), а для NLMS-алгоритма — соотношению (4.8) или (4.9).

Так как параметры μ и $\hat{\mu}$ — это положительные действительные числа, не равные нулю, то, исходя из физических соображений, на каждом шаге алгоритмов (5.170) и (5.175) необходимо ограничивать значения $\mu(k)$ и $\hat{\mu}(k)$ как $\mu_{\min} \leq \mu(k) \leq \mu_{\max}$ и $\mu_{\min} \leq \hat{\mu}(k) \leq \mu_{\max}$, где $\mu_{\min} > 0$ выбирается из соображения минимальнодопустимой скорости сходимости LMS- и NLMS-алгоритмов, а $\mu_{\max} > 0$ — из соотношений (5.141), (5.143), (4.8), (4.9).

5.7. Линейно-ограниченные LMS- и NLMS-алгоритмы

В разделе 3.7 было представлено решение LC задачи винеровской фильтрации. Аналогичным образом можно получить уравнение LC LMS-алгоритма. Для этого выполним процедуры, во многом совпадающие с процедурами, рассмотренными в разделах 3.7 и 5.2. Определим линейные ограничения вида

$$\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \quad (5.176)$$

и модифицированную с помощью множителей Лагранжа целевую функцию

$$\begin{aligned} \xi_c(k) = & d(k)d^*(k) - \mathbf{h}_N^H(k)\hat{\mathbf{r}}_N(k) - \hat{\mathbf{r}}_N^H(k)\mathbf{h}_N(k) + \\ & + \mathbf{h}_N^H(k)\hat{\mathbf{R}}_N(k)\hat{\mathbf{h}}_N(k) + \boldsymbol{\lambda}_J^H(k)[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k)] + [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k)]^H \boldsymbol{\lambda}_J(k). \end{aligned} \quad (5.177)$$

Значение функции (5.177) минимально при

$$\begin{aligned} \hat{\nabla}_{\mathbf{h}_N^*} \xi_c(k) = & \frac{\partial \xi_c(k)}{\partial \mathbf{h}_N^*(k)} = -\hat{\mathbf{r}}_N(k) + \hat{\mathbf{R}}_N(k)\mathbf{h}_N(k) - \mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k) = \\ = & \nabla_{\mathbf{h}_N^*} \xi(k) - \mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k) = \mathbf{0}_N, \end{aligned} \quad (5.178)$$

где $\hat{\nabla}_{\mathbf{h}_N^*} \xi_c(k)$ — оценка градиента задачи линейно-ограниченной оптимизации, $\hat{\nabla}_{\mathbf{h}_N^*} \xi(k) = -\hat{\mathbf{r}}_N(k) + \hat{\mathbf{R}}_N(k)\mathbf{h}_N(k)$ — оценка градиента задачи оптимизации без ограничений, а $\hat{\mathbf{R}}_N(k) = \mathbf{x}_N(k)\mathbf{x}_N^H(k)$ и $\hat{\mathbf{r}}_N = \mathbf{x}_N(k)d^*(k)$, см. раздел 5.1. В случае действительных ВК

$$\begin{aligned} \hat{\nabla}_{\tilde{\mathbf{h}}_N} \xi_c(k) = & \frac{\partial \xi_c(k)}{\partial \tilde{\mathbf{h}}_N(k)} = -2\hat{\mathbf{r}}_N(k) + 2\hat{\mathbf{R}}_N(k)\tilde{\mathbf{h}}_N(k) - 2\mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k) = \\ = & \nabla_{\tilde{\mathbf{h}}_N} \xi(k) - 2\mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k) = \mathbf{0}_N. \end{aligned} \quad (5.179)$$

Используем выражение градиента (5.178) в уравнении LMS-алгоритма (5.5)

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \left[-\hat{\nabla}_{\mathbf{h}_N^*} \xi_c(k-1) \right] = \mathbf{h}_N(k-1) - \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) + \mu \mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k-1). \quad (5.180)$$

Уравнение (5.180) должно удовлетворять ограничению (5.176), т. е.

$$\begin{aligned} \mathbf{C}_{NJ}^H \mathbf{h}_N(k) &= \mathbf{C}_{NJ}^H \left[\mathbf{h}_N(k-1) - \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) + \mu \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k-1) \right] = \\ &= \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1) - \mu \mathbf{C}_{NJ}^H \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) + \mu \mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k-1) = \mathbf{f}_J. \end{aligned} \quad (5.181)$$

Из уравнения (5.181) можно определить $\boldsymbol{\lambda}_J(k-1)$ как

$$\boldsymbol{\lambda}_J(k-1) = \mu^{-1} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1) \right] + \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{C}_{NJ}^H \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) \right]. \quad (5.182)$$

С учетом (5.182), уравнение (5.180) можно преобразовать как

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) - \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) + \mu \mathbf{C}_{NJ} \left\{ \mu^{-1} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1) \right] + \right. \\ &+ \left. \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) \right\} = \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \right\} \mathbf{h}_N(k-1) - \\ &- \mu \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \right\} \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) + \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J = \\ &= \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \right\} \left[\mathbf{h}_N(k-1) + \mu \hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) \right] + \\ &+ \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J = \mathbf{P}_N \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J. \end{aligned} \quad (5.183)$$

Из-за умножения на квадратную матрицу \mathbf{P}_N правой части уравнения (5.183) вычислительная сложность алгоритма LC LMS-алгоритма пропорциональна $O(N^2)$ арифметическим операциям. В уравнении (5.183) матрица

$$\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \quad (5.184)$$

— проекционная матрица.

Вычислительная процедура LC LMS-алгоритма приведена в табл. 5.3.

Таблица 5.3. LC LMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J$,	(5.3.0)
$\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H$, $\hat{\mathbf{f}}_N = \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J$	
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(5.3.1)
$\mathbf{h}_N(k) = \mathbf{P}_N \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \hat{\mathbf{f}}_N$	(5.3.2)
End for k	

Из уравнения (5.183) следует, что начальное значение вектора $\mathbf{h}_N(0)$ определяется как

$$\mathbf{h}_N(0) = \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J. \quad (5.185)$$

Сложность алгоритма (см. табл. 5.3) можно уменьшить до $O(N)$ арифметических операций, если уравнение (5.183) преобразовать как

$$\begin{aligned} \mathbf{h}_N(k) &= \left[\mathbf{I}_N - \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{C}_{NJ}^H \right] \times \\ &\times \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J = \\ &= \left[\mathbf{I}_N - \mathbf{Q}_{NJ} \mathbf{C}_{NJ}^H \right] \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \mathbf{Q}_{NJ} \mathbf{f}_J = \\ &= \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^* + \mathbf{Q}_{NJ} \left\{ \mathbf{f}_J - \mathbf{C}_{NJ}^H \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] \right\} = \\ &= \mathbf{h}'_N(k) + \mathbf{Q}_{NJ} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k) \right], \end{aligned} \quad (5.186)$$

где

$$\mathbf{Q}_{NJ} = \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \quad (5.187)$$

и

$$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k). \quad (5.188)$$

С учетом (5.39), LC LMS-алгоритм принимает вид, представленный в табл. 5.4.

Таблица 5.4. LC LMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{Q}_{NJ} = \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1}$, $\mathbf{h}_N(0) = \mathbf{Q}_{NJ} \mathbf{f}_J$	(5.4.0)
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(5.4.1)
$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k)$	(5.4.2)
$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ} \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k) \right]$	(5.4.3)
End for k	

Из табл. 5.4 следует, что вычислительная сложность LC LMS-алгоритма равна $2N + 2NJ$ операциям умножения и $3N + 3NJ$ операциям сложения.

При $d(k) = 0$ алгоритм (см. табл. 5.3) эквивалентен LC LMS-алгоритму Фроста (Frost) [58, 59].

Значение шага сходимости μ , обеспечивающее устойчивое функционирование LC LMS-алгоритма для адаптивного фильтра с комплексными ВК, определяется как

$$0 < \mu < \frac{1}{\sum_{n=1}^N R_{nn}} = \frac{1}{\text{tr}\{\mathbf{R}_N\}}, \quad (5.189)$$

а для адаптивного фильтра с действительными ВК — как

$$0 < \mu < \frac{1}{3 \sum_{n=1}^N R_{nn}} = \frac{1}{3 \text{tr}\{\mathbf{R}_N\}}, \quad (5.190)$$

т. е. условия (5.189) и (5.190) аналогичны условиям (5.141) и (5.143) LMS-алгоритма без ограничений.

Геометрическая интерпретация влияния ограничений на процесс адаптации приведена ниже. Из выражений (5.176) и (5.183) следует, что

$$\begin{aligned} \mathbf{C}_{NJ}^H \mathbf{h}_N(k) &= \mathbf{C}_{NJ}^H \left[\mathbf{P}_N \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \mathbf{C}_{NJ} \left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^{-1} \mathbf{f}_J \right] = \\ &= \mathbf{C}_{NJ}^H \mathbf{P}_N \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] + \mathbf{f}_J = \mathbf{f}_J, \end{aligned} \quad (5.191)$$

если только

$$\mathbf{C}_{NJ}^H \mathbf{P}_N \left[\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k) \right] = \mathbf{C}_{NJ}^H \mathbf{P}_N \mathbf{h}'_N(k) = \mathbf{C}_{NJ}^H \mathbf{h}''_N(k) = \mathbf{0}_J. \quad (5.192)$$

Таким образом, матрица \mathbf{P}_N проецирует вектор $\mathbf{h}'_N(k)$ в плоскость, ортогональную гиперплоскости, образованной столбцами матрицы $\mathbf{C}_{NJ} = [\mathbf{c}_{N,1}, \mathbf{c}_{N,2}, \dots, \mathbf{c}_{N,J}]$, см. раздел 3.7. Поэтому эта матрица называется проекционной.

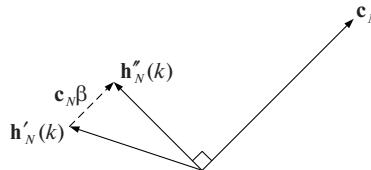


Рис. 5.13. Проецирование вектора: $\mathbf{C}_{NJ} = \mathbf{c}_N$

Действительно, из рис. 5.13 следует, что проецирование вектора $\mathbf{h}'_N(k)$ на вектор $\mathbf{h}''_N(k)$, ортогональный вектору \mathbf{c}_N , можно осуществить путем добавления к $\mathbf{h}'_N(k)$ некоторого корректирующего вектора, параллельного \mathbf{c}_N [6]. Этот вектор должен совпадать по направлению с вектором \mathbf{c}_N . Поэтому его можно определить как $\mathbf{c}_N \beta$, где β — масштабный множитель. Тогда

$$\mathbf{h}''_N(k) = \mathbf{h}'_N(k) + \mathbf{c}_N \beta. \quad (5.193)$$

Если требуется, чтобы вектор $\mathbf{h}''_N(k)$ был ортогонален одновременно нескольким векторам $\mathbf{c}_{N,1}, \mathbf{c}_{N,2}, \dots, \mathbf{c}_{N,J}$, то корректирующий вектор можно определить как $\mathbf{C}_{NJ} \beta_J$, где $\beta_J = [\beta_1, \beta_2, \dots, \beta_J]^T$ (рис. 5.14).

Таким образом, в случае, показанном на рис. 5.14,

$$\mathbf{h}''_N(k) = \mathbf{h}'_N(k) + \mathbf{C}_{NJ} \beta_J. \quad (5.194)$$

Так как в соответствии с рис. 5.14

$$\mathbf{C}_{NJ}^H \mathbf{h}''_N(k) = \mathbf{0}_J, \quad (5.195)$$

то из этого условия можно найти вектор β_J , т. е.

$$\mathbf{C}_{NJ}^H \mathbf{h}_N''(k) = \mathbf{C}_{NJ}^H [\mathbf{h}'_N(k) + \mathbf{C}_{NJ} \beta_J] = \mathbf{C}_{NJ}^H \mathbf{h}'_N(k) + \mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \beta_J = \mathbf{0}_J. \quad (5.196)$$

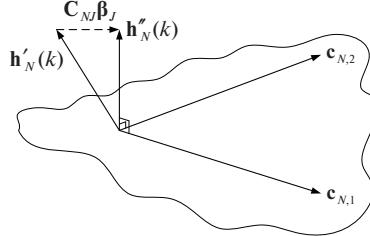


Рис. 5.14. Проецирование вектора: $\mathbf{C}_{NJ} = [\mathbf{c}_{N,1}, \mathbf{c}_{N,2}]$

Из (5.196) следует, что

$$\beta_J = -[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \mathbf{h}'_N(k). \quad (5.197)$$

С учетом (5.197)

$$\begin{aligned} \mathbf{h}''_N(k) &= \mathbf{h}'_N(k) + \mathbf{C}_{NJ} \beta_J = \mathbf{h}'_N(k) - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \mathbf{h}'_N(k) = \\ &= \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\} \mathbf{h}'_N(k) = \mathbf{P}_N \mathbf{h}'_N(k). \end{aligned} \quad (5.198)$$

Следовательно, выражение для проекционной матрицы в уравнении (5.198) совпадает с выражением (5.184).

Геометрическая интерпретация операций в процедуре вычисления вектора ВК в LC LMS-алгоритме [58] показана на рис. 5.15.

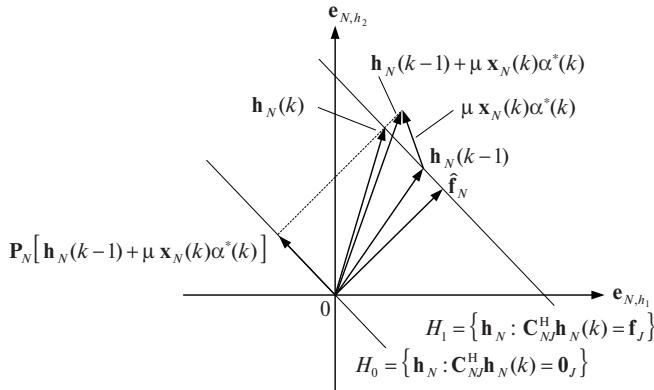


Рис. 5.15. Геометрическая интерпретация операций в LC LMS-алгоритме

Вычисленный на предыдущей итерации алгоритма вектор $\mathbf{h}_N(k-1)$ упирается в плоскость ограничений $H_1 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \}$, так как в задачах линейно-ограниченной оптимизации (см. рис. 3.15) ВК адаптивного фильтра

принимают лишь значения, лежащие на линии, проходящей через плоскость ограничений. Прибавление согласно процедуре (5.183) к этому вектору приращения $-\mu \nabla_{\mathbf{h}_N} \xi = \mu \mathbf{x}_N(k) \alpha^*(k)$, где $\nabla_{\mathbf{h}_N} \xi = -\mathbf{x}_N(k) \alpha^*(k)$ — градиент целевой функции в отсутствии ограничений, смешает конец результирующего вектора $\mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k)$ с плоскости ограничений. Однако, будучи спроектированным с помощью проекционной матрицы \mathbf{P}_N (5.184) на плоскость $\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{0}_J$ и сложенным с вектором $\hat{\mathbf{f}}_N = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \mathbf{h}_N(0)$, результирующий вектор $\mathbf{h}_N(k)$ снова упирается в плоскость ограничений $H_1 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \}$, т. е. удовлетворяет условию ограничений (5.176).

Так как вектор $\hat{\mathbf{f}}_N$ является линейной комбинацией столбцов матрицы \mathbf{C}_{NJ} , то это значит (см. рис. 5.14), что он лежит в плоскости, охваченной этими столбцами. Следовательно, данная плоскость и любой вектор, лежащий в ней, в том числе и вектор $\hat{\mathbf{f}}_N$, ортогональны векторам, спроектированным с помощью матрицы \mathbf{P}_N .

Вектор $\hat{\mathbf{f}}_N$ является кратчайшим вектором от начала координат и плоскости $H_0 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{0}_J \}$ до плоскости $H_1 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \}$, так как проекция вектора из плоскости $H_1 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \}$ на плоскость $H_0 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{0}_J \}$ равна нуль-вектору. Действительно,

$$\begin{aligned} \mathbf{P}_N \hat{\mathbf{f}}_N &= \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\} \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \\ &= \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \mathbf{0}_J, \end{aligned} \quad (5.199)$$

что означает ортогональность вектора $\hat{\mathbf{f}}_N$ и плоскостей $H_1 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J \}$ и $H_0 = \{ \mathbf{h}_N : \mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{0}_J \}$, а также коллинеарность этих плоскостей друг другу.

LC LMS-алгоритм обладает замечательным свойством устранения накопления ошибок в процедуре вычисления вектора ВК (5.183). Если в результате накопленных ошибок вектор $\mathbf{h}_N(k-1)$ выходит за пределы плоскости огранич-

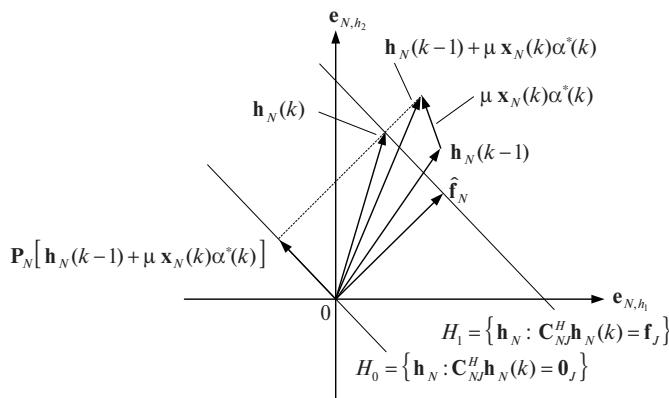


Рис. 5.16. Геометрическая интерпретация операций в LC LMS-алгоритме при наличии ошибки при вычислении вектора $\mathbf{h}_N(k-1)$

ний, то на следующей итерации k процедура проецирования возвращает новый вектор $\mathbf{h}_N(k)$ на плоскость ограничений, как это показано на рис. 5.16.

Проекционная матрица \mathbf{P}_N является идемпотентной, т. е.

$$\begin{aligned}
 \mathbf{P}_N^H \mathbf{P}_N &= \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\} = \\
 &= \mathbf{I}_N - \left\{ \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H + \\
 &\quad + \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-H} \mathbf{C}_{NJ}^H \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H = \\
 &= \mathbf{I}_N - \left\{ \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H + \\
 &\quad + \left\{ \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H = \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H = \mathbf{P}_N.
 \end{aligned} \tag{5.200}$$

Справедливо также соотношение $\mathbf{P}_N^H = \mathbf{P}_N$, так как

$$\begin{aligned}
 \left\{ \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H &= \mathbf{I}_N^H - \left\{ \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H \right\}^H = \\
 &= \mathbf{I}_N - (\mathbf{C}_{NJ}^H)^H \left\{ \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \right\}^H = \mathbf{I}_N - \mathbf{C}_{NJ} \left\{ [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \right\}^H (\mathbf{C}_{NJ})^H = \\
 &= \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-H} \mathbf{C}_{NJ}^H = \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H,
 \end{aligned} \tag{5.201}$$

поскольку $[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-H} = [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1}$, см. свойство 1 корреляционной матрицы (раздел 3.2). Это равенство справедливо, поскольку матрица $\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}$ является эрмитовой, а значит, $[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1}$ — также эрмитова матрица. Действительно,

$$\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} = \begin{bmatrix} \mathbf{c}_{N,1}^H \\ \mathbf{c}_{N,2}^H \\ \vdots \\ \mathbf{c}_{N,J}^H \end{bmatrix} \begin{bmatrix} \mathbf{c}_{N,1}, \mathbf{c}_{N,2}, \dots, \mathbf{c}_{N,J} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{N,1}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,1}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,1}^H \mathbf{c}_{N,J} \\ \mathbf{c}_{N,2}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,2}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,2}^H \mathbf{c}_{N,J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{N,J}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,J}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,J}^H \mathbf{c}_{N,J} \end{bmatrix}, \tag{5.202}$$

а значит,

$$\left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^H = \begin{bmatrix} \mathbf{c}_{N,1}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,1}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,1}^H \mathbf{c}_{N,J} \\ \mathbf{c}_{N,2}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,2}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,2}^H \mathbf{c}_{N,J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{N,J}^H \mathbf{c}_{N,1} & \mathbf{c}_{N,J}^H \mathbf{c}_{N,2} & \dots & \mathbf{c}_{N,J}^H \mathbf{c}_{N,J} \end{bmatrix}, \tag{5.203}$$

т. е. $\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}$ — эрмитова матрица, так как $\left[\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right]^H = \mathbf{C}_{NJ}^H \mathbf{C}_{NJ}$.

Свойство (5.201) используется при получении LC NLMS-алгоритма [89], приведенного в табл. 5.5.

Из-за умножения на квадратную матрицу \mathbf{P}_N в уравнении (5.5.2) вычислительная сложность алгоритма (см. табл. 5.5) подобно LC LMS-алгоритму (см. табл. 5.3) пропорциональна $O(N^2)$ арифметическим операциям.

Таблица 5.5. LC NLMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J$,	(5.5.0)
$\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H$, $\hat{\mathbf{f}}_N = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J$	
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(5.5.1)
$\mathbf{h}_N(k) = \mathbf{P}_N \left[\mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{P}_N \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k) \right] + \hat{\mathbf{f}}_N$	(5.5.2)
End for k	

Вычислительная сложность алгоритма (см. табл. 5.5) может быть уменьшена подобно LC LMS-алгоритму (см. табл. 5.4). Кроме того, выражение $[\mathbf{x}_N^H(k) \mathbf{P}_N \mathbf{x}_N(k) + \delta^2]^{-1}$ в уравнении (5.5.2) также можно упростить как

$$\begin{aligned} \mathbf{x}_N^H(k) \mathbf{P}_N \mathbf{x}_N(k) &= \mathbf{x}_N^H(k) [\mathbf{I}_N - \mathbf{Q}_{NJ} \mathbf{C}_{NJ}^H] \mathbf{x}_N(k) = \\ &= \mathbf{x}_N^H(k) \mathbf{x}_N(k) + [\mathbf{x}_N^H(k) \mathbf{Q}_{NJ}] [\mathbf{C}_{NJ}^H \mathbf{x}_N(k)]. \end{aligned} \quad (5.204)$$

Полученный таким образом LC LMS-алгоритм приведен в табл. 5.6.

Таблица 5.6. LC NLMS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{Q}_{NJ} = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1}$, $\mathbf{h}_N(0) = \mathbf{Q}_{NJ} \mathbf{f}_J$	(5.6.0)
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(5.6.1)
$\tilde{\alpha}(k) = \alpha(k) [\mathbf{x}_N^H(k) \mathbf{x}_N(k) + [\mathbf{x}_N^H(k) \mathbf{Q}_{NJ}] [\mathbf{C}_{NJ}^H \mathbf{x}_N(k)] + \delta^2]^{-1}$	
$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \hat{\mu} \mathbf{x}_N(k) \tilde{\alpha}^*(k)$	(5.6.2)
$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k)]$	(5.6.3)
End for k	

Из табл. 5.6 следует, что вычислительная сложность LC NLMS-алгоритма равна $3N + 4NJ$ операциям умножения, $4N + 5NJ$ операциям сложения и одной операции деления.

При использовании алгоритмов (см. табл. 5.1—5.6) для обработки действительных сигналов в вычислениях необходимо исключить операции комплексного сопряжения скалярных переменных, операции эрмитова сопряжения векторов и матриц заменить на операции транспонирования и в уравнениях вычисления ВК адаптивного фильтра вместо μ и $\hat{\mu}$ использовать 2μ и $2\hat{\mu}$, соответственно.



5.8. Выводы по главе

В главе 5 рассмотрены LMS-, NLMS-, LC LMS- и LC NLMS-алгоритмы поиска ВК адаптивного линейного сумматора (адаптивного фильтра). Приведены оценки предельнодопустимых значений шагов сходимости этих алгоритмов, а также рассмотрен способ изменения шагов сходимости по градиентному закону, позволяющий увеличить среднюю скорость сходимости алгоритмов.

Показано, что с увеличением шага сходимости не только увеличивается скорость сходимости адаптивного фильтра, но и увеличивается дополнительная среднеквадратическая ошибка ξ_{exc} . Поэтому на практике значения шага сходимости в градиентных алгоритмах необходимо выбирать, исходя из компромисса между скоростью сходимости алгоритмов фильтрации и среднеквадратическим значением остаточных ошибок выходного сигнала адаптивного фильтра в установленном режиме.

Кроме рассмотренных, существует ряд других разновидностей градиентных алгоритмов адаптивной фильтрации, в частности алгоритмы, использующие субполосную фильтрацию [18, 32, 90–92], и алгоритмы в частотной области [18, 65, 93]. Эти алгоритмы позволяют реализовать адаптивные фильтры с большим числом ВК. При этом вычислительная сложность таких фильтров, как правило, меньше, чем при прямой реализации в полной полосе частот, что достигается за счет субполосной фильтрации и прореживания отсчетов обрабатываемых сигналов, блочной обработки и процедур быстрого преобразования Фурье (БПФ). Кроме того, если существует разброс собственных чисел корреляционной матрицы входного сигнала адаптивного фильтра, субполосные алгоритмы характеризуются большей скоростью сходимости, чем алгоритмы в полной полосе частот. Это достигается за счет того, что разброс собственных чисел в субполосах уменьшается и в каждой субполосе можно выбрать свой шаг сходимости, в результате чего сходимость всей гребенки адаптивных фильтров, обрабатывающих узкополосные сигналы, в среднем выравнивается.

Недостатками субполосных адаптивных фильтров является то, что в большинстве случаев они реализуют лишь приближенную адаптивную фильтрацию из-за банков анализа-синтеза сигналов, а также имеет место задержка выходного сигнала, обусловленная блочным характером обработки сигналов. Эти недостатки частично компенсируются в адаптивных алгоритмах [93].

ГЛАВА 6

РЕКУРСИВНАЯ АДАПТИВНАЯ ФИЛЬТРАЦИЯ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ

В настоящей главе рассматриваются постановка и решение задачи идентификации линейных объектов по критерию наименьших квадратов, а также RLS-алгоритм адаптивной фильтрации, вытекающий из этой задачи и полученный в результате рекурсивного обращения корреляционной матрицы входных сигналов адаптивного фильтра. Анализируются свойства этого алгоритма и приводятся результаты его численного моделирования.

6.1. Введение

В основе решения задачи оптимальной винеровской фильтрации, рассмотренной в главе 3, лежит минимизация унимодальной квадратичной целевой функции (среднеквадратической ошибки между требуемым и выходным сигналами адаптивного фильтра), параметры которой (форма, минимальное значение) зависят от статистических характеристик этих сигналов. Рассматриваемая в настоящей главе RLS-фильтрация сигналов также основана на минимизации квадратичной целевой функции. Для получения решения этой задачи используется обобщенный метод наименьших квадратов (Least Squares, LS). Фильтрация по методу наименьших квадратов является детерминированным аналогом винеровской фильтрации. Винеровская фильтрация базируется на усреднении по ансамблю реализаций стационарных в широком смысле стохастических процессов [94], в то время как LS-фильтрация использует усреднение по времени, в результате чего находятся ВК оптимального фильтра, которые различаются для каждой реализации стохастического процесса.

В настоящей главе использованы материалы из [18, 20, 29, 30].

6.2. Задача наименьших квадратов и ее решение

В предыдущем разделе было показано, что адаптивные алгоритмы могут быть получены путем реализации стратегии градиентного поиска минимума



целевой функции, в которой используется оценка ее градиента, основанная на одном текущем отсчете значений требуемого сигнала $d(k)$ и вектора сигналов линейного сумматора $\mathbf{x}_N(k)$. При этом на каждом шаге алгоритмов вектор ВК адаптивного фильтра $\mathbf{h}_N(k)$ «движется» в направлении оптимального значения $\mathbf{h}_{N,o}$.

Достоинствами таких алгоритмов является их простота (малая вычислительная и алгоритмическая сложность), а недостатком — медленная и неравномерная сходимость вектора $\mathbf{h}_N(k)$ к оптимальному вектору $\mathbf{h}_{N,o}$, а также то, что даже по мере приближения к оптимальному решению вектор $\mathbf{h}_N(k)$ обычно хаотично движется в окрестности этого решения, а не сходится к нему. Это происходит из-за того, что движение к $\mathbf{h}_{N,o}$ осуществляется не с помощью истинного градиента (3.60), (3.61), а с помощью его простой аппроксимации (5.2), (5.4). Сказанное демонстрируется на рис. 6.1, где показан фрагмент рис. 5.11 в увеличенном масштабе.

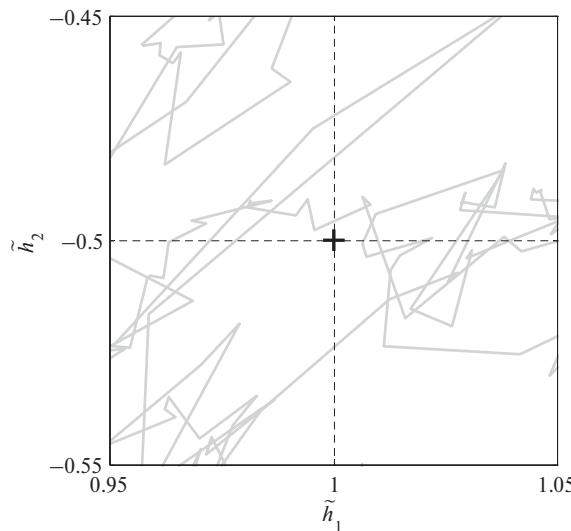


Рис. 6.1. LMS-алгоритм, поиск вектора $\tilde{\mathbf{h}}_{N,o} = [1, -0.5]^T$ (обозначено крестиком в центре рисунка) при $\mu = 0.005$ и различных начальных значениях вектора ВК $\tilde{\mathbf{h}}_N(0)$

Рассматриваемый в настоящей главе способ поиска вектора $\mathbf{h}_{N,o}$ основан на другом подходе. Адаптивный алгоритм также использует входные сигналы $d(k)$ и $\mathbf{x}_N(k)$ для поиска оптимального решения. Однако этот поиск осуществляется таким образом, что оптимальное решение обеспечивается не только для текущего отсчета k , но и для всех предыдущих отсчетов.

Исходя из вышесказанного, задача фильтрации сигналов по критерию наименьших квадратов формулируется следующим образом. Пусть имеется k от-

счетов наблюдаемых сигналов $\{x(i)\}$ и $\{d(i)\}$, где $\{x(i)\}$ — отсчеты входного сигнала, а $\{d(i)\}$ — отсчеты выходного сигнала некоторой неизвестной системы.

Предполагается, что система является линейной, т. е. между ее входным и выходным сигналами существует линейная зависимость

$$d(i) = \sum_{n=1}^N h_{n,o}^* x(i-n+1) + e_o(i) = \mathbf{h}_{N,o}^H \mathbf{x}_N(i) + e_o(i), \quad (6.1)$$

где $(N-1)$ — порядок линейной модели системы, $h_{n,o}$ — ВК линейной модели системы, а $e_o(i)$ — сигнал ошибок (шума на выходе системы), $\mathbf{h}_{N,o} = [h_{1,o}, h_{2,o}, \dots, h_{N,o}]^T$ и $\mathbf{x}_N(i) = [x(i), x(i-1), \dots, x(i-N+1)]^T$.

Требуется найти оценку вектора $\mathbf{h}_{N,o}$ на основе анализа отсчетов $\{x(i)\}$ и $\{d(i)\}$, т. е. идентифицировать неизвестную линейную систему (см. главу 1). Предполагается, что $e_o(i)$ — это белый шум с нулевым средним значением $E\{e_o(i)\} = 0$ и дисперсией $\sigma_o^2 = E\{e_o(i)e_o^*(i)\} = E\{|e_o(i)|^2\}$. Кроме того, $E\{e_o(i)e_o^*(j)\} = \sigma_o^2 \delta(i-j)$, где $\delta(i-j) = \begin{cases} 1, & \text{при } i=j \\ 0, & \text{при } i \neq j \end{cases}$ — дельта-функция. Так как отсчеты $\{x(i)\}$ известны, то среднее значение сигнала (6.1) определяется как

$$E\{d(i)\} = E\{\mathbf{h}_{N,o}^H \mathbf{x}_N(i) + e_o(i)\} = E\{\mathbf{h}_{N,o}^H \mathbf{x}_N(i)\} + E\{e_o(i)\} = \mathbf{h}_{N,o}^H \mathbf{x}_N(i). \quad (6.2)$$

Оценку вектора $\mathbf{h}_{N,o}$, т. е. вектор \mathbf{h}_N , можно найти путем решения системы линейных уравнений

$$\mathbf{h}_N^H \mathbf{x}_N(i) = d(i), \quad (6.3)$$

где $i = N, N+1, \dots, k$.

При $k \geq 2$ выражение (6.3) представляет собой переопределенную систему линейных уравнений, так как число уравнений $k - N + 1$ больше числа неизвестных N . Решение системы уравнений (6.3) может быть получено в результате приближения значений отсчетов сигнала

$$y(i) = \mathbf{h}_N^H \mathbf{x}_N(i) \quad (6.4)$$

к значениям сигнала $d(i)$, т. е. путем минимизации среднеквадратического значения ошибок

$$e(i) = d(i) - y(i). \quad (6.5)$$

Переменные (6.4) и (6.5) можно интерпретировать как сигналы трансверсального фильтра (рис. 6.2). Поэтому задачу нахождения вектора $\mathbf{h}_{N,o}$ можно сформулировать как задачу нахождение вектора ВК \mathbf{h}_N этого фильтра, при которых его выходной сигнал $y(i)$ наилучшим образом воспроизводит, т. е. моделирует, сигнал $d(i)$. В этой постановке предполагается, что число ВК N модели линейной системы известно и оно совпадает с числом ВК фильтра.

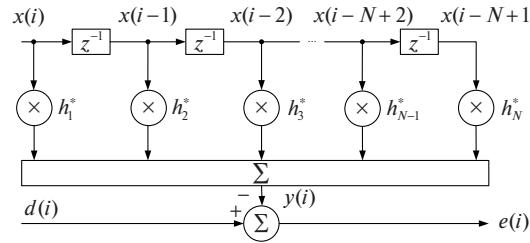


Рис. 6.2. Решение LS-задачи с помощью линейного трансверсального фильтра

Другими словами, требуется найти вектор ВК \mathbf{h}_N , обеспечивающий минимизацию суммы квадратов модулей ошибок между сигналами $d(i)$ и $y(i)$, т. е. минимизацию целевой функции $J(\mathbf{h}_N) = \sum_{i=N}^k e(i)e^*(i)$. С учетом выражений (6.4) и (6.5), эта функция может быть представлена как

$$\begin{aligned} J(\mathbf{h}_N) &= \sum_{i=N}^k e(i)e^*(i) = \\ &= \sum_{i=N}^k d(i)d(i)^* - \sum_{i=N}^k d(i)\mathbf{x}_N^H(i)\mathbf{h}_N - \sum_{i=N}^k \mathbf{h}_N^H \mathbf{x}_N(i)d^*(i) + \sum_{i=N}^k \mathbf{h}_N^H \mathbf{x}_N(i)\mathbf{x}_N^H(i)\mathbf{h}_N = \\ &= \sum_{i=N}^k d(i)d(i)^* - \hat{\mathbf{r}}_N^H \mathbf{h}_N - \mathbf{h}_N^H \hat{\mathbf{r}}_N + \mathbf{h}_N^H \hat{\mathbf{R}}_N \mathbf{h}_N, \end{aligned} \quad (6.6)$$

где

$$\hat{\mathbf{R}}_N = \sum_{i=N}^k \mathbf{x}_N(i)\mathbf{x}_N^H(i) \quad (6.7)$$

и

$$\hat{\mathbf{r}}_N = \sum_{i=N}^k \mathbf{x}_N(i)d^*(i). \quad (6.8)$$

Условие

$$\nabla_{\mathbf{h}_N} J = \frac{\partial J}{\partial \mathbf{h}_N^*} = -\hat{\mathbf{r}}_N + \hat{\mathbf{R}}_N \mathbf{h}_N = \mathbf{0}_N \quad (6.9)$$

позволяет найти координаты минимума функции (6.6), т. е. вектор

$$\mathbf{h}_N = \hat{\mathbf{R}}_N^{-1} \hat{\mathbf{r}}_N. \quad (6.10)$$

В случае действительных ВК $\tilde{\mathbf{h}}_N$ целевая функция определяется как

$$\begin{aligned} J(\tilde{\mathbf{h}}_N) &= \sum_{i=N}^k e^2(i) = \sum_{i=N}^k d^2(i) - 2 \sum_{i=N}^k \tilde{\mathbf{h}}_N^T \mathbf{x}_N(i)d(i) + \sum_{i=N}^k \tilde{\mathbf{h}}_N^T \mathbf{x}_N(i)\mathbf{x}_N^T(i)\mathbf{h}_N = \\ &= \sum_{i=N}^k d^2(i) - 2 \tilde{\mathbf{h}}_N^T \hat{\mathbf{r}}_N + \tilde{\mathbf{h}}_N^T \hat{\mathbf{R}}_N \tilde{\mathbf{h}}_N, \end{aligned} \quad (6.11)$$

где

$$\hat{\mathbf{R}}_N = \sum_{i=N}^k \mathbf{x}_N(i)\mathbf{x}_N^T(i) \quad (6.12)$$

и

$$\hat{\mathbf{r}}_N = \sum_{i=N}^k \mathbf{x}_N(i) d(i), \quad (6.13)$$

а координаты ее минимума находятся из соотношения

$$\nabla_{\tilde{\mathbf{h}}_N} J = \frac{\partial J}{\partial \tilde{\mathbf{h}}_N} = -2\hat{\mathbf{r}}_N + 2\hat{\mathbf{R}}_N \tilde{\mathbf{h}}_N = \mathbf{0}_N \quad (6.14)$$

как

$$\tilde{\mathbf{h}}_N = \hat{\mathbf{R}}_N^{-1} \hat{\mathbf{r}}_N. \quad (6.15)$$

Минимальные значения J_{\min} функций (6.6) и (6.11) определяются при подстановке в уравнения (6.6) и (6.11) выражений $\hat{\mathbf{r}}_N = \hat{\mathbf{R}}_N \mathbf{h}_N$ и $\tilde{\mathbf{h}}_N = \hat{\mathbf{R}}_N \tilde{\mathbf{h}}_N$, определяемых уравнениями (6.9) и (6.14). Эти значения равны соответственно

$$J_{\min} = \sum_{i=N}^k d(i) d(i)^* - \hat{\mathbf{r}}_N^H \mathbf{h}_N - \mathbf{h}_N^H \hat{\mathbf{r}}_N + \mathbf{h}_N^H \hat{\mathbf{r}}_N = \sum_{i=N}^k d(i) d(i)^* - \hat{\mathbf{r}}_N^H \mathbf{h}_N \quad (6.16)$$

и

$$J_{\min} = \sum_{i=N}^k d^2(i) - \hat{\mathbf{r}}_N^T \tilde{\mathbf{h}}_N - \tilde{\mathbf{h}}_N^T \hat{\mathbf{r}}_N + \tilde{\mathbf{h}}_N^T \hat{\mathbf{r}}_N = \sum_{i=N}^k d^2(i) - \hat{\mathbf{r}}_N^T \tilde{\mathbf{h}}_N. \quad (6.17)$$

Выражения (6.16) и (6.17) схожи с выражением (3.64) для ξ_{\min} при винеровской фильтрации. Отличие заключается в том, что усреднение по ансамблю реализаций в (3.64) при винеровской фильтрации заменяется усреднением по времени в выражениях (6.16) и (6.17) при LS-фильтрации.

6.3. Основные свойства LS-решения

Решения задачи наименьших квадратов (6.10) и (6.15) обладают определенными свойствами. С целью получения соотношений для анализа этих свойств определим матрицу $\mathbf{X}_{N(k-N+1)}$ со столбцами $\mathbf{x}_N(i)$

$$\begin{aligned} \mathbf{X}_{N(k-N+1)} &= [\mathbf{x}_N(N), \mathbf{x}_N(N+1), \dots, \mathbf{x}_N(k)] = \\ &= \begin{bmatrix} x(N) & x(N+1) & \dots & x(k) \\ x(N-1) & x(N) & \dots & x(k-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(1) & x(2) & \dots & x(k-N+1) \end{bmatrix} \end{aligned} \quad (6.18)$$

и вектор

$$\mathbf{d}_{k-N+1} = [d(N), d(N+1), \dots, d(k-N+1)]^T, \quad (6.19)$$

где $N \leq i \leq k$.

Для больших значений k и стационарных процессов $\{x(i)\}$ и $\{d(i)\}$ матрицу (6.7) и вектор (6.8) можно выразить как

$$\hat{\mathbf{R}}_N = \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H = \sum_{i=N}^k \mathbf{x}_N(i) \mathbf{x}_N^H(i) \approx (k-N+1) \mathbf{R}_N \quad (6.20)$$

и

$$\hat{\mathbf{r}}_N = \mathbf{X}_{N(k-N+1)} \mathbf{d}_{k-N+1}^* = \sum_{i=N}^k \mathbf{x}_N(i) d^*(i) \approx (k-N+1) \mathbf{r}_N. \quad (6.21)$$

В этом случае согласно уравнению (6.10) вектор ВК \mathbf{h}_N определяется как

$$\mathbf{h}_N = \hat{\mathbf{R}}_N^{-1} \hat{\mathbf{r}}_N \approx [(k-N+1) \mathbf{R}_N]^{-1} [(k-N+1) \mathbf{r}_N] = \mathbf{R}_N^{-1} \mathbf{r}_N, \quad (6.22)$$

где \mathbf{R}_N — корреляционная матрица сигналов $x(i)$ в линейном фильтре (см. рис. 6.2), а \mathbf{r}_N — вектор взаимной корреляции этих сигналов и сигнала $d(i)$.

Аналогично для случая действительных ВК матрица $\hat{\mathbf{R}}_N$ и вектор $\hat{\mathbf{r}}_N$ определяются как

$$\hat{\mathbf{R}}_N = \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^T = \sum_{i=N}^k \mathbf{x}_N(i) \mathbf{x}_N^T(i) \approx (k-N+1) \mathbf{R}_N, \quad (6.23)$$

$$\hat{\mathbf{r}}_N = \mathbf{X}_{N(k-N+1)} \mathbf{d}_{k-N+1} = \sum_{i=N}^k \mathbf{x}_N(i) d(i) \approx (k-N+1) \mathbf{r}_N, \quad (6.24)$$

а вектор $\tilde{\mathbf{h}}_N$ определяется как

$$\tilde{\mathbf{h}}_N \approx \mathbf{R}_N^{-1} \mathbf{r}_N. \quad (6.25)$$

Из уравнений (6.20) и (6.23) следует, что матрица $\hat{\mathbf{R}}_N$ обладает всеми свойствами корреляционной матрицы \mathbf{R}_N (см. главу 3), а из уравнений (6.22) и (6.25) — что решения задачи наименьших квадратов являются схожими с решениями задачи оптимальной винеровской фильтрации, что более строго доказывается ниже.

Из уравнений (6.20), (6.21) следует, что вектор ВК (6.10) может быть представлен как

$$\mathbf{h}_N = \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{d}_{k-N+1}^*. \quad (6.26)$$

Если уравнение (6.1) в векторном виде записать как

$$\mathbf{d}_{k-N+1} = \mathbf{y}_{k-N+1,o} + \mathbf{e}_{k-N+1,o} = \mathbf{X}_{N(k-N+1)}^T \mathbf{h}_{N,o}^* + \mathbf{e}_{k-N+1,o}, \quad (6.27)$$

то с учетом уравнения (6.27) уравнение (6.26) примет вид

$$\begin{aligned} \mathbf{h}_N &= \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \left[\mathbf{X}_{N(k-N+1)}^T \mathbf{h}_{N,o}^* + \mathbf{e}_{k-N+1,o} \right]^* = \\ &= \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \mathbf{h}_{N,o} + \\ &+ \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^* = \\ &= \mathbf{h}_{N,o} + \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^*. \end{aligned} \quad (6.28)$$

Так как матрица $\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)}$ известна, поскольку она полностью определена известными отсчетами наблюдаемого процесса $\{x(i)\}$, среднее значение шума $E\{e_o(i)\} = 0$, а значит, и $E\{\mathbf{e}_{k-N+1,o}^*\} = \mathbf{0}_N$, то среднее значение вектора

$$\begin{aligned}
E\{\mathbf{h}_N\} &= \mathbf{h}_{N,o} + E\left\{\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^*\right\} = \\
&= \mathbf{h}_{N,o} + \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} E\{\mathbf{e}_{k-N+1,o}^*\} = \\
&= \mathbf{h}_{N,o} + \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{0}_{k-N+1} = \mathbf{h}_{N,o}
\end{aligned} \tag{6.29}$$

является несмешенной оценкой $\mathbf{h}_{N,o}$, т. е. вектора ВК неизвестной линейной системы.

Ковариационная матрица вектора (6.28), т. е. мера его отклонения от вектора $\mathbf{h}_{N,o}$, обусловленного шумом $\{e_o(i)\}$, оценивается как

$$\begin{aligned}
\hat{\mathbf{K}}_N &= \text{cov}\{\mathbf{h}_N\} = E\{\Delta \mathbf{h}_N \Delta \mathbf{h}_N^H\} = E\left\{\left[\mathbf{h}_N - \mathbf{h}_{N,o}\right] \left[\mathbf{h}_N - \mathbf{h}_{N,o}\right]^H\right\} = \\
&= E\left\{\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^* \times \right. \\
&\quad \left. \times \left(\mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^*\right)^H \left(\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}\right)^H\right\} = \\
&= \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} E\{\mathbf{e}_{k-N+1,o}^* \mathbf{e}_{k-N+1,o}^T\} \times \\
&\quad \times \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} = \\
&= \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \sigma_o^2 \mathbf{I}_{k-N+1} \times \\
&\quad \times \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} = \\
&= \sigma_o^2 \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} = \\
&= \sigma_o^2 \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1},
\end{aligned} \tag{6.30}$$

так как $\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} = \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-H}$ — эрмитова матрица и $E\{\mathbf{e}_{k-N+1,o}^* \mathbf{e}_{k-N+1,o}^T\} = \sigma_o^2 \mathbf{I}_{k-N+1}$.

Выражение (6.30) позволяет доказать, что (6.28) является наилучшей оценкой вектора $\mathbf{h}_{N,o}$.

Действительно, пусть аналогично уравнению (6.26) существует некоторая другая оценка вектора $\mathbf{h}_{N,o}$, определяемая как $\hat{\mathbf{h}}_N = \mathbf{B}_{N(k-N+1)} \mathbf{d}_{k-N+1}^*$. С учетом выражения (6.27) вектор $\hat{\mathbf{h}}_N$ определяется как

$$\begin{aligned}
\hat{\mathbf{h}}_N &= \mathbf{B}_{N(k-N+1)} \mathbf{d}_{k-N+1}^* = \mathbf{B}_{N(k-N+1)} \left[\mathbf{X}_{N(k-N+1)}^T \mathbf{h}_{N,o}^* + \mathbf{e}_{k-N+1,o}^* \right]^* = \\
&= \mathbf{B}_{N(k-N+1)} \mathbf{X}_{(k-N+1)N}^H \mathbf{h}_{N,o} + \mathbf{B}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^* = \mathbf{h}_{N,o} + \mathbf{B}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^*,
\end{aligned} \tag{6.31}$$

если

$$\mathbf{B}_{N(k-N+1)} \mathbf{X}_{(k-N+1)N}^H = \mathbf{I}_N. \tag{6.32}$$

Среднее значение вектора (6.31) также является несмешенной оценкой вектора $\mathbf{h}_{N,o}$ по аналогии (6.29) $E\{\hat{\mathbf{h}}_N\} = \mathbf{h}_{N,o}$.

Кроме того, аналогично (6.30) ковариационная матрица вектора (6.31) определяется как

$$\begin{aligned}\hat{\mathbf{K}}_N &= \text{cov}\{\hat{\mathbf{h}}_N\} = E\{\Delta\mathbf{h}_N \Delta\mathbf{h}_N^H\} = E\left\{\left[\mathbf{h}_N - \mathbf{h}_{N,o}\right]\left[\mathbf{h}_N - \mathbf{h}_{N,o}\right]^H\right\} = \\ &= E\left\{\mathbf{B}_{N(k-N+1)} \mathbf{e}_{k-N+1,o}^* \mathbf{e}_{k-N+1,o} \mathbf{B}_{N(k-N+1)}^H\right\} = \sigma_o^2 \left[\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H\right].\end{aligned}\quad (6.33)$$

Из уравнений (6.28) и (6.31) видно, что различие между векторами \mathbf{h}_N и $\hat{\mathbf{h}}_N$ обусловлено множителями (матрицами) $\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)}$ и $\mathbf{B}_{N(k-N+1)}$ во втором слагаемом этих уравнений.

С целью анализа этого различия определим матрицу разности указанных множителей

$$\Psi_{N(k-N+1)} = \mathbf{B}_{N(k-N+1)} - \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)}, \quad (6.34)$$

а затем с учетом уравнения (6.32) определим матрицу

$$\begin{aligned}\Psi_{N(k-N+1)} \Psi_{N(k-N+1)}^H &= \mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H - \\ &- \mathbf{B}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} - \\ &- \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H + \\ &+ \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1} = \\ &= \mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H - \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}.\end{aligned}\quad (6.35)$$

Диагональные элементы матриц $\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H$ и $\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}$ подобно диагональным элементам корреляционной матрицы (3.5), определяемой произведением векторов $\mathbf{x}_N(k) \mathbf{x}_N^H(k)$, являются неотрицательными. Таким образом, матрица $\Psi_{N(k-N+1)} \Psi_{N(k-N+1)}^H$ имеет структуру корреляционной матрицы с неотрицательными диагональными элементами. Поэтому из уравнения (6.35) можно заключить, что

$$\begin{aligned}\text{diag}\left\{\Psi_{N(k-N+1)} \Psi_{N(k-N+1)}^H\right\} &= \\ &= \text{diag}\left\{\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H - \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}\right\} = \\ &= \text{diag}\left\{\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H\right\} - \text{diag}\left\{\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}\right\} \geq \mathbf{O}_N\end{aligned}\quad (6.36)$$

или что

$$\text{diag}\left\{\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H\right\} \geq \text{diag}\left\{\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}\right\}, \quad (6.37)$$

где запись $\text{diag}\{\mathbf{A}_N\}$ служит для обозначения диагональной матрицы, диагональные элементы которой составлены из диагональных элементов матрицы \mathbf{A}_N . Умножив обе части неравенства (6.37) на σ_o^2 , получим, что

$$\sigma_o^2 \text{diag}\left\{\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H\right\} \geq \sigma_o^2 \text{diag}\left\{\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H\right]^{-1}\right\}. \quad (6.38)$$

Так как $\sigma_o^2 [\mathbf{B}_{N(k-N+1)} \mathbf{B}_{N(k-N+1)}^H] = \text{cov}\{\hat{\mathbf{h}}_N\}$, см. (6.33), а $\sigma_o^2 [\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H]^{-1} = \text{cov}\{\mathbf{h}_N\}$, см. уравнение (6.30), то из (6.29) и (6.38) следует, что вектор \mathbf{h}_N , см. уравнение (6.26), полученный в результате решения задачи среднеквадратического приближения процессов $\{x(i)\}$ и $\{d(i)\}$, является наилучшей несмещенной оценкой вектора $\mathbf{h}_{N,o}$ в том смысле, что каждый элемент вектора \mathbf{h}_N имеет минимально возможное среднеквадратическое отклонение от элементов вектора $\mathbf{h}_{N,o}$, определяемое диагональными элементами матрицы $\hat{\mathbf{K}}_N$.

6.4. Геометрическая интерпретация метода наименьших квадратов

В главе 3 было показано, что если ВК фильтра соответствуют винеровскому решению $\mathbf{h}_{N,o}$, то ошибка $e_o(k)$ ортогональна компонентам вектора входного сигнала фильтра $x(k-n+1)$, $n=1, 2, \dots, N$ и выходному сигналу фильтра $y_o(k)$, т. е. $E\{x_N(k-n+1)e_o^*(k)\}=0$ и $E\{y_o(k)e_o^*(k)\}=0$, см. уравнения (3.68) и (3.69). Эти соотношения называются принципом ортогональности для фильтров Винера.

Аналогичные результаты могут быть получены и для фильтров, значения ВК которых являются результатом решения LS-задачи.

Если в (6.9) использовать уравнения (6.20) и (6.21), то получим, что

$$\begin{aligned} \hat{\mathbf{R}}_N \mathbf{h}_N - \hat{\mathbf{r}}_N &= \mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \mathbf{h}_N - \mathbf{X}_{N(k-N+1)} \mathbf{d}_{k-N+1}^* = \\ &= \mathbf{X}_{N(k-N+1)} [\mathbf{y}_{k-N+1}^* - \mathbf{d}_{k-N+1}^*] = -\mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1}^* = \mathbf{0}_N, \end{aligned} \quad (6.39)$$

где

$$\mathbf{y}_{k-N+1}^* = \mathbf{X}_{N(k-N+1)}^H \mathbf{h}_N \quad (6.40)$$

и

$$\mathbf{e}_{k-N+1}^* = \mathbf{d}_{k-N+1}^* - \mathbf{y}_{k-N+1}^*. \quad (6.41)$$

Из уравнения (6.39) следует, что векторы, образованные элементами столбцов матрицы $\mathbf{X}_{N(k-N+1)}^H$, ортогональны вектору ошибок \mathbf{e}_{k-N+1}^* , так как скалярное произведение этих векторов равно нулю, т. е.

$$[\mathbf{x}_{k-N+1}^*(n)]^H \mathbf{e}_{k-N+1}^* = \mathbf{x}_{k-N+1}^T(n) \mathbf{e}_{k-N+1}^* = 0, \quad (6.42)$$

где

$$\mathbf{x}_{(k-N+1)}^*(n) = [x^*(N-n+1), x^*(N-n+2), \dots, x^*(N-n+k-N+1)]^T \quad (6.43)$$

— столбцы матрицы $\mathbf{X}_{N(k-N+1)}^H$, а

$$\mathbf{x}_{k-N+1}^T(n) = [x(N-n+1), x(N-n+2), \dots, x(N-n+k-N+1)] \quad (6.44)$$

— строки матрицы $\mathbf{X}_{N(k-N+1)}$, $n=1, 2, \dots, N$.

Умножив выражение (6.39) слева на \mathbf{h}_N^H , получим, что

$$\mathbf{h}_N^H \mathbf{X}_{N(k-N+1)} \mathbf{e}_{k-N+1}^* = \mathbf{y}_{k-N+1}^T \mathbf{e}_{k-N+1}^* = (\mathbf{y}_{k-N+1}^*)^H \mathbf{e}_{k-N+1}^* = 0, \quad (6.45)$$

т. е. векторы \mathbf{y}_{k-N+1}^* и \mathbf{e}_{k-N+1}^* также ортогональны, так как их скалярное произведение равно нулю.

Сравнение уравнений (3.68), (3.39) и (6.42), (6.45) показывает, что принцип ортогональности при винеровской фильтрации определяется в терминах статистического усреднения по ансамблю реализаций, в то время как в LS-фильтрации — в терминах скалярного произведения векторов.

Если уравнения (6.42) и (6.45) представить как

$$\frac{1}{k-N+1} \mathbf{x}_{k-N+1}^T(n) \mathbf{e}_{k-N+1}^* = \frac{1}{k-N+1} \sum_{i=N}^k x(i-n+1) e^*(i) = 0 \quad (6.46)$$

и

$$\frac{1}{k-N+1} \mathbf{y}_{k-N+1}^T \mathbf{e}_{k-N+1}^* = \frac{1}{k-N+1} \sum_{i=N}^k y(i) e^*(i) = 0, \quad (6.47)$$

т. е. разделить левую и правую их части на $k-N+1$ (число наблюдаемых отсчетов сигналов, участвующих в LS-оценивании), то видно что при LS-фильтрации ортогональность в терминах скалярного произведения векторов можно заменять на ортогональность в терминах усреднения по времени.

Другими словами, если ВК фильтра оптимизированы в среднеквадратическом смысле, т. е. вычислены с помощью уравнения (6.26), то отсчеты сигналов в этом фильтре, а также выходной сигнал фильтра ортогональны к ошибкам оценивания требуемого сигнала.

Если в уравнении (6.40) использовать (6.26), то получим, что

$$\begin{aligned} \mathbf{y}_{k-N+1}^* &= \mathbf{X}_{N(k-N+1)}^H \mathbf{h}_N = \\ &= \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \mathbf{d}_{k-N+1}^* = \mathbf{P}_{k-N+1} \mathbf{d}_{k-N+1}^*. \end{aligned} \quad (6.48)$$

В уравнении (6.48) квадратная матрица \mathbf{P}_{k-N+1} с числом элементов $(k-N+1) \times (k-N+1)$ — это проекционная матрица

$$\mathbf{P}_{k-N+1} = \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)}, \quad (6.49)$$

которая проецирует вектор \mathbf{d}_{k-N+1}^* в гиперплоскость, охваченную столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$.

Векторы, образованные столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$, не являются ортогональными, а плоскость, охваченная этими столбцами, в $(k-N+1)$ -мерном пространстве может в общем случае иметь произвольную ориентацию. Для геометрической иллюстрации сказанного рассмотрим действительную ма-

трицу $\mathbf{X}_{23}^T = \begin{bmatrix} 6 & 0 \\ 2 & 6 \\ 0.5 & 2 \end{bmatrix}$. Плоскость, охваченная столбцами этой матрицы, показана на рис. 6.3.

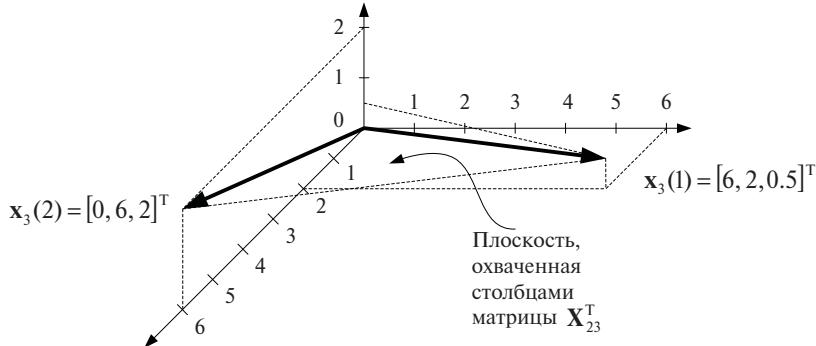


Рис. 6.3. Геометрическая иллюстрация трехмерной плоскости, охваченной столбцами матрицы \mathbf{X}_{23}^T

Из уравнения (6.49) следует, что

$$\begin{aligned} \mathbf{P}_{k-N+1}^H &= \left[\mathbf{X}_{N(k-N+1)} \right]^H \left\{ \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \right\}^H = \\ &= \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-H} \left[\mathbf{X}_{N(k-N+1)}^H \right]^H = \\ &= \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} = \mathbf{P}_{k-N+1}, \end{aligned} \quad (6.50)$$

так как матрица (6.20) является эрмитовой, а значит, и обратная матрица тоже эрмитова (см. главу 3), т. е. $\left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-H} = \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1}$.

Кроме того, используя уравнения (6.49) и (6.50), получим, что

$$\begin{aligned} \mathbf{P}_{k-N+1}^H \mathbf{P}_{k-N+1} &= \left\{ \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \right\} \times \\ &\quad \times \left\{ \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} \right\} = \\ &= \mathbf{X}_{N(k-N+1)}^H \left[\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H \right]^{-1} \mathbf{X}_{N(k-N+1)} = \mathbf{P}_{k-N+1}. \end{aligned} \quad (6.51)$$

Так как согласно уравнениям (6.41) и (6.48)

$$\mathbf{e}_{k-N+1}^* = \mathbf{d}_{k-N+1}^* - \mathbf{y}_{k-N+1}^* = \mathbf{d}_{k-N+1}^* - \mathbf{P}_{k-N+1} \mathbf{d}_{k-N+1}^* = (\mathbf{I}_{k-N+1} - \mathbf{P}_{k-N+1}) \mathbf{d}_{k-N+1}^* = \mathbf{Q}_{k-N+1} \mathbf{d}_{k-N+1}^*, \quad (6.52)$$

то это означает, что матрица \mathbf{Q}_{k-N+1} проецирует вектор \mathbf{d}_{k-N+1}^* на гиперплоскость, являющуюся ортогональным дополнением к гиперплоскости, охваченной столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$.

Действительно, согласно уравнениям (6.39) и (6.42) столбцы матрицы $\mathbf{X}_{N(k-N+1)}^H$ и вектор \mathbf{e}_{k-N+1}^* являются попарно ортогональными. Значит, в соответствии с определением ортогональных плоскостей, плоскость, в которой лежит вектор \mathbf{e}_{k-N+1}^* , является ортогональной гиперплоскости, охваченной столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$. Две гиперплоскости ортогональны, если каждая пара векторов, один из которых (\mathbf{e}_{k-N+1}^* в рассматриваемом случае) взят из первой гиперплоскости, а второй (любой из векторов $\mathbf{x}_{k-N+1}^*(n)$, $n=1, 2, \dots, N$), взятый из второй гиперплоскости, являются ортогональными.

Из уравнения (6.45) видно, что векторы \mathbf{e}_{k-N+1}^* и \mathbf{y}_{k-N+1}^* также являются ортогональными, а это означает, что вектор \mathbf{y}_{k-N+1}^* лежит в гиперплоскости, охваченной столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$. Следовательно, согласно уравнению (6.48) матрица \mathbf{P}_{k-N+1} проецирует вектор \mathbf{d}_{k-N+1}^* на гиперплоскость, охваченную столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$.

Таким образом, векторы \mathbf{e}_{k-N+1}^* и \mathbf{y}_{k-N+1}^* представляют собой ортогональные компоненты вектора \mathbf{d}_{k-N+1}^* , как показано на рис. 6.4.

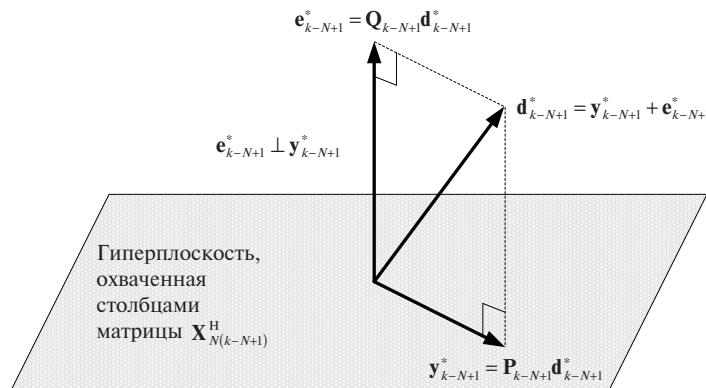


Рис. 6.4. Геометрическая интерпретация LS-метода

Согласно рис. 6.4 и уравнениям (6.41), (6.48) и (6.52) вектор

$$\mathbf{d}_{k-N+1}^* = \mathbf{y}_{k-N+1}^* + \mathbf{e}_{k-N+1}^* = \mathbf{P}_{k-N+1} \mathbf{d}_{k-N+1}^* + \mathbf{Q}_{k-N+1} \mathbf{d}_{k-N+1}^* = (\mathbf{P}_{k-N+1} + \mathbf{Q}_{k-N+1}) \mathbf{d}_{k-N+1}^*. \quad (6.53)$$

Следовательно,

$$\mathbf{P}_{k-N+1} + \mathbf{Q}_{k-N+1} = \mathbf{I}_{k-N+1}. \quad (6.54)$$

Кроме того,

$$\begin{aligned} \mathbf{P}_{k-N+1}^H \mathbf{Q}_{k-N+1} &= \mathbf{P}_{k-N+1}^H (\mathbf{I}_{k-N+1} - \mathbf{P}_{k-N+1}) = \\ &= \mathbf{P}_{k-N+1}^H - \mathbf{P}_{k-N+1}^H \mathbf{P}_{k-N+1} = \mathbf{P}_{k-N+1}^H - \mathbf{P}_{k-N+1} = \\ &= \mathbf{P}_{k-N+1} - \mathbf{P}_{k-N+1} = \mathbf{O}_{k-N+1}. \end{aligned} \quad (6.55)$$

Таким образом, для заданных \mathbf{d}_{k-N+1}^* и $\mathbf{X}_{N(k-N+1)}^H$ мы определили вектор \mathbf{h}_N , который минимизирует расстояние между векторами $\mathbf{y}_{k-N+1}^* = \mathbf{X}_{N(k-N+1)}^H \mathbf{h}_N$ и \mathbf{d}_{k-N+1}^* . В этом случае, вектор \mathbf{y}_{k-N+1}^* рассматривается как наилучшая среднеквадратическая аппроксимация вектора \mathbf{d}_{k-N+1}^* . Эта аппроксимация получается путем проекции вектора \mathbf{d}_{k-N+1}^* гиперплоскость, охваченную столбцами матрицы $\mathbf{X}_{N(k-N+1)}^H$.

Ниже приводится пример [18], демонстрирующий рассмотренный метод решения LS-задачи для действительных переменных.

Рассмотрим матрицу $\mathbf{X}_{23} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0.1 \end{bmatrix}$ и вектор $\mathbf{d}_3 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$. В этом случае $\mathbf{X}_{23} \mathbf{X}_{23}^T = \begin{bmatrix} 5 & 4 \\ 4 & 5.01 \end{bmatrix}$, $\mathbf{X}_{23} \mathbf{d}_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, вектор ВК фильтра (см. рис. 6.2) определяется как $\tilde{\mathbf{h}}_2 = [\mathbf{X}_{23} \mathbf{X}_{23}^T]^{-1} \mathbf{X}_{23} \mathbf{d}_3 = \begin{bmatrix} 0.9956 \\ -0.9945 \end{bmatrix}$, вектор отсчетов выходного

сигнала фильтра — как $\mathbf{y}_3 = \mathbf{X}_{23}^T \tilde{\mathbf{h}}_2 = \begin{bmatrix} 0.9967 \\ -0.9934 \\ 0.0033 \\ -0.0994 \end{bmatrix}$, вектор отсчетов сигнала ошибок — как $\mathbf{e}_3 = \mathbf{d}_3 - \mathbf{y}_3 = \begin{bmatrix} -0.0066 \\ 0.0994 \end{bmatrix}$, а также выполняются соотношения (6.39)

$\mathbf{X}_{23} \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}_2$ и (6.35) $\mathbf{y}_3^T \mathbf{e}_3 = 0$. Проекционная матрица \mathbf{P}_3 определяется как

$\mathbf{P}_3 = \mathbf{P}_3^T = \mathbf{X}_{23}^T [\mathbf{X}_{23} \mathbf{X}_{23}^T]^{-1} \mathbf{X}_{23} = \begin{bmatrix} 0.9989 & 0.0022 & -0.0331 \\ 0.0022 & 0.9956 & 0.0663 \\ -0.0331 & 0.0663 & 0.0055 \end{bmatrix}$, а проекционная матрица

\mathbf{Q}_3 — как $\mathbf{Q}_3 = \mathbf{I}_3 - \mathbf{P}_3 = \begin{bmatrix} 0.0011 & -0.0022 & 0.0331 \\ -0.0022 & 0.0044 & -0.0663 \\ 0.0331 & -0.0663 & 0.9945 \end{bmatrix}$. Справедливо также соотношение (6.55) $\mathbf{P}_3^T \mathbf{Q}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{0}_3$.

Таким образом, рассмотренный пример демонстрирует справедливость приведенных в настоящем разделе соотношений.

6.5. Рекурсивная задача наименьших квадратов

Уравнение (6.26) является вычислительно-сложным, что обусловлено необходимостью обращения матрицы $\mathbf{X}_{N(k-N+1)} \mathbf{X}_{N(k-N+1)}^H$ для каждого нового значения k .

Уменьшить вычислительную сложность этого уравнения можно с помощью рекурсивной процедуры, рассматриваемой в настоящем разделе.

Рекурсивная задача по критерию наименьших квадратов является модификацией рассмотренной ранее нерекурсивной задачи. Модифицированная задача формулируется следующим образом.

Пусть к дискретному моменту времени k имеется $k - 1$ отсчетов наблюдаемых процессов $\{x(i)\}$ и $\{d(i)\}$, где $i = 1, 2, \dots, k - 1$. Предположим, что вектор ВК $\mathbf{h}_N(k-1)$ линейного фильтра (см. рис. 6.2), который решает LS-задачу, для $k - 1$ наблюдений уже определен. Ставится задача получить вектор $\mathbf{h}_N(k)$ путем коррекции предыдущего LS-решения, используя только $\mathbf{h}_N(k-1)$ и новые данные $x(k)$ и $d(k)$, вместо получения нового решения $\mathbf{h}_N(k)$ с помощью вычислений в соответствии с уравнением (6.26), в котором при любом k необходимо использовать все k значений наблюдаемых процессов $\{x(i)\}$ и $\{d(i)\}$.

В такой постановке ВК $\mathbf{h}_N(k)$ можно найти путем минимизации целевой функции

$$E_N(k) = \sum_{i=1}^k e_N(i) e_N^*(i), \quad (6.56)$$

где

$$e_N(i) = d(i) - y_N(i) = d(i) - \sum_{n=1}^N h_n^*(k) x(i-n+1) = d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i) \quad (6.57)$$

и $\mathbf{h}_N(k) = [h_1(k), h_2(k), \dots, h_N(k)]^T$. Здесь и далее при описании RLS-алгоритмов в ряде переменных будет использоваться нижний индекс, указывающий на то, что они являются функциями многих переменных. Такое уточнение удобно с точки зрения описания быстрых версий RLS-алгоритмов, рассматриваемых в следующих главах, поскольку в этом описании будут возникать одноименные функции от $N-1$, N и $N+1$ переменных.

В уравнении (6.56) используются два индекса i и k , потому что в каждый момент времени k решается LS-задача, основанная на всех отсчетах наблюдаемых процессов вплоть до текущего т. е. включая отсчет k . Поэтому вектор $\mathbf{h}_N(k)$, полученный в результате решения этой задачи, зависит от всех значений $\mathbf{x}_N(i)$ и $d(i)$, где $i = 1, 2, \dots, k$.

Обрабатываемые сигналы в целевой функции (6.56) часто взвешиваются с помощью параметра λ , т. е. как

$$E_N(k) = \sum_{i=1}^k \lambda^{k-i} e_N(i) e_N^*(i), \quad (6.58)$$

(см. рис. 6.5). Здесь и далее параметр λ не следует путать с собственными числами λ_n , $n = 1, 2, \dots, N$ корреляционной матрицы \mathbf{R}_N . Так исторически сложилось, что в литературе по адаптивной фильтрации для обозначения этих величин используется один и тот же символ λ .

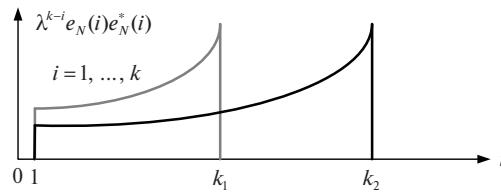


Рис. 6.5. Экспоненциальное взвешивание обрабатываемых сигналов

Взвешивание сигналов с помощью параметра λ позволяет учитывать небольшие изменения в статистике сигналов, если эти сигналы нестационарные, т.е. их статистические параметры меняются во времени. Параметр λ называется параметром экспоненциального взвешивания или параметром «забывания» («forgetting factor»). Его значение обычно ограничено пределами $(1-0.4/N) \leq \lambda \leq 1$ [95, 96].

Чтобы понять, как осуществляется экспоненциальное взвешивание, рассмотрим функцию (6.58) при нескольких значениях k . При $k=1$ $E_N(1)=\lambda^0 e_N(1)e_N^*(1)=e_N(1)e_N^*(1)$. При $k=2$ $E_N(2)=\lambda^0 e_N(2)e_N^*(2)+\lambda^1 e_N(1)e_N^*(1)=\lambda E_N(1)+e_N(2)e_N^*(2)$. При $k=3$ $E_N(3)=\lambda^0 e_N(3)e_N^*(3)+\lambda^1 e_N(2)e_N^*(2)+\lambda^2 e_N(1)e_N^*(1)=\lambda^0 e_N(3)e_N^*(3)+\lambda [e_N(2)e_N^*(2)+\lambda e_N(1)e_N^*(1)]=\lambda E_N(2)+e_N(3)e_N^*(3)$.

Можно заметить, что для любого k справедливо соотношение

$$E_N(k)=\lambda E_N(k-1)+e_N(k)e_N^*(k). \quad (6.59)$$

Таким образом, в уравнении (6.58) каждое слагаемое $e_N(i)e_N^*(i)$ взвешивается с помощью коэффициента, равного λ^{k-i} , т.е. умножается на число, которое уменьшается по мере того, как увеличивается разность $k-i$, если $0 < \lambda < 1$.

Например, при $\lambda=0.9$ $\lambda^0=1$, $\lambda^1=0.9$, $\lambda^2=0.81$, $\lambda^3=0.72$, $\lambda^4=0.66$, $\lambda^5=0.59$, ..., $\lambda^{10}=0.35$, ..., $\lambda^{20}=0.12$, ..., $\lambda^{30}=0.04$, ..., $\lambda^{40}=0.015$, ..., $\lambda^{50}=0.005$, ..., $\lambda^{100}=0.000027$, т.е. параметр λ определяет «память» алгоритма решения рекурсивной задачи по критерию наименьших квадратов. При $0 < \lambda < 1$ и малых значениях разности $k-i$ суммируемые члены взвешиваются с большими весами, а при больших значениях этой разности — с меньшими весами. При $\lambda=1$ эта «память» равна k отсчетам наблюдаемых сигналов. При $0 < \lambda < 1$ вклад одних и тех же i -х отсчетов в сумму является разным при разных λ . Этот вклад больший при больших λ и меньший при меньших λ . То есть, с уменьшением λ эффективная память уменьшается и наоборот.

6.6. Решение рекурсивной задачи наименьших квадратов

Используя уравнение (6.57), целевую функцию (6.58) можно представить как

$$\begin{aligned}
E_N(k) &= \sum_{i=1}^k \lambda^{k-i} e_N(i) e_N^*(i) = \sum_{i=1}^k \lambda^{k-i} [d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i)] [d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i)]^* = \\
&= \sum_{i=1}^k \lambda^{k-i} d(i) d(i)^* - \sum_{i=1}^k \lambda^{k-i} d(i) \mathbf{x}_N^H(i) \mathbf{h}_N(k) - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^H(k) \mathbf{x}_N(i) d^*(i) + \\
&+ \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^H(k) \mathbf{x}_N(i) \mathbf{x}_N^H(i) \mathbf{h}_N(k) = E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) - \mathbf{h}_N^H(k) \mathbf{r}_N + \\
&+ \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{h}_N(k),
\end{aligned} \tag{6.60}$$

где

$$E_d(k) = \sum_{i=1}^k \lambda^{k-i} d(i) d^*(i), \tag{6.61}$$

$$\mathbf{r}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) \tag{6.62}$$

и

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i). \tag{6.63}$$

В дальнейшем для простоты выражения (6.62) и (6.63) будем именовать «вектором взаимной корреляции» и «корреляционной матрицей», хотя, как будет показано далее, лишь при $\lambda \rightarrow 1$ и больших k вектор (6.62) и матрица (6.63) соответствуют вектору взаимной корреляции и корреляционной матрице обрабатываемых сигналов с некоторым коэффициентом.

Аналогично (6.60) в случае действительных ВК

$$\begin{aligned}
E_N(k) &= \sum_{i=1}^k \lambda^{k-i} e_N^2(i) = \sum_{i=1}^k \lambda^{k-i} d^2(i) - 2 \sum_{i=1}^k \lambda^{k-i} \tilde{\mathbf{h}}_N^T(k) \mathbf{x}_N(i) d(i) + \\
&+ \sum_{i=1}^k \lambda^{k-i} \tilde{\mathbf{h}}_N^T(k) \mathbf{x}_N(i) \mathbf{x}_N^T(i) \mathbf{h}_N(k) = E_d(k) - 2 \tilde{\mathbf{h}}_N^T(k) \mathbf{r}_N(k) + \tilde{\mathbf{h}}_N^T(k) \mathbf{R}_N(k) \tilde{\mathbf{h}}_N(k),
\end{aligned} \tag{6.64}$$

где

$$E_d(k) = \sum_{i=1}^k \lambda^{k-i} d^2(i), \tag{6.65}$$

$$\mathbf{r}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d(i) \tag{6.66}$$

и

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^T(i). \tag{6.67}$$

Значение функции (6.60) минимально при

$$\nabla_{\mathbf{h}_N^*} E_N(k) = \frac{\partial E_N(k)}{\partial \mathbf{h}_N^*(k)} = -\mathbf{r}_N(k) + \mathbf{R}_N(k) \mathbf{h}_N(k) = \mathbf{0}_N, \tag{6.68}$$

откуда

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k), \tag{6.69}$$

а значение функции (6.64) минимально при

$$\nabla_{\tilde{\mathbf{h}}_N} E_N(k) = \frac{\partial E_N(k)}{\partial \tilde{\mathbf{h}}_N(k)} = -2\mathbf{r}_N(k) + 2\mathbf{R}_N(k)\tilde{\mathbf{h}}_N(k) = \mathbf{0}_N, \quad (6.70)$$

откуда

$$\tilde{\mathbf{h}}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{r}_N(k). \quad (6.71)$$

Минимальные значения $E_{N,\min}(k)$ функций (6.60) и (6.64) определяются по аналогии (6.16) и (6.17) путем подстановки в (6.60) и (6.64) выражений $\mathbf{r}_N(k) = \mathbf{R}_N(k)\mathbf{h}_N(k)$ и $\mathbf{r}_N(k) = \mathbf{R}_N(k)\tilde{\mathbf{h}}_N(k)$, определяемых уравнениями (6.68) и (6.70). Эти значения равны соответственно

$$E_{N,\min}(k) = E_d(k) - \mathbf{r}_N^H(k)\mathbf{h}_N(k) \quad (6.72)$$

и

$$E_{N,\min}(k) = E_d(k) - \mathbf{r}_N^T(k)\tilde{\mathbf{h}}_N(k). \quad (6.73)$$

Рекурсивное решение LS-задачи минимизации функций (6.60) и (6.64), т. е. Recursive Least Squares (RLS), осуществляется следующим образом.

Сначала, подобно (6.59), представим уравнения (6.62) и (6.63) с помощью рекуррентных соотношений

$$\mathbf{r}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) = \lambda \mathbf{r}_N(k-1) + \mathbf{x}_N(k) d^*(k) \quad (6.74)$$

и

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k). \quad (6.75)$$

Тогда, используя уравнения (6.74) и (6.75), можно для каждого значения k решать LS-задачу, т. е. вычислять $\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{r}_N(k)$, не вычисляя $\mathbf{R}_N^{-1}(k)$ каждый раз заново.

В этом и заключается суть рекурсивного решения LS-задачи путем минимизации функций (6.60) и (6.64). Это решение базируется на рекурсивном обращении матрицы (6.75) с помощью MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где $\mathbf{C} = \lambda \mathbf{R}_N(k-1)$, $\mathbf{C}^{-1} = \lambda^{-1} \mathbf{R}_N^{-1}(k-1)$, $\mathbf{B}^H = \mathbf{x}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k)$ и $\mathbf{A}^{-1} = -1$.

Для этого, используя уравнение (6.75), определим матрицу $\mathbf{R}_N^{-1}(k)$ как

$$\begin{aligned} \mathbf{R}_N^{-1}(k) &= \lambda^{-1} \mathbf{R}_N^{-1}(k-1) - \frac{\lambda^{-1} \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \lambda^{-1} \mathbf{R}_N^{-1}(k-1)}{1 + \mathbf{x}_N^H(k) \lambda^{-1} \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)} = \\ &= \lambda^{-1} \mathbf{R}_N^{-1}(k-1) - \frac{\lambda^{-2} \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)}{\lambda^{-1} [\lambda + \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k)]} = \\ &= \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)], \end{aligned} \quad (6.76)$$

где вектор

$$\mathbf{g}_N(k) = \frac{\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)}{\lambda + \mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)} \quad (6.77)$$

называется вектором коэффициентов Калмана.

Из выражения (6.77) следует, что $\mathbf{g}_N(k)[\lambda + \mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)] = \mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)$, а используя выражение (6.76), — что

$$\begin{aligned} \mathbf{g}_N(k) &= \lambda^{-1} [\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)] = \\ &= \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)]\mathbf{x}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k). \end{aligned} \quad (6.78)$$

Таким образом, используя матрицу $\mathbf{R}_N^{-1}(k)$ и вектор $\mathbf{x}_N(k)$, вектор коэффициентов Калмана можно определять как

$$\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k), \quad (6.79)$$

а используя матрицу $\mathbf{R}_N^{-1}(k-1)$ и вектор $\mathbf{x}_N(k)$ — как (6.77). Вектор коэффициентов Калмана играет фундаментальную роль в RLS-алгоритмах, включая их быстрые (вычислительно-эффективные) версии, где он используется в качестве переменной, участвующей в рекурсивном вычислении вектора ВК $\mathbf{h}_N(k)$.

Для рекурсивного вычисления $\mathbf{h}_N(k)$ в уравнении (6.69) необходимо использовать выражения (6.74) и (6.76). Тогда

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{R}_N^{-1}(k)\mathbf{r}_N(k) = \mathbf{R}_N^{-1}(k)[\lambda \mathbf{r}_N(k-1) + \mathbf{x}_N(k)d^*(k)] = \\ &= \lambda \mathbf{R}_N^{-1}(k)\mathbf{r}_N(k-1) + \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)d^*(k) = \\ &= \lambda \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)]\mathbf{r}_N(k-1) + \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)d^*(k) = \\ &= \mathbf{R}_N^{-1}(k-1)\mathbf{r}_N(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{r}_N(k-1) + \\ &+ \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)d^*(k) = \mathbf{h}_N(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{h}_N(k-1) + \mathbf{g}_N(k)d^*(k) = \\ &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)[d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k-1)] = \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k), \end{aligned} \quad (6.80)$$

где

$$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k) \quad (6.81)$$

— априорная ошибка оценивания процесса $\{d(i)\}$ (требуемого сигнала), поскольку это ошибка между наблюдаемыми значениями этого процесса и значениями выходного сигнала трансверсального фильтра $\{y(i)\}$, измеренная перед обновлением вектора ВК фильтра, т. е. на основе ВК $\mathbf{h}_N(k-1)$, см. также уравнение (1.2). Апостериорная ошибка, см. уравнение (1.3), оценивания процесса $\{d(i)\}$ определяется как

$$e_N(k) = d(k) - \mathbf{h}_N^H(k)\mathbf{x}_N(k) \quad (6.82)$$

после вычисления вектора текущего значения ВК с помощью уравнения (6.80).

Можно также получить уравнение для рекурсивного вычисления минимального значения энергии апостериорных ошибок (6.72). Для этого подобно уравнению (6.59) определим

$$E_d(k) = \sum_{i=1}^k \lambda^{k-i} d(i) d^*(i) = \lambda E_d(k-1) + d(k) d^*(k), \quad (6.83)$$

а также используем уравнение (6.80). Тогда согласно уравнению (6.72)

$$\begin{aligned} E_{N,\min}(k) &= E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) = \lambda E_d(k-1) + d(k) d^*(k) - \\ &- \mathbf{r}_N^H(k) [\mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)] = \lambda E_d(k-1) + d(k) d^*(k) - \\ &- [\lambda \mathbf{r}_N^H(k-1) + d(k) \mathbf{x}_N^H(k)] \mathbf{h}_N(k-1) - \mathbf{r}_N^H(k) \mathbf{g}_N(k) \alpha_N^*(k) = \\ &= \lambda E_d(k-1) + d(k) d^*(k) - \lambda \mathbf{r}_N^H(k-1) \mathbf{h}_N(k-1) - d(k) \mathbf{x}_N^H(k) \mathbf{h}_N(k-1) - \\ &- \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \alpha_N^*(k) = \lambda E_{N,\min}(k-1) + \\ &+ d(k) [d^*(k) - \mathbf{x}_N^H(k) \mathbf{h}_N(k-1)] - \mathbf{h}_N^H(k) \mathbf{x}_N(k) \alpha_N^*(k) = \\ &= \lambda E_{N,\min}(k-1) + d(k) \alpha_N^*(k) - \mathbf{h}_N^H(k) \mathbf{x}_N(k) \alpha_N^*(k) = \\ &= \lambda E_{N,\min}(k-1) + [d(k) - \mathbf{h}_N^H(k) \mathbf{x}_N(k)] \alpha_N^*(k) = \lambda E_{N,\min}(k-1) + e_N(k) \alpha_N^*(k). \end{aligned} \quad (6.84)$$

Здесь также использованы уравнения (6.68) $\mathbf{r}_N(k) = \mathbf{R}_N(k) \mathbf{h}_N(k)$, (3.7) $\mathbf{R}_N^H(k) = \mathbf{R}_N(k)$, (6.74) $\mathbf{r}_N(k) = \lambda \mathbf{r}_N(k-1) + \mathbf{x}_N(k) d^*(k)$ и (6.79) $\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)$.

Чтобы значения уравнения (6.84) оставались действительными положительными величинами, необходимо, чтобы результаты произведения комплексных чисел $e_N(k) \alpha_N^*(k)$ были действительными, т. е.

$$\begin{aligned} &[\operatorname{Re}\{e_N(k)\} + j \operatorname{Im}\{e_N(k)\}] [\operatorname{Re}\{\alpha_N(k)\} - j \operatorname{Im}\{\alpha_N(k)\}] = \\ &= \operatorname{Re}\{e_N(k)\} \operatorname{Re}\{\alpha_N(k)\} + \operatorname{Im}\{e_N(k)\} \operatorname{Im}\{\alpha_N(k)\} + j \operatorname{Im}\{e_N(k)\} \operatorname{Re}\{\alpha_N(k)\} - \\ &- j \operatorname{Re}\{e_N(k)\} \operatorname{Im}\{\alpha_N(k)\} = \operatorname{Re}\{e_N(k)\} \operatorname{Re}\{\alpha_N(k)\} + \operatorname{Im}\{e_N(k)\} \operatorname{Im}\{\alpha_N(k)\}. \end{aligned} \quad (6.85)$$

Это означает, что $j \operatorname{Im}\{e_N(k)\} \operatorname{Re}\{\alpha_N(k)\} - j \operatorname{Re}\{e_N(k)\} \operatorname{Im}\{\alpha_N(k)\} = 0$ или

$$\operatorname{Im}\{e_N(k)\} \operatorname{Re}\{\alpha_N(k)\} = \operatorname{Re}\{e_N(k)\} \operatorname{Im}\{\alpha_N(k)\}. \quad (6.86)$$

Если равенство (6.86) верно, то также верно и равенство

$$E_{N,\min}(k) = \lambda E_{N,\min}(k-1) + \alpha_N(k) e_N^*(k), \quad (6.87)$$

так как в этом случае произведение

$$\begin{aligned} \alpha_N(k) e_N^*(k) &= [\operatorname{Re}\{\alpha_N(k)\} + j \operatorname{Im}\{\alpha_N(k)\}] [\operatorname{Re}\{e_N(k)\} - j \operatorname{Im}\{e_N(k)\}] \\ &= \operatorname{Re}\{\alpha_N(k)\} \operatorname{Re}\{e_N(k)\} + \operatorname{Im}\{\alpha_N(k)\} \operatorname{Im}\{e_N(k)\} + j \operatorname{Im}\{\alpha_N(k)\} \operatorname{Re}\{e_N(k)\} - \\ &- j \operatorname{Re}\{\alpha_N(k)\} \operatorname{Im}\{e_N(k)\} = \operatorname{Re}\{\alpha_N(k)\} \operatorname{Re}\{e_N(k)\} + \operatorname{Im}\{\alpha_N(k)\} \operatorname{Im}\{e_N(k)\} \end{aligned} \quad (6.88)$$

также является действительным.



Таким образом, основываясь на соотношениях, приведенных в настоящем разделе, RLS-алгоритм минимизации среднеквадратичного значения ошибок между процессами $\{d(i)\}$ и $\{y(i)\}$ можно представить, как это показано в табл. 6.1.

Таблица 6.1. RLS-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0)=\mathbf{0}_N$, $\mathbf{R}_N^{-1}(0)=\delta^{-2}\mathbf{I}_N$, $\mathbf{h}_N(0)=\mathbf{0}_N$	(6.1.0)
For $k = 1, 2, \dots, K$	
$\mathbf{g}_N(k) = \frac{\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)}{\lambda + \mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{x}_N(k)}$	(6.1.1)
$\mathbf{R}_N^{-1}(k) = \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)]$	(6.1.2)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$	(6.1.3)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k)$	(6.1.4)
End for k	

Инициализация вектора ВК в RLS-алгоритме (см. табл. 6.1) осуществляется как $\mathbf{h}_N(0)=\mathbf{0}_N$, так как при $k=0$ переменные $\mathbf{x}_N(0)=\mathbf{0}_N$ и $d(0)=0$ вектор $\mathbf{r}_N(0)=\mathbf{x}_N(0)d^*(0)=\mathbf{0}_N$, а значит, и $\mathbf{h}_N(0)=\mathbf{R}_N^{-1}(0)\mathbf{r}_N(0)=\mathbf{0}_N$.

С помощью уравнения (6.1.2) матрица $\mathbf{R}_N^{-1}(k)$ вычисляется рекурсивно путем использования матрицы $\mathbf{R}_N^{-1}(k-1)$, вычисленной на предыдущей итерации, и текущего значения вектора $\mathbf{x}_N(k)$. Так как итерации начинаются с $k=1$, то необходимо задать начальное значение $\mathbf{R}_N^{-1}(0)$. Вопросы выбора начального значения $\mathbf{R}_N^{-1}(0)$ будут подробно рассмотрены в разделе 14.8, а пока можно предположить, что инициализация матрицы $\mathbf{R}_N(k)$ осуществляется как

$$\mathbf{R}_N(0)=\delta^2\mathbf{I}_N, \quad (6.89)$$

а значит, и инициализация обратной матрицы — как

$$\mathbf{R}_N^{-1}(0)=\delta^{-2}\mathbf{I}_N. \quad (6.90)$$

Чтобы понять, как экспоненциальное взвешивание и инициализация (6.89) влияют на матрицу $\mathbf{R}_N(k)$, рассмотрим уравнение (6.75) для нескольких значений k .

При $k=1$

$$\mathbf{R}_N(1)=\lambda\mathbf{R}_N(0)+\mathbf{x}_N(1)\mathbf{x}_N^H(1)=\lambda\delta^2\mathbf{I}_N+\mathbf{x}_N(1)\mathbf{x}_N^H(1).$$

При $k=2$

$$\begin{aligned} \mathbf{R}_N(2) &= \lambda\mathbf{R}_N(1)+\mathbf{x}_N(2)\mathbf{x}_N^H(2)=\lambda[\mathbf{x}_N(1)\mathbf{x}_N^H(1)+\lambda\delta^2\mathbf{I}_N]+\mathbf{x}_N(2)\mathbf{x}_N^H(2)= \\ &= \lambda\mathbf{x}_N(1)\mathbf{x}_N^H(1)+\mathbf{x}_N(2)\mathbf{x}_N^H(2)+\lambda^2\delta^2\mathbf{I}_N. \end{aligned}$$

При $k=3$

$$\begin{aligned}\mathbf{R}_N(3) &= \lambda \left[\lambda \mathbf{x}_N(1) \mathbf{x}_N^H(1) + \mathbf{x}_N(2) \mathbf{x}_N^H(2) + \lambda^2 \delta^2 \mathbf{I}_N \right] + \mathbf{x}_N(3) \mathbf{x}_N^H(3) = \\ &= \lambda^2 \mathbf{x}_N(1) \mathbf{x}_N^H(1) + \lambda \mathbf{x}_N(2) \mathbf{x}_N^H(2) + \mathbf{x}_N(3) \mathbf{x}_N^H(3) + \lambda^3 \delta^2 \mathbf{I}_N.\end{aligned}$$

Из этих уравнений следует, что для любого k справедливо соотношение

$$\mathbf{R}_N(k) = \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right] + \lambda^k \delta^2 \mathbf{I}_N. \quad (6.91)$$

При $k \rightarrow \infty$ матрица $\lambda^k \delta^2 \mathbf{I}_N \rightarrow \mathbf{0}_N$, так как $\lambda < 1$ и множитель $\lambda^k \rightarrow 0$, а $\mathbf{R}_N(k) \rightarrow \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i)$. Следовательно, влияние инициализации $\mathbf{R}_N(0) = \delta^2 \mathbf{I}_N$ на значение матрицы $\mathbf{R}_N(k)$ уменьшается с увеличением k . То есть инициализация мало сказывается на поведении RLS-алгоритма в установившемся режиме (при больших значениях k). Кроме того, при достаточно малых значениях δ^2 влияние $\mathbf{R}_N(0) = \delta^2 \mathbf{I}_N$ мало сказывается и на переходном процессе RLS-алгоритма. Согласно [80] на практике обычно используется значение $\delta^2 > 0.01\sigma_x^2$, где σ_x^2 — дисперсия входного сигнала $x(k)$.

Наличие второго слагаемого в уравнении (6.91) приводит к смещению вектора ВК, вычисляемого при использовании матрицы (6.91) относительно несмещенного вектора ВК, вычисляемого с помощью матрицы (6.67). Чтобы оценить это смещение, используем матрицу (6.91) в уравнении (6.68), т. е.

$$\mathbf{R}_N(k) \mathbf{h}_N(k) = \left\{ \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right] + \lambda^k \delta^2 \mathbf{I}_N \right\} \mathbf{h}_N(k) = \mathbf{r}_N(k). \quad (6.92)$$

Затем, умножив обе части уравнения (6.92) слева на матрицу $\left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1}$, получим, что

$$\begin{aligned}\left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1} \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right] \mathbf{h}_N(k) + \\ + \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1} \lambda^k \delta^2 \mathbf{h}_N(k) = \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1} \mathbf{r}_N(k),\end{aligned} \quad (6.93)$$

так как $\mathbf{I}_N \mathbf{h}_N(k) = \mathbf{h}_N(k)$.

Поскольку сигнал $x(k)$, формирующий векторы $\mathbf{x}_N(k)$, и сигнал $d(k)$ являются отсчетами эргодических случайных процессов (см. раздел 3.4), то для анализа уравнения (6.93) усреднение по ансамблю реализаций можно заменить усреднением по времени. Тогда при $k \rightarrow \infty$ и $\lambda \rightarrow 1$ получим, что

$$\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \approx \sum_{i=1}^k \lambda^{k-i} E \left\{ \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right\} = \sum_{i=1}^k \lambda^{k-i} \mathbf{R}_N = \mathbf{R}_N \sum_{i=1}^k \lambda^{k-i}, \quad (6.94)$$

см. уравнение (3.5).

Аналогично

$$\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) \approx \sum_{i=1}^k \lambda^{k-i} E\{\mathbf{x}_N(i) d^*(i)\} = \sum_{i=1}^k \lambda^{k-i} \mathbf{r}_N = \mathbf{r}_N \sum_{i=1}^k \lambda^{k-i}. \quad (6.95)$$

Используя формулу суммы k членов геометрической прогрессии (4.5) для определения значения

$$\sum_{i=1}^k \lambda^{k-i} = \lambda^{k-1} + \lambda^{k-2} + \dots + \lambda^2 + \lambda^1 + \lambda^0 = \frac{a_1 - a_k q}{1 - q} = \frac{1 - \lambda^{k-1} \lambda}{1 - \lambda} = \frac{1 - \lambda^k}{1 - \lambda}, \quad (6.96)$$

где $a_1 = \lambda^0 = 1$ — первый член, $a_k = \lambda^{k-1}$ — последний член и $q = \frac{a_k}{a_{k-1}} = \frac{\lambda^{k-1}}{\lambda^{k-2}} = \lambda$ — знаменатель геометрической прогрессии, уравнения (6.94) и (6.95) можно преобразовать к виду

$$\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \approx \mathbf{R}_N \sum_{i=1}^k \lambda^{k-i} = \frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \quad (6.97)$$

и

$$\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) \approx \mathbf{r}_N \sum_{i=1}^k \lambda^{k-i} = \frac{1 - \lambda^k}{1 - \lambda} \mathbf{r}_N, \quad (6.98)$$

а уравнение (6.93) — к виду

$$\begin{aligned} & \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right] \mathbf{h}_N(k) + \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \lambda^k \delta^2 \mathbf{h}_N(k) = \\ & = \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{r}_N \right]. \end{aligned} \quad (6.99)$$

Из уравнения (6.99) следует, что

$$\mathbf{h}_N(k) + \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \lambda^k \delta^2 \mathbf{h}_N(k) = \mathbf{h}_{N,o}, \quad (6.100)$$

а из уравнения (6.100) — что смещение вектора ВК, обусловленное начальной инициализацией корреляционной матрицы, определяется как

$$\Delta \mathbf{h}_N(k) = \mathbf{h}_N(k) - \mathbf{h}_{N,o}(k) = - \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \lambda^k \delta^2 \mathbf{h}_N(k). \quad (6.101)$$

Так как в рассматриваемом случае (т. е. при $k \rightarrow \infty$ и $\lambda \rightarrow 1$) вектор $\mathbf{h}_N(k) \approx \mathbf{h}_{N,o}$, что следует из соотношения

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) \approx \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{R}_N \right]^{-1} \left[\frac{1 - \lambda^k}{1 - \lambda} \mathbf{r}_N \right] = \\ &= \left[\frac{1 - \lambda^k}{1 - \lambda} \right]^{-1} \left[\frac{1 - \lambda^k}{1 - \lambda} \right] \mathbf{R}_N^{-1} \mathbf{r}_N = \mathbf{R}_N^{-1} \mathbf{r}_N = \mathbf{h}_{N,o}, \end{aligned} \quad (6.102)$$

то для оценки смещения вектора ВК $\Delta \mathbf{h}_N(k)$ за счет инициализации (6.89) или (6.90) в правой части уравнения (6.101) можно воспользоваться соотношением (6.102). Тогда уравнение (6.101) преобразуется к виду

$$\Delta \mathbf{h}_N(k) \approx - \left[\frac{1-\lambda^k}{1-\lambda} \mathbf{R}_N \right]^{-1} \lambda^k \delta^2 \mathbf{h}_{N,o} = \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1} \lambda^k \delta^2 \mathbf{h}_{N,o}. \quad (6.103)$$

Из уравнения (6.103) следует, что инициализация $\mathbf{R}_N(0) = \delta^2 \mathbf{I}_N$ приводит к смещению значений вектора ВК $\mathbf{h}_N(k)$ на величину $\left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right]^{-1} \lambda^k \delta^2 \mathbf{h}_{N,o}$. Однако в установившемся режиме (при больших k) это смещение стремится к $\mathbf{0}_N$, так как $\lambda^k \rightarrow 0$, и в результате итерационного выполнения RLS-алгоритма находится вектор $\mathbf{h}_N(k) \approx \mathbf{h}_{N,o}$, см. уравнение (6.102).

Алгоритм, приведенный в табл. 6.1, называется RLS-алгоритмом адаптивной фильтрации. С его помощью минимизируется среднеквадратичное значение ошибок между процессами $\{d(i)\}$ и $\{y(i)\}$. Природа процесса $\{d(i)\}$ определяется приложением адаптивного фильтра (см. главу 1). Например, в эхокомпенсаторах — это эхо-сигнал, в эквалайзерах каналов связи — тренировочная последовательность известных информационных символов или последовательность оценок неизвестных принимаемых символов (собственно, передаваемой информации), получаемых с выходного сигнала эквалайзера после обучения, т.е. когда предполагается, что эти символы являются достоверными.

Для получения математических соотношений RLS-алгоритма адаптивной фильтрации действительных сигналов можно воспользоваться преобразованиями (6.64)–(6.81), сделав тривиальный переход от арифметики действительных чисел к арифметике комплексных чисел, т.е. заменив операции эрмитова сопряжения векторов и матриц операциями обычного транспонирования и опустив операцию комплексного сопряжения скалярных переменных. В этом случае преобразования (6.64)–(6.81) приводят к алгоритму (см. табл. 6.1), в котором операции комплексной арифметики просто заменены на операции действительной арифметики.

Таким образом, в отличие от градиентных алгоритмов, рассмотренных в главах 4 и 5, в RLS-алгоритме переход от комплексной арифметики и наоборот не сопровождается появлением и исчезновением в вычислениях множителя, равного 2. Этот множитель, см. уравнение (6.70), исчезает в уравнении (6.71) и в дальнейших преобразованиях не участвует.

Вычислительная сложность RLS-алгоритма (см. табл. 6.1) пропорциональна $O(N^2)$ арифметическим операциям (комплексным или действительным, в зависимости от арифметики). Из этой таблицы следует, что вычислительная сложность RLS-алгоритма равна $3N^2 + 3N$ операциям умножения, $2N^2 + 3N$ операциям сложения и одной операции деления. Здесь учитывается, что произведение $\mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)$ можно не вычислять, так как $[\mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k)]^H = \mathbf{x}_N^H(k) \mathbf{R}_N^{-H}(k-1) = \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)$. То есть, вместо вычисления вектора $\mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)$ можно использовать эрмитово-сопряженное значение ранее вычисленного вектора $\mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k)$.



Из-за ошибок округления при вычислениях матрица (6.76) может потерять симметрию [97], в результате чего может нарушиться соотношение (3.7) $\mathbf{R}_N = \mathbf{R}_N^H$, а алгоритм (см. табл. 6.1) — стать неустойчивым. Простым решением этой проблемы является вычисление в уравнении (6.1.2) только диагональных элементов и верхней или нижней треугольной части матрицы $\mathbf{R}_N^{-1}(k)$ с последующим их копированием соответственно в нижнюю или верхнюю часть этой матрицы и комплексным сопряжением в случае обработки комплексных сигналов $x(k)$. Этот прием позволяет сэкономить в уравнении (6.1.2) $2(N^2 - N)/2 = N^2 - N$ умножений и $(N^2 - N)/2$ сложений. В этом случае вычислительная сложность RLS-алгоритма (см. табл. 6.1) будет равна $2N^2 + 4N$ операциям умножения, $1.5N^2 + 3.5N$ операциям сложения и одной операции деления.

Таким образом, если сравнить сложность RLS-алгоритма со сложностью прямого решения LS-задачи (6.22) $\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)$, то можно заметить, что выигрыш от рекурсивного вычисления матрицы $\mathbf{R}_N^{-1}(k)$ по сравнению с ее обращением с помощью любого из большинства известных алгоритмов обращения матриц пропорционален N , так как эти алгоритмы, как правило, требуют $O(N^3)$ арифметических операций, тогда как рассмотренный RLS-алгоритм требует $O(N^2)$ арифметических операций на одну итерацию. Разумеется, сложность $O(N^2)$ также может оказаться большой, начиная с некоторых N . Поэтому семейство RLS-алгоритмов не ограничивается только рассмотренным алгоритмом на основе MIL. Эффективные алгоритмы вычисления вектора $\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)$ могут базироваться также на использовании прямого или обратного QR-разложения, на использовании так называемых быстрых, т. е. вычислительно-эффективных, процедур, включая и лестничные алгоритмы. Такие алгоритмы рассматриваются в следующих главах.

6.7. Качество адаптивной фильтрации с помощью RLS-алгоритма

На рис. 6.6 и рис. 6.7 показаны результаты сравнительного моделирования LMS- и RLS-алгоритмов при решении задачи линейного предсказания авторегрессионного процесса второго порядка в условиях, аналогичных представленным на рис. 5.2 и рис. 5.10.

Из рис. 6.6 видно, что RLS-алгоритм демонстрирует существенно большую скорость сходимости ВК к оптимальным значениям по сравнению с алгоритмом наискорейшего спуска и LMS-алгоритмом (см. рис. 5.2). На рис. 6.6 темной линией показан переходной процесс ВК LMS-алгоритма (при $\mu = 0.005$), а светлой — RLS-алгоритма при $\lambda = 0.9999$.

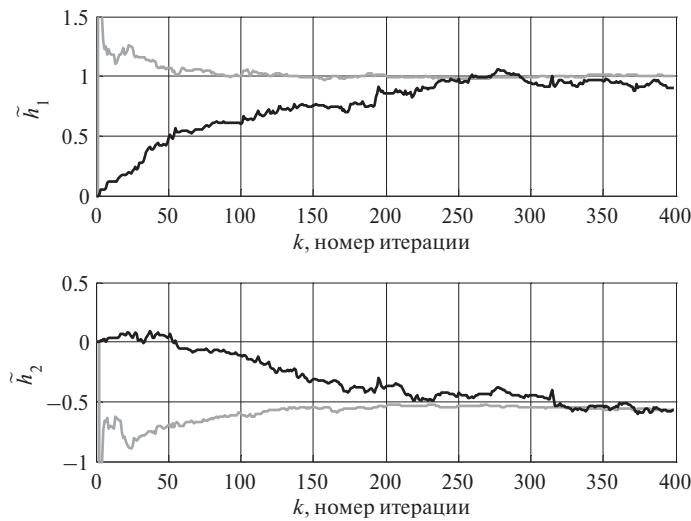


Рис. 6.6. Переходные процессы ВК: одна реализация

Из рис. 6.7 также следует, что RLS-алгоритм демонстрирует наибольшую скорость сходимости в терминах обучающей кривой, т.е. среднеквадратической ошибки $\xi(k) = E\{\alpha_N(k)\alpha_N^*(k)\}$ моделирования требуемого сигнала $d(k)$, и наименьшее значение остаточной ошибки ξ_{\min} в установившемся режиме. Причем для обеспечения этих свойств в RLS-алгоритме не требуется такой неоднозначный параметр, как шаг сходимости μ , используемый в LMS-алгоритме.

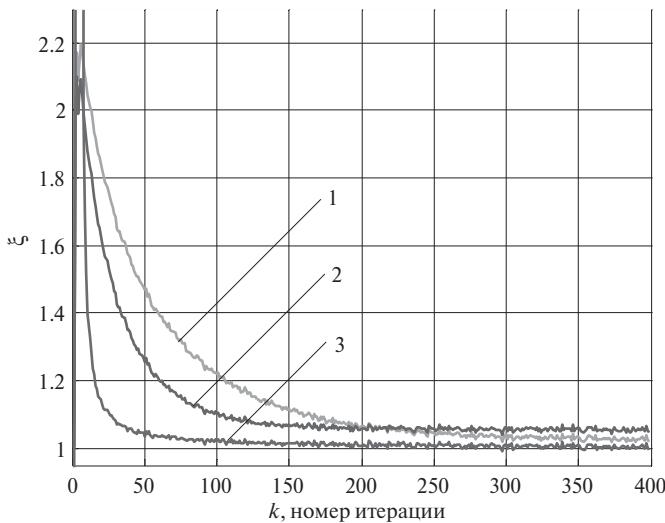


Рис. 6.7. Обучающие кривые при усреднении по 100000 реализациям: 1 — LMS-алгоритм, $\mu = 0.005$; 2 — LMS-алгоритм, $\mu = 0.01$; 3 — RLS-алгоритм, $\lambda = 0.9999$

На рис. 6.8 показаны переходные процессы обучающей кривой сравниваемых LMS- и RLS-алгоритмов при двух соотношениях собственных чисел χ корреляционной матрицы \mathbf{R}_N входного сигнала $x(k)$. Для LMS-алгоритма шаг сходимости выбран в пределах допустимых значений (см. главу 5).

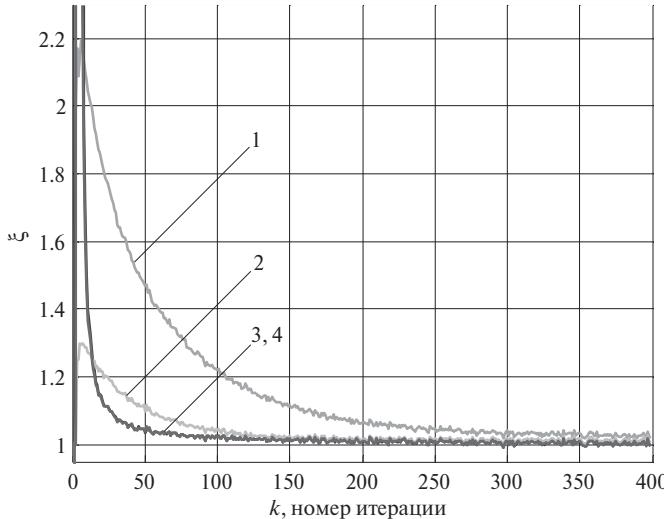


Рис. 6.8. Обучающие кривые при усреднении по 100000 реализациям: 1 — LMS-алгоритм, $\mu = 0.005$, $\chi = 5$; 2 — LMS-алгоритм, $\mu = 0.005$, $\chi = 1$; 3 — RLS-алгоритм, $\lambda = 0.9999$, $\chi = 5$, 4 — RLS-алгоритм, $\lambda = 0.9999$, $\chi = 1$

Из рис. 6.8 следует, что с увеличением χ сходимость LMS-алгоритма ухудшается, т. е. увеличивается длительность переходного процесса и увеличивается ξ_{exc} , см. уравнение (5.147), в то время как длительность переходного процесса и ξ_{exc} в RLS-алгоритме не зависят от значения χ , см. также рис. 6.9.

Это происходит оттого, что вектор коэффициентов Калмана $\mathbf{g}_N(k)$ фактически выступает в качестве N различных переменных шагов сходимости (т. е. для каждого ВК используется свой шаг сходимости, обновляемый на каждой k -й итерации), зависящих согласно уравнению (6.79) от всех обрабатываемых сигналов как $\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)$.

Поэтому можно провести некоторую аналогию между LMS-, NLMS- и RLS-алгоритмами, где первые два алгоритма можно рассматривать как результат упрощения последнего алгоритма [47]. Действительно, согласно уравнениям (6.79) и (6.80) вектор ВК в RLS-алгоритме определяется как

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k) = \mathbf{h}_N(k-1) + \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \alpha_N^*(k). \quad (6.104)$$

Если в (6.104) матрицу $\mathbf{R}_N^{-1}(k)$ заменить ее оценкой на одном отсчете обрабатываемых сигналов $\hat{\mathbf{R}}_N(k) = \mathbf{x}_N(k) \mathbf{x}_N^H(k)$ и использовать регуляризацию для

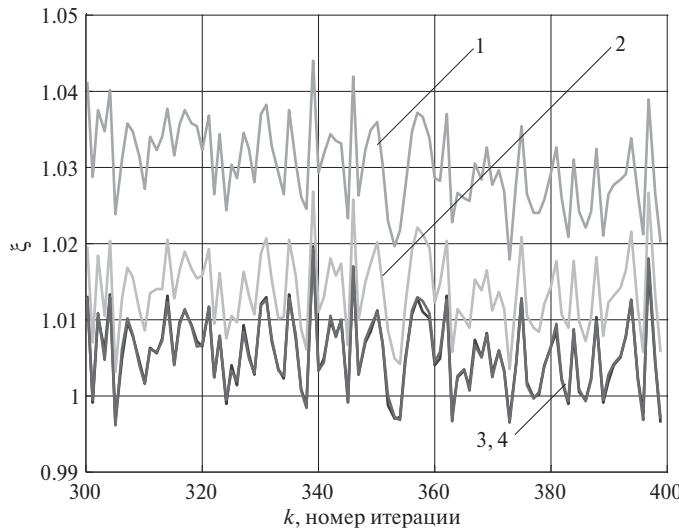


Рис. 6.9. Обучающие кривые: рис. 6.8 в увеличенном масштабе

обращения этой матрицы, то (6.104) становится уравнением вычисления ВК в NLMS-алгоритме, т. е.

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \hat{\mathbf{R}}_N^{-1}(k) \mathbf{x}_N(k) \alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha_N^*(k). \end{aligned} \quad (6.105)$$

Если с целью дальнейшего упрощения уравнения (6.105) множитель

$$\frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2}$$

заменить на

$$\mu = \frac{\hat{\mu}}{\sum_{n=1}^N x^*(k-n+1) x(k-n+1)} \leq 2/\lambda_{\max},$$

то получим уравнение вычисления ВК в LMS-алгоритме, т. е.

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\sum_{n=1}^N x^*(k-n+1) x(k-n+1)} \mathbf{x}_N(k) \alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + \frac{2}{\lambda_{\max}} \mathbf{x}_N(k) \alpha_N^*(k) = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha_N^*(k). \end{aligned} \quad (6.106)$$

Таким образом, преобразования (6.104)–(6.106) показывают, что NLMS- и LMS-алгоритмы можно рассматривать как результат последовательного упрощения RLS-алгоритма. А раз это так, то не следует ожидать, что характеристи-



ки упрощенных алгоритмов в терминах длительности переходных процессов и остаточных ошибок в установившемся режиме будут лучше, чем у исходного, более эффективного, но и более сложного RLS-алгоритма.

Квадратичная вычислительная сложность RLS-алгоритмов является ценой их более высокой эффективности по сравнению с вычислительно-простыми алгоритмами на базе стратегий градиентного спуска, что было продемонстрировано на рис. 6.6—6.9.

Для анализа сходимости RLS-алгоритма, используя уравнение (6.1), представим требуемый сигнал как

$$d(k) = \sum_{n=1}^N h_{n,o}^* x(k-n+1) + e_o(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k). \quad (6.107)$$

Уравнение (6.107) в векторном виде представляется как

$$\mathbf{d}_k(k) = [d(1), d(2), \dots, d(N), \dots, d(k-1), d(k)]^T = \mathbf{X}_{Nk}^T(k) \mathbf{h}_{N,o}^* + \mathbf{e}_{k,o}(k), \quad (6.108)$$

где

$$\begin{aligned} \mathbf{X}_{Nk}(k) &= [\mathbf{x}_N(1), \mathbf{x}_N(2), \dots, \mathbf{x}_N(N), \dots, \mathbf{x}_N(k-1), \mathbf{x}_N(k)] = \\ &= \begin{bmatrix} x(1) & x(2) & \cdots & x(N) & \cdots & x(k-1) & x(k) \\ 0 & x(1) & \cdots & x(N-1) & \cdots & x(k-2) & x(k-1) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & x(1) & \cdots & x(k-N) & x(k-N+1) \end{bmatrix} \end{aligned} \quad (6.109)$$

и

$$\mathbf{e}_{k,o} = [e_o(1), e_o(2), \dots, e_o(k)]^T. \quad (6.110)$$

Уравнение (6.58) в этом случае принимает вид

$$E_N(k) = \mathbf{e}_k^T(k) \Lambda_k(k) \mathbf{e}_k^*(k), \quad (6.111)$$

где диагональная матрица $\Lambda_k(k)$ определяется как

$$\Lambda_k(k) = \text{diag}\{\lambda^{k-1}, \lambda^{k-2}, \dots, \lambda, 1\} = \begin{bmatrix} \lambda^{k-1} & 0 & \cdots & 0 & 0 \\ 0 & \lambda^{k-2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & \lambda & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \quad (6.112)$$

Матрицу коэффициентов $\Lambda_k(k)$ с числом элементов $k \times k$ не следует путать с матрицей (3.42), т. е. Λ_N — матрицей собственных чисел с числом элементов $N \times N$.

Следуя приемам раздела 6.3, уравнение (6.69) можно представить как

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) = [\mathbf{X}_{Nk}(k) \Lambda_k(k) \mathbf{X}_{Nk}^H(k)]^{-1} \mathbf{X}_{Nk}(k) \Lambda_k(k) \mathbf{d}_k^*, \quad (6.113)$$

где, см. уравнения (6.62) и (6.63),

$$\mathbf{r}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) = \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{d}_k^* \quad (6.114)$$

и

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k). \quad (6.115)$$

Используя уравнение (6.108) в (6.113), получим, что

$$\begin{aligned} \mathbf{h}_N(k) &= \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \left[\mathbf{X}_{Nk}^T(k) \mathbf{h}_{N,o}^* + \mathbf{e}_{k,o}^*(k) \right]^* = \\ &= \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right] \mathbf{h}_{N,o} + \\ &\quad + \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k) = \\ &= \mathbf{h}_{N,o} + \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k). \end{aligned} \quad (6.116)$$

С учетом независимости $\mathbf{X}_{Nk}(k)$ и $\mathbf{e}_{k,o}^*(k)$ и того, что $E\{\mathbf{e}_{k,o}^*(k)\} = \mathbf{0}_N$, см. также уравнение (6.29), среднее значение вектора ВК (6.116) определяется как

$$E\{\mathbf{h}_N(k)\} = \mathbf{h}_{N,o}, \quad (6.117)$$

т.е. RLS-алгоритм обеспечивает нахождение несмещенной оценки вектора $\mathbf{h}_{N,o}$, или оптимального винеровского решения.

Отклонение вектора $\mathbf{h}_N(k)$ от оптимального решения $\mathbf{h}_{N,o}$, обусловленное шумом $e_o(k)$, согласно уравнению (6.116) определяется как

$$\begin{aligned} \Delta \mathbf{h}_N(k) &= \mathbf{h}_N(k) - \mathbf{h}_{N,o}(k) = \mathbf{h}_{N,o} + \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k) - \mathbf{h}_{N,o}(k) = \\ &= \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k). \end{aligned} \quad (6.118)$$

С учетом независимости $\mathbf{X}_{Nk}(k)$ и $\mathbf{e}_{k,o}^*(k)$, а также того, что $E\{\mathbf{e}_{k,o}^*(k) \mathbf{e}_{k,o}^T(k)\} = \sigma_o^2 \mathbf{I}_k$ ковариационная матрица вектора (6.116), или, что то же самое, корреляционная матрица вектора (6.118), определяется аналогично выражению (6.30) как

$$\begin{aligned} \hat{\mathbf{K}}_N(k) &= \text{cov}\{\mathbf{h}_N(k)\} = E\{\Delta \mathbf{h}_N(k) \Delta \mathbf{h}_N^H(k)\} = \\ &= E\left\{ \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k) \times \right. \\ &\quad \times \left. \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{e}_{k,o}^*(k) \right]^H \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \right\} = \\ &= \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) E\{\mathbf{e}_{k,o}^*(k) \mathbf{e}_{k,o}^T(k)\} \times \\ &\quad \times \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} = \\ &= \sigma_o^2 \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \times \\ &\quad \times \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} = \\ &= \sigma_o^2 \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k^2(k) \mathbf{X}_{Nk}^H(k) \right] \left[\mathbf{X}_{Nk}(k) \boldsymbol{\Lambda}_k(k) \mathbf{X}_{Nk}^H(k) \right]^{-1}. \end{aligned} \quad (6.119)$$

Используя (6.97), уравнение (6.119) преобразуется к виду

$$\begin{aligned}
 \hat{\mathbf{K}}_N(k) &= \sigma_o^2 \left[\frac{1-\lambda^k}{1-\lambda} \mathbf{R}_N \right]^{-1} \left[\frac{1-\lambda^{2k}}{1-\lambda^2} \mathbf{R}_N \right] \left[\frac{1-\lambda^k}{1-\lambda} \mathbf{R}_N \right]^{-1} = \\
 &= \sigma_o^2 \left(\frac{1-\lambda}{1-\lambda^k} \right) \left(\frac{1-\lambda^{2k}}{1-\lambda^2} \right) \left(\frac{1-\lambda}{1-\lambda^k} \right) \mathbf{R}_N^{-1} \mathbf{R}_N \mathbf{R}_N^{-1} = \\
 &= \sigma_o^2 \left(\frac{1-\lambda}{1-\lambda^k} \right)^2 \left(\frac{1-\lambda^k}{1-\lambda} \right) \left(\frac{1+\lambda^k}{1+\lambda} \right) \mathbf{R}_N^{-1} = \sigma_o^2 \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) \mathbf{R}_N^{-1}.
 \end{aligned} \tag{6.120}$$

Из уравнения (6.120) следует, что при $k \rightarrow \infty$

$$\hat{\mathbf{K}}_N(\infty) = \sigma_o^2 \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^\infty}{1-\lambda^\infty} \right) \mathbf{R}_N^{-1} = \sigma_o^2 \frac{1-\lambda}{1+\lambda} \mathbf{R}_N^{-1}. \tag{6.121}$$

Далее, используя (6.81) и (6.107), априорную ошибку моделирования требуемого сигнала можно выразить как

$$\begin{aligned}
 \alpha_N(k) &= d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \mathbf{h}_{N,o}^H \mathbf{x}_N(k) + e_o(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) = \\
 &= e_o(k) + [\mathbf{h}_{N,o}^H - \mathbf{h}_N^H(k-1)] \mathbf{x}_N(k) = e_o(k) - \Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k),
 \end{aligned} \tag{6.122}$$

см. также (5.29).

Используя уравнения (6.122) и (3.70), получим выражение для обучающей кривой RLS-алгоритма

$$\begin{aligned}
 \xi(k) &= E\{\alpha_N(k) \alpha_N^*(k)\} = E\{e_o(k) e_o^*(k) - e_o(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1) - \\
 &\quad - \Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) e_o^*(k) + \Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1)\} = \\
 &= E\{e_o(k) e_o^*(k)\} - E\{e_o(k)\} \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1) - \\
 &\quad - \Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) E\{e_o^*(k)\} + E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \Delta \mathbf{h}_N(k-1)\} = \\
 &= E\{e_o(k) e_o^*(k)\} + E\{\Delta \mathbf{h}_N^H(k-1) E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\} \Delta \mathbf{h}_N(k-1)\} = \\
 &= \sigma_o^2 + E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1)\},
 \end{aligned} \tag{6.123}$$

где $\sigma_o^2 = E\{|e_o(k)|^2\} = \xi_{\min}$, см. (5.31), и $\mathbf{R}_N = E\{\mathbf{x}_N(k) \mathbf{x}_N^H(k)\}$, см. (3.5).

В уравнении (6.123) использовано соотношение (5.50), а также учитывается, что $E\{e_o(k)\} = 0$ и $E\{e_o^*(k)\} = 0$.

Второе слагаемое в (6.123) представляет собой скалярную величину. Так как след скалярной величины равен самой величине, то, используя выражения (3.52) $\text{tr}\{\mathbf{A}_N \mathbf{B}_N\} = \text{tr}\{\mathbf{B}_N \mathbf{A}_N\}$ и (6.119) $\hat{\mathbf{K}}_N(k) = E\{\Delta \mathbf{h}_N(k) \Delta \mathbf{h}_N^H(k)\}$, второе слагаемое в уравнении (6.123) можно преобразовать как

$$\begin{aligned}
 E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1)\} &= \\
 &= \text{tr}\{E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1)\}\} = E\{\text{tr}\{\Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1)\}\} = \\
 &= E\{\text{tr}\{\Delta \mathbf{h}_N^H(k-1) \Delta \mathbf{h}_N(k-1) \mathbf{R}_N\}\} = \text{tr}\{E\{\Delta \mathbf{h}_N^H(k-1) \Delta \mathbf{h}_N(k-1) \mathbf{R}_N\}\} = \\
 &= \text{tr}\{E\{\Delta \mathbf{h}_N^H(k-1) \Delta \mathbf{h}_N(k-1)\} \mathbf{R}_N\} = \text{tr}\{\hat{\mathbf{K}}_N(k-1) \mathbf{R}_N\}.
 \end{aligned} \tag{6.124}$$

С учетом (6.124) уравнение (6.123) можно записать в виде

$$\xi(k) = \sigma_o^2 + \text{tr} \left\{ \hat{\mathbf{K}}_N(k-1) \mathbf{R}_N \right\}. \quad (6.125)$$

Анализ уравнения (6.125) представляет интерес при $\lambda=1$ и при $\lambda < 1$.

С учетом (6.20), уравнение (6.119) при $\lambda=1$ записывается как

$$\begin{aligned} \hat{\mathbf{K}}_N(k) &= \sigma_o^2 \left[\mathbf{X}_{Nk}(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \left[\mathbf{X}_{Nk}(k) \mathbf{X}_{Nk}^H(k) \right] \left[\mathbf{X}_{Nk}(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} = \\ &= \sigma_o^2 \left[\mathbf{X}_{Nk}(k) \mathbf{X}_{Nk}^H(k) \right]^{-1} \approx \sigma_o^2 \frac{\mathbf{R}_N^{-1}}{k}. \end{aligned} \quad (6.126)$$

В этом случае при больших k , т. е. в установившемся режиме,

$$\hat{\mathbf{K}}_N(k-1) \approx \hat{\mathbf{K}}_N(k), \quad (6.127)$$

а

$$\begin{aligned} \xi(k) &= \sigma_o^2 + \text{tr} \left\{ \hat{\mathbf{K}}_N(k-1) \mathbf{R}_N \right\} = \sigma_o^2 + \text{tr} \left\{ \sigma_o^2 \frac{\mathbf{R}_N^{-1}}{k} \mathbf{R}_N \right\} = \\ &= \sigma_o^2 + \frac{\sigma_o^2}{k} \text{tr} \{ \mathbf{R}_N^{-1} \mathbf{R}_N \} = \sigma_o^2 + \frac{\sigma_o^2}{k} \text{tr} \{ \mathbf{I}_N \} = \sigma_o^2 + \frac{\sigma_o^2}{k} N = \xi_{\min} + \frac{\xi_{\min}}{k} N. \end{aligned} \quad (6.128)$$

Таким образом, из уравнения (6.128) следует, что при $k \rightarrow \infty$ значение

$$\xi(\infty) = \xi_{\min}. \quad (6.129)$$

Это означает, что

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} = \frac{\xi(\infty) - \xi_{\min}}{\xi_{\min}} = \frac{\xi_{\min} - \xi_{\min}}{\xi_{\min}} = 0, \quad (6.130)$$

т. е. при $\lambda=1$ относительное значение дополнительной среднеквадратической ошибки $\xi_{\text{exc}} = \xi(\infty) - \xi_{\min}$ RLS-алгоритма в установившемся режиме равно нулю.

При $\lambda < 1$ и с учетом (6.129) в уравнении (6.125) необходимо использовать формулу (6.120). Тогда

$$\begin{aligned} \xi(k) &= \sigma_o^2 + \text{tr} \left\{ \hat{\mathbf{K}}_N(k-1) \mathbf{R}_N \right\} = \sigma_o^2 + \text{tr} \left\{ \sigma_o^2 \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) \mathbf{R}_N^{-1} \mathbf{R}_N \right\} = \\ &= \sigma_o^2 + \text{tr} \left\{ \sigma_o^2 \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) \mathbf{I}_N \right\} = \sigma_o^2 + \sigma_o^2 \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) N = \\ &= \sigma_o^2 \left[1 + \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) N \right] = \xi_{\min} \left[1 + \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) N \right], \end{aligned} \quad (6.131)$$

где $\text{tr} \{ \mathbf{I}_N \} = N$. При $k \rightarrow \infty$

$$\xi(\infty) = \xi_{\min} \left[1 + N \frac{1-\lambda}{1+\lambda} \right]. \quad (6.132)$$

Это означает, что

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} = \frac{\xi(\infty) - \xi_{\min}}{\xi_{\min}} = \frac{\xi_{\min} \left[1 + N \frac{1-\lambda}{1+\lambda} \right] - \xi_{\min}}{\xi_{\min}} = \frac{\xi_{\min} \left[1 + N \frac{1-\lambda}{1+\lambda} \right] - 1}{\xi_{\min}} = N \frac{1-\lambda}{1+\lambda}. \quad (6.133)$$

Таким образом, в RLS-алгоритме значение $\xi_{\text{exc}} = \xi(\infty) - \xi_{\text{min}}$ в установившемся режиме и значение M не зависят от разброса собственных чисел корреляционной матрицы входных сигналов адаптивного фильтра. Это отличает RLS-алгоритм от LMS-алгоритма, в котором такая зависимость наблюдается, см. выражения (5.120) и (5.121), а также рис. 5.6—5.10, рис. 6.8 и рис. 6.9.

Из выражения (6.131) следует, что при $k \neq \infty$ относительное значение дополнительной среднеквадратической ошибки определяется как

$$\xi_{\text{exc}} = \xi(\infty) - \xi_{\text{min}} = \xi_{\text{min}} \left[1 + N \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) \right] - \xi_{\text{min}} = \xi_{\text{min}} N \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right). \quad (6.134)$$

Используя уравнение (6.134) при $N=2$, $\lambda=0.9999$ и $k=400$, значение ξ_{exc} можно оценить как

$$\xi_{\text{exc}} = 1 \cdot 2 \left(\frac{1-0.9999}{1+0.9999} \right) \left(\frac{1+0.9999^{400}}{1-0.9999^{400}} \right) \approx 0.005,$$

что хорошо совпадает с результатом на рис. 6.9, где $\xi_{\text{exc}} \approx 0.005$, так как $\xi_{\text{min}} = 1$.

В разделе 6.5 параметр экспоненциального взвешивания λ был введен с целью обеспечения возможности сложения за «медленно меняющейся статистикой» обрабатываемых адаптивным фильтром сигналов. Эффективность использования параметра λ уменьшается по мере роста N , так как при $\lambda \rightarrow 1$ увеличивается «память алгоритма». Поэтому, если $\lambda \rightarrow 1$, то согласно уравнению (6.130) $M \rightarrow 0$, что хорошо с точки зрения «нулевой» дополнительной среднеквадратической ошибки ξ_{exc} в установившемся режиме. Однако при этом не обеспечивается возможность сложения адаптивного фильтра за изменяющимися условиями функционирования.

Значения $\lambda < 1$ обеспечивают возможность такого сложения, однако цена этой возможности — ненулевая дополнительная среднеквадратическая ошибка, см. уравнение (6.133)

$$\xi_{\text{exc}} = \xi(\infty) - \xi_{\text{min}} = \xi_{\text{min}} \left[1 + N \frac{1-\lambda}{1+\lambda} \right] - \xi_{\text{min}} = \xi_{\text{min}} N \frac{1-\lambda}{1+\lambda} \approx \xi_{\text{min}} N \frac{1-\lambda}{2}, \quad (6.135)$$

пропорциональная $(1-\lambda)$, так как $\lambda \approx 1$ и $1+\lambda \approx 2$.

Поэтому на практике величину λ выбирают из компромисса, исходя из вышеизложенных свойств RLS-алгоритма, а также его устойчивости, которая гарантируется, если параметр λ удовлетворяет соотношению $(1-0.4/N) \leq \lambda \leq 1$ [95, 96]. При значениях λ , меньших указанной нижней границы, RLS-алгоритм и его разновидности, как правило, являются неустойчивыми.

Уравнение (6.131) получено в предположении больших k , т. е. оно описывает движение обучающей кривой после начального переходного процесса к установившемуся значению (6.132). Это движение согласно развернутому выражению уравнения (6.131)

$$\begin{aligned}
 \xi(k) &= \xi_{\min} + \xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^k}{1-\lambda^k} \right) N \approx \\
 &\approx \xi_{\min} + \xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) (1+\lambda^k) N = \\
 &= \xi_{\min} + \xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) N + \xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) \lambda^k N
 \end{aligned} \tag{6.136}$$

определяется экспоненциально затухающим слагаемым

$$\xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) \lambda^k N = C \lambda^k = C e^{-\frac{k}{k_{\tau_{\xi}}}}, \tag{6.137}$$

зависящим от k , где $C = \xi_{\min} \left(\frac{1-\lambda}{1+\lambda} \right) N$.

Следуя уравнениям (4.10)–(4.13), постоянную времени $k_{\tau_{\xi}}$ переходного процесса обучающей кривой при больших k можно определить как число отсчетов относительно текущего отсчета k , при котором значение $C \lambda^{k+k_{\tau_{\xi}}}$ уменьшится в $e \approx 2.7183$ раза по сравнению со значением $C \lambda^k$, т. е.

$$\frac{C e^{-\frac{k}{k_{\tau_{\xi}}}}}{C e^{-\frac{k+k_{\tau_{\xi}}}{k_{\tau_{\xi}}}}} = \frac{C \lambda^k}{C \lambda^{k+k_{\tau_{\xi}}}} = \lambda^{-k_{\tau_{\xi}}} = e. \tag{6.138}$$

Из уравнения (6.138) постоянная времени $k_{\tau_{\xi}}$ определяется как

$$k_{\tau_{\xi}} = -\log_{\lambda} \lambda^{-k_{\tau_{\xi}}} = -\log_{\lambda} e = -\frac{\ln e}{\ln \lambda} = -\frac{1}{\ln \lambda} = -\frac{1}{\ln[1-(1-\lambda)]} \approx -\frac{1}{-(1-\lambda)} = \frac{1}{1-\lambda} \tag{6.139}$$

при $1-\lambda \ll 1$. Если $\lambda=0.9$, то $k_{\tau_{\xi}} = -1/\ln 0.9 = 9.5 \approx 10$ отсчетов. При $\lambda=0.99$ $k_{\tau_{\xi}} \approx 100$ отсчетов; при $\lambda=0.999$ $k_{\tau_{\xi}} \approx 1000$ отсчетов и т. д.

Как видно из (6.139), постоянная времени обучающей кривой RLS-алгоритма не зависит от собственных чисел корреляционной матрицы входных сигналов адаптивного фильтра, что подтверждено рис. 6.8 и рис. 6.9, на которых обучающие кривые, полученные для двух разных отношений собственных чисел χ практически совпадают, в то время как обучающие кривые LMS-алгоритма имеют заметные отличия при сравнении друг с другом.

Анализ уравнения (6.101) показывает, что инициализация корреляционной матрицы приводит к смещению вектора ВК. Это смещение $\Delta \mathbf{h}_N(k) \rightarrow 0$ при $k \rightarrow \infty$. Результаты анализа обучающей кривой (6.131)–(6.136) также справедливы при больших k , т. е. в установленном режиме.

Справедливость этих результатов продемонстрирована на рис. 6.8, рис. 6.9 и подтверждена расчетом, следующим за уравнением (6.134).

Различное поведение обучающей кривой RLS-алгоритма при разных значениях k («малых» и «больших») объясняется следующим образом.

На начальном участке, когда $k < N$ и среднеквадратичная ошибка остается большой. Это обусловлено существованием бесконечного числа допустимых решений системы уравнений

$$\mathbf{h}_N^H(k)\mathbf{x}_N(i)=d(i), \quad (6.140)$$

где $i=1, 2, \dots, k$.

Для подтверждения сказанного рассмотрим пример при $k=2$ и $N=4$. В этом случае матрица $\mathbf{X}_{Nk}(k)$ определяется как

$$\mathbf{X}_{Nk}(k)=\begin{bmatrix} x(1) & x(2) \\ 0 & x(1) \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

а матрица $\mathbf{R}_N(k)$ согласно уравнению (6.115) — как

$$\begin{aligned} \mathbf{R}_N(k) &= \mathbf{X}_{Nk}(k)\Lambda_k(k)\mathbf{X}_{Nk}^H(k) = \begin{bmatrix} x(1) & x(2) \\ 0 & x(1) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^*(1) & 0 & 0 & 0 \\ x^*(2) & x^*(1) & 0 & 0 \end{bmatrix} = \\ &= \begin{bmatrix} \lambda x(1)x^*(1) + x(2)x^*(2) & x(2)x^*(1) & 0 & 0 \\ x(1)x^*(2) & x(1)x^*(1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{11}(k) & R_{12}(k) & 0 & 0 \\ R_{21}(k) & R_{22}(k) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Вектор $\mathbf{r}_N(k)$ согласно уравнению (6.114) записывается как

$$\begin{aligned} \mathbf{r}_N(k) &= \mathbf{X}_{Nk}(k)\Lambda_k(k)\mathbf{d}_k^* = \\ &= \begin{bmatrix} x(1) & x(2) \\ 0 & x(1) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d^*(1), & d^*(2), & 0, & 0 \end{bmatrix} = \begin{bmatrix} \lambda x(1)d^*(1) + x(2)d^*(2) \\ x(1)d^*(2) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_1(k) \\ r_2(k) \\ 0 \\ 0 \end{bmatrix}. \end{aligned}$$

Определитель матрицы $\mathbf{R}_N(k)$ в этом случае равен нулю, так как эта матрица содержит нулевые диагональные элементы. Это означает, что матрица $\mathbf{R}_N^{-1}(k)$ не может быть получена, см. уравнение (3.13), а значит, нельзя однозначно найти и вектор (6.113), т. е. получить среднеквадратическое решение системы линейных уравнений (6.140).

Эта проблема решается путем инициализации корреляционной матрицы (6.89) $\mathbf{R}_N(0)=\delta^2\mathbf{I}_N$. Это позволяет, используя в (6.113) выражения (6.92) и (6.114), получить среднеквадратическое решение системы уравнений (6.140) как

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) = \left\{ \left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) \right] + \lambda^k \delta^2 \mathbf{I}_N \right\}^{-1} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i). \quad (6.141)$$

И хотя вектор (6.141) достаточно хорошо удовлетворяет системе уравнений (6.140), он может быть не вполне хорошей оценкой оптимального винеровского решения $\mathbf{h}_{N,o}$.

В результате, апостериорная ошибка оценивания процесса $\{d(i)\}$ (6.82) $e_N(k) = d(k) - \mathbf{h}_N^H(k) \mathbf{x}_N(k)$ может быть очень маленькой, а априорная ошибка (6.81) $\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$ — очень большой. По этой причине при определении обучающей кривой (6.123) $\xi(k) = E\{\alpha_N(k) \alpha_N^*(k)\}$ используются априорные, а не апостериорные ошибки.

При $k > N$ число уравнений k в системе (6.140) больше числа неизвестных ВК N . Однако в этом случае вектор ВК $\mathbf{h}_N(k)$ может быть однозначно найден с помощью уравнения (6.116). Точность оценки вектора $\mathbf{h}_{N,o}$ с помощью вектора $\mathbf{h}_N(k)$ определяется числом отсчетов k обрабатываемых сигналов $x(k)$ и $d(k)$ и уровнем шума $e_o(k)$. При $e_o(k) = 0$ согласно выражению (6.117) $\mathbf{h}_N(k) = \mathbf{h}_{N,o}$ и $\xi(k) = \sigma_o^2 + E\{\Delta \mathbf{h}_N^H(k-1) \mathbf{R}_N \Delta \mathbf{h}_N(k-1)\} = 0$ для всех $k \geq N$, так как $\sigma_o^2 = E\{|e_o(k)|^2\} = 0$ и $\Delta \mathbf{h}_N(k) = \mathbf{0}_N$.

Таким образом, RLS-алгоритм находит хорошую оценку $\mathbf{h}_N(k)$ вектора $\mathbf{h}_{N,o}$ как только имеется достаточное число отсчетов обрабатываемых сигналов $x(i)$ и $d(i)$, где $i = 1, 2, \dots, k$. Этим объясняется резкое уменьшение значений обучающей кривой при $k > N$.

После этого RLS-алгоритм сходится к установившемуся состоянию со значительно меньшей скоростью, которая определяется постоянной времени (6.139) $k_{\xi} = 1/(1-\lambda)$, зависящей от параметра экспоненциального взвешивания λ .

Результаты анализа уравнения (6.26), представленные в разделах 6.3 и 6.4, также применимы к уравнению (6.113), если матрицу $\mathbf{X}_{Nk}(k)$ и векторы $\mathbf{d}_k, \mathbf{y}_k, \mathbf{e}_k$ в уравнениях этих разделов определить соответственно как

$$\begin{aligned} & \left[\lambda^{0.5(k-1)} \mathbf{x}_N(1), \lambda^{0.5(k-2)} \mathbf{x}_N(2), \dots, \lambda^{0.5(0)} \mathbf{x}_N(k) \right] = \\ & = \begin{bmatrix} \lambda^{0.5(k-1)} x(1) & \lambda^{0.5(k-2)} x(2) & \dots & \lambda^{0.5(0)} x(k) \\ 0 & \lambda^{0.5(k-2)} x(1) & \dots & \lambda^{0.5(0)} x(k-1) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda^{0.5(0)} x(k-N+1) \end{bmatrix} \end{aligned}$$

и

$$\left[\lambda^{0.5(k-1)} d(1), \lambda^{0.5(k-2)} d(2), \dots, \lambda^{0.5(0)} d(k) \right]^T,$$

$$\left[\lambda^{0.5(k-1)} y(1), \lambda^{0.5(k-2)} y(2), \dots, \lambda^{0.5(0)} y(k) \right]^T,$$

$$\left[\lambda^{0.5(k-1)} e(1), \lambda^{0.5(k-2)} e(2), \dots, \lambda^{0.5(0)} e(k) \right]^T.$$



6.8. Выводы по главе

В главе 6 сформулирована постановка задачи LS-приближения двух процессов, между которыми существует линейная зависимость, и получено решение этой задачи. Приведен рекурсивный аналог этого решения, приводящий к RLS-алгоритму на основе MIL, широко используемому в теории и практике адаптивной фильтрации. Кроме того, проведен анализ качества адаптивной фильтрации сигналов с помощью RLS-алгоритма. Дополнительные детали такого анализа приведены в [98]. В следующих главах будут описаны математически эквивалентные аналоги рассмотренного RLS-алгоритма, отличающиеся вычислительными процедурами, а значит, и числом арифметических операций, требуемых для реализации одной итерации этих алгоритмов.

ГЛАВА 7

РЕКУРСИВНАЯ АДАПТИВНАЯ ФИЛЬТРАЦИЯ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ НА ОСНОВЕ ОБРАТНОГО QR-РАЗЛОЖЕНИЯ

В настоящей главе рассматривается RLS-алгоритм адаптивной фильтрации сигналов на основе обратного QR-разложения с использованием вращений Гивенса. Представлены основные шаги получения алгоритма, его вычислительная процедура, а также вопросы аппаратной реализации, ориентированной на параллельные вычисления.

7.1. Введение

В главе 6 было рассмотрено рекурсивное решение LS-задачи адаптивной фильтрации сигналов на основе использования MIL для обращения корреляционной матрицы входных сигналов адаптивного фильтра, приводящее к RLS-алгоритму. Эта же задача также может решаться с помощью приведения матрицы отсчетов входных сигналов адаптивного фильтра к треугольному виду (триангуляризации). Одним из методов триангуляризации является QR-разложение (QR Decomposition, QRD) [99], основанное на вращениях Гивенса. Однако при использовании QR-разложения для нахождения вектора ВК \mathbf{h}_N в задаче RLS-фильтрации после триангуляризации необходимо выполнять обратные подстановки, подобно тому, как это делается при решении системы линейных уравнений с помощью метода Гаусса.

Поскольку при решении RLS-задачи с помощью QR-разложения используется не корреляционная матрица (6.115)

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \mathbf{X}_{Nk}(k) \Lambda_k(k) \mathbf{X}_{Nk}^H(k), \quad (7.1)$$

а матрица преобразованных с помощью этого разложения отсчетов входных сигналов адаптивного фильтра \mathbf{X}_{Nk} , то при этом уменьшается динамический диапазон чисел, участвующих в вычислениях, по сравнению с числами, фигу-

рирующими при решении этой задачи с помощью MIL, что напрямую связано с вопросами устойчивости QRD RLS-алгоритмов. Это полезное свойство получило подтверждение в работах [20, 100]. Другое полезное свойство QRD RLS-алгоритмов состоит в том, что большинство вычислений в них могут быть выполнены параллельно. Как следствие, QRD RLS-алгоритмы нашли широкое применение в теории и практике адаптивной фильтрации [101–107].

Среди методов, на которых базируются RLS-алгоритмы адаптивной фильтрации, важное место занимает обратное QR-разложение (Inverse QRD, IQRD) [102, 103]. При его использовании в QRD RLS-алгоритмах вектор ВК \mathbf{h}_N вычисляется без обратных подстановок, т. е. более эффективно в терминах числа арифметических операций, так как при использовании прямого QR-разложения обратные подстановки для вычисления вектора \mathbf{h}_N необходимо выполнять на каждой итерации k .

7.2. RLS-фильтрация с помощью прямого QR-разложения

Задача RLS-фильтрации, сформулированная в главе 6 как задача минимизации целевой функции (6.60)

$$E_N(k) = \sum_{i=1}^k \lambda^{k-i} e_N(i) e_N^*(i) = \sum_{i=1}^k \lambda^{k-i} [d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i)] [d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i)]^*, \quad (7.2)$$

может быть выражена в терминах векторов и матриц обрабатываемых сигналов $\{x(i)\}$, $\{d(i)\}$ и $\{e(i)\}$ как

$$E_N(k) = \mathbf{e}_k^T(k) \mathbf{\Lambda}_k(k) \mathbf{e}_k^*(k) \rightarrow \min, \quad (7.3)$$

где

$$\mathbf{e}_k(k) = [e(1), e(2), \dots, e(k)]^T, \quad (7.4)$$

$$\mathbf{d}_k(k) = [d(1), d(2), \dots, d(k)]^T, \quad (7.5)$$

$$\mathbf{e}_k^T(k) = \mathbf{d}_k^T(k) - \mathbf{h}_N^H(k) \mathbf{X}_{Nk}(k), \quad (7.6)$$

$$\mathbf{e}_k^*(k) = \mathbf{d}_k^*(k) - \mathbf{X}_{Nk}^H(k) \mathbf{h}_N(k), \quad (7.7)$$

$$\mathbf{X}_{Nk}(k) = \begin{bmatrix} x(1) & x(2) & \dots & x(N) & \dots & x(k-1) & x(k) \\ 0 & x(1) & \dots & x(N-1) & \dots & x(k-2) & x(k-1) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & x(1) & \dots & x(k-N) & x(k-N+1) \end{bmatrix}. \quad (7.8)$$

Определим в соответствии с уравнением (6.112) диагональную матрицу

$$\begin{aligned}\sqrt{\Lambda_k(k)} &= \text{diag}\left\{\sqrt{\lambda^{k-1}}, \sqrt{\lambda^{k-2}}, \dots, \sqrt{\lambda^1}, \sqrt{\lambda^0}\right\}^T = \\ &= \text{diag}\left\{\sqrt{\lambda^{k-1}}, \sqrt{\lambda^{k-2}}, \dots, \sqrt{\lambda}, 1\right\}^T.\end{aligned}\quad (7.9)$$

Далее для компактности записей будем использовать обозначения $\sqrt{\lambda} = \lambda^{0.5}$, $1/\sqrt{\lambda} = \lambda^{-0.5}$ и $\sqrt{\Lambda_k(k)} = \Lambda_k^{0.5}(k)$. Тогда, с учетом выражений (7.3)–(7.9) уравнение (7.2) можно представить как

$$\begin{aligned}E_N(k) &= [\mathbf{e}_k^T(k) \Lambda_k^{0.5}(k)] [\Lambda_k^{0.5}(k) \mathbf{e}_k^*(k)] = \\ &= [\mathbf{d}_k^T(k) \Lambda_k^{0.5}(k) - \mathbf{h}_N^H(k) \mathbf{X}_{Nk}(k) \Lambda_k^{0.5}(k)] [\Lambda_k^{0.5}(k) \mathbf{d}_k^*(k) - \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) \mathbf{h}_N(k)].\end{aligned}\quad (7.10)$$

Из теории матриц известно, что существует такая унитарная (в случае действительных чисел ортонормированная) матрица $\mathbf{Q}_k(k)$, что для любой матрицы \mathbf{A}_{kN} может быть получено разложение

$$\mathbf{Q}_k^H \mathbf{A}_{kN} = \begin{bmatrix} \Phi_N \\ \mathbf{O}_{(k-N)N} \end{bmatrix}, \quad (7.11)$$

где Φ_N — верхняя треугольная квадратная матрица, называемая разложением Холецкого, $\mathbf{O}_{(k-N)N}$ — нулевая прямоугольная матрица. Для унитарных матриц $\mathbf{Q}_k(k)$ справедливы соотношения $\mathbf{Q}_k(k) \mathbf{Q}_k^H(k) = \mathbf{Q}_k^H(k) \mathbf{Q}_k(k) = \mathbf{I}_k$ и $\mathbf{Q}_k^H(k) = \mathbf{Q}_k^{-1}(k)$, см. раздел 3.3.

Если обозначить

$$\mathbf{A}_{kN} = \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k), \quad (7.12)$$

то согласно уравнению (7.11)

$$\mathbf{Q}_k^H(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix}, \quad (7.13)$$

а

$$\mathbf{X}_{Nk}(k) \Lambda_k^{0.5}(k) \mathbf{Q}_k(k) = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix}^H = \begin{bmatrix} \tilde{\mathbf{R}}_N^H(k), \mathbf{O}_{(k-N)N}^T \end{bmatrix}. \quad (7.14)$$

Использование символов \mathbf{Q} и \mathbf{R} для обозначение матриц в преобразованиях, аналогичных (7.13) и (7.14), дало название этим преобразованиям как QR-разложение.

QR-разложение можно применить и к вектору $\mathbf{d}_k(k)$ как

$$\mathbf{Q}_k^H(k) \Lambda_k^{0.5}(k) \mathbf{d}_k^*(k) = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix} \quad (7.15)$$

и

$$\mathbf{d}_k(k) \Lambda_k^{0.5}(k) \mathbf{Q}_k(k) = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix}^H = \begin{bmatrix} \tilde{\mathbf{r}}_N^H(k), \tilde{\mathbf{v}}_{k-N}^H(k) \end{bmatrix}. \quad (7.16)$$



Используя выражения (7.13)–(7.16), уравнение (7.10) можно представить как

$$\begin{aligned}
 E_N(k) &= [\mathbf{e}_k^T(k) \Lambda_k^{0.5}(k)] \mathbf{Q}_k(k) \mathbf{Q}_k^H(k) [\Lambda_k^{0.5}(k) \mathbf{e}_k^*(k)] = [\tilde{\mathbf{r}}_N^H(k), \tilde{\mathbf{v}}_{k-N}^H(k)] \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix} - \\
 &- \mathbf{h}_N^H(k) [\tilde{\mathbf{R}}_N^H(k), \mathbf{O}_{(k-N)N}^T] \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix} - [\tilde{\mathbf{r}}_N^H(k), \tilde{\mathbf{v}}_{k-N}^H(k)] \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix} \mathbf{h}_N(k) + \\
 &+ \mathbf{h}_N^H(k) [\tilde{\mathbf{R}}_N^H(k), \mathbf{O}_{(k-N)N}^T] \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix} \mathbf{h}_N(k) = \left\| \begin{bmatrix} \tilde{\mathbf{r}}_N(k) - \tilde{\mathbf{R}}_N(k) \mathbf{h}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix} \right\|_2^2.
 \end{aligned} \quad (7.17)$$

Уравнения (7.10) и (7.17) являются математически эквивалентными, так как $\mathbf{Q}_k(k) \mathbf{Q}_k^H(k) = \mathbf{I}_k$. Условие минимума функции $E_N(k)$ определяется из уравнения (7.17) как

$$\begin{aligned}
 \nabla_{\mathbf{h}_N^*} E_N(k) &= \frac{\partial E_N(k)}{\partial \mathbf{h}_N^*(k)} = -\tilde{\mathbf{R}}_N^H(k) \tilde{\mathbf{r}}_N(k) + \tilde{\mathbf{R}}_N^H(k) \tilde{\mathbf{R}}_N(k) \mathbf{h}_N(k) = \\
 &= \tilde{\mathbf{R}}_N^H(k) [-\tilde{\mathbf{r}}_N(k) + \tilde{\mathbf{R}}_N(k) \mathbf{h}_N(k)] = -\tilde{\mathbf{r}}_N(k) + \tilde{\mathbf{R}}_N(k) \mathbf{h}_N(k) = \mathbf{0}_N,
 \end{aligned} \quad (7.18)$$

так как $\tilde{\mathbf{R}}_N^H(k) \neq \mathbf{O}_N$.

Аналогичное уравнение для действительных сигналов и ВК $\tilde{\mathbf{h}}_N(k)$ определяется как

$$\nabla_{\tilde{\mathbf{h}}_N} E_N(k) = \frac{\partial E_N(k)}{\partial \tilde{\mathbf{h}}_N(k)} = -2\tilde{\mathbf{r}}_N(k) + 2\tilde{\mathbf{R}}_N(k) \tilde{\mathbf{h}}_N(k) = \mathbf{0}_N. \quad (7.19)$$

Так как в уравнениях (7.18) и (7.19) матрица $\tilde{\mathbf{R}}_N(k)$ является верхней треугольной, то из этих уравнений можно найти векторы ВК, не обращая эту матрицу, а используя обратные подстановки.

Уравнения (7.18) и (7.19) используются для решения RLS-задачи на базе прямого QR-разложения. Для рекурсивного вычисления фигурирующих в этих уравнениях матриц $\tilde{\mathbf{R}}_N(k)$ и векторов $\tilde{\mathbf{r}}_N(k)$ необходимо воспользоваться соотношениями [108], устанавливающими связь между $\tilde{\mathbf{R}}_N(k-1)$ и $\tilde{\mathbf{R}}_N(k)$, а также между $\tilde{\mathbf{r}}_N(k-1)$ и $\tilde{\mathbf{r}}_N(k)$ как

$$\mathbf{T}_{N+1}(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{0}_N^T \end{bmatrix} \quad (7.20)$$

и

$$\mathbf{T}_{N+1}(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \\ d^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\alpha}_N^*(k) \end{bmatrix}. \quad (7.21)$$

Матрица $\mathbf{T}_{N+1}(k)$ в уравнениях (7.20) и (7.21) является унитарной. Эту матрицу можно сформировать с помощью вращений Гивенса, что, однако, не исключает необходимости использования обратных подстановок для получения вектора ВК $\mathbf{h}_N(k)$.

Рассматриваемое далее обратное QR-разложение также позволяет найти вектор $\mathbf{h}_N(k)$ рекурсивным способом, но без обратных подстановок. Поэтому

в настоящей книге сначала рассматривается RLS-алгоритм на базе обратного QR-разложения как более вычислительно-эффективный, а затем — RLS-алгоритм на базе прямого QR-разложения.

7.3. Рекурсивное вычисление вектора весовых коэффициентов

Из уравнения (7.18) следует, что

$$\mathbf{h}_N(k) = \tilde{\mathbf{R}}_N^{-1}(k) \tilde{\mathbf{r}}_N(k). \quad (7.22)$$

Для получения уравнения (7.22) в рекурсивной форме $\mathbf{T}_{N+1}(k)$ необходимо представить в виде клеточной матрицы как

$$\mathbf{T}_{N+1}(k) = \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix}, \quad (7.23)$$

где $\delta_N(k)$ — действительная переменная, и применить матрицу (7.23) в уравнениях (7.20) и (7.21) как

$$\begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{0}_N^T \end{bmatrix} \quad (7.24)$$

и

$$\begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \\ d^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \xi_N^*(k) \end{bmatrix}. \quad (7.25)$$

Из уравнений (7.24) и (7.25) следует, что

$$\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{R}}_N(k-1) + \mathbf{b}_N(k) \mathbf{x}_N^H(k) = \tilde{\mathbf{R}}_N(k) \quad (7.26)$$

и

$$\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{r}}_N(k-1) + \mathbf{b}_N(k) d^*(k) = \tilde{\mathbf{r}}_N(k). \quad (7.27)$$

Если выражения (7.26) и (7.27) использовать в уравнении (7.22), то получим, что

$$\mathbf{h}_N(k) = \left[\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{R}}_N(k-1) + \mathbf{b}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \left[\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{r}}_N(k-1) + \mathbf{b}_N(k) d^*(k) \right]. \quad (7.28)$$

Так как на k -й итерации переменные $\tilde{\mathbf{R}}_N(k-1)$, $\mathbf{x}_N(k)$, $\tilde{\mathbf{r}}_N(k-1)$ и $d^*(k)$ известны, то для вычислений ВК с помощью выражения (7.28) требуется определить переменные $\mathbf{A}_N(k)$ и $\mathbf{b}_N(k)$. Эти переменные можно получить следующим образом. Поскольку матрица $\mathbf{T}_{N+1}(k)$ является унитарной, то из условия

$$\begin{aligned} \mathbf{T}_{N+1}(k) \mathbf{T}_{N+1}^H(k) &= \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix}^H = \\ &= \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{b}_N(k) \\ \mathbf{c}_N^H(k) & \delta_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{A}_N^H(k) & \mathbf{c}_N(k) \\ \mathbf{b}_N^H(k) & \delta_N(k) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N^T & 1 \end{bmatrix} = \mathbf{I}_{N+1} \end{aligned} \quad (7.29)$$

следует, что

$$\mathbf{A}_N(k)\mathbf{A}_N^H(k) + \mathbf{b}_N(k)\mathbf{b}_N^H(k) = \mathbf{I}_N, \quad (7.30)$$

$$\mathbf{A}_N(k)\mathbf{c}_N(k) + \mathbf{b}_N(k)\delta_N(k) = \mathbf{0}_N^T, \quad (7.31)$$

$$\mathbf{c}_N^H(k)\mathbf{c}_N(k) + \delta_N^2(k) = 1. \quad (7.32)$$

Выразив из уравнения (7.31) вектор

$$\mathbf{b}_N(k) = -\mathbf{A}_N(k)\mathbf{c}_N(k)\delta_N^{-1}(k) \quad (7.33)$$

и подставив его в уравнение (7.28), можно исключить матрицу $\mathbf{A}_N(k)$, т. е.

$$\begin{aligned} \mathbf{h}_N(k) &= \left[\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{R}}_N(k-1) - \mathbf{A}_N(k) \mathbf{c}_N(k) \delta_N^{-1}(k) \mathbf{x}_N^H(k) \right]^{-1} \times \\ &\times \left[\lambda^{0.5} \mathbf{A}_N(k) \tilde{\mathbf{r}}_N(k-1) - \mathbf{A}_N(k) \mathbf{c}_N(k) \delta_N^{-1}(k) d^*(k) \right] = \\ &= \left[\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) - \mathbf{c}_N(k) \delta_N^{-1}(k) \mathbf{x}_N^H(k) \right]^{-1} \mathbf{A}^{-1}(k) \times \\ &\times \mathbf{A}_N(k) \left[\lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) - \mathbf{c}_N(k) \delta_N^{-1}(k) d^*(k) \right] = \\ &= \left[\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) - \mathbf{c}_N(k) \delta_N^{-1}(k) \mathbf{x}_N^H(k) \right]^{-1} \times \\ &\times \left[\lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) - \mathbf{c}_N(k) \delta_N^{-1}(k) d^*(k) \right]. \end{aligned} \quad (7.34)$$

В уравнении (7.34) использовано соотношение (3.50) $[\mathbf{A}_N \mathbf{B}_N]^{-1} = \mathbf{B}_N^{-1} \mathbf{A}_N^{-1}$, справедливое для обратимых квадратных матриц.

Далее, из уравнения (7.24) следует, что

$$\lambda^{0.5} \mathbf{c}_N^H(k) \tilde{\mathbf{R}}_N(k-1) + \delta_N(k) \mathbf{x}_N^H(k) = \mathbf{0}_N^T, \quad (7.35)$$

откуда

$$\mathbf{c}_N^H(k) = -\lambda^{-0.5} \delta_N(k) \mathbf{x}_N^H(k) \tilde{\mathbf{R}}_N^{-1}(k-1) \quad (7.36)$$

или

$$\mathbf{c}_N(k) = -\lambda^{-0.5} \left[\delta_N(k) \mathbf{x}_N^H(k) \tilde{\mathbf{R}}_N^{-1}(k-1) \right]^H = -\lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \mathbf{x}_N(k) \delta_N(k). \quad (7.37)$$

В уравнении (7.37) использовано соотношение (3.9) $[\mathbf{A}_{NM} \mathbf{B}_{MN}]^H = \mathbf{B}_{MN}^H \mathbf{A}_{NM}^H$. Запись $\tilde{\mathbf{R}}_N^{-H}(k)$ обозначает эрмитово сопряжение обратной матрицы, т. е. матрицы $[\tilde{\mathbf{R}}_N^{-1}(k)]^H$.

Если ввести обозначение

$$\mathbf{a}_N(k) = \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \mathbf{x}_N(k), \quad (7.38)$$

то уравнение (7.37) примет вид

$$\mathbf{c}_N(k) = -\mathbf{a}_N(k) \delta_N(k). \quad (7.39)$$

Теперь, используя (7.39), уравнение (7.34) можно представить как

$$\begin{aligned} \mathbf{h}_N(k) &= \left[\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) + \mathbf{a}_N(k) \delta_N(k) \delta_N^{-1}(k) \mathbf{x}_N^H(k) \right]^{-1} \times \\ &\times \left[\lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \mathbf{a}_N(k) \delta_N(k) \delta_N^{-1}(k) d^*(k) \right] = \\ &= \left[\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) + \mathbf{a}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \left[\lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \mathbf{a}_N(k) d^*(k) \right]. \end{aligned} \quad (7.40)$$

В уравнении (7.40) для обращения матрицы $\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) + \mathbf{a}_N(k) \mathbf{x}_N^H(k)$ необходимо использовать MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где $\mathbf{C} = \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1)$, $\mathbf{C}^{-1} = \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1)$, $\mathbf{B}^H = -\mathbf{a}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k)$ и $\mathbf{A}^{-1} = 1$.

Тогда

$$\begin{aligned} &\left[\lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) + \mathbf{a}_N(k) \mathbf{x}_N^H(k) \right]^{-1} = \\ &= \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) - \frac{\lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \mathbf{a}_N(k) \mathbf{x}_N^H(k) \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1)}{1 + \mathbf{x}_N^H(k) \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \mathbf{a}_N(k)} = \\ &= \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) - \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \frac{\mathbf{a}_N(k) \mathbf{a}_N^H(k)}{1 + \mathbf{a}_N^H(k) \mathbf{a}_N(k)}, \end{aligned} \quad (7.41)$$

так как согласно уравнению (7.38) $\mathbf{a}_N^H(k) = \left[\lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \mathbf{x}_N(k) \right]^H = \lambda^{-0.5} \mathbf{x}_N^H(k) \tilde{\mathbf{R}}_N^{-1}(k-1)$.

Введя обозначения

$$b_N^2(k) = 1 + \mathbf{a}_N^H(k) \mathbf{a}_N(k) \quad (7.42)$$

и

$$\mathbf{u}_N(k) = \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \mathbf{a}_N(k) b_N^{-1}(k), \quad (7.43)$$

уравнение для вычисления вектора ВК $\mathbf{h}_N(k)$ (7.40) можно привести к виду

$$\begin{aligned} \mathbf{h}_N(k) &= \left[\lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) - \mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \times \\ &\times \left[\lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \mathbf{a}_N(k) d^*(k) \right] = \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) - \\ &- \left[\mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \mathbf{a}_N(k) d^*(k) - \\ &- \left[\mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \mathbf{a}_N(k) d^*(k) = \mathbf{h}_N(k-1) - \\ &- \left[\mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \mathbf{u}_N(k) b_N(k) d^*(k) - \\ &- \left[\mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \mathbf{a}_N(k) d^*(k) = \mathbf{h}_N(k-1) + \\ &+ \left[-\mathbf{u}_N(k) \mathbf{a}_N^H(k) \mathbf{a}_N(k) d^*(k) + \mathbf{u}_N(k) b_N^2(k) d^*(k) \right] b_N^{-1}(k) - \\ &- \left[\mathbf{u}_N(k) \mathbf{a}_N^H(k) b_N^{-1}(k) \right] \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) = \mathbf{h}_N(k-1) - \\ &+ \mathbf{u}_N(k) b_N^{-1}(k) d^*(k) \left[-\mathbf{a}_N^H(k) \mathbf{a}_N + 1 + \mathbf{a}_N^H(k) \mathbf{a}_N \right] - \\ &- \mathbf{u}_N(k) b_N^{-1}(k) \mathbf{a}_N^H(k) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) = \\ &= \mathbf{h}_N(k-1) + \mathbf{u}_N(k) b_N^{-1}(k) \left[d^*(k) - \mathbf{a}_N^H(k) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \right]. \end{aligned} \quad (7.44)$$

Если в (7.44) использовать уравнение (7.38), то

$$\begin{aligned}\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{u}_N(k)b_N^{-1}(k) \left[d^*(k) - \mathbf{a}_N^H(k)\lambda^{0.5}\tilde{\mathbf{r}}_N(k-1) \right] = \\ &= \mathbf{h}_N(k-1) + \mathbf{u}_N(k)b_N^{-1}(k) \left[d^*(k) - \lambda^{-0.5}\mathbf{x}_N^H(k)\tilde{\mathbf{R}}_N^{-1}(k-1)\lambda^{0.5}\tilde{\mathbf{r}}_N(k-1) \right] = \\ &= \mathbf{h}_N(k-1) + \mathbf{u}_N(k)b_N^{-1}(k) \left[d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k-1) \right] = \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k),\end{aligned}\quad (7.45)$$

где $\mathbf{g}_N(k) = \mathbf{u}_N(k)b_N^{-1}(k)$.

Таким образом, уравнение вычисления вектора ВК $\mathbf{h}_N(k)$ (7.45) является идентичным уравнению (6.80). Различие между ними заключается в способе вычисления вектора коэффициентов Калмана $\mathbf{g}_N(k)$. В уравнении (6.80) для вычисления этого вектора используется MIL, а в уравнении (7.45) — обратное QR-разложение, определяемое с помощью приемов, показанных в следующем разделе.

7.4. Рекурсивное вычисление некоторых переменных

Из уравнения (7.45) следует, что для рекурсивного вычисления вектора ВК на каждой k -й итерации требуется вычислять переменные $\mathbf{u}_N(k)$ и $b_N(k)$. Обе эти переменные, см. (7.42) и (7.43), зависят от вектора $\mathbf{a}_N(k)$, см. (7.38), который, в свою очередь, зависит от матрицы $\tilde{\mathbf{R}}_N^{-H}(k-1)$. Поэтому для получения соотношений для рекурсивного вычисления переменных $\mathbf{u}_N(k)$ и $b_N(k)$ необходимо получить сосотношение для рекурсивного вычисления матрицы $\tilde{\mathbf{R}}_N^{-H}(k)$.

Для этого выполним перемножение левой и правой частей уравнения (7.20) как

$$\begin{aligned} & \left[\lambda^{0.5}\tilde{\mathbf{R}}_N^H(k-1), \mathbf{x}_N(k) \right] \mathbf{T}_{N+1}^H(k) \mathbf{T}_{N+1}(k) \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{R}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \\ &= \left[\lambda^{0.5}\tilde{\mathbf{R}}_N^H(k-1), \mathbf{x}_N(k) \right] \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{R}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \left[\tilde{\mathbf{R}}_N^H(k), \mathbf{0}_N(k) \right] \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{0}_N^T \end{bmatrix}.\end{aligned}\quad (7.46)$$

В уравнении (7.46) учтено, что матрица $\mathbf{T}_{N+1}(k)$ является унитарной, т. е. $\mathbf{T}_{N+1}^H(k)\mathbf{T}_{N+1}(k) = \mathbf{I}_{N+1}$. Из уравнения (7.46) следует, что

$$\lambda\tilde{\mathbf{R}}_N^H(k-1)\tilde{\mathbf{R}}_N(k-1) + \mathbf{x}_N(k)\mathbf{x}_N^H(k) = \tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k). \quad (7.47)$$

Легко заметить, что (7.46), подобно (6.75), представляет собой рекурсивное уравнение вычисления произведения матриц $\tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k)$. Из сравнения уравнений (7.47) и (6.75) следует, что

$$\mathbf{R}_N(k) = \tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k). \quad (7.48)$$

Здесь необходимо напомнить, что $\tilde{\mathbf{R}}_N(k)$ — это верхняя треугольная матрица. Следовательно, $\tilde{\mathbf{R}}_N^T(k)$ — это нижняя треугольная матрица. Поэтому

$\tilde{\mathbf{R}}_N^{-1}(k)$ — это также верхняя треугольная матрица, а $\tilde{\mathbf{R}}_N^{-H}(k)$ — нижняя треугольная матрица.

Для обращения матрицы $\tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k)$ по аналогии с уравнениями (6.75) и (6.76), используется MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$ где $\mathbf{C} = \lambda \tilde{\mathbf{R}}_N^H(k-1) \tilde{\mathbf{R}}_N(k-1)$, $\mathbf{C}^{-1} = \lambda^{-1} \tilde{\mathbf{R}}_N^{-1}(k-1) \tilde{\mathbf{R}}_N^{-H}(k-1)$, $\mathbf{B}^H = \mathbf{x}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k)$ и $\mathbf{A}^{-1} = -1$.

Тогда, с учетом уравнений (7.42) и (7.43) имеем, что

$$\begin{aligned} [\tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k)]^{-1} &= \tilde{\mathbf{R}}_N^{-1}(k)\tilde{\mathbf{R}}_N^{-H}(k) = \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) - \\ &- \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k)\tilde{\mathbf{R}}_N^{-H}(k-1)\mathbf{x}_N(k) \left[1 + \mathbf{x}_N^H(k)\lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1)\mathbf{x}_N(k) \right]^{-1} \times \\ &\times \mathbf{x}_N^H(k)\lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) = \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) - \\ &- \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-1}(k)\mathbf{a}_N(k) \left[1 + \mathbf{a}_N^H(k)\mathbf{a}_N(k) \right]^{-1} \lambda^{-0.5}\mathbf{a}_N^H(k)\tilde{\mathbf{R}}_N^{-H}(k-1) = \\ &= \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) - \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k)\mathbf{a}_N(k)b_N^{-2}(k)\mathbf{a}_N^H(k)\tilde{\mathbf{R}}_N^{-H}(k-1) = \\ &= \lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) - \mathbf{u}_N(k)\mathbf{u}_N^H(k). \end{aligned} \quad (7.49)$$

Из (7.49) следует, что

$$\lambda^{-1}\tilde{\mathbf{R}}_N^{-1}(k-1)\tilde{\mathbf{R}}_N^{-H}(k-1) = \tilde{\mathbf{R}}_N^{-1}(k)\tilde{\mathbf{R}}_N^{-H}(k) + \mathbf{u}_N(k)\mathbf{u}_N^H(k). \quad (7.50)$$

Уравнение (7.50) подразумевает, что существует такая унитарная матрица $\mathbf{P}_{N+1}(k)$, что

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N^{-H}(k) \\ \mathbf{u}_N^H(k) \end{bmatrix}. \quad (7.51)$$

Действительно, из (7.51) следует, что

$$\begin{aligned} &[\lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1), \mathbf{0}_N^T] \mathbf{P}_{N+1}^H(k) \mathbf{P}_{N+1}(k) \begin{bmatrix} \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \\ &= [\lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1), \mathbf{0}_N^T] \begin{bmatrix} \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = [\tilde{\mathbf{R}}_N^{-1}(k), \mathbf{u}_N(k)] \begin{bmatrix} \tilde{\mathbf{R}}_N^{-H}(k) \\ \mathbf{u}_N^H(k) \end{bmatrix}. \end{aligned} \quad (7.52)$$

Выражение (7.52) эквивалентно выражению (7.50), что подтверждает справедливость выражения (7.51).

Различие между унитарными матрицами $\mathbf{T}_{N+1}(k)$ и $\mathbf{P}_{N+1}(k)$ заключается в следующем. Уравнение (7.20) определяет матрицу $\mathbf{T}_{N+1}(k)$ как последовательность вращений Гивенса, которые последовательно обнуляют вектор $\mathbf{x}_N^H(k)$. В уравнении (7.51) элементы последней строки матрицы в правой части уравнения не обнуляются, а вводятся, т. е. формируется вектор $\mathbf{u}_N^H(k)$.

Уравнение (7.51) подразумевает, что матрица $\mathbf{P}_{N+1}(k)$ имеет структуру вида

$$\mathbf{P}_{N+1}(k) = \begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix}, \quad (7.53)$$

где $w_N(k)$ — действительная скалярная величина. Применяя (7.53) в уравнении (7.51), получим, что

$$\begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix} \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \lambda^{-0.5} \mathbf{E}_N(k) \tilde{\mathbf{R}}_N^{-H}(k-1) \\ \lambda^{-0.5} \mathbf{j}_N^H(k) \tilde{\mathbf{R}}_N^{-H}(k-1) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N^{-H}(k) \\ \mathbf{u}_N^H(k) \end{bmatrix}, \quad (7.54)$$

откуда

$$\lambda^{-0.5} \mathbf{E}_N(k) \tilde{\mathbf{R}}_N^{-H}(k-1) = \tilde{\mathbf{R}}_N^{-H}(k) \quad (7.55)$$

и

$$\lambda^{-0.5} \mathbf{j}_N^H(k) \tilde{\mathbf{R}}_N^{-H}(k-1) = \mathbf{u}_N^H(k). \quad (7.56)$$

Переменные, входящие в состав матрицы $\mathbf{P}_{N+1}(k)$, могут быть найдены из условия унитарности этой матрицы

$$\begin{aligned} \mathbf{P}_{N+1}(k) \mathbf{P}_{N+1}^H(k) &= \begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix}^H = \\ &= \begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{E}_N^H(k) & \mathbf{j}_N(k) \\ \mathbf{f}_N^H(k) & w_N(k) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N^T & 1 \end{bmatrix} = \mathbf{I}_{N+1}. \end{aligned} \quad (7.57)$$

Используя (7.43), уравнение (7.56) можно представить как

$$\lambda^{-0.5} \mathbf{j}_N^H(k) \tilde{\mathbf{R}}_N^{-H}(k-1) = \left[\lambda^{-0.5} \tilde{\mathbf{R}}_N^{-1}(k-1) \mathbf{a}_N(k) b_N^{-1}(k) \right]^H = \lambda^{-0.5} \mathbf{a}_N^H(k) \tilde{\mathbf{R}}_N^{-H}(k-1) b_N^{-1}(k), \quad (7.58)$$

откуда следует, что

$$\mathbf{j}_N^H(k) = \mathbf{a}_N^H(k) b_N^{-1}(k). \quad (7.59)$$

Из уравнения (7.57) следует, что

$$\mathbf{j}_N^H(k) \mathbf{j}_N(k) + w_N^2(k) = 1. \quad (7.60)$$

Далее, используя выражение (7.59), уравнение (7.60) можно преобразовать как

$$\mathbf{a}_N^H(k) \mathbf{a}_N(k) b_N^{-2}(k) + w_N^2(k) = 1. \quad (7.61)$$

Из уравнений (7.61) и (7.42) следует, что

$$\mathbf{a}_N^H(k) \mathbf{a}_N(k) + b_N^2(k) w_N^2(k) = b_N^2(k) = 1 + \mathbf{a}_N^H(k) \mathbf{a}_N(k), \quad (7.62)$$

а из уравнения (7.62) — что $w_N^2(k) = b_N^{-2}(k)$ или

$$w_N(k) = b_N^{-1}(k). \quad (7.63)$$

Переменная $\delta_N(k)$ в уравнении (7.23) и переменная $w_N(k)$ в уравнении (7.53), как ранее отмечалось, являются действительными скалярными величинами. Это будет доказано далее при рассмотрении структур матриц $\mathbf{T}_{N+1}(k)$ и $\mathbf{P}_{N+1}(k)$,

составленных из произведения матриц, содержащих в себе элементы матриц вращения Гивенса.

Из уравнения (7.57) также следует, что

$$\mathbf{E}_N(k)\mathbf{j}_N(k) + \mathbf{f}_N(k)w_N(k) = \mathbf{0}_N. \quad (7.64)$$

Используя в (7.64) уравнения (7.59) и (7.63), получим, что

$$\mathbf{E}_N(k)\mathbf{a}_N(k)b_N^{-1}(k) + \mathbf{f}_N(k)b_N^{-1}(k) = \mathbf{0}_N, \quad (7.65)$$

откуда

$$\mathbf{f}_N(k) = -\mathbf{E}_N(k)\mathbf{a}_N(k). \quad (7.66)$$

Далее, используя полученные результаты и применив преобразование, аналогичное (7.51), к вектору $[\mathbf{a}_N^T(k), 1]^T$, получим, что

$$\begin{aligned} \mathbf{P}_{N+1}(k) \begin{bmatrix} \mathbf{a}_N(k) \\ 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{E}_N(k) & \mathbf{f}_N(k) \\ \mathbf{j}_N^H(k) & w_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{a}_N(k) \\ 1 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{E}_N(k) & -\mathbf{E}_N(k)\mathbf{a}_N(k) \\ \mathbf{a}_N^H(k)b_N^{-1}(k) & b_N^{-1}(k) \end{bmatrix} \begin{bmatrix} \mathbf{a}_N(k) \\ 1 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{E}_N(k)\mathbf{a}_N(k) - \mathbf{E}_N(k)\mathbf{a}_N(k) \\ \mathbf{a}_N^H(k)\mathbf{a}_N(k)b_N^{-1}(k) + b_N^{-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ b_N^2(k)b_N^{-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ b_N(k) \end{bmatrix}. \end{aligned} \quad (7.67)$$

Уравнения (7.51) и (7.67) можно представить в компактной форме как

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \mathbf{a}_N(k) & \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1) \\ 1 & \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N & \tilde{\mathbf{R}}_N^{-H}(k) \\ b_N(k) & \mathbf{u}_N^H(k) \end{bmatrix}. \quad (7.68)$$

7.5. Обратное QR-разложение

Уравнение (7.68) представляет собой IQRD RLS-алгоритм, позволяющий вычислять вектор коэффициентов Калмана $\mathbf{g}_N(k) = \mathbf{u}_N(k)b_N^{-1}(k)$ и, как следствие, вычислять вектор ВК $\mathbf{h}_N(k)$ с помощью уравнения (7.45), т. е. без обратных подстановок. Для каждого индекса дискретного времени k эти вычисления последовательно осуществляются с помощью следующих шагов.

Шаг 1. С помощью уравнения (7.38) вычисляется вектор $\mathbf{a}_N(k)$, для чего используется матрица $\tilde{\mathbf{R}}_N^{-H}(k-1)$, полученная на предыдущей итерации $k-1$, и вектор входных сигналов адаптивного фильтра $\mathbf{x}_N(k)$ на текущей итерации k .

Шаг 2. Формируется матрица

$$\begin{bmatrix} \mathbf{a}_N(k) & \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1) \\ 1 & \mathbf{0}_N^T \end{bmatrix}.$$



Шаг 3. Выполняется преобразование (7.68), где

$$\mathbf{P}_{N+1}(k) = \prod_{i=1}^N \mathbf{P}_{N+1}^{(i)}(k). \quad (7.69)$$

Умножение в левой части уравнения (7.68) производится в течение N итераций, в результате чего обнуляются первые N элементов первого столбца матрицы в левой части уравнения (7.68) и формируется матрица

$$\begin{bmatrix} \mathbf{0}_N & \tilde{\mathbf{R}}_N^{-H}(k) \\ b_N(k) & \mathbf{u}_N^H(k) \end{bmatrix}$$

в правой части этого уравнения.

Шаг 4. После выполнения шага 3 в уравнении (7.68) образуются переменные $\mathbf{u}_N(k)$ и $b_N(k)$, необходимые для вычисления вектора $\mathbf{g}_N(k)$, который затем используется в уравнении вычисления ВК $\mathbf{h}_N(k)$ (7.45), а также образуется матрица $\tilde{\mathbf{R}}_N^{-H}(k)$, используемая на следующей $(k+1)$ -й итерации IQRD RLS-алгоритма.

Используемые в шаге 3 матрицы $\mathbf{P}_{N+1}^{(i)}(k)$ содержат элементы матрицы вращения Гивенса [99].

В случае арифметики действительных чисел матрица Гивенса определяется как

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \text{ или } \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad (7.70)$$

где переменные $c=\cos\theta$ и $s=\sin\theta$ удовлетворяют известному тригонометрическому соотношению

$$c^2 + s^2 = 1, \quad (7.71)$$

так как эта матрица является ортогональной. Умножение вектора на эту матрицу эквивалентно его повороту на угол θ по часовой стрелке, что дало название этой матрице как матрица вращения.

Действительно, для вектора $\mathbf{a} = [a_1, a_2]^T$ в произведении

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \mathbf{a} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} ca_1 + sa_2 \\ -sa_1 + ca_2 \end{bmatrix} = \mathbf{a}', \quad (7.72)$$

норма повернутого вектора \mathbf{a}' , т. е.

$$\begin{aligned} \|\mathbf{a}'\|_2 &= \sqrt{\mathbf{a}'^T \mathbf{a}'} = \sqrt{[ca_1 + sa_2, -sa_1 + ca_2] \begin{bmatrix} ca_1 + sa_2 \\ -sa_1 + ca_2 \end{bmatrix}} = \sqrt{(ca_1 + sa_2)^2 + (-sa_1 + ca_2)^2} = \\ &= \sqrt{c^2 a_1^2 + 2csa_1 a_2 + s^2 a_2^2 + s^2 a_1^2 - 2csa_1 a_2 + c^2 a_2^2} = \sqrt{a_1^2 (c^2 + s^2) + a_2^2 (c^2 + s^2)} = \sqrt{a_1^2 + a_2^2}, \end{aligned} \quad (7.73)$$

совпадает с нормой исходного вектора \mathbf{a} , т. е.

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T \mathbf{a}} = \sqrt{\begin{bmatrix} a_1, a_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}} = \sqrt{a_1^2 + a_2^2}. \quad (7.74)$$

Следовательно,

$$\left\| \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \mathbf{a} \right\|_2 = \|\mathbf{a}\|_2. \quad (7.75)$$

Угол θ между векторами \mathbf{a} и \mathbf{a}' можно найти из скалярного произведения этих векторов как

$$\begin{aligned} \cos \theta &= \frac{\mathbf{a}^T \mathbf{a}'}{\|\mathbf{a}\|_2 \|\mathbf{a}'\|_2} = \left([a_1, a_2] \begin{bmatrix} ca_1 + sa_2 \\ -sa_1 + ca_2 \end{bmatrix} \right) / \left(\sqrt{a_1^2 + a_2^2} \sqrt{a_1^2 + a_2^2} \right) = \\ &= (ca_1^2 + sa_1a_2 - sa_1a_2 + ca_2^2) / (a_1^2 + a_2^2) = c(a_1^2 + a_2^2) / (a_1^2 + a_2^2) = c. \end{aligned} \quad (7.76)$$

Таким образом, из уравнений (7.75) и (7.76) следует, что матрица $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ поворачивает вектор \mathbf{a} на угол θ , не меняя нормы этого вектора. Геометрическая интерпретация вращения показана на рис. 7.1 [109]. Вращение вектора \mathbf{a} по часовой стрелке (см. рис. 7.1a) эквивалентно вращению исходной системы координат $\mathbf{e}_1, \mathbf{e}_2$ против часовой стрелки на тот же угол (см. рис. 7.1b).

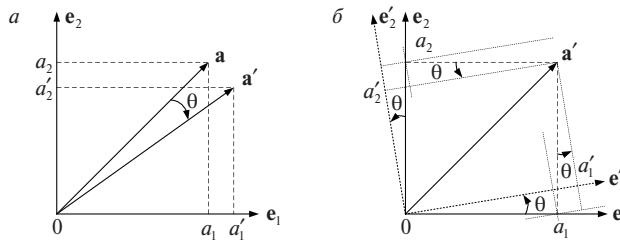


Рис. 7.1. Геометрия вращений Гивенса

Из рис. 7.1б следует, что координаты вектора \mathbf{a}' в повернутой системе координат $\mathbf{e}'_1, \mathbf{e}'_2$ определяются как

$$a'_1 = ca_1 + sa_2 \quad (7.77)$$

и

$$a'_2 = -sa_1 + ca_2. \quad (7.78)$$

Эти выражения и рис. 7.1б показывают, что при повороте координата a'_1 увеличивается, а координата a'_2 уменьшается. Это означает, что матрица $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ поворачивает векторы именно по часовой стрелке. Аналогично можно показать, что матрица $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ поворачивает векторы против часовой стрелки.

Первая матрица обычно используется в алгоритмах на основе прямого QR-разложения, а вторая — в алгоритмах на основе обратного QR-разложения. Далее везде для однообразия мы будем использовать вторую матрицу в алгоритмах на основе как прямого, так и обратного QR-разложения.



В случае комплексной арифметики матрицу Гивенса определяют как единичную матрицу

$$\begin{bmatrix} c & s^* \\ -s & c \end{bmatrix} \text{ или } \begin{bmatrix} c & -s^* \\ s & c \end{bmatrix}, \quad (7.79)$$

где переменные c и s также удовлетворяют тригонометрическому соотношению

$$c^2 + ss^* = c^2 + |s|^2 = 1, \quad (7.80)$$

переменная c является действительной величиной и $0 \leq c \leq 1$, переменная

$$s = |s|e^{i\varphi_s} \quad (7.81)$$

— комплексной величиной, в которой $0 \leq |s| \leq 1$, а значение φ_s определяется как это будет показано дальше.

Норма повернутого вектора

$$\begin{bmatrix} c & s^* \\ -s & c \end{bmatrix} \mathbf{a} = \begin{bmatrix} c & s^* \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} ca_1 + s^* a_2 \\ -sa_1 + ca_2 \end{bmatrix} = \mathbf{a}', \quad (7.82)$$

определяемая как

$$\begin{aligned} \|\mathbf{a}'\|_2 &= \sqrt{\mathbf{a}'^H \mathbf{a}'} = \sqrt{\left[(ca_1 + s^* a_2)^*, (-sa_1 + ca_2)^* \right] \begin{bmatrix} ca_1 + s^* a_2 \\ -sa_1 + ca_2 \end{bmatrix}} = \\ &= \sqrt{(ca_1 + s^* a_2)^* (ca_1 + s^* a_2) + (-sa_1 + ca_2)^* (-sa_1 + ca_2)} = \\ &= \sqrt{c^2 |a_1|^2 + c s a_1 a_2^* + c s^* a_1^* a_2 + |s|^2 |a_2|^2 + |s|^2 |a_1|^2 - c s a_1 a_2^* - c s^* a_1^* a_2 + c^2 |a_2|^2} = \\ &= \sqrt{(c^2 + |s|^2) |a_1|^2 + (c^2 + |s|^2) |a_2|^2} = \sqrt{|a_1|^2 + |a_2|^2}, \end{aligned} \quad (7.83)$$

совпадает с нормой исходного вектора \mathbf{a} , т. е.

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^H \mathbf{a}} = \sqrt{\left[a_1^*, a_2^* \right] \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}} = \sqrt{|a_1|^2 + |a_2|^2}. \quad (7.84)$$

Следовательно,

$$\left\| \begin{bmatrix} c & s^* \\ -s & c \end{bmatrix} \mathbf{a} \right\|_2 = \|\mathbf{a}\|_2. \quad (7.85)$$

Угол θ между векторами \mathbf{a} и \mathbf{a}' можно найти из их скалярного произведения как

$$\begin{aligned} \cos \theta &= \frac{\mathbf{a}^H \mathbf{a}'}{\|\mathbf{a}\|_2 \|\mathbf{a}'\|_2} = \left(\left[a_1^*, a_2^* \right] \begin{bmatrix} ca_1 + s^* a_2 \\ -sa_1 + ca_2 \end{bmatrix} \right) \Big/ \left(\sqrt{|a_1|^2 + |a_2|^2} \sqrt{|a_1|^2 + |a_2|^2} \right) = \\ &= (c |a_1|^2 + s^* a_1^* a_2 - s a_1 a_2^* + c |a_2|^2) \Big/ (|a_1|^2 + |a_2|^2) = \\ &= \left[c (|a_1|^2 + |a_2|^2) - j 2 \operatorname{Im} \{ s a_1 a_2^* \} \right] \Big/ (|a_1|^2 + |a_2|^2). \end{aligned} \quad (7.86)$$

Чтобы $\cos\theta$ оставался действительной величиной и $0 \leq |\cos\theta| \leq 1$, необходимо, чтобы

$$\operatorname{Im}\{sa_1a_2^*\} = 0. \quad (7.87)$$

Это выполняется при $\varphi_s + \varphi_{a_1} - \varphi_{a_2} = 0$, откуда следует, что

$$\varphi_s = \varphi_{a_2} - \varphi_{a_1}, \quad (7.88)$$

где φ_s , φ_{a_1} и φ_{a_2} — фазы комплексных переменных s , a_1 и a_2 , соответственно.

Таким образом, из уравнений (7.85)–(7.88) также следует, что матрица $\begin{bmatrix} c & s^* \\ -s & c \end{bmatrix}$ поворачивает вектор \mathbf{a} на угол θ , не меняя нормы этого вектора.

Матрица $\mathbf{P}_{N+1}(k)$ в уравнении (7.68) может быть сформирована с помощью матриц вращения Гивенса (7.79) следующим образом. Элементы этой матрицы можно определить, используя значения вектора $\mathbf{a}_N(k)$, полученные на первом шаге IQRD RLS-алгоритма. Действительно, уравнение (7.67) устанавливает, что

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \mathbf{a}_N(k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ b_N(k) \end{bmatrix}, \quad (7.89)$$

а из уравнения (7.89) следует, что

$$\left\| \begin{bmatrix} \mathbf{a}_N(k) \\ 1 \end{bmatrix} \right\|_2 = \sqrt{1 + \mathbf{a}_N^H(k)\mathbf{a}_N(k)} = b_N(k) = \left\| \begin{bmatrix} \mathbf{0}_N \\ b_N(k) \end{bmatrix} \right\|_2. \quad (7.90)$$

Это означает, что умножение на матрицу $\mathbf{P}_{N+1}(k)$ не меняет норму вектора $[\mathbf{a}_N^T(k), 1]^T$. Следовательно, возможным кандидатом для построения этой матрицы может служить матрица вращения Гивенса (7.79).

Покажем, как на каждой итерации k по времени можно выполнять преобразование (7.89) за N шагов. В течение этих шагов осуществляется обнуление элементов вектора $\mathbf{a}_N(k)$ в векторе $[\mathbf{a}_N^T(k), 1]^T$ и формирование вектора $[\mathbf{0}_N^T, b_N(k)]^T$.

Пусть на первом из N упомянутых шагов выполняется преобразование, которое обнуляет первый элемент вектора $[\mathbf{a}_N^T(k), 1]^T$ как

$$\mathbf{P}_{N+1}^{(1)}(k) \begin{bmatrix} a_{N,1}(k) \\ a_{N,2}(k) \\ \vdots \\ a_{N,N}(k) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ a_{N,2}(k) \\ \vdots \\ a_{N,N}(k) \\ b_N^{(1)}(k) \end{bmatrix}, \quad (7.91)$$

где

$$b_N^{(1)}(k) = \sqrt{1 + \sum_{l=1}^i a_{N,l}^*(k)a_{N,l}(k)}. \quad (7.92)$$



Здесь и далее нижний индекс N показывает зависимость скалярной величины от N -мерных вычислений, связанных с размером вектора ВК $\mathbf{h}_N(k)$. Верхний индекс (i) указывает на i -й шаг вычислений (в данном случае связанных с вращениями Гивенса), а нижний индекс i — на номер элемента в векторе или матрице, где $i=1, 2, \dots, N$.

Для вычисления параметра $b_N^{(i)}(k)$ вместо выражения (7.92) может быть использована рекурсивная формула

$$b_N^{(i)}(k) = \sqrt{\left[b_N^{(i-1)}(k) \right]^2 + a_{N,i}^*(k) a_{N,i}(k)}, \quad (7.93)$$

где $b_N^{(0)}(k)=1$.

Из уравнения (7.93) следует, что

$$b_N^{(N)}(k) = \sqrt{1 + \sum_{l=1}^N a_{N,l}^*(k) a_{N,l}(k)} = \sqrt{1 + \mathbf{a}_N^H(k) \mathbf{a}_N(k)} = b_N(k). \quad (7.94)$$

Так как в уравнении (7.91) в результате умножения на матрицу $\mathbf{P}_{N+1}^{(1)}(k)$ видоизменяются только два элемента вектора $[\mathbf{a}_N^T(k), 1]^T$ (первый и последний), то эту матрицу можно сформировать из элементов матрицы Гивенса (7.79) как

$$\mathbf{P}_{N+1}^{(1)}(k) = \begin{bmatrix} c_{N,1}(k) & 0 & \cdots & 0 & -s_{N,1}^*(k) \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \mathbf{I}_{N-4} & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ s_{N,1}(k) & 0 & \cdots & 0 & c_{N,1}(k) \end{bmatrix}. \quad (7.95)$$

Тогда из уравнения (7.91) следует, что

$$c_{N,1}(k) a_{N,1}(k) - s_{N,1}^*(k) = 0, \quad (7.96)$$

откуда

$$s_{N,1}^*(k) = c_{N,1}(k) a_{N,1}(k) \quad (7.97)$$

или

$$s_{N,1}(k) = c_{N,1}(k) a_{N,1}^*(k). \quad (7.98)$$

Из уравнения (7.91) также следует, что

$$s_{N,1}(k) a_{N,1}(k) + c_{N,1}(k) = b_N^{(1)}(k), \quad (7.99)$$

а с учетом (7.98) — что

$$\begin{aligned} c_{N,1}(k) a_{N,1}^*(k) a_{N,1}(k) + c_{N,1}(k) &= c_{N,1}(k) \left[a_{N,1}^*(k) a_{N,1}(k) + 1 \right] = \\ &= c_{N,1}(k) \left[b_N^{(1)}(k) \right]^2 = b_N^{(1)}(k), \end{aligned} \quad (7.100)$$

откуда

$$c_{N,1}(k) = b_N^{(1)}(k) \left[b_N^{(1)}(k) \right]^{-2} = \left[b_N^{(1)}(k) \right]^{-1} = b_N^{(0)}(k) \left[b_N^{(1)}(k) \right]^{-1}. \quad (7.101)$$

Используя (7.101), уравнения (7.97) и (7.98) можно представить как

$$s_{N,1}^*(k) = c_{N,1}(k) a_{N,1}(k) = \left[b_N^{(1)}(k) \right]^{-1} a_{N,1}(k) \quad (7.102)$$

и

$$s_{N,1}(k) = c_{N,1}(k) a_{N,1}^*(k) = \left[b_N^{(1)}(k) \right]^{-1} a_{N,1}^*(k). \quad (7.103)$$

Чтобы получить представление о том, как выполнять обозначенные выше вычисления в общем виде, рассмотрим еще один шаг при $i=2$, т. е. выполним действие

$$\mathbf{P}_{N+1}^{(2)}(k) \mathbf{P}_{N+1}^{(1)}(k) \begin{bmatrix} a_{N,1}(k) \\ a_{N,2}(k) \\ a_{N,3}(k) \\ \vdots \\ a_{N,N}(k) \\ 1 \end{bmatrix} = \mathbf{P}_{N+1}^{(1)}(k) \begin{bmatrix} 0 \\ a_{N,2}(k) \\ a_{N,3}(k) \\ \vdots \\ a_{N,N}(k) \\ b_N^{(1)}(k) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ a_{N,3}(k) \\ \vdots \\ a_{N,N}(k) \\ b_N^{(2)}(k) \end{bmatrix}, \quad (7.104)$$

где

$$\mathbf{P}_{N+1}^{(2)}(k) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & c_{N,2}(k) & 0 & \cdots & 0 & -s_{N,2}^*(k) \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \mathbf{I}_{N-5} & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & s_{N,2}(k) & 0 & \cdots & 0 & c_{N,2}(k) \end{bmatrix}. \quad (7.105)$$

Тогда из уравнения (7.104) следует, что

$$c_{N,2}(k) a_{N,2}(k) - s_{N,2}^*(k) b_N^{(1)}(k) = 0, \quad (7.106)$$

откуда

$$s_{N,2}^*(k) = c_{N,2}(k) a_{N,2}(k) \left[b_N^{(1)}(k) \right]^{-1} \quad (7.107)$$

или

$$s_{N,2}(k) = c_{N,2}(k) a_{N,2}^*(k) \left[b_N^{(1)}(k) \right]^{-1}. \quad (7.108)$$

Из уравнения (7.104) также следует, что

$$s_{N,2}(k) a_{N,2}(k) + c_{N,2}(k) b_N^{(1)}(k) = b_N^{(2)}(k), \quad (7.109)$$

а с учетом (7.108), что

$$\begin{aligned}
& c_{N,2}(k) a_{N,2}^*(k) \left[b_N^{(1)}(k) \right]^{-1} a_{N,2}(k) + c_{N,2}(k) b_N^{(1)}(k) = c_{N,2}(k) \left[a_{N,2}^*(k) a_{N,2}(k) \right] + \\
& + c_{N,2}(k) \left[b_N^{(1)}(k) \right]^2 = c_{N,2}(k) \left\{ a_{N,2}^*(k) a_{N,2}(k) + \left[b_N^{(1)}(k) \right]^2 \right\} = \\
& = c_{N,2}(k) \left[b_N^{(2)}(k) \right]^{-2} = b_N^{(1)}(k) b_N^{(2)}(k),
\end{aligned} \tag{7.110}$$

откуда

$$c_{N,2}(k) = b_N^{(1)}(k) b_N^{(2)}(k) \left[b_N^{(2)}(k) \right]^{-2} = b_N^{(1)}(k) \left[b_N^{(2)}(k) \right]^{-1}. \tag{7.111}$$

С учетом (7.111), уравнения (7.107) и (7.108) можно представить как

$$\begin{aligned}
s_{N,2}^*(k) &= c_{N,2}(k) a_{N,2}(k) \left[b_N^{(1)}(k) \right]^{-1} = \\
&= b_N^{(1)}(k) \left[b_N^{(2)}(k) \right]^{-1} a_{N,2}(k) \left[b_N^{(1)}(k) \right]^{-1} = \left[b_N^{(2)}(k) \right]^{-1} a_{N,2}(k)
\end{aligned} \tag{7.112}$$

и

$$s_{N,2}(k) = \left[b_N^{(2)}(k) \right]^{-1} a_{N,2}^*(k). \tag{7.113}$$

Аналогичные преобразования можно выполнить для всех $i=1, 2, \dots, N$. Однако из анализа уравнений (7.101), (7.103), (7.104), (7.111) и (7.113) уже можно заметить, что

$$c_{N,i}(k) = b_N^{(i-1)}(k) \left[b_N^{(i)}(k) \right]^{-1} \tag{7.114}$$

и

$$s_{N,i}(k) = \left[b_N^{(i)}(k) \right]^{-1} a_{N,i}^*(k), \tag{7.115}$$

а

$$\mathbf{P}_{N+1}^{(i)}(k) = \prod_{l=1}^i \mathbf{P}_{N+1}^{(l)}(k). \tag{7.116}$$

Таким образом, уравнения (7.94), (7.114)–(7.116) позволяют определить элементы матрицы вращения Гивенса (7.79), вычислить за N шагов матрицу

$$\mathbf{P}_{N+1}(k) = \prod_{l=1}^N \mathbf{P}_{N+1}^{(l)}(k) = \prod_{l=1}^N \begin{bmatrix} \mathbf{I}_{l-1} & \mathbf{0}_{l-1} & \mathbf{O}_{l-1, N-l-2} & \mathbf{0}_{l-1} \\ \mathbf{0}_{l-1}^T & c_{N,l}(k) & \mathbf{0}_{N-l-2}^T & -s_{N,l}^*(k) \\ \mathbf{O}_{N-l-2, l-1} & \mathbf{0}_{N-l-2} & \mathbf{I}_{N-l-2} & \mathbf{0}_{N-l-2} \\ \mathbf{0}_{l-1}^T & s_{N,l}(k) & \mathbf{0}_{N-l-2}^T & c_{N,l}(k) \end{bmatrix} \tag{7.117}$$

и в конечном итоге выполнить преобразование (7.89). Из уравнения (7.117) следует, что правый нижний элемент матрицы $\mathbf{P}_{N+1}(k)$ определяется как $\prod_{l=1}^N c_{N,l}(k)$, т. е. представляет собой произведение косинусов углов вращения, а значит, является действительной скалярной величиной. Следовательно, сделанное ранее утверждение, что элемент $w_N(k)$ матрицы (7.53), равный правому нижнему элементу матрицы (7.117), есть действительная скалярная величина, является верным.

7.6. Вычисление обратного разложения Холецкого

Матрица (7.117) согласно уравнению (7.68) также может быть использована для обновления матрицы $\tilde{\mathbf{R}}_N^{-H}(k-1)$, т. е. получения матрицы $\tilde{\mathbf{R}}_N^{-H}(k)$ и вычисления вектора $\mathbf{u}_N^H(k)$.

Как уже отмечалось, $\tilde{\mathbf{R}}_N^{-H}(k)$ — это нижняя треугольная матрица. Тогда

$$\begin{aligned}
 & \mathbf{P}_{N+1}^{(1)}(k) \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} c_{N,1}(k) & 0 & \cdots & 0 & -s_{N,1}^*(k) \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \mathbf{I}_{N-3} & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ s_{N,1}(k) & 0 & \cdots & 0 & c_{N,1}(k) \end{bmatrix} \times \\
 & \times \begin{bmatrix} \frac{\tilde{R}_{N,11}^{-H}(k-1)}{\lambda^{0.5}} & 0 & 0 & \cdots & 0 \\ \frac{\tilde{R}_{N,21}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,22}^{-H}(k-1)}{\lambda^{0.5}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{R}_{N,N1}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N2}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N3}^{-H}(k-1)}{\lambda^{0.5}} & \cdots & \frac{\tilde{R}_{N,NN}^{-H}(k-1)}{\lambda^{0.5}} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \\
 & = \begin{bmatrix} c_{N,1}(k) \lambda^{-0.5} \frac{\tilde{R}_{N,11}^{-H}(k-1)}{\lambda^{0.5}} & 0 & 0 & \cdots & 0 \\ \frac{\tilde{R}_{N,21}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,22}^{-H}(k-1)}{\lambda^{0.5}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{R}_{N,N1}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N2}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N3}^{-H}(k-1)}{\lambda^{0.5}} & \cdots & \frac{\tilde{R}_{N,NN}^{-H}(k-1)}{\lambda^{0.5}} \\ s_{N,1}(k) \tilde{R}_{N,11}^{-H}(k-1) & 0 & 0 & \cdots & 0 \end{bmatrix} = \\
 & = \begin{bmatrix} \tilde{R}_{N,11}^{-H}(k) & 0 & 0 & \cdots & 0 \\ \frac{\tilde{R}_{N,21}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,22}^{-H}(k-1)}{\lambda^{0.5}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{R}_{N,N1}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N2}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N3}^{-H}(k-1)}{\lambda^{0.5}} & \cdots & \frac{\tilde{R}_{N,NN}^{-H}(k-1)}{\lambda^{0.5}} \\ u_{N,1}^{(1)*} & 0 & 0 & \cdots & 0 \end{bmatrix},
 \end{aligned} \tag{7.118}$$

т. е.

$$\tilde{R}_{N,11}^{-H}(k) = c_{N,1}(k) \lambda^{-0.5} \tilde{R}_{N,11}^{-H}(k-1) \tag{7.119}$$



и

$$u_{N,1}^{(1)*} = s_{N,1}(k) \lambda^{-0.5} \tilde{R}_{N,11}^{-H}(k-1), \quad (7.120)$$

где символами $\tilde{R}_{N,ij}^{-H}$ обозначены элементы матрицы $\tilde{\mathbf{R}}_N^{-H}(k)$.

Аналогично, выполнив умножение уравнения (7.118) на матрицу $\mathbf{P}_{N+1}^{(2)}(k)$ как

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & c_{N,2}(k) & 0 & \cdots & 0 & -s_{N,2}^*(k) \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \mathbf{I}_{N-4} & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & s_{N,2}(k) & 0 & \cdots & 0 & c_{N,2}(k) \end{bmatrix} \times \begin{bmatrix} \tilde{R}_{N,11}^{-H}(k) & 0 & 0 & \cdots & 0 \\ \frac{\tilde{R}_{N,21}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,22}^{-H}(k-1)}{\lambda^{0.5}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\tilde{R}_{N,N1}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N2}^{-H}(k-1)}{\lambda^{0.5}} & \frac{\tilde{R}_{N,N3}^{-H}(k-1)}{\lambda^{0.5}} & \cdots & \frac{\tilde{R}_{N,NN}^{-H}(k-1)}{\lambda^{0.5}} \\ u_{N,1}^{(1)*} & 0 & 0 & \cdots & 0 \end{bmatrix},$$

получим, что

$$\tilde{R}_{N,21}^{-H}(k) = c_{N,2}(k) \lambda^{-0.5} \tilde{R}_{N,21}^{-H}(k-1) - s_{N,2}^*(k) u_{N,1}^{(1)*}, \quad (7.121)$$

$$u_{N,1}^{(2)*} = s_{N,2}(k) \lambda^{-0.5} \tilde{R}_{N,21}^{-H}(k-1) + c_{N,2}(k) u_{N,1}^{(1)*}, \quad (7.122)$$

$$\tilde{R}_{N,22}^{-H}(k) = c_{N,2}(k) \lambda^{-0.5} \tilde{R}_{N,22}^{-H}(k-1), \quad (7.123)$$

$$u_{N,2}^{(2)*} = s_{N,2}(k) \lambda^{-0.5} \tilde{R}_{N,22}^{-H}(k-1). \quad (7.124)$$

Из уравнений (7.118)–(7.120) следует, что умножение на матрицу $\mathbf{P}_{N+1}^{(1)}(k)$ обеспечивает обновление первой строки обратного разложения Холецкого и начало формирования первого элемента вектора $\mathbf{u}_N^H(k)$. Полное формирование этого элемента занимает N шагов, номер шага указан в уравнении (7.121) в виде верхнего индекса в круглых скобках. Из уравнений (7.121)–(7.124) следует, что умножение полученного результата на матрицу $\mathbf{P}_{N+1}^{(2)}(k)$ обеспечивает обновление второй строки обратного разложения Холецкого и начало формирования второго элемента вектора $\mathbf{u}_N^H(k)$. Полное формирование этого элемента занимает $N-1$ шагов, номер шага указан в уравнении (7.124) в виде верхнего индекса в круглых скобках. Таким образом, полное вычисление матрицы $\tilde{\mathbf{R}}_N^{-H}(k)$ и вектора $\mathbf{u}_N^H(k)$ на каждой итерации k выполняется за N шагов с помощью преобразования (7.68).

Следовательно, уравнения (7.38), (7.93), (7.114), (7.115), (7.121)–(7.124), (7.45), а также уравнение вычисления ошибки $\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$ составляют RLS-алгоритм на основе обратного QR-разложения.

Исходя из (7.48) и (3.50) $[\mathbf{A}_N \mathbf{B}_N]^{-1} = \mathbf{B}_N^{-1} \mathbf{A}_N^{-1}$, матрицу $\mathbf{R}_N^{-1}(k)$ можно представить как

$$\mathbf{R}_N^{-1}(k) = \tilde{\mathbf{R}}_N^{-1}(k) \tilde{\mathbf{R}}_N^{-H}(k). \quad (7.125)$$

Поэтому инициализацию матрицы $\tilde{\mathbf{R}}_N^{-H}(0)$ можно по аналогии с (6.1.0) (см. табл. 6.1) можно выполнить как $\mathbf{R}_N^{-H}(0) = \mathbf{R}_N^{-1}(0) = \delta^{-1} \mathbf{I}_N$.

Однако далее эту инициализацию будем выполнять как $\tilde{\mathbf{R}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5}$, см. уравнение (7.1.0) в табл. 7.1, где представлена вычислительная процедура рассмотренного IQRD RLS-алгоритма. Такая инициализация математически эквивалентна инициализации быстрых RLS-алгоритмов (см. раздел 14.8)

Таблица 7.1. RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса (с операциями извлечения квадратного корня)

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\tilde{\mathbf{R}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5}$, $\mathbf{h}_N(0) = \mathbf{0}_N$, $\Lambda_N = \text{diag}\{1, \lambda, \dots, \lambda^{N-1}\}$	(7.1.0)
For $k = 1, 2, \dots, K$	
$u_{N,j}^{(j-1)*}(k) = 0$, $1 \leq j \leq N$, $b_N^{(0)}(k) = 1$	(7.1.1)
For $i = 1, 2, \dots, N$	
$a_{N,i}(k) = \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \Big _{i,1:i} \mathbf{x}_N(k) \Big _{1:i}$	(7.1.2)
$b_N^{(i)}(k) = \sqrt{\left[b_N^{(i-1)}(k) \right]^2 + a_{N,i}^*(k) a_{N,i}(k)}$	(7.1.3)
$s_{N,i}(k) = \left[b_N^{(i)}(k) \right]^{-1} a_{N,i}^*(k)$	(7.1.4)
$c_{N,i}(k) = b_N^{(i-1)}(k) \left[b_N^{(i)}(k) \right]^{-1}$	(7.1.5)
For $j = 1, 2, \dots, i$	
$\tilde{\mathbf{R}}_{N,ij}^{-H}(k) = c_{N,i}(k) \lambda^{-0.5} \tilde{\mathbf{R}}_{N,ij}^{-H}(k-1) - s_{N,i}^*(k) u_{N,j}^{(i-1)*}(k)$	(7.1.6)
$u_{N,j}^{(i)*}(k) = s_{N,i}(k) \lambda^{-0.5} \tilde{\mathbf{R}}_{N,ij}^{-H}(k-1) + c_{N,i}(k) u_{N,j}^{(i-1)*}(k)$	(7.1.7)
End for j	
End for i	
$\mathbf{g}_N(k) = \mathbf{u}_N^{(N)}(k) \left[b_N^{(N)}(k) \right]^{-1}$	(7.1.8)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(7.1.9)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$	(7.1.10)
End for k	



в случае экспоненциального взвешивания обрабатываемых сигналов с помощью параметра λ .

Так как согласно уравнению (7.42) переменная $b_N(k)$ определяется как $b_N(k) = \sqrt{1 + \mathbf{a}_N^H(k)\mathbf{a}_N(k)}$, то при ее рекурсивном вычислении как $b_N^{(i)}(k) = \sqrt{\left[b_N^{(i-1)}(k)\right]^2 + a_{N,i}^*(k)a_{N,i}(k)}$, см. (7.93), вполне логично определить начальное значение $b_N^{(0)}(k) = 1$, см. уравнение (7.1.1).

Кроме того, так как согласно уравнению (7.68)

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \mathbf{a}_N(k) & \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \\ 1 & \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N & \tilde{\mathbf{R}}_N^{-H}(k) \\ b_N(k) & \mathbf{u}_N^H(k) \end{bmatrix},$$

то при рекурсивном вычислении вектора $\mathbf{u}_N^H(k)$ его начальное значение определяется как $u_{N,j}^{(j-1)*}(k) = 0$, $1 \leq j \leq N$, см. уравнение (7.1.1).

В табл. 7.1 обозначение $(\)_{i,1:i}$ указывает на номера используемых в вычислениях элементов матриц, а $(\)_{1:i}$ — на номера элементов векторов.

IQRD RLS-алгоритм (см. табл. 7.1) содержит N операций извлечения квадратного корня, которые в уравнении (7.1.3) обусловлены вычислением переменных $b_N(k)$ в течение N шагов ($i=1, \dots, N$) на каждой k -й итерации алгоритма адаптивной фильтрации. Кроме того, этот алгоритм содержит $3N^2 + 9N$ операций умножения, $1.5N^2 + 4.5N$ операций сложения и N операций деления. Если при $j=i$ вместо уравнений (7.1.6) и (7.1.7) использовать уравнения, аналогичные (7.119), (7.120), (7.123) и (7.124), т. е.

$$\tilde{\mathbf{R}}_{N,ii}^{-H}(k) = c_{N,i}(k)\lambda^{-0.5} \tilde{\mathbf{R}}_{N,ii}^{-H}(k-1), \quad (7.126)$$

$$u_{N,i}^{(2)*} = s_{N,i}(k)\lambda^{-0.5} \tilde{\mathbf{R}}_{N,ii}^{-H}(k-1), \quad (7.127)$$

то это позволяет сэкономить $2N$ операций умножения и $2N$ операций сложения. Тогда, с учетом уравнений (7.126) и (7.127), вычислительная сложность алгоритма (см. табл. 7.1) составит $3N^2 + 7N$ операций умножения, $1.5N^2 + 2.5N$ операций сложения, N операций деления и N операций извлечения квадратного корня. Таким образом, оценка вычислительной сложности IQRD RLS-алгоритма (см. табл. 7.1), подобно оценке вычислительной сложности алгоритма RLS-алгоритма на базе MIL (см. табл. 6.1), равна $O(N^2)$ арифметическим операциям.

7.7. Параллельная реализация IQRD RLS-алгоритма

Треугольная форма матрицы $\tilde{\mathbf{R}}_N^{-H}(k)$ позволяет для реализации алгоритма (см. табл. 7.1) использовать систолические вычислители. Архитектура такого вычислителя приведена в [102]. Аналогичные архитектуры обычно приводятся в большинстве работ, посвященных алгоритмам на основе различных модифи-

каций QR-разложения. Привлекательность этих архитектур обусловлена тем, что большую часть вычислений в QRD RLS-алгоритмах можно выполнять параллельно. Однако нужно понимать, что аппаратная реализация каждой операции в алгоритме (см. табл. 7.1) требует наличия соответствующего арифметического устройства. Поэтому такая реализация на сегодняшний день возможна лишь для сравнительно небольших значений N .

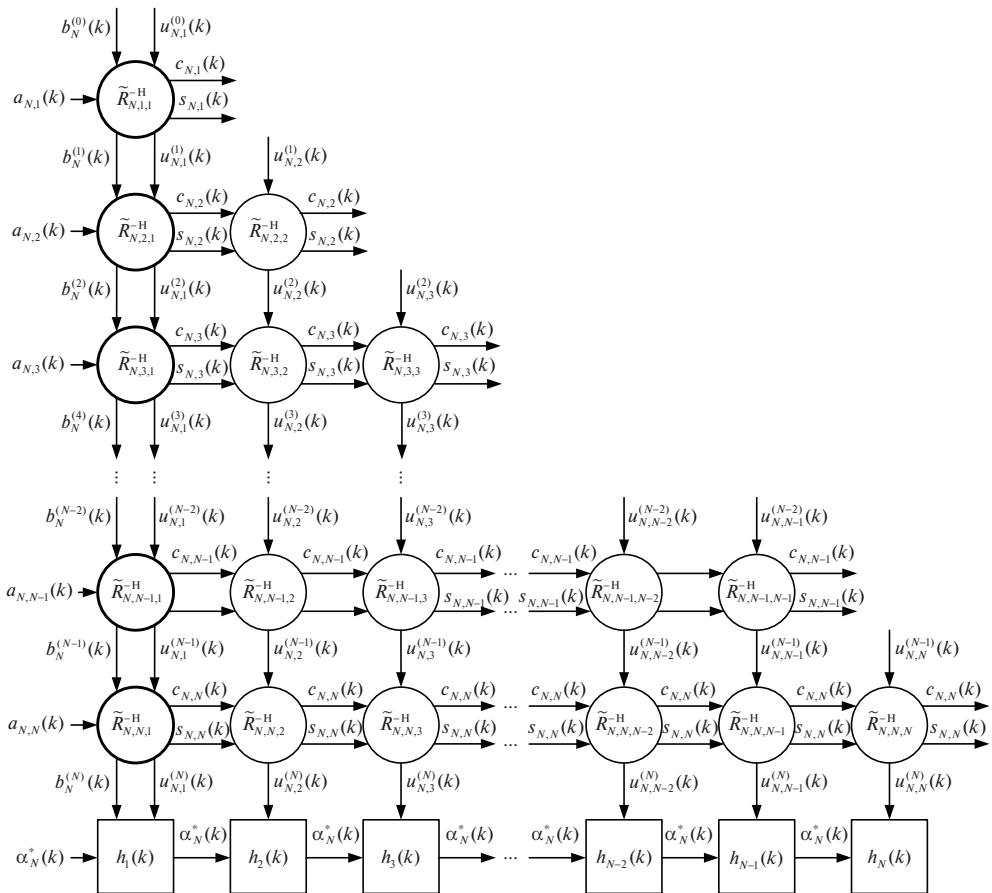


Рис. 7.2. Реализация вычислений (7.1.3)–(7.1.7) и (7.1.10)

Пример аппаратной реализации алгоритма (см. табл. 7.1) приведен на рис. 7.2, где показаны все вычисления, за исключением (7.1.2) и (7.1.9). Здесь разные ячейки характеризуются разной вычислительной сложностью. Самая большая сложность у крайнего левого столбца (кроме элементов последней строки), так как в ней присутствуют все вычисления, обозначенные в уравнениях (7.1.3)–(7.1.7).

Остальные ячейки, кроме последней строки, вычисляют только уравнения (7.1.6) и (7.1.7). В правых крайних диагональных ячейках в уравнениях (7.1.6)



и (7.1.7) вычисляются только первые слагаемые, подобно (7.126) и (7.127). Элементы ячеек нижней строки содержат вычисления из уравнения (7.1.8) (на рис. 7.2 не показаны)

$$g_{N,i}(k) = u_{N,i}^{(N)}(k) \left[b_N^{(N)}(k) \right]^{-1} \quad (7.128)$$

и из уравнения (7.1.10)

$$h_{N,i}(k) = h_{N,i}(k-1) + g_{N,i}(k) \alpha_N^*(k). \quad (7.129)$$

На рис. 7.3 для случая $N=5$ показано, как в архитектуре (см. рис. 7.2) могут быть использованы параллельные вычисления.

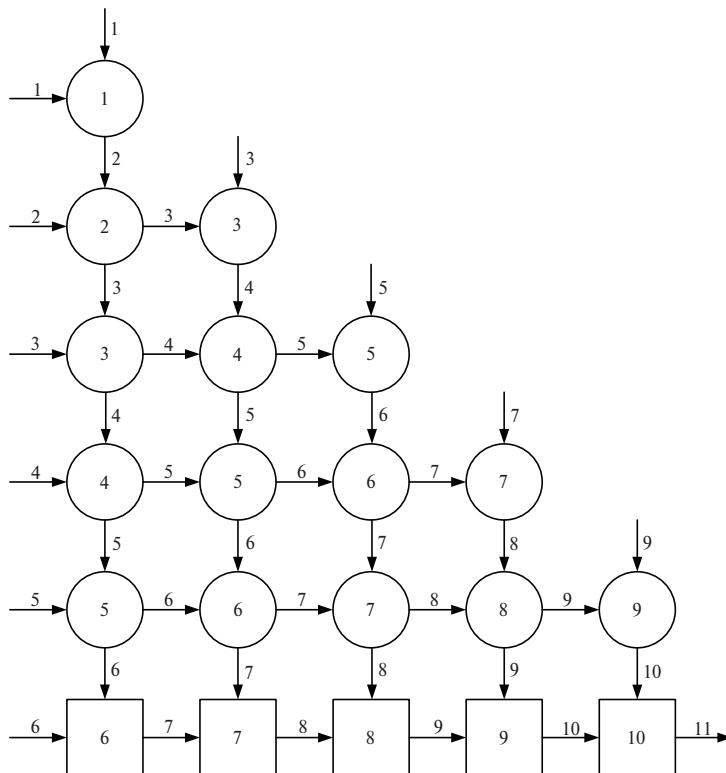


Рис. 7.3. Параллельные вычисления в IQRD RLS-алгоритме

На рис. 7.3 одинаковыми цифрами обозначены информационные потоки и соответствующие им параллельные вычисления, выполняемые одновременно. Проанализировав аналогичные архитектуры для других значений N , можно легко заметить, что для каждого значения k требуется $2N + 1$ тактов (шагов) для вычислений в соответствии со структурой на рис. 7.3. Длительность одного такта определяется вычислениями в ячейках с максимальной длительностью, т. е. содержащих наибольшее число арифметических операций, требующих

последовательного выполнения в пределах ячейки. Здесь, как уже указывалось, это ячейки крайнего левого столбца (см. рис. 7.2 и рис. 7.3).

На рис. 7.2 и рис. 7.3 отсутствуют вычисления элементов $a_{N,i}(k)$ и $\alpha_N^*(k)$. Эти вычисления также можно выполнить параллельно, так, что к нужным тактам на рис. 7.2 и рис. 7.3 они будут готовы.

Вычисление элементов $a_{N,i}(k)$ и $\alpha_N^*(k)$ показаны на рис. 7.4. Согласно данному рисунку эти вычисления требуют $N+1$ тактов (шагов). Из сравнения рис. 7.2—7.4 видно, что вычисления (см. рис. 7.3) должны быть задержаны на один такт по сравнению с вычислениями (см. рис. 7.4), так как на каждом i -м такте схемы (см. рис. 7.3) требуется наличие переменной $a_{N,i}(k)$, вычисление которой занимает один такт.

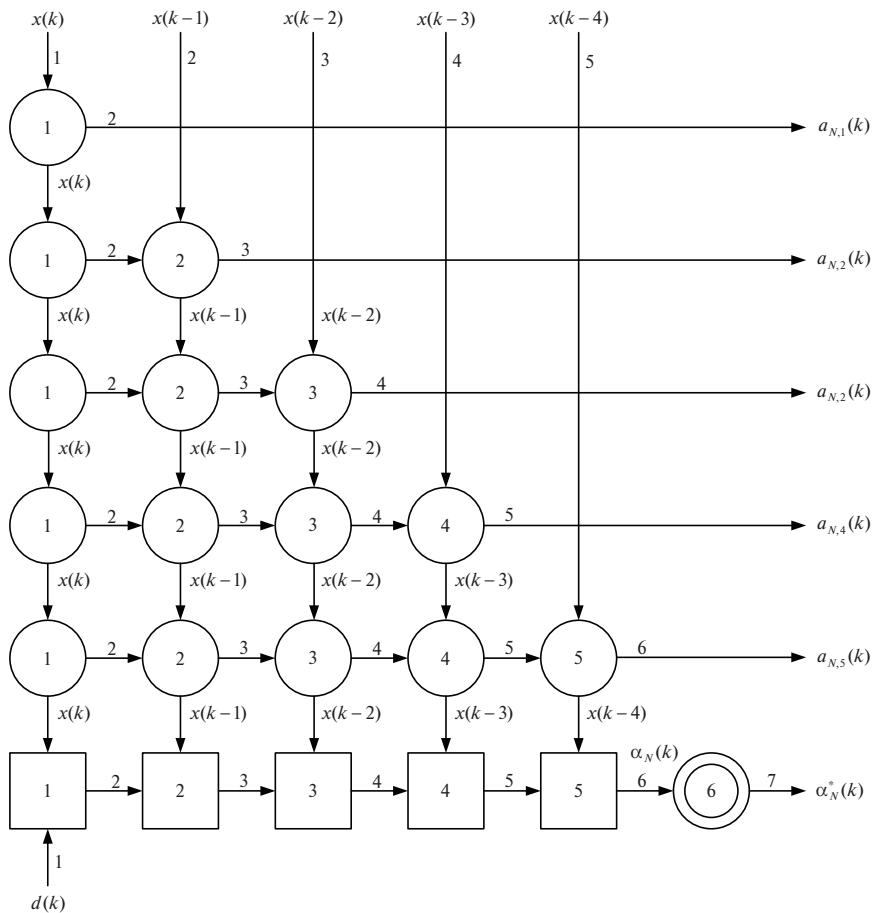


Рис. 7.4. Параллельные вычисления $a_{N,i}(k)$ и $\alpha_N^*(k)$

Таким образом, с учетом времени выполнения операций в структурах (см. рис. 7.3 и рис. 7.4), параллельная реализация алгоритма (см. табл. 7.1) требует $2N+1$ тактов, т. е. максимального числа тактов, необходимого для структуры



(см. рис. 7.3). Кроме того, требуется задержка на один такт за счет задержки вычислений в структуре (см. рис. 7.4).

Для достижения полного эффекта от использования параллельных вычислений требуется буфер для хранения $2N+2$ значений отсчетов сигнала $x(k)$ и $2N+2$ отсчетов сигнала $d(k)$, т.е. входных данных адаптивного фильтра $x(k), x(k-1), \dots, x(k-2N-1)$ и $d(k), d(k-1), \dots, d(k-2N-1)$.

Буфер служит для конвейерной организации вычислений, так как по мере продвижения вычислений сверху вниз (см. рис. 7.3) следующие друг за другом верхние ячейки высвобождаются и могут быть использованы для вычислений при других значениях k . Поскольку при этом длительность всех вычислений равна длительности расчетов в одной (самой сложной) ячейке, то частота дискретизации обрабатываемых сигналов может быть повышена в $2N+2$ раза по сравнению с последовательной реализацией всех вычислений IQRD RLS-алгоритма, а длительность общей задержки выходных сигналов адаптивного фильтра при этом составит $(2N+2)T_s = (2N+2)/F_s$, где T_s — период и F_s — частота дискретизации обрабатываемых сигналов.

В нижнем ряду ячеек 1—5 последовательно выполняются вычисления в соответствии с уравнением (7.1.9) как

$$\alpha_N^{(i)}(k) = \alpha_N^{(i-1)}(k) - h_{N,i}^*(k-1)x_N(k-i+1), \quad (7.130)$$

где $\alpha_N^{(0)}(k) = d(k)$. В ячейке 6 осуществляется комплексное сопряжение переменной $\alpha_N^{(N)}(k) = \alpha_N(k)$.

Преобразования, рассмотренные в настоящей главе для арифметики комплексных чисел, легко применимы и к арифметике действительных чисел. Для этого нужно исключить операции комплексного сопряжения скалярных переменных, а операции эрмитова сопряжения векторов и матриц заменить на операции транспонирования.

7.8. Выводы по главе

В главе 7 сформулирована постановка задачи RLS-адаптивной фильтрации с использованием обратного QR-разложения на базе вращений Гивенса, приводящих к вычислительной процедуре, содержащей операции извлечения квадратного корня. В главе 9 будет описана процедура, позволяющая избавиться от этих операций, при которой оба алгоритма (с корнями и без корней) остаются математически эквивалентными друг другу, равно как и алгоритму из табл. 6.1, при условии их реализации в арифметике с плавающей точкой, одинаковых параметрах адаптивных фильтров и математически эквивалентной инициализации. Структура вычислений в IQRD RLS-алгоритме адаптивной фильтрации позволяет реализовать эти вычисления параллельно.

ГЛАВА 8

РЕКУРСИВНАЯ АДАПТИВНАЯ ФИЛЬТРАЦИЯ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ НА БАЗЕ ПРЯМОГО QR-РАЗЛОЖЕНИЯ

В настоящей главе рассматривается RLS-алгоритм адаптивной фильтрации сигналов, основанный на прямом QR-разложении с использованием вращений Гивенса. Представлены основные шаги получения этого алгоритма, его вычислительная процедура, а также вопросы аппаратной реализации, ориентированной на параллельные вычисления.

8.1. Введение

В главе 7 было рассмотрено решение RLS-задачи адаптивной фильтрации сигналов на основе использовании обратного QR-разложения. Кроме того, в разделе 7.2 были изложены основные идеи прямого QR-разложения. Отмечалось, что при его использовании для вычисления вектора ВК адаптивного фильтра требуется выполнять обратные подстановки, являющиеся источником дополнительной вычислительной сложности адаптивного алгоритма.

В настоящей главе рассматривается RLS-алгоритм [101] на базе прямого QR-разложения, позволяющий вычислять априорные ошибки $\alpha_N(k) = d(k) - y_N(k)$, не вычисляя в явном виде $y_N(k)$, что исключает необходимость вычисления в явном виде и ВК, исключая, таким образом, обратные подстановки. Отсутствие обратных подстановок уменьшает вычислительную сложность алгоритма и, как следствие, делает его более устойчивым.

Ограничение рассматриваемого QRD RLS-алгоритма состоит в том, что он может быть использован лишь в приложениях, где наблюдаемым выходным сигналом является только сигнал ошибки $\alpha_N(k)$. Примером таких приложений являются адаптивные компенсаторы сигналов акустического и электрического эха.



8.2. Рекурсивная форма прямого QR-разложения

Матрица входных сигналов адаптивного фильтра $\mathbf{X}_{Nk}(k)$, приведенная с помощью уравнения (7.13) к верхней треугольной матрице $\tilde{\mathbf{R}}_N(k)$, и вектор $\mathbf{d}_k^*(k)$, преобразованный с помощью уравнения (7.15) к вектору $\tilde{\mathbf{r}}_N(k)$, позволяют с помощью уравнения (7.18) $-\tilde{\mathbf{r}}_N(k) + \tilde{\mathbf{R}}_N(k)\mathbf{h}_N(k) = \mathbf{0}_N$ или (7.19) $-2\tilde{\mathbf{r}}_N(k) + 2\tilde{\mathbf{R}}_N(k)\tilde{\mathbf{h}}_N(k) = \mathbf{0}_N$ вычислять вектор ВК адаптивного фильтра $\mathbf{h}_N(k)$ или $\tilde{\mathbf{h}}_N(k)$, для чего на каждой k -й итерации необходимо выполнять обратные подстановки.

Приведение матрицы данных $\mathbf{X}_{Nk}^H(k)$ (см. раздел 7.2) к треугольному виду $\tilde{\mathbf{R}}_N(k)$ может быть выполнено различными способами, среди которых находятся вращения Гивенса (см. раздел 7.5). Подобно другим RLS-алгоритмам, получение матрицы $\tilde{\mathbf{R}}_N(k)$ можно осуществить рекурсивно во времени, т. е. используя результаты с предыдущей $(k-1)$ -й итерации, выполнять вычисления для k -й итерации. Это осуществляется следующим образом.

Допустим, что на итерации $k-1$ уже существует разложение (7.13), т. е.

$$\mathbf{Q}_{k-1}^H(k-1)\Lambda_{k-1}^{0.5}(k-1)\mathbf{X}_{N(k-1)}^H(k-1) = \begin{bmatrix} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{O}_{(k-1-N)N} \end{bmatrix} \quad (8.1)$$

и требуется выполнить преобразование

$$\mathbf{Q}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k) = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix}, \quad (8.2)$$

используя результат (8.1).

Определим матрицу

$$\bar{\mathbf{Q}}_k^H(k) = \begin{bmatrix} \mathbf{Q}_{k-1}^H(k-1) & \mathbf{0}_{k-1} \\ \mathbf{0}_{k-1}^T & 1 \end{bmatrix}. \quad (8.3)$$

Если умножить матрицу $\Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k)$ слева на матрицу (8.3), то эта операция модифицирует уравнение (8.1), добавляя к матрице в его правой части $(k+1)$ -ю (нижнюю) строку $\mathbf{x}_N^H(k)$, т. е.

$$\begin{aligned} \bar{\mathbf{Q}}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k) &= \bar{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5} \Lambda_{k-1}^{0.5}(k-1) \mathbf{X}_{N(k-1)}^H(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{Q}_{k-1}^H(k-1) \lambda^{0.5} \Lambda_{k-1}^{0.5}(k-1) \mathbf{X}_{N(k-1)}^H(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{O}_{(k-1-N)N} \\ \mathbf{x}_N^H(k) \end{bmatrix}. \end{aligned} \quad (8.4)$$

Чтобы выполнить преобразование (8.2), в уравнении (8.4) необходимо обнулить последнюю строку. Действительно, из уравнений (8.2) и (8.4) следует, что

$$\begin{aligned} \mathbf{Q}_k^H(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) &= \hat{\mathbf{Q}}_k^H(k) \bar{\mathbf{Q}}_k^H(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) = \\ &= \hat{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{O}_{(k-1-N)N} \\ \mathbf{x}_N^H(k) \\ \mathbf{0}_N \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-1-N)N} \\ \mathbf{0}_N \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix}, \end{aligned} \quad (8.5)$$

где матрица $\mathbf{Q}_k^H(k)$ представляет собой произведение двух матриц

$$\mathbf{Q}_k^H(k) = \hat{\mathbf{Q}}_k^H(k) \bar{\mathbf{Q}}_k^H(k). \quad (8.6)$$

Таким образом, приведение матрицы $\Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k)$ к треугольному виду с помощью матрицы $\mathbf{Q}_k^H(k)$, см. уравнение (8.2), на итерации k можно осуществить путем обнуления последней строки в матрице (8.4), т.е. используя результат приведения матрицы $\Lambda_{k-1}^{0.5}(k-1) \mathbf{X}_{N(k-1)}^H(k-1)$ к треугольному виду, полученный на итерации $k-1$. Это обнуление осуществляется с помощью матрицы $\hat{\mathbf{Q}}_k^H(k)$, которая представляет собой произведение матриц, составленных из матриц вращения Гивенса.

Рекурсивная связь между $\tilde{\mathbf{R}}_N(k-1)$ и $\tilde{\mathbf{R}}_N(k)$ в более компактном виде, т.е. при использовании на каждой итерации k матриц с фиксированным числом элементов $(N+1) \times (N+1)$ может быть представлена с помощью уравнения (7.20), где фиксированная по размеру матрица $\mathbf{T}_{N+1}(k)$ представляет собой «сжатую» до размера $(N+1) \times (N+1)$, матрицу $\hat{\mathbf{Q}}_k^H(k)$ с переменным числом элементов $(k+1) \times (k+1)$.

Элементы матриц Гивенса определяются из уравнения (7.20), где

$$\mathbf{T}_{N+1}(k) = \prod_{l=1}^N \mathbf{T}_{N+1}^{(l)}(k), \quad (8.7)$$

а структура матриц $\mathbf{T}_{N+1}^{(l)}(k)$ аналогична структуре матриц $\mathbf{P}_{N+1}^{(l)}(k)$, см. уравнение (7.117). Матрицы $\mathbf{T}_{N+1}^{(l)}(k)$ и $\mathbf{P}_{N+1}^{(l)}(k)$ различаются способом получения косинусов и синусов углов вращения.

Рассмотрим для примера первый из N шагов, приводящих к преобразованию (7.20), для чего используется матрица $\mathbf{T}_{N+1}^{(l)}(k)$. Тогда

$$\begin{aligned} \mathbf{T}_{N+1}^{(1)}(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \tilde{\mathbf{x}}_N^{(0)H}(k) \end{bmatrix} &= \begin{bmatrix} c_{N,1}(k) & 0 & \cdots & 0 & -s_{N,1}^*(k) \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \mathbf{I}_{N-3} & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ s_{N,1}(k) & 0 & \cdots & 0 & c_{N,1}(k) \end{bmatrix} \times \\ &\times \begin{bmatrix} \lambda^{0.5} \tilde{R}_{N,11}(k-1) & \lambda^{0.5} \tilde{R}_{N,12}(k-1) & \cdots & \lambda^{0.5} \tilde{R}_{N,1(N-1)}(k-1) & \lambda^{0.5} \tilde{R}_{N,1N}(k-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda^{0.5} \tilde{R}_{N,(N-1)(N-1)}(k-1) & \lambda^{0.5} \tilde{R}_{N,(N-1)N}(k-1) \\ 0 & 0 & \cdots & 0 & \lambda^{0.5} \tilde{R}_{N,NN}(k-1) \\ \tilde{x}_{N,1}^{(0)*}(k) & \tilde{x}_{N,2}^{(0)*}(k) & \tilde{x}_{N,3}^{(0)*}(k) & \cdots & \tilde{x}_{N,N}^{(0)*}(k) \end{bmatrix} = \end{aligned}$$

$$= \begin{bmatrix} \tilde{R}_{N,11}(k) & \lambda^{0.5} \tilde{R}_{N,12}(k-1) & \cdots & \lambda^{0.5} \tilde{R}_{N,1(N-1)}(k-1) & \lambda^{0.5} \tilde{R}_{N,1N}(k-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & \lambda^{0.5} \tilde{R}_{N,(N-1)(N-1)}(k-1) & \lambda^{0.5} \tilde{R}_{N,(N-1)N}(k-1) \\ 0 & 0 & \cdots & 0 & \tilde{R}_{N,NN}(k-1) \\ 0 & \tilde{x}_{N,2}^{(0)*}(k) & \tilde{x}_{N,3}^{(0)*}(k) & \cdots & \tilde{x}_{N,N}^{(0)*}(k) \end{bmatrix}. \quad (8.8)$$

В уравнении (8.8) вектор $\tilde{\mathbf{x}}_N^{(0)*}(k)$ определяется как

$$\tilde{\mathbf{x}}_N^{(0)*}(k) = [\tilde{x}_{N,1}^{(0)*}(k), \tilde{x}_{N,2}^{(0)*}(k), \dots, \tilde{x}_{N,N}^{(0)*}(k)] = \mathbf{x}_N^H(k) = [x^*(k), x^*(k-1), \dots, x^*(k-N+1)]. \quad (8.9)$$

Здесь, подобно переменным в IQRD RLS-алгоритме, верхний индекс в скобках означает номер преобразования, совершающегося над переменной $\tilde{x}_{N,i}^{(0)*}(k)$ с номером i в векторе. Такое преобразование над $\tilde{x}_{N,1}^{(0)*}(k)$ совершается один раз (после первого раза переменная обнуляется), над $\tilde{x}_{N,2}^{(0)*}(k)$ — два раза (после второго раза переменная обнуляется) и т. д. и над $\tilde{x}_{N,N}^{(0)*}(k)$ — N раз (после N -го раза переменная обнуляется).

Из уравнения (8.9) следует, что

$$c_{N,1}(k) \lambda^{0.5} \tilde{R}_{N,11}(k-1) - s_{N,1}^*(k) \tilde{x}_{N,1}^{(0)*}(k) = \tilde{R}_{N,11}(k) \quad (8.10)$$

и

$$s_{N,1}(k) \lambda^{0.5} \tilde{R}_{N,11}(k-1) + c_{N,1}(k) \tilde{x}_{N,1}^{(0)*}(k) = 0. \quad (8.11)$$

Выполнив аналогичные преобразования всех $i=1, 2, \dots, N$, можно установить, что

$$c_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ii}(k-1) - s_{N,i}^*(k) \tilde{x}_{N,i}^{(i-1)*}(k) = \tilde{R}_{N,ii}(k), \quad (8.12)$$

$$s_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ii}(k-1) + c_{N,i}(k) \tilde{x}_{N,i}^{(i-1)*}(k) = 0. \quad (8.13)$$

Затем из уравнения (8.12) можно определить, что

$$s_{N,i}(k) = -c_{N,i}(k) \tilde{x}_{N,i}^{(i-1)*}(k) \left[\lambda^{0.5} \tilde{R}_{N,ii}(k-1) \right]^{-1} \quad (8.14)$$

и, учитывая, что $c^2 + ss^* = 1$, из уравнения

$$\begin{aligned} c_{N,i}^2(k) + c_{N,i}(k) \tilde{x}_{N,i}^{(i-1)*}(k) \left[\lambda^{0.5} \tilde{R}_{N,ii}(k-1) \right]^{-1} c_{N,i}(k) \tilde{x}_{N,i}^{(i-1)}(k) \left[\lambda^{0.5} \tilde{R}_{N,ii}(k-1) \right]^{-1} = \\ = c_{N,i}^2(k) \left[\lambda^{0.5} \tilde{R}_{N,ii}(k-1) \right]^2 + c_{N,i}^2(k) \left[\tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k) \right] = \\ = c_{N,i}^2(k) \left[\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k) \right] \left[\lambda \tilde{R}_{N,ii}^2(k-1) \right]^{-1} = 1 \end{aligned} \quad (8.15)$$

можно определить, что

$$\begin{aligned} c_{N,i}(k) &= \sqrt{\lambda \tilde{R}_{N,ii}^2(k-1) \left[\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k) \right]^{-1}} = \\ &= \sqrt{\lambda \tilde{R}_{N,ii}^2(k-1)} \sqrt{\left[\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k) \right]^{-1}} = \lambda^{0.5} \tilde{R}_{N,ii}(k-1) \tilde{R}_{N,ii}^{-1}(k), \end{aligned} \quad (8.16)$$

где

$$\tilde{R}_{N,ii}(k) = \sqrt{\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k)}. \quad (8.17)$$

Здесь учтено, что диагональные элементы $\tilde{R}_{N,ii}(k)$ матрицы $\tilde{\mathbf{R}}_N(k)$, подобно диагональным элементам матрицы $\mathbf{R}_N(k)$, являются действительными числами.

Затем, используя (8.16) в уравнении (8.14), можно определить

$$\begin{aligned} s_{N,i}(k) &= -\lambda^{0.5} \tilde{R}_{N,ii}(k-1) \tilde{R}_{N,ii}^{-1}(k) \tilde{x}_{N,i}^{(i-1)*}(k) [\lambda^{0.5} \tilde{R}_{N,ii}(k-1)]^{-1} = \\ &= -\tilde{x}_{N,i}^{(i-1)*}(k) \tilde{R}_{N,ii}^{-1}(k). \end{aligned} \quad (8.18)$$

Таким образом, уравнения (8.16)–(8.18) позволяют вычислять косинус и синус некоторого угла вращения, которые согласно уравнению (8.8) вычисляют элементы i -й строки матрицы $\tilde{\mathbf{R}}_N(k)$, обнуляют i -й элемент вектора $\tilde{\mathbf{x}}_N^{(i-1)\text{H}}(k)$ и модифицируют оставшиеся ненулевые элементы этого вектора как $[0, 0, \dots, 0, 0, \tilde{x}_{N,i+1}^{(i)*}(k), \dots, \tilde{x}_{N,N}^{(i)*}(k)]$, т. е.

$$\tilde{\mathbf{x}}_N^{(i-1)\text{H}}(k) = [0, 0, \dots, 0, \tilde{x}_{N,i}^{(i-1)*}(k), \tilde{x}_{N,i+1}^{(i-1)*}(k), \dots, \tilde{x}_{N,N}^{(i-1)*}(k)], \quad (8.19)$$

а

$$\tilde{\mathbf{x}}_N^{(i)\text{H}}(k) = [0, 0, \dots, 0, 0, \tilde{x}_{N,i+1}^{(i)*}(k), \dots, \tilde{x}_{N,N}^{(i)*}(k)]. \quad (8.20)$$

Эти преобразования для каждого значения i включают вычисления (8.16)–(8.18), а для всех $j=i+1, \dots, N$ – вычисления, аналогичные (8.12) и (8.13), т. е.

$$\tilde{R}_{N,ij}(k) = c_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ij}(k-1) - s_{N,i}^*(k) \tilde{x}_{N,j}^{(i-1)*}(k) \quad (8.21)$$

и

$$\tilde{x}_{N,j}^{(i)*}(k) = s_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ij}(k-1) + c_{N,i}(k) \tilde{x}_{N,j}^{(i-1)*}(k). \quad (8.22)$$

Таким образом, преобразование (8.5) можно осуществлять **либо** как $\mathbf{Q}_k^{\text{H}}(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^{\text{H}}(k)$, применяя на каждой итерации матрицу $\mathbf{Q}_k^{\text{H}}(k)$ размера $k \times k$ к матрице $\Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^{\text{H}}(k)$ размера $k \times N$, **либо** с помощью матрицы

$$\hat{\mathbf{Q}}_k^{\text{H}}(k) = \prod_{l=1}^N \hat{\mathbf{Q}}_k^{(l)\text{H}}(k) = \prod_{l=1}^N \begin{bmatrix} \mathbf{I}_{l-1} & \mathbf{0}_{l-1} & \mathbf{O}_{l-1, k-l-3} & \mathbf{0}_{l-1} \\ \mathbf{0}_{l-1}^{\text{T}} & c_{N,l}(k) & \mathbf{0}_{k-l-3}^{\text{T}} & -s_{N,l}^*(k) \\ \mathbf{O}_{k-l-3, l-1} & \mathbf{0}_{k-l-3} & \mathbf{I}_{k-l-3} & \mathbf{0}_{k-l-3} \\ \mathbf{0}_{l-1}^{\text{T}} & s_{N,l}(k) & \mathbf{0}_{k-l-3}^{\text{T}} & c_{N,l}(k) \end{bmatrix}, \quad (8.23)$$

применяемой к матрице $\begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{O}_{(k-l-N)N} \\ \mathbf{x}_N^{\text{H}}(k) \end{bmatrix}$ размера $k \times N$.



Аналогичным образом выполним преобразование $\mathbf{Q}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{d}_k^*(k)$, используя результат преобразования, полученный на итерации $k-1$ в виде (7.15)

$$\mathbf{Q}_{k-1}^H(k)\Lambda_{k-1}^{0.5}(k)\mathbf{d}_{k-1}^*(k-1) = \begin{bmatrix} \tilde{\mathbf{r}}_N(k-1) \\ \tilde{\mathbf{v}}_{k-1-N}(k-1) \end{bmatrix}. \quad (8.24)$$

Тогда

$$\begin{aligned} \bar{\mathbf{Q}}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) &= \bar{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5}\Lambda_{k-1}^{0.5}(k-1)\mathbf{d}_{k-1}^*(k-1) \\ d^*(k) \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{Q}_{k-1}^H(k-1)\lambda^{0.5}\Lambda_{k-1}^{0.5}(k-1)\mathbf{d}_{k-1}^*(k-1) \\ d^*(k) \end{bmatrix} = \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{r}}_N(k-1) \\ \lambda^{0.5}\tilde{\mathbf{v}}_{k-1-N}(k-1) \\ d^*(k) \end{bmatrix} \end{aligned} \quad (8.25)$$

и, см. уравнение (8.5),

$$\begin{aligned} \mathbf{Q}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) &= \hat{\mathbf{Q}}_k^H(k)\bar{\mathbf{Q}}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) = \\ &= \hat{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{r}}_N(k-1) \\ \lambda^{0.5}\tilde{\mathbf{v}}_{k-1-N}(k-1) \\ d^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \end{bmatrix}. \end{aligned} \quad (8.26)$$

Согласно уравнению (8.23) матрица $\hat{\mathbf{Q}}_k^H(k)$ преобразует только последний элемент вектора $\tilde{\mathbf{v}}_{k-N}(k)$, обозначаемый как

$$\tilde{\alpha}_N^*(k) = v_{k-N}(k), \quad (8.27)$$

т. е.

$$\tilde{\mathbf{v}}_{k-N}(k) = \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{v}}_{k-1-N}^T(k-1), \tilde{\alpha}_N^*(k) \end{bmatrix}^T. \quad (8.28)$$

Чтобы получить выражения для расчета переменных $\tilde{\mathbf{r}}_N(k)$ и $\tilde{\alpha}_N^*(k)$ при использовании $\lambda^{0.5}\tilde{\mathbf{r}}_N(k-1)$ и $d^*(k)$, необходимо воспользоваться соотношением (7.21). Для этого, как и в случае уравнения (8.8), рассмотрим первый шаг преобразования (7.21)

$$\begin{aligned} \mathbf{T}_{N+1}^{(1)}(k) \begin{bmatrix} \lambda^{0.5}\tilde{\mathbf{r}}_N(k-1) \\ d^*(k) \end{bmatrix} &= \\ &= \begin{bmatrix} c_{N,1}(k) & 0 & \dots & 0 & -s_{N,1}^*(k) \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \mathbf{I}_{N-3} & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ s_{N,1}(k) & 0 & \dots & 0 & c_{N,1}(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5}\tilde{r}_{N,1}(k-1) \\ \lambda^{0.5}\tilde{r}_{N,2}(k-1) \\ \vdots \\ \lambda^{0.5}\tilde{r}_{N,N}(k-1) \\ \tilde{\alpha}_N^{(0)*}(k) \end{bmatrix} = \begin{bmatrix} \tilde{r}_{N,1}(k) \\ \lambda^{0.5}\tilde{r}_{N,2}(k-1) \\ \vdots \\ \lambda^{0.5}\tilde{r}_{N,N}(k-1) \\ \tilde{\alpha}_N^{(1)*}(k) \end{bmatrix}, \end{aligned} \quad (8.29)$$

где

$$\tilde{\alpha}_N^{(0)*}(k) = d^*(k), \quad (8.30)$$

$$c_{N,1}(k)\lambda^{0.5}\tilde{r}_{N,1}(k-1)-s_{N,1}^*(k)\tilde{\alpha}_N^{(0)*}(k)=\tilde{r}_{N,1}(k) \quad (8.31)$$

и

$$s_{N,1}(k)\lambda^{0.5}\tilde{r}_{N,1}(k-1)+c_{N,1}(k)\tilde{\alpha}_N^{(0)*}(k)=\tilde{\alpha}_N^{(1)*}(k). \quad (8.32)$$

Здесь верхний индекс в скобках также означает номер преобразования, совершающегося над переменной $\tilde{\alpha}_N^{(0)*}(k)$. Это преобразование совершается N раз на каждой k -й итерации алгоритма.

Таким образом, уравнения (8.30)–(8.32) позволяют модифицировать один раз переменную $\tilde{\alpha}_N^{(0)*}(k)$ и вычислить первый элемент вектора $\tilde{\mathbf{r}}_N(k)$. В общем случае для $i=1, 2, \dots, N$ уравнения (8.31) и (8.32) принимают вид

$$c_{N,i}(k)\lambda^{0.5}\tilde{r}_{N,i}(k-1)-s_{N,i}^*(k)\tilde{\alpha}_N^{(i-1)*}(k)=\tilde{r}_{N,i}(k) \quad (8.33)$$

и

$$s_{N,i}(k)\lambda^{0.5}\tilde{r}_{N,i}(k-1)+c_{N,i}(k)\tilde{\alpha}_N^{(i-1)*}(k)=\tilde{\alpha}_N^{(i)*}(k). \quad (8.34)$$

Следовательно, преобразование (8.26), подобно преобразованию (8.5), также можно выполнить с помощью матрицы (8.23).

8.3. Рекурсивное вычисление ошибок

Так как матрица $\tilde{\mathbf{R}}_N(k)$ является верхней треугольной, то результаты вычислений, получаемые с помощью уравнений (8.5) и (8.26), или, что то же самое с помощью уравнений (7.20) и (7.21), позволяют находить векторы ВК адаптивного фильтра $\mathbf{h}_N(k)$, решая с помощью обратных подстановок уравнение (7.18)

$$\tilde{\mathbf{R}}_N(k)\mathbf{h}_N(k)=\tilde{\mathbf{r}}_N(k). \quad (8.35)$$

Обратные подстановки необходимо выполнять на каждой итерации k , что является источником дополнительной вычислительной сложности такого алгоритма. То есть, в отличие от RLS-алгоритмов на основе MIL (см. главу 6) и обратного QR-разложения (см. главу 7), вычисление ВК в алгоритме на основе прямого QR-разложения не является рекурсивной процедурой, так как согласно уравнению (8.35) ВК не обновляются при переходе от итерации k к итерации $k+1$, а вычисляются каждый раз заново путем использования обратных подстановок.

Однако если в качестве «полезного» (наблюданного) выходного сигнала требуется вычислять только сигнал ошибки $\alpha_N(k)=d(k)-\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$, то в алгоритме на основе прямого QR-разложения вычислять вектор ВК $\mathbf{h}_N(k)$ в явном виде не обязательно.

Действительно, для начала заметим, что переменная $e_N^*(k) = d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k)$ является последним (k -м) элементом вектора

$$\Lambda_k^{0.5}(k)\mathbf{e}_k^*(k) = \Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) - \Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k)\mathbf{h}_N(k), \quad (8.36)$$

см. уравнение (7.7). Из уравнений (8.5) и (8.26) следует, что

$$\mathbf{Q}_k^H(k) \begin{bmatrix} \Lambda_k^{0.5}(k)\mathbf{e}_k^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N} \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix} \mathbf{h}_N(k), \quad (8.37)$$

а из уравнения (8.37) — что матрицу $\mathbf{Q}_k^H(k)$ можно представить как

$$\mathbf{Q}_k^H(k) = \begin{bmatrix} \mathbf{W}_{Nk}(k) \\ \mathbf{S}_{(k-N)k}(k) \end{bmatrix}, \quad (8.38)$$

где $\mathbf{W}_{Nk}(k)$ — прямоугольная матрица размером $N \times k$, а $\mathbf{S}_{(k-N)k}(k)$ — прямоугольная матрица размером $(k-N) \times k$.

Затем, используя (8.38), преобразуем уравнение (8.36). Для этого, умножив уравнение (8.5) справа на матрицу $\mathbf{Q}_k(k) = [\mathbf{W}_{Nk}^H(k), \mathbf{S}_{(k-N)k}^H(k)]$, получим, что

$$\begin{aligned} \mathbf{Q}_k(k)\mathbf{Q}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k) &= \\ &= \Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k) = \mathbf{Q}_k(k) \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{W}_{Nk}^H(k), \mathbf{S}_{(k-N)k}^H(k) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N)N} \end{bmatrix} = \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{R}}_N(k). \end{aligned} \quad (8.39)$$

Далее, используя выражения (8.26) и (8.38), получим, что

$$\begin{aligned} \mathbf{Q}_k(k)\mathbf{Q}_k^H(k)\Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) &= \Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) = \mathbf{Q}_k(k) \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N} \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{W}_{Nk}^H(k), \mathbf{S}_{(k-N)k}^H(k) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N} \end{bmatrix} = \\ &= \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{r}}_N(k) + \mathbf{S}_{(k-N)k}^H(k)\tilde{\mathbf{v}}_{k-N}. \end{aligned} \quad (8.40)$$

Если теперь в уравнении (8.36) использовать выражения (8.39) и (8.40)

$$\begin{aligned} \Lambda_k^{0.5}(k)\mathbf{e}_k^*(k) &= \Lambda_k^{0.5}(k)\mathbf{d}_k^*(k) - \Lambda_k^{0.5}(k)\mathbf{X}_{Nk}^H(k)\mathbf{h}_N(k) = \\ &= \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{r}}_N(k) + \mathbf{S}_{(k-N)k}^H(k)\tilde{\mathbf{v}}_{k-N} - \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{R}}_N(k)\mathbf{h}_N(k) = \\ &= \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{r}}_N(k) + \mathbf{S}_{(k-N)k}^H(k)\tilde{\mathbf{v}}_{k-N} - \mathbf{W}_{Nk}^H(k)\tilde{\mathbf{r}}_N(k) = \mathbf{S}_{(k-N)k}^H(k)\tilde{\mathbf{v}}_{k-N}, \end{aligned} \quad (8.41)$$

то можно заметить, что уравнение (8.41) не содержит вектора ВК $\mathbf{h}_N(k)$ в явном виде. Кроме того, как уже упоминалось, последний элемент вектора (8.41) равен $e_N^*(k)$.

Ниже показано, как можно осуществить преобразование (8.41) и как соотносятся между собой переменные $e_N(k)$ и $\alpha_N(k)$.

Значение переменной $e_N^*(k)$ можно определить, исходя из структуры унитарных матриц, формируемых в процессе использования вращений Гивенса.

Матрица $\hat{\mathbf{Q}}_k^H(k)$ формируется с помощью уравнения (8.23). Чтобы понять ее структуру, выполним перемножение двух из N матриц $\hat{\mathbf{Q}}_k^{H(2)}(k)$ и $\hat{\mathbf{Q}}_k^{H(1)}(k)$ и по полученному результату сделаем вывод о том, какова структура матрицы $\hat{\mathbf{Q}}_k^H(k)$.

Произведение $\hat{\mathbf{Q}}_k^{H(2)}(k)\hat{\mathbf{Q}}_k^{H(1)}(k)$ формируется как

$$\begin{aligned}
 \hat{\mathbf{Q}}_k^{H(2)}(k)\hat{\mathbf{Q}}_k^{H(1)}(k) &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & c_{k,2}(k) & 0 & \cdots & 0 & -s_{k,2}^*(k) \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \mathbf{I}_{N-4} & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & s_{k,2}(k) & 0 & \cdots & 0 & c_{k,2}(k) \end{bmatrix} \times \\
 &\times \begin{bmatrix} c_{k,1}(k) & 0 & 0 & \cdots & 0 & -s_{k,1}^*(k) \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \mathbf{I}_{k-4} & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ s_{k,1}(k) & 0 & 0 & \cdots & 0 & c_{k,1}(k) \end{bmatrix} = \\
 &= \begin{bmatrix} c_{k,1}(k) & 0 & 0 & \cdots & 0 & -s_{k,1}^*(k) \\ 0 & c_{k,2}(k) & 0 & \cdots & 0 & -s_{k,2}^*(k)c_{k,1}(k) \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \mathbf{I}_{k-4} & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ c_{k,2}(k)s_{k,1}(k) & s_{k,2}(k) & 0 & \cdots & 0 & c_{k,2}(k)c_{k,1}(k) \end{bmatrix} = \\
 &= \begin{bmatrix} \mathbf{A}_2(k) & \mathbf{O}_{2(k-3)} & \mathbf{a}_2(k) \\ \mathbf{O}_{(k-3)2} & \mathbf{I}_{k-3} & \mathbf{0}_{k-3} \\ \mathbf{b}_2^H(k) & \mathbf{0}_{k-3}^H & \tilde{\phi}_2(k) \end{bmatrix}.
 \end{aligned} \tag{8.42}$$

Можно убедиться, что, продолжив умножение (8.42) на матрицы $\hat{\mathbf{Q}}_k^{H(3)}(k)$, ..., $\hat{\mathbf{Q}}_k^{H(N)}(k)$, каждый раз будем получать матрицу со структурой, аналогичной структуре матрицы (8.42). Следовательно, уравнение (8.23) можно представить как

$$\hat{\mathbf{Q}}_k^H(k) = \prod_{l=1}^N \hat{\mathbf{Q}}_k^{(l)H}(k) = \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{O}_{N(k-l-N)} & \mathbf{a}_N(k) \\ \mathbf{O}_{(k-l-N)N} & \mathbf{I}_{k-l-N} & \mathbf{0}_{k-l-N} \\ \mathbf{b}_N^H(k) & \mathbf{0}_{k-l-N}^H & \tilde{\phi}_N(k) \end{bmatrix}, \tag{8.43}$$

где, согласно матрице (8.42), $\tilde{\varphi}(k)$ — действительная скалярная величина, не превышающая единицу по абсолютной величине, так как

$$\tilde{\varphi}_N(k) = c_{k,N}(k) \cdots c_{k,2}(k) c_{k,1}(k) = \prod_{l=1}^N c_{k,l}(k). \quad (8.44)$$

Из уравнений (8.3), (8.6), (8.38) и (8.43) следует, что

$$\begin{aligned} \mathbf{Q}_k^H(k) &= \hat{\mathbf{Q}}_k^H(k) \bar{\mathbf{Q}}_k^H(k) = \\ &= \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{O}_{N(k-1-N)} & \mathbf{a}_N(k) \\ \mathbf{O}_{(k-1-N)N} & \mathbf{I}_{k-1-N} & \mathbf{0}_{k-1-N} \\ \mathbf{b}_N^H(k) & \mathbf{0}_{k-1-N}^H & \tilde{\varphi}_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{k-1}^H(k-1) & \mathbf{0}_{k-1} \\ \mathbf{0}_{k-1}^T & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{O}_{N(k-1-N)} & \mathbf{a}_N(k) \\ \mathbf{O}_{(k-1-N)N} & \mathbf{I}_{k-1-N} & \mathbf{0}_{k-1-N} \\ \mathbf{b}_N^H(k) & \mathbf{0}_{k-1-N}^H & \tilde{\varphi}_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{W}_{N(k-1)}(k-1) & \mathbf{0}_N \\ \mathbf{S}_{(k-1-N)(k-1)}(k-1) & \mathbf{0}_{k-1-N} \\ \mathbf{0}_{k-1}^T & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{A}_N(k)\mathbf{W}_{N(k-1)}(k-1) & \mathbf{a}_N(k) \\ \mathbf{S}_{(k-1-N)(k-1)}(k-1) & \mathbf{0}_{k-1-N} \\ \mathbf{b}_N^H(k)\mathbf{W}_{N(k-1)}(k-1) & \tilde{\varphi}_N(k) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{Nk}(k) \\ \mathbf{S}_{(k-N)k}(k) \end{bmatrix}, \end{aligned} \quad (8.45)$$

а из уравнений (8.26) и (8.43) — что

$$\begin{aligned} \hat{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \\ \lambda^{0.5} \tilde{\mathbf{v}}_{k-1-N}(k-1) \\ d^*(k) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{O}_{N(k-1-N)} & \mathbf{a}_N(k) \\ \mathbf{O}_{(k-1-N)N} & \mathbf{I}_{k-1-N} & \mathbf{0}_{k-1-N} \\ \mathbf{b}_N^H(k) & \mathbf{0}_{k-1-N}^H & \tilde{\varphi}_N(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \\ \lambda^{0.5} \tilde{\mathbf{v}}_{k-1-N}(k-1) \\ d^*(k) \end{bmatrix} = \\ &= \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-N}(k) \\ \tilde{\alpha}_N^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\mathbf{v}}_{k-1-N}(k) \\ \tilde{\alpha}_N^*(k) \end{bmatrix} = \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{v}}_{k-1-N}(k-1) \\ \tilde{\alpha}_N^*(k) \end{bmatrix}, \end{aligned} \quad (8.46)$$

откуда

$$\mathbf{b}_N^H(k) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) + \tilde{\varphi}_N(k) d^*(k) = \tilde{\alpha}_N^*(k) \quad (8.47)$$

и

$$\tilde{\mathbf{v}}_{k-N}(k) = [\lambda^{0.5} \tilde{\mathbf{v}}_{k-1-N}^T(k-1), \tilde{\alpha}_N^*(k)]^T. \quad (8.48)$$

Из уравнения (8.45) также следует, что

$$\mathbf{S}_{(k-N)k}(k) = \begin{bmatrix} \mathbf{S}_{(k-1-N)(k-1)}(k-1) & \mathbf{0}_{k-1-N} \\ \mathbf{b}_N^H(k)\mathbf{W}_{N(k-1)}(k-1) & \tilde{\varphi}_N(k) \end{bmatrix} \quad (8.49)$$

и

$$\mathbf{S}_{(k-N)k}^H(k) = \begin{bmatrix} \mathbf{S}_{(k-1-N)(k-1)}^H(k-1) & \mathbf{W}_{N(k-1)}^H(k-1)\mathbf{b}_N(k) \\ \mathbf{0}_{k-1-N}^T & \tilde{\varphi}_N(k) \end{bmatrix}. \quad (8.50)$$

Тогда, используя уравнения (8.41) (8.48) и (8.50), получим, что

$$\begin{aligned}
 \Lambda_k^{0.5}(k) \mathbf{e}_k^*(k) &= \mathbf{S}_{(k-N)k}^H(k) \tilde{\mathbf{v}}_{k-N} = \\
 &= \begin{bmatrix} \mathbf{S}_{(k-l-N)(k-l)}^H(k-1) & \mathbf{W}_{N(k-l)}^H(k-1) \mathbf{b}_N(k) \\ \mathbf{0}_{k-l-N}^T & \tilde{\varphi}_N(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{v}}_{k-l-N}(k-1) \\ \tilde{\alpha}_N^*(k) \end{bmatrix} = \\
 &= \begin{bmatrix} \mathbf{S}_{(k-l-N)(k-l)}^H(k-1) \lambda^{0.5} \tilde{\mathbf{v}}_{k-l-N}(k-1) \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{W}_{N(k-l)}^H(k-1) \mathbf{b}_N(k) \tilde{\alpha}_N^*(k) \\ \tilde{\varphi}_N(k) \tilde{\alpha}_N^*(k) \end{bmatrix} = \\
 &= \lambda^{0.5} \begin{bmatrix} \mathbf{S}_{(k-l-N)(k-l)}^H(k-1) \tilde{\mathbf{v}}_{k-l-N}(k-1) \\ 0 \end{bmatrix} + \tilde{\alpha}_N^*(k) \begin{bmatrix} \mathbf{W}_{N(k-l)}^H(k-1) \mathbf{b}_N(k) \\ \tilde{\varphi}_N(k) \end{bmatrix}.
 \end{aligned} \tag{8.51}$$

Из (8.51) следует, что последний (k -й) элемент полученного вектора равен $\tilde{\alpha}_N^*(k) \tilde{\varphi}_N(k)$, т. е.

$$e_N(k) = \tilde{\alpha}_N^*(k) \tilde{\varphi}_N(k). \tag{8.52}$$

Далее установим взаимосвязь между переменными $e_N(k)$ и $\alpha_N(k)$, так как «обычно» в адаптивном фильтре в качестве «требуемого выходного сигнала» используется вторая переменная, потому что «в начале» k -й итерации можно вычислять только $y_N(k) = \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$ и $\alpha_N(k) = d(k) - y_N(k)$, так как значение $\mathbf{h}_N^H(k)$ может «появиться» (быть рассчитано) только «в конце» k -й итерации.

Для этого, используя (8.43), уравнение (8.5) можно представить как

$$\hat{\mathbf{Q}}_k^H(k) \begin{bmatrix} \lambda^{0.5} \mathbf{R}_N(k-1) \\ \mathbf{O}_{(k-l-N)N} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_N(k) & \mathbf{O}_{N(k-l-N)} & \mathbf{a}_N(k) \\ \mathbf{O}_{(k-l-N)N} & \mathbf{I}_{k-l-N} & \mathbf{0}_{k-l-N} \\ \mathbf{x}_N^H(k) & \mathbf{b}_N^H(k) & \mathbf{0}_{k-l-N}^T \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \mathbf{R}_N(k-1) \\ \mathbf{O}_{(k-l-N)N} \\ \tilde{\mathbf{R}}_N(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-l-N)N} \\ \mathbf{0}_N^T \end{bmatrix}. \tag{8.53}$$

Тогда из уравнения (8.53) следует, что

$$\mathbf{b}_N^H(k) \lambda^{0.5} \mathbf{R}_N(k-1) + \tilde{\varphi}_N(k) \mathbf{x}_N^H(k) = \mathbf{0}_N^T, \tag{8.54}$$

откуда

$$\mathbf{x}_N^H(k) = -\tilde{\varphi}_N^{-1}(k) \mathbf{b}_N^H(k) \lambda^{0.5} \mathbf{R}_N(k-1). \tag{8.55}$$

Затем, с учетом уравнений (8.35) и (8.47), используем выражение (8.55) в уравнении

$$\begin{aligned}
 \alpha_N^*(k) &= d^*(k) - \mathbf{x}_N^H(k) \mathbf{h}_N(k-1) = \\
 &= d^*(k) + \tilde{\varphi}_N^{-1}(k) \mathbf{b}_N^H(k) \lambda^{0.5} \mathbf{R}_N(k-1) \mathbf{h}_N(k-1) = \\
 &= d^*(k) + \tilde{\varphi}_N^{-1}(k) \mathbf{b}_N^H(k) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) = \\
 &= \tilde{\varphi}_N^{-1}(k) [\tilde{\varphi}_N(k) d^*(k) + \mathbf{b}_N^H(k) \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1)] = \tilde{\alpha}_N^*(k) \tilde{\varphi}_N^{-1}(k).
 \end{aligned} \tag{8.56}$$

Так как согласно уравнению (8.52)

$$\tilde{\alpha}_N(k) = e_N(k) \tilde{\varphi}_N^{-1}(k), \tag{8.57}$$

то

$$\alpha_N(k) = \tilde{\alpha}_N(k) \tilde{\varphi}_N^{-1}(k) = e_N(k) \tilde{\varphi}_N^{-1}(k) \tilde{\varphi}_N^{-1}(k) = e_N(k) \tilde{\varphi}_N^{-2}(k). \quad (8.58)$$

Таким образом, уравнения (8.16)–(8.18), выполняемые $i=1, 2, \dots, N$ раз на каждой итерации по времени k ; уравнения (8.21) и (8.22), выполняемые $j=i+1, \dots, N$ раз при каждом i ; уравнения (8.33), (8.34) и уравнение (8.44) в виде $\tilde{\varphi}_N^{(i)}(k) = c_{N,i}(k) \tilde{\varphi}_N^{(i-1)}(k)$ при $\tilde{\varphi}_N^{(0)}(k) = 1$, выполняемые $i=1, 2, \dots, N$ раз на каждой итерации по времени k ; уравнение (8.58) в виде $\alpha_N(k) = \tilde{\alpha}_N(k) \tilde{\varphi}_N^{-1}(k)$ при $\tilde{\varphi}_N(k) = \tilde{\varphi}_N^{(N)}(k)$ представляют собой RLS-алгоритм на базе прямого QR-разложения, в котором не используются обратные подстановки, но в качестве единственного «требуемого выходного сигнала» могут использоваться только значения $\alpha_N(k)$. Вычислительная процедура такого алгоритма приведена в табл. 8.1.

Таблица 8.1. RLS-алгоритм на основе прямого QR-разложения с использованием вращений Гивенса (с операциями извлечения квадратного корня)

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\tilde{\mathbf{R}}_N(0) = \delta \Lambda_N^{-0.5}$, $\mathbf{h}_N(0) = \mathbf{0}_N$, $\tilde{\mathbf{r}}_N(0) = \mathbf{0}_N$, $\Lambda_N = \text{diag}\{1, \lambda, \dots, \lambda^{N-1}\}$	(8.1.0)
For $k = 1, 2, \dots, K$	
$\tilde{\mathbf{x}}_N^{(0)\text{H}}(k) = \mathbf{x}_N^{\text{H}}(k)$, $\tilde{\alpha}_N^{*(0)}(k) = d^*(k)$, $\tilde{\varphi}_N^{(0)}(k) = 1$	(8.1.1)
For $i = 1, 2, \dots, N$	
$\tilde{\mathbf{R}}_{N,ii}(k) = \sqrt{\lambda \tilde{\mathbf{R}}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k)}$	(8.1.2)
$s_{N,i}(k) = -\tilde{x}_{N,i}^{(i-1)*}(k) \tilde{\mathbf{R}}_{N,ii}^{-1}(k)$	(8.1.3)
$c_{N,i}(k) = \lambda^{0.5} \tilde{\mathbf{R}}_{N,ii}(k-1) \tilde{\mathbf{R}}_{N,ii}^{-1}(k)$	(8.1.4)
For $j = i+1, \dots, N$	
$\tilde{\mathbf{R}}_{N,ij}(k) = c_{N,i}(k) \lambda^{0.5} \tilde{\mathbf{R}}_{N,ij}(k-1) - s_{N,i}^*(k) \tilde{x}_{N,j}^{(i-1)*}(k)$	(8.1.5)
$\tilde{x}_{N,j}^{(i)}(k) = s_{N,i}(k) \lambda^{0.5} \tilde{\mathbf{R}}_{N,ij}(k-1) + c_{N,i}(k) \tilde{x}_{N,j}^{(i-1)*}(k)$	(8.1.6)
End for j	
$\tilde{\mathbf{r}}_{N,i}(k) = c_{N,i}(k) \lambda^{0.5} \tilde{\mathbf{r}}_{N,i}(k-1) - s_{N,i}^*(k) \tilde{\alpha}_N^{(i-1)*}(k)$	(8.1.7)
$\tilde{\alpha}_N^{(i)*}(k) = s_{N,i}(k) \lambda^{0.5} \tilde{\mathbf{r}}_{N,i}(k-1) + c_{N,i}(k) \tilde{\alpha}_N^{(i-1)*}(k)$	(8.1.8)
$\tilde{\varphi}_N^{(i)}(k) = c_{N,i}(k) \tilde{\varphi}_N^{(i-1)}(k)$	(8.1.9)
End for i	
$\alpha_N(k) = \tilde{\alpha}_N^{(N)}(k) \left[\tilde{\varphi}_N^{(N)}(k) \right]^{-1}$	(8.1.10)
End for k	

Нижние индексы i и j в обозначениях переменных в табл. 8.1 указывают на номера используемых в вычислениях элементов векторов и матриц.

Инициализация разложения Холецкого в табл. 8.1 как $\tilde{\mathbf{R}}_N(0)=\delta \Lambda_N^{-0.5}$, где

$$\Lambda_N^{-0.5} = \sqrt{\Lambda_N^{-1}} = \sqrt{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & & \lambda^{N-2} & 0 \\ 0 & 0 & \cdots & 0 & \lambda^{N-1} \end{bmatrix}^{-1}} =$$

$$= \sqrt{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda^{-1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & & \lambda^{-(N-2)} & 0 \\ 0 & 0 & \cdots & 0 & \lambda^{-(N-1)} \end{bmatrix}} = \sqrt{\text{diag}\{1, \lambda^{-1}, \dots, \lambda^{-(N-1)}\}}, \quad (8.59)$$

математически эквивалентна инициализации алгоритма (см. табл. 6.1) как $\mathbf{R}_N(0)=\delta^2 \Lambda_N^{-1}$, эквивалентна инициализации алгоритма (см. табл. 7.1) как $\tilde{\mathbf{R}}_N^{-H}(0)=\delta^{-1} \Lambda_N^{0.5}$ и эквивалентна другим RLS-алгоритмам в случае экспоненциального взвешивания обрабатываемых сигналов с помощью параметра λ , см. раздел 14.8. Данная инициализация следует из уравнения (7.48) $\mathbf{R}_N(k)=\tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k)$ и из уравнений (7.13), (7.14), (6.115), так как

$$\begin{aligned} \mathbf{R}_N(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \mathbf{X}_{Nk}(k) \Lambda_k(k) \mathbf{X}_{Nk}^H(k) = \\ &= \mathbf{X}_{Nk}(k) \Lambda_k^{0.5}(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) = \mathbf{X}_{Nk}(k) \Lambda_k^{0.5}(k) \mathbf{Q}_k(k) \mathbf{Q}_k^H(k) \Lambda_k^{0.5}(k) \mathbf{X}_{Nk}^H(k) = \\ &= [\tilde{\mathbf{R}}_N^H(k), \mathbf{O}_{(k-N),N}^T] \begin{bmatrix} \tilde{\mathbf{R}}_N(k) \\ \mathbf{O}_{(k-N),N} \end{bmatrix} = \tilde{\mathbf{R}}_N^H(k) \tilde{\mathbf{R}}_N(k). \end{aligned} \quad (8.60)$$

Поскольку согласно уравнению (8.44) $\tilde{\varphi}_N(k)=c_{k,N}(k) \cdots c_{k,2}(k) c_{k,1}(k)=\prod_{l=1}^N c_{k,l}(k)$

и уравнению (8.1.1) $\tilde{\varphi}_N^{(i)}(k)=c_{N,i}(k) \tilde{\varphi}_N^{(i-1)}(k)$ переменная $\tilde{\varphi}_N(k)=\tilde{\varphi}_N^{(N)}(k)$ вычисляется в виде рекурсивного произведения, то вполне логичным является то, что начальное значение этого произведения $\tilde{\varphi}_N^{(0)}(k)=1$.

Кроме того, так как согласно уравнению (7.21)

$$\mathbf{T}_{N+1}(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{r}}_N(k-1) \\ d^*(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_N(k) \\ \tilde{\alpha}_N^*(k) \end{bmatrix},$$

то при рекурсивном вычислении переменной $\tilde{\alpha}_N^*(k)$ ее начальное значение определяется как $\tilde{\alpha}_N^{*(0)}(k)=d^*(k)$, см. уравнения (8.1.1) и (8.30).

Также, поскольку согласно уравнению (8.35) $\tilde{\mathbf{R}}_N(0)\mathbf{h}_N(0)=\tilde{\mathbf{r}}_N(0)$, то при $\tilde{\mathbf{R}}_N(0) \neq \mathbf{0}_N$ и $\mathbf{h}_N(0)=\mathbf{0}_N$ начальное значение вектора $\tilde{\mathbf{r}}_N(0)$ определяется как $\tilde{\mathbf{r}}_N(0)=\mathbf{0}_N$.

Оценка вычислительной сложности алгоритма (см. табл. 8.1) равна $O(N^2)$ арифметическим операциям. Она состоит из $2.5N^2 + 7.5N$ операций умножения,

$N^2 + 2N$ операций сложения, $N + 1$ операций деления и N операций извлечения квадратного корня.

8.4. Параллельная реализация QRD RLS-алгоритма

Треугольная структура матрицы $\tilde{\mathbf{R}}_N(k)$ позволяет для реализации алгоритма (см. табл. 8.1) использовать систолические вычислители. Архитектура основных вычислений алгоритма с помощью таких вычислителей приведена в [101] и на рис. 8.1. Основные идеи такой реализации совпадают с идеями, рассмотренными в разделе 7.7.

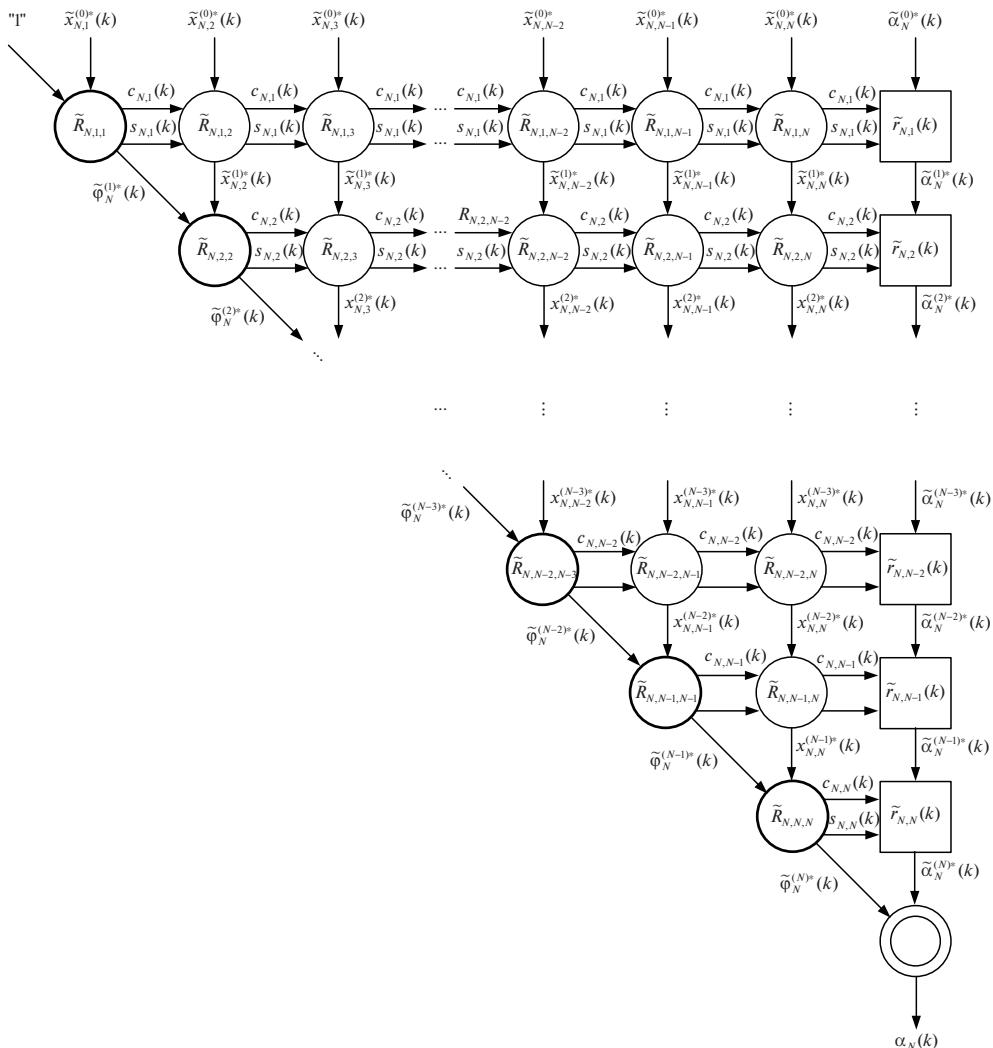


Рис. 8.1. Аппаратная реализация QR RLS-алгоритма

Здесь крайние левые (диагональные) ячейки (кроме самой нижней) выполняют все операции, кроме (8.1.5), (8.1.6) и (8.1.10). Ячейки справа от диагональных выполняют операции (8.1.5), (8.1.6), а нижняя диагональная ячейка выполняет операцию (8.1.10).

На рис. 8.2 при $N = 5$ показан пример того, как в архитектуре (см. рис. 8.1) могут быть использованы параллельные вычисления. Здесь одинаковыми цифрами обозначены информационные потоки и соответствующие им параллельные вычисления, которые могут быть выполнены одновременно. Проанализировав эту архитектуру при разных значениях N , можно легко заметить, что для полного вычисления при одном значении k требуется $2N + 2$ тактов (шагов). Длительность одного такта определяется вычислениями в ячейках с максимальной длительностью по времени, т. е. содержащих наибольшее число арифметических операций, требующих последовательного выполнения в пределах ячейки.

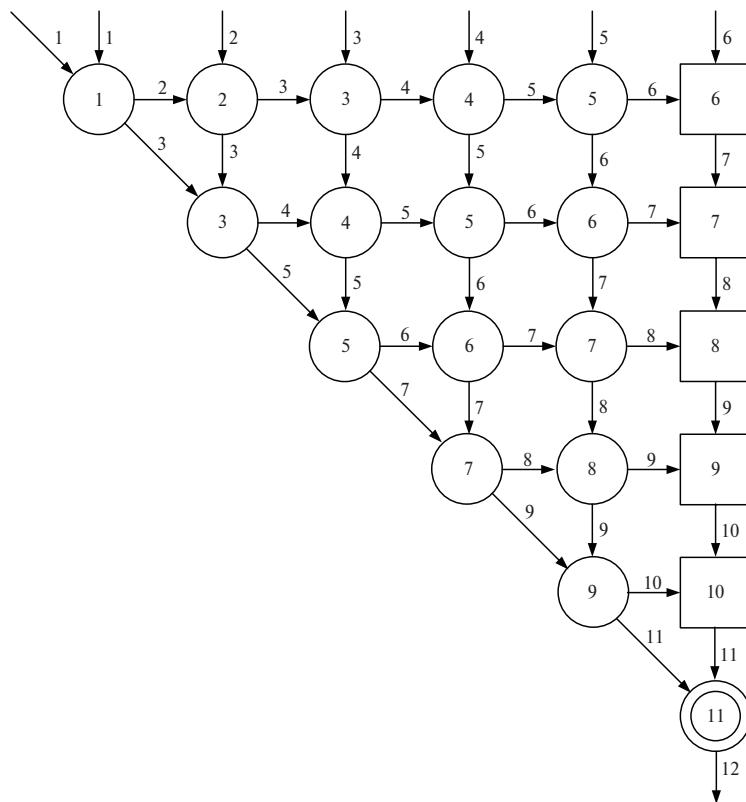


Рис. 8.2. Параллельные вычисления в QR RLS-алгоритме

Для достижения полного эффекта от использования параллельных вычислений требуется буфер для хранения $2N + 2$ значений отсчетов сигнала



$x(k)$ и $2N + 2$ отсчетов сигнала $d(k)$, т. е. входных данных адаптивного фильтра $x(k)$, $x(k-1)$, ..., $x(k-2N-1)$ и $d(k)$, $d(k-1)$, ..., $d(k-2N-1)$.

Буфер нужен для выполнения конвейерной организации вычислений, так как по мере продвижения вычислений сверху вниз (см. рис. 8.2) следующие друг за другом верхние ячейки высвобождаются и могут быть использованы для вычислений при других значениях k .

Так как при этом длительность всех вычислений равна длительности расчетов в одной (самой сложной) ячейке, то частота дискретизации обрабатываемых сигналов может быть повышена в $2N + 2$ раз по сравнению с последовательной реализацией вычислений, а длительность общей задержки при этом составит $(2N + 2)T_s = (2N + 2)/F_s$, где T_s — период и F_s — частота дискретизации сигналов.

Преобразования, рассмотренные в настоящей главе для арифметики комплексных чисел, легко приметить к арифметике действительных чисел. Для этого нужно исключить операции комплексного сопряжения скалярных переменных и операции эрмитова сопряжения векторов и матриц заменить на операции транспонирования.

8.5. Выводы по главе

В главе 8 сформулирована постановка задачи RLS-адаптивной фильтрации сигналов с использованием прямого QR-разложения на базе вращений Гивенса, приводящих к вычислительной процедуре, содержащей операции извлечения квадратного корня. Структура вычислений в алгоритме адаптивной фильтрации на базе прямого QR-разложения позволяет реализовать эти вычисления параллельно.

ГЛАВА 9

УМЕНЬШЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ РЕКУРСИВНЫХ АЛГОРИТМОВ АДАПТИВНОЙ ФИЛЬТРАЦИИ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ

В настоящей главе рассматривается RLS-алгоритм адаптивной фильтрации сигналов, основанный на обратном QR-разложении с использованием преобразования Хаусхолдера. Алгоритм характеризуется наличием только одной операции извлечения квадратного корня и одной операции деления. Также рассматривается способ исключения операций извлечения квадратного корня в RLS-алгоритмах на основе прямого и обратного QR-разложений с использованием вращений Гивенса.

9.1. Введение

Известно, что RLS-алгоритмы на основе прямого и обратного QR-разложений характеризуются хорошей устойчивостью, что объясняет их популярность в приложениях адаптивной фильтрации. Однако такие алгоритмы, как правило, содержат операции извлечения квадратного корня и деления, см. примеры этих операций в табл. 7.1 и табл. 8.1. Это уравнения (7.1.3), (8.1.2) и (7.1.4), (8.1.3), (8.1.10). Число операций (7.1.3), (8.1.2) и (7.1.4), (8.1.3) на каждой итерации по времени k равно числу ВК адаптивного фильтра N .

Операции извлечения квадратного корня и деления в большинстве современных ЦСП не имеют аппаратной поддержки и реализуются с помощью различных функциональных преобразований на базе таблично-аналитических методов [110, 111]. Это означает, что операции извлечения квадратного корня или деления реализуются программным способом и требуют некоторого числа простейших арифметических операций умножения и сложения.

Так, например, в прикладных библиотеках сигнальных контроллеров отечественной серии «Мультикор» [44] начальные приближения для операций из-



влечения квадратного корня и деления хранятся в виде таблиц, а конечные результаты итерационно уточняются с помощью метода Ньютона — Рафсона [112]. Для реализации одной операции извлечения квадратного корня в арифметике с плавающей точкой используется 13 умножений и 3 сложения, а для реализации операции деления — 7 умножений и 3 сложения.

Таким образом, казалось бы, линейная зависимость числа операций извлечения квадратного корня и деления от числа ВК N в RLS-алгоритмах на основе QR-разложения приводит к существенному усложнению этих алгоритмов в терминах числа операций сложения и умножения.

Исключение операций извлечения квадратного корня и деления позволяет уменьшить вычислительную сложность алгоритмов адаптивной фильтрации, что в конечном итоге, при прочих равных условиях позволяет уменьшить длительность выполнения одной итерации алгоритмов, а значит уменьшить период дискретизации обрабатываемых сигналов, ограничиваемый этой длительностью, т. е. повысить частоту дискретизации.

Далее будут рассмотрены два способа исключения сложных арифметических операций в QRD RLS-алгоритмах.

Первый способ основан на применении преобразования Хаусхолдера [104, 105, 113]. В вычислениях в RLS-алгоритме на основе этого способа используется не корреляционная матрица, а квадратный корень из обратной корреляционной матрицы, имеется только одна операция извлечения квадратного корня и две операции деления на одну итерацию. То есть, число этих операций не зависит от числа ВК N адаптивного фильтра.

Второй способ [101, 106, 114] базируется на масштабировании переменных, участвующих в вычислениях в алгоритмах на основе прямого и обратного QR-разложений. Данный способ позволяет исключить операции извлечения квадратного корня, присутствующие в этих алгоритмах.

9.2. Преобразование Хаусхолдера

В отличие от матриц вращений Гивенса, которые за один раз образуют («вычисляют») или исключают («обнуляют») только один элемент вектора (см. главы 7 и 8), преобразование Хаусхолдера может быть использовано для исключения нескольких элементов вектора за один раз. Это преобразование рассматривается ниже.

Пусть $\mathbf{e}_{N,i} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$ — единичный направляющий вектор в N -мерном евклидовом пространстве, где «1» — i -й ненулевой (единичный) элемент вектора, i может принимать любое из значений $1, 2, \dots, N$, $\mathbf{x}_N = [x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_N]^T$ — N -мерный действительный вектор. Предположим, что требуется выполнить преобразование

$$\mathbf{P}_N \mathbf{x}_N = \alpha \mathbf{e}_{N,i}, \quad (9.1)$$

т. е. некоторым образом спроецировать \mathbf{x}_N на направляющий вектор $\mathbf{e}_{N,i}$. Здесь \mathbf{P}_N — ортонормированная проекционная матрица и α — скаляр, которые для выполнения операции (9.1) нужно определить.

Значение α можно определить из соотношения

$$(\mathbf{P}_N \mathbf{x}_N)^T \mathbf{P}_N \mathbf{x}_N = \mathbf{x}_N^T \mathbf{P}_N^T \mathbf{P}_N \mathbf{x}_N = \mathbf{x}_N^T \mathbf{x}_N = \|\mathbf{x}_N\|_2^2 = (\alpha \mathbf{e}_{N,i})^T \alpha \mathbf{e}_{N,i} = \alpha^2 \mathbf{e}_{N,i}^T \mathbf{e}_{N,i} = \alpha^2 \quad (9.2)$$

как

$$\alpha = \sqrt{\|\mathbf{x}_N\|_2^2} = \pm \|\mathbf{x}_N\|_2, \quad (9.3)$$

а одним из способов получения соотношения (9.1) является отражение Хаусхольдера — унитарное преобразование (рис. 9.1), не изменяющее норму спроецированного вектора, т. е.

$$\|\alpha \mathbf{e}_{N,i}\|_2 = \|\mathbf{x}_N\|_2. \quad (9.4)$$

Из рис. 9.1 видно, что для выполнения условия (9.4) необходимо, чтобы вектор \mathbf{x}_N совпал «по длине» с вектором $\alpha \mathbf{e}_{N,i}$. Для этого вектор \mathbf{x}_N нужно отразить через линию, совпадающую с биссектрисой угла, образованного векторами \mathbf{x}_N и $\alpha \mathbf{e}_{N,i}$. В этом случае вектор $\alpha \mathbf{e}_{N,i}$ определяется как сумма двух векторов

$$\alpha \mathbf{e}_{N,i} = \mathbf{x}_N + \Delta \mathbf{x}_N. \quad (9.5)$$

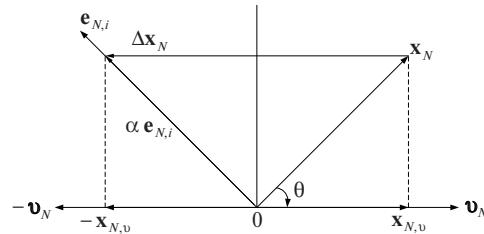


Рис. 9.1. Геометрия отражений Хаусхольдера

Из выражения (9.5) следует, что для нахождения вектора $\alpha \mathbf{e}_{N,i}$ необходимо найти вектор $\Delta \mathbf{x}_N$, который согласно рис. 9.1 определяется как

$$\Delta \mathbf{x}_N = -2 \mathbf{x}_{N,u}, \quad (9.6)$$

где $\mathbf{x}_{N,u}$ — проекция вектора \mathbf{x}_N на вектор \mathbf{v}_N , параллельный вектору $\Delta \mathbf{x}_N$.

Согласно уравнению (9.6) и рис. 9.1 вектор $\Delta \mathbf{x}_N$ можно определить как удвоенное произведение (с отрицательным знаком) единичного направляющего вектора $\hat{\mathbf{v}}_N$ на норму (длину) вектора $\mathbf{x}_{N,u}$, т. е. как

$$\Delta \mathbf{x}_N = -2 \hat{\mathbf{v}}_N \|\mathbf{x}_{N,u}\|_2 = -2 \hat{\mathbf{v}}_N \{ \|\mathbf{x}_N\|_2 \cos \theta \}, \quad (9.7)$$

где

$$\hat{\mathbf{v}}_N = \mathbf{v}_N / \|\mathbf{v}_N\| = \mathbf{v}_N / \sqrt{\mathbf{v}_N^T \mathbf{v}_N} \quad (9.8)$$

— ортонормированный (единичный) направляющий вектор, а

$$\cos \theta = \frac{\mathbf{v}_N^T \mathbf{x}_N}{\|\mathbf{v}_N\|_2 \|\mathbf{x}_N\|_2} = \frac{\mathbf{v}_N^T \mathbf{x}_N}{\sqrt{\mathbf{v}_N^T \mathbf{v}_N} \|\mathbf{x}_N\|_2}. \quad (9.9)$$

С учетом (9.8) и (9.9) уравнение (9.7) определяется как

$$\Delta \mathbf{x}_N = -2 \frac{\mathbf{v}_N}{\sqrt{\mathbf{v}_N^T \mathbf{v}_N}} \|\mathbf{x}_N\| \frac{\mathbf{v}_N^T \mathbf{x}_N}{\sqrt{\mathbf{v}_N^T \mathbf{v}_N} \|\mathbf{x}_N\|_2} = -2 \frac{\mathbf{v}_N \mathbf{v}_N^T \mathbf{x}_N}{\mathbf{v}_N^T \mathbf{v}_N}. \quad (9.10)$$

Следовательно, вектор $\alpha \mathbf{e}_{N,i}$, с учетом уравнений (9.5) и (9.10), определяется как

$$\alpha \mathbf{e}_{N,i} = \mathbf{x}_N + \Delta \mathbf{x}_N = \mathbf{x}_N - 2 \frac{\mathbf{v}_N \mathbf{v}_N^T \mathbf{x}_N}{\mathbf{v}_N^T \mathbf{v}_N} = \left(\mathbf{I}_N - 2 \frac{\mathbf{v}_N \mathbf{v}_N^T}{\mathbf{v}_N^T \mathbf{v}_N} \right) \mathbf{x}_N = \mathbf{P}_N \mathbf{x}_N, \quad (9.11)$$

где

$$\mathbf{P}_N = \mathbf{I}_N - 2 \frac{\mathbf{v}_N \mathbf{v}_N^T}{\mathbf{v}_N^T \mathbf{v}_N} \quad (9.12)$$

— матрица отражения (преобразования) Хаусхолдера.

Согласно [99] вектор \mathbf{E}_N определяется как

$$\mathbf{v}_N = \mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}, \quad (9.13)$$

т. е. векторы \mathbf{E}_N и \mathbf{x}_N отличаются только одной i -й компонентой. Чтобы убедиться в правильности этого утверждения, подставим данное значение вектора в уравнение (9.11). Тогда

$$\begin{aligned} \mathbf{P}_N \mathbf{x}_N &= \mathbf{x}_N - 2 \frac{\mathbf{v}_N \mathbf{v}_N^T \mathbf{x}_N}{\mathbf{v}_N^T \mathbf{v}_N} = \mathbf{x}_N - 2 \frac{(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})^T \mathbf{x}_N}{(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})^T (\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})} = \\ &= \mathbf{x}_N - 2 \frac{(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})(\mathbf{x}_N^T \mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N)}{\mathbf{x}_N^T \mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N \pm \mathbf{x}_N^T \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i} + \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}} = \\ &= \mathbf{x}_N - 2 \frac{(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})(\|\mathbf{x}_N\|_2^2 \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N)}{\|\mathbf{x}_N\|_2^2 \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N \pm \mathbf{x}_N^T \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i} + \|\mathbf{x}_N\|_2^2} = \\ &= \mathbf{x}_N - 2 \frac{(\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i})(\|\mathbf{x}_N\|_2^2 \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N)}{2\|\mathbf{x}_N\|_2^2 \pm 2\|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}^T \mathbf{x}_N} = \\ &= \mathbf{x}_N - (\mathbf{x}_N \pm \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i}) = \mp \|\mathbf{x}_N\|_2 \mathbf{e}_{N,i} = \alpha \mathbf{e}_{N,i}, \end{aligned} \quad (9.14)$$

т. е. уравнение (9.14) совпадает с уравнением (9.1), что подтверждает справедливость уравнения (9.13).

Таким образом, матрица (9.13) отражает вектор \mathbf{x}_N в вектор $\alpha \mathbf{e}_{N,i}$, содержащий только одну ненулевую компоненту, равную $\alpha = \pm \|\mathbf{x}_N\|_2$, в i -м направлении N -мерного евклидового пространства, т. е. в направлении вектора $\mathbf{e}_{N,i}$.

В случае комплексных переменных уравнения (9.1)–(9.11) приводят к матрице Хаусхолдера вида

$$\mathbf{P}_N = \mathbf{I}_N - 2 \frac{\mathbf{v}_N \mathbf{v}_N^H}{\mathbf{v}_N^H \mathbf{v}_N}. \quad (9.15)$$

В этом случае справедливо соотношение

$$\|\mathbf{x}_N\|_2^2 = |\alpha|^2, \quad (9.16)$$

где

$$\alpha = \sqrt{\|\mathbf{x}_N\|_2^2} e^{j\phi_i} = \pm \|\mathbf{x}_N\|_2 e^{j\phi_i} \quad (9.17)$$

и

$$x_i = |x_i| e^{j\phi_i}. \quad (9.18)$$

Действительно, согласно выражению (9.2)

$$\|\mathbf{x}_N\|_2^2 = \mathbf{x}_N^H \mathbf{x}_N = (\alpha \mathbf{e}_{N,i})^H \alpha \mathbf{e}_{N,i} = \mathbf{e}_{N,i}^T \alpha^* \alpha \mathbf{e}_{N,i} = \alpha^* \alpha \mathbf{e}_{N,i}^T \mathbf{e}_{N,i} = |\alpha|^2, \quad (9.19)$$

т. е.

$$\|\mathbf{x}_N\|_2 = |\alpha|. \quad (9.20)$$

Так как $\mathbf{P}_N^H = \mathbf{P}_N$, см. уравнение (9.15), то матрица \mathbf{P}_N является эрмитовой. Тогда согласно выражению (3.12) произведение

$$\mathbf{x}_N^H \mathbf{P}_N \mathbf{x}_N = \mathbf{x}_N^H \alpha \mathbf{e}_{N,i} \geq 0, \quad (9.21)$$

т. е. является положительной действительной величиной.

Из уравнения (9.21) следует, что

$$x_i^* \alpha = |x_i| e^{-j\phi_i} |\alpha| e^{j\psi} \geq 0, \quad (9.22)$$

а значит,

$$\psi = \phi_i, \quad (9.23)$$

что подтверждает справедливость уравнения (9.17).

При использовании в RLS-алгоритмах преобразований Хаусхолдера (9.12) и (9.15) обычно не требуется знание точных выражений, описывающих элементы матрицы отражения, а требуется лишь знание структуры этой матрицы.



9.3. RLS-алгоритм, использующий преобразование Хаусхолдера

Уравнение (7.48) устанавливает связь между матрицами $\mathbf{R}_N(k)$ и $\tilde{\mathbf{R}}_N(k)$ как $\mathbf{R}_N(k) = \tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k)$. Определим такую унитарную матрицу $\mathbf{F}_N(k)$, что

$$\mathbf{F}_N(k)\tilde{\mathbf{R}}_N(k) = \tilde{\mathbf{A}}_N(k), \quad (9.24)$$

где $\tilde{\mathbf{A}}_N(k)$ — квадратная матрица со всеми ненулевыми элементами, в то время как в верхней треугольной матрице $\tilde{\mathbf{R}}_N(k)$ элементы, находящиеся ниже основной диагонали, являются нулевыми.

Так как

$$\begin{aligned} \tilde{\mathbf{A}}_N^H(k)\tilde{\mathbf{A}}_N(k) &= \left[\mathbf{F}_N(k)\tilde{\mathbf{R}}_N(k) \right]^H \left[\mathbf{F}_N(k)\tilde{\mathbf{R}}_N(k) \right] = \\ &= \tilde{\mathbf{R}}_N^H(k)\mathbf{F}_N^H(k)\mathbf{F}_N(k)\tilde{\mathbf{R}}_N(k) = \tilde{\mathbf{R}}_N^H(k)\tilde{\mathbf{R}}_N(k) = \mathbf{R}_N(k), \end{aligned} \quad (9.25)$$

то из уравнения (9.25) следует, что

$$\tilde{\mathbf{A}}_N(k) = \sqrt{\mathbf{R}_N(k)}. \quad (9.26)$$

В уравнении (9.25) учтено, что матрица $\mathbf{F}_N(k)$ является унитарной, т. е. $\mathbf{F}_N^H(k)\mathbf{F}_N(k) = \mathbf{I}_N(k)$.

Так как $[\mathbf{A}_N \mathbf{B}_N]^{-1} = \mathbf{B}_N^{-1}\mathbf{A}_N^{-1}$, $[\mathbf{A}_{NM} \mathbf{B}_{MN}]^H = \mathbf{B}_{MN}^H \mathbf{A}_{NM}^H$ и $\mathbf{F}_N^H(k) = \mathbf{F}_N^{-1}(k)$, то

$$[\mathbf{F}_N(k)\tilde{\mathbf{R}}_N(k)]^{-1} = \tilde{\mathbf{R}}_N^{-1}(k)\mathbf{F}_N^{-1}(k) = \tilde{\mathbf{R}}_N^{-1}(k)\mathbf{F}_N^H(k) = \tilde{\mathbf{A}}_N^{-1}(k) \quad (9.27)$$

и

$$[\tilde{\mathbf{R}}_N^{-1}(k)\mathbf{F}_N^H(k)]^H = \mathbf{F}_N(k)\tilde{\mathbf{R}}_N^{-H}(k) = \tilde{\mathbf{A}}_N^{-H}(k). \quad (9.28)$$

Тогда с учетом уравнений (7.38) $\mathbf{a}_N(k) = \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1)\mathbf{x}_N(k)$ и (9.28) получим, что

$$\begin{aligned} \mathbf{F}_N(k)\mathbf{a}_N(k) &= \mathbf{F}_N(k)\left[\lambda^{-0.5}\tilde{\mathbf{R}}_N^{-H}(k-1)\mathbf{x}_N(k)\right] = \\ &= \lambda^{-0.5}\mathbf{F}_N(k)\tilde{\mathbf{R}}_N^{-H}(k-1)\mathbf{x}_N(k) = \lambda^{-0.5}\tilde{\mathbf{A}}_N^{-H}(k-1)\mathbf{x}_N(k) = \tilde{\mathbf{a}}_N(k), \end{aligned} \quad (9.29)$$

а

$$1 + \tilde{\mathbf{a}}_N^H(k)\tilde{\mathbf{a}}_N(k) = 1 + \mathbf{a}_N^H(k)\mathbf{F}_N^H(k)\mathbf{F}_N(k)\mathbf{a}_N(k) = 1 + \mathbf{a}_N^H(k)\mathbf{a}_N(k) = b_N^2(k), \quad (9.30)$$

см. уравнение (7.42), и

$$\begin{aligned} \lambda^{-0.5}\tilde{\mathbf{A}}_N^{-1}(k-1)\tilde{\mathbf{a}}_N(k)b_N^{-1}(k) &= \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-1}(k-1)\mathbf{F}_N^H(k)\mathbf{F}_N(k)\mathbf{a}_N(k)b_N^{-1}(k) = \\ &= \lambda^{-0.5}\tilde{\mathbf{R}}_N^{-1}(k-1)\mathbf{a}_N(k)b_N^{-1}(k) = \mathbf{u}_N(k), \end{aligned} \quad (9.31)$$

см. уравнение (7.43).

Уравнения (9.24)–(9.31) позволяют, по аналогии с уравнением (7.68), установить соотношение

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) & \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \\ 1 & \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N & \tilde{\mathbf{A}}_N^{-H}(k) \\ -b_N(k) & -\mathbf{u}_N^H(k) \end{bmatrix}, \quad (9.32)$$

где в качестве $\mathbf{P}_{N+1}(k)$ может быть использована матрица Хаусхолдера

$$\mathbf{P}_{N+1}(k) = \mathbf{I}_{N+1}(k) - 2 \frac{\mathbf{v}_{N+1}(k) \mathbf{v}_{N+1}^H(k)}{\mathbf{v}_{N+1}^H(k) \mathbf{v}_{N+1}(k)}, \quad (9.33)$$

определенная уравнением (9.15).

Справедливость соотношения (9.32) может быть доказана следующим образом.

Умножив левую и правую части верхней строчки уравнения (7.20) на $\mathbf{F}_N(k)$, получим, что

$$\mathbf{T}_{N+1}(k) \begin{bmatrix} \sqrt{\lambda} \mathbf{F}_N(k) \tilde{\mathbf{R}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_N(k) \tilde{\mathbf{R}}_N(k) \\ \mathbf{0}_N^T \end{bmatrix}. \quad (9.34)$$

Затем, перемножив полученные прямоугольные матрицы как

$$\begin{aligned} & \left[\lambda^{0.5} \tilde{\mathbf{A}}_N^H(k-1), \mathbf{x}_N(k) \right] \mathbf{T}_{N+1}^H(k) \mathbf{T}_{N+1}(k) \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{A}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \\ & = \left[\lambda^{0.5} \tilde{\mathbf{A}}_N^H(k-1), \mathbf{x}_N(k) \right] \begin{bmatrix} \lambda^{0.5} \tilde{\mathbf{A}}_N(k-1) \\ \mathbf{x}_N^H(k) \end{bmatrix} = \left[\tilde{\mathbf{A}}_N^H(k), \mathbf{0}_N(k) \right] \begin{bmatrix} \tilde{\mathbf{A}}_N(k) \\ \mathbf{0}_N^T \end{bmatrix} \end{aligned} \quad (9.35)$$

и раскрыв выражение (9.35), получим, что

$$\lambda \tilde{\mathbf{A}}_N^H(k-1) \tilde{\mathbf{A}}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) = \tilde{\mathbf{A}}_N^H(k) \tilde{\mathbf{A}}_N(k). \quad (9.36)$$

В уравнениях (9.35) учтено, что матрица $\mathbf{T}_{N+1}(k)$ является унитарной.

Уравнение (9.36), подобно (6.75) и (7.47), представляет собой рекурсивное уравнение вычисления $\tilde{\mathbf{A}}_N^H(k) \tilde{\mathbf{A}}_N(k)$.

Для обращения матрицы (9.36) необходимо применить MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где $\mathbf{C} = \lambda \tilde{\mathbf{A}}_N^H(k-1) \tilde{\mathbf{A}}_N(k-1)$, $\mathbf{C}^{-1} = \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1)$, $\mathbf{B}^H = \mathbf{x}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k)$ и $\mathbf{A}^{-1} = -1$, а также использовать соотношение (3.50) $[\mathbf{A}_N \mathbf{B}_N]^{-1} = \mathbf{B}_N^{-1} \mathbf{A}_N^{-1}$, справедливое для обратимых квадратных матриц. Тогда, аналогично уравнению (7.49) и с учетом уравнений (9.30) и (9.31), получим, что

$$\begin{aligned} & [\tilde{\mathbf{A}}_N^H(k) \tilde{\mathbf{A}}_N(k)]^{-1} = \tilde{\mathbf{A}}_N^{-1}(k) \tilde{\mathbf{A}}_N^{-H}(k) = \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) - \\ & - \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k) \tilde{\mathbf{A}}_N^{-H}(k-1) \mathbf{x}_N(k) \left[1 + \mathbf{x}_N^H(k) \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) \mathbf{x}_N(k) \right]^{-1} \times \\ & \times \mathbf{x}_N^H(k) \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) = \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) - \\ & - \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-1}(k) \tilde{\mathbf{a}}_N(k) \left[1 + \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k) \right]^{-1} \lambda^{-0.5} \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{A}}_N^{-H}(k-1) = \\ & = \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) - \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k) \tilde{\mathbf{a}}_N(k) b_N^{-2}(k) \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{A}}_N^{-H}(k-1) = \\ & = \lambda^{-1} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{R}}_N^{-H}(k-1) - \mathbf{u}_N(k) \mathbf{u}_N^H(k). \end{aligned} \quad (9.37)$$

Из (9.37) следует, что

$$\lambda^{-1} \tilde{\mathbf{A}}_N^{-H}(k-1) \tilde{\mathbf{A}}_N^{-H}(k-1) = \tilde{\mathbf{A}}_N^{-1}(k) \tilde{\mathbf{A}}_N^{-H}(k) + \mathbf{u}_N(k) \mathbf{u}_N^H(k). \quad (9.38)$$

Уравнение (9.38) подразумевает, что существует такая унитарная матрица $\mathbf{P}_{N+1}(k)$, что

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_N^{-H}(k) \\ -\mathbf{u}_N^H(k) \end{bmatrix}. \quad (9.39)$$

Действительно, из (9.39) следует, что

$$\begin{aligned} & \left[\lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1), \mathbf{0}_N^T \right] \mathbf{P}_{N+1}^H(k) \mathbf{P}_{N+1}(k) \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \\ & = \left[\lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1), \mathbf{0}_N^T \right] \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \left[\tilde{\mathbf{A}}_N^{-1}(k), -\mathbf{u}_N(k) \right] \begin{bmatrix} \tilde{\mathbf{A}}_N^{-H}(k) \\ -\mathbf{u}_N^H(k) \end{bmatrix}, \end{aligned} \quad (9.40)$$

а раскрытие выражения (9.40) приводит к уравнению (9.38), что подтверждает справедливость выражения (9.39), т. е. правой части уравнения (9.32).

Справедливость соотношения

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ -b_N(k) \end{bmatrix} \quad (9.41)$$

в уравнении (9.32) докажем аналогично (9.40) как

$$\begin{aligned} & \left[\tilde{\mathbf{a}}_N^H(k), 1 \right] \mathbf{P}_N^H(k) \mathbf{P}_{N+1}(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} = \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k) + 1 = b_N^2(k) = \\ & = \left[\mathbf{0}_N^T(k), -b_N(k) \right] \begin{bmatrix} \mathbf{0}_N \\ -b_N(k) \end{bmatrix} = b_N^2(k). \end{aligned} \quad (9.42)$$

Таким образом, матрица $\mathbf{P}_{N+1}(k)$ не только позволяет рекурсивно вычислять матрицу $\tilde{\mathbf{A}}_N^{-H}(k)$, но и по аналогии с уравнением (7.68) позволяет вычислять $b_N(k)$ и $\mathbf{u}_N^H(k)$ для определения вектора коэффициентов Калмана $\mathbf{g}_N(k) = \mathbf{u}_N(k) b_N^{-1}(k)$.

Матрица $\mathbf{P}_{N+1}(k)$, подобно уравнению (9.1), воздействует на вектор $\left[\tilde{\mathbf{a}}_N^T(k), 1 \right]^T$. Первые N элементов этого вектора «зануляются», но сохраняется норма результирующего вектора, так как

$$\left\| \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} \right\|_2 = \sqrt{1 + \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k)} = b_N(k) = \left\| \begin{bmatrix} \mathbf{0}_N \\ -b_N(k) \end{bmatrix} \right\|_2. \quad (9.43)$$

Из уравнений (9.32), (9.1) и (9.43) следует, что

$$\mathbf{P}_{N+1}(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} = - \left\| \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} \right\|_2 \mathbf{e}_{N+1} = \begin{bmatrix} \mathbf{0}_N \\ -b_N(k) \end{bmatrix}, \quad (9.44)$$

а из уравнений (9.13) и (9.43) — что

$$\mathbf{v}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} + \left\| \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} \right\|_2 \mathbf{e}_{N+1} = \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 \end{bmatrix} + b_N(k) \mathbf{e}_{N+1} = \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 + b_N(k) \end{bmatrix}. \quad (9.45)$$

Для определенности в уравнениях (9.32) и (9.44) использован знак «минус», а во втором слагаемом уравнения (9.45) — знак «плюс», что обусловлено сочетанием двух противоположных знаков слагаемых в этих уравнениях согласно (9.14) и (9.13). Это сочетание арифметических знаков обусловлено и тем, что если в указанных уравнениях знаки поменять на противоположные, то возникающее в этом случае в RLS-алгоритме деление на $1 - b_N(k)$ может быть источником численных проблем, например переполнения разрядной сетки, так как $b_N(k) \approx 1$. В главе 11 будет показано, что

$$e_N(k) = \alpha_N(k) \varphi_N(k) = \alpha_N(k) b_N^{-2}(k), \quad (9.46)$$

где $\varphi_N(k)$ — функция, значения которой — это числа, близкие к единице и не превышающие единицу, а значит, и значение $b_N(k)$ также близко к единице.

Используя уравнения (9.45) и (9.30), определим

$$\begin{aligned} \mathbf{v}_{N+1}^H(k) \mathbf{v}_{N+1}(k) &= \begin{bmatrix} \tilde{\mathbf{a}}_N^H(k), 1 + b_N(k) \\ 1 + b_N(k) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 + b_N(k) \end{bmatrix} = \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k) + [1 + b_N(k)]^2 = \\ &= \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k) + 1 + 2b_N(k) + b_N^2(k) = b_N^2(k) + 2b_N(k) + b_N^2(k) = \\ &= 2b_N^2(k) + 2b_N(k) = 2[b_N^2(k) + b_N(k)] = 2b_N(k)[b_N(k) + 1]. \end{aligned} \quad (9.47)$$

Тогда, с учетом (9.47), уравнение (9.33) приобретает вид

$$\mathbf{P}_{N+1}(k) = \mathbf{I}_{N+1}(k) - \beta_N(k) \mathbf{v}_{N+1}(k) \mathbf{v}_{N+1}^H(k), \quad (9.48)$$

где

$$\beta_N(k) = [b_N^2(k) + b_N(k)]^{-1}, \quad (9.49)$$

а с учетом (9.49), уравнение (9.48) приобретает вид

$$\begin{aligned} \mathbf{P}_{N+1}(k) &= \mathbf{I}_{N+1}(k) - \beta_N(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \\ 1 + b_N(k) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{a}}_N^H(k), 1 + b_N(k) \\ 1 + b_N(k) \end{bmatrix} = \\ &= \mathbf{I}_{N+1}(k) - \beta_N(k) \begin{bmatrix} \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) & \tilde{\mathbf{a}}_N(k)[1 + b_N(k)] \\ [1 + b_N(k)] \tilde{\mathbf{a}}_N^H(k) & [1 + b_N(k)]^2 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{I}_N(k) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) & -\beta_N(k) \tilde{\mathbf{a}}_N(k)[1 + b_N(k)] \\ -\beta_N(k)[1 + b_N(k)] \tilde{\mathbf{a}}_N^H(k) & 1 - \beta_N(k)[1 + b_N(k)]^2 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{I}_N(k) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) & -\frac{\tilde{\mathbf{a}}_N(k)[1 + b_N(k)]}{b_N(k)[1 + b_N(k)]} \\ -\frac{[1 + b_N(k)] \tilde{\mathbf{a}}_N^H(k)}{b_N(k)[1 + b_N(k)]} & 1 - \beta_N(k)[1 + b_N(k)]^2 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{I}_N(k) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) & -\tilde{\mathbf{a}}_N(k) b_N^{-1}(k) \\ -\tilde{\mathbf{a}}_N^H(k) b_N^{-1}(k) & 1 - b_N^{-1}(k)[1 + b_N(k)] \end{bmatrix}. \end{aligned} \quad (9.50)$$

Используя (9.50), уравнение (9.39) можно преобразовать как

$$\begin{bmatrix} \mathbf{I}_N(k) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) & -\tilde{\mathbf{a}}_N(k) b_N^{-1}(k) \\ -\tilde{\mathbf{a}}_N^H(k) b_N^{-1}(k) & 1 - b_N^{-1}(k)[1 + b_N(k)] \end{bmatrix} \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \\ \mathbf{0}_N^T \end{bmatrix} = \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{A}}_N^{-H}(k-1) \\ -\tilde{\mathbf{a}}_N^H(k) b_N^{-1}(k) \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_N^{-H}(k) \\ -\mathbf{u}_N^H(k) \end{bmatrix}. \quad (9.51)$$

Таким образом, нижняя строка матрицы (9.51) подтверждает формулу (9.31) для вычисления $\mathbf{u}_N(k)$ и устанавливает соотношение для рекурсивного вычисления матрицы $\tilde{\mathbf{A}}_N^{-H}(k)$, используемой в формулах RLS-алгоритма. Из верхней части уравнения (9.51) следует, что

$$\begin{aligned} \tilde{\mathbf{A}}_N^{-H}(k) &= \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) - \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{a}}_N^H(k) \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) = \\ &= \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) - \lambda^{-0.5} \beta_N(k) \tilde{\mathbf{a}}_N(k) \tilde{\mathbf{u}}_N^H(k) = \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) - \tilde{\mathbf{a}}_N(k) \hat{\mathbf{u}}_N(k), \end{aligned} \quad (9.52)$$

где

$$\tilde{\mathbf{u}}_N(k) = \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{a}}_N \quad (9.53)$$

и

$$\hat{\mathbf{u}}_N(k) = \lambda^{-0.5} \beta_N(k) \tilde{\mathbf{u}}_N(k). \quad (9.54)$$

Используя выражение для вектора $\mathbf{u}_N(k)$ из уравнения (9.51) и уравнение (9.53), вектор коэффициентов Калмана, см. уравнение (7.45), можно определить как

$$\mathbf{g}_N(k) = \mathbf{u}_N(k) b_N^{-1}(k) = \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{a}}_N(k) b_N^{-1}(k) b_N^{-1}(k) = \lambda^{-0.5} \tilde{\mathbf{u}}_N(k) b_N^{-2}(k) \quad (9.55)$$

и затем использовать выражение (9.55) в уравнении (7.45) для вычисления вектора ВК

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k), \quad (9.56)$$

где

$$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k). \quad (9.57)$$

Таким образом, последовательность уравнений (9.29), (9.30), (9.49), (9.53), (9.54), (9.52), (9.55), (9.57) и (9.56) представляет собой вычислительную процедуру RLS-алгоритма, использующего преобразование Хаусхолдера, которая представлена в табл. 9.1.

Оценка вычислительной сложности RLS-алгоритма (см. табл. 9.1) равна $O(N^2)$ арифметическим операциям. Она состоит из $4N^2 + 5N$ операций умножения, $3N^2 + 3N$ операций сложения, одной операции извлечения квадратного корня и двух операций деления.

Таким образом, вычислительная сложность этого алгоритма в части операций умножения и сложения несколько больше, чем у алгоритмов (см. табл. 7.1

и табл. 8.1). Однако в алгоритме (см. табл. 9.1) число операций извлечения квадратного корня и деления не зависит от числа ВК адаптивного фильтра N .

Таблица 9.1. RLS-алгоритм, использующий преобразование Хаусхолдера

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\tilde{\mathbf{A}}_N^{-1}(0) = \tilde{\mathbf{A}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5}$, $\mathbf{h}_N(0) = \mathbf{0}_N$, $\Lambda_N = \text{diag}\{1, \lambda, \dots, \lambda^{N-1}\}$	(9.1.0)
For $k = 1, 2, \dots, K$	
$\tilde{\mathbf{a}}_N(k) = \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) \mathbf{x}_N(k)$	(9.1.1)
$b_N(k) = \sqrt{1 + \tilde{\mathbf{a}}_N^H(k) \tilde{\mathbf{a}}_N(k)}$	(9.1.2)
$\beta_N(k) = [b_N(k) + b_N^2(k)]^{-1}$	(9.1.3)
$\tilde{\mathbf{u}}_N(k) = \tilde{\mathbf{A}}_N^{-1}(k-1) \tilde{\mathbf{a}}_N$	(9.1.4)
$\hat{\mathbf{u}}_N(k) = \lambda^{-0.5} \beta_N(k) \tilde{\mathbf{u}}_N(k)$	(9.1.5)
$\tilde{\mathbf{A}}_N^{-H}(k) = \lambda^{-0.5} \tilde{\mathbf{A}}_N^{-H}(k-1) - \tilde{\mathbf{a}}_N(k) \hat{\mathbf{u}}_N^H(k)$	(9.1.6)
$\mathbf{g}_N(k) = \lambda^{-0.5} \tilde{\mathbf{u}}_N(k) b_N^{-2}(k)$	(9.1.7)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(9.1.8)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$	(9.1.9)
End for k	

В силу векторного представления вычислений алгоритм (см. табл. 9.1), в первую очередь, ориентирован на программную реализацию с помощью ЦСП. Однако существует возможность реализации этого алгоритма с помощью систолических архитектур, подобных архитектурам, рассмотренным в главах 7 и 8. Детали такой реализации приведены в [104]. Матрицы $\tilde{\mathbf{A}}_N^{-1}(0)$ и $\tilde{\mathbf{A}}_N^{-H}(0)$ инициализируются аналогично IQRD RLS-алгоритму (см. табл. 7.1) диагональной действительной матрицей как

$$\tilde{\mathbf{A}}_N^{-1}(0) = \tilde{\mathbf{A}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5}. \quad (9.58)$$

9.4. Исключение операций извлечения квадратного корня в алгоритмах на основе обратного QR-разложения

Как уже отмечалось, в алгоритмах из табл. 7.1 и табл. 8.1 присутствует N операций извлечения квадратного корня и их программная реализация требует дополнительных вычислительных затрат. При аппаратной реализации IQRD

RLS-алгоритма (см. рис. 7.2) операции извлечения квадратного корня могут потребовать большую площадь, занимаемую на кристалле СБИС. Это в ряде случаев делает такие варианты реализации трудновыполнимыми. Поэтому уменьшение числа операций извлечения квадратного корня или их полного исключения из адаптивного алгоритма является актуальной задачей. Исключить эти операции удается путем масштабирования ряда переменных, участвующих в вычислениях [101, 106, 114].

Рассмотрим такое масштабирование в алгоритме (см. табл. 7.1). Согласно (7.89)–(7.124) на каждой итерации по времени k осуществляется следующая модификация элементов уравнения (7.68):

$$\begin{bmatrix} c_{N,i}(k) & -s_{N,i}^*(k) \\ s_{N,i}(k) & c_{N,i}(k) \end{bmatrix} \begin{bmatrix} a_{N,i}(k) & \lambda^{-0.5} \tilde{R}_{N,i1}^{-H}(k-1) & \dots & \lambda^{-0.5} \tilde{R}_{N,ii}^{-H}(k-1) \\ b_N^{(i-1)}(k) & u_{N,1}^{(i-1)*}(k) & \dots & u_{N,i}^{(i-1)*}(k) \end{bmatrix} = \\ = \begin{bmatrix} 0 & \tilde{R}_{N,i1}^{-H}(k) & \dots & \tilde{R}_{N,ii}^{-H}(k) \\ b_N^{(i)}(k) & u_{N,1}^{(i)*}(k) & \dots & u_{N,i}^{(i)*}(k) \end{bmatrix}, \quad (9.59)$$

т. е.

$$\tilde{R}_{N,ij}^{-H}(k) = c_{N,i}(k) \lambda^{-0.5} \tilde{R}_{N,ij}^{-H}(k-1) - s_{N,i}^*(k) u_{N,j}^{(i-1)*}(k), \quad (9.60)$$

$$u_{N,j}^{(i)*}(k) = s_{N,i}(k) \lambda^{-0.5} \tilde{R}_{N,ij}^{-H}(k-1) + c_{N,i}(k) u_{N,j}^{(i-1)*}(k), \quad (9.61)$$

где $i = 1, 2, \dots, N$, $j = 1, 2, \dots, i$. Эта модификация обеспечивается соответствующим выбором элементов матрицы вращения Гивенса

$$s_{N,i}(k) = a_{N,i}^*(k) / b_N^{(i)}(k) \quad (9.62)$$

и

$$c_{N,i}(k) = b_N^{(i-1)}(k) / b_N^{(i)}(k), \quad (9.63)$$

которые находятся из условия обнуления элемента $a_{N,i}(k)$ в уравнении (9.59), приводящего к вычислению

$$b_N^{(i)}(k) = \sqrt{\left[b_N^{(i-1)}(k) \right]^2 + a_{N,i}^*(k) a_{N,i}(k)}. \quad (9.64)$$

Суть масштабирования заключается в том, что уравнение (9.59) можно представить как

$$\begin{bmatrix} c_{N,i}(k) & -s_{N,i}^*(k) \\ s_{N,i}(k) & c_{N,i}(k) \end{bmatrix} \times \\ \times \begin{bmatrix} \lambda^{-0.5} K_{N,i}^{0.5R}(k-1) & 0 \\ 0 & K_N^{0.5B(i-1)}(k) \end{bmatrix} \begin{bmatrix} \bar{a}_{N,i}(k) & \bar{R}_{N,i1}^{-H}(k-1) & \dots & \bar{R}_{N,ii}^{-H}(k-1) \\ \bar{b}_N^{(i-1)}(k) & \bar{u}_{N,1}^{(i-1)*}(k) & \dots & \bar{u}_{N,i}^{(i-1)*}(k) \end{bmatrix} = \\ = \begin{bmatrix} \lambda^{-0.5} K_{N,i}^{0.5R}(k) & 0 \\ 0 & K_N^{0.5B(i)}(k) \end{bmatrix} \begin{bmatrix} 0 & \bar{R}_{N,i1}^{-H}(k) & \dots & \bar{R}_{N,ii}^{-H}(k) \\ \bar{b}_N^{(i)}(k) & \bar{u}_{N,1}^{(i)*}(k) & \dots & \bar{u}_{N,i}^{(i)*}(k) \end{bmatrix}, \quad (9.65)$$

где масштабирующие множители $K_{N,i}^{0.5R}(k-1)$, $K_N^{0.5B(i-1)}(k)$, $K_{N,i}^{0.5R}(k)$ и $K_N^{0.5B(i)}(k)$ находятся соответственно из условий масштабирования элементов уравнения (9.59) $\lambda^{-0.5} \tilde{R}_{N,ii}^{-H}(k-1)$, $b_N^{(i-1)}(k)$, $\tilde{R}_{N,ii}^{-H}(k)$ и $b_N^{(i)}(k)$ как

$$\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \tilde{R}_{N,ii}^{-H}(k-1) = \lambda^{-0.5} \tilde{R}_{N,ii}^{-H}(k-1), \quad (9.66)$$

$$K_N^{0.5B(i-1)}(k) \bar{b}_N^{(i-1)}(k) = b_N^{(i-1)}(k), \quad (9.67)$$

$$K_{N,i}^{0.5R}(k) \bar{R}_{N,ii}^{-H}(k) = \tilde{R}_{N,ii}^{-H}(k), \quad (9.68)$$

$$K_N^{0.5B(i)}(k) \bar{b}_N^{(i)}(k) = b_N^{(i)}(k), \quad (9.69)$$

где $K_{N,i}^{0.5R}(0) = 1$ и $K_N^{0.5B(0)}(k) = 1$.

С учетом (9.66)–(9.69), а также следующего из уравнений (9.59) и (9.65) соотношения

$$\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \bar{a}_{N,i}(k) = a_{N,i}(k) \quad (9.70)$$

уравнения (9.62) и (9.63) выражаются как

$$\begin{aligned} s_{N,i}(k) &= \frac{a_{N,i}^*(k)}{\sqrt{\left[b_N^{(i-1)}(k) \right]^2 + a_{N,i}^*(k) a_{N,i}(k)}} = \\ &= \frac{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \bar{a}_{N,i}^*(k)}{\sqrt{\left[K_N^{0.5B(i-1)}(k) \right]^2 \left[\bar{b}_N^{(i-1)}(k) \right]^2 + \left[\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \right]^2 \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k)}} = \\ &= \frac{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \bar{a}_{N,i}^*(k)}{\sqrt{\bullet}} \end{aligned} \quad (9.71)$$

И

$$\begin{aligned} c_{N,i}(k) &= \frac{b_N^{(i-1)}(k)}{\sqrt{\left[b_N^{(i-1)}(k) \right]^2 + a_{N,i}^*(k) a_{N,i}(k)}} = \\ &= \frac{K_N^{0.5B(i-1)}(k) \bar{b}_N^{(i-1)}(k)}{\sqrt{\left[K_N^{0.5B(i-1)}(k) \right]^2 \left[\bar{b}_N^{(i-1)}(k) \right]^2 + \left[\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \right]^2 \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k)}} = \\ &= \frac{K_N^{0.5B(i-1)}(k) \bar{b}_N^{(i-1)}(k)}{\sqrt{\bullet}}, \end{aligned} \quad (9.72)$$

где символом $\sqrt{\bullet}$ обозначено выражение знаменателя в уравнениях (9.71) и (9.72), т. е.

$$\begin{aligned} \sqrt{\bullet} &= \sqrt{\left[K_N^{0.5B(i-1)}(k) \right]^2 \left[\bar{b}_N^{(i-1)}(k) \right]^2 + \left[\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \right]^2 \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k)} = \\ &= \sqrt{K_N^{B(i-1)}(k) \left[\bar{b}_N^{(i-1)}(k) \right]^2 + \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k)}. \end{aligned} \quad (9.73)$$

Тогда, с учетом (9.66), (9.68), (9.71)–(9.73), а также следующего из уравнений (9.59) и (9.65) соотношения

$$K_N^{0.5B(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)=\tilde{u}_{N,j}^{(i-1)*}(k) \quad (9.74)$$

уравнение (9.60) можно преобразовать к виду

$$\begin{aligned} K_{N,i}^{0.5R}(k)\bar{R}_{N,ij}^{-H}(k) &= \frac{K_N^{0.5B(i-1)}(k)\bar{b}_N^{(i-1)}(k)}{\sqrt{\bullet}}\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)\bar{R}_{N,ij}^{-H}(k-1)- \\ &- \frac{\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)\bar{a}_{N,i}(k)}{\sqrt{\bullet}}K_N^{0.5B(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)= \\ &= \frac{\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)K_N^{0.5B(i-1)}(k)}{\sqrt{\bullet}}\left[\bar{b}_N^{(i-1)}(k)\bar{R}_{N,ij}^{-H}(k-1)-\bar{a}_{N,i}(k)\bar{u}_{N,j}^{(i-1)*}(k)\right], \end{aligned} \quad (9.75)$$

откуда

$$\bar{R}_{N,ij}^{-H}(k)=\frac{\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)K_N^{0.5B(i-1)}(k)}{K_{N,i}^{0.5R}(k)\sqrt{\bullet}}\left[\bar{b}_N^{(i-1)}(k)\bar{R}_{N,ij}^{-H}(k-1)-\bar{a}_{N,i}(k)\bar{u}_{N,j}^{(i-1)*}(k)\right]. \quad (9.76)$$

Аналогично, с учетом (9.71)–(9.74), а также следующего из уравнений (9.59) и (9.65) соотношения

$$K_N^{0.5B(i)}(k)\bar{u}_{N,j}^{(i)*}(k)=\tilde{u}_{N,j}^{(i)*}(k), \quad (9.77)$$

уравнение (9.61) преобразуется как

$$\begin{aligned} K_N^{0.5B(i)}(k)\bar{u}_{N,j}^{(i)*}(k) &= \frac{\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)\bar{a}_{N,i}^*(k)}{\sqrt{\bullet}}\lambda^{-0.5}K_{N,i}^{0.5R}(k-1)\bar{R}_{N,ij}^{-H}(k-1)+ \\ &+ \frac{K_N^{0.5B(i-1)}(k)\bar{b}_N^{(i-1)}(k)}{\sqrt{\bullet}}K_N^{0.5B(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)= \\ &= \frac{\lambda^{-1}K_{N,i}^R(k-1)}{\sqrt{\bullet}}\bar{a}_{N,i}^*(k)\bar{R}_{N,ij}^{-H}(k-1)+\frac{K_N^{B(i-1)}(k)}{\sqrt{\bullet}}\bar{b}_N^{(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)= \\ &= \frac{1}{\sqrt{\bullet}}\left[\lambda^{-1}K_{N,i}^R(k-1)\bar{a}_{N,i}^*(k)\bar{R}_{N,ij}^{-H}(k-1)+K_N^{B(i-1)}(k)\bar{b}_N^{(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)\right], \end{aligned} \quad (9.78)$$

откуда

$$\bar{u}_{N,j}^{(i)*}(k)=\frac{1}{K_N^{0.5B(i)}(k)\sqrt{\bullet}}\left[\lambda^{-1}K_{N,i}^R(k-1)\bar{a}_{N,i}^*(k)\bar{R}_{N,ij}^{-H}(k-1)+K_N^{B(i-1)}(k)\bar{b}_N^{(i-1)}(k)\bar{u}_{N,j}^{(i-1)*}(k)\right]. \quad (9.79)$$

С учетом (9.67), (9.69), (9.70) и (9.73), уравнение (9.64) также можно представить как

$$K_N^{0.5B(i)}(k)\bar{b}_N^{(i)}(k)=\sqrt{K_N^{B(i-1)}(k)\left[\bar{b}_N^{(i-1)}(k)\right]^2+\lambda^{-1}K_{N,i}^R(k-1)\bar{a}_{N,i}^*(k)\bar{a}_{N,i}(k)}=\sqrt{\bullet}, \quad (9.80)$$

откуда следует, что

$$\bar{b}_N^{(i)}(k)=\sqrt{\bullet}/K_N^{0.5B(i)}(k). \quad (9.81)$$

Чтобы в уравнениях (9.76) и (9.79) избежать операций извлечения квадратного корня, необходимо некоторым образом определить параметры $K_{N,i}^R(k)$ и $K_N^{0.5B(i)}(k)$. Если параметр $K_{N,i}^R(k)$ определить как

$$K_{N,i}^R(k) = \frac{\lambda^{-1} K_{N,i}^R(k-1) K_N^{B(i-1)}(k)}{\left(\sqrt{\bullet}\right)^2}, \quad (9.82)$$

тогда в уравнении (9.76) множитель $\frac{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5B(i-1)}(k)}{K_{N,i}^{0.5R}(k) \sqrt{\bullet}}$ можно преобразовать к виду

$$\frac{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5B(i-1)}(k)}{K_{N,i}^{0.5R}(k) \sqrt{\bullet}} = \frac{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5B(i-1)}(k) \sqrt{\bullet}}{\lambda^{-0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5B(i-1)}(k) \sqrt{\bullet}} = 1, \quad (9.83)$$

а уравнение (9.76) примет вид

$$\bar{R}_{N,ij}^{-H}(k) = \bar{b}_N^{(i-1)}(k) \bar{R}_{N,ij}^{-H}(k-1) - \bar{a}_{N,i}(k) \bar{u}_{N,j}^{(i-1)*}(k). \quad (9.84)$$

Аналогично, если параметр $K_N^{B(i)}(k)$ определить как

$$K_N^{B(i)}(k) = \left(\sqrt{\bullet}\right)^2, \quad (9.85)$$

тогда в уравнении (9.79) множитель $\frac{1}{K_N^{0.5B(i)}(k) \sqrt{\bullet}}$ преобразуется как

$$\frac{1}{K_N^{0.5B(i)}(k) \sqrt{\bullet}} = \frac{1}{\sqrt{\bullet} \sqrt{\bullet}} = \frac{1}{\left(\sqrt{\bullet}\right)^2}, \quad (9.86)$$

а уравнение (9.79) примет вид

$$\bar{u}_{N,j}^{(i)*}(k) = \frac{1}{\left(\sqrt{\bullet}\right)^2} \left[\lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) \bar{R}_{N,ij}^{-H}(k-1) + K_N^{B(i-1)}(k) \bar{b}_N^{(i-1)}(k) \bar{u}_{N,j}^{(i-1)*}(k) \right]. \quad (9.87)$$

В этом случае из уравнений (9.81) и (9.85) также следует, что

$$\bar{b}_N^{(i)}(k) = \sqrt{\bullet} / K_N^{0.5B(i)}(k) = \sqrt{\bullet} / \sqrt{\bullet} = 1. \quad (9.88)$$

С учетом (9.88), уравнение (9.85), см. (9.73), приобретает вид

$$\begin{aligned} K_N^{B(i)}(k) &= K_N^{B(i-1)}(k) \left[\bar{b}_N^{(i-1)}(k) \right]^2 + \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k) = \\ &= K_N^{B(i-1)}(k) + \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k), \end{aligned} \quad (9.89)$$

а уравнение (9.84) — вид

$$\bar{R}_{N,ij}^{-H}(k) = \bar{R}_{N,ij}^{-H}(k-1) - \bar{a}_{N,i}(k) \bar{u}_{N,j}^{(i-1)*}(k). \quad (9.90)$$

Если ввести обозначения

$$\bar{s}_{N,i}(k) = \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) / K_N^{B(i)}(k), \quad (9.91)$$

$$\bar{c}_{N,i}(k) = K_N^{B(i-1)}(k) / K_N^{B(i)}(k), \quad (9.92)$$

то с учетом (9.88) уравнение (9.87) примет вид

$$\bar{u}_{N,j}^{(i)*}(k) = \bar{s}_{N,i}(k) \bar{R}_{N,j}^{-H}(k-1) + \bar{c}_{N,i}(k) \bar{u}_{N,j}^{(i-1)*}(k), \quad (9.93)$$

а уравнение (9.82) — как

$$K_{N,i}^R(k) = \lambda^{-1} K_{N,i}^R(k-1) \bar{c}_{N,i}(k). \quad (9.94)$$

Таким образом, уравнения, которые получаются в процессе рассмотренного масштабирования, составляют RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса без операций извлечения квадратного корня.

Вычисление вектора ВК $\mathbf{h}_N(k)$ в этом алгоритме осуществляется с использованием таких же уравнений, как в алгоритмах (см. табл. 6.1 и табл. 7.1), для чего используется вектор коэффициентов Калмана $\mathbf{g}_N(k)$ и значения ошибок $\alpha_N(k)$. С учетом масштабирования переменных, см. уравнения (7.45), (7.1.8) и (9.88), вектор $\mathbf{g}_N(k)$ определяется как

$$\mathbf{g}_N(k) = \frac{\mathbf{u}_N(k)}{b_N(k)} = \frac{\mathbf{u}_N^{(N)}(k)}{b_N^{(N)}(k)} = \frac{K_N^{0.5B}(k) \bar{\mathbf{u}}_N^{(N)}(k)}{K_N^{0.5B}(k) \bar{b}_N^{(N)}(k)} = \bar{\mathbf{u}}_N^{(N)}(k), \quad (9.95)$$

а уравнение (7.1.2), с учетом уравнений (9.66) и (9.70), — как

$$\begin{aligned} a_{N,i}(k) &= \lambda^{-0.5} \tilde{\mathbf{R}}_N^{-H}(k-1) \Big|_{i,i} \mathbf{x}_N(k) \Big|_{i,i} = \\ &= \lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \bar{\mathbf{R}}_N^{-H}(k-1) \Big|_{i,i} \mathbf{x}_N(k) \Big|_{i,i} = \\ &= \lambda^{-0.5} K_{N,i}^{0.5R}(k-1) \bar{a}_{N,i}(k), \end{aligned} \quad (9.96)$$

откуда следует, что

$$\bar{a}_{N,i}(k) = \bar{\mathbf{R}}_N^{-H}(k-1) \Big|_{i,i} \mathbf{x}_N(k) \Big|_{i,i}. \quad (9.97)$$

Таким образом, уравнения (9.97), (9.89), (9.91), (9.92), (9.90), (9.93), (9.94), (9.95), а также уравнения вычисления ВК $\mathbf{h}_N(k)$ и ошибки $\alpha_N(k)$ составляют RLS-алгоритм на основе обратного QR-разложения без операций извлечения квадратного корня, который представлен в табл. 9.2.

В табл. 9.2 $\mathbf{K}_N^R(k)$ — это диагональная матрица, составленная из элементов $K_{N,i}^R(k)$. Начальное значение этой матрицы задается как $\mathbf{K}_N^R(0) = \mathbf{I}_N$, см. уравнение (9.66). Начальное значение матрицы $\bar{\mathbf{R}}_N^{-H}(0)$ в алгоритме (см. табл. 9.2) задается аналогично начальному значению матрицы $\tilde{\mathbf{R}}_N^{-H}(0)$ в алгоритме (см. табл. 7.1), так как с учетом уравнения (9.66)

$$\tilde{\mathbf{R}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5} = \mathbf{K}_N^{0.5R}(0) \bar{\mathbf{R}}_N^{-H}(0) = \mathbf{I}_N \bar{\mathbf{R}}_N^{-H}(0) = \bar{\mathbf{R}}_N^{-H}(0). \quad (9.98)$$

Кроме того, при рекурсивном вычислении вектора $\mathbf{u}_N^H(k)$ его начальное значение определяется как $\mathbf{u}_N^{(0)H}(k) = \mathbf{0}_N^T$, см. раздел 7.6, а так как согласно уравнениям (9.65) и (9.95) $\mathbf{u}_N^H(k) = \mathbf{u}_N^{(N)H}(k) = K_N^{0.5B}(k) \bar{\mathbf{u}}_N^{(N)H}(k)$, то $\bar{u}_{N,j}^{(j-1)*}(k) = 0$, $1 \leq j \leq N$, см. уравнение (9.2.1), и $K_N^{B(0)}(k) = 1$, см. уравнение (9.67).

Таблица 9.2. RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса (без операций извлечения квадратного корня)

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\bar{\mathbf{R}}_N^{-H}(0) = \delta^{-1} \Lambda_N^{0.5}$, $\mathbf{h}_N(0) = \mathbf{0}_N$, $\mathbf{K}_N^R(0) = \mathbf{I}_N$, $\Lambda_N = \text{diag}\{1, \lambda, \dots, \lambda^{N-1}\}$	(9.2.0)
For $k = 1, 2, \dots, K$	
$\bar{u}_{N,j}^{(j-1)*}(k) = 0$, $1 \leq j \leq N$, $K_N^{B(0)}(k) = 1$	(9.2.1)
For $i = 1, 2, \dots, N$	
$\bar{a}_{N,i}(k) = \bar{\mathbf{R}}_N^{-H}(k-1)_{i,1:i} \left \mathbf{x}_N(k) \right _{1:i}$	(9.2.2)
$K_N^{B(i)}(k) = K_N^{B(i-1)}(k) + \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) \bar{a}_{N,i}(k)$	(9.2.3)
$\bar{s}_{N,i}(k) = \lambda^{-1} K_{N,i}^R(k-1) \bar{a}_{N,i}^*(k) / K_N^{B(i)}(k)$	(9.2.4)
$\bar{c}_{N,i}(k) = K_N^{B(i-1)}(k) / K_N^{B(i)}(k)$	(9.2.5)
For $j = 1, 2, \dots, i$	
$\bar{\mathbf{R}}_{N,ij}^{-H}(k) = \bar{\mathbf{R}}_{N,ij}^{-H}(k-1) - \bar{a}_{N,i}(k) \bar{u}_{N,j}^{(i-1)*}(k)$	(9.2.6)
$\bar{u}_{N,j}^{(i)*}(k) = \bar{s}_{N,i}(k) \bar{\mathbf{R}}_{N,ij}^{-H}(k-1) + \bar{c}_{N,i}(k) \bar{u}_{N,j}^{(i-1)*}(k)$	(9.2.7)
End for j	
$K_{N,i}^R(k) = \lambda^{-1} K_{N,i}^R(k-1) \bar{c}_{N,i}(k)$	(9.2.8)
End for i	
$\mathbf{g}_N(k) = \bar{\mathbf{u}}_N^{(N)}(k)$	(9.2.9)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(9.2.10)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$	(9.2.11)
End for k	

Оценка вычислительной сложности RLS-алгоритма в табл. 9.2 равна $O(N^2)$ арифметическим операциям. Она состоит из $2N^2 + 10N$ операций умножения, $1.5N^2 + 4.5N$ операций сложения и N операций деления. Операции извлечения квадратного корня в алгоритме отсутствуют.

Структура вычислителя для реализация алгоритма (см. табл. 9.2) во многом аналогична структуре рассмотренной в разделе 7.7.

9.5. Исключение операций извлечения квадратного корня в алгоритмах на основе прямого QR-разложения

Чтобы избавиться от операций извлечения квадратного корня в алгоритме на основе прямого QR-разложения (см. табл. 8.1), уравнения (7.20) и (7.21) нужно промасштабировать, что осуществляется следующим образом.

На каждой итерации по времени k осуществляется модификация элементов уравнений (7.20) и (7.21), см. уравнения (8.21), (8.22), (8.31) и (8.32), как

$$\begin{bmatrix} c_{N,i}(k) & -s_{N,i}^*(k) \\ s_{N,i}(k) & c_{N,i}(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{R}_{N,ii}(k-1) & \cdots & \lambda^{0.5} \tilde{R}_{N,iN}(k-1) & \lambda^{0.5} \tilde{r}_{N,i}(k-1) \\ \tilde{x}_{N,i}^{(i-1)*}(k) & \cdots & \tilde{x}_{N,N}^{(i-1)*}(k) & \tilde{\alpha}_N^{(i-1)*}(k) \end{bmatrix} = \begin{bmatrix} \tilde{R}_{N,ii}(k) & \tilde{R}_{N,i(i+1)}(k) & \cdots & \tilde{R}_{N,iN}(k) & \tilde{r}_{N,i}(k) \\ 0 & \tilde{x}_{N,i+1}^{(i)*}(k) & \cdots & \tilde{x}_{N,N}^{(i)*}(k) & \tilde{\alpha}_N^{(i)*}(k) \end{bmatrix}, \quad (9.99)$$

т. е.

$$\tilde{R}_{N,ij}(k) = c_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ij}(k-1) - s_{N,i}^*(k) \tilde{x}_{N,j}^{(i-1)*}(k), \quad (9.100)$$

$$\tilde{x}_{N,j}^{(i)*}(k) = s_{N,i}(k) \lambda^{0.5} \tilde{R}_{N,ij}(k-1) + c_{N,i}(k) \tilde{x}_{N,j}^{(i-1)*}(k), \quad (9.101)$$

$$\tilde{r}_{N,i}(k) = c_{N,i}(k) \lambda^{0.5} \tilde{r}_{N,i}(k-1) - s_{N,i}^*(k) \tilde{\alpha}_N^{(i-1)*}(k), \quad (9.102)$$

$$\tilde{\alpha}_N^{(i)*}(k) = s_{N,i}(k) \lambda^{0.5} \tilde{r}_{N,i}(k-1) + c_{N,i}(k) \tilde{\alpha}_N^{(i-1)*}(k), \quad (9.103)$$

где $i = 1, 2, \dots, N$, $j = i+1, \dots, N$, а также согласно уравнению (8.17)

$$\tilde{R}_{N,ii}(k) = \sqrt{\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)}(k) \tilde{x}_{N,i}^{(i-1)*}(k)}. \quad (9.104)$$

Эта модификация обеспечивается соответствующим выбором элементов матрицы вращения Гивенса

$$s_{N,i}(k) = -\tilde{x}_{N,i}^{(i-1)*}(k) / \tilde{R}_{N,ii}(k), \quad (9.105)$$

$$c_{N,i}(k) = \lambda^{0.5} \tilde{R}_{N,ii}(k-1) / \tilde{R}_{N,ii}(k), \quad (9.106)$$

которые получаются из условия обнуления элемента $\tilde{x}_{N,i+1}^{(i)*}(k)$ в уравнении (9.99), см. раздел 8.2.

Суть масштабирования заключается в том, что уравнение (9.99) представляется как

$$\begin{bmatrix} c_{N,i}(k) & -s_{N,i}^*(k) \\ s_{N,i}(k) & c_{N,i}(k) \end{bmatrix} \times \begin{bmatrix} \lambda^{0.5} K_{N,i}^{0.5R}(k-1) & 0 \\ 0 & K_N^{0.5\alpha(i-1)}(k) \end{bmatrix} \begin{bmatrix} \bar{R}_{N,ii}(k-1) & \cdots & \bar{R}_{N,iN}(k-1) & \bar{r}_{N,i}(k-1) \\ \bar{x}_{N,i}^{(i-1)*}(k) & \cdots & \bar{x}_{N,N}^{(i-1)*}(k) & \bar{\alpha}_N^{(i-1)*}(k) \end{bmatrix} = \begin{bmatrix} \lambda^{-0.5} K_{N,i}^{0.5R}(k) & 0 \\ 0 & K_N^{0.5\alpha(i)}(k) \end{bmatrix} \begin{bmatrix} \bar{R}_{N,ii}(k) & \bar{R}_{N,i+1}(k) & \cdots & \bar{R}_{N,iN}(k) & \bar{r}_{N,i}(k) \\ 0 & \bar{x}_{N,i+1}^{(i)*}(k) & \cdots & \bar{x}_{N,N}^{(i)*}(k) & \bar{\alpha}_N^{(i)*}(k) \end{bmatrix}, \quad (9.107)$$

где масштабирующие множители $K_{N,i}^{0.5R}(k-1)$, $K_N^{0.5\alpha(i-1)}(k)$, $K_{N,i}^{0.5R}(k)$ и $K_N^{0.5\alpha(i)}(k)$ находятся из условий масштабирования элементов уравнения (9.99) $\lambda^{0.5} \tilde{R}_{N,ii}(k-1)$, $\tilde{\alpha}_N^{(i-1)*}(k)$, $\tilde{R}_{N,ii}(k)$ и $\tilde{\alpha}_N^{(i)*}(k)$ как

$$\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1) = \lambda^{0.5} \tilde{R}_{N,ii}(k-1), \quad (9.108)$$

$$K_N^{0.5\alpha(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k) = \tilde{\alpha}_N^{(i-1)*}(k), \quad (9.109)$$

$$K_{N,i}^{0.5R}(k) \bar{R}_{N,ii}(k) = \tilde{R}_{N,ii}(k), \quad (9.110)$$

$$K_N^{0.5\alpha(i)}(k) \bar{\alpha}_N^{(i)*}(k) = \tilde{\alpha}_N^{(i)*}(k). \quad (9.111)$$

С учетом (9.104), уравнения (9.105) и (9.106) выражаются как

$$\begin{aligned} s_{N,i}(k) &= -\frac{-\tilde{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k)}} = \\ &= -\frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{[\lambda^{0.5} K_{N,i}^{0.5R}(k-1)]^2 \bar{R}_{N,ii}^2(k-1) + [K_{N,i}^{0.5\alpha(i-1)}(k)]^2 \bar{x}_{N,i}^{(i-1)*}(k) \bar{x}_{N,i}^{(i-1)}(k)}} = \\ &= -\frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\bullet}} \end{aligned} \quad (9.112)$$

и

$$\begin{aligned} c_{N,i}(k) &= \frac{\lambda^{0.5} \tilde{R}_{N,ii}(k-1)}{\sqrt{\lambda \tilde{R}_{N,ii}^2(k-1) + \tilde{x}_{N,i}^{(i-1)*}(k) \tilde{x}_{N,i}^{(i-1)}(k)}} = \\ &= \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{[\lambda^{0.5} K_{N,i}^{0.5R}(k-1)]^2 \bar{R}_{N,ii}^2(k-1) + [K_N^{0.5\alpha(i-1)}(k)]^2 \bar{x}_{N,i}^{(i-1)*}(k) \bar{x}_{N,i}^{(i-1)}(k)}} = \\ &= \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}}, \end{aligned} \quad (9.113)$$

где символом $\sqrt{\bullet}$ обозначено выражение знаменателя в уравнениях (9.112) и (9.113).

Тогда, с учетом (9.112) и (9.113), уравнение (9.100) можно преобразовать как

$$\begin{aligned} K_{N,i}^{0.5R}(k) \bar{R}_{N,ij}(k) &= \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \lambda^{0.5} K_N^{0.5R}(k-1) \bar{R}_{N,ij}(k-1) + \\ &+ \frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k)}{\sqrt{\bullet}} K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,j}^{(i-1)*}(k) = \\ &= \frac{\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \bar{R}_{N,ij}(k-1) + \frac{K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k)}{\sqrt{\bullet}} \bar{x}_{N,j}^{(i-1)*}(k), \end{aligned} \quad (9.114)$$

откуда

$$\begin{aligned} \bar{R}_{N,ij}(k) = & \frac{1}{K_{N,i}^{0.5R}(k)\sqrt{\bullet}} \times \\ & \times \left[\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1) \bar{R}_{N,ij}(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k) \bar{x}_{N,j}^{(i-1)*}(k) \right]. \end{aligned} \quad (9.115)$$

Аналогично уравнение (9.101) можно преобразовать как

$$\begin{aligned} K_N^{0.5\alpha(i)}(k) \bar{x}_{N,j}^{(i)*}(k) = & \\ = & - \frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\bullet}} \lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ij}(k-1) + \\ + & \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,j}^{(i-1)*}(k) = \\ = & - \frac{K_N^{0.5\alpha(i-1)}(k) \lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\bullet}} \bar{R}_{N,ij}(k-1) + \\ + & \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \bar{x}_{N,j}^{(i-1)*}(k) = \\ = & \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k)}{\sqrt{\bullet}} \left[-\bar{x}_{N,i}^{(i-1)*}(k) \bar{R}_{N,ij}(k-1) + \bar{R}_{N,ii}(k-1) \bar{x}_{N,j}^{(i-1)*}(k) \right], \end{aligned} \quad (9.116)$$

откуда

$$\bar{x}_{N,j}^{(i)*}(k) = \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k)}{K_N^{0.5\alpha(i)}(k)\sqrt{\bullet}} \left[-\bar{x}_{N,i}^{(i-1)*}(k) \bar{R}_{N,ij}(k-1) + \bar{R}_{N,ii}(k-1) \bar{x}_{N,j}^{(i-1)*}(k) \right]. \quad (9.117)$$

Затем уравнение (9.102) можно преобразовать как

$$\begin{aligned} K_{N,i}^{0.5R}(k) \bar{r}_{N,i}(k) = & \\ = & \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{r}_{N,i}(k-1) + \\ + & \frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k)}{\sqrt{\bullet}} K_N^{0.5\alpha(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k) = \\ = & \frac{\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \bar{r}_{N,i}(k-1) + \\ + & \frac{K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k)}{\sqrt{\bullet}} \bar{\alpha}_N^{(i-1)*}(k), \end{aligned} \quad (9.118)$$

откуда

$$\begin{aligned} \bar{r}_{N,i}(k) = & \frac{1}{K_{N,i}^{0.5R}(k)\sqrt{\bullet}} \times \\ & \times \left[\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1) \bar{r}_{N,i}(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k) \right]. \end{aligned} \quad (9.119)$$

И уравнение (9.103) можно преобразовать как

$$\begin{aligned}
K_N^{0.5\alpha(i)}(k) \bar{\alpha}_N^{(i)*}(k) &= -\frac{K_N^{0.5\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\bullet}} \lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{r}_{N,i}(k-1) + \\
&+ \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} K_N^{0.5\alpha(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k) = \\
&= -\frac{K_N^{0.5\alpha(i-1)}(k) \lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{x}_{N,i}^{(i-1)*}(k)}{\sqrt{\bullet}} \bar{r}_{N,i}(k-1) + \\
&+ \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k) \bar{R}_{N,ii}(k-1)}{\sqrt{\bullet}} \bar{\alpha}_N^{(i-1)*}(k) = \\
&= \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k)}{\sqrt{\bullet}} \left[-\bar{x}_{N,i}^{(i-1)*}(k) \bar{r}_{N,i}(k-1) + \bar{R}_{N,ii}(k-1) \bar{\alpha}_N^{(i-1)*}(k) \right], \tag{9.120}
\end{aligned}$$

откуда

$$\bar{\alpha}_N^{(i)*}(k) = \frac{\lambda^{0.5} K_{N,i}^{0.5R}(k-1) K_N^{0.5\alpha(i-1)}(k)}{K_N^{0.5\alpha(i)}(k) \sqrt{\bullet}} \left[-\bar{x}_{N,i}^{(i-1)*}(k) \bar{r}_{N,i}(k-1) + \bar{R}_{N,ii}(k-1) \bar{\alpha}_N^{(i-1)*}(k) \right]. \tag{9.121}$$

Уравнение (9.104) также можно представить как

$$\begin{aligned}
K_{N,i}^{0.5R}(k) \bar{R}_{N,ii}(k) &= \sqrt{\left[\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \right]^2 \bar{R}_{N,ii}^2(k-1) + \left[K_N^{0.5\alpha(i-1)}(k) \right]^2 \bar{x}_{N,i}^{(i-1)*}(k) \bar{x}_{N,i}^{(i-1)}(k)} = \\
&= \sqrt{\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}^2(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k) \bar{x}_{N,i}^{(i-1)}(k)} = \sqrt{\bullet}, \tag{9.122}
\end{aligned}$$

откуда

$$\bar{R}_{N,ii}(k) = \sqrt{\bullet} / K_{N,i}^{0.5R}(k). \tag{9.123}$$

Чтобы в уравнениях (9.115), (9.117), (9.119) и (9.121) избежать операций извлечения квадратного корня, необходимо некоторым образом определить параметры $K_{N,i}^R(k)$ и $K_N^{0.5\alpha(i)}(k)$. Если их определить как

$$K_{N,i}^R(k) = \left(\sqrt{\bullet} \right)^2 \tag{9.124}$$

и

$$K_N^{\alpha(i)}(k) = \frac{\lambda K_{N,i}^R(k-1) K_N^{\alpha(i-1)}(k)}{\left(\sqrt{\bullet} \right)^2}, \tag{9.125}$$

то уравнения (9.115), (9.117), (9.119) и (9.121) можно преобразовать как

$$\bar{R}_{N,ij}(k) = \frac{1}{\left(\sqrt{\bullet} \right)^2} \left[\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1) \bar{R}_{N,ij}(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k) \bar{x}_{N,j}^{(i-1)*}(k) \right], \tag{9.126}$$

$$\bar{x}_{N,i}^{(i)*}(k) = -\bar{x}_{N,i}^{(i-1)*}(k) \bar{R}_{N,ij}(k-1) + \bar{R}_{N,ii}(k-1) \bar{x}_{N,j}^{(i-1)*}(k), \tag{9.127}$$

$$\bar{r}_{N,i}(k) = \frac{1}{\left(\sqrt{\bullet} \right)^2} \left[\lambda K_{N,i}^R(k-1) \bar{R}_{N,ii}(k-1) \bar{r}_{N,i}(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k) \right], \tag{9.128}$$

$$\bar{\alpha}_N^{(i)*}(k) = -\bar{x}_{N,i}^{(i-1)*}(k)\bar{r}_{N,i}(k-1) + \bar{R}_{N,ii}(k-1)\bar{\alpha}_N^{(i-1)*}(k). \quad (9.129)$$

В этом случае из уравнений (9.123) и (9.124) также следует, что

$$\bar{R}_{N,ii}(k) = \sqrt{\bullet} / K_{N,i}^{0.5R}(k) = \sqrt{\bullet} / \sqrt{\bullet} = 1. \quad (9.130)$$

С учетом (9.130) уравнение (9.122) приобретает вид

$$K_{N,i}^R(k) = \lambda K_{N,i}^R(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k), \quad (9.131)$$

а уравнения (9.127) и (9.129) — вид

$$\bar{x}_{N,j}^{(i)*}(k) = -\bar{x}_{N,j}^{(i-1)*}(k) \bar{R}_{N,ij}(k-1) + \bar{x}_{N,j}^{(i-1)*}(k), \quad (9.132)$$

$$\bar{\alpha}_N^{(i)*}(k) = -\bar{x}_{N,i}^{(i-1)*}(k) \bar{r}_{N,i}(k-1) + \bar{\alpha}_N^{(i-1)*}(k). \quad (9.133)$$

Если ввести обозначения

$$\bar{s}_{N,i}(k) = -K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k) / K_{N,i}^R(k), \quad (9.134)$$

$$\bar{c}_{N,i}(k) = \lambda K_{N,i}^R(k-1) / K_{N,i}^R(k), \quad (9.135)$$

то с учетом (9.130) уравнения (9.126) и (9.128) приобретут вид

$$\bar{R}_{N,ij}(k) = \bar{c}_{N,i}(k) \bar{R}_{N,ij}(k-1) - \bar{s}_{N,i}^{*}(k) \bar{x}_{N,j}^{(i-1)*}(k), \quad (9.136)$$

$$\bar{r}_{N,i}(k) = \bar{c}_{N,i}(k) \bar{r}_{N,i}(k-1) - \bar{s}_{N,i}^{*}(k) \bar{\alpha}_N^{(i-1)*}(k) \quad (9.137)$$

а уравнение (9.125) — вид

$$K_N^{\alpha(i)}(k) = \bar{c}_{N,i}(k) K_N^{\alpha(i-1)}(k). \quad (9.138)$$

Из уравнений (9.113) и (9.135) также следует, что $c_{N,i}(k) = \bar{c}_{N,i}^2(k)$, а из уравнений (9.138) и (8.1.9) — что

$$[\tilde{\varphi}_N^{(N)}(k)]^2 = [K_N^{0.5\alpha(N)}]^2. \quad (9.139)$$

Тогда из уравнений (9.139) и (8.1.10) следует, что

$$\alpha_N(k) = \tilde{\alpha}_N^{(N)}(k) / \tilde{\varphi}_N^{(N)}(k) = K_N^{0.5\alpha(N)} \bar{\alpha}_N^{(N)}(k) / K_N^{0.5\alpha(N)} = \bar{\alpha}_N^{(N)}(k). \quad (9.140)$$

Таким образом, уравнения, которые получаются в процессе рассмотренного масштабирования, а именно (9.131), (9.134), (9.135), (9.136), (9.132), (9.137), (9.133), (9.138), (9.140), составляют RLS-алгоритм на основе обратного прямого QR-разложения без операций извлечения квадратного корня, который представлен в табл. 9.3.

Из уравнения (9.130) следует, что $\bar{R}_{N,ii}(k) = 1$ независимо от значения k . Поэтому начальное значение элементов матрицы $\bar{\mathbf{R}}_N(k)$ можно определить как $\bar{\mathbf{R}}_N(0) = \mathbf{I}_N$. Фактически для использования в вычислениях алгоритма из табл. 9.3

достаточно, чтобы элементы матрицы $\bar{\mathbf{R}}_N(0)$, находящиеся вне основной диагонали, были нулевыми, а на основной диагонали могут быть произвольными, так как они в вычислениях не участвуют.

Таблица 9.3. RLS-алгоритм на основе прямого QR-разложения с использованием вращений Гивенса (без операций извлечения квадратного корня)

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\bar{\mathbf{r}}_N(0) = \mathbf{0}_N$, $\bar{\mathbf{R}}_N(0) = \mathbf{I}_N$, $\mathbf{K}_N^R(0) = \delta^2 \Lambda_N^{-1}$, $\Lambda_N = \text{diag}\{1, \lambda, \dots, \lambda^{N-1}\}$	(9.3.0)
For $k = 1, 2, \dots, K$	
$\bar{\mathbf{x}}_N^{(0)*}(k) = \mathbf{x}_N^*(k)$, $\bar{\alpha}_N^{(0)*}(k) = d^*(k)$, $K_N^{\alpha(0)}(k) = 1$	(9.3.1)
For $i = 1, 2, \dots, N$	
$\bar{K}_{N,i}^R(k) = \lambda K_{N,i}^R(k-1) + K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k) \bar{x}_{N,i}^{(i-1)*}(k)$	(9.3.2)
$\bar{s}_{N,i}(k) = -K_N^{\alpha(i-1)}(k) \bar{x}_{N,i}^{(i-1)*}(k) / K_{N,i}^R(k)$	(9.3.3)
$\bar{c}_{N,i}(k) = \lambda K_{N,i}^R(k-1) / K_{N,i}^R(k)$	(9.3.4)
For $j = i+1, \dots, N$	
$\bar{R}_{N,ij}(k) = \bar{c}_{N,i}(k) \bar{R}_{N,ij}(k-1) - \bar{s}_{N,i}(k) \bar{x}_{N,j}^{(i-1)*}(k)$	(9.3.5)
$\bar{x}_{N,j}^{(i)*}(k) = -\bar{x}_{N,i}^{(i-1)*}(k) \bar{R}_{N,ij}(k-1) + \bar{x}_{N,j}^{(i-1)*}(k)$	(9.3.6)
End for j	
$\bar{r}_{N,i}(k) = \bar{c}_{N,i}(k) \bar{r}_{N,i}(k-1) - \bar{s}_{N,i}(k) \bar{\alpha}_N^{(i-1)*}(k)$	(9.3.7)
$\bar{\alpha}_N^{(i)*}(k) = -\bar{x}_{N,i}^{(i-1)*}(k) \bar{r}_{N,i}(k-1) + \bar{\alpha}_N^{(i-1)*}(k)$	(9.3.8)
$K_N^{\alpha(i)}(k) = \bar{c}_{N,i}(k) K_N^{\alpha(i-1)}(k)$	(9.3.9)
End for i	
$\alpha_N(k) = \bar{\alpha}_N^{(N)*}(k)$	(9.3.10)
End for k	

Из (9.130) и (9.108) $\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{R}_{N,ii}(k-1) = \lambda^{0.5} \tilde{R}_{N,ii}(k-1)$ также следует, что

$$K_{N,i}^{0.5R}(0) \bar{R}_{N,ii}(0) = K_{N,i}^{0.5R}(0) = \tilde{R}_{N,ii}(0), \quad (9.141)$$

поэтому элементы $K_{N,i}^{0.5R}(0)$ можно определить как диагональные элементы матрицы $\tilde{\mathbf{R}}_N(0) = \delta \Lambda_N^{-0.5}$, см. уравнение (8.1.0), или определить диагональную матрицу $\mathbf{K}_N^R(0)$ как

$$\mathbf{K}_N^R(0) = \begin{bmatrix} K_{N,1}^{0.5R}(0) & 0 & \dots & 0 \\ 0 & K_{N,2}^{0.5R}(0) & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & K_{N,N}^{0.5R}(0) \end{bmatrix}^2 = \delta^2 \Lambda_N^{-1}. \quad (9.142)$$



Кроме того, при рекурсивном вычислении вектора $\tilde{\mathbf{r}}_N(k)$ его начальное значение определяется как $\tilde{\mathbf{r}}_N(0)=\mathbf{0}_N$, см. уравнение (8.1.0), а согласно уравнениям (9.99) и (9.107)

$$\tilde{r}_{N,i}(k-1)=\lambda^{0.5} K_{N,i}^{0.5R}(k-1) \bar{r}_{N,i}(k-1), \quad (9.143)$$

где $K_{N,i}^{0.5R}(k-1) \neq 0$. Поэтому начальное значение $\bar{\mathbf{r}}_N(k)$ можно определить как $\bar{\mathbf{r}}_N(0)=\mathbf{0}_N$.

Поскольку согласно уравнению (9.109) $K_N^{0.5\alpha(i-1)}(k) \bar{\alpha}_N^{(i-1)*}(k)=\tilde{\alpha}_N^{(i-1)*}(k)$ и согласно (8.1.1) $\tilde{\alpha}_N^{(0)*}(k)=d^*(k)$, то

$$K_N^{0.5\alpha(0)}(k) \bar{\alpha}_N^{(0)*}(k)=\tilde{\alpha}_N^{(0)*}(k). \quad (9.144)$$

Поэтому $\bar{\alpha}_N^{(0)*}(k)=d^*(k)$ и $K_N^{0.5\alpha(0)}(k)=1$.

Оценка вычислительной сложности RLS-алгоритма в табл. 9.3 равна $O(N^2)$ арифметическим операциям. Она состоит из $1.5N^2 + 7.5N$ операций умножения, $N^2 + 2N$ операций сложения и N операций деления. Операции извлечения квадратного корня в алгоритме отсутствуют.

Структура вычислителя для реализации алгоритма (см. табл. 9.3) аналогична структуре для реализации алгоритма (см. табл. 8.1), рассмотренной в разделе 8.4.

9.6. Выводы по главе

В главе 9 рассмотрены RLS-алгоритм, использующий преобразование Хаусхолдера и содержащий всего одну операцию извлечения квадратного корня, а также RLS-алгоритмы на основе обратного и прямого QR-разложения с использованием вращений Гивенса, которые, в отличие от одноименных алгоритмов, рассмотренных в главах 7 и 8, не содержат операций извлечения квадратного корня. Исключение этой операции из указанных алгоритмов достигается соответствующим масштабированием ряда переменных, участвующих в вычислениях.

Подобным образом можно исключить операции деления, вычислительная сложность которых при их реализации в виде функционального преобразования вида $1/a$ также соизмерима со сложностью реализации операции \sqrt{a} . Это исключение имеет важное значение в алгоритмах на основе QR-разложения, так как число операций деления в них равно N , т. е. числу ВК адаптивного фильтра. Рассмотрение способа исключения операций деления из QR RLS-алгоритмов, использующих вращения Гивенса, выходит за рамки настоящей книги. Подходы к решению этой задачи могут быть найдены в [115, 116].

ГЛАВА 10

ЛИНЕЙНО-ОГРАНИЧЕННАЯ РЕКУРСИВНАЯ АДАПТИВНАЯ ФИЛЬТРАЦИЯ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ

В настоящей главе рассматриваются алгоритмы линейно-ограниченной рекурсивной адаптивной фильтрации сигналов по критерию наименьших квадратов, основанные на использовании леммы об обращении матриц. Приводится описание трех таких алгоритма с квадратичной и линейной вычислительной сложностью в части процедур линейно-ограниченной адаптивной фильтрации.

10.1. Введение

Линейно-ограниченные алгоритмы адаптивной фильтрации нашли широкое применение в адаптивных антенных и микрофонных решетках [1, 6, 58, 59, 117–123]. Кроме того, они могут применяться и в ряде других приложений, например, в задачах идентификации при наложении ограничений на значения АЧХ и (или) ФЧХ адаптивного фильтра на выбранных частотах.

На практике в качестве алгоритмов линейно-ограниченной адаптивной фильтрации в большинстве случаев используются различные варианты LC LMS-алгоритма Фроста [58], разработанного для управления ААР (см. раздел 5.7). В то же время линейно-ограниченные адаптивные алгоритмы могут строиться и на базе RLS-алгоритмов [122]. Они подобно RLS-алгоритмам без ограничений обладают лучшей сходимостью по сравнению с LC LMS-алгоритмами.

10.2. Постановка и решение линейно-ограниченной RLS-задачи адаптивной фильтрации

Рассмотренные в главах 6–9 RLS-алгоритмы адаптивной фильтрации сигналов основаны на методах безусловной оптимизации. Однако они могут строиться и на методах условной оптимизации. Переход от условной оптимизации к безусловной осуществляется с помощью множителей Лагранжа. Одноканальный

LC RLS-алгоритм рассмотрен в работе [122], лежащей в основе материалов настоящей главы.

Постановка и приемы решения задачи LC RLS-адаптивной фильтрации во многом совпадают с приемами, рассмотренными в разделах 3.7 и 5.7. Идея линейно-ограниченной адаптивной RLS-фильтрации состоит в том, что поиск минимального значения функции (6.60)

$$E_N(k) = \sum_{i=1}^k \lambda^{k-i} e_N(i) e_N^*(i) = E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) - \mathbf{h}_N^H(k) \mathbf{r}_N + \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{h}_N(k)$$

или (6.64)

$$E_N(k) = \sum_{i=1}^k \lambda^{k-i} e_N^2(i) = E_d(k) - 2\tilde{\mathbf{h}}_N^T(k) \mathbf{r}_N(k) + \tilde{\mathbf{h}}_N^T(k) \mathbf{R}_N(k) \tilde{\mathbf{h}}_N(k),$$

осуществляется при наличии линейных ограничений вида

$$\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{f}_J, \quad (10.1)$$

аналогичных ограничениям (5.176). Соотношение (10.1) должно выполняться на каждой k -й итерации, т. е. в результате работы полученного алгоритма находятся ВК адаптивного фильтра $\mathbf{h}_N(k)$, всегда удовлетворяющие этому соотношению. Уравнение, описывающее процедуру вычисления вектора $\mathbf{h}_N(k)$, можно получить аналогично уравнениям (3.87)–(3.93).

Для этого оптимизируемая функция, например (6.60), с помощью множителей Лагранжа модифицируется как

$$\begin{aligned} E_N(k) &= E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) - \mathbf{h}_N^H(k) \mathbf{r}_N + \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{h}_N(k) + \\ &+ \boldsymbol{\lambda}_J^H(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k)] + [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k)]^H \boldsymbol{\lambda}_J(k) = \\ &= E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) - \mathbf{h}_N^H(k) \mathbf{r}_N + \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{h}_N(k) + \boldsymbol{\lambda}_J^H(k) \mathbf{f}_J - \\ &- \boldsymbol{\lambda}_J^H(k) \mathbf{C}_{NJ}^H \mathbf{h}_N(k) + \mathbf{f}_J^H \boldsymbol{\lambda}_J(k) - \mathbf{h}_N^H(k) \mathbf{C}_{NJ}^H \boldsymbol{\lambda}_J(k), \end{aligned} \quad (10.2)$$

где $\boldsymbol{\lambda}_J(k) = [\lambda_1(k), \lambda_2(k), \dots, \lambda_J(k)]^T$ – вектор множителей Лагранжа.

Аналогично в случае действительных ВК, см. выражение (6.64),

$$\begin{aligned} E_N(k) &= E_d(k) - 2\mathbf{r}_N^T(k) \tilde{\mathbf{h}}_N(k) + \tilde{\mathbf{h}}_N^T(k) \mathbf{R}_N(k) \tilde{\mathbf{h}}_N(k) + 2[\mathbf{f}_J - \mathbf{C}_{NJ}^T \mathbf{h}_N(k)]^T \boldsymbol{\lambda}_J(k) = \\ &= E_d(k) - 2\mathbf{r}_N^T(k) \tilde{\mathbf{h}}_N(k) + \tilde{\mathbf{h}}_N^T(k) \mathbf{R}_N(k) \tilde{\mathbf{h}}_N(k) + 2\boldsymbol{\lambda}(k) \mathbf{f}_J^T \boldsymbol{\lambda}_J(k) - 2\tilde{\mathbf{h}}_N^T(k) \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k). \end{aligned} \quad (10.3)$$

Значение функции (10.2) минимально при

$$\nabla_{\mathbf{h}_N} E_N(k) = \frac{\partial E_N(k)}{\partial \mathbf{h}_N^*(k)} = -\mathbf{r}_N(k) + \mathbf{R}_N(k) \mathbf{h}_N(k) - \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k) = \mathbf{0}_N, \quad (10.4)$$

откуда

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) [\mathbf{r}_N(k) + \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k)]. \quad (10.5)$$

Точно так же значение функции (10.3) минимально при

$$\nabla_{\tilde{\mathbf{h}}_N} E_N(k) = \frac{\partial E_N(k)}{\partial \tilde{\mathbf{h}}_N(k)} = -2\mathbf{r}_N(k) + 2\mathbf{R}_N(k) \tilde{\mathbf{h}}_N(k) - 2\mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k) = \mathbf{0}_N, \quad (10.6)$$

откуда

$$\tilde{\mathbf{h}}_N(k) = \mathbf{R}_N^{-1}(k)[\mathbf{r}_N(k) + \mathbf{C}_{NJ}\boldsymbol{\lambda}_J(k)]. \quad (10.7)$$

Вектор $\boldsymbol{\lambda}_J(k)$ можно найти, подставив (10.5) в уравнение (10.1), т. е.

$$\mathbf{C}_{NJ}^H \mathbf{h}_N(k) = \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) + \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k) = \mathbf{f}_J, \quad (10.8)$$

откуда

$$\boldsymbol{\lambda}_J(k) = [\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)]. \quad (10.9)$$

Аналогично для действительных ВК вектор $\boldsymbol{\lambda}_J(k)$ можно найти, подставив (10.7) в уравнение (10.2), т. е.

$$\mathbf{C}_{NJ}^T \tilde{\mathbf{h}}_N(k) = \mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) + \mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ} \boldsymbol{\lambda}_J(k) = \mathbf{f}_J, \quad (10.10)$$

откуда

$$\boldsymbol{\lambda}_J(k) = [\mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)]. \quad (10.11)$$

Подставив (10.9) и (10.11) соответственно в (10.5) и (10.7), получим выражения, являющиеся решением LC RLS-задачи адаптивной фильтрации:

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) + \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)] \quad (10.12)$$

и

$$\tilde{\mathbf{h}}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) + \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)]. \quad (10.13)$$

При отсутствии сигнала $d(k)$ уравнения (10.12) и (10.13) упрощаются. Такие уравнения используются в качестве алгоритмов функционирования ААР, когда информация об источниках полезных сигналов, принимаемых антенной решеткой, задается только матрицей \mathbf{C}_{NJ} .

Инициализация матрицы $\mathbf{R}_N^{-1}(0)$ в этих алгоритмах может быть выполнена аналогично алгоритму (см. табл. 6.1) как $\delta^{-2} \mathbf{I}_N$. В качестве начального значения вектора ВК необходимо использовать значения выражений (10.12) и (10.13) при $k=0$, когда $\mathbf{r}_N(0)=\mathbf{0}_N$ (см. раздел 6.5). В этом случае

$$\begin{aligned} \mathbf{h}_N(0) &= \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \delta^{-2} \mathbf{I}_N \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \delta^{-2} \mathbf{I}_N \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \\ &= \delta^{-2} \mathbf{C}_{NJ} \delta^2 [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J \end{aligned} \quad (10.14)$$

и

$$\tilde{\mathbf{h}}_N(0) = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^T \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J. \quad (10.15)$$

То есть уравнения (10.14) и (10.15) аналогичны уравнению (5.185).



При $J=1$ матрица \mathbf{C}_{NJ} становится вектором \mathbf{c}_N и представляет собой вектор сигналов с единичными амплитудами. Если принять $f=1$ (требуемое усиление антенной решетки в направлении на источник принятого полезного сигнала), то из уравнений (10.14) и (10.15) следует, что $\mathbf{h}_N(0)=\mathbf{c}_N[\mathbf{c}_N^H\mathbf{c}_N]^{-1}f=\mathbf{c}_N[\mathbf{c}_N^H\mathbf{c}_N]^{-1}=\mathbf{c}_N N^{-1}$, так как $\mathbf{c}_N^H\mathbf{c}_N=N$. Если $f=N$, то $\mathbf{h}_N(0)=\mathbf{c}_N$.

Рассматриваемые далее вычислительные процедуры LC RLS-алгоритмов получаются путем преобразования уравнений (10.12) и (10.13). Эти преобразования базируются на использовании MIL (5.115).

10.3. Линейно-ограниченный RLS-алгоритм

Из анализа уравнений (10.12) и (10.13) следует, что первое слагаемое в них совпадает с выражениями для решений аналогичной задачи без линейных ограничений, т. е. (6.69) $\mathbf{h}_{N,unc}(k)=\mathbf{R}_N^{-1}(k)\mathbf{r}_N(k)$ и (6.71) $\tilde{\mathbf{h}}_{N,unc}(k)=\mathbf{R}_N^{-1}(k)\mathbf{r}_N(k)$, а второе слагаемое обусловлено наличием ограничений (10.1), так как содержит матрицу \mathbf{C}_{NJ} и вектор \mathbf{f}_J . Здесь нижний индекс *unc* означает «unconstrained» — «неограниченный» или «без ограничений». Второе слагаемое корректирует вектор $\mathbf{h}_{N,unc}(k)$ так, что в конечном итоге суммарный вектор $\mathbf{h}_N(k)$ удовлетворяет условию (10.1).

В главе 6 было показано, что вектор $\mathbf{h}_{N,unc}(k)$ можно вычислять рекурсивно (см. табл. 6.1) путем использования уравнений вычисления векторов коэффициентов Калмана $\mathbf{g}_N(k)$ (6.77), матрицы $\mathbf{R}_N^{-1}(k)$ (6.76), а также уравнений (6.1.3) и (6.1.4), см. табл. 6.1, $\alpha_{N,unc}(k)=d(k)-\mathbf{h}_{N,unc}^H(k-1)\mathbf{x}_N(k)$ и $\mathbf{h}_{N,unc}(k)=\mathbf{h}_{N,unc}(k-1)+\mathbf{g}_N(k)\alpha_{N,unc}^*(k)$. Вектор $\mathbf{g}_N(k)$ может вычисляться с помощью уравнения (6.77) или других математически эквивалентных уравнений из ранее рассмотренных RLS- и быстрых RLS-алгоритмов, которые будут рассмотрены далее.

Для рекурсивного вычисления с помощью уравнений (10.12) и (10.13) в дополнение к уравнениям табл. 6.1 необходимо получить рекурсивные уравнения для вычисления переменных, фигурирующих в выражениях (10.12) и (10.13),

$$\Gamma_{NJ}(k)=\mathbf{R}_N^{-1}(k)\mathbf{C}_{NJ}, \quad (10.16)$$

$$\Psi_J^{-1}(k)=\left[\mathbf{C}_{NJ}^H\mathbf{R}_N^{-1}(k)\mathbf{C}_{NJ}\right]^{-1}=\left[\mathbf{C}_{NJ}^H\Gamma_{NJ}(k)\right]^{-1}. \quad (10.17)$$

Для рекурсивного вычисления переменной $\Gamma_{NJ}(k)$ необходимо воспользоваться уравнением (6.76) $\mathbf{R}_N^{-1}(k)=\lambda^{-1}\left[\mathbf{R}_N^{-1}(k-1)-\mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\right]$. С учетом этого уравнения матрица $\Gamma_{NJ}(k)$ определяется как

$$\begin{aligned} \Gamma_{NJ}(k) &= \mathbf{R}_N^{-1}(k)\mathbf{C}_{NJ} = \lambda^{-1}\left[\mathbf{R}_N^{-1}(k-1)-\mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\right]\mathbf{C}_{NJ} = \\ &= \lambda^{-1}\mathbf{R}_N^{-1}(k-1)\mathbf{C}_{NJ}-\mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{C}_{NJ} = \\ &= \lambda^{-1}\left[\Gamma_{NJ}(k-1)-\mathbf{g}_N(k)\mathbf{x}_N^H(k)\Gamma_{NJ}(k-1)\right]. \end{aligned} \quad (10.18)$$

Для рекурсивного вычисления обратной матрицы $\Psi_J^{-1}(k)$ используется MIL (5.115), уравнение (10.18) и матрица $\Psi_J(k)$, преобразованная как

$$\begin{aligned}\Psi_J(k) &= \left[\mathbf{C}_{NJ}^H \Gamma_{NJ}(k) \right] = \mathbf{C}_{NJ}^H \left\{ \lambda^{-1} \left[\Gamma_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \Gamma_{NJ}(k-1) \right] \right\} = \\ &= \lambda^{-1} \left[\mathbf{C}_{NJ}^H \Gamma_{NJ}(k-1) - \mathbf{C}_{NJ}^H \mathbf{g}_N(k) \mathbf{x}_N^H(k) \Gamma_{NJ}(k-1) \right] = \lambda^{-1} \left[\Psi(k-1) - \mathbf{v}_J(k) \boldsymbol{\eta}_J^H(k) \right],\end{aligned}\quad (10.19)$$

где

$$\mathbf{v}_J(k) = \mathbf{C}_{NJ}^H \mathbf{g}_N(k) \quad (10.20)$$

и

$$\boldsymbol{\eta}_J^H(k) = \mathbf{x}_N^H(k) \Gamma_{NJ}(k-1). \quad (10.21)$$

Для обращения матрицы $\Psi_J(k)$ с помощью MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, используя уравнение (10.19), необходимо определить переменные $\mathbf{C} = \Psi_J(k-1)$, $\mathbf{C}^{-1} = \Psi_J^{-1}(k-1)$, $\mathbf{B}^H = \mathbf{v}_J(k)$, $\mathbf{D} = \boldsymbol{\eta}_J^H(k)$ и $\mathbf{A}^{-1} = 1$.

Тогда матрицу $\Psi_J^{-1}(k)$ можно представить как

$$\begin{aligned}\Psi_J^{-1}(k) &= \lambda \left[\Psi_J^{-1}(k-1) + \frac{\Psi_J^{-1}(k-1) \mathbf{v}_J(k) \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1)}{1 - \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k)} \right] = \\ &= \lambda \left[\Psi_J^{-1}(k-1) + \mathbf{l}_J(k) \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \right],\end{aligned}\quad (10.22)$$

где

$$\mathbf{l}_J(k) = \frac{\Psi_J^{-1}(k-1) \mathbf{v}_J(k)}{1 - \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k)}. \quad (10.23)$$

Таким образом, уравнение (10.22) является аналогичным уравнению рекурсивного вычисления $\mathbf{R}_N^{-1}(k)$ (6.76), а вектор $\mathbf{l}_J(k)$ аналогичен вектору Калмана $\mathbf{g}_N(k)$ (6.77). Поэтому, используя преобразования, аналогичные (6.78), можно показать, что

$$\begin{aligned}\mathbf{l}_J(k) \left[1 - \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k) \right] &= \\ &= \mathbf{l}_J(k) - \mathbf{l}_J(k) \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k) = \Psi_J^{-1}(k-1) \mathbf{v}_J(k)\end{aligned}\quad (10.24)$$

или

$$\begin{aligned}\mathbf{l}_J(k) &= \Psi_J^{-1}(k-1) \mathbf{v}_J(k) + \mathbf{l}_J(k) \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k) = \\ &= \left[\Psi_J^{-1}(k-1) + \mathbf{l}_J(k) \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \right] \mathbf{v}_J(k) = \lambda^{-1} \Psi_J^{-1}(k) \mathbf{v}_J(k).\end{aligned}\quad (10.25)$$

Поскольку вектор $\mathbf{v}_J(k)$ содержит вектор $\mathbf{g}_N(k)$, то

$$\mathbf{l}_J(k) = \frac{\Psi_J^{-1}(k-1) \mathbf{v}_J(k)}{1 - \boldsymbol{\eta}_J^H(k) \Psi_J^{-1}(k-1) \mathbf{v}_J(k)} = \frac{\Psi_J^{-1}(k-1) \mathbf{C}_{NJ}^H \mathbf{g}_N(k)}{1 - \mathbf{x}_N^H(k) \Gamma_{NJ}(k-1) \Psi_J^{-1}(k-1) \mathbf{C}_{NJ}^H \mathbf{g}_N(k)}, \quad (10.26)$$

т. е. переменная $\mathbf{l}_J(k)$ вычисляется через вектор коэффициентов Калмана (переменную RLS-алгоритма без ограничений) и вектор входных сигналов адаптивного фильтра $\mathbf{x}_N(k)$.



Теперь переменные $\Gamma_{NJ}(k)$ и $\Psi_J^{-1}(k)$ можно использовать в уравнении (10.12) для получения рекурсивной процедуры вычисления вектора ВК. С учетом введенных переменных

$$\mathbf{h}_N(k) = \mathbf{h}_{N,unc}(k) + \Gamma_{NJ}(k) \Psi_J^{-1}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k)]. \quad (10.27)$$

Используя выражение (6.1.4) $\mathbf{h}_{N,unc}(k) = \mathbf{h}_{N,unc}(k-1) + \mathbf{g}_N(k) \alpha_{N,unc}^*(k)$, уравнение (10.27) преобразуем как

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_{N,unc}(k-1) + \mathbf{g}_N(k) \alpha_{N,unc}^*(k) + \\ &+ \Gamma_{NJ}(k) \Psi_J^{-1}(k) \{ \mathbf{f}_J - \mathbf{C}_{NJ}^H [\mathbf{h}_{N,unc}(k-1) + \mathbf{g}_N(k) \alpha_{N,unc}^*(k)] \} = \\ &= \mathbf{h}_{N,unc}(k-1) + \Gamma_{NJ}(k) \Psi_J^{-1}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] + \\ &+ \mathbf{g}_N(k) \alpha_{N,unc}^*(k) - \Gamma_{NJ}(k) \Psi_J^{-1}(k) \mathbf{C}_{NJ}^H \mathbf{g}_N(k) \alpha_{N,unc}^*(k). \end{aligned} \quad (10.28)$$

Затем, используя уравнения (10.18) и (10.22), преобразуем выражение $\Gamma_{NJ}(k) \Psi_J^{-1}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)]$ в уравнении (10.28) как

$$\begin{aligned} \Gamma_{NJ}(k) \Psi_J^{-1}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] &= \lambda^{-1} [\Gamma_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{\eta}_J^H(k)] \times \\ &\times \lambda [\Psi_J^{-1}(k-1) + \mathbf{l}_J(k) \mathbf{\eta}_J^H(k) \Psi_J^{-1}(k-1)] [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] = \\ &= \Gamma_{NJ}(k-1) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] - \\ &- \mathbf{g}_N(k) \mathbf{\eta}_J^H(k) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] + \\ &+ \Gamma_{NJ}(k-1) \mathbf{l}_J(k) \mathbf{\eta}_J^H(k) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] - \\ &- \mathbf{g}_N(k) \mathbf{\eta}_J^H(k) \mathbf{l}_J(k) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)]. \end{aligned} \quad (10.29)$$

Скомбинируем первое слагаемое уравнения (10.28) и первое слагаемое уравнения (10.29). Тогда согласно уравнению (10.27)

$$\mathbf{h}_{N,unc}(k-1) + \Gamma_{NJ}(k-1) [\mathbf{C}_{NJ}^H \Gamma_{NJ}(k-1)]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] = \mathbf{h}_N(k-1). \quad (10.30)$$

Далее скомбинируем второе слагаемое уравнения (10.28) и второе слагаемое уравнения (10.29), т. е.

$$\begin{aligned} \mathbf{g}_N(k) \alpha_{N,unc}^*(k) - \mathbf{g}_N(k) \mathbf{\eta}_J^H(k) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] &= \\ &= \mathbf{g}_N(k) \{ [\mathbf{d}(k) - \mathbf{h}_{N,unc}^H(k-1) \mathbf{x}_N(k)]^* - \\ &- \mathbf{\eta}_J^H(k) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] \} = \mathbf{g}_N(k) \{ \mathbf{d}^*(k) - \\ &- \mathbf{x}_N^H(k) \mathbf{h}_{N,unc}(k-1) - \mathbf{x}_N^H(k) \Gamma_{NJ}(k-1) \Psi_J^{-1}(k-1) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)] \} = \\ &= \mathbf{g}_N(k) \{ \mathbf{d}^*(k) - \mathbf{x}_N^H(k) (\mathbf{h}_{N,unc}(k-1) - \Gamma_{NJ}(k-1) \Psi_J^{-1}(k-1) \times \\ &\times [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)]) \} = \mathbf{g}_N(k) [\mathbf{d}^*(k) - \mathbf{x}_N^H(k) \mathbf{h}_N(k-1)] = \mathbf{g}_N(k) \alpha_N^*(k). \end{aligned} \quad (10.31)$$

Затем, используя определение (10.18), скомбинируем два последних слагаемых в уравнении (10.29), т. е.

$$\begin{aligned}
& \Gamma_{NJ}(k-1)\mathbf{I}_J(k)\boldsymbol{\eta}_J^H(k)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right] - \\
& - \mathbf{g}_N(k)\boldsymbol{\eta}_J^H(k)\mathbf{I}_J(k)\boldsymbol{\eta}_J^H(k)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right] = \\
& = \left[\Gamma_{NJ}(k-1) - \mathbf{g}_N(k)\boldsymbol{\eta}_J^H(k)\right]\mathbf{I}_J(k)\boldsymbol{\eta}_J^H(k)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right] = \\
& = \lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\boldsymbol{\eta}_J^H(k)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right].
\end{aligned} \tag{10.32}$$

Используя выражение (10.20) $\mathbf{v}_J(k) = \mathbf{C}_{NJ}^H \mathbf{g}_N(k)$ и (10.25) $\mathbf{I}_J(k) = \lambda^{-1} \boldsymbol{\Psi}_J^{-1}(k) \mathbf{v}_J(k)$, последнее слагаемое в уравнении (10.28) можно представить как

$$\begin{aligned}
& -\Gamma_{NJ}(k)\boldsymbol{\Psi}_J^{-1}(k)\mathbf{C}_{NJ}^H \mathbf{g}_N(k)\alpha_{N,unc}^*(k) = \\
& = -\Gamma_{NJ}(k)\boldsymbol{\Psi}_J^{-1}(k)\mathbf{v}_J(k)\alpha_{N,unc}^*(k) = -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\alpha_{N,unc}^*(k).
\end{aligned} \tag{10.33}$$

Теперь скомбинируем уравнения (10.33) и (10.32), т. е.

$$\begin{aligned}
& -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\alpha_{N,unc}^*(k) + \lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\boldsymbol{\eta}_J^H(k)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right] = \\
& -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\left\{\alpha_{N,unc}^*(k) - \mathbf{x}_N^H(k)\Gamma_{NJ}(k-1)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right]\right\} = \\
& = -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\left\{d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_{N,unc}(k-1) - \right. \\
& \left. - \mathbf{x}_N^H(k)\Gamma_{NJ}(k-1)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right]\right\} = \\
& -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\left\{d^*(k) - \mathbf{x}_N^H(k)\left(\mathbf{h}_{N,unc}(k-1) - \right. \right. \\
& \left. \left. - \Gamma_{NJ}(k-1)\boldsymbol{\Psi}_J^{-1}(k-1)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_{N,unc}(k-1)\right]\right)\right\} = \\
& = -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\left[d^*(k) - \mathbf{x}_N^H(k)\mathbf{h}_N(k-1)\right] = -\lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\alpha_N^*.
\end{aligned} \tag{10.34}$$

И наконец, используя выражение (10.30), (10.31) и (10.34), уравнение (10.27), а значит, и уравнение (10.12) можно представить как

$$\begin{aligned}
\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k) - \lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\alpha_N^*(k) = \\
& = \mathbf{h}_N(k-1) + [\mathbf{g}_N(k) - \lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)]\alpha_N^*(k).
\end{aligned} \tag{10.35}$$

Таким образом, (10.35) представляет собой рекурсивное уравнение для вычисления вектора ВК адаптивного фильтра на базе LC RLS-алгоритма. В нем $\mathbf{h}_N(k-1)$, $\Gamma_{NJ}(k)$, $\mathbf{I}_J(k)$, а $\alpha_N^*(k)$ — переменные, возникающие в линейно-ограниченной адаптивной фильтрации, а вектор $\mathbf{g}_N(k)$ — переменная, возникающая в адаптивной фильтрации без ограничений. Алгоритм на основании уравнения (10.35) приведен в табл. 10.1.

В табл. 10.1, а также далее в табл. 10.2 и табл. 10.3 настоящей главы, в качестве процедуры вычисления вектора $\mathbf{g}_N(k)$ предлагается использовать уравнения (6.1.1) и (6.1.2) RLS-алгоритма без линейных ограничений (см. табл. 6.1). Поэтому начальное значение матрицы $\mathbf{R}_N^{-1}(0)$ в алгоритме (см. табл. 10.1) задается так же, как в табл. 6.1. Чтобы вектор $\mathbf{g}_N(k)$ был математически эквивалентным вектором коэффициентов Калмана, вычисляемым с помощью алгоритмов из разделов 7—9 и 11, инициализацию матрицы $\mathbf{R}_N^{-1}(0)$ в алгоритмах (см. табл. 6.1 и табл. 10.1—10.3) необходимо выполнять как $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N$.



Таблица 10.1. LC RLS-алгоритм

Вычисления	Ссылки
Initialization : $\mathbf{x}_N(0) = \mathbf{0}_N, \mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{I}_N, \mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}$	(10.1.0)
$\mathbf{\Psi}_J^{-1}(0) = [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1}, \mathbf{h}_N(0) = \mathbf{\Gamma}_{NJ}(0) \mathbf{\Psi}_J^{-1}(0) \mathbf{f}_J$	
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_N(k)$, см., например, (6.1.1), (6.1.2) в табл. 6.1	(10.1.1)
$\mathbf{v}_J(k) = \mathbf{C}_{NJ}^H \mathbf{g}_N(k)$	(10.1.2)
$\mathbf{\eta}_J^H(k) = \mathbf{x}_N^H(k) \mathbf{\Gamma}_{NJ}(k-1)$	(10.1.3)
$\mathbf{l}_J(k) = \frac{\mathbf{\Psi}_J^{-1}(k-1) \mathbf{v}_J(k)}{1 - \mathbf{\eta}_J^H(k) \mathbf{\Psi}_J^{-1}(k-1) \mathbf{v}_J(k)}$	(10.1.4)
$\mathbf{\Psi}_J^{-1}(k) = \lambda [\mathbf{\Psi}_J^{-1}(k-1) + \mathbf{l}_J(k) \mathbf{\eta}_J^H(k) \mathbf{\Psi}_J^{-1}(k-1)]$	(10.1.5)
$\mathbf{\Gamma}_{NJ}(k) = \lambda^{-1} [\mathbf{\Gamma}_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{\eta}_J^H(k)]$	(10.1.6)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(10.1.7)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + [\mathbf{g}_N(k) - \lambda \mathbf{\Gamma}_{NJ}(k) \mathbf{l}_J(k)] \alpha_N^*(k)$	(10.1.8)
End for k	

10.4. Двойное ограничение в RLS-алгоритме

В алгоритме (см. табл. 10.1), в отличие от LC LMS-алгоритма Фроста [58], ограничение в векторе ВК адаптивного фильтра присутствует только при инициализации, см. уравнение (10.1.0). Поэтому в силу погрешностей вычислений со временем вектор ВК $\mathbf{h}_N(k-1)$, а значит, и вектор $\mathbf{h}_N(k)$, может перестать удовлетворять условиям линейных ограничений (10.1). Чтобы вектор ВК, вычисляемых с помощью алгоритма (см. табл. 10.1), удовлетворял условиям (10.1), в нем ограничения должны присутствовать на каждой итерации, как это имеет место в уравнении (10.12).

Отметим, что в уравнении (10.12) второе слагаемое, по сути, является элементом, корректирующим значение вектора ВК $\mathbf{h}_{N,unc}(k)$. Этот элемент пропорционален величине $\mathbf{f}_J - \mathbf{C}_{NJ}^T \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k) = \mathbf{f}_J - \mathbf{C}_{NJ}^T \mathbf{h}_{N,unc}(k)$. Его влияние уменьшается по мере того, как $\mathbf{h}_{N,unc}(k) \rightarrow \mathbf{h}(k)$, а значит, выполняется условие (10.1).

Для обеспечения условия (10.1) в [122] корректирующий элемент предлагается применить **еще раз**, но уже к вектору $\mathbf{h}_N(k-1)$ в уравнении (10.35), так как в нем условие (10.1) в явном виде отсутствует. В этом и заключается суть двойного (или повторного) ограничения.

Влияние корректирующего члена $\mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)]$ будет уменьшаться по мере того, как вектор $\mathbf{h}_N(k-1)$ будет удовлетворять условию ограничения, т. е. множитель $[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)] \rightarrow \mathbf{0}_J$. И наоборот, при $[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)] \neq \mathbf{0}_J$

корректирующий член $\Gamma_{NJ}(k)\Psi_J^{-1}(k)[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)]$ будет изменять $\mathbf{h}_N(k)$ так, чтобы условие (10.1) выполнялось.

Прежде чем применить предлагаемую коррекцию, используя ранее полученное соотношение $\mathbf{I}_J(k) = \lambda^{-1}\Psi_J^{-1}(k)\mathbf{v}_J(k)$, преобразуем уравнение (10.35) как

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k) - \lambda\Gamma_{NJ}(k)\mathbf{I}_J(k)\alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k) - \lambda\Gamma_{NJ}(k)[\lambda^{-1}\Psi_J^{-1}(k)\mathbf{v}_J(k)]\alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k) - \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{C}_{NJ}^H \mathbf{g}_N(k)\alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + [\mathbf{I}_N - \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{C}_{NJ}^H] \mathbf{g}_N(k)\alpha_N^*(k) = \\ &= \mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k), \end{aligned} \quad (10.36)$$

где

$$\mathbf{P}_N(k) = \mathbf{I}_N - \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{C}_{NJ}^H \quad (10.37)$$

— проекционная матрица, аналогичная матрице (5.184).

Далее, по аналогии с уравнением (10.27), в уравнение (10.36) добавим корректирующий член $\Gamma_{NJ}(k)\Psi_J^{-1}(k)[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)]$. Тогда

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k) + \\ &+ \Gamma_{NJ}(k)\Psi_J^{-1}(k)[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N(k-1)] = \\ &= \mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k) + \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{f}_J - \\ &- \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{C}_{NJ}^H \mathbf{h}_N(k-1) = \\ &= [\mathbf{I}_N - \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{C}_{NJ}^H] \mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k) + \\ &+ \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{f}_J = \mathbf{P}_N(k)\mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k) + \\ &+ \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{f}_J = \mathbf{P}_N(k)[\mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k)] + \mathbf{a}_N(k), \end{aligned} \quad (10.38)$$

где

$$\mathbf{a}_N(k) = \Gamma_{NJ}(k)\Psi_J^{-1}(k)\mathbf{f}_J. \quad (10.39)$$

Таким образом, введение корректирующего члена позволяет получить линейно-ограниченный алгоритм (см. табл. 10.2), отличающийся от алгоритма (см. табл. 10.1) лишь уравнением вычисления вектора ВК (10.38).

Уравнения (10.38) и (10.2.10) в некоторой степени схожи с уравнением (5.183) алгоритма [58]. Однако, в отличие от LC LMS-алгоритма [58], в LC RLS-алгоритме матрица $\mathbf{P}_N(k)$ и вектор $\mathbf{a}_N(k)$, обеспечивающие выполнение условия (10.1), не фиксированы, а зависят от входных сигналов. Кроме того, на свойства RLS-алгоритмов аналогичным образом влияет вектор $\mathbf{g}_N(k)$, который в LMS-алгоритмах фактически заменяется на более простой параметр $\mu \mathbf{x}_N(k)$, где μ — фиксированный шаг сходимости градиентного алгоритма, см. уравнения (6.105) и (6.106). Это является причиной того, что, подобно одноименным алгоритмам без ограничений, LC RLS-алгоритмы более эффективны с точки зрения скорости сходимости по сравнению с LC LMS-алгоритмами.

Таблица 10.2. LC RLS-алгоритм с двойным ограничением

Вычисления	Ссылки
Initialization : $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{I}_N$, $\mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}$, $\mathbf{\Psi}_J^{-1}(0) = [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1}$, $\mathbf{h}_N(0) = \mathbf{\Gamma}_{NJ}(0) \mathbf{\Psi}_J^{-1}(0) \mathbf{f}_J$	(10.2.0)
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_N(k)$, см., например, (6.1.1), (6.1.2), в табл. 6.1	(10.2.1)
$\mathbf{v}_J(k) = \mathbf{C}_{NJ}^H \mathbf{g}_N(k)$	(10.2.2)
$\mathbf{\eta}_J^H(k) = \mathbf{x}_N^H(k) \mathbf{\Gamma}_{NJ}(k-1)$	(10.2.3)
$\mathbf{l}_J(k) = \frac{\mathbf{\Psi}_J^{-1}(k-1) \mathbf{v}_J(k)}{1 - \mathbf{\eta}_J^H(k) \mathbf{\Psi}_J^{-1}(k-1) \mathbf{v}_J(k)}$	(10.2.4)
$\mathbf{\Psi}_J^{-1}(k) = \lambda [\mathbf{\Psi}_J^{-1}(k-1) + \mathbf{l}_J(k) \mathbf{\eta}_J^H(k) \mathbf{\Psi}_J^{-1}(k-1)]$	(10.2.5)
$\mathbf{\Gamma}_{NJ}(k) = \lambda^{-1} [\mathbf{\Gamma}_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{\eta}_J^H(k) \mathbf{\Psi}_J^{-1}(k-1)]$	(10.2.6)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(10.2.7)
$\mathbf{P}_N(k) = \mathbf{I}_N - \mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k) \mathbf{C}_{NJ}^H$	(10.2.8)
$\mathbf{a}_N(k) = \mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k) \mathbf{f}_J$	(10.2.9)
$\mathbf{h}_N(k) = \mathbf{P}_N(k) [\mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)] + \mathbf{a}_N(k)$	(10.2.10)
End for k	

Из-за умножений на матрицу $\mathbf{P}_N(k)$ оценка вычислительной сложности уравнения (10.38) равна $O(N^2)$ арифметическим операциям. Эта величина является соизмеримой со сложностью выражений для вычисления векторов Калмана $\mathbf{g}_N(k)$ с помощью соответствующих процедур RLS-алгоритмов без ограничений. Однако в случае использования быстрых (вычислительно-эффективных) RLS-алгоритмов (см. раздел 11), в которых вектор $\mathbf{g}_N(k)$ вычисляется со сложностью $O(N)$, сложность всех шагов LC RLS-алгоритмов должна быть тоже равна $O(N)$. Поэтому алгоритм (см. табл. 10.2) желательно преобразовать в математически эквивалентный алгоритм с меньшей, по возможности линейной, т. е. $O(N)$, вычислительной сложностью. Такой алгоритм представлен в следующем разделе.

10.5. RLS-алгоритм с двойным ограничением и линейной вычислительной сложностью

Уменьшить вычислительную сложность алгоритма (см. табл. 10.2) можно следующим образом. Если обозначить

$$\mathbf{Q}_{NJ}(k) = \mathbf{\Gamma}_{NJ}(k) \mathbf{\Psi}_J^{-1}(k), \quad (10.40)$$

$$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k), \quad (10.41)$$

то уравнение (10.38) можно преобразовать как

$$\begin{aligned} \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{P}_N(k)\mathbf{g}_N(k)\alpha_N^*(k) + \mathbf{\Gamma}_{NJ}(k)\mathbf{\Psi}_J^{-1}(k)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H\mathbf{h}_N(k-1)\right] = \\ &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k) - \mathbf{\Gamma}_{NJ}(k)\mathbf{\Psi}_J^{-1}(k)\mathbf{C}_{NJ}^H\mathbf{g}_N(k)\alpha_N^*(k) + \\ &+ \mathbf{\Gamma}_{NJ}(k)\mathbf{\Psi}_J^{-1}(k)\mathbf{f}_J - \mathbf{\Gamma}_{NJ}(k)\mathbf{\Psi}_J^{-1}(k)\mathbf{C}_{NJ}^H\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \\ &+ \mathbf{g}_N(k)\alpha_N^*(k) - \mathbf{Q}_{NJ}(k)\mathbf{C}_{NJ}^H\mathbf{g}_N(k)\alpha_N^*(k) + \mathbf{Q}_{NJ}(k)\mathbf{f}_J - \\ &- \mathbf{Q}_{NJ}(k)\mathbf{C}_{NJ}^H\mathbf{h}_N(k-1) = \left[\mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k)\right] + \\ &+ \mathbf{Q}_{NJ}(k)\left\{\mathbf{f}_J - \mathbf{C}_{NJ}^H\left[\mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k)\right]\right\} = \\ &= \mathbf{h}'_N(k) + \mathbf{Q}_{NJ}(k)\left[\mathbf{f}_J - \mathbf{C}_{NJ}^H\mathbf{h}'_N(k)\right]. \end{aligned} \quad (10.42)$$

Далее необходимо получить рекурсивную процедуру вычисления матрицы $\mathbf{Q}_{NJ}(k)$. Для этого в уравнение (10.40) нужно подставить (10.18) и (10.22) как

$$\begin{aligned} \mathbf{Q}_{NJ}(k) &= \mathbf{\Gamma}_{NJ}(k)\mathbf{\Psi}_J^{-1}(k) = \lambda^{-1}\left[\mathbf{\Gamma}_{NJ}(k-1) - \mathbf{g}_N(k)\mathbf{\eta}_J^H(k)\right] \times \\ &\times \lambda \left[\mathbf{\Psi}_J^{-1}(k-1) + \frac{\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)\mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)}{1 - \mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)}\right] = \\ &= \mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{\eta}_J^H(k)\mathbf{\Psi}(k-1) + \\ &+ \mathbf{\Gamma}_{NJ}(k-1)\frac{\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)\mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)}{1 - \mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)} - \\ &- \mathbf{g}_N(k)\mathbf{\eta}_J^H(k)\frac{\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)\mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)}{1 - \mathbf{\eta}_J^H(k)\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)} = \\ &= \mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1) + \\ &+ \mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1)\frac{\mathbf{v}_J(k)\mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1)}{1 - \mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)} - \\ &- \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}(k-1)\frac{\mathbf{v}_J(k)\mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1)}{1 - \mathbf{x}_N^H(k)\mathbf{\Gamma}_{NJ}(k-1)\mathbf{\Psi}_J^{-1}(k-1)\mathbf{v}_J(k)} = \\ &= \mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k)\mathbf{v}_J^H(k) + \mathbf{Q}_{NJ}(k-1)\frac{\mathbf{v}_J(k)\mathbf{v}_J^H(k)}{1 - \mathbf{v}_J^H(k)\mathbf{v}_J(k)} - \\ &- \mathbf{g}_N(k)\mathbf{v}_J^H(k)\frac{\mathbf{v}_J(k)\mathbf{v}_J^H(k)}{1 - \mathbf{x}_N^H(k)\mathbf{v}_J^H(k)} = \left[\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k)\mathbf{v}_J^H(k)\right] + \\ &+ \left[\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k)\mathbf{v}_J^H(k)\right]\frac{\mathbf{v}_J(k)\mathbf{v}_J^H(k)}{1 - \mathbf{v}_J^H(k)\mathbf{v}_J(k)} = \\ &= \left[\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k)\mathbf{v}_J^H(k)\right]\left[\mathbf{I}_J + \frac{\mathbf{v}_J(k)\mathbf{v}_J^H(k)}{1 - \mathbf{v}_J^H(k)\mathbf{v}_J(k)}\right], \end{aligned} \quad (10.43)$$

где

$$\mathbf{v}_J^H(k) = \mathbf{x}_N^H(k)\mathbf{Q}_{NJ}(k-1). \quad (10.44)$$

Легко заметить, см. уравнения (10.17) и (10.40), что

$$\mathbf{C}_{NJ}^H \mathbf{Q}_{NJ}(k) = \mathbf{C}_{NJ}^H [\Gamma_{NJ}(k) \Psi_J^{-1}(k)] = [\mathbf{C}_{NJ}^H \Gamma_{NJ}(k)] \Psi_J^{-1}(k) = \Psi_J(k) \Psi_J^{-1}(k) = \mathbf{I}_J. \quad (10.45)$$

Выражение (10.45) формально можно рассматривать как линейное ограничение, аналогичное (10.1). Его можно использовать в уравнении (10.43) для стабилизации вычислений.

Для этого обозначим уравнение (10.43) как

$$\mathbf{Q}'_{NJ}(k) = [\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{v}_J^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_J(k) \mathbf{v}_J^H(k)}{1 - \mathbf{v}_J^H(k) \mathbf{v}_J(k)} \right], \quad (10.46)$$

т. е. как результат, содержащий накопленные ошибки округления.

Для коррекции этого результата составим градиентное уравнение

$$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) - \nabla_{\mathbf{Q}_{NJ}^*}(k), \quad (10.47)$$

где

$$\nabla_{\mathbf{Q}_{NJ}^*}(k) = \frac{\partial \mathbf{G}_J(k)}{\partial \mathbf{Q}_{NJ}^*} \quad (10.48)$$

и

$$\mathbf{G}_J(k) = \Lambda_J^H(k) [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}_{NJ}(k)] + [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}_{NJ}(k)]^H \Lambda_J(k), \quad (10.49)$$

а $\Lambda_J(k)$ — матрица множителей Лагранжа. Каждый столбец этой матрицы отвечает за одно значение ограничивающего параметра, определяемого одним диагональным элементом единичной матрицы \mathbf{I}_J .

Так как согласно уравнению (10.49)

$$\frac{\partial \mathbf{G}_J(k)}{\partial \mathbf{Q}_{NJ}^*(k)} = -\mathbf{C}_{NJ} \Lambda_J(k), \quad (10.50)$$

то

$$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ} \Lambda_J(k). \quad (10.51)$$

Используя (10.51) в уравнении (10.45), т. е.

$$\mathbf{C}_{NJ}^H \mathbf{Q}_{NJ}(k) = \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \Lambda_J(k) = \mathbf{I}_J, \quad (10.52)$$

из уравнения (10.52) можно определить матрицу $\Lambda_J(k)$ как

$$\Lambda_J(k) = [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ}(k)] \quad (10.53)$$

и использовать ее в уравнении (10.51). Тогда

$$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ}(k)]. \quad (10.54)$$

Таким образом, (10.54) является уравнением для линейно-ограниченного вычисления матрицы $\mathbf{Q}_{NJ}(k)$. Это уравнение, а также уравнения (10.41) — (10.43)

представляют собой версию LC RLS-алгоритма с двойным ограничением, вычислительная процедура которого приведена в табл. 10.3.

Таблица 10.3. LC RLS-алгоритм с двойным ограничением

Вычисления	Ссылки
Initialization : $\mathbf{x}_N(0), \mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N, \mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ},$	(10.3.0)
$\mathbf{Q}_{NJ}(0) = \mathbf{\Gamma}_{NJ}(0) [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1}, \mathbf{h}_N(0) = \mathbf{Q}_{NJ}(0) \mathbf{f}_J$	
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_N(k)$, см., например, (6.1.1), (6.1.2), в табл. 6.1	(10.3.1)
$\mathbf{v}_J(k) = \mathbf{C}_{NJ}^H \mathbf{g}_N(k)$	(10.3.2)
$\mathbf{v}_J^H(k) = \mathbf{x}_N^H(k) \mathbf{Q}_{NJ}(k-1)$	(10.3.3)
$\mathbf{Q}'_{NJ}(k) = [\mathbf{Q}_{NJ}(k-1) - \mathbf{g}_N(k) \mathbf{v}_J^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_J(k) \mathbf{v}_J^H(k)}{1 - \mathbf{v}_J^H(k) \mathbf{v}_J(k)} \right]$	(10.3.4)
$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ}(k)]$	(10.3.5)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(10.3.6)
$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$	(10.3.7)
$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k)]$	(10.3.8)
End for k	

Из уравнений (10.40)–(10.42) следует, что начальное значение вектора ВК в алгоритме (см. табл. 10.3) определяется как $\mathbf{h}_N(0) = \mathbf{Q}_{NJ}(0) \mathbf{f}_J$. Легко заметить, что оно совпадает с (10.14).

Таким образом, линейно-ограниченный алгоритм (см. табл. 10.3) является результатом последовательных эквивалентных преобразований алгоритмов (см. табл. 10.1 и табл. 10.2). Сложность алгоритма (см. табл. 10.3) без учета вычисления векторов Калмана $\mathbf{g}_N(k)$ оценивается как $O(N)$ арифметических операций (комплексных или действительных, в зависимости от арифметики) и является наименьшей среди трех рассмотренных алгоритмов. Это позволяет использовать алгоритм (см. табл. 10.3) как базовый при реализации различных линейно-ограниченных алгоритмов, многообразие которых обусловлено различными процедурами вычисления векторов Калмана, заимствованными из RLS-алгоритмов разделов 6–9, а также быстрых RLS-алгоритмов из раздела 11.

Вычислительная сложность алгоритма (см. табл. 10.3) в части, связанной с линейными ограничениями, равна $3NJ^2 + 5NJ + J^2 + J + 2N$ операциям умножения, $3NJ^2 + 6NJ + 4J + 3N$ операциям сложения и одной операции деления. Из этих оценок следует, что вычислительная сложность данной части алгорит-



ма пропорциональна $O(N)$, т. е. при использовании быстрых процедур вычисления векторов Калмана $\mathbf{g}_N(k)$ достигается линейная вычислительная сложность этого алгоритма.

Структура матрицы \mathbf{C}_{Nj} и вектора \mathbf{f}_j зависит от вида задачи, решаемой с помощью линейно-ограниченного адаптивного фильтра. Примеры определения матрицы \mathbf{C}_{Nj} и вектора \mathbf{f}_j были рассмотрены в разделе 3.7.

10.6. Выводы по главе

В главе 10 определены постановка задачи LC RLS-адаптивной фильтрации и ее решение в виде трех различных адаптивных алгоритмов. Эти алгоритмы могут быть использованы в ряде известных приложений линейно-ограниченной адаптивной фильтрации, в частности в ААР, как альтернатива алгоритму [58].

ГЛАВА II

БЫСТРЫЕ РЕКУРСИВНЫЕ АДАПТИВНЫЕ АЛГОРИТМЫ ПО КРИТЕРИЮ НАИМЕНЬШИХ КВАДРАТОВ

В настоящей главе рассматриваются быстрые, т.е. вычислительно эффективные, RLS-алгоритмы адаптивной фильтрации, основанные на использовании лемм об обращении клеточных матриц. Эти алгоритмы являются результатом математически эквивалентного преобразования RLS-алгоритма, рассмотренного в главе 6.

II.1. Введение

С точки зрения вычислительной сложности RLS-алгоритмы адаптивной фильтрации, рассмотренные в главах 6—9, являются вычислительно более эффективными (т.е. характеризуются примерно в N раз меньшим числом арифметических операций) по сравнению с алгоритмом вычисления ВК адаптивного фильтра с помощью уравнения (6.69) $\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{r}_N(k)$, в котором на каждой итерации k фигурирует операция обращения корреляционной матрицы обрабатываемых сигналов. Данный выигрыш обусловлен тем, что обращение матрицы, например с помощью алгоритма, описываемого уравнением (3.13), требует на каждой итерации выполнять арифметические операции, число которых пропорционально N^3 , в то время как RLS-алгоритмы, рассмотренные в главах 6—9, требуют на каждой итерации выполнять арифметические операции, число которых пропорционально N^2 . Это позволяет использовать RLS-алгоритмы в адаптивных фильтрах со сравнительно небольшим числом ВК, так как такие фильтры сегодня уже можно реализовать на современных ЦСП [43].

Однако на практике существуют и задачи, где требуются адаптивные фильтры, в которых число ВК исчисляется десятками, сотнями или даже тысячами. Примерами использования таких адаптивных фильтров являются компенсаторы сигналов электрического или акустического эха [37, 38], эквалайзеры каналов связи [39, 40, 124]. Применение RLS-алгоритмов с квадратичной вы-



числительной сложностью $O(N^2)$ арифметических операций в таких приложениях сегодня, по-прежнему, является проблематичным. Уменьшить вычислительную сложность адаптивных фильтров удается путем использования в них быстрых (т.е. с малым числом арифметических операций на одну итерацию) алгоритмов вычисления ВК.

В основе быстрых RLS-алгоритмов находятся эффективные (с линейной сложностью $O(N)$ арифметических операций) процедуры вычисления векторов коэффициентов Калмана $\mathbf{g}_N(k)$, основанные на теории линейного предсказания [8]. Эти процедуры составляют основу быстрых RLS-алгоритмов, известных как Fast Kalman (FK), Fast Transversal Filter (FTF) и Fast a Posteriori Error Sequential Technique (FAEST), которые рассматриваются в настоящей главе. Название «линейное предсказание» происходит от того, что для получения вычислительных процедур рассматриваемых алгоритмов решается задача предсказания текущего отсчета сигнала с помощью линейной функции прошлых отсчетов.

Эта функция может быть реализована с помощью трансверсального линейного фильтра с числом ВК, равным N . При его использовании текущая выборка сигнала $x(k)$ предсказывается с помощью линейной комбинации N наиболее ранних отсчетов, определяемых вектором $\mathbf{x}_N(k-1)$ входных сигналов фильтра. Такое предсказание называется «forward prediction» или «предсказание вперед». Для вычисления различных переменных, возникающих при его реализации, далее будет использоваться верхний индекс « f ».

Аналогично можно «предсказывать» значение прошлого отсчета $x(k-N)$, используя линейную комбинацию будущих отсчетов, определяемых вектором $\mathbf{x}_N(k)$. Такое предсказание называется «backward prediction» или «предсказание назад». Для вычисления различных переменных, возникающих при его реализации, далее будет использоваться верхний индекс « b ».

Участвующие в линейных предсказаниях сигналы образуют векторы $\mathbf{x}_{N+1}(k)$ с числом элементов $N+1$

$$\mathbf{x}_{N+1}(k) = \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix}, \quad (11.1)$$

демонстрирующие свойство инвариантности (т.е. независимости) последовательности $N+1$ отсчетов обрабатываемых сигналов к их сдвигу. Выражение (11.1) показывает, что вектор сигналов $\mathbf{x}_{N+1}(k)$ с числом элементов $N+1$ можно представить либо как текущую выборку входного сигнала $x(k)$ и задержанный на один отсчет вектор $\mathbf{x}_N(k-1)$ с числом элементов N , либо как текущий вектор $\mathbf{x}_N(k)$ и задержанную на N отсчетов выборку входного сигнала $x(k-N)$.

В основе дальнейшего изложения лежат материалы, представленные в [15, 66, 125–134].

11.2. Линейное предсказание вперед

Линейный трансверсальный фильтр, формирующий предсказание вперед, показан на рис. 11.1.

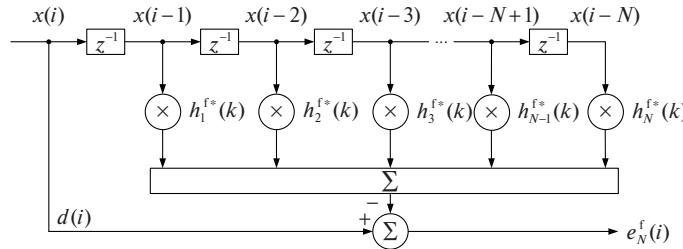


Рис. 11.1. Трансверсальный фильтр, формирующий ошибку линейного предсказания вперед

Здесь отсчеты сигнала $x(i)$ играют роль требуемого сигнала $d(i)$ (см. рис. 6.2). В каждый момент времени k имеется вектор ВК фильтра линейного предсказания вперед $\mathbf{h}_N^f(k) = [h_1^f(k), h_2^f(k), \dots, h_N^f(k)]^T$ с числом элементов, равным N . Для любого значения i определим апостериорную ошибку

$$e_N^f(i) = x(i) - \sum_{n=1}^N h_n^{f*}(k)x(i-n) = x(i) - \mathbf{h}_N^{fH}(k)\mathbf{x}_N(i-1), \quad (11.2)$$

где $\mathbf{x}_N(i-1) = [x(i-1), x(i-2), \dots, x(i-N)]^T$ и $d(i) = x(i)$. В уравнении (11.2) используются индексы i и k , потому что в каждый момент времени k решается LS-задача, которая основана на всех отсчетах наблюдаемых процессов вплоть до момента k (см. раздел 6.6). Поэтому вектор $\mathbf{h}_N^f(k)$ зависит от всех $\mathbf{x}_N(i-1)$ и $x(i)$, где $i=1, \dots, k$.

LS-задачу линейного предсказания вперед можно формализовать как задачу поиска вектора ВК $\mathbf{h}_N^f(k)$, который минимизирует энергию ошибок

$$E_N^f(k) = \sum_{i=1}^k \lambda^{k-i} e_N^f(i) e_N^{f*}(i). \quad (11.3)$$

Используя уравнение (11.2), функцию (11.3) можно представить как

$$\begin{aligned} E_N^f(k) &= \sum_{i=1}^k \lambda^{k-i} e_N^f(i) e_N^{f*}(i) = \sum_{i=1}^k \lambda^{k-i} [x(i) - \mathbf{h}_N^{fH}(k)\mathbf{x}_N(i-1)] [\mathbf{x}_N(i-1) - \mathbf{h}_N^{fH}(k)\mathbf{x}_N(i-1)]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x(i)x(i)^* - \sum_{i=1}^k \lambda^{k-i} x(i)\mathbf{x}_N^H(i-1)\mathbf{h}_N^f(k) - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{fH}(k)\mathbf{x}_N(i-1)x^*(i) + \\ &+ \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{fH}(k)\mathbf{x}_N(i-1)\mathbf{x}_N^H(i-1)\mathbf{h}_N^f(k) = \\ &= \sigma^{f2}(k) - \mathbf{r}_N^{fH}(k)\mathbf{h}_N^f(k) - \mathbf{h}_N^{fH}(k)\mathbf{r}_N^f + \mathbf{h}_N^{fH}(k)\mathbf{R}_N(k-1)\mathbf{h}_N^f(k), \end{aligned} \quad (11.4)$$

где

$$\sigma^{f2}(k) = \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i) = \lambda \sigma^{f2}(k-1) + x(k) x^*(k), \quad (11.5)$$

$$\mathbf{r}_N^f(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) x^*(i) = \lambda \mathbf{r}_N^f(k-1) + \mathbf{x}_N(k-1) x^*(k) \quad (11.6)$$

и

$$\mathbf{R}_N(k-1) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) = \lambda \mathbf{R}_N(k-2) + \mathbf{x}_N(k-1) \mathbf{x}_N^H(k-1) \quad (11.7)$$

по аналогии с уравнениями (6.74) и (6.75).

Значение функции (11.4) минимально при ВК $\mathbf{h}_N^f(k)$, удовлетворяющих условию

$$\nabla_{\mathbf{h}_N^{f*}} E_N^f(k) = \frac{\partial E_N^f(k)}{\partial \mathbf{h}_N^{f*}(k)} = -\mathbf{r}_N^f(k) + \mathbf{R}_N(k-1) \mathbf{h}_N^f(k) = \mathbf{0}_N, \quad (11.8)$$

откуда

$$\mathbf{h}_N^f(k) = \mathbf{R}_N^{-1}(k-1) \mathbf{r}_N^f(k). \quad (11.9)$$

Минимальное значение функций (11.4) находится при подстановке в ее выражение значения

$$\mathbf{r}_N^f(k) = \mathbf{R}_N(k-1) \mathbf{h}_N^f(k), \quad (11.10)$$

определяемого уравнением (11.8). В этом случае

$$E_{N,\min}^f(k) = \sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{h}_N^f(k). \quad (11.11)$$

Для рекурсивного решения LS-задачи минимизации функции (11.4) определим обратное значение матрицы $\mathbf{R}_N(k-1)$ с помощью MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где согласно выражению (11.7) $\mathbf{C} = \lambda \mathbf{R}_N(k-2)$, $\mathbf{C}^{-1} = \lambda^{-1} \mathbf{R}_N^{-1}(k-2)$, $\mathbf{B}^H = \mathbf{x}_N(k)$, $\mathbf{D} = \mathbf{x}_N^H(k-1)$ и $\mathbf{A}^{-1} = -1$. Тогда

$$\begin{aligned} \mathbf{R}_N^{-1}(k-1) &= \lambda^{-1} \mathbf{R}_N^{-1}(k-2) - \frac{\lambda^{-1} \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \mathbf{x}_N^H(k-1) \lambda^{-1} \mathbf{R}_N^{-1}(k-2)}{1 + \mathbf{x}_N^H(k-1) \lambda^{-1} \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1)} = \\ &= \lambda^{-1} \mathbf{R}_N^{-1}(k-2) - \frac{\lambda^{-2} \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2)}{\lambda^{-1} [\lambda + \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1)]} = \\ &= \lambda^{-1} [\mathbf{R}_N^{-1}(k-2) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2)], \end{aligned} \quad (11.12)$$

где

$$\mathbf{g}_N(k-1) = \frac{\mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1)}{\lambda + \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1)}. \quad (11.13)$$

Из уравнения (11.13) следует, что

$$\mathbf{g}_N(k-1) [\lambda + \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1)] = \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \quad (11.14)$$

или, используя уравнение (11.12), что

$$\begin{aligned}\mathbf{g}_N(k-1) &= \lambda^{-1} \left[\mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \right] = \\ &= \lambda^{-1} \left[\mathbf{R}_N^{-1}(k-2) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \right] \mathbf{x}_N(k-1) = \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1),\end{aligned}$$

т. е.

$$\mathbf{g}_N(k-1) = \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1). \quad (11.15)$$

Преобразования (11.12)–(11.15) можно также получить из уравнений (6.76)–(6.79), просто заменив в них k на $k-1$.

С учетом (11.6) и (11.12), уравнение (11.9) преобразуется как

$$\begin{aligned}\mathbf{h}_N^f(k) &= \mathbf{R}_N^{-1}(k-1) \mathbf{r}_N^f(k) = \mathbf{R}_N^{-1}(k-1) \left[\lambda \mathbf{r}_N^f(k-1) + \mathbf{x}_N(k-1) x^*(k) \right] = \\ &= \lambda \mathbf{R}_N^{-1}(k-1) \mathbf{r}_N^f(k-1) + \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) x^*(k) = \\ &= \lambda \lambda^{-1} \left[\mathbf{R}_N^{-1}(k-2) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \right] \mathbf{r}_N^f(k-1) + \\ &\quad + \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) x^*(k) = \mathbf{R}_N^{-1}(k-2) \mathbf{r}_N^f(k-1) - \\ &\quad - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{r}_N^f(k-1) + \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) x^*(k) = \\ &= \mathbf{h}_N^f(k-1) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1) x^*(k) = \\ &= \mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1) \left[x^*(k) - \mathbf{x}_N^H(k-1) \mathbf{h}_N^f(k-1) \right] = \mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1) \alpha_N^{f*}(k),\end{aligned} \quad (11.16)$$

где

$$\alpha_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k-1) \mathbf{x}_N(k-1) \quad (11.17)$$

— априорная ошибка линейного предсказания процесса $\{x(i)\}$, поскольку это ошибка между наблюдаемыми значениями этого процесса и значениями выходного сигнала линейного фильтра (см. рис. 11.1), измеренная перед обновлением вектора ВК фильтра, т. е. на основе ВК $\mathbf{h}_N^f(k-1)$.

Можно также получить уравнение для рекурсивного вычисления минимального значения энергии апостериорных ошибок линейного предсказания вперед процесса $\{x(i)\}$. Для этого в (11.11) используем уравнения (11.5), (11.6) и (11.16). Тогда получим

$$\begin{aligned}E_{N,\min}^f(k) &= \sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{h}_N^f(k) = \lambda \sigma^{f2}(k-1) + x(k) x^*(k) - \\ &\quad - \mathbf{r}_N^{fH}(k) \left[\mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1) \alpha_N^{f*}(k) \right] = \lambda \sigma^{f2}(k-1) + x(k) x^*(k) - \\ &\quad - \left[\lambda \mathbf{r}_N^{fH}(k-1) + x(k) \mathbf{x}_N^H(k-1) \right] \mathbf{h}_N^f(k-1) - \mathbf{r}_N^{fH}(k) \mathbf{g}_N(k-1) \alpha_N^{f*}(k) = \\ &= \lambda \sigma^{f2}(k-1) + x(k) x^*(k) - \lambda \mathbf{r}_N^{fH}(k-1) \mathbf{h}_N^f(k-1) - x(k) \mathbf{x}_N^H(k-1) \mathbf{h}_N^f(k-1) - \\ &\quad - \mathbf{h}_N^{fH}(k) \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) \alpha_N^{f*}(k) = \lambda E_{N,\min}^f(k-1) + \\ &\quad + x(k) \left[x^*(k) - \mathbf{x}_N^H(k-1) \mathbf{h}_N^f(k-1) \right] - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1) \alpha_N^{f*}(k) = \\ &= \lambda E_{N,\min}^f(k-1) + x(k) \alpha_N^{f*}(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1) \alpha_N^{f*}(k) = \\ &= \lambda E_{N,\min}^f(k-1) + \left[x(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1) \right] \alpha_N^{f*}(k) = \lambda E_{N,\min}^f(k-1) + e_N^f(k) \alpha_N^{f*}(k),\end{aligned} \quad (11.18)$$



где

$$e_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1). \quad (11.19)$$

В (11.18) также использованы уравнение (11.11)
 $E_{N,\min}^f(k-1) = \sigma^{f2}(k-1) - \mathbf{r}_N^{fH}(k-1) \mathbf{h}_N^f(k-1)$ при $k-1$, уравнение (11.10) и свойство 1 корреляционной матрицы $\mathbf{R}_N^H(k-1) = \mathbf{R}_N(k-1)$, см. раздел 3.2. Кроме того, аналогично $E_{N,\min}^f(k)$, см. (6.87), также верно, что

$$E_{N,\min}^f(k) = \lambda E_{N,\min}^f(k-1) + \alpha_N^f(k) e_N^{f*}(k). \quad (11.20)$$

Введя обозначение для вектора ВК фильтра линейного предсказания вперед с числом элементов, равным $N+1$,

$$\mathbf{a}_{N+1}(k) = \begin{bmatrix} 1, -\mathbf{h}_N^{fT}(k) \end{bmatrix}^T \quad (11.21)$$

и используя (11.1) и (11.16), уравнения (11.17), (11.19), (11.21) можно представить как

$$\alpha_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1) = \mathbf{a}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k), \quad (11.22)$$

$$e_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1) = \mathbf{a}_{N+1}^H(k) \mathbf{x}_{N+1}(k), \quad (11.23)$$

$$\mathbf{a}_{N+1}(k) = \mathbf{a}_{N+1}(k-1) + \begin{bmatrix} 0 \\ \mathbf{g}_N(k-1) \end{bmatrix} \alpha_N^{f*}(k), \quad (11.24)$$

а корреляционную матрицу с числом элементов $(N+1) \times (N+1)$ на основе векторов (11.1) — как

$$\begin{aligned} \mathbf{R}_{N+1}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) = \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} x(i) \\ \mathbf{x}_N^H(i-1) \end{bmatrix} \begin{bmatrix} x^*(i), \mathbf{x}_N^H(i-1) \end{bmatrix}^T = \\ &= \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i) & \sum_{i=1}^k \lambda^{k-i} x(i) \mathbf{x}_N^H(i-1) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) x^*(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) \end{bmatrix} = \begin{bmatrix} \sigma^{f2}(k) & \mathbf{r}_N^{fH}(k) \\ \mathbf{r}_N^f(k) & \mathbf{R}_N(k-1) \end{bmatrix}. \end{aligned} \quad (11.25)$$

Используя (11.25), (11.21), (11.11) и (11.10), можно получить выражение для произведения

$$\begin{aligned} \mathbf{R}_{N+1}(k) \mathbf{a}_{N+1}(k) &= \begin{bmatrix} \sigma^{f2}(k) & \mathbf{r}_N^{fH}(k) \\ \mathbf{r}_N^f(k) & \mathbf{R}_N(k-1) \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k) \end{bmatrix} = \\ &= \begin{bmatrix} \sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{h}_N^f(k) \\ \mathbf{r}_N^f(k) - \mathbf{R}_N(k-1) \mathbf{h}_N^f(k) \end{bmatrix} = \begin{bmatrix} E_{N,\min}^f(k) \\ \mathbf{0}_N \end{bmatrix} = \mathbf{r}_{N+1}^f(k), \end{aligned} \quad (11.26)$$

которое является аналогичным уравнению $\mathbf{R}_N(k) \mathbf{h}_N(k) = \mathbf{r}_N(k)$, следующему из (6.68).

11.3. Линейное предсказание назад

Линейное предсказание назад выполняется подобно линейному предсказанию вперед. Линейный трансверсальный фильтр, формирующий это предсказание, показан на рис. 11.2. Здесь отсчеты сигнала $x(i - N)$ играют роль требуемого сигнала $d(i)$, см. рис. 6.2. В каждый момент времени k имеется вектор ВК фильтра линейного предсказания назад $\mathbf{h}_N^b(k) = [h_1^b(k), h_2^b(k), \dots, h_N^b(k)]^T$ с числом элементов, равным N . Для любых значений $\mathbf{x}_N(i) = [x(i), x(i-1), \dots, x(i-N+1)]^T$ и $d(i) = x(i - N)$ определим апостериорную ошибку

$$e_N^b(i) = x(i - N) - \sum_{n=1}^N h_n^b(k) x(i) = x(i - N) - \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i). \quad (11.27)$$

Здесь используются индексы i и k , потому что в каждый момент времени k решается LS-задача, которая основана на всех отсчетах наблюдаемых процессов вплоть до момента k (см. раздел 6.6). Поэтому вектор $\mathbf{h}_N^b(k)$ зависит от всех $\mathbf{x}_N(i)$ и $x(i - N)$, где $i = 1, \dots, k$.

LS-задачу линейного предсказания назад можно формализовать как задачу поиска вектора ВК $\mathbf{h}_N^b(k)$, который минимизирует энергию ошибок

$$E_N^b(k) = \sum_{i=1}^k \lambda^{k-i} e_N^b(i) e_N^{b*}(i). \quad (11.28)$$

Используя уравнение (11.27), функцию (11.28) можно представить как

$$\begin{aligned} E_N^b(k) &= \sum_{i=1}^k \lambda^{k-i} e_N^b(i) e_N^{b*}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} [x(i - N) - \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i)] [x(i - N) - \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i)]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x(i - N) x(i - N)^* - \sum_{i=1}^k \lambda^{k-i} x(i - N) \mathbf{x}_N^H(i) \mathbf{h}_N^b(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i) x^*(i - N) + \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i) \mathbf{x}_N^H(i) \mathbf{h}_N^b(k) = \\ &= \sigma^{b2}(k) - \mathbf{r}_N^b(k) \mathbf{h}_N^b(k) - \mathbf{h}_N^{bH}(k) \mathbf{r}_N^b + \mathbf{h}_N^{bH}(k) \mathbf{R}_N(k) \mathbf{h}_N^b(k), \end{aligned} \quad (11.29)$$

где

$$\sigma^{b2}(k) = \sum_{i=1}^k \lambda^{k-i} x(i - N) x^*(i - N) = \lambda \sigma^{b2}(k-1) + x(k - N) x^*(k - N), \quad (11.30)$$

$$\mathbf{r}_N^b(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) x^*(i - N) = \lambda \mathbf{r}_N^b(k-1) + \mathbf{x}_N(k) x^*(k - N) \quad (11.31)$$

и

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \quad (11.32)$$

по аналогии с уравнениями (6.74) и (6.75).

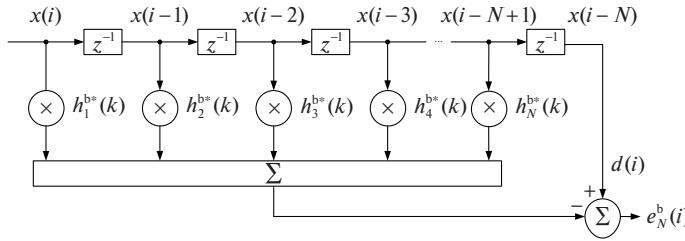


Рис. 11.2. Трансверсальный фильтр, формирующий ошибку линейного предсказания назад

Значение функции (11.29) минимально при ВК $\mathbf{h}_N^b(k)$, удовлетворяющих условию

$$\nabla_{\mathbf{h}_N^b} E_N^b(k) = \frac{\partial E_N^b(k)}{\partial \mathbf{h}_N^b(k)} = -\mathbf{r}_N^b(k) + \mathbf{R}_N(k)\mathbf{h}_N^b(k) = \mathbf{0}_N, \quad (11.33)$$

откуда

$$\mathbf{h}_N^b(k) = \mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k). \quad (11.34)$$

Минимальное значение функций (11.29) определяется при подстановке в ее выражение значения

$$\mathbf{r}_N^b(k) = \mathbf{R}_N(k)\mathbf{h}_N^b(k) \quad (11.35)$$

из уравнения (11.33). В этом случае

$$E_{N,\min}^b(k) = \sigma^{b2}(k) - \mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k). \quad (11.36)$$

Рекурсивное решение LS-задачи минимизации функций (11.29) осуществляется следующим образом.

Обратное значение матрицы (11.32) совпадает с выражением (6.76), т. е.

$$\mathbf{R}_N^{-1}(k) = \lambda^{-1} \left[\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1) \right], \quad (11.37)$$

а значение вектора Калмана $\mathbf{g}_N(k)$ — с уравнениями (6.77) и (6.79), т. е.

$$\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k). \quad (11.38)$$

С учетом (11.37) и (11.31) уравнение (11.34) можно преобразовать как

$$\begin{aligned} \mathbf{h}_N^b(k) &= \mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k) = \mathbf{R}_N^{-1}(k) \left[\lambda \mathbf{r}_N^b(k-1) + \mathbf{x}_N(k)x^*(k-N) \right] = \\ &= \lambda \mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k-1) + \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)x^*(k-N) = \\ &= \lambda \lambda^{-1} \left[\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1) \right] \mathbf{r}_N^b(k-1) + \\ &+ \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)x^*(k-N) = \mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^b(k-1) - \\ &- \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^b(k-1) + \mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)x^*(k-N) = \\ &= \mathbf{h}_N^b(k-1) - \mathbf{g}_N(k)\mathbf{x}_N^H(k)\mathbf{h}_N^b(k-1) + \mathbf{g}_N(k)x^*(k-N) = \\ &= \mathbf{h}_N^b(k-1) + \mathbf{g}_N(k) \left[x^*(k-N) - \mathbf{x}_N^H(k)\mathbf{h}_N^b(k-1) \right] = \mathbf{h}_N^b(k-1) + \mathbf{g}_N(k)\alpha_N^{b*}(k), \end{aligned} \quad (11.39)$$

где

$$\alpha_N^b(k) = x(k-N) - \mathbf{h}_N^{bH}(k-1)\mathbf{x}_N(k) \quad (11.40)$$

— априорная ошибка линейного предсказания процесса $\{x(i-N)\}$, поскольку это ошибка между наблюдаемыми значениями этого процесса и значениями выходного сигнала трансверсального фильтра линейного предсказания, измеренная перед обновлением вектора ВК фильтра, т. е. на основе ВК $\mathbf{h}_N^b(k-1)$.

Можно также получить уравнение рекурсивного вычисления минимального значения энергии апостериорных ошибок линейного предсказания вперед процесса $\{x(i-N)\}$. Для этого в (11.36) используем (11.30), (11.31) и (11.39). Тогда

$$\begin{aligned} E_{N,\min}^b(k) &= \sigma^{b2}(k) - \mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k) = \lambda\sigma^{b2}(k-1) + x(k-N)x^*(k-N) - \\ &- \mathbf{r}_N^{bH}(k)[\mathbf{h}_N^b(k-1) + \mathbf{g}_N(k)\alpha_N^{b*}(k)] = \lambda\sigma^{b2}(k-1) + x(k-N)x^*(k-N) - \\ &- [\lambda\mathbf{r}_N^{bH}(k-1) + x(k-N)\mathbf{x}_N^H(k)]\mathbf{h}_N^b(k-1) - \mathbf{r}_N^{bH}(k)\mathbf{g}_N(k)\alpha_N^{b*}(k) = \\ &= \lambda\sigma^{b2}(k-1) + x(k-N)x^*(k-N) - \lambda\mathbf{r}_N^{bH}(k-1)\mathbf{h}_N^b(k-1) - \\ &- x(k-N)\mathbf{x}_N^H(k)\mathbf{h}_N^b(k-1) - \mathbf{h}_N^{bH}(k)\mathbf{R}_N(k)\mathbf{R}_N^{-1}(k)\mathbf{x}_N(k)\alpha_N^{b*}(k) = \\ &= \lambda E_{N,\min}^b(k-1) + x(k-N)[x^*(k-N) - \mathbf{x}_N^H(k)\mathbf{h}_N^b(k-1)] - \\ &- \mathbf{h}_N^{bH}(k)\mathbf{x}_N(k)\alpha_N^{b*}(k) = \lambda E_{N,\min}^b(k-1) + x(k-N)\alpha_N^{b*}(k) - \\ &- \mathbf{h}_N^{bH}(k)\mathbf{x}_N(k)\alpha_N^{b*}(k) = \lambda E_{N,\min}^b(k-1) + [x(k-N) - \mathbf{h}_N^{bH}(k)\mathbf{x}_N(k)]\alpha_N^{b*}(k) = \\ &= \lambda E_{N,\min}^b(k-1) + e_N^b(k)\alpha_N^{b*}(k), \end{aligned} \quad (11.41)$$

где

$$e_N^b(k) = x(k-N) - \mathbf{h}_N^{bH}(k)\mathbf{x}_N(k), \quad (11.42)$$

Здесь также использованы уравнение (11.36) $E_{N,\min}^b(k-1) = \sigma^{b2}(k-1) - \mathbf{r}_N^{bH}(k-1)\mathbf{h}_N^b(k-1)$ при $k-1$, уравнение (11.35) и свойство 1 корреляционной матрицы (3.7) $\mathbf{R}_N^H(k) = \mathbf{R}_N(k)$, см. раздел 3.2. Кроме того, аналогично $E_{N,\min}^b(k)$, см. (6.87), также верно, что

$$E_{N,\min}^b(k) = \lambda E_{N,\min}^b(k-1) + \alpha_N^b(k)e_N^{b*}(k). \quad (11.43)$$

Введя обозначение для вектора ВК фильтра линейного предсказания назад с числом элементов, равным $N+1$,

$$\mathbf{b}_{N+1}(k) = [-\mathbf{h}_N^{bT}(k), 1]^T \quad (11.44)$$

и используя (11.1) и (11.39), уравнения (11.40), (11.42), (11.44) можно представить как

$$\alpha_N^b(k) = x(k-N) - \mathbf{h}_N^{bH}(k-1)\mathbf{x}_N(k) = \mathbf{b}_{N+1}^H(k-1)\mathbf{x}_{N+1}(k), \quad (11.45)$$

$$e_N^b(k) = x(k-N) - \mathbf{h}_N^{bH}(k)\mathbf{x}_N(k) = \mathbf{b}_{N+1}^H(k)\mathbf{x}_{N+1}(k), \quad (11.46)$$

$$\mathbf{b}_{N+1}(k) = \mathbf{b}_{N+1}(k-1) + \begin{bmatrix} \mathbf{g}_N(k) \\ 0 \end{bmatrix} \alpha_N^{b*}(k), \quad (11.47)$$

а корреляционную матрицу на основе векторов (11.1) — как

$$\begin{aligned} \mathbf{R}_{N+1}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) = \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_N(i) \\ \mathbf{x}(i-N) \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^H(i) \\ \mathbf{x}^*(i-N) \end{bmatrix} = \\ &= \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^H(i) \mathbf{x}^*(i-N) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i-N) \mathbf{x}_N^H(i-1) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^*(i) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_N(k) & \mathbf{r}_N^b(k) \\ \mathbf{r}_N^{bH}(k) & \sigma^{b2}(k) \end{bmatrix}. \end{aligned} \quad (11.48)$$

Используя (11.48), (11.44), (11.35) и (11.36), можно получить произведение

$$\begin{aligned} \mathbf{R}_{N+1}(k) \mathbf{b}_{N+1}(k) &= \begin{bmatrix} \mathbf{R}_N(k) & \mathbf{r}_N^b(k) \\ \mathbf{r}_N^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} \begin{bmatrix} -\mathbf{h}_N^b(k) \\ 1 \end{bmatrix} = \\ &= \begin{bmatrix} -\mathbf{R}_N(k) \mathbf{h}_N^b(k) + \mathbf{r}_N^b(k) \\ -\mathbf{r}_N^{bH}(k) \mathbf{h}_N^b(k) + \sigma^{b2}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ E_{N,\min}^b(k) \end{bmatrix} = \mathbf{r}_{N+1}^b(k), \end{aligned} \quad (11.49)$$

которое является аналогичным уравнению $\mathbf{R}_N(k) \mathbf{h}_N(k) = \mathbf{r}_N(k)$, следующему из уравнения (6.68).

11.4. Быстрое вычисление вектора коэффициентов Калмана

В уравнении (6.80) для рекурсивного вычисления векторов ВК $\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^{*}(k)$ адаптивного фильтра, функционирующего по критерию наименьших квадратов, используется вектор коэффициентов Калмана (6.79) $\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)$. Согласно материалам глав 6—9 сложность вычисления этого вектора оценивается как $O(N^2)$ арифметических операций на одну итерацию, что определяет результирующую сложность RLS-алгоритмов адаптивной фильтрации (см. табл. 6.1, табл. 7.1, табл. 8.1, табл. 9.1—9.3).

Рассматриваемые в настоящем разделе быстрые RLS-алгоритмы используют способы вычисления вектора $\mathbf{g}_N(k)$, которые основаны на следующих из теории линейного предсказания соотношениях между векторами Калмана в фильтрах с числом ВК N и $N+1$. Эти вычисления имеют сложность $O(N)$ арифметических операций на одну итерацию. Поэтому результирующие RLS-алгоритмы называются быстрыми, т. е. вычислительно эффективными в терминах требуемого числа арифметических операций, так как полная вычислительная сложность этих алгоритмов равна не $O(N^2)$, а $O(N)$ арифметическим операциям.

Быстрые RLS-алгоритмы являются математически эквивалентными друг другу и рассмотренным в главах 6–9 RLS-алгоритмам. Эквивалентность означает демонстрацию на каждой итерации одинаковых выходных сигналов в фильтрах, использующих сравниваемые алгоритмы, при соответствующей эквивалентной их инициализации (см. раздел 14.8), реализации алгоритмов в арифметике с плавающей точкой, одинаковых параметрах N, λ, δ и при обработке одинаковых входных сигналов.

Вектор коэффициентов Калмана (6.79) называется апостериорным, так как для его вычисления используется матрица $\mathbf{R}_N^{-1}(k)$, определяемая на итерации k , по аналогии с апостериорными ошибками, использующими в вычислениях векторы $\mathbf{h}_N(k)$, $\mathbf{h}_N^f(k)$ и $\mathbf{h}_N^b(k)$.

По аналогии с уравнением (6.79), вектор коэффициентов Калмана с числом элементов $N+1$ определяется как

$$\mathbf{g}_{N+1}(k) = \mathbf{R}_{N+1}^{-1}(k) \mathbf{x}_{N+1}(k), \quad (11.50)$$

где $\mathbf{R}_{N+1}^{-1}(k)$ и $\mathbf{x}_{N+1}(k)$ определяются уравнениями (11.25), (11.48) и (11.1).

Чтобы установить соотношения между векторами $\mathbf{g}_N(k)$ и $\mathbf{g}_{N+1}(k)$, необходимо определить значения матрицы $\mathbf{R}_{N+1}^{-1}(k)$. Поскольку эта матрица согласно выражениям (11.25) и (11.48) представляется в виде клеточной матрицы, то для ее обращения могут быть использованы леммы об обращении таких матриц [8]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{C}^{-1} \\ -\mathbf{C}^{-1}\mathbf{B}^H(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1} & \mathbf{C}^{-1}\mathbf{B}^H(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{C}^{-1} + \mathbf{C}^{-1} \end{bmatrix} \quad (11.51)$$

и

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A}^{-1} & (\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}. \quad (11.52)$$

Для обращения матрицы (11.25) необходимо воспользоваться леммой (11.51). Если представить обратное значение матрицы (11.25) как

$$\mathbf{R}_{N+1}^{-1}(k) = \begin{bmatrix} \mathbf{R}_{1,1}^{-1}(k) & \mathbf{R}_{1,2}^{-1}(k) \\ \mathbf{R}_{2,1}^{-1}(k) & \mathbf{R}_{2,2}^{-1}(k) \end{bmatrix}, \quad (11.53)$$

то с учетом уравнения (11.11)

$$\begin{aligned} \mathbf{R}_{1,1}^{-1}(k) &= \left[\sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{R}_N^{-1}(k-1) \mathbf{r}_N^f(k) \right]^{-1} = \\ &= \left[\sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{h}_N^f(k) \right]^{-1} = \left[E_{N,\min}^f(k) \right]^{-1}, \end{aligned} \quad (11.54)$$

$$\begin{aligned} \mathbf{R}_{1,2}^{-1}(k) &= - \left[\sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{R}_N^{-1}(k-1) \mathbf{r}_N^f(k) \right]^{-1} \mathbf{r}_N^{fH}(k) \mathbf{R}_N^{-1}(k-1) = \\ &= \left[\sigma^{f2}(k) - \mathbf{r}_N^{fH}(k) \mathbf{h}_N^f(k) \right]^{-1} = - \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{h}_N^{fH}(k), \end{aligned} \quad (11.55)$$

$$\begin{aligned}\mathbf{R}_{2,1}^{-1}(k) &= -\mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^f(k)\left[\sigma^{f2}(k)-\mathbf{r}_N^{fH}(k)\mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^f(k)\right]^{-1} = \\ &= -\mathbf{h}_N^f(k)\left[\sigma^{f2}(k)-\mathbf{r}_N^{fH}(k)\mathbf{h}_N^f(k)\right]^{-1} = -\mathbf{h}_N^f(k)\left[E_{N,\min}^f(k)\right]^{-1},\end{aligned}\quad (11.56)$$

$$\begin{aligned}\mathbf{R}_{2,2}^{-1}(k) &= \mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^f(k)\left[\sigma^{f2}(k)-\mathbf{r}_N^{fH}(k)\mathbf{R}_N^{-1}(k-1)\mathbf{r}_N^f(k)\right]^{-1} \times \\ &\quad \times \mathbf{r}_N^{fH}(k)\mathbf{R}_N^{-1}(k-1)+\mathbf{R}_N^{-1}(k-1) = \\ &= \mathbf{h}_N^f(k)\left[\sigma^{f2}(k)-\mathbf{r}_N^{fH}(k)\mathbf{h}_N^f(k)\right]^{-1}\mathbf{h}_N^{fH}(k)+\mathbf{R}_N^{-1}(k-1) = \\ &= \mathbf{h}_N^f(k)\left[E_{N,\min}^f(k)\right]^{-1}\mathbf{h}_N^{fH}(k)+\mathbf{R}_N^{-1}(k-1).\end{aligned}\quad (11.57)$$

Используя выражения (11.54)–(11.57), матрицу (11.53) можно представить как

$$\begin{aligned}\mathbf{R}_{N+1}^{-1}(k) &= \begin{bmatrix} \left[E_{N,\min}^f(k)\right]^{-1} & -\left[E_{N,\min}^f(k)\right]^{-1}\mathbf{h}_N^{fH}(k) \\ -\mathbf{h}_N^f(k)\left[E_{N,\min}^f(k)\right]^{-1} & \mathbf{h}_N^f(k)\left[E_{N,\min}^f(k)\right]^{-1}\mathbf{h}_N^{fH}(k)+\mathbf{R}_N^{-1}(k-1) \end{bmatrix} = \\ &= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k) \end{bmatrix}\left[E_{N,\min}^f(k)\right]^{-1}\begin{bmatrix} 1, -\mathbf{h}_N^{fH}(k) \end{bmatrix} = \\ &= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k)\left[E_{N,\min}^f(k)\right]^{-1}\mathbf{a}_{N+1}^H(k).\end{aligned}\quad (11.58)$$

Для обращения матрицы (11.48) необходимо воспользоваться леммой (11.52). Тогда элементы обратной матрицы (11.53) с учетом (11.36) определяются как

$$\begin{aligned}\mathbf{R}_{1,1}^{-1}(k) &= \mathbf{R}_N^{-1}(k)+\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\right]^{-1} \times \\ &\quad \times \mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k) = \mathbf{R}_N^{-1}(k)+\mathbf{h}_N^b(k)\left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k)\right]^{-1}\mathbf{h}_N^{bH}(k) = \\ &= \mathbf{R}_N^{-1}(k)+\mathbf{h}_N^b(k)\left[E_{N,\min}^b(k)\right]^{-1}\mathbf{h}_N^{bH}(k),\end{aligned}\quad (11.59)$$

$$\begin{aligned}\mathbf{R}_{1,2}^{-1}(k) &= -\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\right]^{-1} = \\ &= -\mathbf{h}_N^b(k)\left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k)\right]^{-1} = -\mathbf{h}_N^b(k)\left[E_{N,\min}^b(k)\right]^{-1},\end{aligned}\quad (11.60)$$

$$\begin{aligned}\mathbf{R}_{2,1}^{-1}(k) &= -\left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\right]^{-1}\mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k) = \\ &= \left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k)\right]^{-1} = -\left[E_{N,\min}^b(k)\right]^{-1}\mathbf{h}_N^{bH}(k),\end{aligned}\quad (11.61)$$

$$\begin{aligned}\mathbf{R}_{2,2}^{-1}(k) &= \left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{R}_N^{-1}(k)\mathbf{r}_N^b(k)\right]^{-1} = \\ &= \left[\sigma^{b2}(k)-\mathbf{r}_N^{bH}(k)\mathbf{h}_N^b(k)\right]^{-1} = \left[E_{N,\min}^b(k)\right]^{-1}.\end{aligned}\quad (11.62)$$

Используя выражения (11.59)–(11.62), матрицу (11.53) можно представить как

$$\begin{aligned}
\mathbf{R}_{N+1}^{-1}(k) &= \begin{bmatrix} \mathbf{R}_N^{-1}(k) + \mathbf{h}_N^b(k) [E_{N,\min}^b(k)]^{-1} \mathbf{h}_N^{bH}(k) & -\mathbf{h}_N^b(k) [E_{N,\min}^b(k)]^{-1} \\ -[E_{N,\min}^b(k)]^{-1} \mathbf{h}_N^{bH}(k) & [E_{N,\min}^b(k)]^{-1} \end{bmatrix} = \\
&= \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_N^b(k) \\ 1 \end{bmatrix} [E_{N,\min}^b(k)]^{-1} \begin{bmatrix} -\mathbf{h}_N^{bH}(k), 1 \end{bmatrix} = \\
&= \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}(k) [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}^H(k).
\end{aligned} \tag{11.63}$$

Теперь, используя выражения (11.1), (11.58) и (11.23) в уравнении (11.50), получим, что

$$\begin{aligned}
\mathbf{g}_{N+1}(k) &= \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k) \mathbf{a}_{N+1}^H(k) \right\} \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k) [E_{N,\min}^f(k)]^{-1} \mathbf{a}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) & \end{bmatrix} + \mathbf{a}_{N+1}(k) [E_{N,\min}^f(k)]^{-1} \mathbf{a}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} 0 \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k) [E_{N,\min}^f(k)]^{-1} e_N^f(k),
\end{aligned} \tag{11.64}$$

и аналогично, используя выражения (11.1), (11.63) и (11.46) в уравнении (11.50), получим, что

$$\begin{aligned}
\mathbf{g}_{N+1}(k) &= \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}(k) [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}^H(k) \right\} \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} + \mathbf{b}_{N+1}(k) [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k) [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
&= \begin{bmatrix} \mathbf{g}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k) [E_{N,\min}^b(k)]^{-1} e_N^b(k).
\end{aligned} \tag{11.65}$$

Определим также векторы коэффициентов Калмана, использующие матрицы $\mathbf{R}_N^{-1}(k-1)$ и $\mathbf{R}_{N+1}^{-1}(k-1)$ как

$$\mathbf{t}_N(k) = \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k), \tag{11.66}$$

$$\mathbf{t}_{N+1}(k) = \mathbf{R}_{N+1}^{-1}(k-1) \mathbf{x}_{N+1}(k). \tag{11.67}$$

Векторы коэффициентов Калмана (11.66) и (11.67) называются априорными, так как для их вычисления используются матрицы $\mathbf{R}_N^{-1}(k-1)$ и $\mathbf{R}_{N+1}^{-1}(k-1)$, определяемые на итерации $k-1$, по аналогии с априорными ошибками, использующими в вычислениях векторы $\mathbf{h}_N(k-1)$, $\mathbf{h}_N^f(k-1)$ и $\mathbf{h}_N^b(k-1)$.

Используя выражения (11.58) и (11.63) при $k-1$, выражения (11.22) и (11.45), уравнение (11.67) можно представить как

$$\begin{aligned}
 \mathbf{t}_{N+1}(k) &= \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-2) \end{bmatrix} + \mathbf{a}_{N+1}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^H(k-1) \right\} \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-2) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} 0 \\ \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} 0 \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \alpha_N^f(k)
 \end{aligned} \tag{11.68}$$

и как

$$\begin{aligned}
 \mathbf{t}_{N+1}(k) &= \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}(k-1) \left[E_{N,\min}^b(k-1) \right]^{-1} \mathbf{b}_{N+1}^H(k-1) \right\} \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} + \mathbf{b}_{N+1}(k-1) \left[E_{N,\min}^b(k-1) \right]^{-1} \mathbf{b}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k-1) \left[E_{N,\min}^b(k-1) \right]^{-1} \mathbf{b}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= \begin{bmatrix} \mathbf{t}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k-1) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_N^b(k).
 \end{aligned} \tag{11.69}$$

Соотношение между векторами $\mathbf{t}_N(k)$ и $\mathbf{g}_N(k)$ можно установить следующим образом. Из (6.79) определим вектор $\mathbf{x}_N(k) = \mathbf{R}_N(k) \mathbf{g}_N(k)$ и используем его в уравнении (11.66). Тогда с учетом уравнения (6.75)

$$\begin{aligned}
 \mathbf{t}_N(k) &= \mathbf{R}_N^{-1}(k-1) \mathbf{R}_N(k) \mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k-1) \left[\lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right] \mathbf{g}_N(k) = \\
 &= \mathbf{R}_N^{-1}(k-1) \lambda \mathbf{R}_N(k-1) \mathbf{g}_N(k) + \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k) \mathbf{x}_N^H(k) \mathbf{g}_N(k) = \\
 &= \lambda \mathbf{g}_N(k) + \mathbf{t}_N(k) \mathbf{x}_N^H(k) \mathbf{g}_N(k),
 \end{aligned} \tag{11.70}$$

откуда

$$\mathbf{t}_N(k) = \lambda \mathbf{g}_N(k) \left[1 - \mathbf{x}_N^H(k) \mathbf{g}_N(k) \right]^{-1}. \tag{11.71}$$

Произведение $\mathbf{x}_N^H(k) \mathbf{g}_N(k)$ является действительной неотрицательной скалярной величиной, так как аналогично уравнению (3.12)

$$\begin{aligned}
 \mathbf{x}_N^H(k) \mathbf{g}_N(k) &= \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) = \mathbf{g}_N^H(k) \mathbf{x}_N(k) = \\
 &= \left[\mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \right]^H \mathbf{x}_N(k) = \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \geq 0.
 \end{aligned} \tag{11.72}$$

Поэтому

$$\mathbf{t}_N(k) = \lambda \mathbf{g}_N(k) \left[1 - \mathbf{x}_N^H(k) \mathbf{g}_N(k) \right]^{-1} = \lambda \mathbf{g}_N(k) \left[1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k) \right]^{-1}. \tag{11.73}$$

Уравнения (11.64), (11.65), (11.68) и (11.69) позволяют рекурсивно вычислять векторы $\mathbf{g}_N(k)$, используя $\mathbf{g}_{N+1}(k)$, и векторы $\mathbf{t}_N(k)$, используя $\mathbf{t}_{N+1}(k)$. В этих вычислениях отсутствуют матричные операции, что обеспечивает линейную сложность $O(N)$ быстрых RLS-алгоритмов, основанных на этих вычислениях.

11.5. Соотношение между апостериорными и априорными ошибками

Отношение апостериорных и априорных ошибок также играет важную роль в быстрых RLS-алгоритмах.

Это отношение для ошибок оценивания процесса $\{d(i)\}$ с учетом уравнений (6.82) и (6.80) определяется как

$$\begin{aligned}
 \varphi_N(k) &= \frac{e_N(k)}{\alpha_N(k)} = \frac{d(k) - \mathbf{h}_N^H(k) \mathbf{x}_N(k)}{\alpha_N(k)} = \\
 &= \frac{d(k) - [\mathbf{h}_N^H(k-1) + \alpha_N(k) \mathbf{g}_N^H(k)] \mathbf{x}_N(k)}{\alpha_N(k)} = \\
 &= \frac{d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) - \alpha_N(k) \mathbf{g}_N^H(k) \mathbf{x}_N(k)}{\alpha_N(k)} = \\
 &= \frac{\alpha_N(k) - \alpha_N(k) \mathbf{g}_N^H(k) \mathbf{x}_N(k)}{\alpha_N(k)} = \frac{\alpha_N(k) [1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k)]}{\alpha_N(k)} = 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k).
 \end{aligned} \tag{11.74}$$

Поскольку, как уже указывалось, $\mathbf{g}_N^H(k) \mathbf{x}_N(k)$ является действительной скалярной величиной, то $\varphi_N(k)$ — также действительная величина.

Аналогично, с учетом уравнений (11.19) и (11.16) для ошибок линейного предсказания вперед отношение

$$\begin{aligned}
 \frac{e_N^f(k)}{\alpha_N^f(k)} &= \frac{x(k) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(k-1)}{\alpha_N^f(k)} = \\
 &= \frac{x(k) - [\mathbf{h}_N^{fH}(k-1) + \alpha_N^f(k) \mathbf{g}_N^H(k-1)] \mathbf{x}_N(k-1)}{\alpha_N^f(k)} = \\
 &= \frac{x(k) - \mathbf{h}_N^{fH}(k-1) \mathbf{x}_N(k-1) - \alpha_N^f(k) \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1)}{\alpha_N^f(k)} = \\
 &= \frac{\alpha_N^f(k) - \alpha_N^f(k) \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1)}{\alpha_N^f(k)} = \frac{\alpha_N^f(k) [1 - \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1)]}{\alpha_N^f(k)} = \\
 &= 1 - \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1) = \varphi_N(k-1)
 \end{aligned} \tag{11.75}$$

и с учетом уравнений (11.42) и (11.39) для ошибок линейного предсказания назад отношение

$$\begin{aligned}
\frac{e_N^b(k)}{\alpha_N^b(k)} &= \frac{x(k-N) - \mathbf{h}_N^{bH}(k) \mathbf{x}_N(k)}{\alpha_N^b(k)} = \\
&= \frac{x(k-N) - [\mathbf{h}_N^{bH}(k-1) + \alpha_N^b(k) \mathbf{g}_N^H(k)] \mathbf{x}_N(k)}{\alpha_N^b(k)} = \\
&= \frac{x(k-N) - \mathbf{h}_N^{bH}(k-1) \mathbf{x}_N(k) - \alpha_N^b(k) \mathbf{g}_N^H(k) \mathbf{x}_N(k)}{\alpha_N^b(k)} = \\
&= \frac{\alpha_N^b(k) - \alpha_N^b(k) \mathbf{g}_N^H(k) \mathbf{x}_N(k)}{\alpha_N^b(k)} = \frac{\alpha_N^b(k) [1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k)]}{\alpha_N^b(k)} = \\
&= 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k) = \varphi_N(k).
\end{aligned} \tag{11.76}$$

Уравнения (11.74)–(11.76) позволяют по известным априорным ошибкам вычислять апостериорные ошибки до того, как будут определены соответствующие векторы ВК. Кроме того, сложность этих вычислений не N операций сложения и N операций умножения, как в уравнениях (6.82), (11.19) и (11.42), а всего одна операция умножения, так как из уравнений (11.74)–(11.76) следует, что

$$e_N(k) = \alpha_N(k) \varphi_N(k), \tag{11.77}$$

$$e_N^f(k) = \alpha_N^f(k) \varphi_N(k-1), \tag{11.78}$$

$$e_N^b(k) = \alpha_N^b(k) \varphi_N(k). \tag{11.79}$$

Кроме того, используя (11.74) или (11.76) в уравнении (11.73), можно установить следующее соотношение между векторами коэффициентов Калмана

$$\mathbf{t}_N(k) = \lambda \mathbf{g}_N(k) \varphi_N^{-1}(k) \tag{11.80}$$

или

$$\mathbf{g}_N(k) = \lambda^{-1} \mathbf{t}_N(k) \varphi_N(k). \tag{11.81}$$

11.6. Рекурсивное вычисление отношения ошибок

Чтобы воспользоваться уравнениями (11.77)–(11.79), необходимо знать $\varphi_N(k)$. Как будет показано ниже, переменную $\varphi_N(k)$ или ее обратное значение можно определять из $\varphi_{N+1}(k)$, т. е. одноименной переменной для фильтра с числом ВК, равным $N+1$.

Из уравнения (11.74) следует, что

$$\varphi_{N+1}(k) = \frac{e_{N+1}(k)}{\alpha_{N+1}(k)} = 1 - \mathbf{g}_{N+1}^H(k) \mathbf{x}_{N+1}(k). \tag{11.82}$$

С учетом (11.1), (11.64), (11.75) и (11.23) уравнение (11.82) можно записать как

$$\begin{aligned}
 \varphi_{N+1}(k) &= 1 - \mathbf{g}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{e}_N^f(k) \right\}^H \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \\
 &= 1 - \left[0, \mathbf{g}_N^H(k-1) \right] \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} - \mathbf{e}_N^{f*}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{a}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \quad (11.83) \\
 &= 1 - \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1) - \mathbf{e}_N^{f*}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{e}_N^f(k) = \\
 &= \varphi_N(k-1) - \mathbf{e}_N^{f*}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{e}_N^f(k),
 \end{aligned}$$

а с учетом (11.1), (11.65) и (11.76) и (11.46) — как

$$\begin{aligned}
 \varphi_{N+1}(k) &= 1 - \mathbf{g}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k) \right\}^H \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} = \\
 &= 1 - \left[\mathbf{g}_N^H(k), 0 \right] \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} - \mathbf{e}_N^{b*}(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{b}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \quad (11.84) \\
 &= 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k) - \mathbf{e}_N^{b*}(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k) = \\
 &= \varphi_N(k) - \mathbf{e}_N^{b*}(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k).
 \end{aligned}$$

Таким образом, из (11.83) и (11.84) следует, что, используя известное из предыдущей итерации алгоритма значение $\varphi_N(k-1)$, с помощью уравнения (11.83) можно определить $\varphi_{N+1}(k)$, а затем из (11.84) определить $\varphi_N(k)$ при условии, что другие переменные в этих уравнениях известны, т. е. вычислены.

Обратное отношение ошибок вычисляется следующим образом. Используя уравнения (11.74) и (11.81), определим

$$\begin{aligned}
 \varphi_N(k) &= 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k) = \\
 &= 1 - \left[\lambda^{-1} \mathbf{t}_N(k) \varphi_N(k) \right]^H \mathbf{x}_N(k) = \quad (11.85) \\
 &= 1 - \lambda^{-1} \varphi_N(k) \mathbf{t}_N^H(k) \mathbf{x}_N(k),
 \end{aligned}$$

откуда

$$\varphi_N(k) = \left[1 + \lambda^{-1} \mathbf{t}_N^H(k) \mathbf{x}_N(k) \right]^{-1} \quad (11.86)$$

или

$$\varphi_N^{-1}(k) = 1 + \lambda^{-1} \mathbf{t}_N^H(k) \mathbf{x}_N(k). \quad (11.87)$$

По аналогии с уравнением (11.87) определим

$$\varphi_{N+1}^{-1}(k) = 1 + \lambda^{-1} \mathbf{t}_{N+1}^H(k) \mathbf{x}_{N+1}(k). \quad (11.88)$$

Далее, используя (11.1), (11.68), (11.22) и (11.87) в уравнении (11.88), получим, что

$$\begin{aligned}
 \varphi_{N+1}^{-1}(k) &= 1 + \lambda^{-1} \mathbf{t}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
 &= 1 + \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{a}_{N+1}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \alpha_{N+1}^f(k) \right\}^H \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \\
 &= 1 - \lambda^{-1} \begin{bmatrix} 0, \mathbf{t}_N^H(k-1) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \\
 &\quad + \lambda^{-1} \alpha_N^{f*}(k) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= 1 + \lambda^{-1} \mathbf{t}_N^H(k-1) \mathbf{x}_N(k-1) + \lambda^{-1} \alpha_N^{f*}(k) \left[E_{N,\min}^f(k-1) \right]^{-1} \alpha_N^f(k) = \\
 &= \varphi_N^{-1}(k-1) + \lambda^{-1} \alpha_N^{f*}(k) \left[E_{N,\min}^f(k-1) \right]^{-1} \alpha_N^f(k),
 \end{aligned} \tag{11.89}$$

а используя (11.1), (11.69) и (11.45) и (11.87) в уравнении (11.88), получим, что

$$\begin{aligned}
 \varphi_{N+1}^{-1}(k) &= 1 + \lambda^{-1} \mathbf{t}_{N+1}^H(k) \mathbf{x}_{N+1}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}(k-1) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_N^b(k) \right\}^H \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \begin{bmatrix} \mathbf{t}_N^H(k), 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix} + \\
 &\quad + \lambda^{-1} \alpha_{N+1}^{b*}(k) \left[E_{N,\min}^b(k-1) \right]^{-1} \mathbf{b}_{N+1}^H(k-1) \mathbf{x}_{N+1}(k) = \\
 &= 1 + \lambda^{-1} \mathbf{t}_{N+1}^H(k) \mathbf{x}_N(k) + \lambda^{-1} \alpha_{N+1}^{b*}(k) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_{N+1}^b(k) = \\
 &= \varphi_N^{-1}(k) + \lambda^{-1} \alpha_{N+1}^{b*}(k) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_{N+1}^b(k).
 \end{aligned} \tag{11.90}$$

Уравнения (11.89) и (11.90) подобно уравнениям (11.83) и (11.84) позволяют, используя полученное на предыдущей итерации алгоритма значение $\varphi_N^{-1}(k-1)$, определить $\varphi_{N+1}^{-1}(k)$ с помощью (11.89), а затем из (11.90) определить $\varphi_N^{-1}(k)$ при условии, что другие переменные в этих уравнениях известны, т. е. вычислены.

Рассмотренные в разделах 11.4—11.6 соотношения лежат в основе трех основных разновидностей быстрых RLS-алгоритмов адаптивной фильтрации: FK, FTF и FAEST.

11.7. Быстрый алгоритм Калмана

В основе FK-алгоритма [125] находится рекурсивное вычисление вектора $\mathbf{g}_N(k)$ с помощью соотношений (11.64) и (11.65). Для этого сначала с помощью уравнения (11.64) вычисляется вектор $\mathbf{g}_{N+1}(k)$, который можно представить в виде двухкомпонентного вектора как

$$\mathbf{g}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix}. \quad (11.91)$$

Поскольку векторы $\mathbf{g}_{N+1}(k)$, определяемые уравнениями (11.64) и (11.65), равны друг другу, то можно записать

$$\mathbf{g}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_N(k) \\ 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_N^b(k) \\ 1 \end{bmatrix} \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k). \quad (11.92)$$

Из уравнения (11.92) следует, что

$$\tilde{q}_N(k) = \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k), \quad (11.93)$$

а

$$\tilde{\mathbf{q}}_N(k) = \mathbf{g}_N(k) - \mathbf{h}_N^b(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k) = \mathbf{g}_N(k) - \mathbf{h}_N^b(k) \tilde{q}_N(k). \quad (11.94)$$

В уравнении (11.94) вектор $\mathbf{h}_N^b(k)$ является неизвестным, поскольку согласно (11.39) для своего вычисления он также требует пока еще не определенного вектора $\mathbf{g}_N(k)$. Для нахождения $\mathbf{g}_N(k)$ в уравнение (11.94) необходимо подставить (11.39). Тогда

$$\begin{aligned} \tilde{\mathbf{q}}_N(k) &= \mathbf{g}_N(k) - \mathbf{h}_N^b(k) \left[E_{N,\min}^b(k) \right]^{-1} \mathbf{e}_N^b(k) = \\ &= \mathbf{g}_N(k) - \mathbf{h}_N^b(k) \tilde{q}_N(k) = \\ &= \mathbf{g}_N(k) - \tilde{q}_N(k) \left[\mathbf{h}_N^b(k-1) + \mathbf{g}_N(k) \alpha_N^{b*}(k) \right] = \\ &= \mathbf{g}_N(k) - \mathbf{h}_N^b(k-1) \tilde{q}_N(k) - \mathbf{g}_N(k) \alpha_N^{b*}(k) \tilde{q}_N(k) = \\ &= -\mathbf{g}_N(k) \left[1 - \alpha_N^{b*}(k) \tilde{q}_N(k) \right] - \mathbf{h}_N^b(k-1) \tilde{q}_N(k), \end{aligned} \quad (11.95)$$

откуда

$$\mathbf{g}_N(k) = \left[\tilde{\mathbf{q}}_N(k) + \mathbf{h}_N^b(k-1) \tilde{q}_N(k) \right] \left[1 - \alpha_N^{b*}(k) \tilde{q}_N(k) \right]^{-1}. \quad (11.96)$$

Таким образом, уравнения (11.64) и (11.96) позволяют рекурсивно вычислять вектор коэффициентов Калмана $\mathbf{g}_N(k)$ при наличии значений других переменных, входящих в эти уравнения.

Вычислительная процедура FK-алгоритма, приведенного в табл. 11.1, на каждой итерации k последовательно использует уравнения (11.17), (11.40), (11.16), (11.19), (11.18), (11.64), (11.96), (11.39), (6.81) и (6.80).

Особенности инициализации переменной $E_{N,\min}^f(0)$ в рассматриваемом алгоритме, а также переменной $E_{N,\min}^b(0)$ в других алгоритмах настоящей главы будут приведены в разд. 14.8.

Вычислительная сложность FK-алгоритма равна $10N$ операциям умножения, $9N$ операциям сложения и двум операциям деления, т. е. $O(N)$, что позволяет считать данный алгоритм вычислительно эффективным (по сравнению с RLS-алгоритмами из глав 6—9) или быстрым.

Таблица 11.1. FK-алгоритм

Вычисления	Ссылки
Initialization: $E_{N,\min}^f(0) = \delta^2$, $\mathbf{h}_N^f(0) = \mathbf{0}_N$, $\mathbf{h}_N^b(0) = \mathbf{0}_N$, $\mathbf{g}_N(0) = \mathbf{0}_N$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	(11.1.0)
For $k = 1, 2, \dots, K$	
$\alpha_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k-1)\mathbf{x}_N(k-1)$	(11.1.1)
$\alpha_N^b(k) = x(k-N) - \mathbf{h}_N^{bH}(k-1)\mathbf{x}_N(k)$	(11.1.2)
$\mathbf{h}_N^f(k) = \mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1)\alpha_N^{f*}(k)$	(11.1.3)
$e_N^f(k) = x(k) - \mathbf{h}_N^{fH}(k)\mathbf{x}_N(k-1)$	(11.1.4)
$E_{N,\min}^f(k) = \lambda E_{N,\min}^f(k-1) + e_N^f(k)\alpha_N^{f*}(k)$	(11.1.5)
$\mathbf{g}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{g}_N(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k) \end{bmatrix} \frac{e_N^f(k)}{E_{N,\min}^f(k)}$	(11.1.6)
$\mathbf{g}_N(k) = \frac{\tilde{\mathbf{q}}_N(k) + \mathbf{h}_N^b(k-1)\tilde{q}_N(k)}{1 - \alpha_N^{b*}(k)\tilde{q}_N(k)}$	(11.1.7)
$\mathbf{h}_N^b(k) = \mathbf{h}_N^b(k-1) + \mathbf{g}_N(k)\alpha_N^{b*}(k)$	(11.1.8)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$	(11.1.9)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k)\alpha_N^*(k)$	(11.1.10)
End for k	

11.8. FTF-алгоритм

FTF-алгоритм [127] основан на вычислении априорных векторов коэффициентов Калмана $\mathbf{t}_N(k)$ и отношений ошибок $\varphi_N(k)$. Апостериорные ошибки, также требуемые для вычисления энергий ошибок линейного предсказания, при этом определяются с помощью $\varphi_N(k)$ на основе соотношений (11.77)–(11.79).

Рекурсивное вычисление вектора $\mathbf{t}_N(k)$ осуществляется посредством соотношений (11.68) и (11.69). Для этого с помощью уравнения (11.68), используя известный вектор $\mathbf{t}_N(k-1)$, вычисляется вектор $\mathbf{t}_{N+1}(k)$, который можно представить в виде двухкомпонентного вектора как

$$\mathbf{t}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix}. \quad (11.97)$$

Поскольку векторы $\mathbf{t}_{N+1}(k)$, определяемые уравнениями (11.68) и (11.69), равны, то

$$\mathbf{t}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} \mathbf{t}_N(k) \\ 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_N^b(k-1) \\ 1 \end{bmatrix} \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_N^b(k). \quad (11.98)$$

Из уравнения (11.98) следует, что

$$\tilde{q}_N(k) = \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_N^b(k), \quad (11.99)$$

$$\tilde{q}_N(k) = \mathbf{t}_N(k) - \mathbf{h}_N^b(k) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_N^b(k) = \mathbf{t}_N(k) - \mathbf{h}_N^b(k) \tilde{q}_N(k), \quad (11.100)$$

а из уравнения (11.100) — что

$$\mathbf{t}_N(k) = \tilde{q}_N(k) + \mathbf{h}_N^b(k-1) \tilde{q}_N(k). \quad (11.101)$$

Таким образом, уравнения (11.68) и (11.101) позволяют рекурсивно вычислять вектор коэффициентов Калмана $\mathbf{t}_N(k)$ при наличии значений других переменных, входящих в эти уравнения. Кроме того, переменную $\alpha_N^b(k)$ можно определять из уравнения (11.99) как

$$\alpha_N^b(k) = E_{N,\min}^b(k-1) \tilde{q}_N(k), \quad (11.102)$$

а переменную $e_N^b(k)$ — из уравнения (11.79).

Для рекурсивного вычисления $\varphi_N(k)$ преобразуем уравнения (11.83) с учетом (11.78) и (11.18) как

$$\begin{aligned} \varphi_{N+1}(k) &= \varphi_N(k-1) - e_N^{f*}(k) \left[E_{N,\min}^f(k) \right]^{-1} e_N^f(k) = \\ &= \varphi_N(k-1) - \varphi_N(k-1) \alpha_N^{f*}(k) \left[E_{N,\min}^f(k) \right]^{-1} e_N^f(k) = \\ &= \frac{\varphi_N(k-1) E_{N,\min}^f(k) - \varphi_N(k-1) \alpha_N^{f*}(k) e_N^f(k)}{E_{N,\min}^f(k)} = \\ &= \varphi_N(k-1) \frac{\left[E_{N,\min}^f(k) - \alpha_N^{f*}(k) e_N^f(k) \right]}{E_{N,\min}^f(k)} = \\ &= \varphi_N(k-1) \frac{\lambda E_{N,\min}^f(k-1)}{E_{N,\min}^f(k)}. \end{aligned} \quad (11.103)$$

Аналогичным образом преобразуем уравнение (11.84) с учетом (11.41) и (11.79) как

$$\begin{aligned} \varphi_{N+1}(k) &= \varphi_N(k) - e_N^{b*}(k) \left[E_{N,\min}^b(k) \right]^{-1} e_N^b(k) = \\ &= \varphi_N(k) - \varphi_N(k) \alpha_N^{b*}(k) \left[E_{N,\min}^b(k) \right]^{-1} e_N^b(k) = \\ &= \frac{\varphi_N(k) E_{N,\min}^b(k) - \varphi_N(k) \alpha_N^{b*}(k) e_N^b(k)}{E_{N,\min}^b(k)} = \\ &= \frac{\varphi_N(k) \left[E_{N,\min}^b(k) - \alpha_N^{b*}(k) e_N^b(k) \right]}{E_{N,\min}^b(k)} = \\ &= \varphi_N(k) \frac{\lambda E_{N,\min}^b(k-1)}{E_{N,\min}^b(k)}, \end{aligned} \quad (11.104)$$

откуда с учетом (11.99)

$$\begin{aligned}\varphi_N(k) &= \varphi_{N+1}(k) \frac{E_{N,\min}^b(k)}{\lambda E_{N,\min}^b(k-1)} = \\ &= \varphi_{N+1}(k) \frac{[\lambda E_{N,\min}^b(k-1) + e_N^b(k) \alpha_N^{b*}(k)]}{\lambda E_{N,\min}^b(k-1)} = \\ &= \varphi_{N+1}(k) + \varphi_{N+1}(k) \frac{e_N^b(k) \tilde{q}_N^*(k)}{\lambda}\end{aligned}\quad (11.105)$$

или

$$\begin{aligned}\lambda \varphi_N(k) &= \lambda \varphi_{N+1}(k) + \varphi_{N+1}(k) e_N^b(k) \tilde{q}_N^*(k) = \\ &= \lambda \varphi_{N+1}(k) + \varphi_{N+1}(k) \alpha_N^b(k) \varphi_N(k) \tilde{q}_N^*(k).\end{aligned}\quad (11.106)$$

Из уравнения (11.106) можно определить

$$\begin{aligned}\varphi_N(k) &= \lambda \varphi_{N+1}(k) \lambda^{-1} \left[1 - \lambda^{-1} \varphi_{N+1}(k) \tilde{q}_N^*(k) \alpha_N^b(k) \right]^{-1} = \\ &= \varphi_{N+1}(k) \left[1 - \lambda^{-1} \varphi_{N+1}(k) \tilde{q}_N^*(k) \alpha_N^b(k) \right]^{-1}.\end{aligned}\quad (11.107)$$

Таким образом, уравнения (11.103) и (11.107) позволяют рекурсивно вычислять переменную $\varphi_N(k)$.

С учетом (11.81) и (11.77)–(11.79) уравнения для вычисления векторов ВК фильтров линейного предсказания и собственно адаптивного фильтра преобразуются как

$$\begin{aligned}\mathbf{h}_N^f(k) &= \mathbf{h}_N^f(k-1) + \mathbf{g}_N(k-1) \alpha_N^{f*}(k) = \mathbf{h}_N^f(k-1) + \\ &+ \lambda^{-1} \mathbf{t}_N(k-1) \varphi_N(k-1) \alpha_N^{f*}(k) = \mathbf{h}_N^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) e_N^{f*}(k),\end{aligned}\quad (11.108)$$

$$\begin{aligned}\mathbf{h}_N^b(k) &= \mathbf{h}_N^b(k-1) + \mathbf{g}_N(k) \alpha_N^{b*}(k) = \mathbf{h}_N^b(k-1) + \\ &+ \lambda^{-1} \mathbf{t}_N(k) \varphi_N(k) \alpha_N^{b*}(k) = \mathbf{h}_N^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^{b*}(k),\end{aligned}\quad (11.109)$$

$$\begin{aligned}\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k) = \mathbf{h}_N(k-1) + \\ &+ \lambda^{-1} \mathbf{t}_N(k) \varphi_N(k) \alpha_N^*(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^*(k).\end{aligned}\quad (11.110)$$

Таким образом, вычислительная процедура FTF-алгоритма приведено-го в табл. 11.2, на каждой итерации k последовательно использует уравнения (11.17), (11.78), (11.68), (11.108), (11.102), (11.101), (11.18), (11.104), (11.107), (11.79), (11.41), (11.109), (6.81), (11.77) и (11.110).

Поскольку согласно уравнению (11.74) $\varphi_N(k) = 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k)$, то при инициа-лизации $\mathbf{x}_N(0) = \mathbf{0}_N$ начальное значение $\varphi_N(0) = 1 - \mathbf{g}_N^H(0) \mathbf{0}_N = 1$. Аналогично опре-деляется начальное значение вектора (11.66) $\mathbf{t}_N(0) = \mathbf{R}_N^{-1}(-1) \mathbf{x}_N(0) = \mathbf{R}_N^{-1}(-1) \mathbf{0}_N = \mathbf{0}_N$.

Вычислительная сложность FTF-алгоритма равна $7N$ операциям умноже-ния, $7N$ операциям сложения и трем операциям деления, т. е. $O(N)$.

Таблица 11.2. FTF-алгоритм

Вычисления	Ссылки
Initialization: $E_{N,\min}^f(0) = \delta^2$, $E_{N,\min}^b(0) = \delta^2 \lambda^{-N}$, $\mathbf{h}_N^f(0) = \mathbf{0}_N$,	(11.2.0)
$\mathbf{h}_N^b(0) = \mathbf{0}_N$, $\mathbf{t}_N(0) = \mathbf{0}_N$, $\varphi_N(0) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
$\alpha_N^f(k) = \mathbf{x}(k) - \mathbf{h}_N^{fH}(k-1) \mathbf{x}_N(k-1)$	(11.2.1)
$e_N^f(k) = \alpha_N^f(k) \varphi_N(k-1)$	(11.2.2)
$\mathbf{t}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t}_N(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k-1) \end{bmatrix} \frac{\alpha_N^f(k)}{E_{N,\min}^f(k-1)}$	(11.2.3)
$\mathbf{h}_N^f(k) = \mathbf{h}_N^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) e_N^{f*}(k)$	(11.2.4)
$\alpha_N^b(k) = E_{N,\min}^b(k-1) \tilde{q}_N(k)$	(11.2.5)
$\mathbf{t}_N(k) = \tilde{\mathbf{q}}_N(k) + \mathbf{h}_N^b(k-1) \tilde{q}_N(k)$	(11.2.6)
$E_{N,\min}^f(k) = \lambda E_{N,\min}^f(k-1) + e_N^f(k) \alpha_N^{f*}(k)$	(11.2.7)
$\varphi_{N+1}(k) = \varphi_N(k-1) \frac{\lambda E_{N,\min}^f(k-1)}{E_{N,\min}^f(k)}$	(11.2.8)
$\varphi_N(k) = \frac{\varphi_{N+1}(k)}{1 - \lambda^{-1} \varphi_{N+1}(k) \tilde{q}_N(k) \alpha_N^b(k)}$	(11.2.9)
$e_N^b(k) = \alpha_N^b(k) \varphi_N(k)$	(11.2.10)
$E_{N,\min}^b(k) = \lambda E_{N,\min}^b(k-1) + e_N^b(k) \alpha_N^{b*}(k)$	(11.2.11)
$\mathbf{h}_N^b(k) = \mathbf{h}_N^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^{b*}(k)$	(11.2.12)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(11.2.13)
$e_N(k) = \alpha_N(k) \varphi_N(k)$	(11.2.14)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^*(k)$	(11.2.15)
End for k	

11.9. FAEST-алгоритм

Различие между FTF- и FAEST-алгоритмами состоит в том, что в первом рекуррентно вычисляется отношение $\varphi_N(k)$, а во втором — его обратное значение $\varphi_N^{-1}(k)$. Для этого сначала используется уравнение (11.89), а затем из (11.90) с учетом (11.99) определяется

$$\begin{aligned} \varphi_N^{-1}(k) &= \varphi_{N+1}^{-1}(k) - \lambda^{-1} \alpha_{N+1}^{b*}(k) \left[E_{N,\min}^b(k-1) \right]^{-1} \alpha_{N+1}^b(k) = \\ &= \varphi_{N+1}^{-1}(k) - \lambda^{-1} \alpha_{N+1}^{b*}(k) \tilde{q}_N(k). \end{aligned} \quad (11.111)$$

Вычислительная процедура FTF-алгоритма, приведенная в табл. 11.3 на каждой итерации k последовательно использует уравнения (11.17), (11.78),



Таблица 11.3. FAEST-алгоритм

Вычисления	Ссылки
Initialization: $E_{N,\min}^f(0) = \delta^2$, $E_{N,\min}^b(0) = \delta^2 \lambda^{-N}$, $\mathbf{h}_N^f(0) = \mathbf{0}_N$,	(11.3.0)
$\mathbf{h}_N^b(0) = \mathbf{0}_N$, $\mathbf{t}_N(0) = \mathbf{0}_N$, $\varphi_N(0) = \varphi_N^{-1}(0) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
$\alpha_N^f(k) = \mathbf{x}(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k-1)$	(11.3.1)
$e_N^f(k) = \alpha_N^f(k) \varphi_N(k-1)$	(11.3.2)
$\mathbf{t}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t}_N(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k-1) \end{bmatrix} \frac{\alpha_N^f(k)}{E_{N,\min}^f(k-1)}$	(11.3.3)
$\mathbf{h}_N^f(k) = \mathbf{h}_N^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) e_N^{f*}(k)$	(11.3.4)
$\alpha_N^b(k) = E_{N,\min}^b(k-1) \tilde{q}_N(k)$	(11.3.5)
$\mathbf{t}_N(k) = \mathbf{q}_N(k) + \mathbf{h}_N^b(k-1) \tilde{q}_N(k)$	(11.3.6)
$E_{N,\min}^f(k) = \lambda E_{N,\min}^f(k-1) + e_N^f(k) \alpha_N^{f*}(k)$	(11.3.7)
$\frac{1}{\varphi_{N+1}(k)} = \frac{1}{\varphi_N(k-1)} + \lambda^{-1} \frac{\alpha_N^{f*}(k) \alpha_N^f(k)}{E_{N,\min}^f(k-1)}$	(11.3.8)
$\frac{1}{\varphi_N(k)} = \frac{1}{\varphi_{N+1}(k)} - \lambda^{-1} \alpha_N^{b*}(k) \tilde{q}_N(k)$	(11.3.9)
$e_N^b(k) = \alpha_N^b(k) \varphi_N(k)$	(11.3.10)
$E_{N,\min}^b(k) = \lambda E_{N,\min}^b(k-1) + e_N^b(k) \alpha_N^{b*}(k)$	(11.3.11)
$\mathbf{h}_N^b(k) = \mathbf{h}_N^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^{b*}(k)$	(11.3.12)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(11.3.13)
$e_N(k) = \alpha_N(k) \varphi_N(k)$	(11.3.14)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^{*}(k)$	(11.3.15)
End for k	

(11.68), (11.108), (11.102), (11.101), (11.18), (11.89), (11.111), (11.79), (11.41), (11.109), (6.81), (11.77) и (11.110).

Так как в FTF-алгоритме начальное значение отношения ошибок инициализируется как $\varphi_N(0) = 1$, то в FAEST-алгоритме начальное значение обратного отношения ошибок также инициализируется $\varphi_N^{-1}(0) = 1$.

Вычислительная сложность многоканального FAEST-алгоритма равна $7N$ операциям умножения, $7N$ операциям сложения и двум операциям деления, т. е. $O(N)$.

Известно [127, 130], что при ограниченной разрядности вычислений в быстрых RLS-алгоритмах может наблюдаться неустойчиваое функционирование. В этих случаях можно применить стабилизированные версии таких алгоритмов

Таблица 11.4. Стабилизированный FAEST-алгоритм

Вычисления	Ссылки
Initialization: $E_{N,\min}^f(0) = \delta^2$, $E_{N,\min}^b(0) = \delta^2 \lambda^{-N}$, $\mathbf{h}_N^f(0) = \mathbf{0}_N$, $\mathbf{h}_N^b(0) = \mathbf{0}_N$, $\mathbf{t}_N(0) = \mathbf{0}_N$, $\varphi_N(0) = \varphi_N^{-1}(0) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	(11.4.0)
For $k = 1, 2, \dots, K$	
$\alpha_N^f(k) = \mathbf{x}(k) - \mathbf{h}_N^{fH}(k-1) \mathbf{x}_N(k-1)$	(11.4.1)
$e_N^f(k) = \alpha_N^f(k) \varphi_N(k-1)$	(11.4.2)
$\mathbf{t}_{N+1}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{q}_N(k) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t}_N(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k-1) \end{bmatrix} \frac{\alpha_N^f(k)}{E_{N,\min}^f(k-1)}$	(11.4.3)
$\mathbf{h}_N^f(k) = \mathbf{h}_N^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) e_N^{f*}(k)$	(11.4.4)
$E_{N,\min}^f(k) = \lambda E_{N,\min}^f(k-1) + e_N^f(k) \alpha_N^{f*}(k)$	(11.4.5)
$\frac{1}{\varphi_{N+1}(k)} = \frac{1}{\varphi_N(k-1)} + \lambda^{-1} \frac{\alpha_N^{f*}(k) \alpha_N^f(k)}{E_{N,\min}^f(k-1)}$	(11.4.6)
$\alpha_N^b(k) = \mathbf{x}(k-N) - \mathbf{h}_N^{bH}(k-1) \mathbf{x}_N(k)$	(11.4.7)
$q_N(k) = \alpha_N^b(k) / E_{N,\min}^b(k-1)$	(11.4.8)
$\tilde{\alpha}_N^b(k) = E_{N,\min}^b(k-1) \tilde{q}_N(k)$	(11.4.9)
$\alpha_N^{b(1)}(k) = K_1 \alpha_N^b(k) + (1-K_1) \tilde{\alpha}_N^b(k)$	(11.4.10)
$\alpha_N^{b(2)}(k) = K_2 \alpha_N^b(k) + (1-K_2) \tilde{\alpha}_N^b(k)$	(11.4.11)
$\alpha_N^{b(5)}(k) = K_5 \alpha_N^b(k) + (1-K_5) \tilde{\alpha}_N^b(k)$	(11.4.12)
$t_{N+1}^N(k) = K_4 q_N(k) + (1-K_4) \tilde{q}_N(k)$	(11.4.13)
$\mathbf{t}_N(k) = \tilde{\mathbf{q}}_N(k) + \mathbf{h}_N^b(k-1) t_{N+1}^N(k)$	(11.4.14)
$\frac{1}{\hat{\varphi}_N(k)} = 1 + \lambda^{-1} \mathbf{t}_N^H(k) \mathbf{x}_N(k)$	(11.4.15)
$\frac{1}{\tilde{\varphi}_N(k)} = \frac{1}{\varphi_{N+1}(k)} - \lambda^{-1} \alpha_N^{b(5)*}(k) \tilde{q}_N(k)$	(11.4.16)
$\varphi_N(k) = \left[K_3 \frac{1}{\hat{\varphi}_N(k)} + (1-K_3) \frac{1}{\tilde{\varphi}_N(k)} \right]^{-1}$	(11.4.17)
$e_N^{b(1)}(k) = \alpha_N^{b(1)}(k) \tilde{\varphi}_N(k)$	(11.4.18)
$e_N^{b(2)}(k) = \alpha_N^{b(2)}(k) \tilde{\varphi}_N(k)$	(11.4.19)
$E_N^b(k) = \lambda E_N^b(k-1) + e_N^{b(2)}(k) \alpha_N^{b(2)*}(k)$	(11.4.20)
$\mathbf{h}_N^b(k) = \mathbf{h}_N^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^{b(1)*}(k)$	(11.4.21)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(11.4.22)
$e_N(k) = \alpha_N(k) \varphi_N(k)$	(11.4.23)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^*(k)$	(11.4.24)
End for k	



[95, 135], быстрые алгоритмы на основе обратного QR-разложения [66] или лестничные RLS-алгоритмы, рассматриваемые в главе 12.

В основе метода [95] стабилизации FAEST-алгоритма, приведенного в табл. 11.4, лежит альтернативное вычисление одних и тех же переменных, таких как ошибки линейного предсказания назад и обратные отношения ошибок линейного предсказания, которые затем комбинируются с некоторыми весами K_1, \dots, K_5 перед применением в других вычислениях. Значения этих весов могут быть выбраны близкими к значениям, рекомендуемым в [95], как $K_1=1.5, K_2=2.5, K_3=1, K_4=0, K_5=1$. Различия в вычислениях в FAEST- и в стабилизированном FAEST-алгоритме легко увидеть, сравнивая содержание табл. 11.3 и табл. 11.4.

Вычислительная сложность алгоритма (см. табл. 11.4) равна $9N$ операциям умножения, $9N$ операциям сложения и трем операциям деления, т. е. $O(N)$. Она несколько выше, чем у FTF- и FAEST-алгоритма, что является следствием дополнительных вычислительных затрат, обеспечивающих устойчивое функционирование адаптивного фильтра.

11.10. Выводы по главе

Оценка вычислительной сложности RLS-алгоритмов, рассмотренных в настоящей главе, из табл. 11.1—11.4 равна $O(N)$, что позволяет их считать вычислительно эффективными или быстрыми по сравнению с RLS-алгоритмами с вычислительной сложностью $O(N^2)$, рассмотренными в главах 6—9.

ГЛАВА 12

БЫСТРЫЕ ЛЕСТНИЧНЫЕ АЛГОРИТМЫ

В настоящей главе рассматриваются вопросы построения быстрых лестничных RLS-алгоритмов. Приводятся вычислительные процедуры десяти разновидностей таких алгоритмов. Лестничные алгоритмы, подобно алгоритмам, рассмотренным в главе 11, также характеризуются линейной вычислительной сложностью.

12.1. Введение

Лестничные алгоритмы адаптивной фильтрации относятся к классу быстрых RLS-алгоритмов, т. е. они характеризуются вычислительной сложностью $O(N)$ арифметических операций на одну итерацию. Эти алгоритмы получили свое название из-за структуры процедур вычисления ошибок линейного предсказания и моделирования обрабатываемых сигналов, напоминающей по форме лестницу («ladder»), состоящую из отдельных звеньев. Другое широко используемое название таких алгоритмов — это «решетчатые» («lattice») из-за структуры процедур вычислений ошибок линейного предсказания, напоминающей решетку или косичку с перекрещивающимися связями. Иногда используется комбинация этих названий. Далее для простоты будет использоваться первое название как наиболее часто используемое применительно к адаптивным фильтрам.

Подобно RLS-алгоритмам на основе прямого QR-разложения, рассмотренным в главе 8, в лестничных алгоритмах отсутствуют вычисления ВК адаптивного фильтра в явном виде и, как следствие, отсутствует в явном виде вычисление выходного сигнала адаптивного фильтра $y_N(k)$. Все вычисления в лестничных алгоритмах являются скалярными и выполняются на каждой k -й итерации по времени в течение N шагов при изменении числа ВК адаптивного фильтра от 1 до N .

Таким образом, лестничные алгоритмы базируются на рекурсивных вычислениях не только по времени, но и по числу ВК адаптивного фильтра, что отличает их от быстрых RLS-алгоритмов, рассмотренных в главе 11.

Основные разновидности одноканальных адаптивных лестничных алгоритмов с возрастающим окном представлены в работе [67]. Это алгоритмы без обратных связей и с обратными связями на основе априорных и апостериорных ошибок линейного предсказания входного сигнала и моделирования требуемого сигнала, без обратных связей и с обратными связями на основе только априорных или только апостериорных ошибок, нормализованный алгоритм и алгоритмы на основе QR-разложения, в том числе в пространстве состояний с операциями извлечения квадратного корня и без таких операций.

Вычислительная сложность лестничных алгоритмов несколько больше сложности быстрых RLS-алгоритмов, рассмотренных в главе 11. Однако лестничные алгоритмы известны своей устойчивостью [129], что способствует их широкому использованию в приложениях. Кроме того, каскадная структура вычислений, характерная для лестничных алгоритмов, обеспечивает не только их эффективную программную, но и аппаратную реализацию в виде СБИС.

В основе дальнейшего изложения лежат материалы, представленные в [15, 20, 22, 29, 30, 67, 129, 136—146].

12.2. Рекурсивное вычисление энергий ошибок линейного предсказания

Быстрые RLS-алгоритмы из главы 11 были получены на основе теории линейного предсказания, в которой центральную роль играют выражения обратных корреляционных матриц $\mathbf{R}_{N+1}^{-1}(k)$ с числом элементов $(N+1) \times (N+1)$, см. уравнения (11.58) и (11.63), вычисление которых основано на использовании матриц $\mathbf{R}_N^{-1}(k)$ и $\mathbf{R}_N^{-1}(k-1)$ с числом элементов $N \times N$, энергий ошибок линейного предсказания $E_{N,\min}^f(k)$ и $E_{N,\min}^b(k)$, а также векторов ВК фильтров линейного предсказания $\mathbf{a}_{N+1}(k)$ и $\mathbf{b}_{N+1}(k)$. Посредством этих уравнений устанавливаются соотношения между векторными переменными с числом элементов N и $N+1$. Данные соотношения справедливы для любых значений N , что позволяет их использовать для вычислений в лестничных фильтрах.

Рассмотрим соотношения, связывающие векторы $\mathbf{a}_n(k)$ и $\mathbf{a}_{n+1}(k)$, а также векторы $\mathbf{b}_n(k)$ и $\mathbf{b}_{n+1}(k)$, где $n=1, \dots, N$.

Согласно уравнению (11.9) вектор ВК фильтра линейного предсказания $\mathbf{h}_n^f(k)$ определяется как $\mathbf{h}_n^f(k) = \mathbf{R}_n^{-1}(k-1) \mathbf{r}_n^f(k)$ в соответствии с уравнением (11.63)

$$\mathbf{R}_n^{-1}(k-1) = \begin{bmatrix} \mathbf{R}_{n-1}^{-1}(k-1) & \mathbf{0}_{n-1} \\ \mathbf{0}_{n-1}^T & 0 \end{bmatrix} + \mathbf{b}_n(k-1) \left[E_{n-1,\min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1), \quad (12.1)$$

а вектор $\mathbf{r}_n^f(k)$ согласно уравнению (11.6) определяется как

$$\begin{aligned}
 \mathbf{r}_n^f(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_n(i-1) x^*(i) = \\
 &= \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} x(i-1) \\ x(i-2) \\ \vdots \\ x(i-n+1) \\ x(i-n) \end{bmatrix} x^*(i) = \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n-1}(i-1) x^*(i) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n) x^*(i) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{n-1}^f(k) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n) x^*(i) \end{bmatrix}. \quad (12.2)
 \end{aligned}$$

Используя уравнения (12.1) и (12.2), вектор $\mathbf{h}_n^f(k)$ можно представить как

$$\begin{aligned}
 \mathbf{h}_n^f(k) &= \left\{ \begin{bmatrix} \mathbf{R}_{n-1}^{-1}(k-1) & \mathbf{0}_{n-1} \\ \mathbf{0}_{n-1}^T & 0 \end{bmatrix} + \mathbf{b}_n(k-1) \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \right\} \times \\
 &\times \begin{bmatrix} \mathbf{r}_{n-1}^f(k) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n) x^*(i) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{n-1}^{-1}(k-1) \mathbf{r}_{n-1}^f(k) \\ 0 \end{bmatrix} + \\
 &+ \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n(k-1) \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \\
 &= \begin{bmatrix} \mathbf{h}_{n-1}^f(k) \\ 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-1) \\ 1 \end{bmatrix} \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k). \quad (12.3)
 \end{aligned}$$

Тогда из уравнений (11.21) и (12.3) следует, что

$$\begin{aligned}
 \mathbf{a}_{n+1}(k) &= \begin{bmatrix} 1 \\ -\mathbf{h}_n^f(k) \end{bmatrix} = \begin{bmatrix} 1 \\ -\mathbf{h}_{n-1}^f(k) \end{bmatrix} - \begin{bmatrix} 0 \\ -\mathbf{h}_{n-1}^b(k-1) \end{bmatrix} \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \\
 &= \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} \Gamma_n^f(k), \quad (12.4)
 \end{aligned}$$

где

$$\Gamma_n^f(k) = -\left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k). \quad (12.5)$$

Аналогично определяются соотношения, связывающие векторы $\mathbf{b}_n(k)$ и $\mathbf{b}_{n+1}(k)$. Для этого используются уравнения (11.34) $\mathbf{h}_N^b(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N^b(k)$, (11.58)

$$\mathbf{R}_n^{-1}(k) = \begin{bmatrix} 0 & \mathbf{0}_{n-1}^T \\ \mathbf{0}_{n-1} & \mathbf{R}_{n-1}^{-1}(k-1) \end{bmatrix} + \mathbf{a}_n(k) \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \quad (12.6)$$

и (11.31)

$$\mathbf{r}_n^b(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_n(i) x^*(i-n) = \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} x(i) \\ x(i-1) \\ \vdots \\ x(i-n+2) \\ x(i-n+1) \end{bmatrix} x^*(i-n) = \quad (12.7)$$

$$= \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i-n) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n-1}(i-1) x^*(i-n) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i-n) \\ \mathbf{r}_{n-1}^b(k-1) \end{bmatrix}.$$

Затем с помощью уравнений (12.6) и (12.7) вектор $\mathbf{h}_n^b(k)$ можно представить как

$$\begin{aligned} \mathbf{h}_n^b(k) &= \left\{ \begin{bmatrix} 0 & \mathbf{0}_{n-1}^T \\ \mathbf{0}_{n-1} & \mathbf{R}_{n-1}^{-1}(k-1) \end{bmatrix} + \mathbf{a}_n(k) \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \right\} \times \\ &\times \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i-n) \\ \mathbf{r}_{n-1}^b(k-1) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{R}_{n-1}^{-1}(k-1) \mathbf{r}_{n-1}^b(k-1) \end{bmatrix} + \mathbf{a}_n(k) \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \mathbf{r}_n^b(k) = \\ &= \begin{bmatrix} 0 \\ \mathbf{h}_{n-1}^b(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_{n-1}^f(k) \end{bmatrix} \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \mathbf{r}_n^b(k). \end{aligned} \quad (12.8)$$

Тогда из уравнений (11.44) и (12.8) следует, что

$$\begin{aligned} \mathbf{b}_{n+1}(k) &= \begin{bmatrix} -\mathbf{h}_n^b(k) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{h}_{n-1}^b(k-1) \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ -\mathbf{h}_{n-1}^f(k) \\ 0 \end{bmatrix} \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \mathbf{r}_n^b(k) = \\ &= \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} - \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} \left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \mathbf{r}_n^b(k) = \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} \Gamma_n^b(k), \end{aligned} \quad (12.9)$$

где

$$\Gamma_n^b(k) = -\left[E_{n-1, \min}^f(k) \right]^{-1} \mathbf{a}_n^H(k) \mathbf{r}_n^b(k). \quad (12.10)$$

Выражения (12.4) и (12.9) далее будут использоваться для получения уравнений вычисления ошибок линейного предсказания для фильтров с изменяемым на единицу числом ВК. Для этого также необходимо получить выражения для вычисления переменных $E_{n-1, \min}^f(k)$, $E_{n-1, \min}^b(k)$, $\Gamma_n^f(k)$ и $\Gamma_n^b(k)$, используемых в указанных вычислениях. Переменные $\Gamma_n^f(k)$ и $\Gamma_n^b(k)$ называются коэффициентами отражения.

Для вычисления переменных $E_{n-1, \min}^f(k)$ и $\Gamma_n^f(k)$ используем выражения (11.26), (11.48) и (11.25). Из (11.26) следует, что

$$\mathbf{R}_{n+1}(k) \mathbf{a}_{n+1}(k) = \begin{bmatrix} E_{n, \min}^f(k) \\ \mathbf{0}_n \end{bmatrix}, \quad (12.11)$$

$$\mathbf{R}_n(k) \mathbf{a}_n(k) = \begin{bmatrix} E_{n-1, \min}^f(k) \\ \mathbf{0}_{n-1} \end{bmatrix}, \quad (12.12)$$

а из (11.25) и (11.48) — что

$$\mathbf{R}_{n+1}(k) = \begin{bmatrix} \mathbf{R}_n(k) & \mathbf{r}_n^b(k) \\ \mathbf{r}_n^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} = \begin{bmatrix} \sigma^{f2}(k) & \mathbf{r}_n^{fH}(k) \\ \mathbf{r}_n^f(k) & \mathbf{R}_n(k-1) \end{bmatrix}. \quad (12.13)$$

Используя уравнения (12.4), (12.11)–(12.13), получим, что

$$\begin{aligned} \mathbf{R}_{n+1}(k) \mathbf{a}_{n+1}(k) &= \mathbf{R}_{n+1}(k) \left\{ \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} \Gamma_n^f(k) \right\} = \\ &= \begin{bmatrix} \mathbf{R}_n(k) & \mathbf{r}_n^b(k) \\ \mathbf{r}_n^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma^{f2}(k) & \mathbf{r}_n^{fH}(k) \\ \mathbf{r}_n^f(k) & \mathbf{R}_n(k-1) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} \Gamma_n^f(k) = \\ &= \begin{bmatrix} \mathbf{R}_n(k) \mathbf{a}_n(k) \\ \mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) \end{bmatrix} + \begin{bmatrix} \mathbf{r}_n^{fH}(k) \mathbf{b}_n(k-1) \\ \mathbf{R}_n(k-1) \mathbf{b}_n(k-1) \end{bmatrix} \Gamma_n^f(k) = \\ &= \begin{bmatrix} E_{n-1, \min}^f(k) \\ \mathbf{0}_{n-1} \end{bmatrix} + \begin{bmatrix} \mathbf{r}_n^{fH}(k) \mathbf{b}_n(k-1) \\ \mathbf{0}_{n-1} \end{bmatrix} \Gamma_n^f(k) = \begin{bmatrix} E_{n, \min}^f(k) \\ \mathbf{0}_{n-1} \\ 0 \end{bmatrix}. \end{aligned} \quad (12.14)$$

Для получения первого слагаемого в (12.14) использована левая часть уравнения (12.13), а второго — правая. Из (12.14) следует, что (см. верхнюю строку в последней части уравнения)

$$E_{n, \min}^f(k) = E_{n-1, \min}^f(k) + \mathbf{r}_n^{fH}(k) \mathbf{b}_n(k-1) \Gamma_n^f(k) \quad (12.15)$$

и (см. нижнюю строку)

$$\begin{aligned} \mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) + E_{n-1, \min}^b(k-1) \Gamma_n^f(k) &= \mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) - \\ &- E_{n-1, \min}^b(k-1) \left[E_{n-1, \min}^b(k-1) \right]^{-1} \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \\ &= \mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) - \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = 0. \end{aligned} \quad (12.16)$$

В уравнении (12.16) также использовано соотношение (12.5).

Из (12.16) следует, что

$$\mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) = \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \left[\mathbf{r}_n^{fH}(k) \mathbf{b}_n(k-1) \right]^* = \Delta_{n-1}(k), \quad (12.17)$$

из уравнений (12.5) и (12.17) следует, что

$$\Gamma_n^f(k) = - \left[E_{n-1, \min}^b(k-1) \right]^{-1} \Delta_{n-1}(k), \quad (12.18)$$

а из (12.15), (12.17) и (12.18) — что

$$\begin{aligned} E_{n, \min}^f(k) &= E_{n-1, \min}^f(k) + \mathbf{r}_n^{fH}(k) \mathbf{b}_n(k-1) \Gamma_n^f(k) = \\ &= E_{n-1, \min}^f(k) + \Delta_{n-1}^*(k) \left\{ - \left[E_{n-1, \min}^b(k-1) \right]^{-1} \Delta_{n-1}(k) \right\} = \\ &= E_{n-1, \min}^f(k) - \Delta_{n-1}^*(k) \left[E_{n-1, \min}^b(k-1) \right]^{-1} \Delta_{n-1}(k). \end{aligned} \quad (12.19)$$

Для вычисления переменных $E_{n-1, \min}^b(k)$ и $\Gamma_n^b(k)$ используем выражение (11.49), из которого следует, что

$$\mathbf{R}_{n+1}(k)\mathbf{b}_{n+1}(k) = \begin{bmatrix} \mathbf{0}_n \\ E_{n,\min}^b(k) \end{bmatrix}, \quad (12.20)$$

$$\mathbf{R}_n(k)\mathbf{b}_n(k) = \begin{bmatrix} \mathbf{0}_{n-1} \\ E_{n-1,\min}^b(k) \end{bmatrix}. \quad (12.21)$$

Используя (12.9), (12.20), (12.21) и (12.13), получим

$$\begin{aligned} \mathbf{R}_{n+1}(k)\mathbf{b}_{n+1}(k) &= \mathbf{R}_{n+1}(k)\left\{\begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix}\Gamma_n^b(k)\right\} = \\ &= \begin{bmatrix} \sigma^{f2}(k) & \mathbf{r}_n^{fH}(k) \\ \mathbf{r}_n^f(k) & \mathbf{R}_n(k-1) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{R}_n(k) & \mathbf{r}_n^b(k) \\ \mathbf{r}_n^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix}\Gamma_n^b(k) = \\ &= \begin{bmatrix} \mathbf{r}_n^f(k)\mathbf{b}_n(k-1) \\ \mathbf{R}_n(k-1)\mathbf{b}_n(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{R}_n(k)\mathbf{a}_n(k) \\ \mathbf{r}_n^{bH}(k)\mathbf{a}_n(k) \end{bmatrix}\Gamma_n^b(k) = \\ &= \begin{bmatrix} \mathbf{r}_n^f(k)\mathbf{b}_n(k-1) \\ \mathbf{0}_{n-1} \end{bmatrix} + \begin{bmatrix} E_{n-1,\min}^f(k) \\ \mathbf{r}_n^{bH}(k)\mathbf{a}_n(k) \end{bmatrix}\Gamma_n^b(k) = \begin{bmatrix} 0 \\ \mathbf{0}_{n-1} \end{bmatrix} = \begin{bmatrix} E_{n,\min}^b(k) \\ E_{n,\min}^b(k) \end{bmatrix}. \end{aligned} \quad (12.22)$$

Здесь в первом слагаемом использована правая часть уравнения (12.13), а во втором — левая.

Из (12.22) следует, что (см. нижнюю строку в последней части уравнения)

$$E_{n,\min}^b(k) = E_{n-1,\min}^b(k-1) + \mathbf{r}_n^{bH}(k)\mathbf{a}_n(k)\Gamma_n^b(k) \quad (12.23)$$

и (см. верхнюю строку)

$$\begin{aligned} \mathbf{r}_n^f(k)\mathbf{b}_n(k-1) + E_{n-1,\min}^f(k)\Gamma_n^b(k) &= \mathbf{r}_n^f(k)\mathbf{b}_n(k-1) - \\ &- E_{n-1,\min}^f(k)\left[E_{n-1,\min}^f(k)\right]^{-1}\mathbf{a}_n^H(k)\mathbf{r}_n^b(k) = \mathbf{r}_n^f(k)\mathbf{b}_n(k-1) - \mathbf{a}_n^H(k)\mathbf{r}_n^b(k) = 0. \end{aligned} \quad (12.24)$$

В (12.24) также использовано выражение (12.10).

Из (12.24) и (12.17) следует, что

$$\mathbf{r}_n^f(k)\mathbf{b}_n(k-1) = \mathbf{a}_n^H(k)\mathbf{r}_n^b(k) = \left[\mathbf{r}_n^{bH}(k)\mathbf{a}_n(k)\right]^* = \Delta_{n-1}^*(k), \quad (12.25)$$

из (12.10) и (12.25) следует, что

$$\Gamma_n^b(k) = -\left[E_{n-1,\min}^f(k)\right]^{-1}\Delta_{n-1}^*(k), \quad (12.26)$$

а из (12.23), (12.25) и (12.26) — что

$$\begin{aligned} E_{n,\min}^b(k) &= E_{n-1,\min}^b(k-1) + \mathbf{r}_n^{bH}(k)\mathbf{a}_n(k)\Gamma_n^b(k) = \\ &= E_{n-1,\min}^b(k-1) + \Delta_{n-1}(k)\left\{-\left[E_{n-1,\min}^f(k)\right]^{-1}\Delta_{n-1}^*(k)\right\} = \\ &= E_{n-1,\min}^b(k-1) - \Delta_{n-1}(k)\left[E_{n-1,\min}^f(k)\right]^{-1}\Delta_{n-1}^*(k). \end{aligned} \quad (12.27)$$

12.3. Рекурсивное вычисление ошибок линейного предсказания и коэффициентов отражения

Для произвольного значения n справедливы ряд соотношений, устанавливаемых с помощью уравнений (11.1), (11.17), (11.19), (11.40) и (11.42). Это $\mathbf{x}_{n+1}(k) = \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_n(k) \\ x(k-n) \end{bmatrix}$, $\alpha_n^f(k) = \mathbf{a}_{n+1}^H(k-1) \mathbf{x}_{n+1}(k)$, $e_n^f(k) = \mathbf{a}_{n+1}^H(k) \mathbf{x}_{n+1}(k)$, $\alpha_n^b(k) = \mathbf{b}_{n+1}^H(k-1) \mathbf{x}_{n+1}(k)$ и $e_n^b(k) = \mathbf{b}_{n+1}^H(k) \mathbf{x}_{n+1}(k)$. С их помощью, а также с помощью уравнений (12.4) и (12.9) ошибки линейного предсказания вычисляются как

$$\begin{aligned} \alpha_n^f(k) &= \mathbf{a}_{n+1}^H(k-1) \mathbf{x}_{n+1}(k) = \left\{ \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{b}_n(k-2) \end{bmatrix} \Gamma_n^f(k-1) \right\}^H \mathbf{x}_{n+1}(k) = \\ &= \left[\begin{bmatrix} \mathbf{a}_n^H(k-1), 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_n(k) \\ x(k-n) \end{bmatrix} + \Gamma_n^{f*}(k-1) \begin{bmatrix} 0, \mathbf{b}_n^H(k-2) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} \right] = \\ &= \mathbf{a}_n^H(k-1) \mathbf{x}_n(k) + \Gamma_n^{f*}(k) \mathbf{b}_n^H(k-2) \mathbf{x}_n(k-1) = \\ &= \alpha_{n-1}^f(k) + \Gamma_n^{f*}(k-1) \alpha_{n-1}^b(k-1), \end{aligned} \quad (12.28)$$

$$\begin{aligned} e_n^f(k) &= \mathbf{a}_{n+1}^H(k) \mathbf{x}_{n+1}(k) = \left\{ \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} \Gamma_n^f(k) \right\}^H \mathbf{x}_{n+1}(k) = \\ &= \left[\begin{bmatrix} \mathbf{a}_n^H(k), 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_n(k) \\ x(k-n) \end{bmatrix} + \Gamma_n^{f*}(k) \begin{bmatrix} 0, \mathbf{b}_n^H(k-1) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} \right] = \\ &= \mathbf{a}_n^H(k) \mathbf{x}_n(k) + \Gamma_n^{f*}(k) \mathbf{b}_n^H(k-1) \mathbf{x}_n(k-1) = e_{n-1}^f(k) + \Gamma_n^{f*}(k) e_{n-1}^b(k-1), \end{aligned} \quad (12.29)$$

$$\begin{aligned} \alpha_n^b(k) &= \mathbf{b}_{n+1}^H(k-1) \mathbf{x}_{n+1}(k) = \left\{ \begin{bmatrix} 0 \\ \mathbf{b}_n(k-2) \end{bmatrix} + \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} \Gamma_n^b(k-1) \right\}^H \mathbf{x}_{n+1}(k) = \\ &= \left[\begin{bmatrix} 0, \mathbf{b}_n^H(k-2) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} + \Gamma_n^{b*}(k-1) \begin{bmatrix} \mathbf{a}_n^H(k-1), 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_n(k) \\ x(k-n) \end{bmatrix} \right] = \\ &= \mathbf{b}_n^H(k-2) \mathbf{x}_n(k-1) + \Gamma_n^{b*}(k-1) \mathbf{a}_n^H(k-1) \mathbf{x}_n(k) = \\ &= \alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k), \end{aligned} \quad (12.30)$$

$$\begin{aligned} e_n^b(k) &= \mathbf{b}_{n+1}^H(k) \mathbf{x}_{n+1}(k) = \left\{ \begin{bmatrix} 0 \\ \mathbf{b}_n(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{a}_n(k) \\ 0 \end{bmatrix} \Gamma_n^b(k) \right\}^H \mathbf{x}_{n+1}(k) = \\ &= \left[\begin{bmatrix} 0, \mathbf{b}_n^H(k-1) \end{bmatrix} \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} + \Gamma_n^{b*}(k) \begin{bmatrix} \mathbf{a}_n^H(k), 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_n(k) \\ x(k-n) \end{bmatrix} \right] = \\ &= \mathbf{b}_n^H(k-1) \mathbf{x}_n(k-1) + \Gamma_n^{b*}(k) \mathbf{a}_n^H(k) \mathbf{x}_n(k) = \\ &= e_{n-1}^b(k-1) + \Gamma_n^{b*}(k) e_{n-1}^f(k). \end{aligned} \quad (12.31)$$

Следовательно, уравнения (12.28)–(12.31) позволяют рекурсивно вычислять ошибки линейного предсказания с помощью фильтра с числом ВК, равным n ,

через эти же ошибки для фильтра с числом ВК, равным $n-1$. Входящие в эти уравнения коэффициенты отражения вычисляются через энергию ошибок линейного предсказания и переменную $\Delta_n(k)$. Энергия ошибок линейного предсказания также вычисляется рекурсивно с помощью уравнений (12.19) и (12.27). Поэтому для расчета коэффициентов отражения с помощью уравнений (12.5) и (12.10) необходимо также получить процедуры рекурсивного вычисления входящей в них переменной $\Delta_{n-1}(k)$.

Согласно уравнениям (12.14) и (12.25)

$$\begin{aligned} \left[0, \mathbf{b}_n^H(k-1) \right] \mathbf{R}_{n+1}(k) &= \begin{bmatrix} \mathbf{r}_n^f(k) \mathbf{b}_n(k-1) \\ \mathbf{0}_{n-1} \\ E_{n-1,\min}^b(k-1) \end{bmatrix}^H = \begin{bmatrix} \Delta_{n-1}^*(k) \\ \mathbf{0}_{n-1} \\ E_{n-1,\min}^b(k-1) \end{bmatrix}^H = \\ &= \left[\Delta_{n-1}(k), \mathbf{0}_{n-1}^T, E_{n-1,\min}^b(k-1) \right]. \end{aligned} \quad (12.32)$$

Тогда, с учетом (12.32),

$$\begin{aligned} \left[0, \mathbf{b}_n^H(k-1) \right] \mathbf{R}_{n+1}(k) &= \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} = \\ &= \left[\Delta_{n-1}(k), \mathbf{0}_{n-1}^T, E_{n-1,\min}^b(k-1) \right] \begin{bmatrix} 1 \\ -\mathbf{h}_{n-1}^f(k-1) \\ 0 \end{bmatrix} = \Delta_{n-1}(k). \end{aligned} \quad (12.33)$$

С другой стороны, согласно уравнению (6.75)

$$\mathbf{R}_{n+1}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n+1}(i) \mathbf{x}_{n+1}^H(i) = \lambda \mathbf{R}_{n+1}(k-1) + \mathbf{x}_{n+1}(k) \mathbf{x}_{n+1}^H(k). \quad (12.34)$$

Подставляя выражения (12.34), (12.14) и (12.17) в (12.33), получим

$$\begin{aligned} \left[0, \mathbf{b}_n^H(k-1) \right] \mathbf{R}_{n+1}(k) &= \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} = \\ &= \left[0, \mathbf{b}_n^H(k-1) \right] \left[\lambda \mathbf{R}_{n+1}(k-1) + \mathbf{x}_{n+1}(k) \mathbf{x}_{n+1}^H(k) \right] \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} = \\ &= \lambda \left[0, \mathbf{b}_n^H(k-1) \right] \mathbf{R}_{n+1}(k-1) \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} + \\ &+ \left[0, \mathbf{b}_n^H(k-1) \right] \mathbf{x}_{n+1}(k) \mathbf{x}_{n+1}^H(k) \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} = \lambda \left[0, \mathbf{b}_n^H(k-1) \right] \begin{bmatrix} E_{n-1,\min}^f(k-1) \\ \mathbf{0}_{n-1} \\ \Delta_{n-1}(k-1) \end{bmatrix} + \\ &+ \left[0, \mathbf{b}_n^H(k-1) \right] \begin{bmatrix} x(k) \\ \mathbf{x}_n(k-1) \end{bmatrix} \left[\mathbf{x}_n^H(k), x^*(k-n) \right] \begin{bmatrix} \mathbf{a}_n(k-1) \\ 0 \end{bmatrix} = \\ &= \lambda \left[0, -\mathbf{h}_{n-1}^{BH}(k-1), 1 \right] \begin{bmatrix} E_{n-1,\min}^f(k-1) \\ \mathbf{0}_{n-1} \\ \Delta_{n-1}(k-1) \end{bmatrix} + \mathbf{b}_n^H(k-1) \mathbf{x}_n(k-1) \mathbf{x}_n^H(k) \mathbf{a}_n(k-1) = \\ &= \lambda \Delta_{n-1}(k-1) + e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k). \end{aligned} \quad (12.35)$$

Тогда из уравнений (12.33) и (12.35) следует, что

$$\Delta_{n-1}(k) = \lambda \Delta_{n-1}(k-1) + e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k). \quad (12.36)$$

Уравнение (12.36) может быть получено и другим способом. Для этого с учетом (11.39) $\mathbf{h}_{n-1}^b(k) = \mathbf{h}_{n-1}^b(k-1) + \mathbf{g}_{n-1}(k) \alpha_{n-1}^{b*}(k)$, (6.79) $\mathbf{g}_{n-1}(k) = \mathbf{R}_{n-1}^{-1}(k) \mathbf{x}_{n-1}(k)$ и (11.9) $\mathbf{h}_{n-1}^f(k) = \mathbf{R}_{n-1}^{-1}(k-1) \mathbf{r}_{n-1}^f(k)$ преобразуем уравнение (12.17) как

$$\begin{aligned} \Delta_{n-1}(k) &= \mathbf{b}_n^H(k-1) \mathbf{r}_n^f(k) = \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-1) \\ 1 \end{bmatrix}^H \begin{bmatrix} \mathbf{r}_{n-1}^f(k) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n) x^*(i) \end{bmatrix} = \\ &= \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-2) - \mathbf{g}_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1) \\ 1 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}^f(k-1) + \mathbf{x}_{n-1}(k-1) x^*(k) \\ \lambda r_{n-1}^f(k-1) + x(k-n) x^*(k) \end{bmatrix} = \\ &= \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-2) \\ 1 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}^f(k-1) \\ \lambda r_{n-1}^f(k-1) \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-2) \\ 1 \end{bmatrix}^H \begin{bmatrix} \mathbf{x}_{n-1}(k-1) x^*(k) \\ x(k-n) x^*(k) \end{bmatrix} + \\ &\quad + \begin{bmatrix} -\mathbf{g}_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1) \\ 0 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}^f(k-1) \\ \lambda r_{n-1}^f(k-1) \end{bmatrix} + \\ &\quad + \begin{bmatrix} -\mathbf{g}_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1) \\ 0 \end{bmatrix}^H \begin{bmatrix} \mathbf{x}_{n-1}(k-1) x^*(k) \\ x(k-n) x^*(k) \end{bmatrix} = \\ &= \lambda \mathbf{b}_n^H(k-2) \mathbf{r}_n^f(k-1) + \mathbf{b}_n^H(k-2) \mathbf{x}_n(k-1) x^*(k) - \\ &\quad - \alpha_{n-1}^b(k-1) \mathbf{g}_{n-1}^H(k-1) \lambda \mathbf{r}_{n-1}^f(k-1) - \alpha_{n-1}^b(k-1) \mathbf{g}_{n-1}^H(k-1) \mathbf{x}_{n-1}(k-1) x^*(k) = \quad (12.37) \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) x^*(k) - \\ &\quad - \alpha_{n-1}^b(k-1) \mathbf{g}_{n-1}^H(k-1) [\lambda \mathbf{r}_n^f(k-1) + \mathbf{x}_{n-1}(k-1) x^*(k)] = \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) x^*(k-1) - \alpha_{n-1}^b(k-1) \mathbf{g}_{n-1}^H(k-1) \mathbf{r}_{n-1}^f(k) = \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) [x^*(k) - \mathbf{x}_{n-1}^H(k-1) \mathbf{R}_{n-1}^{-1}(k-1) \mathbf{r}_n^f(k)] = \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) [x^*(k) - \mathbf{x}_{n-1}^H(k-1) \mathbf{h}_{n-1}^f(k)] = \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) e_{n-1}^{f*}(k). \end{aligned}$$

С учетом выражений (11.75) $\varphi_{n-1}(k-1) = e_{n-1}^f(k) / \alpha_{n-1}^f(k)$ и (11.76) $e_{n-1}^b(k) / \alpha_{n-1}^b(k) = \varphi_{n-1}(k)$ уравнение (12.37) можно преобразовать как

$$\begin{aligned} \Delta_{n-1}(k) &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) e_{n-1}^{f*}(k) = \\ &= \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k) = \quad (12.38) \\ &= \lambda \Delta_{n-1}(k-1) + e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k), \end{aligned}$$

т. е. уравнение (12.38) совпадает с уравнением (12.36).

Аналогичным образом можно получить уравнение для рекурсивного вычисления переменной $\Delta_{n-1}^*(k)$.

Из (12.14) и (12.25) следует, что

$$[\mathbf{a}_n^H(k), 0] \mathbf{R}_{n+1}(k) = \begin{bmatrix} E_{n-1,\min}^f(k) \\ \mathbf{0}_{n-1} \\ \mathbf{r}_n^{bH}(k) \mathbf{a}_n(k) \end{bmatrix}^H = [E_{n-1,\min}^f(k), \mathbf{0}_{n-1}^T, \Delta_{n-1}^*(k)]. \quad (12.39)$$

Тогда с учетом (12.39)

$$[\mathbf{a}_n^H(k), 0] \mathbf{R}_{n+1}(k) \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} = [E_{n-1,\min}^f(k), \mathbf{0}_{n-1}^T, \Delta_{n-1}^*(k)] \begin{bmatrix} 0 \\ -\mathbf{h}_{n-1}^b(k-2) \\ 1 \end{bmatrix} = \Delta_{n-1}^*(k). \quad (12.40)$$

С другой стороны, с учетом (12.34), (12.22), (12.25) уравнение (12.40) можно преобразовать как

$$\begin{aligned} & [\mathbf{a}_n^H(k), 0] \mathbf{R}_{n+1}(k) \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} = \\ & = [\mathbf{a}_n^H(k), 0] [\lambda \mathbf{R}_{n+1}(k-1) + \mathbf{x}_{n+1}(k) \mathbf{x}_{n+1}^H(k)] \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} = \\ & = \lambda [\mathbf{a}_n^H(k), 0] \mathbf{R}_{n+1}(k-1) \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} + \\ & + [\mathbf{a}_n^H(k), 0] \mathbf{x}_{n+1}(k) \mathbf{x}_{n+1}^H(k) \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} = \lambda [\mathbf{a}_n^H(k), 0] \begin{bmatrix} \mathbf{r}_n^f(k-1) \mathbf{b}_n(k-2) \\ \mathbf{0}_{n-1} \\ E_{n-1,\min}^b(k-2) \end{bmatrix} + \quad (12.41) \\ & + [\mathbf{a}_n^H(k), 0] \begin{bmatrix} \mathbf{x}_n^H(k) \\ x^*(k-n) \end{bmatrix} [\mathbf{x}^*(k), \mathbf{x}_n^H(k-1)] \begin{bmatrix} 0 \\ \mathbf{b}_n^H(k-2) \end{bmatrix} = \\ & = \lambda [1, -\mathbf{h}_{n-1}^{fH}(k), 0] \begin{bmatrix} \Delta_{n-1}^*(k-1) \\ \mathbf{0}_{n-1} \\ E_{n-1,\min}^b(k-2) \end{bmatrix} + \mathbf{a}_n^H(k) \mathbf{x}_n^H(k) \mathbf{x}_n^H(k-1) \mathbf{b}_n^H(k-2) = \\ & = \lambda \Delta_{n-1}^*(k-1) + e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1). \end{aligned}$$

Таким образом, из уравнений (12.40) и (12.41) следует, что

$$\Delta_{n-1}^*(k) = \lambda \Delta_{n-1}^*(k-1) + e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1) \quad (12.42)$$

и что выражение (12.36) можно рассматривать как комплексно-сопряженное значение выражения (12.42), и наоборот.

Следовательно, эти выражения являются формулами для рекурсивного вычисления переменной $\Delta_{n-1}(k)$ и ее комплексно-сопряженного значения, используемых в формулах расчета коэффициентов отражения (12.18) и (12.26). Выражение (12.42) также можно получить с помощью тех же приемов, что и при получении уравнения (12.37), используя другие соотношения, фигурирующие в уравнениях (12.17) и (12.25).

12.4. Рекурсивное вычисление ошибок моделирования требуемого сигнала адаптивного фильтра

Для рекурсивного вычисления ошибок $e_N(k)$ и $\alpha_N(k)$ моделирования требуемого сигнала $d(k)$ с помощью соотношений, подобных (12.28)–(12.31), необходимо получить уравнение рекурсивного вычисления вектора ВК $\mathbf{h}_N(k)$, подобное уравнениям (12.4) и (12.9).

Согласно уравнению (6.81) $\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$, а согласно уравнению (11.48)

$$\begin{aligned} \mathbf{R}_n(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_n(i) \mathbf{x}_n^H(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_{n-1}(i) \\ x(i-n+1) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{n-1}^H(i), x^*(i-n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{n-1}(k) & \mathbf{r}_{n-1}^b(k) \\ \mathbf{r}_{n-1}^{bH}(k) & \sigma^{b2}(k) \end{bmatrix}, \end{aligned} \quad (12.43)$$

где

$$\sigma^{b2}(k) = \sum_{i=1}^k \lambda^{k-i} x(i-n+1)x^*(i-n+1), \quad (12.44)$$

$$\mathbf{r}_n^b(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n-1}(i) x^*(i-n+1), \quad (12.45)$$

$$\mathbf{R}_{n-1}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n-1}(i) \mathbf{x}_{n-1}^H(i). \quad (12.46)$$

Так как

$$\mathbf{R}_n(k)\mathbf{h}_n(k) = \mathbf{r}_n(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_n(i) d^*(i) = \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) \end{bmatrix} \quad (12.47)$$

и

$$\begin{aligned} \mathbf{R}_n(k) \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_{n-1}(k) & \mathbf{r}_{n-1}^b(k) \\ \mathbf{r}_{n-1}^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{n-1}(k)\mathbf{h}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k)\mathbf{h}_{n-1}(k) \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k)\mathbf{h}_{n-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k)\mathbf{R}_{n-1}^{-1}(k)\mathbf{r}_{n-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{h}_{n-1}^{bH}(k)\mathbf{r}_{n-1}(k) \end{bmatrix}, \end{aligned} \quad (12.48)$$

то разность уравнений (12.47) и (12.48) равна

$$\begin{aligned} \mathbf{R}_n(k)\mathbf{h}_n(k) - \mathbf{R}_n(k) \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} &= \mathbf{R}_n(k) \left\{ \mathbf{h}_n(k) - \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} \right\} = \\ &= \left[\sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) \right] - \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{h}_{n-1}^{bH}(k)\mathbf{r}_{n-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n-1} \\ \kappa_{n-1}(k) \end{bmatrix}, \end{aligned} \quad (12.49)$$

где, подобно уравнениям (12.17) и (12.25),

$$\begin{aligned}
 \kappa_{n-1}(k) &= \left[-\mathbf{h}_{n-1}^{\text{bH}}(k), 1 \right] \left[\sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) \right] = \\
 &= \begin{bmatrix} -\mathbf{h}_{n-1}^{\text{b}}(k) \\ 1 \end{bmatrix}^{\text{H}} \left[\sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) \right] = \mathbf{b}_n^{\text{H}}(k) \mathbf{r}_n(k) = \\
 &= \sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) - \mathbf{h}_{n-1}^{\text{bH}}(k) \mathbf{r}_{n-1}(k) = \\
 &= \sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) - \mathbf{h}_{n-1}^{\text{bH}}(k) \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{n-1}(i) d^*(i) = \\
 &= \sum_{i=1}^k \lambda^{k-i} \left[x(i-n+1) - \mathbf{h}_{n-1}^{\text{bH}}(k) \mathbf{x}_{n-1}(i) \right] d^*(i) = \sum_{i=1}^k \lambda^{k-i} e_{n-1}^{\text{b}}(i) d^*(i). \tag{12.50}
 \end{aligned}$$

Если уравнение (12.49) умножить на $\kappa_{n-1}^{-1}(k) E_{n-1,\min}^{\text{b}}(k)$, то получим, что

$$\begin{aligned}
 \mathbf{R}_n(k) \left\{ \mathbf{h}_n(k) - \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} \right\} \kappa_{n-1}^{-1}(k) E_{n-1,\min}^{\text{b}}(k) = \\
 = \begin{bmatrix} \mathbf{0}_{n-1} \\ \kappa_{n-1}(k) \end{bmatrix} \kappa_{n-1}^{-1}(k) E_{n-1,\min}^{\text{b}}(k) = \begin{bmatrix} \mathbf{0}_{n-1} \\ E_{n-1,\min}^{\text{b}}(k) \end{bmatrix}. \tag{12.51}
 \end{aligned}$$

Так как согласно уравнению (11.49)

$$\mathbf{R}_n(k) \mathbf{b}_n(k) = \begin{bmatrix} \mathbf{0}_N \\ E_{n-1,\min}^{\text{b}}(k) \end{bmatrix}, \tag{12.52}$$

то, сравнивая уравнения (12.51) и (12.52), можно заключить, что

$$\mathbf{b}_n(k) = \left\{ \mathbf{h}_n(k) - \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} \right\} \kappa_{n-1}^{-1}(k) E_{n-1,\min}^{\text{b}}(k),$$

откуда

$$\mathbf{h}_n(k) = \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} + \mathbf{b}_n(k) \left[E_{n-1,\min}^{\text{b}}(k) \right]^{-1} \kappa_{n-1}(k). \tag{12.53}$$

Тогда, с учетом уравнения (12.53),

$$\begin{aligned}
 \alpha_n(k) &= d(k) - \mathbf{h}_n^{\text{H}}(k-1) \mathbf{x}_n(k) = d(k) - \\
 &- \left\{ \begin{bmatrix} \mathbf{h}_{n-1}(k-1) \\ 0 \end{bmatrix} + \mathbf{b}_n(k-1) \left[E_{n-1,\min}^{\text{b}}(k-1) \right]^{-1} \kappa_{n-1}(k-1) \right\}^{\text{H}} \begin{bmatrix} \mathbf{x}_{n-1}(k) \\ x(k-n+1) \end{bmatrix} = \\
 &= d(k) - \mathbf{h}_{n-1}^{\text{H}}(k-1) \mathbf{x}_{n-1}(k) - \kappa_{n-1}^*(k-1) \left[E_{n-1,\min}^{\text{b}}(k-1) \right]^{-1} \mathbf{b}_n^{\text{H}}(k-1) \mathbf{x}_n(k) = \\
 &= \alpha_{n-1}(k) - \kappa_{n-1}^*(k-1) \left[E_{n-1,\min}^{\text{b}}(k-1) \right]^{-1} \alpha_{n-1}^{\text{b}}(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^{\text{b}}(k), \tag{12.54}
 \end{aligned}$$

где

$$\gamma_n(k) = \left[E_{n-1, \min}^b(k) \right]^{-1} \kappa_{n-1}(k). \quad (12.55)$$

Используя (12.53), уравнение для рекурсивного вычисления ошибки $e_n(k)$ можно записать как

$$\begin{aligned} e_n(k) &= d(k) - \mathbf{h}_n^H(k) \mathbf{x}_n(k) = d(k) - \\ &- \left\{ \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} + \mathbf{b}_n(k) \left[E_{n-1, \min}^b(k) \right]^{-1} \kappa_{n-1}(k) \right\}^H \begin{bmatrix} \mathbf{x}_{n-1}(k) \\ x(k-n+1) \end{bmatrix} = \\ &= d(k) - \mathbf{h}_{n-1}^H(k) \mathbf{x}_{n-1}(k) - \kappa_{n-1}^*(k) \left[E_{n-1, \min}^b(k) \right]^{-1} \mathbf{b}_n^H(k) \mathbf{x}_n(k) = \\ &= e_{n-1}(k) - \kappa_{n-1}^*(k) \left[E_{n-1, \min}^b(k) \right]^{-1} e_{n-1}^b(k) = e_{n-1}(k) + \gamma_n^*(k) e_{n-1}^b(k). \end{aligned} \quad (12.56)$$

Таким образом, (12.54) и (12.56) являются формулами рекурсивного вычисления ошибок моделирования требуемого сигнала. Для их использования необходимо получить уравнение рекурсивного вычисления переменной $\kappa_{n-1}(k)$, используемой для вычисления $\gamma_n(k)$, подобно тому, как вычисляется переменная $\Delta_{n-1}(k)$, см. уравнения (12.38) и (12.42), используемая для вычисления коэффициентов отражения $\Gamma_n^f(k)$ и $\Gamma_n^b(k)$.

Для этого с учетом уравнений (11.39) $\mathbf{h}_{n-1}^b(k) = \mathbf{h}_{n-1}^b(k-1) + \mathbf{g}_{n-1}(k) \alpha_{n-1}^{b*}(k)$, (6.79) $\mathbf{g}_{n-1}(k) = \mathbf{R}_{n-1}^{-1}(k) \mathbf{x}_{n-1}(k)$ и (6.69) $\mathbf{h}_{n-1}(k) = \mathbf{R}_{n-1}^{-1}(k) \mathbf{r}_{n-1}(k)$ уравнение (12.50) преобразуем к виду

$$\begin{aligned} \kappa_{n-1}(k) &= \mathbf{b}_n^H(k) \mathbf{r}_n(k) = \begin{bmatrix} -\mathbf{h}_{n-1}^b(k) \\ 1 \end{bmatrix}^H \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \sum_{i=1}^k \lambda^{k-i} x(i-n+1) d^*(i) \end{bmatrix} = \\ &= \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-1) - \mathbf{g}_{n-1}(k) \alpha_{n-1}^{b*}(k) \\ 1 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}(k-1) + \mathbf{x}_{n-1}(k) d^*(k) \\ \lambda \mathbf{r}_n(k-1) + x(k-n+1) d^*(k) \end{bmatrix} = \\ &= \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-1) \\ 1 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}(k-1) \\ \lambda \mathbf{r}_n(k-1) \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_{n-1}^b(k-1) \\ 1 \end{bmatrix}^H \begin{bmatrix} \mathbf{x}_{n-1}(k) d^*(k) \\ x(k-n+1) d^*(k) \end{bmatrix} + \\ &+ \begin{bmatrix} -\mathbf{g}_{n-1}(k) \alpha_{n-1}^{b*}(k) \\ 0 \end{bmatrix}^H \begin{bmatrix} \lambda \mathbf{r}_{n-1}(k-1) \\ \lambda \mathbf{r}_n(k-1) \end{bmatrix} + \begin{bmatrix} -\mathbf{g}_{n-1}(k) \alpha_{n-1}^{b*}(k) \\ 0 \end{bmatrix}^H \begin{bmatrix} \mathbf{x}_{n-1}(k) d^*(k) \\ x(k-n+1) d^*(k) \end{bmatrix} = \quad (12.57) \\ &= \lambda \mathbf{b}_n^H(k-1) \mathbf{r}_n(k-1) + \mathbf{b}_n^H(k-1) \mathbf{x}_n(k) d^*(k) - \alpha_{n-1}^b(k) \mathbf{g}_{n-1}^H(k) \lambda \mathbf{r}_{n-1}(k-1) + \\ &- \alpha_{n-1}^b(k) \mathbf{g}_{n-1}^H(k) \mathbf{x}_{n-1}(k) d^*(k) = \\ &= \lambda \kappa_{n-1}(k-1) + \alpha_{n-1}^b(k) d^*(k) - \alpha_{n-1}^b(k) \mathbf{g}_{n-1}^H(k) [\lambda \mathbf{r}_{n-1}(k-1) + \mathbf{x}_{n-1}(k) d^*(k)] = \\ &= \lambda \kappa_{n-1}(k-1) + \alpha_{n-1}^b(k) d^*(k) - \alpha_{n-1}^b(k) \mathbf{g}_{n-1}^H(k) \mathbf{r}_{n-1}(k) = \\ &= \lambda \kappa_{n-1}(k-1) + \alpha_{n-1}^b(k) [d^*(k) - \mathbf{x}_{n-1}^H(k) \mathbf{h}_{n-1}(k)] = \lambda \kappa_{n-1}(k-1) + \alpha_{n-1}^b(k) e_{n-1}^*(k). \end{aligned}$$

В дальнейшем для получения алгоритмов адаптивной фильтрации потребуются соотношения для рекурсивного вычисления отношения ошибок линейного предсказания, которое согласно (11.83) и (11.84) для произвольного n определяется как

$$\varphi_n(k) = \varphi_{n-1}(k-1) - e_{n-1}^{f*}(k) \left[E_{n-1,\min}^f(k) \right]^{-1} e_{n-1}^f(k) \quad (12.58)$$

или

$$\varphi_n(k) = \varphi_{n-1}(k) - e_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k). \quad (12.59)$$

Подобно процедурам (12.19) и (12.27), минимальное значение ошибок моделирования требуемого сигнала также может вычисляться рекурсивно. Для этого уравнение (6.72) необходимо преобразовать с использованием выражений для входящих в него переменных, т. е.

$$\begin{aligned} E_{n,\min}(k) &= E_d(k) - \mathbf{r}_n^H(k) \mathbf{h}_n(k) = \\ &= E_d(k) - \mathbf{h}_n^H(k) \mathbf{R}_n(k) \mathbf{h}_n(k) = \\ &= E_d(k) - \mathbf{h}_n^H(k) \begin{bmatrix} \mathbf{R}_{n-1}(k) & \mathbf{r}_{n-1}^b(k) \\ \mathbf{r}_{n-1}^{bH}(k) & \sigma^{b2}(k) \end{bmatrix} \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} + \\ &+ \mathbf{b}_n(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) \Big\} = E_d(k) - \mathbf{h}_n^H(k) \begin{bmatrix} \mathbf{R}_{n-1}(k) \mathbf{h}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k) \mathbf{h}_{n-1}(k) \end{bmatrix} + \\ &+ \begin{bmatrix} -\mathbf{R}_{n-1}(k) \mathbf{h}_{n-1}^b(k) + \mathbf{r}_{n-1}^b(k) \\ -\mathbf{r}_{n-1}^b(k) \mathbf{h}_{n-1}^b(k) + \sigma^{b2}(k) \end{bmatrix} \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) \Big\} = \\ &= E_d(k) - \mathbf{h}_n^H(k) \left\{ \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k) \mathbf{h}_{n-1}(k) \end{bmatrix} + \begin{bmatrix} -\mathbf{r}_{n-1}^b(k) + \mathbf{r}_{n-1}^b(k) \\ E_{n-1,\min}^b(k) \end{bmatrix} \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) \right\} = \\ &= E_d(k) - \left\{ \begin{bmatrix} \mathbf{h}_{n-1}(k) \\ 0 \end{bmatrix} + \mathbf{b}_n(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) \right\}^H \times \quad (12.60) \\ &\times \left\{ \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k) \mathbf{h}_{n-1}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ E_{n-1,\min}^b(k) \end{bmatrix} \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) \right\} = \\ &= E_d(k) - \mathbf{h}_{n-1}^H(k) \mathbf{r}_{n-1}(k) - \kappa_{n-1}^*(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \mathbf{b}_n^H(k) \begin{bmatrix} \mathbf{r}_{n-1}(k) \\ \mathbf{r}_{n-1}^{bH}(k) \mathbf{h}_{n-1}(k) \end{bmatrix} - \\ &- \kappa_{n-1}^*(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \mathbf{b}_n^H(k) \begin{bmatrix} 0 \\ E_{n-1,\min}^b(k) \end{bmatrix} \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) = \\ &= E_{n-1,\min}(k) - \kappa_{n-1}^*(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \left[-\mathbf{h}_{n-1}^{bH}(k) \mathbf{r}_{n-1}(k) + \mathbf{r}_{n-1}^{bH}(k) \mathbf{h}_{n-1}(k) \right] - \\ &- \kappa_{n-1}^*(k) \left[E_{n-1,\min}^b(k) \right]^{-1} E_{n-1,\min}^b(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) = \\ &= E_{n-1,\min}(k) - \kappa_{n-1}^*(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k). \end{aligned}$$

В уравнении (12.60) использовано соотношение (12.53) для $\mathbf{h}_n(k)$. Кроме того, $-\mathbf{h}_{n-1}^{\text{bH}}(k)\mathbf{r}_{n-1}(k) + \mathbf{r}_{n-1}^{\text{bH}}(k)\mathbf{h}_{n-1}(k) = 0$, так как $\mathbf{h}_{n-1}^{\text{b}}(k) = \mathbf{R}_{n-1}^{-1}(k)\mathbf{r}_{n-1}^{\text{b}}(k)$ и $\mathbf{r}_{n-1}(k) = \mathbf{R}_{n-1}(k)\mathbf{h}_{n-1}(k)$, а значит,

$$\begin{aligned}\mathbf{h}_{n-1}^{\text{bH}}(k)\mathbf{r}_{n-1}(k) &= \left[\mathbf{R}_{n-1}^{-1}(k)\mathbf{r}_{n-1}^{\text{b}}(k) \right]^{\text{H}} \mathbf{R}_{n-1}(k)\mathbf{h}_{n-1}(k) = \\ &= \mathbf{r}_{n-1}^{\text{bH}}(k)\mathbf{R}_{n-1}^{-1}(k)\mathbf{R}_{n-1}(k)\mathbf{h}_{n-1}(k) = \mathbf{r}_{n-1}^{\text{bH}}(k)\mathbf{h}_{n-1}(k).\end{aligned}\quad (12.61)$$

12.5. Рекурсивные адаптивные алгоритмы на основе вычисления априорных и апостериорных ошибок

Полученные в разделах 12.1—12.4 соотношения, используемые для вычисления различных переменных в фильтрах с числом ВК, равным n , с помощью переменных для фильтров с числом ВК $n-1$, могут быть использованы для построения адаптивных фильтров с числом ВК, равным N .

Одним из таких алгоритмов является алгоритм на основе априорных и апостериорных ошибок линейного предсказания, приведенный в табл. 12.1. Для его формирования необходимо последовательно применять уравнения (11.18), (11.41), (12.36), (12.57), (12.28), (12.30), (12.54), (12.18), (12.26), (12.55), (12.29), (12.31) и (12.56). Данная последовательность обусловлена необходимостью вычисления используемых друг за другом переменных.

Из табл. 12.1 следует, что на каждой итерации по времени k выполняется N итераций при изменении числа ВК. Вычисления в алгоритме зависят только от переменных с индексами n и $n-1$, что отличает этот алгоритм от алгоритмов из глав 6—9, где большинство вычислений являются векторными.

Для хранения значений $E_{n-1,\min}^{\text{f}}(k-1)$, $E_{n-1,\min}^{\text{b}}(k-1)$, $\Delta_{n-1}(k-1)$, $\kappa_{n-1}(k-1)$, $e_{n-1}^{\text{b}}(k-1)$, $\alpha_{n-1}^{\text{b}}(k-1)$ требуется N ячеек памяти для каждой из этих переменных.

Кроме того, инициализация алгоритма осуществляется в два этапа. Первый этап заключается в общей инициализации переменных, как это показано в уравнении (12.1.0). Здесь инициализация минимальных энергий ошибок линейного предсказания совпадает с аналогичной инициализацией в алгоритмах главы 11.

Инициализация ошибок на каждой k -й итерации как это показано в уравнении (12.1.1) обусловлена тем, что фильтры с числом ВК, равным 0, не содержат ВК, а потому ошибки таких фильтров являются самими сигналами. Аналогична ситуация с ошибками моделирования требуемого сигнала. Поэтому согласно уравнениям (12.28)–(12.31), (12.54) и (12.56) справедливы соотношения, определяемые уравнением (12.1.1).

Таблица 12.1. Лестничный RLS-алгоритм на основе априорных и апостериорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0)=\delta^2$, $E_{n-1,\min}^b(0)=\delta^2\lambda^{-(n-1)}$,	(12.1.0)
$\Delta_{n-1}(0)=0$, $\kappa_{n-1}(0)=0$, $\alpha_{n-1}^b(0)=0$, $e_{n-1}^b(0)=0$,	
$\Gamma_n^f(0)=0$, $\Gamma_n^b(0)=0$, $\gamma_n(0)=0$, $n=1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$\alpha_0^f(k)=x(k)$, $\alpha_0^b(k)=x(k)$, $e_0^f(k)=x(k)$, $e_0^b(k)=x(k)$,	(12.1.1)
$\alpha_0(k)=d(k)$, $e_0(k)=d(k)$	
For $n=1, 2, \dots, N$	
$E_{n-1,\min}^f(k)=\lambda E_{n-1,\min}^f(k-1)+e_{n-1}^f(k)\alpha_{n-1}^{f*}(k)$	(12.1.2)
$E_{n-1,\min}^b(k)=\lambda E_{n-1,\min}^b(k-1)+e_{n-1}^b(k)\alpha_{n-1}^{b*}(k)$	(12.1.3)
$\Delta_{n-1}(k)=\lambda\Delta_{n-1}(k-1)+e_{n-1}^b(k-1)\alpha_{n-1}^{f*}(k)$	(12.1.4)
$\kappa_{n-1}(k)=\lambda\kappa_{n-1}(k-1)+\alpha_{n-1}^b(k)e_{n-1}^{f*}(k)$	(12.1.5)
$\alpha_n^f(k)=\alpha_{n-1}^f(k)+\Gamma_n^{f*}(k-1)\alpha_{n-1}^b(k-1)$	(12.1.6)
$\alpha_n^b(k)=\alpha_{n-1}^b(k-1)+\Gamma_n^{b*}(k-1)\alpha_{n-1}^f(k)$	(12.1.7)
$\alpha_n(k)=\alpha_{n-1}(k)+\gamma_n^*(k-1)\alpha_{n-1}^b(k)$	(12.1.8)
$\Gamma_n^f(k)=-\left[E_{n-1,\min}^b(k-1)\right]^{-1}\Delta_{n-1}(k)$	(12.1.9)
$\Gamma_n^b(k)=-\left[E_{n-1,\min}^f(k)\right]^{-1}\Delta_{n-1}^*(k)$	(12.1.10)
$\gamma_n(k)=\left[E_{n-1,\min}^b(k)\right]^{-1}\kappa_{n-1}(k)$	(12.1.11)
$e_n^f(k)=e_{n-1}^f(k)+\Gamma_n^{f*}(k)e_{n-1}^b(k-1)$	(12.1.12)
$e_n^b(k)=e_{n-1}^b(k-1)+\Gamma_n^{b*}(k)e_{n-1}^f(k)$	(12.1.13)
$e_n(k)=e_{n-1}(k)+\gamma_n^*(k)e_{n-1}^b(k)$	(12.1.14)
End for n	
End for k	

На рис. 12.1 и рис. 12.2 показаны структуры вычислений ошибок в алгоритме, приведенном в табл. 12.1. Из этой таблицы следует, что ошибки $\alpha_n^f(k)$ и $e_n^f(k)$ являются взвешенными функциями ошибок $\alpha_{n-1}^b(k-1)$ и $e_{n-1}^b(k-1)$, а ошибки $\alpha_n^b(k)$ и $e_n^b(k)$ — взвешенными функциями ошибок $\alpha_{n-1}^f(k)$ и $e_{n-1}^f(k)$. Другими словами, с помощью коэффициентов $\Gamma_n^f(k-1)$, $\Gamma_n^f(k)$, $\Gamma_n^b(k-1)$ и $\Gamma_n^b(k)$ эти ошибки как бы отражаются друг на друга, что обуславливает решеточную структуру процедуры вычисления этих ошибок (см. верхнюю часть рис. 12.1 и рис. 12.2) и название этих коэффициентов.

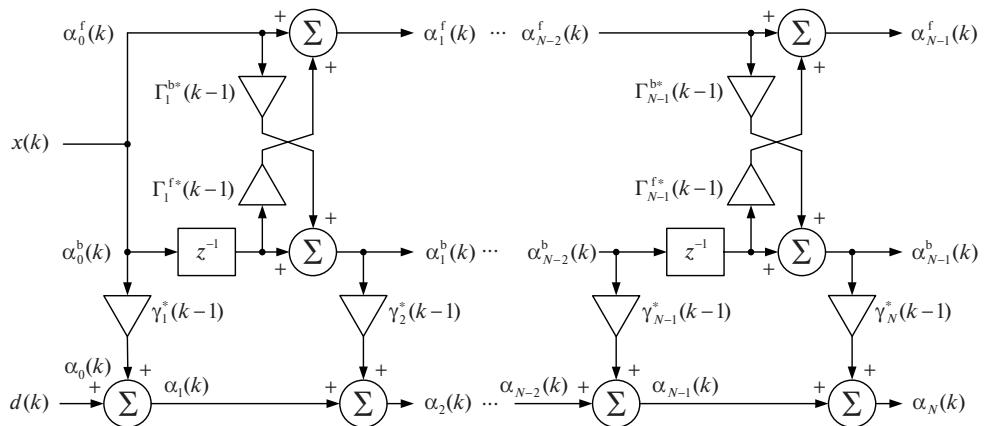


Рис. 12.1. Адаптивный фильтр на основе априорных ошибок

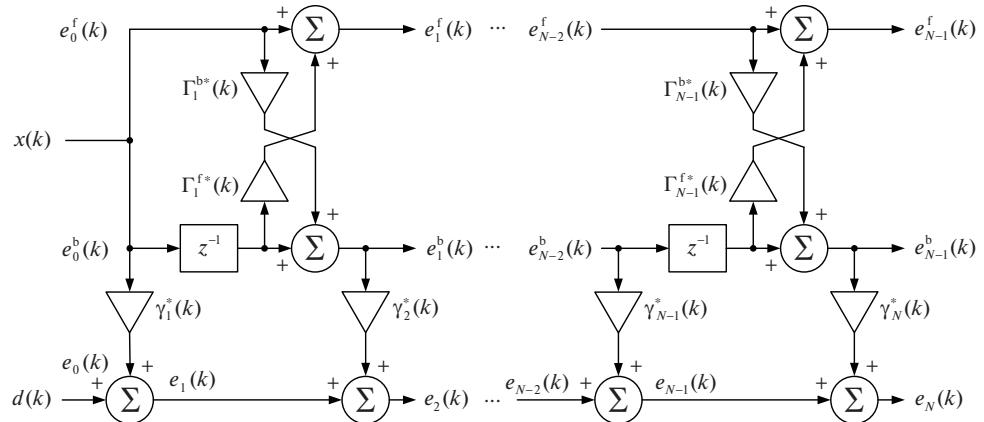


Рис. 12.2. Адаптивный фильтр на основе апостериорных ошибок

Таким образом, вычисление ошибок $\alpha(k)=\alpha_N(k)$ и $e(k)=e_N(k)$ можно осуществить рекурсивно с помощью уравнений (12.54) и (12.56) в соответствии со структурами, показанными на рис. 12.1 и рис. 12.2, не определяя ВК фильтра и его выходной сигнал в явном виде.

Из уравнений (12.54) и (12.56), а также рис. 12.1 и рис. 12.2 видно, что вычисления, связанные с линейными предсказаниями для фильтра в числом ВК, равным N , нужно выполнять $N-1$ раз, а вычисления, связанные с ошибками моделирования, — N раз. Для простоты и однообразности представления вычислительной процедуры алгоритма в табл. 12.1 все вычисления выполняются N раз. Этот момент нужно учитывать при программной реализации алгоритма (см. табл. 12.1) и других алгоритмов, рассматриваемых в настоящей главе. Аппаратная реализация, как видно из рис. 12.1 и рис. 12.2, отмеченной избыточностью не обладает.

Алгоритм (см. табл. 12.1) может быть преобразован в два алгоритма, которые функционируют на основе вычисления только априорных или только апостериорных ошибок. Эти алгоритмы приведены в табл. 12.2 и табл. 12.3. Здесь в вычислениях используется только один из видов ошибок предсказания и моделирования, а необходимые недостающие ошибки определяются на основе соотношений между априорными и апостериорными ошибками (11.74) $\varphi_{n-1}(k) = e_{n-1}(k)/\alpha_{n-1}(k)$, (11.75) $\varphi_{n-1}(k-1) = e_{n-1}^f(k)/\alpha_{n-1}^f(k)$ и (11.76) $e_{n-1}^b(k)/\alpha_{n-1}^b(k) = \varphi_{n-1}(k)$. Рекурсивное вычисление $\varphi_n(k)$ осуществляется с помощью уравнения (12.59) как

$$\begin{aligned}\varphi_n(k) &= \varphi_{n-1}(k) - e_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) = \\ &= \varphi_{n-1}(k) - \varphi_{n-1}(k) \alpha_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) = \\ &= \varphi_{n-1}(k) \left\{ 1 - \alpha_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \right\}.\end{aligned}\quad (12.62)$$

Таблица 12.2. Лестничный RLS-алгоритм на основе априорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0) = \delta^2$, $E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$,	(12.2.0)
$\Delta_{n-1}(0) = 0$, $\kappa_{n-1}(0) = 0$, $\alpha_{n-1}^b(0) = 0$, $\varphi_{n-1}(0) = 1$,	
$\Gamma_n^f(0) = 0$, $\Gamma_n^b(0) = 0$, $\gamma_n(0) = 0$, $n = 1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$\alpha_0^f(k) = x(k)$, $\alpha_0^b(k) = x(k)$, $\alpha_0(k) = d(k)$, $\varphi_{n-1}(k) = 1$, $n = 1, 2, \dots, N$	(12.2.1)
For $n = 1, 2, \dots, N$	
$E_{n-1,\min}^f(k) = \lambda E_{n-1,\min}^f(k-1) + \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)$	(12.2.2)
$E_{n-1,\min}^b(k) = \lambda E_{n-1,\min}^b(k-1) + \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^{b*}(k)$	(12.2.3)
$\Delta_{n-1}(k) = \lambda \Delta_{n-1}(k-1) + \alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)$	(12.2.4)
$\kappa_{n-1}(k) = \lambda \kappa_{n-1}(k-1) + \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^{b*}(k)$	(12.2.5)
$\varphi_n(k) = \varphi_{n-1}(k) \left\{ 1 - \alpha_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \right\}$	(12.2.6)
$\alpha_n^f(k) = \alpha_{n-1}^f(k) + \Gamma_n^f(k-1) \alpha_{n-1}^b(k-1)$	(12.2.7)
$\alpha_n^b(k) = \alpha_{n-1}^b(k-1) + \Gamma_n^b(k-1) \alpha_{n-1}^f(k)$	(12.2.8)
$\alpha_n(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k)$	(12.2.9)
$\Gamma_n^f(k) = - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \Delta_{n-1}(k)$	(12.2.10)
$\Gamma_n^b(k) = - \left[E_{n-1,\min}^f(k) \right]^{-1} \Delta_{n-1}^*(k)$	(12.2.11)
$\gamma_n(k) = \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k)$	(12.2.12)
End for n	
End for k	

В табл. 12.2 инициализация $\varphi_{n-1}(0)=1$ следует из соотношения $\varphi_{n-1}(k-1)=e_{n-1}^f(k)/\alpha_{n-1}^f(k)=x(k)/x(k)=1$ для всех $k=1, \dots, K$. Инициализация $\varphi_{n-1}(k)=1$, $n=1, 2, \dots, N$ осуществляется, исходя из условий (11.74) $\varphi_{n-1}(k)=e_{n-1}(k)/\alpha_{n-1}(k)$ и (11.76) $\varphi_{n-1}(k)=e_{n-1}^b(k)/\alpha_{n-1}^b(k)=x(k)/x(k)=e_{n-1}(k)/\alpha_{n-1}(k)=d(k)/d(k)=1$.

Таблица 12.3. Лестничный RLS-алгоритм на основе апостериорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0)=\delta^2$, $E_{n-1,\min}^b(0)=\delta^2\lambda^{-(n-1)}$,	(12.3.0)
$\Delta_{n-1}(0)=0$, $\kappa_{n-1}(0)=0$, $e_{n-1}^b(0)=0$, $\varphi_{n-1}(0)=1$	
$\Gamma_n^f(0)=0$, $\Gamma_n^b(0)=0$, $\gamma_n(0)=0$, $n=1, 2, \dots, N$	
For $k=1, 2, \dots, K$	
$e_0^f(k)=x(k)$, $e_0^b(k)=x(k)$, $e_0(k)=d(k)$, $\varphi_{n-1}(k)=1$, $n=1, 2, \dots, N$	(12.3.1)
For $n=1, 2, \dots, N$	
$E_{n-1,\min}^f(k)=\lambda E_{n-1,\min}^f(k-1)+e_{n-1}^f(k)\varphi_{n-1}^{-1}(k-1)e_{n-1}^{f*}(k)$	(12.3.2)
$E_{n-1,\min}^b(k)=\lambda E_{n-1,\min}^b(k-1)+e_{n-1}^b(k)\varphi_{n-1}^{-1}(k)e_{n-1}^{b*}(k)$	(12.3.3)
$\Delta_{n-1}(k)=\lambda \Delta_{n-1}(k-1)+e_{n-1}^b(k-1)\varphi_{n-1}^{-1}(k-1)e_{n-1}^{f*}(k)$	(12.3.4)
$\kappa_{n-1}(k)=\lambda \kappa_{n-1}(k-1)+e_{n-1}^b(k)\varphi_{n-1}^{-1}(k)e_{n-1}^{b*}(k)$	(12.3.5)
$\varphi_n(k)=\varphi_{n-1}(k)-e_{n-1}^{b*}(k)\left[E_{n-1,\min}^b(k)\right]^{-1}e_{n-1}^b(k)$	(12.3.6)
$\Gamma_n^f(k)=-\left[E_{n-1,\min}^b(k-1)\right]^{-1}\Delta_{n-1}(k)$	(12.3.7)
$\Gamma_n^b(k)=-\left[E_{n-1,\min}^f(k)\right]^{-1}\Delta_{n-1}^*(k)$	(12.3.8)
$\gamma_n(k)=\left[E_{n-1,\min}^b(k)\right]^{-1}\kappa_{n-1}(k)$	(12.3.9)
$e_n^f(k)=e_{n-1}^f(k)+\Gamma_n^f(k)e_{n-1}^b(k-1)$	(12.3.10)
$e_n^b(k)=e_{n-1}^b(k-1)+\Gamma_n^b(k)e_{n-1}^f(k)$	(12.3.11)
$e_n(k)=e_{n-1}(k)+\gamma_n^*(k)e_{n-1}^b(k)$	(12.3.12)
End for n	
$\alpha_N(k)=e_N(k)\varphi_N^{-1}(k)$	(12.3.13)
End for k	

Поскольку выходной сигнал адаптивного фильтра формируется как $y_N(k)=\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$, т.е. на текущей итерации k используются ВК, вычисленные на предыдущей итерации $k-1$, то в качестве физически наблюдаемого сигнала ошибки моделирования требуемого сигнала $d(k)$ необходимо использовать априорную ошибку $\alpha_N(k)=d(k)-\mathbf{h}_N^H(k-1)\mathbf{x}_N(k)$, которая для алгоритма (см. табл. 12.3) на каждой итерации k определяется из уравнения (11.74) через

переменную $\varphi_N(k)$ как это показано в уравнении (12.3.13) после завершения итераций по n , т. е. при $n=N$.

12.6. Рекурсивные адаптивные алгоритмы с обратными связями на основе вычисления априорных и апостериорных ошибок

Дальнейшая модификация алгоритмов в табл. 12.1—12.3 может быть осуществлена за счет рекурсивного вычисления коэффициентов отражения. Такие алгоритмы называются алгоритмами с обратными связями [141]. Для получения соответствующих вычислительных процедур в уравнениях (12.1.9)–(12.1.11) используются выражения для переменных, определяемые остальными строками табл. 12.1.

Определим из уравнения (12.36) переменную

$$\Delta_{n-1}(k-1) = \lambda^{-1} \left[\Delta_{n-1}(k) - e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k) \right], \quad (12.63)$$

а из уравнения (12.1.2) — переменную

$$E_{n-1,\min}^b(k-2) = \lambda^{-1} \left[E_{n-1,\min}^b(k-1) - e_{n-1}^b(k-1) \alpha_{n-1}^{b*}(k-1) \right]. \quad (12.64)$$

Затем используем уравнения (12.63) и (12.64) в уравнении (12.1.9) как

$$E_{n-1,\min}^b(k-2) \Gamma_n^f(k-1) = -\Delta_{n-1}(k-1), \quad (12.65)$$

т. е.

$$\begin{aligned} & \lambda^{-1} \left[E_{n-1,\min}^b(k-1) - e_{n-1}^b(k-1) \alpha_{n-1}^{b*}(k-1) \right] \Gamma_n^f(k-1) = \\ & = -\lambda^{-1} \left[\Delta_{n-1}(k) - e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k) \right], \end{aligned} \quad (12.66)$$

откуда

$$\begin{aligned} & E_{n-1,\min}^b(k-1) \Gamma_n^f(k-1) - e_{n-1}^b(k-1) \alpha_{n-1}^{b*}(k-1) \Gamma_n^f(k-1) = \\ & = -\Delta_{n-1}(k) + e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k). \end{aligned} \quad (12.67)$$

Из уравнения (12.67) можно определить

$$\begin{aligned} & \Gamma_n^f(k-1) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_{n-1}^{b*}(k-1) \Gamma_n^f(k-1) = \\ & = - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \Delta_{n-1}(k) + \left[E_{n-1,\min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k), \end{aligned} \quad (12.68)$$

или

$$\begin{aligned} & \Gamma_n^f(k-1) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_{n-1}^{b*}(k-1) \Gamma_n^f(k-1) = \\ & = \Gamma_n^f(k) + \left[E_{n-1,\min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_{n-1}^{f*}(k), \end{aligned} \quad (12.69)$$

откуда, с учетом (12.1.6),

$$\begin{aligned}\Gamma_n^f(k) &= \Gamma_n^f(k-1) - \left[E_{n-1, \min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \left[\alpha_{n-1}^{f*}(k) + \alpha_{n-1}^{b*}(k-1) \Gamma_n^f(k-1) \right] = \\ &= \Gamma_n^f(k-1) - \left[E_{n-1, \min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_n^{f*}(k).\end{aligned}\quad (12.70)$$

Аналогично из уравнения (12.42) определим переменную

$$\Delta_{n-1}^*(k-1) = \lambda^{-1} \left[\Delta_{n-1}^*(k) - e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1) \right], \quad (12.71)$$

а из уравнения (12.1.1) — переменную

$$E_{n-1, \min}^f(k-1) = \lambda^{-1} \left[E_{n-1, \min}^f(k) - e_{n-1}^f(k) \alpha_{n-1}^{f*}(k) \right], \quad (12.72)$$

а затем используем (12.71) и (12.72) в уравнении (12.1.10) как

$$E_{n-1, \min}^f(k-1) \Gamma_n^b(k-1) = -\Delta_{n-1}^*(k-1), \quad (12.73)$$

т. е.

$$\begin{aligned}\lambda^{-1} \left[E_{n-1, \min}^f(k) - e_{n-1}^f(k) \alpha_{n-1}^{f*}(k) \right] \Gamma_n^b(k-1) &= \\ &= -\lambda^{-1} \left[\Delta_{n-1}^*(k) - e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1) \right],\end{aligned}\quad (12.74)$$

откуда

$$E_{n-1}^f(k) \Gamma_n^b(k-1) - e_{n-1}^f(k) \alpha_{n-1}^{f*}(k) \Gamma_n^b(k-1) = -\Delta_{n-1}^*(k) + e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1). \quad (12.75)$$

Из уравнения (12.75) можно определить

$$\begin{aligned}\Gamma_n^b(k-1) - \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \alpha_{n-1}^{f*}(k) \Gamma_n^b(k-1) &= \\ &= - \left[E_{n-1, \min}^f(k) \right]^{-1} \Delta_{n-1}^*(k) + \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1),\end{aligned}\quad (12.76)$$

или

$$\begin{aligned}\Gamma_n^b(k-1) - \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \alpha_{n-1}^{f*}(k) \Gamma_n^b(k-1) &= \\ &= \Gamma_n^b(k) + \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \alpha_{n-1}^{b*}(k-1),\end{aligned}\quad (12.77)$$

откуда, с учетом (12.1.7),

$$\begin{aligned}\Gamma_n^b(k) &= \Gamma_n^b(k-1) - \\ &- \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \left[\alpha_{n-1}^{b*}(k-1) + \alpha_{n-1}^{f*}(k) \Gamma_n^b(k-1) \right] = \\ &= \Gamma_n^b(k-1) - \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k) \alpha_n^{b*}(k).\end{aligned}\quad (12.78)$$

Затем из уравнения (12.57) определим переменную

$$\kappa_{n-1}(k-1) = \lambda^{-1} \left[\kappa_{n-1}(k) - \alpha_{n-1}^b(k) e_{n-1}^*(k) \right], \quad (12.79)$$

а из уравнения (12.1.2) — переменную

$$E_{n-1, \min}^b(k-1) = \lambda^{-1} \left[E_{n-1, \min}^b(k) - e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \right]. \quad (12.80)$$

После этого используем (12.79) и (12.80) в уравнении (12.1.11) как

$$E_{n-1,\min}^b(k-1)\gamma_n(k-1) = -\kappa_{n-1}(k-1), \quad (12.81)$$

т. е.

$$\begin{aligned} \lambda^{-1} & \left[E_{n-1,\min}^b(k) - e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \right] \gamma_n(k-1) = \\ & = -\lambda^{-1} \left[\kappa_{n-1}(k) - \alpha_{n-1}^b(k) e_{n-1}^*(k) \right], \end{aligned} \quad (12.82)$$

откуда

$$E_{n-1,\min}^b(k) \gamma_n(k-1) - e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = -\kappa_{n-1}(k) + \alpha_{n-1}^b(k) e_{n-1}^*(k). \quad (12.83)$$

Из уравнения (12.83) можно определить

$$\begin{aligned} \gamma_n(k-1) - & \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = - \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) + \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) e_{n-1}^*(k), \end{aligned} \quad (12.84)$$

или

$$\begin{aligned} \gamma_n(k-1) - & \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = \gamma_n(k) + \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) e_{n-1}^*(k), \end{aligned} \quad (12.85)$$

откуда, с учетом (12.1.8), получим, что

$$\begin{aligned} \gamma_n(k) = & \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) e_{n-1}^*(k) - \\ & - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k) - \\ & - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k) - \\ & - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^*(k) - \\ & - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^{b*}(k) \gamma_n(k-1) = \\ & = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \left[\alpha_{n-1}^*(k) + \alpha_{n-1}^{b*}(k) \gamma_n(k-1) \right] = \\ & = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^*(k). \end{aligned} \quad (12.86)$$

Таким образом, если в алгоритме (см. табл. 12.1) исключить уравнения (12.1.4) и (12.1.5), а вместо уравнений (12.1.9)–(12.1.11) использовать соответственно уравнения (12.70), (12.78) и (12.86), то получится лестничный RLS-алгоритм с обратными связями на основе априорных и апостериорных ошибок линейного предсказания. Этот алгоритм приведен в табл. 12.4.

Таблица 12.4. Лестничный RLS-алгоритм с обратными связями на основе априорных и апостериорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0) = \delta^2$, $E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$,	(12.4.0)
$\alpha_{n-1}^b(0) = 0$, $e_{n-1}^b(0) = 0$, $\Gamma_n^f(0) = 0$, $\Gamma_n^b(0) = 0$, $\gamma_n(0) = 0$, $n = 1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$\alpha_0^f(k) = x(k)$, $\alpha_0^b(k) = x(k)$, $e_0^f(k) = x(k)$, $e_0^b(k) = x(k)$,	(12.4.1)
$\alpha_0(k) = d(k)$, $e_0(k) = d(k)$	
For $n = 1, 2, \dots, N$	
$E_{n-1,\min}^f(k) = \lambda E_{n-1,\min}^f(k-1) + e_{n-1}^f(k) \alpha_{n-1}^{f*}(k)$	(12.4.2)
$E_{n-1,\min}^b(k) = \lambda E_{n-1,\min}^b(k-1) + e_{n-1}^b(k) \alpha_{n-1}^{b*}(k)$	(12.4.3)
$\alpha_n^f(k) = \alpha_{n-1}^f(k) + \Gamma_n^{f*}(k-1) \alpha_{n-1}^b(k-1)$	(12.4.4)
$\alpha_n^b(k) = \alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k)$	(12.4.5)
$\alpha_n(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k)$	(12.4.6)
$\Gamma_n^f(k) = \Gamma_n^f(k-1) - [E_{n-1,\min}^b(k-1)]^{-1} e_{n-1}^b(k-1) \alpha_n^{f*}(k)$	(12.4.7)
$\Gamma_n^b(k) = \Gamma_n^b(k-1) - [E_{n-1,\min}^f(k)]^{-1} e_{n-1}^f(k) \alpha_n^{b*}(k)$	(12.4.8)
$\gamma_n(k) = \gamma_n(k-1) - [E_{n-1,\min}^b(k)]^{-1} e_{n-1}^b(k) \alpha_{n-1}^*(k)$	(12.4.9)
$e_n^f(k) = e_{n-1}^f(k) + \Gamma_n^{f*}(k) e_{n-1}^b(k-1)$	(12.4.10)
$e_n^b(k) = e_{n-1}^b(k-1) + \Gamma_n^{b*}(k) e_{n-1}^f(k)$	(12.4.11)
$e_n(k) = e_{n-1}(k) + \gamma_n^*(k) e_{n-1}^b(k)$	(12.4.12)
End for n	
End for k	

Для получения лестничного RLS-алгоритма с обратными связями на основе априорных ошибок линейного предсказания в уравнениях (12.70), (12.78) и (12.86) необходимо заменить апостериорные ошибки на априорные ошибки подобно тому, как это было сделано в уравнениях алгоритма (см. табл. 12.2), т. е.

$$\Gamma_n^f(k) = \Gamma_n^f(k-1) - [E_{n-1,\min}^b(k-1)]^{-1} \alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_n^{f*}(k), \quad (12.87)$$

$$\Gamma_n^b(k) = \Gamma_n^b(k-1) - [E_{n-1,\min}^f(k)]^{-1} \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_n^{b*}(k), \quad (12.88)$$

$$\gamma_n(k) = \gamma_n(k-1) - [E_{n-1,\min}^b(k)]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_n^*(k), \quad (12.89)$$

а затем использовать уравнения (12.87)–(12.89) вместо уравнений (12.2.10)–(12.2.12) алгоритма, приведенного в табл. 12.2, и исключить уравнения (12.2.4) и (12.2.5). Этот алгоритм приведен в табл. 12.5.

Таблица 12.5. Лестничный RLS-алгоритм с обратными связями на основе априорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0) = \delta^2$, $E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$,	(12.5.0)
$\alpha_{n-1}^b(0) = 0$, $\varphi_{n-1}(0) = 1$, $\Gamma_n^f(0) = 0$, $\Gamma_n^b(0) = 0$, $\gamma_n(0) = 0$, $n = 1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$\alpha_0^f(k) = x(k)$, $\alpha_0^b(k) = x(k)$, $\alpha_0(k) = d(k)$, $\varphi_{n-1}(k) = 1$, $n = 1, 2, \dots, N$	(12.5.1)
For $n = 1, 2, \dots, N$	
$E_{n-1,\min}^f(k) = \lambda E_{n-1,\min}^f(k-1) + \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)$	(12.5.2)
$E_{n-1,\min}^b(k) = \lambda E_{n-1,\min}^b(k-1) + \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^{b*}(k)$	(12.5.3)
$\varphi_n(k) = \varphi_{n-1}(k) \left\{ 1 - \alpha_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \right\}$	(12.5.4)
$\alpha_n^f(k) = \alpha_{n-1}^f(k) + \Gamma_n^{f*}(k-1) \alpha_{n-1}^b(k-1)$	(12.5.5)
$\alpha_n^b(k) = \alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k)$	(12.5.6)
$\alpha_n(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k)$	(12.5.7)
$\Gamma_n^f(k) = \Gamma_n^f(k-1) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_n^f(k)$	(12.5.8)
$\Gamma_n^b(k) = \Gamma_n^b(k-1) - \left[E_{n-1,\min}^f(k) \right]^{-1} \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_n^{b*}(k)$	(12.5.9)
$\gamma_n(k) = \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k)$	(12.5.10)
End for n	
End for k	

Для получения алгоритма с обратными связями, аналогичного алгоритму (см. табл. 12.3), нужно сначала определить переменные $\left[E_{n-1,\min}^f(k) \right]^{-1}$ и $\left[E_{n-1,\min}^b(k) \right]^{-1}$ из уравнений (12.3.2) и (12.3.3).

Для этого преобразуем уравнение (12.58) к виду

$$\begin{aligned} \varphi_n(k) &= \varphi_{n-1}(k-1) - e_{n-1}^{f*}(k) \left[E_{n-1,\min}^f(k) \right]^{-1} e_{n-1}^f(k) = \\ &= \varphi_{n-1}(k-1) \left\{ 1 - \varphi_{n-1}^{-1}(k-1) e_{n-1}^{f*}(k) \left[E_{n-1,\min}^f(k) \right]^{-1} e_{n-1}^f(k) \right\}. \end{aligned} \quad (12.90)$$

Тогда, с учетом (12.90), уравнение (12.3.2) принимает вид

$$\begin{aligned} \lambda E_{n-1,\min}^f(k-1) &= E_{n-1,\min}^f(k) - \frac{e_{n-1}^f(k) e_{n-1}^{f*}(k)}{\varphi_{n-1}(k-1)} = \\ &= E_{n-1,\min}^f(k) \left[1 - \frac{e_{n-1}^f(k) e_{n-1}^{f*}(k)}{E_{n-1,\min}^f(k) \varphi_{n-1}(k-1)} \right] = \\ &= E_{n-1,\min}^f(k) \frac{\varphi_n(k)}{\varphi_{n-1}(k-1)}, \end{aligned} \quad (12.91)$$

откуда

$$\frac{1}{\lambda E_{n-1,\min}^f(k-1)} = \frac{1}{E_{n-1,\min}^f(k)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \right]^{-1}, \quad (12.92)$$

или

$$\frac{1}{E_{n-1,\min}^f(k)} = \frac{1}{\lambda E_{n-1,\min}^f(k-1)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \right]. \quad (12.93)$$

Аналогично преобразуем уравнение (12.59) к виду

$$\begin{aligned} \varphi_n(k) &= \varphi_{n-1}(k) - e_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) = \\ &= \varphi_{n-1}(k) \left\{ 1 - \varphi_{n-1}^{-1}(k) e_{n-1}^{b*}(k) \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \right\}. \end{aligned} \quad (12.94)$$

Тогда, с учетом (12.94), уравнение (12.3.3) принимает вид

$$\begin{aligned} \lambda E_{n-1,\min}^b(k-1) &= E_{n-1,\min}^b(k) - \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{\varphi_{n-1}(k)} = \\ &= E_{n-1,\min}^b(k) \left[1 - \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \right] = E_{n-1,\min}^b(k) \frac{\varphi_n(k)}{\varphi_{n-1}(k)}, \end{aligned} \quad (12.95)$$

откуда

$$\frac{1}{\lambda E_{n-1,\min}^b(k-1)} = \frac{1}{E_{n-1,\min}^b(k)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k)} \right]^{-1} \quad (12.96)$$

или

$$\frac{1}{E_{n-1,\min}^b(k)} = \frac{1}{\lambda E_{n-1,\min}^b(k-1)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k)} \right]. \quad (12.97)$$

Для получения формулы рекурсивного вычисления коэффициента отражения $\Gamma_n^f(k)$ необходимо преобразовать уравнение (12.3.7) с использованием уравнений (12.36) и (12.93), т. е.

$$\begin{aligned} \Gamma_n^f(k) &= - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \Delta_{n-1}(k) = \\ &= - \frac{1}{\lambda E_{n-1,\min}^b(k-2)} \left[\frac{\varphi_n(k-1)}{\varphi_{n-1}(k-1)} \right] \left[\lambda \Delta_{n-1}(k-1) + \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{\varphi_{n-1}(k-1)} \right] = \\ &= \frac{\varphi_n(k-1)}{\varphi_{n-1}(k-1)} \left[- \frac{\lambda \Delta_{n-1}(k-1)}{\lambda E_{n-1,\min}^b(k-2)} - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{\lambda E_{n-1,\min}^b(k-2) \varphi_{n-1}(k-1)} \right] = \\ &= \frac{\varphi_n(k-1)}{\varphi_{n-1}(k-1)} \left[\Gamma_n^f(k-1) - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{\lambda E_{n-1,\min}^b(k-2) \varphi_{n-1}(k-1)} \right]. \end{aligned} \quad (12.98)$$

Затем для получения формулы рекурсивного вычисления коэффициента отражения $\Gamma_n^b(k)$ необходимо преобразовать уравнение (12.3.8) с использованием уравнений (12.42) и (12.97), т. е.

$$\begin{aligned}
 \Gamma_n^b(k) &= -\left[E_{n-1,\min}^f(k) \right]^{-1} \Delta_{n-1}^*(k) = \\
 &= -\frac{1}{\lambda E_{n-1,\min}^f(k-1)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \right] \left[\lambda \Delta_{n-1}^*(k-1) + \frac{e_{n-1}^f(k) e_{n-1}^{b*}(k-1)}{\varphi_{n-1}(k-1)} \right] = \\
 &= \frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \left[-\frac{\lambda \Delta_{n-1}^*(k-1)}{\lambda E_{n-1,\min}^f(k-1)} - \frac{e_{n-1}^f(k) e_{n-1}^{b*}(k-1)}{\lambda E_{n-1,\min}^f(k-1) \varphi_{n-1}(k-1)} \right] = \\
 &= \frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \left[\Gamma_n^b(k-1) - \frac{e_{n-1}^f(k) e_{n-1}^{b*}(k-1)}{\lambda E_{n-1,\min}^f(k-1) \varphi_{n-1}(k-1)} \right].
 \end{aligned} \tag{12.99}$$

И для получения формулы рекурсивного вычисления коэффициента $\gamma_n(k)$ необходимо преобразовать уравнение (12.3.9) с использованием уравнений (12.57) и (12.97), т. е.

$$\begin{aligned}
 \gamma_n(k) &= \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}(k) = \\
 &= -\frac{1}{\lambda E_{n-1,\min}^b(k-1)} \left[\frac{\varphi_n(k)}{\varphi_{n-1}(k)} \right] \left[\lambda \kappa_{n-1}(k-1) + \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{\varphi_{n-1}(k)} \right] = \\
 &= \frac{\varphi_n(k)}{\varphi_{n-1}(k)} \left[-\frac{\lambda \kappa_{n-1}(k-1)}{\lambda E_{n-1,\min}^b(k-1)} - \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{\lambda E_{n-1,\min}^b(k-1) \varphi_{n-1}(k)} \right] = \\
 &= \frac{\varphi_n(k)}{\varphi_{n-1}(k)} \left[\gamma_n(k-1) - \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{\lambda E_{n-1,\min}^b(k-1) \varphi_{n-1}(k)} \right].
 \end{aligned} \tag{12.100}$$

Теперь уравнения (12.3.4) и (12.3.5) можно исключить из табл. 12.3, а уравнения (12.98)–(12.100) использовать вместо уравнений (12.3.7)–(12.3.9). Такой алгоритм, называемый лестничным RLS-алгоритмом на основе апостериорных ошибок с обратными связями, представлен в табл. 12.6.

Поскольку в уравнении (12.6.5) используются переменные $E_{n-1,\min}^b(k-2)$, то их инициализация осуществляется одинаковым образом как $E_{n-1,\min}^b(-1) = E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$, см. уравнение (12.6.1). Эти значения используются в вычислениях при $k=1$, т. е. $E_{n-1,\min}^b(k-2) = E_{n-1,\min}^b(-1) = \delta^2 \lambda^{-(n-1)}$ и $E_{n-1,\min}^b(k-1) = E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$. При $k=2$ используются значения $E_{n-1,\min}^b(k-2) = E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$ и $E_{n-1,\min}^b(k-1) = E_{n-1,\min}^b(1)$. В последующих вычислениях инициализированные переменные $E_{n-1,\min}^b$ уже не используются. Справедливость такой инициализации, приводящей к $\Gamma_n^f(1) = 0$ для всех $n=1, 2, \dots, N$, следует из прямой подстановки данных для $k=1$ в уравнение (12.6.5) из табл. 12.6 и в уравнение 12.3.7 из табл. 12.3.

Таблица 12.6. Лестничный RLS-алгоритм с обратными связями на основе апостериорных ошибок

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0) = \delta^2$, $E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$,	(12.6.0)
$E_{n-1,\min}^b(-1) = \delta^2 \lambda^{-(n-1)}$, $e_{n-1}^b(0) = 0$, $\varphi_{n-1}(0) = 1$,	
$\Gamma_n^f(0) = 0$, $\Gamma_n^b(0) = 0$, $\gamma_n(0) = 0$, $n = 1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$e_0^f(k) = x(k)$, $e_0^b(k) = x(k)$, $e_0(k) = d(k)$, $\varphi_{n-1}(k) = 1$, $n = 1, 2, \dots, N$	(12.6.1)
For $n = 1, 2, \dots, N$	
$E_{n-1,\min}^f(k) = \lambda E_{n-1,\min}^f(k-1) + e_{n-1}^f(k) \varphi_{n-1}^{-1}(k-1) e_{n-1}^{f*}(k)$	(12.6.2)
$E_{n-1,\min}^b(k) = \lambda E_{n-1,\min}^b(k-1) + e_{n-1}^b(k) \varphi_{n-1}^{-1}(k) e_{n-1}^{b*}(k)$	(12.6.3)
$\varphi_n(k) = \varphi_{n-1}(k) - e_{n-1}^{b*}(k) [E_{n-1,\min}^b(k)]^{-1} e_{n-1}^b(k)$	(12.6.4)
$\Gamma_n^f(k) = \frac{\varphi_n(k-1)}{\varphi_{n-1}(k-1)} \left[\Gamma_n^f(k-1) - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{\lambda E_{n-1,\min}^b(k-2) \varphi_{n-1}(k-1)} \right]$	(12.6.5)
$\Gamma_n^b(k) = \frac{\varphi_n(k)}{\varphi_{n-1}(k-1)} \left[\Gamma_n^b(k-1) - \frac{e_{n-1}^f(k) e_{n-1}^{b*}(k-1)}{\lambda E_{n-1,\min}^f(k-1) \varphi_{n-1}(k-1)} \right]$	(12.6.6)
$\gamma_n(k) = \frac{\varphi_n(k)}{\varphi_{n-1}(k)} \left[\gamma_n(k-1) - \frac{e_{n-1}^b(k) e_{n-1}^{f*}(k)}{\lambda E_{n-1,\min}^b(k-1) \varphi_{n-1}(k)} \right]$	(12.6.7)
$e_n^f(k) = e_{n-1}^f(k) + \Gamma_n^f(k) e_{n-1}^b(k-1)$	(12.6.8)
$e_n^b(k) = e_{n-1}^b(k-1) + \Gamma_n^b(k) e_{n-1}^f(k)$	(12.6.9)
$e_n(k) = e_{n-1}(k) + \gamma_n^*(k) e_{n-1}^b(k)$	(12.6.10)
End for n	
$\alpha_N(k) = e_N(k) \varphi_N^{-1}(k)$	(12.6.11)
End for k	

12.7. Нормализованный лестничный алгоритм

Следующей разновидностью лестничных алгоритмов является нормализованный алгоритм [138]. Он представляет собой модификацию алгоритма на основе апостериорных ошибок с обратными связями. Особенностью нормализованного лестничного алгоритма является то, что в нем вместо двух различных видов коэффициентов отражения в части операций линейного предсказания используются коэффициенты отражения одного вида. В нем также требуется выполнять операции извлечения квадратного корня. Его особенностью является и то, что все переменные, участвующие в вычислениях, меньше едини-



цы, что полезно в случае реализации алгоритма в арифметике с фиксированной точкой.

В процессе функционирования нормализованного алгоритма используются нормализованные ошибки

$$\hat{e}_{n-1}^f(k) = \frac{e_{n-1}^f(k)}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}, \quad (12.101)$$

$$\hat{e}_{n-1}^b(k) = \frac{e_{n-1}^b(k)}{\left[E_{n-1,\min}^b(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k)}, \quad (12.102)$$

$$\hat{e}_{n-1}(k) = \frac{e_{n-1}(k)}{\left[E_{n-1,\min}(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k)}, \quad (12.103)$$

где согласно уравнениям (6.84) и (11.74)

$$\begin{aligned} E_{n-1,\min}(k) &= \lambda E_{n-1,\min}(k-1) + e_{n-1}(k) \alpha_{n-1}^*(k) = \\ &= \lambda E_{n-1,\min}(k-1) + e_{n-1}(k) e_{n-1}^*/\varphi_{n-1}(k). \end{aligned} \quad (12.104)$$

Кроме того, используются нормализованные значения коэффициентов $\Gamma_n^f(k)$, $\Gamma_n^b(k)$ и $\gamma_n(k)$, которые согласно уравнениям (12.18), (12.26) и (12.55) определяются как

$$\begin{aligned} \hat{\Gamma}_n^f(k) &= \Gamma_n^f(k) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = -\frac{\Delta_{n-1}(k)}{E_{n-1,\min}^b(k-1)} \cdot \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = \\ &= -\frac{\Delta_{n-1}(k)}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5} \left[E_{n-1,\min}^f(k) \right]^{0.5}}, \end{aligned} \quad (12.105)$$

$$\begin{aligned} \hat{\Gamma}_n^b(k) &= \Gamma_n^b(k) \frac{\left[E_{n-1,\min}^f(k) \right]^{0.5}}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} = -\frac{\Delta_{n-1}(k)}{E_{n-1,\min}^f(k)} \cdot \frac{\left[E_{n-1,\min}^f(k) \right]^{0.5}}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} = \\ &= -\frac{\Delta_{n-1}(k)}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \left[E_{n-1,\min}^b(k-1) \right]^{0.5}} \end{aligned} \quad (12.106)$$

и

$$\begin{aligned} \hat{\gamma}_n(k) &= \gamma_n(k) \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} = -\frac{\kappa_{n-1}(k)}{E_{n-1,\min}^b(k)} \cdot \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} = \\ &= -\frac{\kappa_{n-1}(k)}{\left[E_{n-1,\min}^b(k) \right]^{0.5} \left[E_{n-1,\min}(k) \right]^{0.5}}. \end{aligned} \quad (12.107)$$

Из уравнений (12.105) и (12.106) следует, что

$$\hat{\Gamma}_n^f(k) = \hat{\Gamma}_n^b(k) = \hat{\Gamma}_n(k). \quad (12.108)$$

Согласно уравнениям (12.70), (12.28), (12.101) и (12.102)

$$\begin{aligned}
 \Gamma_n^f(k) &= \Gamma_n^f(k-1) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} e_{n-1}^b(k-1) \alpha_n^{f*}(k) = \\
 &= \Gamma_n^f(k-1) - \frac{e_{n-1}^b(k-1)}{E_{n-1,\min}^b(k-1)} \left[\alpha_{n-1}^{f*}(k) + \Gamma_n^f(k-1) \alpha_{n-1}^{b*}(k-1) \right] = \\
 &= \Gamma_n^f(k-1) - \frac{e_{n-1}^b(k-1)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} \left[e_{n-1}^{f*}(k) + \Gamma_n^f(k-1) e_{n-1}^{b*}(k-1) \right] = \\
 &= \Gamma_n^f(k-1) - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} - \frac{e_{n-1}^b(k-1) e_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} \Gamma_n^f(k-1) = \\
 &= \Gamma_n^f(k-1) \left[1 - \frac{e_{n-1}^b(k-1) e_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} \right] - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} = \\
 &= \Gamma_n^f(k-1) \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)}.
 \end{aligned} \tag{12.109}$$

Далее, используя уравнение (12.109) в (12.105), получим, что

$$\begin{aligned}
 \hat{\Gamma}_n(k) &= \Gamma_n^f(k) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = \left\{ \Gamma_n^f(k-1) \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \right. \\
 &\quad \left. - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} \right\} \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = \\
 &= \Gamma_n^f(k-1) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \\
 &\quad - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1) \varphi_{n-1}(k-1)} \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = \\
 &= \Gamma_n^f(k-1) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \\
 &\quad - \frac{e_{n-1}^b(k-1) e_{n-1}^{f*}(k)}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5} \varphi_{n-1}(k-1) \left[E_{n-1,\min}^f(k) \right]^{0.5}} = \\
 &= \Gamma_n^f(k-1) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{f*}(k).
 \end{aligned} \tag{12.110}$$

Теперь для использования в (12.110) уравнение (12.3.2) преобразуем как

$$\begin{aligned}
 E_{n-1,\min}^f(k) &= \lambda E_{n-1,\min}^f(k-1) + e_{n-1}^f(k) e_{n-1}^{f*}(k) / \varphi_{n-1}(k-1) = \lambda E_{n-1,\min}^f(k-1) + \\
 &\quad + \frac{e_{n-1}^f(k) e_{n-1}^{f*}(k)}{\varphi_{n-1}(k-1)} \cdot \frac{E_{n-1,\min}^f(k)}{E_{n-1,\min}^f(k)} = \lambda E_{n-1,\min}^f(k-1) + \hat{e}_{n-1}^f(k) \hat{e}_{n-1}^{f*}(k) E_{n-1,\min}^f(k),
 \end{aligned} \tag{12.111}$$

откуда

$$E_{n-1,\min}^f(k) = \frac{\lambda E_{n-1,\min}^f(k-1)}{1 - \hat{e}_{n-1}^f(k) \hat{e}_{n-1}^{f*}(k)}. \quad (12.112)$$

Аналогичным образом преобразуем уравнение (12.3.3)

$$\begin{aligned} E_{n-1,\min}^b(k) &= \lambda E_{n-1,\min}^b(k-1) + e_{n-1}^b(k) e_{n-1}^{b*}(k) / \varphi_{n-1}(k) = \lambda E_{n-1,\min}^b(k-1) + \\ &+ \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{\varphi_{n-1}(k)} \cdot \frac{E_{n-1,\min}^b(k)}{E_{n-1,\min}^b(k)} = \lambda E_{n-1,\min}^b(k-1) + \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) E_{n-1,\min}^b(k), \end{aligned} \quad (12.113)$$

откуда

$$E_{n-1,\min}^b(k) = \frac{\lambda E_{n-1,\min}^b(k-1)}{1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k)}. \quad (12.114)$$

Затем используем выражения (12.112) и (12.114) в (12.110), как

$$\begin{aligned} \hat{\Gamma}_n(k) &= \Gamma_n^f(k-1) \frac{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{f*}(k) = \\ &= \Gamma_n^f(k-1) \frac{\left[\frac{\lambda E_{n-1,\min}^b(k-2)}{1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1)} \right]^{0.5}}{\left[\frac{\lambda E_{n-1,\min}^f(k-1)}{1 - \hat{e}_{n-1}^f(k) \hat{e}_{n-1}^{f*}(k)} \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \\ &- \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{f*}(k) = \Gamma_n^f(k-1) \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5}}{\left[\lambda E_{n-1,\min}^f(k-1) \right]^{0.5}} \cdot \frac{\left[1 - \hat{e}_{n-1}^f(k) \hat{e}_{n-1}^{f*}(k) \right]^{0.5}}{\left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right]^{0.5}} \times \\ &\times \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right] - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{f*}(k) = \\ &= \hat{\Gamma}_n^f(k-1) \left[1 - \hat{e}_{n-1}^f(k) \hat{e}_{n-1}^{f*}(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{b*}(k-1) \right]^{0.5} - \hat{e}_{n-1}^b(k-1) \hat{e}_{n-1}^{f*}(k). \end{aligned} \quad (12.115)$$

Потом получим выражение для рекурсивного вычисления $\hat{\gamma}_n(k)$. Согласно уравнениям (12.86), (12.54), (12.102) и (12.103)

$$\begin{aligned} \gamma_n(k) &= \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^b(k) \alpha_{n-1}^*(k) = \\ &= \gamma_n(k-1) - \frac{e_{n-1}^b(k)}{E_{n-1,\min}^b(k)} \left[\alpha_{n-1}^*(k) + \gamma_n(k-1) \alpha_{n-1}^{b*}(k) \right] = \\ &= \gamma_n(k-1) - \frac{e_{n-1}^b(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \left[e_{n-1}^*(k) + \gamma_n(k-1) e_{n-1}^{b*}(k) \right] = \\ &= \gamma_n(k-1) - \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} - \gamma_n(k-1) \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} = \\ &= \gamma_n(k-1) \left[1 - \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \right] - \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} = \\ &= \gamma_n(k-1) \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \frac{e_{n-1}^b(k) e_{n-1}^*(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)}. \end{aligned} \quad (12.116)$$

Далее, используя (12.116) в (12.107), получим, что

$$\begin{aligned}
 \hat{\gamma}_n(k) &= \gamma_n(k) \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} = \\
 &= \left\{ \gamma_n(k-1) \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \frac{e_{n-1}^b(k) e_{n-1}^{*}(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \right\} \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} = \\
 &= \gamma_n(k-1) \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} - \frac{e_{n-1}^b(k) e_{n-1}^{*}(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \times \\
 &\quad \times \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} = \gamma_n(k-1) \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \\
 &\quad - \frac{e_{n-1}^b(k) e_{n-1}^{*}(k)}{\left[E_{n-1,\min}^b(k) \right]^{0.5} \varphi_{n-1}(k) \left[E_{n-1,\min}(k) \right]^{0.5}} = \\
 &= \gamma_n(k-1) \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{*}(k).
 \end{aligned} \tag{12.117}$$

Также преобразуем уравнение (12.104) как

$$\begin{aligned}
 E_{n-1,\min}(k) &= \lambda E_{n-1,\min}(k-1) + e_{n-1}(k) e_{n-1}^{*}(k) / \varphi_{n-1}(k) = \lambda E_{n-1,\min}(k-1) + \\
 &+ \frac{e_{n-1}(k) e_{n-1}^{*}(k)}{\varphi_{n-1}(k)} \cdot \frac{E_{n-1,\min}(k)}{E_{n-1,\min}(k)} = \lambda E_{n-1,\min}(k-1) + \hat{e}_{n-1}(k) \hat{e}_{n-1}^{*}(k) E_{n-1,\min}(k),
 \end{aligned} \tag{12.118}$$

откуда

$$E_{n-1,\min}(k) = \frac{\lambda E_{n-1,\min}(k-1)}{1 - \hat{e}_{n-1}(k) \hat{e}_{n-1}^{*}(k)}. \tag{12.119}$$

Затем используем выражения (12.114) и (12.119) в (12.117). Тогда

$$\begin{aligned}
 \hat{\gamma}_n(k) &= \gamma_n(k-1) \frac{\left[E_{n-1,\min}^b(k) \right]^{0.5}}{\left[E_{n-1,\min}(k) \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{*}(k) = \\
 &= \gamma_n(k-1) \frac{\left[\frac{\lambda E_{n-1,\min}^b(k-1)}{1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k)} \right]^{0.5}}{\left[\frac{\lambda E_{n-1,\min}(k-1)}{1 - \hat{e}_{n-1}(k) \hat{e}_{n-1}^{*}(k)} \right]^{0.5}} \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{*}(k) = \\
 &= \gamma_n(k-1) \frac{\left[\lambda E_{n-1,\min}^b(k-1) \right]^{0.5}}{\left[\lambda E_{n-1,\min}(k-1) \right]^{0.5}} \cdot \frac{\left[1 - \hat{e}_{n-1}(k) \hat{e}_{n-1}^{*}(k) \right]^{0.5}}{\left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right]^{0.5}} \times \\
 &\quad \times \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right] - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{*}(k) = \\
 &= \hat{\gamma}_n(k-1) \left[1 - \hat{e}_{n-1}(k) \hat{e}_{n-1}^{*}(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{b*}(k) \right]^{0.5} - \hat{e}_{n-1}^b(k) \hat{e}_{n-1}^{*}(k).
 \end{aligned} \tag{12.120}$$

Потом преобразуем уравнения (12.19), (12.27) и (12.60) как

$$\begin{aligned}
E_{n,\min}^f(k) &= E_{n-1,\min}^f(k) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \Delta_{n-1}^*(k) \Delta_{n-1}(k) = \\
&= E_{n-1,\min}^f(k) \left[1 - \frac{\Delta_{n-1}^*(k) \Delta_{n-1}(k)}{E_{n-1,\min}^b(k-1) E_{n-1,\min}^f(k)} \right] = E_{n-1,\min}^f(k) \left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right], \tag{12.121}
\end{aligned}$$

$$\begin{aligned}
E_{n,\min}^b(k) &= E_{n-1,\min}^b(k-1) - \left[E_{n-1,\min}^f(k) \right]^{-1} \Delta_{n-1}(k) \Delta_{n-1}^*(k) = \\
&= E_{n-1,\min}^b(k-1) \left[1 - \frac{\Delta_{n-1}(k) \Delta_{n-1}^*(k)}{E_{n-1,\min}^f(k) E_{n-1,\min}^b(k-1)} \right] = E_{n-1,\min}^b(k-1) \left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right], \tag{12.122}
\end{aligned}$$

$$\begin{aligned}
E_{n,\min}(k) &= E_{n-1,\min}(k) + \left[E_{n-1,\min}^b(k) \right]^{-1} \kappa_{n-1}^*(k) \kappa_{n-1}(k) = \\
&= E_{n-1,\min}(k) \left[1 - \frac{\kappa_{n-1}^*(k) \kappa_{n-1}(k)}{E_{n-1,\min}^b(k) E_{n-1,\min}(k)} \right] = E_{n-1,\min}(k) \left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k) \right]. \tag{12.123}
\end{aligned}$$

Из уравнения (12.90) также следует, что

$$\varphi_n(k) = \varphi_{n-1}(k-1) \left[1 - \frac{e_{n-1}^{f*}(k) e_{n-1}^f(k)}{E_{n-1,\min}^f(k) \varphi_{n-1}(k-1)} \right] = \varphi_{n-1}(k-1) \left[1 - e_{n-1}^{f*}(k) e_{n-1}^f(k) \right], \tag{12.124}$$

а из уравнения (12.59) следует, что

$$\begin{aligned}
\varphi_n(k) &= \varphi_{n-1}(k) - \left[E_{n-1,\min}^b(k) \right]^{-1} e_{n-1}^{b*}(k) e_{n-1}^b(k) = \\
&= \varphi_{n-1}(k) \left[1 - \frac{e_{n-1}^{b*}(k) e_{n-1}^b(k)}{E_{n-1,\min}^b(k) \varphi_{n-1}(k)} \right] = \varphi_{n-1}(k) \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k) \right]. \tag{12.125}
\end{aligned}$$

Используя полученные выше соотношения, можно составить уравнения для вычисления нормализованных апостериорных ошибок линейного предсказания и моделирования с помощью уравнений (12.101)–(12.103).

Используя (12.101), (12.29), (12.105), (12.121) и (12.125), получим, что

$$\begin{aligned}
\hat{e}_n^f(k) &= \frac{e_n^f(k)}{\left[E_{n,\min}^f(k) \right]^{0.5} \varphi_n^{0.5}(k-1)} = \\
&= \frac{e_{n-1}^f(k) + \Gamma_n^{f*}(k) e_{n-1}^b(k-1)}{\left[E_{n-1,\min}^f(k) \left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right] \right]^{0.5} \left\{ \varphi_{n-1}(k-1) \left[1 - \hat{e}_{n-1}^{b*}(k-1) \hat{e}_{n-1}^b(k-1) \right] \right\}^{0.5}} = \\
&= \frac{e_{n-1}^f(k) + \Gamma_n^{f*}(k) e_{n-1}^b(k-1)}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} = \\
&= \frac{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k-1) \hat{e}_{n-1}^b(k-1) \right]^{0.5}}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} = \tag{12.126} \\
&= \frac{\frac{e_{n-1}^f(k)}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} + \frac{\Gamma_n^{f*}(k) e_{n-1}^b(k-1)}{\left[E_{n-1,\min}^f(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} \cdot \frac{\left[E_{n,\min}^b(k) \right]^{0.5}}{\left[E_{n,\min}^b(k) \right]^{0.5}}}{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k-1) \hat{e}_{n-1}^b(k-1) \right]^{0.5}} = \\
&= \frac{\hat{e}_{n-1}^f(k) + \hat{\Gamma}_n^*(k) \hat{e}_{n-1}^b(k-1)}{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k-1) \hat{e}_{n-1}^b(k-1) \right]^{0.5}}.
\end{aligned}$$

Также, используя (12.102), (12.31), (12.106), (12.122) и (12.124), получим, что

$$\begin{aligned}
 \hat{e}_n^b(k) &= \frac{e_n^b(k)}{\left[E_{n,\min}^b(k)\right]^{0.5} \varphi_n^{0.5}(k)} = \\
 &= \frac{e_{n-1}^b(k-1) + \Gamma_n^{b*}(k) e_{n-1}^f(k)}{\left\{E_{n-1,\min}^b(k-1) \left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k)\right]\right\}^{0.5} \left\{\varphi_{n-1}(k-1) \left[1 - \hat{e}_{n-1}^{f*}(k) \hat{e}_{n-1}^f(k)\right]\right\}^{0.5}} = \\
 &= \frac{\frac{e_{n-1}^b(k-1) + \Gamma_n^{b*}(k) e_{n-1}^f(k)}{\left[E_{n-1,\min}^b(k-1)\right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}}{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]^{0.5}} = \\
 &= \frac{\frac{e_{n-1}^b(k-1)}{\left[E_{n-1,\min}^b(k-1)\right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} + \frac{\Gamma_n^{b*}(k) e_{n-1}^f(k)}{\left[E_{n-1,\min}^b(k-1)\right]^{0.5} \varphi_{n-1}^{0.5}(k-1)} \cdot \frac{\left[E_{n,\min}^f(k)\right]^{0.5}}{\left[E_{n,\min}^f(k)\right]^{0.5}}}{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{f*}(k) \hat{e}_{n-1}^f(k)\right]^{0.5}} = \\
 &= \frac{\hat{e}_{n-1}^b(k-1) + \hat{\Gamma}_n(k) \hat{e}_{n-1}^f(k)}{\left[1 - \hat{\Gamma}_n(k) \hat{\Gamma}^*(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]^{0.5}}.
 \end{aligned} \tag{12.127}$$

Затем, используя (12.103), (12.56), (12.107), (12.123) и (12.125), получим, что

$$\begin{aligned}
 \hat{e}_n(k) &= \frac{e_n(k)}{\left[E_{n,\min}(k)\right]^{0.5} \varphi_n^{0.5}(k)} = \\
 &= \frac{e_{n-1}(k) + \gamma_n^*(k) e_{n-1}^b(k)}{\left\{E_{n-1,\min}(k) \left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k)\right]\right\}^{0.5} \left\{\varphi_{n-1}(k) \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]\right\}^{0.5}} = \\
 &= \frac{\frac{e_{n-1}(k) + \gamma_n^*(k) e_{n-1}^b(k)}{\left[E_{n-1,\min}(k)\right]^{0.5} \varphi_{n-1}^{0.5}(k)}}{\left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]^{0.5}} = \\
 &= \frac{\frac{e_{n-1}(k)}{\left[E_{n-1,\min}(k)\right]^{0.5} \varphi_{n-1}^{0.5}(k)} + \frac{\gamma_n^*(k) e_{n-1}^b(k)}{\left[E_{n-1,\min}(k)\right]^{0.5} \varphi_{n-1}^{0.5}(k)} \cdot \frac{\left[E_{n,\min}^b(k)\right]^{0.5}}{\left[E_{n,\min}^b(k)\right]^{0.5}}}{\left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]^{0.5}} = \\
 &= \frac{\hat{e}_{n-1}(k) + \hat{\gamma}_n^*(k) \hat{e}_{n-1}^b(k)}{\left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k)\right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k)\right]^{0.5}}.
 \end{aligned} \tag{12.128}$$

Из уравнения (12.128) следует, что

$$e_n(k) = \left[E_{n,\min}(k)\right]^{0.5} \varphi_n^{0.5}(k) \hat{e}_n(k) = \sigma_n(k) \hat{e}_n(k), \tag{12.129}$$

где согласно уравнениям (12.123) и (12.125)

$$\begin{aligned}
 \sigma_n(k) &= \left[E_{n,\min}(k) \right]^{0.5} \varphi_n^{0.5}(k) = \\
 &= \left\{ E_{n-1,\min}(k) \left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k) \right] \right\}^{0.5} \left\{ \varphi_{n-1}(k) \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k) \right] \right\}^{0.5} = \\
 &= \left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k) \right]^{0.5} \left[E_{n-1,\min}(k) \right]^{0.5} \varphi_{n-1}^{0.5}(k) = \\
 &= \left[1 - \hat{\gamma}_n^*(k) \hat{\gamma}_n(k) \right]^{0.5} \left[1 - \hat{e}_{n-1}^{b*}(k) \hat{e}_{n-1}^b(k) \right]^{0.5} \sigma_{n-1}(k).
 \end{aligned} \tag{12.130}$$

Последовательность рассмотренных выше соотношений составляет нормализованный лестничный RLS-алгоритм на основе апостериорных ошибок с обратными связями, приведенный в табл. 12.7.

Общая инициализация алгоритма в табл. 12.7 осуществляется так, как показано в уравнении (12.7.0). Здесь обозначенная инициализация энергий ошибок следует из аналогичной инициализации других алгоритмов настоящей главы при $n-1=0$.

Из уравнений (12.6.1)–(12.6.3) и (12.130) следует, что на каждой k -й итерации ряд переменных нормализованного алгоритма дополнительно инициализируются, как показано в уравнениях (12.7.1)–(12.7.4).

Затем последовательно выполняются процедуры (12.110), (12.117), (12.126)–(12.129), (12.130) и (12.125). Однократные множители, участвующие в этих выражениях, вычисляются один раз с помощью формул (12.7.5)–(12.7.7), (12.7.10) и (12.7.11).

Структура вычисления ошибок в соответствии с табл. 12.7 приведена на рис. 12.3, а структура адаптивного фильтра на основе таких ошибок — на рис. 12.4.

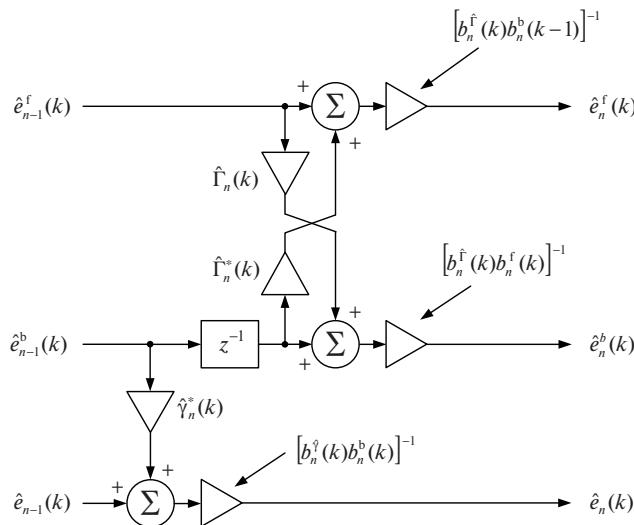


Рис. 12.3. Вычисление нормализованных апостериорных ошибок

Таблица 12.7. Нормализованный лестничный RLS-алгоритм с обратными связями на основе апостериорных ошибок

Вычисления	Ссылки
Initialization: $\hat{e}_{n-1}^b(0) = 0$, $b_n^b(0) = 1$, $\hat{\Gamma}_n(0) = 0$, $\hat{\gamma}_n(0) = 0$, $n = 1, 2, \dots, N$	(12.7.0)
$E_0(0) = \delta^2$, $E_0^f(0) = \delta^2$, $E_0^b(0) = \delta^2$, $\varphi_0(0) = 1$	
For $k = 1, 2, \dots, K$	
$E_{0,\min}^f(k) = \lambda E_{0,\min}^f(k-1) + x(k)x^*(k)$	(12.7.1)
$E_{0,\min}^b(k) = \lambda E_{0,\min}^b(k-1) + x(k)x^*(k)$	(12.7.2)
$E_{0,\min}(k) = \lambda E_{0,\min}(k-1) + d(k)d^*(k)$	(12.7.3)
$\hat{e}_0^f(k) = x(k) / [E_{0,\min}^f(k)]^{0.5}$, $\hat{e}_0^b(k) = x(k) / [E_{0,\min}^b(k)]^{0.5}$,	(12.7.4)
$\hat{e}_0(k) = d(k) / [E_{0,\min}(k)]^{0.5}$, $\sigma_0(k) = [E_{0,\min}(k)]^{0.5}$, $\varphi_0(k) = 1$	
For $n = 1, 2, \dots, N$	
$b_n^f(k) = [1 - \hat{e}_{n-1}^f(k)\hat{e}_{n-1}^{f*}(k)]^{0.5}$	(12.7.5)
$b_n^b(k) = [1 - \hat{e}_{n-1}^b(k)\hat{e}_{n-1}^{b*}(k)]^{0.5}$	(12.7.6)
$b_n(k) = [1 - \hat{e}_{n-1}(k)\hat{e}_{n-1}^*(k)]^{0.5}$	(12.7.7)
$\hat{\Gamma}_n^f(k) = \hat{\Gamma}_n^f(k-1)b_n^f(k)b_n^f(k-1) - \hat{e}_{n-1}^b(k-1)\hat{e}_{n-1}^{f*}(k)$	(12.7.8)
$\hat{\gamma}_n(k) = \hat{\gamma}_n(k-1)b_n(k)b_n^b(k) - \hat{e}_{n-1}^b(k)\hat{e}_{n-1}^*(k)$	(12.7.9)
$b_n^{\hat{f}}(k) = [1 - \hat{\Gamma}_n(k)\hat{\Gamma}_n^*(k)]^{0.5}$	(12.7.10)
$b_n^{\hat{b}}(k) = [1 - \hat{\gamma}_n(k)\hat{\gamma}_n^*(k)]^{0.5}$	(12.7.11)
$\hat{e}_n^f(k) = \frac{\hat{e}_{n-1}^f(k) + \hat{\Gamma}_n^*(k)\hat{e}_{n-1}^b(k-1)}{b_n^{\hat{f}}(k)b_n^f(k-1)}$	(12.7.12)
$\hat{e}_n^b(k) = \frac{\hat{e}_{n-1}^b(k-1) + \hat{\Gamma}_n(k)\hat{e}_{n-1}^f(k)}{b_n^{\hat{b}}(k)b_n^f(k)}$	(12.7.13)
$\hat{e}_n(k) = \frac{\hat{e}_{n-1}(k) + \hat{\gamma}_n(k)\hat{e}_{n-1}^b(k)}{b_n^{\hat{b}}(k)b_n^b(k)}$	(12.7.14)
$\sigma_n(k) = b_n^{\hat{b}}(k)b_n^b(k)\sigma_{n-1}(k)$	(12.7.15)
$\varphi_n(k) = \varphi_{n-1}(k)[b_n^b(k)]^2$	(12.7.16)
End for n	
$e_N(k) = \sigma_N(k)\hat{e}_N(k)$	(12.7.17)
$\alpha_N(k) = e_N(k)/\varphi_N(k)$	(12.6.18)
End for k	

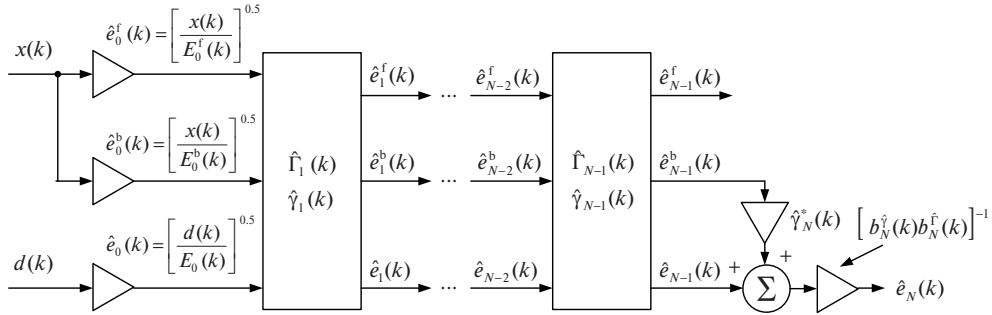


Рис. 12.4. Нормализованный адаптивный фильтр на основе апостериорных ошибок

12.8. Алгоритм на основе QR-разложения с операциями извлечения квадратного корня

Другой тип нормализованного лестничного алгоритма — это алгоритм на основе апостериорных ошибок с использованием вращений Гивенса [146]. В нем нормализация ошибок осуществляется как

$$\tilde{e}_{n-1}^f(k) = e_{n-1}^f(k) / \varphi_{n-1}^{0.5}(k-1), \quad (12.131)$$

$$\tilde{e}_{n-1}^b(k) = e_{n-1}^b(k) / \varphi_{n-1}^{0.5}(k), \quad (12.132)$$

$$\tilde{e}_{n-1}(k) = e_{n-1}(k) / \varphi_{n-1}^{0.5}(k). \quad (12.133)$$

Формулы (12.131)–(12.133) позволяют выразить энергию ошибок линейного предсказания (12.3.2) и (12.3.3) лишь в терминах нормализованных ошибок как

$$\begin{aligned} E_{n-1,\min}^f(k) &= \lambda E_{n-1,\min}^f(k-1) + e_{n-1}^f(k) e_{n-1}^{f*} / \varphi_{n-1}(k-1) = \\ &= [\tilde{E}_{n,\min}^f(k)]^2 = \lambda [\tilde{E}_{n,\min}^f(k-1)]^2 + \tilde{e}_n^f(k) \tilde{e}_n^{f*}(k) \end{aligned} \quad (12.134)$$

и

$$\begin{aligned} E_{n-1,\min}^b(k) &= \lambda E_{n-1,\min}^b(k-1) + e_{n-1}^b(k) e_{n-1}^{b*} / \varphi_{n-1}(k) = \\ &= [\tilde{E}_{n,\min}^b(k)]^2 = \lambda [\tilde{E}_{n,\min}^b(k-1)]^2 + \tilde{e}_n^b(k) \tilde{e}_n^{b*}(k). \end{aligned} \quad (12.135)$$

Из раздела 11.8 следует, что

$$\varphi_n(k) = \frac{\lambda E_{n-1,\min}^b(k-1) \varphi_{n-1}(k)}{E_{n-1,\min}^b(k)}. \quad (12.136)$$

Тогда согласно выражениям (12.131), (12.28) и (12.136)

$$\begin{aligned}
\tilde{e}_n^f(k) &= \frac{e_n^f(k)}{\varphi_n^{0.5}(k-1)} = \frac{\alpha_n^f(k)\varphi_n(k-1)}{\varphi_n^{0.5}(k-1)} = \alpha_n^f(k)\varphi_n^{0.5}(k-1) = \\
&= \left[\alpha_{n-1}^f(k) + \Gamma_n^{f*}(k-1)\alpha_{n-1}^b(k-1) \right] \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} = \\
&= \alpha_{n-1}^f(k) \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} + \\
&+ \Gamma_n^{f*}(k-1)\alpha_{n-1}^b(k-1) \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} = \tilde{e}_{n-1}^f(k) \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5}}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} + \\
&+ \Gamma_n^{f*}(k-1)\tilde{e}_{n-1}^b(k-1) \frac{\left[\lambda E_{n-1,\min}^b(k-2) \right]^{0.5}}{\left[E_{n-1,\min}^b(k-1) \right]^{0.5}} = \tilde{e}_{n-1}^f(k) \frac{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2)}{\tilde{E}_{n-1,\min}^b(k-1)} + \\
&+ \Gamma_n^{f*}(k-1)\tilde{e}_{n-1}^b(k-1) \frac{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2)}{\tilde{E}_{n-1,\min}^b(k-1)} = \tilde{e}_{n-1}^f(k) c_n^b(k-1) + \lambda^{0.5} q_n^{f*}(k-1) s_n^b(k-1),
\end{aligned} \tag{12.137}$$

где

$$\tilde{e}_{n-1}^b(k) = \frac{e_{n-1}^b(k)}{\varphi_{n-1}^{0.5}(k)} = \frac{\alpha_{n-1}^b(k)\varphi_{n-1}(k)}{\varphi_{n-1}^{0.5}(k)} = \alpha_{n-1}^b(k)\varphi_{n-1}^{0.5}(k), \tag{12.138}$$

$$c_n^b(k) = \frac{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1)}{\tilde{E}_{n-1,\min}^b(k)}, \tag{12.139}$$

$$s_n^b(k) = \frac{\tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1,\min}^b(k)}, \tag{12.140}$$

$$q_n^f(k) = \Gamma_n^f(k) \tilde{E}_{n-1,\min}^b(k-1). \tag{12.141}$$

Для рекурсивного вычисления переменной $q_n^f(k)$ в уравнении (12.141) необходимо использовать выражение (12.109) в виде

$$\begin{aligned}
\Gamma_n^f(k) &= \Gamma_n^f(k-1) \left[1 - \frac{e_{n-1}^b(k-1)e_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1)\varphi_{n-1}(k)} \right] - \frac{e_{n-1}^b(k-1)e_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1)\varphi_{n-1}(k)} = \\
&= \Gamma_n^f(k-1) \left[1 - \frac{\tilde{e}_{n-1}^b(k-1)\tilde{e}_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1)} \right] - \frac{\tilde{e}_{n-1}^b(k-1)\tilde{e}_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1)},
\end{aligned} \tag{12.142}$$

а из (12.135) определить

$$\frac{\lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2}{\left[\tilde{E}_{n,\min}^b(k) \right]^2} = 1 - \frac{\tilde{e}_n^b(k)\tilde{e}_n^{b*}(k)}{\left[\tilde{E}_{n,\min}^b(k) \right]^2}. \tag{12.143}$$

Тогда, с учетом (12.142) и (12.143), уравнение (12.141) преобразуется как

$$\begin{aligned}
 q_n^f(k) &= \Gamma_n^f(k) \tilde{E}_{n-1, \min}^b(k-1) = \\
 &= \left\{ \Gamma_n^f(k-1) \left[1 - \frac{\tilde{e}_{n-1}^b(k-1) \tilde{e}_{n-1}^{b*}(k-1)}{E_{n-1, \min}^b(k-1)} \right] - \frac{\tilde{e}_{n-1}^b(k-1) \tilde{e}_{n-1}^{f*}(k)}{E_{n-1, \min}^b(k-1)} \right\} \tilde{E}_{n-1, \min}^b(k-1) = \\
 &= \left\{ \Gamma_n^f(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^b(k-2) \right]^2}{\left[\tilde{E}_{n, \min}^b(k-1) \right]^2} - \frac{\tilde{e}_{n-1}^b(k-1) \tilde{e}_{n-1}^{f*}(k)}{\left[\tilde{E}_{n-1, \min}^b(k-1) \right]^2} \right\} \tilde{E}_{n-1, \min}^b(k-1) = \\
 &= \Gamma_n^f(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^b(k-2) \right]^2}{\left[\tilde{E}_{n, \min}^b(k-1) \right]^2} \tilde{E}_{n-1, \min}^b(k-1) - \\
 &\quad - \frac{\tilde{e}_{n-1}^b(k-1) \tilde{e}_{n-1}^{f*}(k)}{\left[\tilde{E}_{n-1, \min}^b(k-1) \right]^2} \tilde{E}_{n-1, \min}^b(k-1) = \lambda^{0.5} \Gamma_n^f(k-1) \tilde{E}_{n, \min}^b(k-2) \frac{\lambda^{0.5} \tilde{E}_{n, \min}^b(k-2)}{\tilde{E}_{n-1, \min}^b(k-1)} - \\
 &\quad - \frac{\tilde{e}_{n-1}^b(k-1) \tilde{e}_{n-1}^{f*}(k)}{\tilde{E}_{n-1, \min}^b(k-1)} \tilde{e}_{n-1}^f(k) = \lambda^{0.5} q_n^f(k-1) c_n^b(k-1) - s_n^b(k-1) \tilde{e}_{n-1}^{f*}(k). \tag{12.144}
 \end{aligned}$$

Выражения (12.137) и (12.144) можно представить и как результат приведения к треугольному виду с помощью вращения Гивенса, последовательности матриц

$$\begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-2) & \lambda^{0.5} q_n^{f*}(k-1) \\ -\tilde{e}_{n-1}^b(k-1) & \tilde{e}_{n-1}^f(k) \end{bmatrix}$$

как

$$\begin{aligned}
 &\begin{bmatrix} c_n^b(k-1) & -s_n^{b*}(k-1) \\ s_n^b(k-1) & c_n^b(k-1) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-2) & \lambda^{0.5} q_n^{f*}(k-1) \\ -\tilde{e}_{n-1}^b(k-1) & \tilde{e}_{n-1}^f(k) \end{bmatrix} = \\
 &= \begin{bmatrix} \tilde{E}_{n-1, \min}^b(k-1) & q_n^{f*}(k) \\ 0 & \tilde{e}_n^f(k) \end{bmatrix}. \tag{12.145}
 \end{aligned}$$

Справедливость этого уравнения легко проверяется путем перемножения матриц в его левой части и использования выражений (12.139) и (12.140).

Действительно,

$$\begin{aligned}
 &c_n^b(k-1) \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-2) - s_n^{b*}(k-1) \left[-\tilde{e}_{n-1}^b(k-1) \right] = \\
 &= \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-2)}{\tilde{E}_{n-1, \min}^b(k-1)} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-2) - \frac{\tilde{e}_{n-1}^{b*}(k-1)}{\tilde{E}_{n-1, \min}^b(k-1)} \left[-\tilde{e}_{n-1}^b(k-1) \right] = \\
 &= \frac{\lambda \left[\tilde{E}_{n-1, \min}^b(k-2) \right]^2}{\tilde{E}_{n-1, \min}^b(k-1)} + \frac{\tilde{e}_{n-1}^{b*}(k-1) \tilde{e}_{n-1}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k-1)} = \tilde{E}_{n-1, \min}^b(k-1), \tag{12.146}
 \end{aligned}$$

откуда следует уравнение (12.135) для итерации с номером $k-1$.

Аналогично доказывается, что

$$s_n^b(k-1)\lambda^{0.5}\tilde{E}_{n-1,\min}^b(k-2)+c_n^b(k-1)\left[-\tilde{e}_{n-1}^b(k-1)\right]=\frac{\tilde{e}_{n-1}^b(k-1)}{\tilde{E}_{n-1,\min}^b(k-1)}\lambda^{0.5}\tilde{E}_{n-1,\min}^b(k-2)+\frac{\lambda^{0.5}\tilde{E}_{n-1,\min}^b(k-2)}{\tilde{E}_{n-1,\min}^b(k-1)}\left[-\tilde{e}_{n-1}^b(k-1)\right]=0. \quad (12.147)$$

Элементы матрицы $\begin{bmatrix} c_n^b(k-1) & -s_n^{b*}(k-1) \\ s_n^b(k-1) & c_n^b(k-1) \end{bmatrix}$ в уравнении (12.145) носят названия синусов и косинусов углов вращения, так как они удовлетворяют известному тригонометрическому соотношению

$$\begin{aligned} c_n^b(k-1)c_n^b(k-1)+s_n^b(k-1)s_n^{b*}(k-1) &= \frac{\lambda^{0.5}\tilde{E}_{n-1,\min}^b(k-2)}{\tilde{E}_{n-1,\min}^b(k-1)} \cdot \frac{\lambda^{0.5}\tilde{E}_{n-1,\min}^b(k-2)}{\tilde{E}_{n-1,\min}^b(k-1)} + \\ &+ \frac{\tilde{e}_{n-1}^b(k-1)}{\tilde{E}_{n-1,\min}^b(k-1)} \cdot \frac{\tilde{e}_{n-1}^{b*}(k-1)}{\tilde{E}_{n-1,\min}^b(k-1)} = \frac{\lambda\left[\tilde{E}_{n-1,\min}^b(k-2)\right]^2}{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2} + \frac{\tilde{e}_{n-1}^b(k-1)\tilde{e}_{n-1}^{b*}(k-1)}{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2} = 1. \end{aligned} \quad (12.148)$$

Значения косинусов $c_{n-1}^b(k-1) \leq 1$, так как согласно уравнению (12.135) энергия ошибок от итерации к итерации возрастает, а потому отношения этих энергий в соседних итерациях согласно уравнению (12.139) всегда величина, не превышающая единицы.

Аналогично согласно уравнениям (12.140), (12.139) и (12.135) имеем, что

$$\begin{aligned} |s_n^b(k-1)| &= \sqrt{s_n^b(k-1)s_n^{b*}(k-1)} = \sqrt{\frac{\tilde{e}_{n-1}^b(k-1)\tilde{e}_{n-1}^{b*}(k-1)}{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2}} = \\ &= \sqrt{\frac{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2 - \lambda\left[\tilde{E}_{n-1,\min}^b(k-2)\right]^2}{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2}} = \sqrt{1 - \frac{\lambda\left[\tilde{E}_{n-1,\min}^b(k-2)\right]^2}{\left[\tilde{E}_{n-1,\min}^b(k-1)\right]^2}} \leq 1. \end{aligned} \quad (12.149)$$

Таким же образом получаются уравнения для рекурсивного вычисления ошибок $e_n^b(k)$. Из уравнения (11.103) следует, что

$$\varphi_n(k) = \frac{\lambda E_{n-1,\min}^f(k-1)\varphi_{n-1}(k-1)}{E_{n-1,\min}^f(k)}. \quad (12.150)$$

Тогда согласно уравнениям (12.132), (12.29) и (12.150)

$$\begin{aligned} \tilde{e}_{n-1}^b(k) &= \frac{e_{n-1}^b(k)}{\varphi_{n-1}^{0.5}(k)} = \frac{\alpha_{n-1}^b(k)\varphi_{n-1}(k)}{\varphi_{n-1}^{0.5}(k)} = \alpha_{n-1}^b(k)\varphi_{n-1}^{0.5}(k) = \\ &= \left[\alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1)\alpha_{n-1}^f(k) \right] \frac{\left[\lambda E_{n-1,\min}^f(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} = \\ &= \alpha_{n-1}^b(k-1) \frac{\left[\lambda E_{n-1,\min}^f(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1,\min}^f(k) \right]^{0.5}} + \end{aligned} \quad (12.151)$$

$$\begin{aligned}
 & + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k) \frac{\left[\lambda E_{n-1, \min}^f(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k-1)}{\left[E_{n-1, \min}^f(k) \right]^{0.5}} = \\
 & = \tilde{e}_{n-1}^b(k-1) \frac{\left[\lambda E_{n-1, \min}^f(k-1) \right]^{0.5}}{\left[E_{n-1, \min}^f(k) \right]^{0.5}} + \Gamma_n^{b*}(k-1) \tilde{e}_{n-1}^f(k) \frac{\left[\lambda E_{n-1, \min}^f(k-1) \right]^{0.5}}{\left[E_{n-1, \min}^f(k) \right]^{0.5}} = \\
 & = e_{n-1}^b(k-1) \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} + \Gamma_n^{b*}(k-1) e_{n-1}^f(k) \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} = \\
 & = \tilde{e}_{n-1}^b(k-1) c_n^f(k) + \lambda^{0.5} q_n^{b*}(k-1) s_n^f(k),
 \end{aligned}$$

где

$$\tilde{e}_{n-1}^f(k) = \frac{e_{n-1}^f(k)}{\varphi_{n-1}^{0.5}(k-1)} = \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1)}{\varphi_{n-1}^{0.5}(k-1)} = \alpha_{n-1}^f(k) \varphi_{n-1}^{0.5}(k), \quad (12.152)$$

$$c_n^f(k) = \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)}, \quad (12.153)$$

$$s_n^f(k) = \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1, \min}^f(k)}, \quad (12.154)$$

$$q_n^b(k) = \Gamma_n^b(k) \tilde{E}_{n-1, \min}^f(k). \quad (12.155)$$

Для рекурсивного вычисления переменных $q_n^b(k)$ требуется в уравнении (12.155) использовать выражение (12.78) в виде

$$\begin{aligned}
 \Gamma_n^b(k) & = \Gamma_n^b(k-1) - \left[E_{n-1, \min}^f(k) \right]^{-1} e_{n-1}^f(k-1) \alpha_n^{b*}(k) = \\
 & = \Gamma_n^b(k-1) - \frac{e_{n-1}^f(k)}{E_{n-1, \min}^f(k)} \left[\alpha_{n-1}^{b*}(k-1) + \Gamma_n^b(k-1) \alpha_{n-1}^{f*}(k) \right] = \\
 & = \Gamma_n^b(k-1) - \frac{e_{n-1}^f(k)}{E_{n-1, \min}^f(k) \varphi_{n-1}(k-1)} \left[e_{n-1}^{b*}(k-1) + \Gamma_n^b(k-1) e_{n-1}^{f*}(k) \right] = \\
 & = \Gamma_n^b(k-1) - \frac{e_{n-1}^f(k)}{E_{n-1, \min}^f(k) \varphi_{n-1}(k-1)} e_{n-1}^{b*}(k-1) - \frac{e_{n-1}^f(k)}{E_{n-1, \min}^f(k) \varphi_{n-1}(k-1)} \Gamma_n^b(k-1) e_{n-1}^{f*}(k) = \quad (12.156) \\
 & = \Gamma_n^b(k-1) \left[1 - \frac{e_{n-1}^f(k) e_{n-1}^{f*}(k)}{E_{n-1, \min}^f(k) \varphi_{n-1}(k-1)} \right] - \frac{e_{n-1}^f(k) e_{n-1}^{b*}(k-1)}{E_{n-1, \min}^f(k) \varphi_{n-1}(k-1)} = \\
 & = \Gamma_n^b(k-1) \left[1 - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{f*}(k)}{E_{n-1, \min}^f(k)} \right] - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{b*}(k-1)}{E_{n-1, \min}^f(k)},
 \end{aligned}$$

а также из (12.134) определить

$$\frac{\lambda \left[\tilde{E}_{n, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n, \min}^f(k) \right]^2} = 1 - \frac{\tilde{e}_n^f(k) \tilde{e}_n^{f*}(k)}{\left[\tilde{E}_{n, \min}^f(k) \right]^2}. \quad (12.157)$$

Тогда, с учетом (12.156) и (12.157), уравнение (12.155) преобразуется как

$$\begin{aligned}
 q_n^b(k) &= \Gamma_n^b(k) \tilde{E}_{n-1, \min}^f(k) = \\
 &= \left\{ \Gamma_n^b(k-1) \left[1 - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{f*}(k)}{\tilde{E}_{n-1, \min}^f(k)} \right] - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{b*}(k-1)}{\tilde{E}_{n-1, \min}^f(k)} \right\} \tilde{E}_{n-1, \min}^f(k) = \\
 &= \left\{ \Gamma_n^b(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n, \min}^f(k) \right]^2} - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{b*}(k-1)}{\left[\tilde{E}_{n, \min}^f(k) \right]^2} \right\} \tilde{E}_{n-1, \min}^f(k) = \\
 &= \Gamma_n^b(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n, \min}^f(k) \right]^2} \tilde{E}_{n-1, \min}^f(k) - \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{b*}(k-1)}{\left[\tilde{E}_{n, \min}^f(k) \right]^2} \tilde{E}_{n-1, \min}^f(k) = \\
 &= \lambda^{0.5} \Gamma_n^b(k-1) \tilde{E}_{n-1, \min}^f(k-1) \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} - \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1, \min}^f(k)} \tilde{e}_{n-1}^{b*}(k-1) = \\
 &= \lambda^{0.5} q_n^b(k-1) c_n^f(k) - s_n^f(k) \tilde{e}_{n-1}^{b*}(k-1).
 \end{aligned} \tag{12.158}$$

Выражения (12.151) и (12.158) можно представить и как результат приведения к треугольному виду последовательности матриц

$$\begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) & \lambda^{0.5} q_n^{b*}(k-1) \\ -\tilde{e}_{n-1}^f(k) & \tilde{e}_{n-1}^b(k-1) \end{bmatrix}$$

как

$$\begin{aligned}
 \begin{bmatrix} c_n^f(k) & -s_n^{f*}(k) \\ s_n^f(k) & c_n^f(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) & \lambda^{0.5} q_n^{b*}(k-1) \\ -\tilde{e}_{n-1}^f(k) & \tilde{e}_{n-1}^b(k-1) \end{bmatrix} &= \\
 \begin{bmatrix} \tilde{E}_{n-1, \min}^f(k) & q_n^{b*}(k) \\ 0 & \tilde{e}_n^b(k) \end{bmatrix}.
 \end{aligned} \tag{12.159}$$

Справедливость этого уравнения легко проверяется путем перемножения матриц в его левой части и использования (12.153) и (12.154).

Действительно,

$$\begin{aligned}
 c_n^f(k) \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) - s_n^{f*}(k) \left[-\tilde{e}_{n-1}^f(k) \right] &= \\
 = \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) - & \\
 - \frac{\tilde{e}_{n-1}^{f*}(k)}{\tilde{E}_{n-1, \min}^f(k)} \left[-\tilde{e}_{n-1}^f(k) \right] &= \\
 = \frac{\lambda \left[\tilde{E}_{n-1, \min}^f(k-1) \right]^2}{\tilde{E}_{n-1, \min}^f(k)} + \frac{\tilde{e}_{n-1}^{f*}(k) \tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1, \min}^f(k)} &= \tilde{E}_{n-1, \min}^f(k),
 \end{aligned} \tag{12.160}$$

откуда следует уравнение (12.134) для текущей итерации с номером k .

Аналогично доказывается, что

$$\begin{aligned} s_n^f(k) \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) + c_n^f(k) \left[-\tilde{e}_{n-1}^f(k) \right] = \\ = \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1, \min}^f(k)} \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) + \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} \left[-\tilde{e}_{n-1}^f(k) \right] = 0. \end{aligned} \quad (12.161)$$

Элементы матрицы

$$\begin{bmatrix} c_n^f(k) & -s_n^{f*}(k) \\ s_n^f(k) & c_n^f(k) \end{bmatrix}$$

в уравнении (12.161) также носят названия синусов и косинусов углов вращения, так как они удовлетворяют известному тригонометрическому соотношению

$$\begin{aligned} c_n^f(k) c_n^f(k) + s_n^f(k) s_n^{f*}(k) = \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^f(k)} \cdot \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)}{\tilde{E}_{n-1, \min}^b(k)} + \\ + \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1, \min}^f(k)} \cdot \frac{\tilde{e}_{n-1}^{f*}(k)}{\tilde{E}_{n-1, \min}^f(k)} = \frac{\lambda \left[\tilde{E}_{n-1, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2} + \frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{f*}(k)}{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2} = 1. \end{aligned} \quad (12.162)$$

Значения косинусов $c_{n-1}^f(k) \leq 1$, так как согласно уравнению (12.134) энергия ошибок от итерации к итерации возрастает, а потому отношения этих энергий в соседних итерациях согласно уравнению (12.153) всегда величина, не превышающая единицы.

Аналогично, в соответствии с уравнениями (12.154), (12.153) и (12.134)

$$\begin{aligned} |s_n^f(k)| = \sqrt{s_n^f(k) s_n^{f*}(k)} = \sqrt{\frac{\tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{f*}(k)}{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2}} = \\ = \sqrt{\frac{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2 - \lambda \left[\tilde{E}_{n-1, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2}} = \sqrt{1 - \frac{\lambda \left[\tilde{E}_{n-1, \min}^f(k-1) \right]^2}{\left[\tilde{E}_{n-1, \min}^f(k) \right]^2}} \leq 1. \end{aligned} \quad (12.163)$$

Таким же образом получаются уравнения для рекурсивного вычисления ошибок $\tilde{e}_n(k)$. Согласно уравнениям (12.133), (12.54) и (12.136)

$$\begin{aligned} \tilde{e}_n(k) = \frac{e_n(k)}{\varphi_n^{0.5}(k)} = \frac{\alpha_n(k) \varphi_n(k)}{\varphi_n^{0.5}(k)} = \alpha_n(k) \varphi_n^{0.5}(k) = \\ = \left[\alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k) \right] \frac{\left[\lambda E_{n-1, \min}^b(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k)}{\left[E_{n-1, \min}^b(k) \right]^{0.5}} = \\ = \alpha_{n-1}(k) \frac{\left[\lambda E_{n-1, \min}^b(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k)}{\left[E_{n-1, \min}^b(k) \right]^{0.5}} + \end{aligned} \quad (12.164)$$

$$\begin{aligned}
& + \gamma_n^*(k-1) \alpha_{n-1}^b(k) \frac{\left[\lambda E_{n-1, \min}^b(k-1) \right]^{0.5} \varphi_{n-1}^{0.5}(k)}{\left[E_{n-1, \min}^b(k) \right]^{0.5}} = \\
& = \tilde{e}_{n-1}(k) \frac{\left[\lambda E_{n-1, \min}^b(k-1) \right]^{0.5}}{\left[E_{n-1, \min}^b(k) \right]^{0.5}} + \gamma_n^*(k-1) \alpha_{n-1}^b(k) \frac{\left[\lambda E_{n-1, \min}^b(k-1) \right]^{0.5}}{\left[E_{n-1, \min}^b(k) \right]^{0.5}} = \\
& = \tilde{e}_{n-1}(k) \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k)} + \gamma_n^*(k-1) \alpha_{n-1}^b(k) \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k)} = \\
& = \tilde{e}_{n-1}(k) c_n^b(k) + \lambda^{0.5} q_n^*(k-1) s_n^b(k),
\end{aligned}$$

где

$$q_n(k) = \gamma_n(k) \tilde{E}_{n-1, \min}^b(k). \quad (12.165)$$

Для рекурсивного вычисления переменных $q_n^f(k)$ требуется в уравнении (12.165) использовать выражение (12.116) в виде

$$\begin{aligned}
\gamma_n(k) &= \gamma_n(k-1) \left[1 - \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k) \varphi_{n-1}(k)} \right] - \frac{e_{n-1}^b(k) e_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k) \varphi_{n-1}(k)} = \\
&= \gamma_n(k-1) \left[1 - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)} \right] - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)},
\end{aligned} \quad (12.166)$$

а также использовать уравнение (12.143).

Тогда, с учетом выражений (12.166) и (12.143), уравнение (12.165) преобразуется как

$$\begin{aligned}
q_n(k) &= \gamma_n(k) \tilde{E}_{n-1, \min}^b(k) = \gamma_n(k-1) \left[1 - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)} \right] - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)} = \\
&= \left\{ \gamma_n(k-1) \left[1 - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)} \right] - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{E_{n-1, \min}^b(k)} \right\} \tilde{E}_{n-1, \min}^b(k) = \\
&= \left\{ \gamma_n(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^b(k-1) \right]^2}{\left[\tilde{E}_{n, \min}^b(k) \right]^2} - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{\left[\tilde{E}_{n-1, \min}^b(k) \right]^2} \right\} \tilde{E}_{n-1, \min}^b(k) = \\
&= \gamma_n(k-1) \frac{\lambda \left[\tilde{E}_{n, \min}^b(k-1) \right]^2}{\left[\tilde{E}_{n, \min}^b(k) \right]^2} \tilde{E}_{n-1, \min}^b(k) - \frac{\tilde{e}_{n-1}(k) \tilde{e}_{n-1}^{b*}(k)}{\left[\tilde{E}_{n-1, \min}^b(k) \right]^2} \tilde{E}_{n-1, \min}^b(k) = \\
&= \lambda^{0.5} \gamma_n(k-1) \tilde{E}_{n, \min}^b(k-1) \frac{\lambda^{0.5} \tilde{E}_{n, \min}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k)} - \frac{\tilde{e}_{n-1}(k)}{\tilde{E}_{n-1, \min}^b(k)} \tilde{e}_{n-1}^{b*}(k) = \\
&= \lambda^{0.5} q_n(k-1) c_n^b(k) - s_n^b(k) \tilde{e}_{n-1}^{b*}(k).
\end{aligned} \quad (12.167)$$

Выражения (12.164) и (12.167) можно представить и как результат приведения с помощью вращения Гивенса к треугольному виду последовательности матриц

$$\begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1) & \lambda^{0.5} q_n^*(k-1) \\ -\tilde{e}_{n-1}^b(k) & \tilde{e}_{n-1}(k) \end{bmatrix}$$

как

$$\begin{aligned} & \begin{bmatrix} c_n^b(k) & -s_n^{b*}(k) \\ s_n^b(k) & c_n^b(k) \end{bmatrix} \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1) & \lambda^{0.5} q_n^*(k-1) \\ -\tilde{e}_{n-1}^b(k) & \tilde{e}_{n-1}(k) \end{bmatrix} = \\ & = \begin{bmatrix} \tilde{E}_{n-1,\min}^b(k) & q_n^*(k) \\ 0 & \tilde{e}_n(k) \end{bmatrix}. \end{aligned} \quad (12.168)$$

Справедливость этого уравнения легко устанавливается путем перемножения матриц в его левой части и использования (12.139) и (12.140).

Таблица 12.8. Лестничный RLS-алгоритм с использованием вращений Гивенса с операциями извлечения квадратного корня

Вычисления	Ссылки
Initialization: $\tilde{E}_{n-1,\min}^f(0) = \delta$, $\tilde{E}_{n-1,\min}^b(0) = \delta\sqrt{\lambda^{-(n-1)}}$, $\tilde{e}_{n-1}^b(0) = 0$, $c_n^b(0) = 0$, $s_n^b(0) = 0$, $q_n^f(0) = 0$, $q_n^b(0) = 0$, $q_n(0) = 0$, $n = 1, 2, \dots, N$	(12.8.0)
For $k = 1, 2, \dots, K$	
$\tilde{e}_0^f(k) = x(k)$, $\tilde{e}_0^b(k) = x(k)$, $\tilde{e}_0(k) = d(k)$, $\tilde{\phi}_0(k) = 1$	(12.8.1)
For $n = 1, 2, \dots, N$	
$\tilde{E}_{n-1,\min}^f(k) = \sqrt{\lambda \left[\tilde{E}_{n-1,\min}^f(k-1) \right]^2 + \tilde{e}_{n-1}^f(k) \tilde{e}_{n-1}^{f*}(k)}$	(12.8.2)
$c_n^f(k) = \lambda^{0.5} \tilde{E}_{n-1,\min}^f(k-1) / \tilde{E}_{n-1,\min}^f(k)$	(12.3.3)
$s_n^f(k) = \tilde{e}_{n-1}^f(k) / \tilde{E}_{n-1,\min}^f(k)$	(12.8.4)
$\tilde{e}_n^f(k) = \tilde{e}_{n-1}^f(k) c_n^b(k-1) + \lambda^{0.5} q_n^{f*}(k-1) s_n^b(k-1)$	(12.8.5)
$q_n^f(k) = \lambda^{0.5} q_n^f(k-1) c_n^b(k-1) - s_n^b(k-1) \tilde{e}_{n-1}^{f*}(k)$	(12.8.6)
$\tilde{E}_{n-1,\min}^b(k) = \sqrt{\lambda \left[\tilde{E}_{n-1,\min}^b(k-1) \right]^2 + \tilde{e}_{n-1}^b(k) \tilde{e}_{n-1}^{b*}(k)}$	(12.8.7)
$c_n^b(k) = \lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1) / \tilde{E}_{n-1,\min}^b(k)$	(12.8.8)
$s_n^b(k) = \tilde{e}_{n-1}^b(k) / \tilde{E}_{n-1,\min}^b(k)$	(12.8.9)
$\tilde{e}_n^b(k) = \tilde{e}_{n-1}^b(k-1) c_n^f(k) + \lambda^{0.5} q_n^{b*}(k-1) s_n^f(k)$	(12.8.10)
$q_n^b(k) = \lambda^{0.5} q_n^b(k-1) c_n^f(k) - s_n^f(k) \tilde{e}_{n-1}^{b*}(k-1)$	(12.8.11)
$\tilde{e}_n(k) = \tilde{e}_{n-1}(k) c_n^b(k) + \lambda^{0.5} q_n^*(k-1) s_n^b(k)$	(12.8.12)
$q_n(k) = \lambda^{0.5} q_n(k-1) c_n^b(k) - s_n^b(k) \tilde{e}_{n-1}^*(k)$	(12.8.13)
$\tilde{\phi}_n(k) = c_n^b(k) \tilde{\phi}_{n-1}(k)$	(12.8.14)
End for n	
$\alpha_N(k) = \tilde{e}_N(k) / \tilde{\phi}_N(k)$	(12.8.15)
End for k	

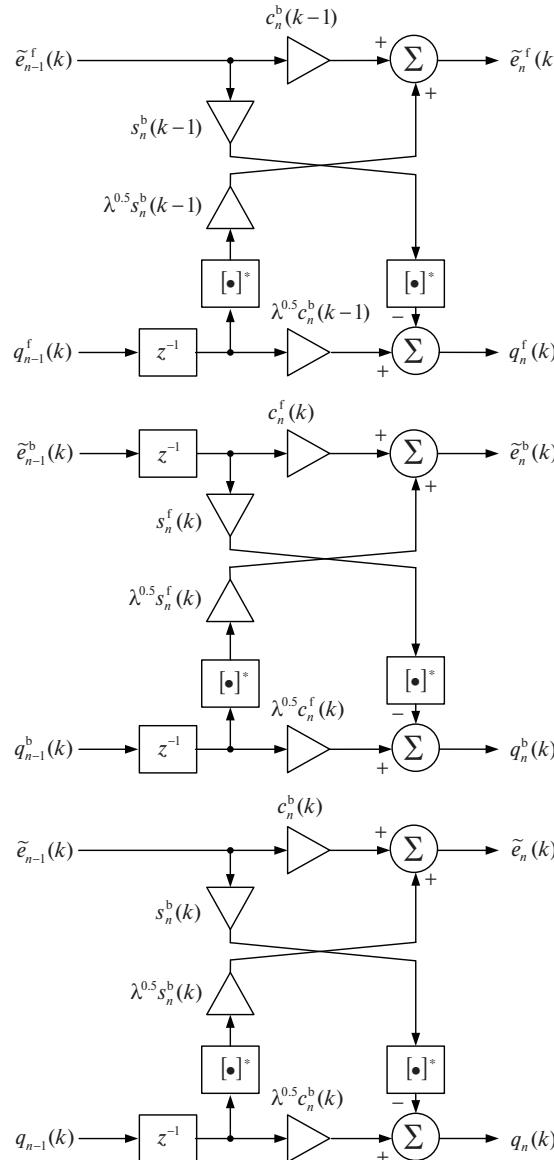


Рис. 12.5. Вычисление нормализованных апостериорных ошибок в лестничном RLS-алгоритме на основе QR-разложения

Действительно,

$$\begin{aligned}
 c_n^b(k) \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) - s_n^{b*}(k) [-\tilde{e}_{n-1}^b(k)] &= \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k)} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) - \\
 - \frac{\tilde{e}_{n-1}^{b*}(k)}{\tilde{E}_{n-1, \min}^b(k)} [-\tilde{e}_{n-1}^b(k)] &= \frac{\lambda [\tilde{E}_{n-1, \min}^b(k-1)]^2}{\tilde{E}_{n-1, \min}^b(k)} + \frac{\tilde{e}_{n-1}^{b*}(k) \tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1, \min}^b(k)} = \tilde{E}_{n-1, \min}^b(k),
 \end{aligned} \tag{12.169}$$

откуда следует уравнение (12.135).

Аналогично доказывается, что

$$\begin{aligned}
 s_n^b(k) \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) + c_n^b(k) [-\tilde{e}_{n-1}^b(k)] &= \\
 = \frac{\tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1, \min}^b(k)} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) + & \\
 + \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1)}{\tilde{E}_{n-1, \min}^b(k)} [-\tilde{e}_{n-1}^b(k)] &= 0.
 \end{aligned} \tag{12.170}$$

Из сравнения уравнений (12.136) и (12.139) также следует, что

$$\begin{aligned}
 \tilde{\varphi}_n(k) = \varphi_n^{0.5}(k) &= \frac{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) \varphi_{n-1}^{0.5}(k)}{\tilde{E}_{n-1, \min}^b(k)} = \\
 = c_n^b(k) \varphi_{n-1}^{0.5}(k) &= c_n^b(k) \tilde{\varphi}_{n-1}(k),
 \end{aligned} \tag{12.171}$$

а из уравнения (12.164) — что

$$\alpha_n(k) = \tilde{e}_n(k) / \varphi_n^{0.5}(k). \tag{12.172}$$

Последовательность полученных в настоящем разделе уравнений, а именно (12.134), (12.153), (12.154), (12.137), (12.144), (12.135), (12.139), (12.140), (12.151), (12.158), (12.164), (12.167), (12.171), а также уравнение (12.172) при $n=N$ составляют лестничный RLS-алгоритм с использованием вращений Гивенса, представленный в табл. 12.8.

Структура вычисления ошибок в соответствии с табл. 12.8 приведена на рис. 12.5, а структура соответствующего адаптивного фильтра — на рис. 12.6.

Инициализация переменных $q_n^f(0)$, $q_n^b(0)$ и $q_n(0)$ как показано в уравнении (12.8.0) следует из определения этих переменных, см. уравнения (12.141), (12.155), (12.165), и инициализации коэффициентов $\gamma_n(0)$, $\Gamma_n^f(0)$ и $\Gamma_n^b(0)$, см. табл. 12.4.

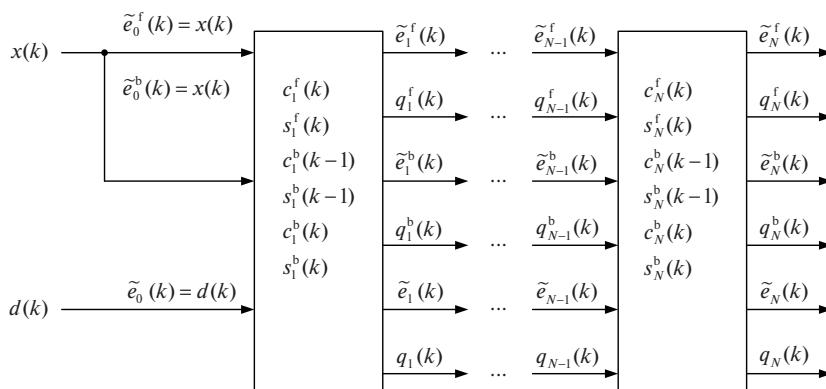


Рис. 12.6. Адаптивный фильтр на основе QR-разложения с операциями извлечения квадратного корня

12.9. Алгоритм на основе QR-разложения без операций извлечения квадратного корня

Известно (см. главу 7), что в RLS-алгоритмах, использующих вращения Гивенса, можно избежать операций извлечения квадратного корня путем масштабирования переменных. Лестничный алгоритм, не содержащий операций квадратного корня [146], может быть получен из лестничного RLS-алгоритма на основе априорных ошибок с обратными связями (табл. 12.5). Для вычисления коэффициентов $\Gamma_n^f(k)$, $\Gamma_n^b(k)$ и $\gamma_n(k)$ с помощью вращений Гивенса уравнения (12.87)–(12.89) необходимо преобразовать аналогично (12.109), (12.156) и (12.116).

Для этого сначала из уравнения (12.2.2) определим

$$\begin{aligned}\lambda E_{n-1,\min}^f(k-1) &= E_{n-1,\min}^f(k) - \varphi_{n-1}(k-1) \alpha_{n-1}^f(k) \alpha_{n-1}^{f*}(k) = \\ &= E_{n-1,\min}^f(k) \left[1 - \frac{\varphi_{n-1}(k-1) \alpha_{n-1}^f(k) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^f(k)} \right],\end{aligned}$$

откуда

$$\frac{\lambda E_{n-1,\min}^f(k-1)}{E_{n-1,\min}^f(k)} = 1 - \frac{\varphi_{n-1}(k-1) \alpha_{n-1}^f(k) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^f(k)}, \quad (12.173)$$

а затем из уравнения (12.2.3) определим

$$\begin{aligned}\lambda E_{n-1,\min}^b(k-1) &= E_{n-1,\min}^b(k) - \varphi_{n-1}(k) \alpha_{n-1}^b(k) \alpha_{n-1}^{b*}(k) = \\ &= E_{n-1,\min}^b(k) \left[1 - \frac{\varphi_{n-1}(k) \alpha_{n-1}^b(k) \alpha_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k)} \right],\end{aligned} \quad (12.174)$$

откуда

$$\frac{\lambda E_{n-1,\min}^b(k-1)}{E_{n-1,\min}^b(k)} = 1 - \frac{\varphi_{n-1}(k) \alpha_{n-1}^b(k) \alpha_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k)}. \quad (12.175)$$

Затем, используя уравнения (12.28) и (12.175), из (12.87) получим, что

$$\begin{aligned}\Gamma_n^f(k) &= \Gamma_n^f(k-1) - \left[E_{n-1,\min}^b(k-1) \right]^{-1} \alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_n^{f*}(k) = \\ &= \Gamma_n^f(k-1) - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1)}{E_{n-1,\min}^b(k-1)} \left[\alpha_{n-1}^{f*}(k) + \Gamma_n^f(k-1) \alpha_{n-1}^{b*}(k-1) \right] = \\ &= \Gamma_n^f(k-1) - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1)} - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \Gamma_n^f(k-1) \alpha_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1)} = \\ &= \Gamma_n^f(k-1) \left[1 - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1)}{E_{n-1,\min}^b(k-1)} \right] - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1)} = \\ &= \Gamma_n^f(k-1) \frac{\lambda E_{n-1,\min}^b(k-2)}{E_{n-1,\min}^b(k-1)} - \frac{\alpha_{n-1}^b(k-1) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^b(k-1)} = \\ &= \Gamma_n^f(k-1) \bar{c}_n^b(k-1) - \bar{s}_n^b(k-1) \alpha_{n-1}^{f*}(k),\end{aligned} \quad (12.176)$$

где

$$\bar{c}_n^b(k) = \frac{\lambda E_{n-1,\min}^b(k-1)}{E_{n-1,\min}^b(k)}, \quad (12.177)$$

$$\bar{s}_n^b(k) = \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k)}{E_{n-1,\min}^b(k)}. \quad (12.178)$$

Далее, используя (12.30) и (12.173), из (12.88) получим, что

$$\begin{aligned} \Gamma_n^b(k) &= \Gamma_n^b(k-1) - \left[E_{n-1,\min}^f(k) \right]^{-1} \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{b*}(k) = \\ &= \Gamma_n^b(k-1) - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1)}{E_{n-1,\min}^f(k)} \left[\alpha_{n-1}^{b*}(k-1) + \Gamma_n^b(k-1) \alpha_{n-1}^{f*}(k) \right] = \\ &= \Gamma_n^b(k-1) - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1)}{E_{n-1,\min}^f(k)} - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \Gamma_n^b(k-1) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^f(k)} = \\ &= \Gamma_n^b(k-1) \left[1 - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{f*}(k)}{E_{n-1,\min}^f(k)} \right] - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1)}{E_{n-1,\min}^f(k)} = \\ &= \Gamma_n^b(k-1) \frac{\lambda E_{n-1,\min}^f(k-1)}{E_{n-1,\min}^f(k)} - \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1) \alpha_{n-1}^{b*}(k-1)}{E_{n-1,\min}^f(k)} = \\ &= \Gamma_n^b(k-1) \bar{c}_n^f(k) - \bar{s}_n^f(k) \alpha_{n-1}^{b*}(k-1), \end{aligned} \quad (12.179)$$

где

$$\bar{c}_n^f(k) = \frac{\lambda E_{n-1,\min}^f(k-1)}{E_{n-1,\min}^f(k)}, \quad (12.180)$$

$$\bar{s}_n^f(k) = \frac{\alpha_{n-1}^f(k) \varphi_{n-1}(k-1)}{E_{n-1,\min}^f(k)}. \quad (12.181)$$

И, наконец, используя уравнения (12.54) и (12.175), из (12.89) получим, что

$$\begin{aligned} \gamma_n(k) &= \gamma_n(k-1) - \left[E_{n-1,\min}^b(k) \right]^{-1} \alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k) = \\ &= \gamma_n(k-1) - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k)}{E_{n-1,\min}^b(k)} \left[\alpha_{n-1}^*(k) + \gamma_n(k-1) \alpha_{n-1}^{b*}(k) \right] = \\ &= \gamma_n(k-1) - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k)}{E_{n-1,\min}^b(k)} - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k) \gamma_n(k-1) \alpha_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k)} = \\ &= \gamma_n(k-1) \left[1 - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^{b*}(k)}{E_{n-1,\min}^b(k)} \right] - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k)}{E_{n-1,\min}^b(k)} = \\ &= \gamma_n(k-1) \frac{\lambda E_{n-1,\min}^b(k-1)}{E_{n-1,\min}^b(k)} - \frac{\alpha_{n-1}^b(k) \varphi_{n-1}(k) \alpha_{n-1}^*(k)}{E_{n-1,\min}^b(k)} = \\ &= \gamma_n(k-1) \bar{c}_n^b(k) - \bar{s}_n^b(k) \alpha_{n-1}^*(k). \end{aligned} \quad (12.182)$$

Из уравнения (12.97) также следует, что

$$\frac{\lambda E_{n-1,\min}^b(k-1)}{E_{n-1,\min}^b(k)} = \frac{\varphi_n(k)}{\varphi_{n-1}(k)} = \bar{c}_n^b(k), \quad (12.183)$$

откуда

$$\varphi_n(k) = \bar{c}_n^b(k) \varphi_{n-1}(k). \quad (12.184)$$

Таким образом, если уравнения алгоритма, приведенного в табл. 12.5, модифицировать уравнениями настоящего раздела, то получится алгоритм на основе использования вращений Гивенса без операций извлечения квадратного корня, приведенный в табл. 12.9.

Таблица 12.9. Лестничный RLS-алгоритм с использованием вращений Гивенса без операций извлечения квадратного корня

Вычисления	Ссылки
Initialization: $E_{n-1,\min}^f(0) = \delta^2$, $E_{n-1,\min}^b(0) = \delta^2 \lambda^{-(n-1)}$, $\bar{c}_n^b(k) = 0$,	(12.9.0)
$\bar{s}_n^b(k) = 0$, $\alpha_{n-1}^b(0) = 0$, $\varphi_{n-1}(0) = 1$, $\Gamma_n^f(0) = 0$, $\Gamma_n^b(0) = 0$, $\gamma_n(0) = 0$, $n = 1, 2, \dots, N$	
For $k = 1, 2, \dots, K$	
$\alpha_0^f(k) = x(k)$, $\alpha_0^b(k) = x(k)$, $\alpha_0(k) = d(k)$, $\varphi_0(k) = 1$	(12.9.1)
For $n = 1, 2, \dots, N$	
$E_{n-1,\min}^f(k) = \lambda E_{n-1,\min}^f(k-1) + \varphi_{n-1}(k-1) \alpha_{n-1}^f(k) \alpha_{n-1}^{f*}(k)$	(12.9.2)
$\bar{c}_n^f(k) = \lambda E_{n-1,\min}^f(k-1) / E_{n-1,\min}^f(k)$	(12.9.3)
$\bar{s}_n^f(k) = \alpha_{n-1}^f(k) \varphi_{n-1}(k-1) / E_{n-1,\min}^f(k)$	(12.9.4)
$\alpha_n^f(k) = \alpha_{n-1}^f(k) + \Gamma_n^{f*}(k-1) \alpha_{n-1}^b(k-1)$	(12.9.5)
$\Gamma_n^f(k) = \Gamma_n^f(k-1) \bar{c}_n^b(k-1) - \bar{s}_n^b(k-1) \alpha_{n-1}^{f*}(k)$	(12.9.6)
$E_{n-1,\min}^b(k) = \lambda E_{n-1,\min}^b(k-1) + \varphi_{n-1}(k) \alpha_{n-1}^b(k) \alpha_{n-1}^{b*}(k)$	(12.9.7)
$\bar{c}_n^b(k) = \lambda E_{n-1,\min}^b(k-1) / E_{n-1,\min}^b(k)$	(12.9.8)
$\bar{s}_n^b(k) = \alpha_{n-1}^b(k) \varphi_{n-1}(k) / E_{n-1,\min}^b(k)$	(12.9.9)
$\alpha_n^b(k) = \alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k)$	(12.9.10)
$\Gamma_n^b(k) = \Gamma_n^b(k-1) \bar{c}_n^f(k) - \bar{s}_n^f(k) \alpha_{n-1}^{b*}(k-1)$	(12.9.11)
$\alpha_n(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k)$	(12.9.12)
$\gamma_n(k) = \gamma_n(k-1) \bar{c}_n^b(k) - \bar{s}_n^b(k) \alpha_{n-1}^{b*}(k)$	(12.9.13)
$\alpha_n^b(k) = \alpha_{n-1}^b(k-1) + \Gamma_n^{b*}(k-1) \alpha_{n-1}^f(k)$	(12.9.14)
$\alpha_n(k) = \alpha_{n-1}(k) + \gamma_n^*(k-1) \alpha_{n-1}^b(k)$	(12.9.15)
$\varphi_n(k) = \bar{c}_n^b(k) \varphi_{n-1}(k)$	(12.9.16)
End for n	
End for k	

Структура адаптивного фильтра на основе алгоритма (см. табл. 12.9) совпадает со структурой (см. рис. 12.1). Отличие от алгоритма из табл. 12.5 заключается лишь в способе вычисления коэффициентов $\Gamma_n^f(k)$, $\Gamma_n^b(k)$ и $\gamma_n(k)$, которые вычисляются с помощью элементов матриц Гивенса.

12.10. Алгоритм на основе QR-разложения с операциями извлечения квадратного корня в пространстве состояний

Возможно и другое представление лестничных алгоритмов, использующих вращения Гивенса, с помощью пространства состояний [20, 22, 30]. В таком алгоритме, приведенном в табл. 12.10, фактически используются те же самые вычисления, что и в алгоритме из табл. 12.8, за исключением того, что вместо уравнений (12.8.2) и (12.8.7) используются уравнения состояний для вычисления энергий ошибок.

Таблица 12.10. Лестничный RLS-алгоритм в пространстве состояний с использованием вращений Гивенса с операциями извлечения квадратного корня

Вычисления	Ссылки
Initialization: $\tilde{E}_{n-1,\min}^f(0) = \delta$, $\tilde{E}_{n-1,\min}^b(0) = \delta\sqrt{\lambda^{-(n-1)}}$, $\tilde{E}_{n-1,\min}^b(-1) = \delta\sqrt{\lambda^{-(n-1)}}$, $\tilde{e}_{n-1}^b(0) = 0$, $c_n^b(0) = 0$, $s_n^b(0) = 0$, $q_n^f(0) = 0$, $q_n^b(0) = 0$, $q_n(0) = 0$, $n = 1, 2, \dots, N$	(12.10.0)
For $k = 1, 2, \dots, K$	
$\tilde{e}_0^f(k) = x(k)$, $\tilde{e}_0^b(k) = x(k)$, $\tilde{e}_0(k) = d(k)$, $\tilde{\varphi}_0(k) = 1$	(12.10.1)
For $n = 1, 2, \dots, N$	
$c_n^b(k-1) = \left\{ 1 + \left[\frac{\tilde{e}_{n-1}^b(k-1)}{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2)} \right] \left[\frac{\tilde{e}_{n-1}^b(k-1)}{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2)} \right]^* \right\}^{-0.5}$	(12.10.2)
$s_n^b(k-1) = c_n^b(k-1) \left[\frac{\tilde{e}_{n-1}^b(k-1)}{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2)} \right]$	(12.10.3)
$\Theta_n^b(k-1) = \begin{bmatrix} c_n^b(k-1) & -s_n^{b*}(k-1) \\ s_n^b(k-1) & c_n^b(k-1) \end{bmatrix}$	(12.10.4)
$\Theta_n^b(k-1) \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-2) & \lambda^{0.5} q_n^{f*}(k-1) \\ -\tilde{e}_{n-1}^b(k-1) & \tilde{e}_{n-1}^f(k) \end{bmatrix} = \begin{bmatrix} \tilde{E}_{n-1,\min}^b(k-1) & q_n^{f*}(k) \\ 0 & \tilde{e}_n^f(k) \end{bmatrix}$	(12.10.5)
$c_n^b(k) = \left\{ 1 + \left[\frac{\tilde{e}_{n-1}^b(k)}{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1)} \right] \left[\frac{\tilde{e}_{n-1}^b(k)}{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1)} \right]^* \right\}^{-0.5}$	(12.10.6)

Таблица 12.10. (окончание)

Вычисления	Ссылки
$s_n^b(k) = c_n^b(k) \left[\frac{\tilde{e}_{n-1}^b(k)}{\lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1)} \right]$	(12.10.7)
$\Theta_n^b(k) = \begin{bmatrix} c_n^b(k) & -s_n^b(k) \\ s_n^{b*}(k) & c_n^b(k) \end{bmatrix}$	(12.10.8)
$\Theta_n^b(k) \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^b(k-1) & \lambda^{0.5} q_n^*(k-1) & \tilde{\varphi}_{n-1}(k) \\ -\tilde{e}_{n-1}^b(k) & \tilde{e}_{n-1}(k) & 0 \end{bmatrix} =$	(12.10.9)
$= \begin{bmatrix} \tilde{E}_{n-1}^b(k) & q_n^*(k) & \tilde{\varphi}_n(k) \\ 0 & \tilde{e}_{n-1}(k) & e_{n-1}^b(k) / \tilde{E}_{n-1, \min}^b(k) \end{bmatrix}$	
$c_n^f(k) = \left\{ 1 + \left[\frac{\tilde{e}_{n-1}^f(k)}{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)} \right] \left[\frac{\tilde{e}_{n-1}^f(k)}{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)} \right]^* \right\}^{-0.5}$	(12.10.10)
$s_n^f(k) = c_n^f(k) \left[\frac{\tilde{e}_{n-1}^f(k)}{\lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1)} \right]$	(12.10.11)
$\Theta_n^f(k) = \begin{bmatrix} c_n^f(k) & -s_n^{f*}(k) \\ s_n^{f*}(k) & c_n^f(k) \end{bmatrix}$	(12.10.12)
$\Theta_n^f(k) \begin{bmatrix} \lambda^{0.5} \tilde{E}_{n-1, \min}^f(k-1) & \lambda^{0.5} q_n^{b*}(k-1) \\ -\tilde{e}_{n-1}^f(k) & \tilde{e}_{n-1}^b(k-1) \end{bmatrix} \Theta_n^f(k) = \begin{bmatrix} \tilde{E}_{n-1, \min}^f(k) & q_n^{b*}(k) \\ 0 & \tilde{e}_n^b(k) \end{bmatrix}$	(12.10.13)
End for n	
$e_N(k) = \tilde{\varphi}_N(k) \tilde{e}_N(k)$	(12.10.14)
$\alpha_N(k) = \tilde{e}_N(k) / \tilde{\varphi}_N(k)$	(12.10.15)
End for k	

В алгоритме, приведенном в табл. 12.10, выражения для синусов и косинусов углов вращения, см. уравнения (12.10.2), (12.10.3), (12.10.6), (12.10.7), (12.10.10) и (12.10.11), можно получить из уравнений (12.139), (12.140), (12.153), (12.154), (12.134) и (12.135) следующим образом.

Из уравнения (12.134) следует, что

$$\begin{aligned} [\tilde{E}_{n, \min}^f(k)]^2 &= \lambda [\tilde{E}_{n, \min}^f(k-1)]^2 + \tilde{e}_n^f(k) \tilde{e}_n^{f*}(k) = \\ &= \lambda [\tilde{E}_{n, \min}^f(k-1)]^2 \left[1 + \frac{\tilde{e}_n^f(k) \tilde{e}_n^{f*}(k)}{\lambda [\tilde{E}_{n, \min}^f(k-1)]^2} \right], \end{aligned} \quad (12.185)$$

откуда

$$\frac{\lambda [\tilde{E}_{n, \min}^f(k-1)]^2}{[\tilde{E}_{n, \min}^f(k)]^2} = \left[1 + \frac{\tilde{e}_n^f(k) \tilde{e}_n^{f*}(k)}{\lambda [\tilde{E}_{n, \min}^f(k-1)]^2} \right]^{-1}. \quad (12.186)$$

Из уравнения (12.135) следует, что

$$\begin{aligned} \left[\tilde{E}_{n,\min}^b(k) \right]^2 &= \lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2 + \tilde{e}_n^b(k) \tilde{e}_n^{b*}(k) = \\ &= \lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2 \left[1 + \frac{\tilde{e}_n^b(k) \tilde{e}_n^{b*}(k)}{\lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2} \right], \end{aligned} \quad (12.187)$$

откуда

$$\frac{\lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2}{\left[\tilde{E}_{n,\min}^b(k) \right]^2} = \left[1 + \frac{\tilde{e}_n^b(k) \tilde{e}_n^{b*}(k)}{\lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2} \right]^{-1}. \quad (12.188)$$

Тогда из уравнений (12.139) и (12.140) следует, что

$$c_n^b(k) = \frac{\lambda^{0.5} \tilde{E}_{n-1,\min}^b(k-1)}{\tilde{E}_{n-1,\min}^b(k)} = \left[1 + \frac{\tilde{e}_n^b(k) \tilde{e}_n^{b*}(k)}{\lambda \left[\tilde{E}_{n,\min}^b(k-1) \right]^2} \right]^{-0.5}, \quad (12.189)$$

$$\begin{aligned} s_n^b(k) &= \frac{\tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1,\min}^b(k)} = \frac{\lambda^{0.5} \tilde{E}_{n,\min}^b(k-1)}{\lambda^{0.5} \tilde{E}_{n,\min}^b(k-1)} \cdot \frac{\tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1,\min}^b(k)} = \\ &= \frac{\lambda^{0.5} \tilde{E}_{n,\min}^b(k-1)}{\tilde{E}_{n-1,\min}^b(k)} \cdot \frac{\tilde{e}_{n-1}^b(k)}{\lambda^{0.5} \tilde{E}_{n,\min}^b(k-1)} = c_n^b(k) \frac{\tilde{e}_{n-1}^b(k)}{\lambda^{0.5} \tilde{E}_{n,\min}^b(k-1)}, \end{aligned} \quad (12.190)$$

а из уравнений (12.153) и (12.154) — что

$$c_n^f(k) = \frac{\lambda^{0.5} \tilde{E}_{n-1,\min}^f(k-1)}{\tilde{E}_{n-1,\min}^f(k)} = \left[1 + \frac{\tilde{e}_n^f(k) \tilde{e}_n^{f*}(k)}{\lambda \left[\tilde{E}_{n,\min}^f(k-1) \right]^2} \right]^{-0.5}, \quad (12.191)$$

$$\begin{aligned} s_n^f(k) &= \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1,\min}^f(k)} = \frac{\lambda^{0.5} \tilde{E}_{n,\min}^f(k-1)}{\lambda^{0.5} \tilde{E}_{n,\min}^f(k-1)} \cdot \frac{\tilde{e}_{n-1}^f(k)}{\tilde{E}_{n-1,\min}^f(k)} = \\ &= \frac{\lambda^{0.5} \tilde{E}_{n,\min}^f(k-1)}{\tilde{E}_{n-1,\min}^f(k)} \cdot \frac{\tilde{e}_{n-1}^f(k)}{\lambda^{0.5} \tilde{E}_{n,\min}^f(k-1)} = c_n^f(k) \frac{\tilde{e}_{n-1}^f(k)}{\lambda^{0.5} \tilde{E}_{n,\min}^f(k-1)}. \end{aligned} \quad (12.192)$$

Выражения (12.10.4) и (12.10.5) представляют собой уравнение (12.145), выражения (12.10.12) и (12.10.13) представляют собой уравнение (12.159), а уравнение (12.10.9) представляет собой уравнения (12.168) и (12.171). Элемент $e_{n-1}^b(k) / \tilde{E}_{n-1,\min}^b(k)$ в (12.10.9) образуется с учетом уравнений (12.140) и (12.151) как

$$s_n^b(k) \tilde{\Phi}_{n-1}(k) = \frac{\tilde{e}_{n-1}^b(k)}{\tilde{E}_{n-1,\min}^b(k)} \tilde{\Phi}_{n-1}(k) = \frac{e_{n-1}^b(k)}{\tilde{E}_{n-1,\min}^b(k)}, \quad (12.193)$$

а выражения (12.10.14) и (12.10.15) следуют из уравнений (12.164) и (12.172).

12.11. Выводы по главе

Рассмотренные в настоящей главе основные разновидности лестничных RLS-алгоритмов адаптивной фильтрации являются результатом тождественных преобразований, а потому они математически эквивалентны друг другу и другим RLS-алгоритмам для одноканальных адаптивных фильтров с комплексными ВК, в которых оценка корреляционной матрицы производится на возрастающем окне с экспоненциальным взвешиванием отсчетов обрабатываемых сигналов.

Сравнительные оценки арифметической сложности рассмотренных алгоритмов приведены в табл. 12.11. Данные алгоритмы принадлежат к классу быстрых RLS-алгоритмов, поскольку их арифметическая сложность равна $O(N)$. В целом, вычислительная сложность лестничных RLS-алгоритмов больше, чем быстрых RLS-алгоритмов из главы 11. Однако, как уже отмечалось, лестничные алгоритмы являются более устойчивыми, что дает им право на существование. Различие между разновидностями лестничных алгоритмов проявляется при ограниченной разрядности вычислений.

Таблица 12.11. Вычислительная сложность лестничных RLS-алгоритмов

№	Алгоритм	Сложения	Умножения	Деления	Квадратные корни
1	Табл. 12.1	$10N$	$16N$	$2N$	—
2	Табл. 12.2	$8N$	$17N$	$2N$	—
3	Табл. 12.3	$8N$	$17N$	$3N+1$	—
4	Табл. 12.4	$11N$	$16N$	$2N$	—
5	Табл. 12.5	$9N$	$17N$	$2N$	—
6	Табл. 12.6	$9N$	$25N$	$5N$	—
7	Табл. 12.7	$9N$	$18N$	$3N$	$5N$
8	Табл. 12.8	$8N$	$32N$	$2N$	$2N$
9	Табл. 12.9	$8N$	$20N$	$2N$	—
10	Табл. 12.10	$12N$	$30N$	$3N$	$3N$

ГЛАВА 13

МНОГОКАНАЛЬНЫЕ RLS-АЛГОРИТМЫ

В настоящей главе рассматриваются два способа построения многоканальных версий быстрых RLS-алгоритмов, рассмотренных в главах 6–11, для адаптивных фильтров с неодинаковым числом весовых коэффициентов в каналах. Эти способы основаны на использовании перестановочных матриц.

13.1. Введение

В главе 1 была приведена структура многоканального адаптивного КИХ-фильтра (см. рис. 1.3) с неодинаковым числом весовых коэффициентов, а также перечислены примеры приложений этих фильтров. В более общем случае такой фильтр может быть также элементом адаптивной системы со многими входами и многими выходами (Multi Input – Multi Output, MIMO). Эта система представляет собой набор фильтров (см. рис. 1.3), обрабатывающих одни и те же входные сигналы $x_m(k)$, где $m=1, 2, \dots, M$, а M – число каналов (входов многоканальной системы), и различные требуемые сигналы $d_p(k)$, где $p=1, 2, \dots, P$, а P – число выходов многоканальной системы. Поскольку сигналы $d_p(k)$, как правило, являются независимыми, то алгоритмы адаптивной фильтрации в MIMO-системах можно рассматривать как набор из P независимых алгоритмов, в которых общей является часть, связанная с вычислением вектора коэффициентов Калмана $\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)$, где $\mathbf{R}_N(k)$ – корреляционная матрица многоканального КИХ-фильтра, определяемая согласно уравнению (3.14) как

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \begin{bmatrix} \mathbf{R}_{N_1}(k) & \mathbf{R}_{N_1N_2}(k) & \cdots & \mathbf{R}_{N_1N_M}(k) \\ \mathbf{R}_{N_1N_2}^H(k) & \mathbf{R}_{N_2}(k) & \cdots & \mathbf{R}_{N_2N_M}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{N_1N_M}^H(k) & \mathbf{R}_{N_2N_{M-1}}^H(k) & \cdots & \mathbf{R}_{N_M}(k) \end{bmatrix}. \quad (13.1)$$

В уравнении (13.1)

$$\mathbf{x}_N(k) = \left[\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \quad (13.2)$$

– вектор входных сигналов многоканального фильтра,

$$\mathbf{x}_{N_m}(k) = \left[x_m(k), x_m(k-1), \dots, x_m(k-N_m+1) \right]^T \quad (13.3)$$

— вектор входных сигналов m -го канала многоканального фильтра, N_m — число ВК в m -м канале фильтра, $N = \sum_{m=1}^M N_m$ — полное (суммарное) число ВК многоканального фильтра. Такая декомпозиция адаптивной MIMO-системы возможна только в случае, если все сигналы $d_p(k)$ формируются на базе линейных моделей, аналогичных показанным на рис. 1.3, с одинаковыми параметрами M и N_m .

Переход от RLS-алгоритмов, перечисленных в табл. 6.1, табл. 7.1, табл. 8.1, табл. 9.1—9.3, табл. 10.1—10.3, с вычислительной сложностью $O(N^2)$ арифметических операций для одноканальных фильтров со структурой, показанной на рис. 1.2, к алгоритмам для многоканальных фильтров со структурой, показанной на рис. 1.3, осуществляется путем тривиальной замены вектора входного сигнала $\mathbf{x}_N(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T$ на вектор входных сигналов (13.2). Инициализация входной матрицы $\mathbf{R}_N(0)$ или ее преобразованных значений в RLS-алгоритмах на основе QR-разложения осуществляется в соответствии со структурой матрицы (13.1) как

$$\mathbf{R}_N(0) = \begin{bmatrix} \mathbf{R}_{N_1}(0) & 0 & \cdots & 0 \\ 0 & \mathbf{R}_{N_2}(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{R}_{N_M}(0) \end{bmatrix} = \delta^2 \begin{bmatrix} \mathbf{\Lambda}_{N_1}^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{\Lambda}_{N_2}^{-1}(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Lambda}_{N_M}^{-1}(0) \end{bmatrix},$$

где $\mathbf{\Lambda}_{N_m}^{-1} = \text{diag}\{1, \lambda^{-1}, \dots, \lambda^{-(N_m-1)}\}$, см. также раздел 14.8. Остальные вычисления в табл. 6.1, табл. 7.1, табл. 8.1, табл. 9.1—9.3, табл. 10.1—10.3 остаются неизменными.

В быстрых RLS-алгоритмах (см. главу 11) такой переход невозможен в силу того, что вектор входного сигнала многоканального фильтра (13.2) из-за своей структуры не обладает свойством инвариантности к сдвигу (11.1). В то же время при использовании технологии перестановочных матриц [66, 147—151] могут быть получены и многоканальные версии алгоритмов, рассмотренных в главе 11. В настоящей главе будут рассмотрены два способа построения многоканальных быстрых RLS-алгоритмов.

13.2. B-SUSD линейное предсказание

По определению [150], апостериорная ошибка линейного предсказания вперед входного сигнала m -го канала адаптивного фильтра вычисляется как

$$e_{N,m}^f(k) = x_m(k) - \sum_{m=1}^M \sum_{n=1}^{N_m} h_{n,m}^{f*}(k) x_m(k-n) = x_m(k) - \mathbf{h}_{N,m}^{fH}(k) \mathbf{x}_N(k-1), \quad (13.4)$$

где

$$\mathbf{x}_N(k-1) = [\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_M}^T(k-1)]^T \quad (13.5)$$

— вектор входного сигнала фильтра линейного предсказания с ВК



$$\mathbf{h}_{N,m}^f(k) = \left[\mathbf{h}_{N_1,m}^{fT}(k), \mathbf{h}_{N_2,m}^{fT}(k), \dots, \mathbf{h}_{N_M,m}^{fT}(k) \right]^T \quad (13.6)$$

и

$$\mathbf{h}_{N_m,m}^f(k) = \left[h_{1,m}^f(k), h_{2,m}^f(k), \dots, h_{N_m,m}^f(k) \right]^T.$$

Аналогично апостериорная ошибка линейного предсказания назад входного сигнала m -го канала адаптивного фильтра определяется как

$$e_{N,m}^b(k) = x_m(k - N_m) - \sum_{m=1}^M \sum_{n=1}^{N_m} h_{n,m}^{b*}(k) x_n(k) = x_m(k - N_m) - \mathbf{h}_{N,m}^{bH}(k) \mathbf{x}_N(k), \quad (13.7)$$

где

$$\mathbf{x}_N(k) = \left[\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \quad (13.8)$$

— вектор входного сигнала фильтра линейного предсказания с ВК

$$\mathbf{h}_{N,m}^b(k) = \left[\mathbf{h}_{N_1,m}^{bT}(k), \mathbf{h}_{N_2,m}^{bT}(k), \dots, \mathbf{h}_{N_M,m}^{bT}(k) \right]^T \quad (13.9)$$

и

$$\mathbf{h}_{N_m,m}^b(k) = \left[h_{1,m}^b(k), h_{2,m}^b(k), \dots, h_{N_m,m}^b(k) \right]^T.$$

Каждый из $m = 1, 2, \dots, M$ векторов фильтров линейного предсказания $\mathbf{h}_{N,m}^f(k)$ и $\mathbf{h}_{N,m}^b(k)$ находится в результате решения LS-задач, аналогичных рассмотренным в разделах 11.2 и 11.3.

Значение энергии ошибок линейного предсказания (13.4), определяемой как

$$\begin{aligned} E_{N,m}^f(k) &= \sum_{i=1}^k \lambda^{k-i} e_{N,m}^f(i) e_{N,m}^{f*}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \left[x_m(i) - \mathbf{h}_{N,m}^{fH}(k) \mathbf{x}_N(i-1) \right] \left[x_m(i) - \mathbf{h}_{N,m}^{fH}(k) \mathbf{x}_N(i-1) \right]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x_m(i) x_m^*(i) - \sum_{i=1}^k \lambda^{k-i} x_m(i) \mathbf{x}_N^H(i-1) \mathbf{h}_{N,m}^f(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_{N,m}^{fH}(k) \mathbf{x}_N(i-1) x_m^*(i) + \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_{N,m}^{fH}(k) \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) \mathbf{h}_{N,m}^f(k) = \\ &= \sigma_m^{f2}(k) - \mathbf{r}_{N,m}^{fH}(k) \mathbf{h}_{N,m}^f(k) - \mathbf{h}_{N,m}^{fH}(k) \mathbf{r}_N^f + \mathbf{h}_{N,m}^{fH}(k) \mathbf{R}_N(k-1) \mathbf{h}_{N,m}^f(k), \end{aligned} \quad (13.10)$$

где

$$\sigma_m^{f2}(k) = \sum_{i=1}^k \lambda^{k-i} x_m(i) x_m^*(i) = \lambda \sigma_m^{f2}(k-1) + x_m(k) x_m^*(k), \quad (13.11)$$

$$\mathbf{r}_{N,m}^f(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) x_m^*(i) = \lambda \mathbf{r}_{N,m}^f(k-1) + \mathbf{x}_N(k-1) x_m^*(k), \quad (13.12)$$

$$\mathbf{R}_N(k-1) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) = \lambda \mathbf{R}_N(k-2) + \mathbf{x}_N(k-1) \mathbf{x}_N^H(k-1), \quad (13.13)$$

минимально при

$$\nabla_{\mathbf{h}_{N,m}^{\text{f}*}} \{E_N^{\text{f}}(k)\} = \frac{\partial E_{N,m}^{\text{f}}(k)}{\partial \mathbf{h}_{N,m}^{\text{f}*}(k)} = -\mathbf{r}_{N,m}^{\text{f}}(k) + \mathbf{R}_N(k-1)\mathbf{h}_{N,m}^{\text{f}}(k) = \mathbf{0}_N,$$

откуда

$$\mathbf{h}_{N,m}^{\text{f}}(k) = \mathbf{R}_N^{-1}(k-1)\mathbf{r}_{N,m}^{\text{f}}(k). \quad (13.14)$$

Для всех $m=1, 2, \dots, M$ уравнение (13.14) можно представить как

$$\mathbf{H}_{NM}^{\text{f}}(k) = \mathbf{R}_N^{-1}(k-1)\mathbf{R}_{NM}^{\text{f}}(k), \quad (13.15)$$

где $\mathbf{H}_{NM}^{\text{f}}(k) = [\mathbf{h}_{N,1}^{\text{f}}(k), \mathbf{h}_{N,2}^{\text{f}}(k), \dots, \mathbf{h}_{N,M}^{\text{f}}(k)]$ и $\mathbf{R}_{NM}^{\text{f}}(k) = [\mathbf{r}_{N,1}^{\text{f}}(k), \mathbf{r}_{N,2}^{\text{f}}(k), \dots, \mathbf{r}_{N,M}^{\text{f}}(k)]$.

Аналогично значение энергии ошибок линейного предсказания (13.7), определяемой как

$$\begin{aligned} E_{N,m}^{\text{b}}(k) &= \sum_{i=1}^k \lambda^{k-i} e_{N,m}^{\text{b}}(i) e_{N,m}^{\text{b}*}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} [x_m(i-N_m) - \mathbf{h}_{N,m}^{\text{bH}}(k)\mathbf{x}_N(i)] [x_m(i-N_m) - \mathbf{h}_{N,m}^{\text{bH}}(k)\mathbf{x}_N(i)]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) x_m^*(i-N_m) - \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) \mathbf{x}_N^{\text{H}}(i) \mathbf{h}_{N,m}^{\text{b}}(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_{N,m}^{\text{bH}}(k) \mathbf{x}_N(i) x_m^*(i-N_m) + \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_{N,m}^{\text{bH}}(k) \mathbf{x}_N(i) \mathbf{x}_N^{\text{H}}(i) \mathbf{h}_{N,m}^{\text{b}}(k) = \\ &= \sigma_m^{\text{b}2}(k) - \mathbf{r}_{N,m}^{\text{bH}}(k) \mathbf{h}_{N,m}^{\text{b}}(k) - \mathbf{h}_{N,m}^{\text{bH}}(k) \mathbf{r}_N^{\text{b}} + \mathbf{h}_{N,m}^{\text{bH}}(k) \mathbf{R}_N(k) \mathbf{h}_{N,m}^{\text{b}}(k), \end{aligned} \quad (13.16)$$

где

$$\sigma_m^{\text{b}2}(k) = \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) x_m^*(i-N_m) = \lambda \sigma_m^{\text{b}2}(k-1) + x_m(k-N_m) x_m^*(k-N_m), \quad (13.17)$$

$$\mathbf{r}_{N,m}^{\text{b}}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) x_m^*(i-N_m) = \lambda \mathbf{r}_{N,m}^{\text{b}}(k-1) + \mathbf{x}_N(k) x_m^*(k-N_m), \quad (13.18)$$

$$\mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^{\text{H}}(i) = \lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^{\text{H}}(k), \quad (13.19)$$

минимально при

$$\nabla_{\mathbf{h}_{N,m}^{\text{b}*}} \{E_N^{\text{b}}(k)\} = \frac{\partial E_{N,m}^{\text{b}}(k)}{\partial \mathbf{h}_{N,m}^{\text{b}*}(k)} = -\mathbf{r}_{N,m}^{\text{b}}(k) + \mathbf{R}_N(k) \mathbf{h}_{N,m}^{\text{b}}(k) = \mathbf{0}_N,$$

откуда

$$\mathbf{h}_{N,m}^{\text{b}}(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_{N,m}^{\text{b}}(k). \quad (13.20)$$

Для всех $m=1, 2, \dots, M$ уравнение (13.20) можно представить как

$$\mathbf{H}_{NM}^{\text{b}}(k) = \mathbf{R}_N^{-1}(k) \mathbf{R}_{NM}^{\text{b}}(k), \quad (13.21)$$

где $\mathbf{H}_{NM}^{\text{b}}(k) = [\mathbf{h}_{N,1}^{\text{b}}(k), \mathbf{h}_{N,2}^{\text{b}}(k), \dots, \mathbf{h}_{N,M}^{\text{b}}(k)]$ и $\mathbf{R}_{NM}^{\text{b}}(k) = [\mathbf{r}_{N,1}^{\text{b}}(k), \mathbf{r}_{N,2}^{\text{b}}(k), \dots, \mathbf{r}_{N,M}^{\text{b}}(k)]$.

По аналогии с уравнением (11.1) определим расширенный $(N+M)$ -мерный вектор входных сигналов

$$\begin{aligned}
 \mathbf{x}_{N+M}(k) &= \left[x_1(k), \mathbf{x}_{N_1}^T(k-1), x_2(k), \mathbf{x}_{N_2}^T(k-1), \right. \\
 &\quad \ldots, x_m(k), \mathbf{x}_{N_m}^T(k-1), \ldots, x_M(k), \mathbf{x}_{N_M}^T(k) \Big]^T = \\
 &= \left[\mathbf{x}_{N_1}^T(k), x_1(k-N_1), \mathbf{x}_{N_2}^T(k), x_2(k-N_2), \right. \\
 &\quad \ldots, \mathbf{x}_{N_m}^T(k), x_m(k-N_m), \ldots, \mathbf{x}_{N_M}^T(k), x_M(k-N_M) \Big]^T. \tag{13.22}
 \end{aligned}$$

Как видно, вектор $\mathbf{x}_{N+M}(k)$ многоканального адаптивного фильтра представляет собой блочный вектор, состоящий из M блоков. Каждый блок является одним из векторов $\mathbf{x}_{N_m+1}(k)$ входных сигналов многоканального фильтра. Из-за этого математические преобразования из главы 11, приводящие к быстрым RLS-алгоритмам, которые основаны на представлении корреляционной матрицы сигналов расширенного вектора в виде клеточные матрицы, состоящей из четырех клеток (11.25) и (11.48), не могут быть использованы напрямую.

Тем не менее такое представление корреляционной матрицы можно осуществить, если с помощью перестановочных матриц выполнить преобразование вектора $\mathbf{x}_{N+M}(k)$ как

$$\mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_m(k) \\ \vdots \\ x_M(k) \\ \mathbf{x}_{N_1}(k-1) \\ \mathbf{x}_{N_2}(k-1) \\ \vdots \\ \mathbf{x}_{N_m}(k-1) \\ \vdots \\ \mathbf{x}_{N_M}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} \tag{13.23}$$

и как

$$\mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) = \begin{bmatrix} \mathbf{x}_{N_1}(k) \\ \mathbf{x}_{N_2}(k) \\ \vdots \\ \mathbf{x}_{N_m}(k) \\ \vdots \\ \mathbf{x}_{N_M}(k) \\ x_1(k-N_1) \\ x_2(k-N_2) \\ \vdots \\ x_m(k-N_m) \\ \vdots \\ x_M(k-N_M) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix}, \tag{13.24}$$

где \mathbf{T}_{N+M} и \mathbf{S}_{N+M} — перестановочные матрицы, $\mathbf{x}_M(k) = [x_1(k), x_2(k), \dots, x_m(k), \dots, x_M(k)]^T$ и $\mathbf{x}_M(k-N) = [x_1(k-N_1), x_2(k-N_2), \dots, x_m(k-N_m), \dots, x_M(k-N_M)]^T$.

Обратная перестановка элементов векторов (13.23) и (13.24) осуществляется как

$$\mathbf{x}_{N+M}(k) = \mathbf{T}_{N+M}^T \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \mathbf{S}_{N+M}^T \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix}. \quad (13.25)$$

Структура перестановочных матриц для $M=3$ приводится ниже:

$$\mathbf{T}_{N+3}^T = \begin{bmatrix} 1 & 0 & 0 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_1} & \mathbf{0}_{N_1} & \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ 0 & 1 & 0 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_2} & \mathbf{0}_{N_2} & \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ 0 & 0 & 1 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_3} & \mathbf{0}_{N_3} & \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix},$$

$$\mathbf{T}_{N+3} = \begin{bmatrix} 1 & \mathbf{0}_{N_1}^T & 0 & \mathbf{0}_{N_2}^T & 0 & \mathbf{0}_{N_3}^T \\ 0 & \mathbf{0}_{N_1}^T & 1 & \mathbf{0}_{N_2}^T & 0 & \mathbf{0}_{N_3}^T \\ 0 & \mathbf{0}_{N_1}^T & 0 & \mathbf{0}_{N_2}^T & 1 & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_3} \\ \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{0}_{N_2} & \mathbf{I}_{N_2} & \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_2} & \mathbf{0}_{N_3} & \mathbf{I}_{N_3} \end{bmatrix},$$

$$\mathbf{S}_{N+3}^T = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} & \mathbf{0}_{N_1} & \mathbf{0}_{N_1} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 1 & 0 & 0 \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} & \mathbf{0}_{N_2} & \mathbf{0}_{N_2} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 0 & 1 & 0 \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} & \mathbf{0}_{N_3} & \mathbf{0}_{N_3} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{S}_{N+3} = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} \\ \mathbf{O}_{N_2 N_1} & \mathbf{0}_{N_2} & \mathbf{I}_{N_2} & \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} \\ \mathbf{O}_{N_3 N_1} & \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_2} & \mathbf{0}_{N_3} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} \\ \mathbf{0}_{N_1}^T & 1 & \mathbf{0}_{N_2}^T & 0 & \mathbf{0}_{N_3}^T & 0 \\ \mathbf{0}_{N_1}^T & 0 & \mathbf{0}_{N_2}^T & 1 & \mathbf{0}_{N_3}^T & 0 \\ \mathbf{0}_{N_1}^T & 0 & \mathbf{0}_{N_2}^T & 0 & \mathbf{0}_{N_3}^T & 1 \end{bmatrix}.$$



Такие матрицы могут быть получены аналогичным образом для любых значений M и N_m . Для этих матриц справедливы соотношения $\mathbf{T}_{N+M}^T = \mathbf{T}_{N+M}^{-1}$ и $\mathbf{S}_{N+M}^T = \mathbf{S}_{N+M}^{-1}$, а значит,

$$\mathbf{T}_{N+M} \mathbf{T}_{N+M}^T = \mathbf{I}_{N+M} \quad (13.26)$$

и

$$\mathbf{S}_{N+M} \mathbf{S}_{N+M}^T = \mathbf{I}_{N+M}. \quad (13.27)$$

Быстрые RLS-алгоритмы, основанные на использовании приведенных перестановочных матриц, относятся к алгоритмам, имеющим в своем названии аббревиатуру Block Step Up Step Down (B-SUSD). С помощью этих матриц осуществляется перестановка элементов векторов и матриц, участвующих в обработке, с числом элементов $N+M$ и $(N+M) \times (N+M)$, таким образом, что за один шаг формируются искомые векторы с числом элементов (с размерностью блока) M и N и матрицы с числом элементов $M \times M$ и $N \times N$, или же эти векторы и матрицы расформированыются обратно тоже за один шаг. В результате, апостериорные векторы коэффициентов Калмана вычисляются путем формирования вектора $\mathbf{g}_{N+M}(k)$ из вектора $\mathbf{g}_N(k-1)$, из которого затем находится интересуемый вектор $\mathbf{g}_N(k)$. Таким образом осуществляется преобразование $\mathbf{g}_N(k-1) \rightarrow \mathbf{g}_{N+M}(k)$ (Block Step Up) $\rightarrow \mathbf{g}_N(k)$ (Block Step Down). Аналогично вычисляются априорные векторы коэффициентов Калмана $\mathbf{t}_N(k-1) \rightarrow \mathbf{t}_{N+M}(k)$ (Block Step Up) $\rightarrow \mathbf{t}_N(k)$ (Block Step Down).

Для выполнения этих вычислений корреляционную матрицу $\mathbf{R}_{N+M}(k)$ необходимо представить как

$$\begin{aligned} \mathbf{R}_{N+M}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+M}(i) \mathbf{x}_{N+M}^H(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \left\{ \mathbf{T}_{N+M}^T \begin{bmatrix} \mathbf{x}_M(i) \\ \mathbf{x}_N(i-1) \end{bmatrix} \left[\mathbf{x}_M^H(i), \mathbf{x}_N^H(i-1) \right] \mathbf{T}_{N+M} \right\} = \\ &= \mathbf{T}_{N+M}^T \left\{ \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_M(i) \\ \mathbf{x}_N(i-1) \end{bmatrix} \left[\mathbf{x}_M^H(i), \mathbf{x}_N^H(i-1) \right] \right\} \mathbf{T}_{N+M} = \quad (13.28) \\ &= \mathbf{T}_{N+M}^T \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i) \mathbf{x}_M^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i) \mathbf{x}_N^H(i-1) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_M^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) \end{bmatrix} \mathbf{T}_{N+M} = \\ &= \mathbf{T}_{N+M}^T \begin{bmatrix} \mathbf{R}_M^{f2}(k) & \mathbf{R}_{NM}^{fH}(k) \\ \mathbf{R}_{NM}^f(k) & \mathbf{R}_N(k-1) \end{bmatrix} \mathbf{T}_{N+M} \end{aligned}$$

и как

$$\begin{aligned}
\mathbf{R}_{N+M}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+M}(i) \mathbf{x}_{N+M}^H(i) = \\
&= \sum_{i=1}^k \lambda^{k-i} \left\{ \mathbf{S}_{N+M}^T \begin{bmatrix} \mathbf{x}_N(i) \\ \mathbf{x}_M(i-N) \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^H(i), \mathbf{x}_M^H(i-N) \end{bmatrix} \mathbf{S}_{N+M} \right\} = \\
&= \mathbf{S}_{N+M}^T \left\{ \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_N(i) \\ \mathbf{x}_M(i-N) \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^H(i), \mathbf{x}_M^H(i-N) \end{bmatrix} \right\} \mathbf{S}_{N+M} = \\
&= \mathbf{S}_{N+M}^T \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_M^H(i-N) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i-N) \mathbf{x}_N^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i-N) \mathbf{x}_M^H(i-N) \end{bmatrix} \mathbf{S}_{N+M} = \\
&= \mathbf{S}_{N+M}^T \begin{bmatrix} \mathbf{R}_N(k) & \mathbf{R}_{NM}^b(k) \\ \mathbf{R}_{NM}^{bH}(k) & \mathbf{R}_M^b(k) \end{bmatrix} \mathbf{S}_{N+M}.
\end{aligned} \tag{13.29}$$

Далее, следуя уравнениям (11.12)–(11.17) и (11.37)–(11.40), матрицы векторов ВК фильтров линейного предсказания (13.15) и (13.21) входных сигналов многоканального фильтра можно представить в виде рекуррентных соотношений

$$\begin{aligned}
\mathbf{H}_{NM}^f(k) &= \mathbf{R}_N^{-1}(k-1) \mathbf{R}_{NM}^f(k) = \\
&= \mathbf{R}_N^{-1}(k-1) [\lambda \mathbf{R}_{NM}^f(k-1) + \mathbf{x}_N(k-1) \mathbf{x}_M^H(k)] = \\
&= \lambda \mathbf{R}_N^{-1}(k-1) \mathbf{R}_{NM}^f(k-1) + \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) \mathbf{x}_M^H(k) = \\
&= \lambda \lambda^{-1} [\mathbf{R}_N^{-1}(k-2) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2)] \mathbf{R}_{NM}^f(k-1) + \\
&+ \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) \mathbf{x}_M^H(k) = \\
&= \mathbf{R}_N^{-1}(k-2) \mathbf{R}_{NM}^f(k-1) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{R}_N^{-1}(k-2) \mathbf{R}_{NM}^f(k-1) + \\
&+ \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) \mathbf{x}_M^H(k) = \\
&= \mathbf{H}_{NM}^f(k-1) - \mathbf{g}_N(k-1) \mathbf{x}_N^H(k-1) \mathbf{H}_{NM}^f(k-1) + \mathbf{g}_N(k-1) \mathbf{x}_M^H(k) = \\
&= \mathbf{H}_{NM}^f(k-1) + \mathbf{g}_N(k-1) [\mathbf{x}_M^H(k) - \mathbf{x}_N^H(k-1) \mathbf{H}_{NM}^f(k-1)] = \\
&= \mathbf{H}_{NM}^f(k-1) + \mathbf{g}_N(k-1) \boldsymbol{\alpha}_M^{fH}(k)
\end{aligned} \tag{13.30}$$

И

$$\begin{aligned}
\mathbf{H}_{NM}^b(k) &= \mathbf{R}_N^{-1}(k) \mathbf{R}_{NM}^b(k) = \mathbf{R}_N^{-1}(k) [\lambda \mathbf{R}_{NM}^b(k-1) + \mathbf{x}_N(k) \mathbf{x}_M^H(k-N)] = \\
&= \lambda \mathbf{R}_N^{-1}(k) \mathbf{R}_{NM}^b(k-1) + \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \mathbf{x}_M^H(k-N) = \\
&= \lambda \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)] \mathbf{R}_{NM}^b(k-1) + \\
&+ \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \mathbf{x}_M^H(k-N) = \\
&= \mathbf{R}_N^{-1}(k-1) \mathbf{R}_{NM}^b(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1) \mathbf{R}_{NM}^b(k-1) + \\
&+ \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \mathbf{x}_M^H(k-N) = \\
&= \mathbf{H}_{NM}^b(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \mathbf{H}_{NM}^b(k-1) + \mathbf{g}_N(k) \mathbf{x}_M^H(k-N) = \\
&= \mathbf{H}_{NM}^b(k-1) + \mathbf{g}_N(k) [\mathbf{x}_M^H(k-N) - \mathbf{x}_N^H(k) \mathbf{H}_{NM}^b(k-1)] = \\
&= \mathbf{H}_{NM}^b(k-1) + \mathbf{g}_N(k) \boldsymbol{\alpha}_M^{bH}(k),
\end{aligned} \tag{13.31}$$

где

$$\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \quad (13.32)$$

— вектор коэффициентов Калмана многоканального адаптивного фильтра, определяемый аналогично вектору коэффициентов Калмана одноканального адаптивного фильтра (6.79), а $\alpha_M^f(k) = [\alpha_{N,1}^f(k), \alpha_{N,2}^f(k), \dots, \alpha_{N,M}^f(k)]^T$ и $\alpha_M^b(k) = [\alpha_{N,1}^b(k), \alpha_{N,2}^b(k), \dots, \alpha_{N,M}^b(k)]^T$ — векторы априорных ошибок линейного предсказания, элементы которых определяются как

$$\alpha_{N,m}^f(k) = x_m(k) - \sum_{m=1}^M \sum_{n=1}^{N_m} h_{n,m}^{f*}(k-1) x_m(k-n) = x_m(k) - \mathbf{h}_{N,m}^{fH}(k-1) \mathbf{x}_N(k-1) \quad (13.33)$$

и

$$\alpha_{N,m}^b(k) = x_m(k - N_m) - \sum_{m=1}^M \sum_{n=1}^{N_m} h_{n,m}^{b*}(k-1) x_m(k) = x(k - N_m) - \mathbf{h}_{N,m}^{bH}(k-1) \mathbf{x}_N(k). \quad (13.34)$$

Соотношения (13.26)–(13.31) далее используются для рекурсивного B-SUSD вычисления векторов коэффициентов Калмана.

13.3. B-SUSD-вычисление векторов коэффициентов Калмана и отношений ошибок линейного предсказания

Для быстрого вычисления векторов коэффициентов Калмана в многоканальных B-SUSD RLS-алгоритмах требуется определить ряд переменных, введенных в главе 11.

Для начала, аналогично уравнениям (11.4) и (11.29), определим матрицы

$$\begin{aligned} \mathbf{E}_M^f(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{e}_M^f(i) \mathbf{e}_M^{fH}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} [\mathbf{x}_M(i) - \mathbf{H}_{NM}^{fH}(k) \mathbf{x}_N(i-1)] [\mathbf{x}_M(i) - \mathbf{H}_{NM}^{fH}(k) \mathbf{x}_N(i-1)]^H = \\ &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i) \mathbf{x}_M^H(i) - \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i) \mathbf{x}_N^H(i-1) \mathbf{H}_{NM}^f(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{H}_{NM}^{fH}(k) \mathbf{x}_N(i-1) \mathbf{x}_M^H(i) + \\ &\quad + \sum_{i=1}^k \lambda^{k-i} \mathbf{H}_{NM}^{fH}(k) \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) \mathbf{H}_{NM}^f(k) = \\ &= \mathbf{R}_M^{f2}(k) - \mathbf{R}_{NM}^{fH}(k) \mathbf{H}_{NM}^f(k) - \mathbf{H}_{NM}^{fH}(k) \mathbf{R}_{NM}^f(k) + \\ &\quad + \mathbf{H}_{NM}^{fH}(k) \mathbf{R}_N(k-1) \mathbf{H}_{NM}^f(k) \end{aligned} \quad (13.35)$$

и

$$\begin{aligned}
\mathbf{E}_M^b(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{e}_M^b(i) \mathbf{e}_M^{bH}(i) = \\
&= \sum_{i=1}^k \lambda^{k-i} [\mathbf{x}_M(i-N) - \mathbf{H}_{NM}^{bH}(k) \mathbf{x}_N(i)] [\mathbf{x}_M(i-N) - \mathbf{H}_{NM}^{bH}(k) \mathbf{x}_N(i)]^H = \\
&= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i-N) \mathbf{x}_M^H(i-N) - \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_M(i-N) \mathbf{x}_N^H(i) \mathbf{H}_{NM}^b(k) - \\
&\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{H}_{NM}^{bH}(k) \mathbf{x}_N(i) \mathbf{x}_M(i-N) + \sum_{i=1}^k \lambda^{k-i} \mathbf{H}_{NM}^{bH}(k) \mathbf{x}_N(i) \mathbf{x}_N^H(i) \mathbf{H}_{NM}^b(k) = \\
&= \mathbf{R}_M^{b2}(k) - \mathbf{R}_{NM}^{bH}(k) \mathbf{H}_{NM}^b - \mathbf{H}_{NM}^{bH}(k) \mathbf{R}_{NM}^b + \mathbf{H}_{NM}^{bH}(k) \mathbf{R}_N(k) \mathbf{H}_{NM}^b(k),
\end{aligned} \tag{13.36}$$

где $\mathbf{e}_M^f(k) = [e_{N,1}^f(k), e_{N,2}^f(k), \dots, e_{N,M}^f(k)]^T$ и $\mathbf{e}_M^b(k) = [e_{N,1}^b(k), e_{N,2}^b(k), \dots, e_{N,M}^b(k)]^T$ — векторы апостериорных ошибок линейного предсказания, элементы которых определяются с помощью уравнений (13.4) и (13.7). Другие элементы уравнений (13.35) и (13.36) определяются уравнениями (13.28)–(13.31).

Подставляя (13.15) и (13.21) в уравнения (13.35) и (13.36), а также используя преобразования, аналогичные (11.18) и (11.41), получим корреляционные матрицы ошибок линейного предсказания, характеризующиеся минимальной нормой Фробениуса, т. е. матрицы вида

$$\mathbf{E}_{M,\min}^f(k) = \mathbf{R}_M^{f2}(k) - \mathbf{R}_{NM}^{fH}(k) \mathbf{H}_{NM}^f(k) = \lambda \mathbf{E}_{M,\min}^f(k-1) + \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k), \tag{13.37}$$

$$\mathbf{E}_{M,\min}^b(k) = \mathbf{R}_M^{b2}(k) - \mathbf{R}_{NM}^{bH}(k) \mathbf{H}_{NM}^b(k) = \lambda \mathbf{E}_{M,\min}^b(k-1) + \mathbf{e}_M^b(k) \boldsymbol{\alpha}_M^{bH}(k). \tag{13.38}$$

Из анализа структуры матриц

$$\mathbf{R}_M^{f2}(k) - \mathbf{R}_{NM}^{fH}(k) \mathbf{H}_{NM}^f(k) = \mathbf{R}_M^{f2}(k) - \mathbf{R}_{NM}^{fH}(k) \mathbf{R}_N^{-1}(k-1) \mathbf{R}_{NM}^f(k), \tag{13.39}$$

$$\mathbf{R}_M^{b2}(k) - \mathbf{R}_{NM}^{bH}(k) \mathbf{H}_{NM}^b(k) = \mathbf{R}_M^{b2}(k) - \mathbf{R}_{NM}^{bH}(k) \mathbf{R}_N^{-1}(k) \mathbf{R}_{NM}^b(k) \tag{13.40}$$

в уравнениях (13.37) и (13.38) следует, что матрицы $\mathbf{E}_{M,\min}^f(k)$ и $\mathbf{E}_{M,\min}^b(k)$ являются эрмитовыми и положительно определенными (см. главу 3), а значит, они обратимы. Это свойство далее используется при получении формулы для вычисления вектора коэффициентов Калмана $\mathbf{g}_N(k)$ с помощью вектора коэффициентов Калмана $\mathbf{g}_{N+M}(k)$.

Введя для матрицы векторов ВК фильтра линейного предсказания вперед обозначение

$$\mathbf{A}_{(N+M)M}(k) = [\mathbf{I}_M, -\mathbf{H}_{NM}^{fT}(k)]^T \tag{13.41}$$

и используя (13.23) и (13.30), векторы априорных и апостериорных ошибок линейного предсказания можно представить как

$$\boldsymbol{\alpha}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k-1) \mathbf{x}_N(k-1) = \mathbf{A}_{(N+M)M}^H(k-1) \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) \tag{13.42}$$

и

$$\mathbf{e}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k) \mathbf{x}_N(k-1) = \mathbf{A}_{(N+M)M}^H(k) \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k), \tag{13.43}$$

а уравнение (13.41) — как

$$\mathbf{A}_{(N+M)M}(k) = \mathbf{A}_{(N+M)M}(k-1) + \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} \boldsymbol{\alpha}_M^{\text{fH}}(k). \quad (13.44)$$

Аналогичные обозначения можно ввести для матриц векторов ВК фильтра линейного предсказания назад

$$\mathbf{B}_{(N+M)M}(k) = \begin{bmatrix} -\mathbf{H}_{NM}^{\text{bT}}(k), \mathbf{I}_M \end{bmatrix}^{\text{T}}. \quad (13.45)$$

Затем используя (13.24) и (13.31), векторы априорных и апостериорных ошибок линейного предсказания можно представить как

$$\boldsymbol{\alpha}_M^{\text{b}}(k) = \mathbf{x}_M(k-N) - \mathbf{H}_{NM}^{\text{bT}}(k-1) \mathbf{x}_N(k) = \mathbf{B}_{(N+M)M}^{\text{H}}(k-1) \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) \quad (13.46)$$

и

$$\boldsymbol{\epsilon}_M^{\text{b}}(k) = \mathbf{x}_M(k-N) - \mathbf{H}_{NM}^{\text{bT}}(k) \mathbf{x}_N(k) = \mathbf{B}_{(N+M)M}^{\text{H}}(k) \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k), \quad (13.47)$$

а уравнение (13.45) — как

$$\mathbf{B}_{(N+M)M}(k) = \mathbf{B}_{(N+M)M}(k-1) + \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} \boldsymbol{\alpha}_M^{\text{bH}}(k). \quad (13.48)$$

Далее, используя лемму об обращении клеточных матриц (11.51) и выполнив преобразования, аналогичные (11.53)–(11.58), можно осуществить обращение матрицы (13.28) как

$$\begin{aligned} \mathbf{R}_{N+M}^{-1}(k) &= \\ &= \mathbf{T}_{N+M}^{\text{T}} \begin{bmatrix} \left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} & -\left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} \mathbf{H}_{NM}^{\text{fH}}(k) \\ -\mathbf{H}_{NM}^{\text{f}}(k) \left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} & \mathbf{H}_{NM}^{\text{f}}(k) \left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} \mathbf{H}_{NM}^{\text{fH}}(k) + \mathbf{R}_N^{-1}(k-1) \end{bmatrix} \mathbf{T}_{N+M} = \\ &= \mathbf{T}_{N+M}^{\text{T}} \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^{\text{T}} \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{I}_M \\ -\mathbf{H}_{NM}^{\text{fH}}(k) \end{bmatrix} \left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} \left[\mathbf{I}_M, -\mathbf{H}_{NM}^{\text{fH}}(k) \right] \right\} \mathbf{T}_{N+M} = \\ &= \mathbf{T}_{N+M}^{\text{T}} \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^{\text{T}} \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\text{min}}^{\text{f}}(k) \right]^{-1} \mathbf{A}_{(N+M)M}^{\text{H}}(k) \right\} \mathbf{T}_{N+M}. \end{aligned} \quad (13.49)$$

Таким же образом используя лемму (11.52) и преобразования, аналогичные (11.59)–(11.63), можно осуществить обращение матрицы (13.29) как

$$\begin{aligned} \mathbf{R}_{N+M}^{-1}(k) &= \\ &= \mathbf{S}_{N+M}^{\text{T}} \begin{bmatrix} \mathbf{R}_N^{-1}(k) + \mathbf{H}_{NM}^{\text{b}}(k) \left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \mathbf{H}_{NM}^{\text{bH}}(k) & -\mathbf{H}_{NM}^{\text{b}} \left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \\ -\left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \mathbf{H}_{NM}^{\text{bH}}(k) & \left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \end{bmatrix} \mathbf{S}_{N+M} = \\ &= \mathbf{S}_{N+M}^{\text{T}} \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^{\text{T}} & \mathbf{O}_M \end{bmatrix} + \begin{bmatrix} -\mathbf{H}_{NM}^{\text{b}}(k) \\ \mathbf{I}_M \end{bmatrix} \left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \left[-\mathbf{H}_{NM}^{\text{bH}}(k), \mathbf{I}_M \right] \right\} \mathbf{S}_{N+M} = \\ &= \mathbf{S}_{N+M}^{\text{T}} \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^{\text{T}} & \mathbf{O}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\text{min}}^{\text{b}}(k) \right]^{-1} \mathbf{B}_{(N+M)M}^{\text{H}}(k) \right\} \mathbf{S}_{N+M}. \end{aligned} \quad (13.50)$$

Теперь, используя уравнения (13.49), (13.50), (13.23), (13.24), (13.43), (13.47), легко получить два выражения для вычисления вектора апостериорных коэффициентов Калмана $\mathbf{g}_{N+M}(k) = \mathbf{R}_{N+M}^{-1}(k) \mathbf{x}_{N+M}(k)$ как

$$\begin{aligned}
 \mathbf{g}_{N+M}(k) &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{A}_{(N+M)M}^H(k) \right\} \times \\
 &\times \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{A}_{(N+M)M}^H(k) \right\} \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \right. \\
 &\quad \left. + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f \right]^{-1} \mathbf{A}_{(N+M)M}^H(k) \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} \right\} = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M & \\ \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k-1) & \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\} = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M & \\ \mathbf{g}_N(k-1) & \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}
 \end{aligned} \tag{13.51}$$

И

$$\begin{aligned}
 \mathbf{g}_{N+M}(k) &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k) \right\} \times \\
 &\times \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k) \right\} \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} + \right. \\
 &\quad \left. + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k) \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} \right\} = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\} = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\}.
 \end{aligned} \tag{13.52}$$

Векторы апостериорных коэффициентов Калмана

$$\mathbf{t}_{N+M}(k) = \mathbf{R}_{N+M}^{-1}(k-1) \mathbf{x}_{N+M}(k) \tag{13.53}$$

определяются аналогично как

$$\begin{aligned}
 \mathbf{t}_{N+M}(k) &= \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-2) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{A}_{(N+M)M}^H(k-1) \right\} \times \\
 &\times \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-2) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{A}_{(N+M)M}^H(k-1) \right\} \times \\
 &\times \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{O}_M & \mathbf{O}_{NM}^T \\ \mathbf{O}_{NM} & \mathbf{R}_N^{-1}(k-2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \right. \\
 &\left. + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{A}_{(N+M)M}^H(k-1) \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} \right\} = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{R}_N^{-1}(k-2) \mathbf{x}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\} = \\
 &= \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\}
 \end{aligned} \tag{13.54}$$

И

$$\begin{aligned}
 \mathbf{t}_{N+M}(k) &= \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k-1) \right\} \times \\
 &\times \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k-1) \right\} \times \\
 &\times \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) & \mathbf{O}_{NM} \\ \mathbf{O}_{NM}^T & \mathbf{O}_M \end{bmatrix} \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} + \right. \\
 &\left. + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \mathbf{B}_{(N+M)M}^H(k-1) \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} \right\} = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\} = \\
 &= \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\}.
 \end{aligned} \tag{13.55}$$

Соотношение между векторами $\mathbf{t}_N(k)$ и $\mathbf{g}_N(k)$ многоканального адаптивного фильтра, а также соотношения между апостериорными и априорными ошибками являются такими же, как и в одноканальных адаптивных фильтрах (см. разделы 11.4 и 11.5). Отношение ошибок $\varphi_{N+M}(k)$ также вычисляется с помощью выражений, аналогичных уравнениям (11.83) и (11.84) с использованием формул (13.23), (13.24), (13.51) и (13.52).

Согласно уравнениям (11.83) и (11.84)

$$\begin{aligned}
 \varphi_{N+M}(k) &= 1 - \mathbf{g}_{N+M}^H(k) \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left(\mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\} \right)^H \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}^H \left(\mathbf{T}_{N+M}^T \right)^T \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}^H \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}^H \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \\
 &= 1 - \left[\mathbf{0}_M^T, \mathbf{g}_N^H(k-1) \right] \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} - \left\{ \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}^H \times \\
 &\quad \times \mathbf{A}_{(N+M)M}^H(k) \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = 1 - \mathbf{g}_N^H(k-1) \mathbf{x}_N(k-1) - \mathbf{e}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) = \\
 &= \varphi_N(k-1) - \mathbf{e}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k)
 \end{aligned} \tag{13.56}$$

И

$$\begin{aligned}
 \varphi_{N+M}(k) &= 1 - \mathbf{g}_{N+M}^H(k) \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left(\mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\} \right)^H \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\}^H \left(\mathbf{S}_{N+M}^T \right)^T \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\}^H \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\}^H \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = \\
 &= 1 - \left[\mathbf{g}_N^H(k), \mathbf{0}_M^T \right] \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} - \left\{ \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\}^H \times \\
 &\quad \times \mathbf{B}_{(N+M)M}^H(k) \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = 1 - \mathbf{g}_N^H(k) \mathbf{x}_N(k) - \mathbf{e}_M^{bH}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) = \\
 &= \varphi_N(k) - \mathbf{e}_M^{bH}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k).
 \end{aligned} \tag{13.57}$$

Обратное отношение ошибок $\varphi_{N+M}^{-1}(k)$ вычисляется согласно уравнениям (11.89) и (11.90) как

$$\begin{aligned}
 \varphi_{N+M}^{-1}(k) &= 1 + \lambda^{-1} \mathbf{t}_{N+M}^H(k) \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left(\mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\} \right)^H \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\}^H \left(\mathbf{T}_{N+M}^T \right)^T \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\}^H \mathbf{T}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\}^H \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \quad (13.58) \\
 &= 1 + \lambda^{-1} \left[\mathbf{0}_M^T, \mathbf{t}_N^H(k-1) \right] \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} + \lambda^{-1} \left\{ \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) \right\}^H \times \\
 &\quad \times \mathbf{A}_{(N+M)M}^H(k-1) \begin{bmatrix} \mathbf{x}_M(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \mathbf{t}_N^H(k-1) \mathbf{x}_N(k-1) + \lambda^{-1} \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k) = \\
 &= \varphi_N^{-1}(k-1) + \lambda^{-1} \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k)
 \end{aligned}$$

II

$$\begin{aligned}
 \varphi_{N+M}^{-1}(k) &= 1 + \lambda^{-1} \mathbf{t}_{N+M}^H(k) \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left(\mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\} \right)^H \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\}^H \left(\mathbf{S}_{N+M}^T \right)^T \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\}^H \mathbf{S}_{N+M} \mathbf{x}_{N+M}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\}^H \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = \quad (13.59) \\
 &= 1 + \lambda^{-1} \left[\mathbf{t}_N^H(k), \mathbf{0}_M^T \right] \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} + \lambda^{-1} \left\{ \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\}^H \times \\
 &\quad \times \mathbf{B}_{(N+M)M}^H(k-1) \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k-N) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \mathbf{t}_N^H(k) \mathbf{x}_N(k) + \lambda^{-1} \boldsymbol{\alpha}_M^{bH}(k) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) = \\
 &= \varphi_N^{-1}(k) + \lambda^{-1} \boldsymbol{\alpha}_M^{bH}(k) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k).
 \end{aligned}$$

13.4. B-SUSD быстрые RLS-алгоритмы

Соотношения, приведенные в разделах 13.2 и 13.3, используются при получении вычислительных процедур многоканальных B-SUSD быстрых RLS-алгоритмов, одноканальные версии которых приведены в разделах 11.7–11.9.

Для получения многоканального FK-алгоритма в уравнениях (13.51) и (13.52) необходимо выделить сомножители, заключенные в фигурные скобки, для чего можно воспользоваться соотношениями (13.26) и (13.27). Тогда получим, что

$$\begin{aligned} \mathbf{T}_{N+M} \mathbf{g}_{N+M}(k) &= \\ &= \mathbf{T}_{N+M} \mathbf{T}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\} = \\ &= \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \end{aligned} \quad (13.60)$$

и

$$\begin{aligned} \mathbf{S}_{N+M} \mathbf{g}_{N+M}(k) &= \mathbf{S}_{N+M} \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \right\} = \\ &= \begin{bmatrix} \mathbf{g}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{\mathbf{q}}_M(k) \end{bmatrix}. \end{aligned} \quad (13.61)$$

Так как векторы $\mathbf{g}_{N+M}(k)$, вычисленные по формулам (13.51) и (13.52), равны между собой, то, вычислив $\mathbf{g}_{N+M}(k)$ по формуле (13.51), а затем выполнив перемножение $\mathbf{S}_{N+M} \mathbf{g}_{N+M}(k)$, можно из уравнения (13.61) определить

$$\tilde{\mathbf{q}}_M(k) = \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) \quad (13.62)$$

и

$$\begin{aligned} \tilde{\mathbf{q}}_N(k) &= \mathbf{g}_N(k) - \mathbf{H}_{NM}^b(k) \left[\mathbf{E}_{M,\min}^b(k) \right]^{-1} \mathbf{e}_M^b(k) = \\ &= \mathbf{g}_N(k) - \mathbf{H}_{NM}^b(k) \tilde{\mathbf{q}}_M(k) = \\ &= \mathbf{g}_N(k) - \left[\mathbf{H}_{NM}^b(k-1) + \mathbf{g}_N(k) \boldsymbol{\alpha}_M^{bH}(k) \right] \tilde{\mathbf{q}}_M(k) = \\ &= \mathbf{g}_N(k) - \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k) - \mathbf{g}_N(k) \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k) = \\ &= \mathbf{g}_N(k) \left[1 - \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k) \right] - \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k). \end{aligned} \quad (13.63)$$

В (13.63) использовано уравнение (13.31) для матрицы $\mathbf{H}_{NM}^b(k)$. Из (13.63) можно определить

$$\mathbf{g}_N(k) = \left[\tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k) \right] \left[1 - \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k) \right]^{-1}. \quad (13.64)$$

Таким образом, уравнения (13.60), (13.61) и (13.64) позволяют рекурсивно вычислять вектор коэффициентов Калмана $\mathbf{g}_N(k)$ при наличии значений других переменных, входящих в эти уравнения.



Вычислительная процедура многоканального B-SUSD FK-алгоритма, основанная на использовании уравнений (13.51), (13.61) и (13.64), приведена в табл. 13.1.

Таблица 13.1. Многоканальный B-SUSD FK-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{E}_{M,\min}^f(0) = \delta^2 \mathbf{I}_M$, $\mathbf{H}_{NM}^f(k) = \mathbf{O}_{NM}$, $\mathbf{H}_{NM}^b(k) = \mathbf{O}_{NM}$,	(13.1.0)
$\mathbf{g}_N(0) = \mathbf{0}_N$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$	
For $k = 1, 2, \dots, K$	
$\boldsymbol{\alpha}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k-1) \mathbf{x}_N(k-1)$	(13.1.1)
$\boldsymbol{\alpha}_M^b(k) = \mathbf{x}_M(k-N) - \mathbf{H}_{NM}^{bT}(k-1) \mathbf{x}_N(k)$	(13.1.2)
$\mathbf{H}_{NM}^f(k) = \mathbf{H}_{NM}^f(k-1) + \mathbf{g}_N(k-1) \boldsymbol{\alpha}_M^{fH}(k)$	(13.1.3)
$\mathbf{e}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k) \mathbf{x}_N(k-1)$	(13.1.4)
$\mathbf{E}_{M,\min}^f(k) = \lambda \mathbf{E}_{M,\min}^f(k-1) + \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k)$	(13.1.5)
$\bar{\mathbf{g}}_{N+M}(k) = \begin{bmatrix} \mathbf{0}_M \\ \mathbf{g}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k)$	(13.1.6)
$\tilde{\mathbf{g}}_{N+M}(k) = \mathbf{S}_{N+M} \mathbf{T}_{N+M}^T \bar{\mathbf{g}}_{N+M}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{\mathbf{q}}_M(k) \end{bmatrix}$	(13.1.7)
$\mathbf{g}_N(k) = \left[\tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k) \right] \left[1 - \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k) \right]^{-1}$	(13.1.8)
$\mathbf{H}_{NM}^b(k) = \mathbf{H}_{NM}^b(k-1) + \mathbf{g}_N(k) \boldsymbol{\alpha}_M^{bH}(k)$	(13.1.9)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(13.1.10)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$	(13.1.11)
End for k	

При выполнении алгоритма, приведенного в табл. 13.1, и других B-SUSD RLS-алгоритмов необходимо обеспечить формирование векторов $\mathbf{x}_M(k)$, $\mathbf{x}_N(k-1)$, $\mathbf{x}_M(k)$ и $\mathbf{x}_M(k-N)$, которое можно, например, осуществить путем формирования вектора $\mathbf{x}_{N+M}(k)$ и выполнения операций (13.23) и (13.24).

Вычислительная сложность B-SUSD FK-алгоритма равна $7MN + 3N + 4M^2$ операциям умножения, $7MN + 3N + 3M^2$ операциям сложения и одной операции деления, т. е. $O(NM)$ при $M \ll N$ без учета операции перестановки элементов вектора $\bar{\mathbf{g}}_{N+M}(k)$ с помощью произведения матриц $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$, а также без учета сложности обращения матрицы $\left[\mathbf{E}_{M,\min}^f(k) \right]^{-1}$. Обращение матрицы $\left[\mathbf{E}_{M,\min}^f(k) \right]^{-1}$ будет рассмотрено в конце настоящего раздела.

Умножение на перестановочные матрицы не влечет за собой выполнения дополнительных арифметических операций, поскольку для конкретных значений N_m в каналах M -канального адаптивного фильтра можно заранее вычис-

лить произведения $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$ и определить правила перестановки элементов в векторах с помощью адресного вектора.

Рекурсивное вычисление вектора $\mathbf{t}_N(k)$ в многоканальных B-SUSD FTF- и FAEST-алгоритмах осуществляется посредством соотношений (13.54) и (13.55). Для этого сначала с помощью уравнения (13.54) вычисляется вектор $\mathbf{t}_{N+M}(k)$, который затем, используя соотношение (13.27), умножается на матрицу \mathbf{S}_{N+M} , чтобы результат можно было представить в виде двухкомпонентного вектора

$$\begin{aligned} \mathbf{S}_{N+M} \mathbf{t}_{N+M}(k) &= \\ &= \mathbf{S}_{N+M} \mathbf{S}_{N+M}^T \left\{ \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) \right\} = \\ &= \begin{bmatrix} \mathbf{t}_N(k) \\ \mathbf{0}_M \end{bmatrix} + \mathbf{B}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{\mathbf{q}}_M(k) \end{bmatrix}. \end{aligned} \quad (13.65)$$

Из уравнения (13.65) следует, что

$$\tilde{\mathbf{q}}_M(k) = \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k), \quad (13.66)$$

$$\begin{aligned} \tilde{\mathbf{q}}_N(k) &= \mathbf{t}_N(k) - \mathbf{H}_{NM}^b(k-1) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) = \\ &= \mathbf{t}_N(k) - \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k) \end{aligned} \quad (13.67)$$

и

$$\mathbf{t}_N(k) = \tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k). \quad (13.68)$$

Таким образом, уравнения (13.54), (13.65)–(13.68) позволяют рекурсивно вычислять вектор коэффициентов Калмана $\mathbf{t}_N(k)$. Значения $\boldsymbol{\alpha}_N^b(k)$ определяются из уравнения (13.66) как

$$\boldsymbol{\alpha}_M^b(k) = \mathbf{E}_{M,\min}^b(k-1) \tilde{\mathbf{q}}_M(k). \quad (13.69)$$

Соотношения между априорными и апостериорными ошибками в многоканальных B-SUSD быстрых RLS-алгоритмах определяются по аналогии с уравнениями (11.77)–(11.79) как

$$\mathbf{e}_N(k) = \boldsymbol{\alpha}_N^f(k) \varphi_N(k), \quad (13.70)$$

$$\mathbf{e}_M^f(k) = \boldsymbol{\alpha}_M^f(k) \varphi_N(k-1), \quad (13.71)$$

$$\mathbf{e}_M^b(k) = \boldsymbol{\alpha}_M^b(k) \varphi_N(k). \quad (13.72)$$

Для рекурсивного вычисления $\varphi_N(k)$ уравнения (13.56) и (13.57), с учетом уравнений (13.71) и (13.72), преобразуются как

$$\begin{aligned} \varphi_{N+M}(k) &= \varphi_N(k-1) - \mathbf{e}_M^f(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) = \\ &= \varphi_N(k-1) - \varphi_N(k-1) \boldsymbol{\alpha}_M^f(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) = \\ &= \varphi_N(k-1) \left\{ 1 - \boldsymbol{\alpha}_M^f(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\} \end{aligned} \quad (13.73)$$



и

$$\begin{aligned}
 \varphi_{N+M}(k) &= \varphi_N(k) - \mathbf{e}_M^{\text{bH}}(k) \left[\mathbf{E}_{M,\min}^{\text{b}}(k) \right]^{-1} \mathbf{e}_M^{\text{b}}(k) = \\
 &= \varphi_N(k) - \varphi_N(k) \boldsymbol{\alpha}_M^{\text{bH}}(k) \left[\mathbf{E}_{M,\min}^{\text{b}}(k) \right]^{-1} \mathbf{e}_M^{\text{b}}(k) = \\
 &= \varphi_N(k) \left\{ 1 - \boldsymbol{\alpha}_M^{\text{bH}}(k) \left[\mathbf{E}_{M,\min}^{\text{b}}(k) \right]^{-1} \mathbf{e}_M^{\text{b}}(k) \right\}.
 \end{aligned} \tag{13.74}$$

Из уравнения (13.73) с учетом уравнений (13.38) и (13.66) и по аналогии с уравнением (11.107) $\varphi_N(k)$ можно представить как

$$\varphi_N(k) = \varphi_{N+M}(k) \left[1 - \lambda^{-1} \varphi_{N+M}(k) \tilde{\mathbf{q}}_M^{\text{H}}(k) \boldsymbol{\alpha}_N^{\text{b}}(k) \right]^{-1}. \tag{13.75}$$

Таким образом, уравнения (13.73) и (13.75) позволяют рекурсивно вычислять переменную $\varphi_N(k)$.

С учетом выражений (11.81) и (13.70)–(13.72), уравнения для вычисления векторов ВК фильтров линейного предсказания и ВК собственно адаптивного фильтра преобразуются как

$$\begin{aligned}
 \mathbf{H}_{NM}^{\text{f}}(k) &= \mathbf{H}_{NM}^{\text{f}}(k-1) + \mathbf{g}_N(k-1) \boldsymbol{\alpha}_M^{\text{fH}}(k) = \mathbf{H}_{NM}^{\text{f}}(k-1) + \\
 &+ \lambda^{-1} \mathbf{t}_N(k-1) \varphi_N(k-1) \boldsymbol{\alpha}_M^{\text{fH}}(k) = \mathbf{H}_{NM}^{\text{f}}(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) \mathbf{e}_M^{\text{fH}}(k),
 \end{aligned} \tag{13.76}$$

$$\begin{aligned}
 \mathbf{H}_{NM}^{\text{b}}(k) &= \mathbf{H}_{NM}^{\text{b}}(k-1) + \mathbf{g}_N(k) \boldsymbol{\alpha}_N^{\text{bs}}(k) = \mathbf{H}_{NM}^{\text{b}}(k-1) + \\
 &+ \lambda^{-1} \mathbf{t}_N(k) \varphi_N(k) \boldsymbol{\alpha}_M^{\text{bH}}(k) = \mathbf{H}_{NM}^{\text{b}}(k-1) + \lambda^{-1} \mathbf{t}_N(k) \mathbf{e}_M^{\text{bH}}(k),
 \end{aligned} \tag{13.77}$$

$$\begin{aligned}
 \mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \boldsymbol{\alpha}_N^*(k) = \mathbf{h}_N(k-1) + \\
 &+ \lambda^{-1} \mathbf{t}_N(k) \varphi_N(k) \boldsymbol{\alpha}_N^*(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) \mathbf{e}_N^*(k).
 \end{aligned} \tag{13.78}$$

Полученные соотношения определяют вычислительную процедуру много-канального B-SUSD FTF-алгоритма, приведенного в табл. 13.2.

Вычислительная сложность этого алгоритма равна $5MN + 2N + 8M^2 + 4M$ операциям умножения, $5MN + 3N + 6M^2 + 2M$ операциям сложения и одной операции деления, т. е. $O(MN)$ при $M \ll N$ без учета операции перестановки элементов вектора $\tilde{\mathbf{t}}_{N+M}(k)$ с помощью произведения матриц $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^{\text{T}}$, а также без учета сложности обращения матрицы $\left[\mathbf{E}_{M,\min}^{\text{b}}(k) \right]^{-1}$. Обращение этой матрицы будет рассмотрено в конце настоящего раздела.

Многоканальный FAEST-алгоритм, приведенный в табл. 13.3, отличается от многоканального FTF-алгоритма всего лишь двумя уравнениями. В FFT-алгоритме (см. также раздел 11.8) рекурсивно вычисляется переменная $\varphi_N(k)$, а в FAEST-алгоритме (см. также раздел 11.9) — ее обратное значение $\varphi_N^{-1}(k)$. Для этого сначала используется уравнение (13.58), а затем из уравнения (13.59) с учетом уравнения (13.66) определяется $\varphi_N^{-1}(k)$ как

Таблица 13.2. Многоканальный B-SUSD FTF-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{E}_{M,\min}^f(0) = \delta^2 \mathbf{I}_M$,	(13.2.0)
$\mathbf{E}_{M,\min}^b(0) = \delta^2 \text{diag}(\lambda^{-N_1}, \lambda^{-N_2}, \dots, \lambda^{-N_M})$,	
$\mathbf{H}_{NM}^f(k) = \mathbf{O}_{NM}$, $\mathbf{H}_{NM}^b(k) = \mathbf{O}_{NM}$, $\mathbf{t}_N(0) = \mathbf{0}_N$, $\varphi_N(0) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$,	
$\mathbf{h}_N(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$	
For $k = 1, 2, \dots, K$	
$\boldsymbol{\alpha}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k-1) \mathbf{x}_N(k-1)$	(13.2.1)
$\mathbf{e}_M^f(k) = \boldsymbol{\alpha}_M^f(k) \varphi_N(k-1)$	(13.2.2)
$\bar{\mathbf{t}}_{N+M}(k) = \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k)$	(13.2.3)
$\tilde{\mathbf{t}}_{N+M}(k) = \mathbf{S}_{N+M} \mathbf{T}_{N+M}^T \bar{\mathbf{t}}_{N+M}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{\mathbf{q}}_M(k) \end{bmatrix}$	(13.2.4)
$\mathbf{H}_{NM}^f(k) = \mathbf{H}_{NM}^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) \mathbf{e}_M^{fH}(k)$	(13.2.5)
$\boldsymbol{\alpha}_M^b(k) = \mathbf{E}_{M,\min}^b(k-1) \tilde{\mathbf{q}}_M(k)$	(13.2.6)
$\mathbf{t}_N(k) = \tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k)$	(13.2.7)
$\mathbf{E}_{M,\min}^f(k) = \lambda \mathbf{E}_{M,\min}^f(k-1) + \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k)$	(13.2.8)
$\varphi_{N+M}(k) = \varphi_N(k-1) \left\{ 1 - \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} \mathbf{e}_M^f(k) \right\}$	(13.2.9)
$\varphi_N(k) = \varphi_{N+M}(k) \left[1 - \lambda^{-1} \varphi_{N+M}(k) \tilde{\mathbf{q}}_M^H(k) \boldsymbol{\alpha}_N^b(k) \right]^{-1}$	(13.2.10)
$\mathbf{e}_M^b(k) = \boldsymbol{\alpha}_M^b(k) \varphi_N(k)$	(13.2.11)
$\mathbf{E}_{M,\min}^b(k) = \lambda \mathbf{E}_{M,\min}^b(k-1) + \mathbf{e}_M^b(k) \boldsymbol{\alpha}_M^{bH}(k)$	(13.2.12)
$\mathbf{H}_{NM}^b(k) = \mathbf{H}_{NM}^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) \mathbf{e}_M^{bH}(k)$	(13.2.13)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(13.2.14)
$e_N(k) = \alpha_N(k) \varphi_N(k)$	(13.2.15)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^*(k)$	(13.2.16)
End for k	

$$\begin{aligned} \varphi_N^{-1}(k) &= \varphi_{N+M}^{-1}(k) - \lambda^{-1} \boldsymbol{\alpha}_M^{bH}(k) \left[\mathbf{E}_{M,\min}^b(k-1) \right]^{-1} \boldsymbol{\alpha}_M^b(k) = \\ &= \varphi_{N+M}^{-1}(k) - \lambda^{-1} \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k). \end{aligned} \quad (13.79)$$

Вычислительная сложность B-SUSD FAEST-алгоритма равна $5MN + 2N + 8M^2 + 4M$ операциям умножения, $5MN + 3N + 6M^2 + 2M$ операциям сложения и одной операции деления, т. е. $O(NM)$ при $M \ll N$ без учета операции перестановки элементов вектора $\bar{\mathbf{t}}_{N+M}(k)$ с помощью произведения матриц $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$, а также без учета сложности обращения матрицы $\left[\mathbf{E}_{M,\min}^f(k) \right]^{-1}$. Обращение этой матрицы рассматривается ниже.



Таблица 13.3. Многоканальный B-SUSD FAEST-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{E}_{M,\min}^f(0) = \delta^2 \mathbf{I}_M$,	(13.3.0)
$\mathbf{E}_{M,\min}^b(0) = \delta^2 \text{diag}(\lambda^{-N_1}, \lambda^{-N_2}, \dots, \lambda^{-N_M})$,	
$\mathbf{H}_{NM}^f(k) = \mathbf{O}_{NM}$, $\mathbf{H}_{NM}^b(k) = \mathbf{O}_{NM}$, $\mathbf{t}_N(0) = \mathbf{0}_N$, $\varphi_N(0) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$,	
$\mathbf{h}_N(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+M} \mathbf{T}_{N+M}^T$	
For $k = 1, 2, \dots, K$	
$\boldsymbol{\alpha}_M^f(k) = \mathbf{x}_M(k) - \mathbf{H}_{NM}^{fT}(k-1) \mathbf{x}_N(k-1)$	(13.3.1)
$\mathbf{e}_M^f(k) = \boldsymbol{\alpha}_M^f(k) \varphi_N(k-1)$	(13.3.2)
$\bar{\mathbf{t}}_{N+M}(k) = \begin{bmatrix} \mathbf{0}_M \\ \mathbf{t}_N(k-1) \end{bmatrix} + \mathbf{A}_{(N+M)M}(k-1) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k)$	(13.3.3)
$\tilde{\mathbf{t}}_{N+M}(k) = \mathbf{S}_{N+M} \mathbf{T}_{N+M}^T \bar{\mathbf{t}}_{N+M}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N(k) \\ \tilde{\mathbf{q}}_M(k) \end{bmatrix}$	(13.3.4)
$\mathbf{H}_{NM}^f(k) = \mathbf{H}_{NM}^f(k-1) + \lambda^{-1} \mathbf{t}_N(k-1) \mathbf{e}_M^{fH}(k)$	(13.3.5)
$\boldsymbol{\alpha}_M^b(k) = \mathbf{E}_{M,\min}^b(k-1) \tilde{\mathbf{q}}_M(k)$	(13.3.6)
$\mathbf{t}_N(k) = \tilde{\mathbf{q}}_N(k) + \mathbf{H}_{NM}^b(k-1) \tilde{\mathbf{q}}_M(k)$	(13.3.7)
$\mathbf{E}_{M,\min}^f(k) = \lambda \mathbf{E}_{M,\min}^f(k-1) + \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k)$	(13.3.8)
$\varphi_{N+M}^{-1}(k) = \varphi_N^{-1}(k-1) + \lambda^{-1} \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \boldsymbol{\alpha}_M^f(k)$	(13.3.9)
$\varphi_N^{-1}(k) = \varphi_{N+M}^{-1}(k) - \lambda^{-1} \boldsymbol{\alpha}_M^{bH}(k) \tilde{\mathbf{q}}_M(k)$	(13.3.10)
$\mathbf{e}_M^b(k) = \boldsymbol{\alpha}_M^b(k) \varphi_N(k)$	(13.3.11)
$\mathbf{E}_{M,\min}^b(k) = \lambda \mathbf{E}_{M,\min}^b(k-1) + \mathbf{e}_M^b(k) \boldsymbol{\alpha}_M^{bH}(k)$	(13.3.12)
$\mathbf{H}_{NM}^b(k) = \mathbf{H}_{NM}^b(k-1) + \lambda^{-1} \mathbf{t}_N(k) \mathbf{e}_M^{bH}(k)$	(13.3.13)
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(13.3.14)
$e_N(k) = \alpha_N(k) \varphi_N(k)$	(13.3.15)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N(k) e_N^*(k)$	(13.2.16)
End for k	

Недостатком рассмотренных B-SUSD алгоритмов является наличие в них операций обращения матрицы $\mathbf{E}_{M,\min}^f(k)$, что при больших значениях M является источником значительного числа дополнительных арифметических операций. Обращение матрицы $\mathbf{E}_{M,\min}^f(k)$ может быть выполнено, например, по аналогии с уравнением (6.76) с помощью MIL (5.115) $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$, где согласно уравнению (13.37) $\mathbf{C} = \lambda \mathbf{E}_{M,\min}^f(k-1)$, $\mathbf{C}^{-1} = \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1}$, $\mathbf{B}^H = \mathbf{e}_M^f(k)$, $\mathbf{D} = \boldsymbol{\alpha}_M^{fH}(k)$ и $\mathbf{A}^{-1} = -1$. В этом случае

$$\begin{aligned}
 & \left[\mathbf{E}_{M,\min}^f(k) \right]^{-1} = \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} - \\
 & - \frac{\lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k) \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1}}{1 + \boldsymbol{\alpha}_M^{fH}(k) \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k)} = \\
 & = \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} - \frac{\lambda^{-2} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1}}{\lambda^{-1} \left[\lambda + \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k) \right]} = \\
 & = \lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} - \frac{\lambda^{-1} \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k) \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1}}{\lambda + \boldsymbol{\alpha}_M^{fH}(k) \left[\mathbf{E}_{M,\min}^f(k-1) \right]^{-1} \mathbf{e}_M^f(k)}.
 \end{aligned} \tag{13.80}$$

Оценка вычислительной сложности уравнения (13.80) равна $O(M^2)$ арифметическим операциям.

В рассматриваемых далее многоканальных One Step Up Step Down (1-SUSD) быстрых RLS-алгоритмах, которые являются математически эквивалентными B-SUSD алгоритмам, необходимость обращения матриц с числом элементов $M \times M$ отсутствует, что выгодно их отличает от алгоритмов, рассмотренных в настоящем разделе.

13.5. 1-SUSD линейное предсказание

One Step Up Step Down (1-SUSD) быстрые RLS-алгоритмы также основаны на использовании перестановочных матриц, которые осуществляют перестановку элементов векторов и матриц, участвующих в вычислениях, с числом элементов $N+1$ и $(N+1) \times (N+1)$. При этом используют $2M$ перестановочных матриц, с помощью которых за M шагов из вектора $\mathbf{g}_N(k-1)$ последовательно формируют M векторов $\mathbf{g}_{N+M}^{(m)}(k)$ (при изменении m от M до 1), из которых на последнем шаге находится интересуемый вектор $\mathbf{g}_N(k)$. Это формирование осуществляется как (1 Step Up) $\mathbf{g}_N(k-1) = \mathbf{g}_N^{(0)}(k-1) = \mathbf{g}_N^{(M)}(k) \rightarrow \mathbf{g}_{N+1}^{(M)}(k) \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(M-1)}(k) \rightarrow$ (1 Step Up) $\mathbf{g}_{N+1}^{(M-1)}(k) \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(M-2)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(m)}(k) \rightarrow$ (1 Step Up) $\mathbf{g}_{N+1}^{(m)}(k) \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(m-1)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(1)}(k) \rightarrow$ (1 Step Up) $\mathbf{g}_{N+1}^{(1)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{g}_N^{(0)}(k) = \mathbf{g}_N(k) = \mathbf{g}_N^{(M)}(k+1)$.

Аналогично вычисляются априорные векторы коэффициентов Калмана как (1 Step Up) $\mathbf{t}_N(k-1) = \mathbf{t}_N^{(0)}(k-1) = \mathbf{t}_N^{(M)}(k) \rightarrow \mathbf{t}_{N+1}^{(M)}(k) \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(M-1)}(k) \rightarrow$ (1 Step Up) $\mathbf{t}_{N+1}^{(M-1)}(k) \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(M-2)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(m)}(k) \rightarrow$ (1 Step Up) $\mathbf{t}_{N+1}^{(m)}(k) \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(m-1)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(1)}(k) \rightarrow$ (1 Step Up) $\mathbf{t}_{N+1}^{(1)}(k) \rightarrow \dots \rightarrow$ (1 Step Down) $\mathbf{t}_N^{(0)}(k) = \mathbf{t}_N(k) = \mathbf{t}_N^{(M)}(k+1)$.

Таким образом, вычислительные процедуры 1-SUSD быстрых RLS-алгоритмов представляют собой комбинацию технологии перестановочных матриц, похожую на используемую в B-SUSD алгоритмах, и вычислений, аналогичных рассмотрен-



ным в главе 11. В результате этого 1-SUBSD алгоритмы можно интерпретировать как M -кратное повторение в части линейного предсказания процедур алгоритмов из главы 11. Очевидно, что вычислительная сложность полученных таким образом быстрых RLS-алгоритмов будет равна $O(NM)$.

В обозначенной выше последовательности преобразований векторов Калмана участвуют фильтры линейного предсказания в которых в качестве векторов входных сигналов используются блочные векторы

$$\begin{aligned}
 \mathbf{x}_N^{(0)}(k) &= \left[\mathbf{x}_{N_1}^T(k), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T = \mathbf{x}_N(k) \\
 \mathbf{x}_N^{(1)}(k) &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \\
 &\vdots \\
 \mathbf{x}_N^{(m)}(k) &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_m}^T(k-1), \mathbf{x}_{N_{m+1}}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \\
 &\vdots \\
 \mathbf{x}_N^{(M)}(k) &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_m}^T(k-1), \dots, \mathbf{x}_{N_M}^T(k-1) \right]^T = \mathbf{x}_N(k-1).
 \end{aligned} \tag{13.81}$$

Используя уравнение (13.81), можно сформировать M расширенных векторов входных сигналов с числом элементов $N+1$

$$\begin{aligned}
 \mathbf{x}_{N+1}^{(1)}(k) &= \\
 &= \left[x_1(k), \mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T = \\
 &= \left[\mathbf{x}_{N_1}^T(k), x_1(k-N_1), \mathbf{x}_{N_2}^T(k), \dots, \mathbf{x}_{N_m}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right] \\
 &\vdots \\
 \mathbf{x}_{N+1}^{(m)}(k) &= \\
 &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, x_m(k), \mathbf{x}_{N_m}^T(k-1), \mathbf{x}_{N_{m+1}}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T = \tag{13.82} \\
 &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_m}^T(k), x_m(k-N_m), \mathbf{x}_{N_{m+1}}^T(k), \dots, \mathbf{x}_{N_M}^T(k) \right]^T \\
 &\vdots \\
 \mathbf{x}_{N+1}^{(M)}(k) &= \\
 &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_m}^T(k-1), \dots, x_M(k), \mathbf{x}_{N_M}^T(k-1) \right]^T = \\
 &= \left[\mathbf{x}_{N_1}^T(k-1), \mathbf{x}_{N_2}^T(k-1), \dots, \mathbf{x}_{N_m}^T(k-1), \dots, \mathbf{x}_{N_M}^T(k), x_M(k-N_M) \right]^T.
 \end{aligned}$$

Для использования выражений (13.82) при вычислении векторов коэффициентов Калмана $\mathbf{g}_{N+1}^{(m)}(k)$ с помощью лемм об обращении клеточных матриц (11.51), (11.52) требуется в этих уравнениях выполнить перестановку элементов $x_1(k), \dots, x_m(k), \dots, x_M(k)$ с помощью M перестановочных матриц $\mathbf{T}_{N+1}^{(m)}$ и перестановку элементов $x_1(k-N_1), \dots, x_m(k-N_m), \dots, x_M(k-N_M)$ с помощью M перестановочных матриц $\mathbf{S}_{N+1}^{(m)}$ как

$$\mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} \tag{13.83}$$

и

$$\mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix}. \quad (13.84)$$

Обратная перестановка элементов векторов (13.83) и (13.84) осуществляется как

$$\mathbf{x}_{N+1}(k) = \mathbf{T}_{N+1}^{(m)\top} \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \mathbf{S}_{N+1}^{(m)\top} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix}. \quad (13.85)$$

Структура перестановочных матриц $\mathbf{T}_{N+1}^{(m)}$, $\mathbf{T}_{N+1}^{(m)\top}$, $\mathbf{S}_{N+1}^{(m)}$ и $\mathbf{S}_{N+1}^{(m)\top}$ для $M=3$, определенная в [151], приводится ниже:

$$\begin{aligned} \mathbf{T}_{N+1}^{(1)} &= \begin{bmatrix} 1 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix}, \quad \mathbf{T}_{N+1}^{(1)\top} = \begin{bmatrix} 1 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix}, \\ \mathbf{T}_{N+1}^{(2)} &= \begin{bmatrix} \mathbf{0}_{N_1}^T & 1 & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{I}_{N_1} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ \mathbf{O}_{N_2 N_1} & \mathbf{0}_{N_2} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{O}_{N_3 N_1} & \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix}, \quad \mathbf{T}_{N+1}^{(2)\top} = \begin{bmatrix} \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ 1 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix}, \\ \mathbf{T}_{N+1}^{(3)} &= \begin{bmatrix} \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & 1 & \mathbf{0}_{N_3}^T \\ \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_3} \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{0}_{N_3} & \mathbf{I}_{N_3} \end{bmatrix}, \quad \mathbf{T}_{N+1}^{(3)\top} = \begin{bmatrix} \mathbf{0}_{N_1} & \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ 1 & \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \\ \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \end{bmatrix}, \\ \mathbf{S}_{N+1}^{(1)} &= \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} \\ \mathbf{O}_{N_2 N_1} & \mathbf{0}_{N_2} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{O}_{N_3 N_1} & \mathbf{0}_{N_3} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} \\ \mathbf{0}_{N_1}^T & 1 & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T \end{bmatrix}, \quad \mathbf{S}_{N+1}^{(1)\top} = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 1 \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} \end{bmatrix}, \\ \mathbf{S}_{N+1}^{(2)} &= \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{0}_{N_1} & \mathbf{O}_{N_1 N_3} \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{0}_{N_2} & \mathbf{O}_{N_2 N_3} \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{0}_{N_3} & \mathbf{I}_{N_3} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & 1 & \mathbf{0}_{N_3}^T \end{bmatrix}, \quad \mathbf{S}_{N+1}^{(2)\top} = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 1 \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} \end{bmatrix}, \\ \mathbf{S}_{N+1}^{(3)} &= \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 1 \end{bmatrix}, \quad \mathbf{S}_{N+1}^{(3)\top} = \begin{bmatrix} \mathbf{I}_{N_1} & \mathbf{O}_{N_1 N_2} & \mathbf{O}_{N_1 N_3} & \mathbf{0}_{N_1} \\ \mathbf{O}_{N_2 N_1} & \mathbf{I}_{N_2} & \mathbf{O}_{N_2 N_3} & \mathbf{0}_{N_2} \\ \mathbf{O}_{N_3 N_1} & \mathbf{O}_{N_3 N_2} & \mathbf{I}_{N_3} & \mathbf{0}_{N_3} \\ \mathbf{0}_{N_1}^T & \mathbf{0}_{N_2}^T & \mathbf{0}_{N_3}^T & 1 \end{bmatrix}. \end{aligned}$$



По аналогии, такие матрицы могут быть получены для любых значений M и N_m . Для этих матриц справедливы соотношения $\mathbf{T}_{N+1}^{T(m)} = \left[\mathbf{T}_{N+1}^{(m)} \right]^{-1}$ и $\mathbf{S}_{N+1}^{T(m)} = \left[\mathbf{S}_{N+1}^{(m)} \right]^{-1}$, а значит по аналогии с уравнениями (13.26) и (13.27) справедливы и соотношения

$$\mathbf{T}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} = \mathbf{I}_{N+1}, \quad (13.86)$$

$$\mathbf{S}_{N+1}^{(m)} \mathbf{S}_{N+1}^{(m)T} = \mathbf{I}_{N+1}. \quad (13.87)$$

Для выполнения последовательности вычислений векторов коэффициентов Калмана, обозначенных в начале настоящего раздела, необходимо определить M корреляционных матриц, каждую из которых можно представить как

$$\begin{aligned} \mathbf{R}_{N+1}^{(m)}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}^{(m)}(i) \mathbf{x}_{N+1}^{(m)H}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \left\{ \mathbf{T}_{N+1}^{(m)T} \begin{bmatrix} x_m(i) \\ \mathbf{x}_N^{(m)}(i) \end{bmatrix} \left[x_m^*(i), \mathbf{x}_N^{(m)H}(i) \right] \mathbf{T}_{N+1}^{(m)} \right\} = \\ &= \mathbf{T}_{N+1}^{(m)T} \left\{ \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} x_m(i) \\ \mathbf{x}_N^{(m)}(i) \end{bmatrix} \left[x_m^*(i), \mathbf{x}_N^{(m)H}(i) \right] \right\} \mathbf{T}_{N+1}^{(m)} = \\ &= \mathbf{T}_{N+1}^{(m)T} \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x_m(i) x_m^*(i) & \sum_{i=1}^k \lambda^{k-i} x_m(i) \mathbf{x}_N^{(m)H}(i) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m)}(i) x_m^*(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m)}(i) \mathbf{x}_N^{(m)H}(i) \end{bmatrix} \mathbf{T}_{N+1}^{(m)} = \\ &= \mathbf{T}_{N+1}^{(m)T} \begin{bmatrix} \mathbf{\sigma}_{f(m)2}^2(k) & \mathbf{r}_N^{f(m)H}(k) \\ \mathbf{r}_N^{f(m)}(k) & \mathbf{R}_N^{(m)}(k) \end{bmatrix} \mathbf{T}_{N+1}^{(m)} \end{aligned} \quad (13.88)$$

или как

$$\begin{aligned} \mathbf{R}_{N+1}^{(m)}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}^{(m)}(i) \mathbf{x}_{N+1}^{(m)H}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \left\{ \mathbf{S}_{N+1}^{(m)T} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(i) \\ x_m(i - N_m) \end{bmatrix} \left[\mathbf{x}_N^{(m-1)H}(i), x_m^*(i - N_m) \right] \mathbf{S}_{N+1}^{(m)} \right\} = \\ &= \mathbf{S}_{N+1}^{(m)T} \left\{ \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(i) \\ x_m(i - N_m) \end{bmatrix} \left[\mathbf{x}_N^{(m-1)H}(i), x_m^*(i - N_m) \right] \right\} \mathbf{S}_{N+1}^{(m)} = \\ &= \mathbf{S}_{N+1}^{(m)T} \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m-1)}(i) \mathbf{x}_N^{(m-1)H}(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m-1)}(i) x_m^*(i - N_m) \\ \sum_{i=1}^k \lambda^{k-i} x_m(i - N_m) \mathbf{x}_N^{(m-1)H}(i) & \sum_{i=1}^k \lambda^{k-i} x_m(i - N_m) x_m^*(i - N_m) \end{bmatrix} \mathbf{S}_{N+1}^{(m)} = \\ &= \mathbf{S}_{N+1}^{(m)T} \begin{bmatrix} \mathbf{R}_N^{(m-1)}(k) & \mathbf{r}_N^{b(m)}(k) \\ \mathbf{r}_N^{b(m)H}(k) & \mathbf{\sigma}_{b(m)2}^2(k) \end{bmatrix} \mathbf{S}_{N+1}^{(m)}. \end{aligned} \quad (13.89)$$

При рекурсивном вычислении векторов коэффициентов Калмана $\mathbf{g}_{N+1}^{(m)}(k)$ и $\mathbf{t}_{N+1}^{(m)}(k)$ также используется линейное предсказание, которое для многоканальных 1-SUBSD RLS-алгоритмов рассматривается в следующем разделе.

13.6. I-SUSD-вычисление векторов коэффициентов Калмана и отношений ошибок линейного предсказания

Апостериорная ошибка линейного предсказания вперед входного сигнала m -го канала I-SUSD адаптивного фильтра определяется как

$$e_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(k), \quad (13.90)$$

где $\mathbf{x}_N^{(m)}(k)$ определяется уравнением (13.81), а

$$\mathbf{h}_N^{f(m)}(k) = \left[\mathbf{h}_{N_1}^{f(m)T}(k), \mathbf{h}_{N_2}^{f(m)T}(k), \dots, \mathbf{h}_{N_M}^{f(m)T}(k) \right]^T \quad (13.91)$$

— вектор ВК фильтра линейного предсказания, в котором $\mathbf{h}_{N_m}^{f(m)}(k) = \left[h_1^{f(m)}(k), h_2^{f(m)}(k), \dots, h_{N_m}^{f(m)}(k) \right]^T$.

Аналогично апостериорная ошибка линейного предсказания назад входного сигнала m -го канала I-SUSD адаптивного фильтра определяется как

$$e_N^{b(m)}(k) = x_m(k - N_m) - \mathbf{h}_N^{b(m)H}(k) \mathbf{x}_N^{(m-1)}(k), \quad (13.92)$$

где $\mathbf{x}_N^{(m-1)}(k)$ также определяется уравнением (13.81), а

$$\mathbf{h}_N^{b(m)}(k) = \left[\mathbf{h}_{N_1}^{b(m)T}(k), \mathbf{h}_{N_2}^{b(m)T}(k), \dots, \mathbf{h}_{N_M}^{b(m)T}(k) \right]^T \quad (13.93)$$

— вектор ВК фильтра линейного предсказания, в котором $\mathbf{h}_{N_m}^{b(m)}(k) = \left[h_1^{b(m)}(k), h_2^{b(m)}(k), \dots, h_{N_m}^{b(m)}(k) \right]^T$.

Каждый из $m=1, 2, \dots, M$ векторов фильтров линейного предсказания вперед $\mathbf{h}_N^{f(m)}(k)$ и $\mathbf{h}_N^{b(m)}(k)$ находится в результате решения LS-задач, аналогичных рассмотренным в разделах 11.2 и 11.3.

Значение энергии ошибок линейного предсказания (13.90), определяемой как

$$\begin{aligned} E_N^{f(m)}(k) &= \sum_{i=1}^k \lambda^{k-i} e_N^{f(m)}(i) e_N^{f(m)*}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} \left[x_m(i) - \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(i) \right] \left[x_m(i) - \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(i) \right]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x_m(i) x_m^*(i) - \sum_{i=1}^k \lambda^{k-i} x_m(i) \mathbf{x}_N^{(m)H}(i) \mathbf{h}_N^{f(m)}(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(i) x_m^*(i) + \\ &\quad + \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(i) \mathbf{x}_N^{(m)H}(i) \mathbf{h}_N^{f(m)}(k) = \\ &= \sigma^{f(m)2}(k) - \mathbf{r}_N^{f(m)H}(k) \mathbf{h}_N^{f(m)}(k) - \mathbf{h}_N^{f(m)H}(k) \mathbf{r}_N^{f(m)}(k) + \\ &\quad + \mathbf{h}_N^{f(m)H}(k) \mathbf{R}_N^{(m)}(k) \mathbf{h}_N^{f(m)}(k), \end{aligned} \quad (13.94)$$

где

$$\sigma^{f(m)2}(k) = \sum_{i=1}^k \lambda^{k-i} x_m(i) x_m^*(i) = \lambda \sigma^{f(m)2}(k-1) + x_m(k) x_m^*(k), \quad (13.95)$$

$$\mathbf{r}_N^{f(m)}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m)}(i) x_m^*(i) = \lambda \mathbf{r}_N^{f(m)}(k-1) + \mathbf{x}_N^{(m)}(k) x_m^*(k), \quad (13.96)$$

$$\mathbf{R}_N^{(m)}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m)}(i) \mathbf{x}_N^{(m)H}(i) = \lambda \mathbf{R}_N^{(m)}(k-1) + \mathbf{x}_N^{(m)}(k) \mathbf{x}_N^{(m)H}(k), \quad (13.97)$$

минимально при

$$\nabla_{\mathbf{h}_N^{f(m)*}} \{E_N^{f(m)}(k)\} = \frac{\partial E_N^{f(m)}(k)}{\partial \mathbf{h}_N^{f(m)*}(k)} = -\mathbf{r}_N^{f(m)}(k) + \mathbf{R}_N^{(m)}(k) \mathbf{h}_N^{f(m)}(k) = \mathbf{0}_N, \quad (13.98)$$

откуда

$$\mathbf{h}_N^{f(m)}(k) = [\mathbf{R}_N^{(m)}(k)]^{-1} \mathbf{r}_N^{f(m)}(k) \quad (13.99)$$

для всех $m=1, 2, \dots, M$.

Аналогично, значение энергии ошибок линейного предсказания (13.92), определяемой как

$$\begin{aligned} E_N^{b(m)}(k) &= \sum_{i=1}^k \lambda^{k-i} e_N^{b(m)}(i) e_N^{b*}(i) = \\ &= \sum_{i=1}^k \lambda^{k-i} [x_m(i-N_m) - \mathbf{h}_N^{b(m)H}(k) \mathbf{x}_N^{(m-1)}(i)] [x_m(i-N_m) - \mathbf{h}_N^{b(m)H}(k) \mathbf{x}_N^{(m-1)}(i)]^* = \\ &= \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) x_m(i-N_m)^* - \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) \mathbf{x}_N^{(m-1)H}(i) \mathbf{h}_N^{b(m)}(k) - \\ &\quad - \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{b(m)H}(k) \mathbf{x}_N^{(m-1)}(i) x^*(i-N_m) + \sum_{i=1}^k \lambda^{k-i} \mathbf{h}_N^{bH}(k) \mathbf{x}_N^{(m-1)}(i) \mathbf{x}_N^{(m-1)H}(i) \mathbf{h}_N^b(k) = \\ &= \sigma^{b(m)2}(k) - \mathbf{r}_N^{b(m)H}(k) \mathbf{h}_N^{b(m)}(k) - \mathbf{h}_N^{b(m)H}(k) \mathbf{r}_N^{b(m)} + \mathbf{h}_N^{b(m)H}(k) \mathbf{R}_N^{(m-1)}(k) \mathbf{h}_N^{b(m)}(k), \end{aligned} \quad (13.100)$$

где

$$\sigma^{b(m)2}(k) = \sum_{i=1}^k \lambda^{k-i} x_m(i-N_m) x_m^*(i-N_m) = \lambda \sigma^{b(m)2}(k-1) + x_m(k-N_m) x_m^*(k-N_m), \quad (13.101)$$

$$\mathbf{r}_N^{b(m)}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m-1)}(i) x_m^*(i-N_m) = \lambda \mathbf{r}_N^{b(m)}(k-1) + \mathbf{x}_N^{(m-1)}(k) x_m^*(k-N_m), \quad (13.102)$$

$$\mathbf{R}_N^{(m-1)}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N^{(m-1)}(i) \mathbf{x}_N^{(m-1)H}(i) = \lambda \mathbf{R}_N^{(m-1)}(k-1) + \mathbf{x}_N^{(m-1)}(k) \mathbf{x}_N^{(m-1)H}(k), \quad (13.103)$$

минимально при

$$\nabla_{\mathbf{h}_N^{b(m)*}} \{E_N^{b(m)}(k)\} = \frac{\partial E_N^{b(m)}(k)}{\partial \mathbf{h}_N^{b(m)*}(k)} = -\mathbf{r}_N^{b(m)}(k) + \mathbf{R}_N^{(m-1)}(k) \mathbf{h}_N^{b(m)}(k) = \mathbf{0}_N. \quad (13.104)$$

Отсюда

$$\mathbf{h}_N^{b(m)}(k) = \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{r}_N^{b(m)}(k) \quad (13.105)$$

для всех $m=1, 2, \dots, M$.

Далее, следуя уравнениям (11.12)–(11.17) и (11.37)–(11.40), векторы ВК фильтров линейного предсказания (13.99) и (13.105) входных сигналов многоканального 1-SUBSD фильтра можно представить в виде рекуррентных соотношений

$$\begin{aligned} \mathbf{h}_N^{f(m)}(k) &= \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{r}_N^{f(m)}(k) = \\ &= \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \left[\lambda \mathbf{r}_N^{f(m)}(k-1) + \mathbf{x}_N^{(m)}(k) \mathbf{x}_m^*(k) \right] = \\ &= \lambda \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{r}_N^{f(m)}(k-1) + \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{x}_N^{(m)}(k) \mathbf{x}_m^*(k) = \\ &= \lambda \lambda^{-1} \left\{ \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} - \mathbf{g}_N^{(m)}(k) \mathbf{x}_N^{(m)}(k) \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \right\} \mathbf{r}_N^{f(m)}(k-1) + \\ &+ \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{x}_N^{(m)}(k) \mathbf{x}_m^*(k) = \\ &= \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \mathbf{r}_N^{f(m)}(k-1) - \mathbf{g}_N^{(m)}(k) \mathbf{x}_N^{(m)}(k) \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \mathbf{r}_N^{f(m)}(k-1) + \\ &+ \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{x}_N^{(m)}(k) \mathbf{x}_m^*(k) = \\ &= \mathbf{h}_N^{f(m)}(k-1) - \mathbf{g}_N^{(m)}(k) \mathbf{x}_N^{(m)H}(k) \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \mathbf{x}_m^*(k) = \\ &= \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \left[\mathbf{x}_m^*(k) - \mathbf{x}_N^{(m)H}(k) \mathbf{h}_N^{f(m)}(k-1) \right] = \\ &= \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{f(m)*}(k) \end{aligned} \quad (13.106)$$

И

$$\begin{aligned} \mathbf{h}_N^{b(m)}(k) &= \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{r}_N^{b(m)}(k) = \\ &= \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \left[\lambda \mathbf{r}_N^{b(m)}(k-1) + \mathbf{x}_N^{(m-1)}(k) \mathbf{x}_m^*(k-N_m) \right] = \\ &= \lambda \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{r}_N^{b(m)}(k-1) + \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) \mathbf{x}_m^*(k-N_m) = \\ &= \lambda \lambda^{-1} \left\{ \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} - \mathbf{g}_N^{(m-1)}(k) \mathbf{x}_N^{(m-1)}(k) \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} \right\} \mathbf{r}_N^{b(m)}(k-1) + \\ &+ \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) \mathbf{x}_m^*(k-N_m) = \\ &= \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} \mathbf{r}_N^{b(m)}(k-1) - \mathbf{g}_N^{(m-1)}(k) \mathbf{x}_N^{(m-1)}(k) \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} \mathbf{r}_N^{b(m)}(k-1) + \\ &+ \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) \mathbf{x}_m^*(k-N_m) = \\ &= \mathbf{h}_N^{b(m)}(k-1) - \mathbf{g}_N^{(m-1)}(k) \mathbf{x}_N^{(m-1)H}(k) \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \mathbf{x}_m^*(k-N_m) = \\ &= \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \left[\mathbf{x}_m^*(k-N_m) - \mathbf{x}_N^{(m-1)H}(k) \mathbf{h}_N^{b(m)}(k-1) \right] = \\ &= \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k), \end{aligned} \quad (13.107)$$

ГДЕ

$$\mathbf{g}_N^{(m)}(k) = \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{x}_N^{(m)}(k) \quad (13.108)$$

И

$$\mathbf{g}_N^{(m-1)}(k) = \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) \quad (13.109)$$

— векторы коэффициентов Калмана многоканального 1-SUBSD адаптивного фильтра.

Для быстрого вычисления векторов коэффициентов Калмана в многоканальных 1-SUSD RLS-алгоритмах требуется определить выражения для вычисления переменных $E_{N,\min}^{f(m)}(k)$, $E_{N,\min}^{b(m)}(k)$, а также выполнить обращение матриц (13.88) и (13.89).

При подстановке уравнений (13.99) и (13.105) в уравнения (13.94) и (13.100), а также используя преобразования, аналогичные (11.18) и (11.41), получим значения энергий ошибок линейного предсказания

$$E_{N,\min}^{f(m)}(k) = \sigma^{f(m)2}(k) - \mathbf{r}_N^{f(m)H}(k) \mathbf{h}_N^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + e_N^{f(m)}(k) \alpha_N^{f(m)*}(k) \quad (13.110)$$

и

$$E_{N,\min}^{b(m)}(k) = \sigma^{b(m)2}(k) - \mathbf{r}_N^{b(m)H}(k) \mathbf{h}_N^{b(m)}(k) = \lambda E_{N,\min}^{b(m)}(k-1) + e_N^{b(m)}(k) \alpha_N^{b(m)*}(k). \quad (13.111)$$

Кроме того, аналогично $E_{N,\min}^{f(m)}(k)$, см. уравнение (6.87), также справедливы соотношения

$$E_{N,\min}^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + \alpha_N^{f(m)}(k) e_N^{f(m)*}(k) \quad (13.112)$$

и

$$E_{N,\min}^{b(m)}(k) = \lambda E_{N,\min}^{b(m)}(k-1) + \alpha_N^{b(m)}(k) e_N^{b(m)*}(k). \quad (13.113)$$

Введя обозначение для матрицы векторов $\mathbf{a}_{N+1}^{(m)}(k)$ ВК фильтра линейного предсказания вперед

$$\mathbf{a}_{N+1}^{(m)}(k) = \left[1, -\mathbf{h}_N^{f(m)T}(k) \right]^T \quad (13.114)$$

и используя уравнения (13.83) и (13.106), векторы априорных и апостериорных ошибок линейного предсказания можно представить как

$$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)H}(k-1) \mathbf{x}_N^{(m)}(k) = \mathbf{a}_{N+1}^{(m)H}(k-1) \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) \quad (13.115)$$

и

$$e_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)H}(k) \mathbf{x}_N^{(m)}(k) = \mathbf{a}_{N+1}^{(m)H}(k) \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k), \quad (13.116)$$

а уравнение (13.114) — как

$$\mathbf{a}_{N+1}^{(m)}(k) = \mathbf{a}_{N+1}^{(m)}(k-1) + \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} \alpha_N^{f(m)*}(k). \quad (13.117)$$

Аналогичные обозначения можно ввести для матриц векторов $\mathbf{b}_{N+1}^{(m)}(k)$ ВК фильтра линейного предсказания назад

$$\mathbf{b}_{N+1}^{(m)}(k) = \left[-\mathbf{h}_N^{b(m)T}(k), 1 \right]^T. \quad (13.118)$$

Тогда используя уравнения (3.84) и (13.107), векторы априорных и апостериорных ошибок линейного предсказания можно представить как

$$\alpha_N^{b(m)}(k) = x(k - N_m) - \mathbf{h}_N^{b(m)H}(k-1) \mathbf{x}_N^{(m-1)}(k) = \mathbf{b}_{N+1}^{(m)H}(k-1) \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k), \quad (13.119)$$

$$e_N^{b(m)} = x(k - N_m) - \mathbf{h}_N^{b(m)H}(k) \mathbf{x}_N^{(m-1)}(k) = \mathbf{b}_{N+1}^{(m)H}(k) \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k), \quad (13.120)$$

а уравнение (13.118) — как

$$\mathbf{b}_{N+1}^{(m)}(k) = \mathbf{b}_{N+1}^{(m)}(k-1) + \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} \alpha_N^{b(m)*}(k). \quad (13.121)$$

Обращение матрицы (13.88) можно осуществить, используя лемму об обращении клеточных матриц (11.51) и преобразования, аналогичные (11.53)–(11.58). Тогда

$$\begin{aligned} & \left[\mathbf{R}_{N+1}^{(m)}(k) \right]^{-1} = \\ &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[E_{N,\min}^{f(m)}(k) \right]^{-1} & -\left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{h}_N^{f(m)H}(k) \\ -\mathbf{h}_N^{f(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} & \mathbf{h}_N^f(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{h}_N^{f(m)H}(k) + \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} \right\} \mathbf{T}_{N+1}^{(m)} = \\ &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k) \end{bmatrix} \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \begin{bmatrix} 1, & -\mathbf{h}_N^{f(m)H}(k) \end{bmatrix} \right\} \mathbf{T}_{N+1}^{(m)} = \\ &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \right\} \mathbf{T}_{N+1}^{(m)}, \end{aligned} \quad (13.122)$$

Таким же образом, используя лемму (11.52) и преобразования, аналогичные (11.59)–(11.63), можно осуществить обращение матрицы (13.89) как

$$\begin{aligned} & \left[\mathbf{R}_{N+1}^{(m)}(k) \right]^{-1} = \\ &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} + \mathbf{h}_N^{b(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{h}_N^{b(m)H}(k) & -\mathbf{h}_N^{b(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \\ -\left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{h}_N^{b(m)H}(k) & \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \end{bmatrix} \right\} \mathbf{S}_{N+1}^{(m)} = \\ &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{h}_N^{b(m)}(k) \\ 1 \end{bmatrix} \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \begin{bmatrix} -\mathbf{h}_N^{b(m)H}(k), 1 \end{bmatrix} \right\} \mathbf{S}_{N+1}^{(m)} = \\ &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k) \right\} \mathbf{S}_{N+1}^{(m)}. \end{aligned} \quad (13.123)$$

Теперь, используя уравнения (13.122), (13.123), (13.85), (13.116) и (13.120), можно определить два выражения для вычисления вектора апостериорных коэффициентов Калмана $\mathbf{g}_{N+1}^{(m)}(k) = \left[\mathbf{R}_{N+1}^{(m)}(k) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k)$ как

$$\begin{aligned}
 \mathbf{g}_{N+1}^{(m)}(k) &= \left[\mathbf{R}_{N+1}^{(m)}(k) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \right\} \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \right\} \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
 &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \end{bmatrix} \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} + \right. \\
 &\quad \left. + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} \right\} = \\
 &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \left[\mathbf{R}_N^{(m)}(k) \right]^{-1} \mathbf{x}_N^{(m)}(k) & 0 \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} \right\} = \\
 &= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\}
 \end{aligned} \tag{13.124}$$

И

$$\begin{aligned}
 \mathbf{g}_{N+1}^{(m)}(k) &= \left[\mathbf{R}_{N+1}^{(m)}(k) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k) \right\} \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k) \right\} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} = \\
 &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} + \right. \\
 &\quad \left. + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k) \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} \right\} = \\
 &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k) \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} \right\} = \\
 &= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\}.
 \end{aligned} \tag{13.125}$$

Векторы апостериорных коэффициентов Калмана $\mathbf{t}_{N+1}^{(m)}(k) =$
 $= \left[\mathbf{R}_{N+1}^{(m)}(k-1) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k)$ определяются аналогично как

$$\begin{aligned}
\mathbf{t}_{N+1}^{(m)}(k) &= \left[\mathbf{R}_{N+1}^{(m)}(k-1) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k) = \\
&= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k-1) \right\} \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
&= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k-1) \right\} \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
&= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \end{bmatrix} \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} + \right. \\
&\quad \left. + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} \right\} = \\
&= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \left[\mathbf{R}_N^{(m)}(k-1) \right]^{-1} \mathbf{x}_N^{(m)}(k) & \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k-1) \right]^{-1} \mathbf{a}_{N+1}^{(m)H}(k) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} \right\} = \\
&= \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^f(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\}
\end{aligned} \tag{13.126}$$

И

$$\begin{aligned}
\mathbf{t}_{N+1}^{(m)}(k) &= \left[\mathbf{R}_{N+1}^{(m)}(k-1) \right]^{-1} \mathbf{x}_{N+1}^{(m)}(k) = \\
&= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k-1) \right\} \times \\
&\times \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
&= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k-1) \right\} \times \\
&\times \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k-N_m) \end{bmatrix} = \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k-N_m) \end{bmatrix} + \right. \\
&\quad \left. + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k-1) \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k-N_m) \end{bmatrix} \right\} = \\
&= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \left[\mathbf{R}_N^{(m-1)}(k-1) \right]^{-1} \mathbf{x}_N^{(m-1)}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \right. \\
&\quad \left. + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \mathbf{b}_{N+1}^{(m)H}(k-1) \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k-N_m) \end{bmatrix} \right\} = \\
&= \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\}.
\end{aligned} \tag{13.127}$$

Соотношения между векторами $\mathbf{t}_N^{(m)}(k)$ и $\mathbf{g}_N^{(m)}(k)$ многоканального 1-SUSD адаптивного фильтра аналогичны соотношениям в одноканальных адаптивных фильтрах (11.80) и (11.81), т. е. $\mathbf{t}_N^{(m)}(k) = \lambda \mathbf{g}_N^{(m)}(k) [\varphi_N^{(m)}(k)]^{-1}$ и $\mathbf{g}_N^{(m)}(k) = \lambda^{-1} \mathbf{t}_N^{(m)}(k) \varphi_N^{(m)}(k)$, а отношения ошибок $\varphi_{N+1}^{(m)}(k)$ вычисляются аналогично выражениям (11.83) и (11.84) при использовании (13.83), (13.84), (13.124) и (13.125).

Согласно уравнению (11.83)

$$\begin{aligned}
 \varphi_{N+1}^{(m)}(k) &= 1 - \mathbf{g}_{N+1}^{(m)H}(k) \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left(\mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\} \right)^H \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\}^H \left(\mathbf{T}_{N+1}^{(m)T} \right)^T \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\}^H \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\}^H \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
 &= 1 - \left[0, \mathbf{g}_N^{(m)H}(k) \right] \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} - \left\{ \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) \right\}^H \mathbf{a}_{N+1}^{(m)H}(k) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
 &= 1 - \mathbf{g}_N^{(m)H}(k) \mathbf{x}_N^{(m)}(k) - \mathbf{e}_N^{f(m)*}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k) = \\
 &= \varphi_N^{(m)}(k) - \mathbf{e}_N^{f(m)*}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} \mathbf{e}_N^{f(m)}(k)
 \end{aligned} \tag{13.128}$$

и согласно уравнению (11.84)

$$\begin{aligned}
 \varphi_{N+1}^{(m)}(k) &= 1 - \mathbf{g}_{N+1}^{(m)H}(k) \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left(\mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\} \right)^H \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\}^H \left(\mathbf{S}_{N+1}^{(m)T} \right)^T \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\}^H \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 - \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\}^H \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} = \\
 &= 1 - \left[\mathbf{g}_N^{(m-1)H}(k), 0 \right] \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} - \left\{ \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) \right\}^H \mathbf{b}_{N+1}^{(m)}(k) \begin{bmatrix} \mathbf{x}_N(k) \\ \mathbf{x}_M(k - N) \end{bmatrix} = \\
 &= 1 - \mathbf{g}_N^{(m-1)H}(k) \mathbf{x}_N^{(m-1)}(k) - \mathbf{e}_N^{b(m)*}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k) = \\
 &= \varphi_N^{(m-1)}(k) - \mathbf{e}_N^{b(m)*}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} \mathbf{e}_N^{b(m)}(k).
 \end{aligned} \tag{13.129}$$

Обратное отношение ошибок $\left[\varphi_{N+1}^{(m)}(k)\right]^{-1}$ вычисляется согласно уравнениям (11.89) и (11.90) как

$$\begin{aligned}
 \left[\varphi_{N+1}^{(m)}(k)\right]^{-1} &= 1 + \lambda^{-1} \mathbf{t}_{N+1}^{(m)H}(k) \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left(\mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\} \right)^H \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\}^H \left(\mathbf{T}_{N+1}^{(m)T} \right)^T \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\}^H \mathbf{T}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\}^H \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \left[0, \mathbf{t}_N^{(m)H}(k) \right] \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} + \lambda^{-1} \left\{ \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) \right\}^H \times \\
 &\quad \times \mathbf{a}_{N+1}^{(m)H}(k-1) \begin{bmatrix} x_m(k) \\ \mathbf{x}_N^{(m)}(k) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \mathbf{t}_N^{(m)H}(k) \mathbf{x}_N^{(m)}(k) + \lambda^{-1} \alpha_N^{f(m)*}(k) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k) = \\
 &= \left[\varphi_N^{(m)}(k) \right]^{-1} + \lambda^{-1} \alpha_N^{f(m)*}(k) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-1} \alpha_N^{f(m)}(k)
 \end{aligned} \tag{13.130}$$

и как

$$\begin{aligned}
 \left[\varphi_{N+1}^{(m)}(k)\right]^{-1} &= 1 + \lambda^{-1} \mathbf{t}_{N+1}^{(m)H}(k) \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left(\mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\} \right)^H \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\}^H \left(\mathbf{S}_{N+1}^{(m)T} \right)^T \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\}^H \mathbf{S}_{N+1}^{(m)} \mathbf{x}_{N+1}^{(m)}(k) = \\
 &= 1 + \lambda^{-1} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\}^H \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} = \\
 &= 1 + \lambda^{-1} \left[\mathbf{t}_N^{(m-1)H}(k), 0 \right] \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} + \lambda^{-1} \left\{ \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\}^H \times \\
 &\quad \times \mathbf{b}_{N+1}^{(m)H}(k-1) \begin{bmatrix} \mathbf{x}_N^{(m-1)}(k) \\ x_m(k - N_m) \end{bmatrix} = 1 + \lambda^{-1} \mathbf{t}_N^{(m-1)H}(k) \mathbf{x}_N^{(m-1)}(k) + \lambda^{-1} \alpha_N^{b(m)*}(k) \times \\
 &\quad \times \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) = \left[\varphi_N^{(m-1)}(k) \right]^{-1} + \lambda^{-1} \alpha_N^{b(m)*}(k) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k).
 \end{aligned} \tag{13.131}$$

13.7. I-SUSD быстрые RLS-алгоритмы

Соотношения, приведенные в разделах 13.5 и 13.6, используются при получении вычислительных процедур многоканальных I-SUSD быстрых RLS-алгоритмов. При разработке многоканального FK-алгоритма в уравнениях (13.124) и (13.125) необходимо выделить сомножители, заключенные в фигурные скобки, для чего можно воспользоваться соотношениями (13.86) и (13.87). Тогда получим, что

$$\begin{aligned} \mathbf{T}_{N+1}^{(m)} \mathbf{g}_{N+1}^{(m)}(k) &= \mathbf{T}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \left\{ \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k) \right]^{-1} e_N^{f(m)}(k) \right\} = \\ &= \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \mathbf{a}_{N+1}^{(m)}(k) \left[E_{N,\min}^f(k) \right]^{-1} e_N^{f(m)}(k) \end{aligned} \quad (13.132)$$

И

$$\begin{aligned} \mathbf{S}_{N+1}^{(m)} \mathbf{g}_{N+1}^{(m)}(k) &= \mathbf{S}_{N+1}^{(m)} \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) \right\} = \\ &= \begin{bmatrix} \mathbf{g}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}. \end{aligned} \quad (13.133)$$

Так как векторы $\mathbf{g}_{N+1}^{(m)}(k)$, вычисленные по формулам (13.124) и (13.125), равны между собой, то, вычислив $\mathbf{g}_{N+1}^{(m)}(k)$ по формуле (13.124), а затем выполнив перемножение $\mathbf{S}_{N+1}^{(m)} \mathbf{g}_{N+1}^{(m)}(k)$, можно из уравнения (13.133) определить

$$\tilde{q}_N^{(m)}(k) = \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) \quad (13.134)$$

И

$$\begin{aligned} \tilde{\mathbf{q}}_N^{(m)}(k) &= \mathbf{g}_N^{(m-1)}(k) - \mathbf{h}_N^{b(m)}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) = \\ &= \mathbf{g}_N^{(m-1)}(k) - \mathbf{h}_N^{b(m)}(k) \tilde{q}_N^{(m)}(k) = \\ &= \mathbf{g}_N^{(m-1)}(k) - \left[\mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) \right] \tilde{q}_N^{(m)}(k) = \\ &= \mathbf{g}_N^{(m-1)}(k) - \mathbf{h}_N^{b(m)}(k-1) \tilde{q}_N^{(m)}(k) - \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k) = \\ &= \mathbf{g}_N^{(m-1)}(k) \left[1 - \alpha_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k) \right] - \mathbf{h}_N^{b(m)}(k-1) \tilde{q}_N^{(m)}(k). \end{aligned} \quad (13.135)$$

В (13.135) также использовано уравнение (13.107) для вектора $\mathbf{h}_N^{b(m)}(k)$. Из (13.135) можно определить

$$\mathbf{g}_N^{(m-1)}(k) = \left[\tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) \tilde{q}_N^{(m)}(k) \right] \left[1 - \alpha_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k) \right]^{-1}. \quad (13.136)$$

Таким образом, уравнения (13.124), (13.133) и (13.135) позволяют рекурсивно вычислять вектор коэффициентов Калмана $\mathbf{g}_N^{(m-1)}(k)$ при наличии значений других переменных, входящих в эти уравнения. За M шагов определяется вектор $\mathbf{g}_N^{(0)}(k) = \mathbf{g}_N(k) = \mathbf{g}_N^{(M)}(k+1)$.

Вычислительная процедура многоканального I-SUSD FK-алгоритма на основе вычислений (13.124), (13.133) и (13.136) приведена в табл. 13.4.

Таблица 13.4. Многоканальный 1-SUSD FK-алгоритм

Вычисления	Ссылки
Initialization: $E_N^{f(m)}(0) = \delta^2$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$, $\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$,	(13.4.0)
create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$, $m = 1, \dots, M$, $\mathbf{g}_N^{(M)}(1) = \mathbf{0}_N$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(13.4.1)
$\alpha_N^{b(m)}(k) = x_m(k - N_m) - \mathbf{h}_N^{b(m)\top}(k-1) \mathbf{x}_N^{(m-1)}(k)$	(13.4.2)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{f(m)*}(k)$	(13.4.3)
$\mathbf{e}_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(13.4.4)
$E_{N,\min}^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + \mathbf{e}_N^{f(m)}(k) \alpha_N^{f(m)*}(k)$	(13.4.5)
$\tilde{\mathbf{g}}_{N+1}^{(m)}(k) = \begin{bmatrix} 0 \\ \mathbf{g}_N^{(m)}(k) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k) \end{bmatrix} \frac{\mathbf{e}_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k)}$	(13.4.6)
$\tilde{\mathbf{g}}_{N+1}^{(m)}(k) = \mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top} \tilde{\mathbf{g}}_{N+1}^{(m)}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}$	(13.4.7)
$\mathbf{g}_N^{(m-1)}(k) = \frac{\tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{b(m)\top}(k-1) \tilde{q}_N^{(m)}(k)}{1 - \alpha_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k)}$	(13.4.8)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k)$	(13.4.9)
End for m	
$\alpha_N(k) = d(k) - \mathbf{h}_N^{H}(k-1) \mathbf{x}_N^{(0)}(k)$	(13.4.10)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N^{(0)}(k) \alpha_N^{*}(k)$	(13.4.11)
$\mathbf{g}_N^{(M)}(k+1) = \mathbf{g}_N^{(0)}(k)$	(13.4.12)
End for k	

При выполнении алгоритма (см. табл. 13.4) и других 1-SUSD RLS-алгоритмов необходимо обеспечить формирование скалярных переменных и векторов $x_m(k)$, $\mathbf{x}_N^{(m)}(k)$, $x_m(k - N_m)$ и $\mathbf{x}_N^{(m-1)}(k)$. Это можно, например, обеспечить путем формирования векторов (13.82) и выполнения над ними операций (13.83) и (13.84). Особенности инициализации переменных $E_{N,\min}^{f(m)}(0)$ в рассматриваемом алгоритме, а также переменных $E_{N,\min}^{b(m)}(0)$ будут рассмотрены в разделе 14.8.

Вычислительная сложность 1-SUSD FK-алгоритма равна $7MN + 3N$ операциям умножения, $7MN + 2N$ операциям сложения и $2M$ операциям деления, т. е. $O(NM)$.

Рекурсивное вычисление векторов $\mathbf{t}_{N+1}^{(m)}(k)$ в многоканальных 1-SUSD FTF- и FAEST-алгоритмах и $\mathbf{t}_N^{(0)}(k) = \mathbf{t}_N(k) = \mathbf{t}_N^{(M)}(k+1)$ осуществляется посредством соотношений (13.126) и (13.127). Для этого сначала с помощью уравнения (13.126) вычисляется вектор $\mathbf{t}_{N+1}^{(m)}(k)$, а затем, используя соотношение (13.87), этот вектор

умножается на матрицу $\mathbf{S}_{N+1}^{(m)}$, чтобы результат можно было представить в виде двухэлементного вектора

$$\begin{aligned} \mathbf{S}_{N+1}^{(m)} \mathbf{t}_{N+1}^{(m)}(k) &= \mathbf{S}_{N+1}^{(m)} \mathbf{S}_{N+1}^{(m)T} \left\{ \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) \right\} = \\ &= \begin{bmatrix} \mathbf{t}_N^{(m-1)}(k) \\ 0 \end{bmatrix} + \mathbf{b}_{N+1}^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}. \end{aligned} \quad (13.137)$$

Из уравнения (13.137) следует, что

$$\tilde{q}_N^{(m)}(k) = \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k), \quad (13.138)$$

$$\begin{aligned} \tilde{\mathbf{q}}_N^{(m)}(k) &= \mathbf{t}_N^{(m-1)}(k) - \mathbf{h}_N^{(m)}(k-1) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k) = \\ &= \mathbf{t}_N^{(m-1)}(k) - \mathbf{h}_N^{(m)}(k-1) \tilde{q}_N^{(m)}(k), \end{aligned} \quad (13.139)$$

а

$$\mathbf{t}_N^{(m-1)}(k) = \tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{(m)}(k-1) \tilde{q}_N^{(m)}(k). \quad (13.140)$$

Таким образом, уравнения (13.126), (13.137)–(13.140) позволяют за M шагов рекурсивно вычислить вектор коэффициентов Калмана $\mathbf{t}_N^{(0)}(k) = \mathbf{t}_N(k) = \mathbf{t}_N^{(M)}(k+1)$. Значения $\alpha_N^{b(m)}(k)$ определяются из уравнения (13.138) как

$$\alpha_N^{b(m)}(k) = E_{N,\min}^{b(m)}(k-1) \tilde{q}_N^{(m)}(k). \quad (13.141)$$

Соотношения между априорными и апостериорными ошибками в многоканальных 1-SUSD быстрых RLS-алгоритмах определяются по аналогии с уравнениями (11.77)–(11.79) как

$$e_N(k) = \alpha_N(k) \varphi_N^{(0)}(k), \quad (13.142)$$

$$e_N^{f(m)}(k) = \alpha_N^{f(m)}(k) \varphi_N^{(m)}(k), \quad (13.143)$$

$$e_N^{b(m)}(k) = \alpha_N^{b(m)}(k) \varphi_N^{(m-1)}(k). \quad (13.144)$$

Для рекурсивного вычисления $\varphi_N^{(m-1)}(k)$ уравнения (13.128) и (13.129), с учетом уравнений (13.143) и (13.144), можно преобразовать как

$$\begin{aligned} \varphi_{N+1}^{(m)}(k) &= \varphi_N^{(m)}(k) - e_N^{f(m)H}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} e_N^{f(m)}(k) = \\ &= \varphi_N^{(m)}(k) - \varphi_N^{(m)}(k) \alpha_N^{f(m)*}(k) \left[E_{N,\min}^{f(m)}(k) \right]^{-1} e_N^{f(m)}(k) = \\ &= \frac{\varphi_N^{(m)}(k) E_{N,\min}^{f(m)}(k) - \varphi_N^{(m)}(k) \alpha_N^{f(m)*}(k) e_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k)} = \\ &= \frac{\varphi_N^{(m)}(k) \left[E_{N,\min}^{f(m)}(k) - \alpha_N^{f(m)*}(k) e_N^{f(m)}(k) \right]}{E_{N,\min}^{f(m)}(k)} = \varphi_N^{(m)}(k) \frac{\lambda E_{N,\min}^{f(m)}(k-1)}{E_{N,\min}^{f(m)}(k)} \end{aligned} \quad (13.145)$$

и

$$\begin{aligned}
\varphi_{N+1}^{(m)}(k) &= \varphi_N^{(m-1)}(k) - e_N^{b(m)*}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) = \\
&= \varphi_N^{(m-1)}(k) - \varphi_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) \left[E_{N,\min}^{b(m)}(k) \right]^{-1} e_N^{b(m)}(k) = \\
&= \frac{\varphi_N^{(m-1)}(k) E_{N,\min}^{b(m)}(k) - \varphi_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) e_N^{b(m)}(k)}{E_{N,\min}^{b(m)}(k)} = \\
&= \frac{\varphi_N^{(m-1)}(k) \left[E_{N,\min}^{b(m)}(k) - \alpha_N^{b(m)*}(k) e_N^{b(m)}(k) \right]}{E_{N,\min}^{b(m)}(k)} = \varphi_N^{(m-1)}(k) \frac{\lambda E_{N,\min}^{b(m)}(k-1)}{E_{N,\min}^{b(m)}(k)}.
\end{aligned} \tag{13.146}$$

Из уравнения (13.146), с учетом (13.111) и (13.138) и подобно (11.107), можно определить $\varphi_N^{(m-1)}(k)$ как

$$\begin{aligned}
\varphi_N^{(m-1)}(k) &= \varphi_{N+1}^{(m)}(k) \frac{E_{N,\min}^{b(m)}(k)}{\lambda E_{N,\min}^{b(m)}(k-1)} = \\
&= \varphi_{N+1}^{(m)}(k) \frac{\left[\lambda E_{N,\min}^{b(m)}(k-1) + e_N^{b(m)}(k) \alpha_N^{b(m)*}(k) \right]}{\lambda E_{N,\min}^{b(m)}(k-1)} = \\
&= \varphi_{N+1}^{(m)}(k) + \varphi_{N+1}^{(m)}(k) \frac{e_N^{b(m)}(k) \tilde{q}_N^{(m)*}(k)}{\lambda}
\end{aligned} \tag{13.147}$$

или

$$\begin{aligned}
\lambda \varphi_N^{(m-1)}(k) &= \lambda \varphi_{N+1}^{(m)}(k) + \varphi_{N+1}^{(m)}(k) e_N^{b(m)}(k) \tilde{q}_N^{(m)*}(k) = \\
&= \lambda \varphi_{N+1}^{(m)}(k) + \varphi_{N+1}^{(m)}(k) \alpha_N^{b(m)}(k) \varphi_N^{(m-1)}(k) \tilde{q}_N^{(m)*}(k).
\end{aligned} \tag{13.148}$$

Из последнего уравнения получаем

$$\begin{aligned}
\varphi_N^{(m-1)}(k) &= \lambda \varphi_{N+1}^{(m)}(k) \lambda^{-1} \left[1 - \lambda^{-1} \varphi_{N+1}^{(m)}(k) \tilde{q}_N^{(m)*}(k) \alpha_N^{b(m)}(k) \right]^{-1} = \\
&= \varphi_{N+1}^{(m)}(k) \left[1 - \lambda^{-1} \varphi_{N+1}^{(m)}(k) \tilde{q}_N^{(m)*}(k) \alpha_N^{b(m)}(k) \right]^{-1}.
\end{aligned} \tag{13.149}$$

Таким образом, уравнения (13.146) и (13.149) позволяют рекурсивно вычислять переменную $\varphi_N^{(m-1)}(k)$.

С учетом соотношений $\mathbf{g}_N^{(m)}(k) = \lambda^{-1} \mathbf{t}_N^{(m)}(k) \varphi_N^{(m)}(k)$, $\mathbf{g}_N^{(m-1)}(k) = \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) \varphi_N^{(m-1)}(k)$ и (13.142)–(13.144) уравнения для вычисления векторов ВК фильтров линейного предсказания и ВК собственно адаптивного фильтра преобразуются как

$$\begin{aligned}
\mathbf{h}_N^{f(m)}(k) &= \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{f(m)*}(k) = \mathbf{h}_N^{f(m)}(k-1) + \\
&+ \lambda^{-1} \mathbf{t}_N^{(m)}(k) \varphi_N^{(m)}(k) \alpha_N^{f(m)*}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m)}(k) e_N^{f(m)*}(k),
\end{aligned} \tag{13.150}$$

$$\begin{aligned}
\mathbf{h}_N^{b(m)}(k) &= \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) = \mathbf{h}_N^{b(m)}(k-1) + \\
&+ \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) \varphi_N^{(m-1)}(k) \alpha_N^{b(m)*}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) e_N^{b(m)*}(k),
\end{aligned} \tag{13.151}$$

$$\begin{aligned}
\mathbf{h}_N(k) &= \mathbf{h}_N(k-1) + \mathbf{g}_N^{(0)}(k) \alpha_N^{*}(k) = \mathbf{h}_N(k-1) + \\
&+ \lambda^{-1} \mathbf{t}_N^{(0)}(k) \varphi_N^{(0)}(k) \alpha_N^{*}(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N^{(0)}(k) e_N^{*}(k).
\end{aligned} \tag{13.152}$$

Полученные выражения определяют вычислительную процедуру многоканального 1-SUSD FTF-алгоритма, которая приведена в табл. 13.5.

Таблица 13.5. Многоканальный 1-SUSD FTF-алгоритм

Вычисления	Ссылки
Initialization: $E_N^{f(m)}(0) = \delta^2$, $E_N^{b(m)}(0) = \delta^2 \lambda^{-N_m}$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$,	(13.5.0)
$\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$, $m = 1, \dots, M$, $\mathbf{t}_N^{(M)}(1) = \mathbf{0}_N$, $\varphi_N^{(M)}(1) = 1$,	
$\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(13.5.1)
$e_N^{f(m)}(k) = \alpha_N^{f(m)}(k) \varphi_N^{(m)}(k)$	(13.5.2)
$\bar{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k-1) \end{bmatrix} \frac{\alpha_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k-1)}$	(13.5.3)
$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top} \bar{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} \mathbf{q}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}$	(13.5.4)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m)}(k) e_N^{f(m)*}(k)$	(13.5.5)
$\alpha_N^{b(m)}(k) = E_{N,\min}^{b(m)}(k-1) \tilde{q}_N^{(m)}(k)$	(13.5.6)
$\mathbf{t}_N^{(m-1)}(k) = \tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) \tilde{q}_N^{(m)}(k)$	(13.5.7)
$E_{N,\min}^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + e_N^{f(m)}(k) \alpha_N^{f(m)*}(k)$	(13.5.8)
$\varphi_{N+1}^{(m)}(k) = \varphi_N^{(m)}(k) \frac{\lambda E_{N,\min}^{f(m)}(k-1)}{E_{N,\min}^{f(m)}(k)}$	(13.5.9)
$\varphi_{N+1}^{(m-1)}(k) = \frac{\varphi_{N+1}^{(m)}(k)}{1 - \lambda^{-1} \varphi_{N+1}^{(m)}(k) \tilde{q}_N^{(m)*}(k) \alpha_N^{b(m)}(k)}$	(13.5.10)
$e_N^{b(m)}(k) = \alpha_N^{b(m)}(k) \varphi_N^{(m-1)}(k)$	(13.5.11)
$E_N^{b(m)}(k) = \lambda E_N^{b(m)}(k-1) + e_N^{b(m)}(k) \alpha_N^{b(m)*}(k)$	(13.5.12)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) e_N^{b(m)*}(k)$	(13.5.13)
End for m	
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N^{(0)}(k)$	(13.5.14)
$e_N(k) = \alpha_N(k) \varphi_N^{(0)}(k)$	(13.5.15)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N^{(0)}(k) e_N^{*}(k)$	(13.5.16)
$\mathbf{t}_N^{(M)}(k+1) = \mathbf{t}_N^{(0)}(k)$, $\varphi_N^{(M)}(k+1) = \varphi_N^{(0)}(k)$	(13.5.17)
End for k	

Вычислительная сложность 1-SUSD FTF-алгоритма равна $5MN + 2N$ операциям умножения, $5MN + 2N$ операциям сложения и $3M$ операциям деления, т.е. $O(NM)$.

Многоканальный 1-SUSD FAEST-алгоритм, приведенный в табл. 13.6, отличается от 1-SUSD FTF-алгоритма (см. табл. 13.5) всего лишь двумя уравнениями. В 1-SUSD FFT-алгоритме (см. также раздел 11.8) рекурсивно вычисляется перемен-

ная $\varphi_N^{(m-1)}(k)$, а в 1-SUSD FAEST-алгоритме (см. также раздел 11.9) — ее обратное значение $[\varphi_N^{(m-1)}(k)]^{-1}$. Для этого сначала используется уравнение (13.130), а затем из уравнения (13.131), с учетом уравнения (13.138), определяется $[\varphi_N^{(m-1)}(k)]^{-1}$ как

$$\begin{aligned} [\varphi_N^{(m-1)}(k)]^{-1} &= [\varphi_{N+1}^{(m)}(k)]^{-1} - e_N^{b(m)*}(k) [E_{N,\min}^{b(m)}(k)]^{-1} e_N^{b(m)}(k) = \\ &= [\varphi_{N+1}^{(m)}(k)]^{-1} - \lambda^{-1} e_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k). \end{aligned} \quad (13.153)$$

Таблица 13.6. Многоканальный 1-SUSD FAEST-алгоритм

Вычисления	Ссылки
Initialization: $E_N^{f(m)}(0) = \delta^2$, $E_N^{b(m)}(0) = \delta^2 \lambda^{-N_m}$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$,	(13.6.0)
$\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T}$, $m = 1, \dots, M$, $\mathbf{t}_N^{(M)}(1) = \mathbf{0}_N$, $\varphi_N^{(M)}(1) = 1$,	
$\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = \mathbf{x}_m(k) - \mathbf{h}_N^{f(m)H}(k-1) \mathbf{x}_N^{(m)}(k)$	(13.6.1)
$e_N^{f(m)}(k) = \alpha_N^{f(m)}(k) \varphi_N^{(m)}(k)$	(13.6.2)
$\tilde{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k-1) \end{bmatrix} \frac{\alpha_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k-1)}$	(13.6.3)
$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)T} \tilde{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} \tilde{\mathbf{q}}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}$	(13.6.4)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m)}(k) e_N^{f(m)*}(k)$	(13.6.5)
$\alpha_N^{b(m)}(k) = E_{N,\min}^{b(m)}(k-1) \tilde{q}_N^{(m)}(k)$	(13.6.6)
$\mathbf{t}_N^{(m-1)}(k) = \tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) \tilde{q}_N^{(m)}(k)$	(13.6.7)
$E_{N,\min}^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + e_N^{f(m)}(k) \alpha_N^{f(m)*}(k)$	(13.6.8)
$\frac{1}{\varphi_{N+1}^{(m)}(k)} = \frac{1}{\varphi_N^{(m)}(k)} + \lambda^{-1} \frac{\alpha_N^{f(m)*}(k) \alpha_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k-1)}$	(13.6.9)
$\frac{1}{\varphi_N^{(m-1)}(k)} = \frac{1}{\varphi_{N+1}^{(m)}(k)} - \lambda^{-1} \alpha_N^{b(m)*}(k) \tilde{q}_N^{(m)}(k)$	(13.6.10)
$e_N^{b(m)}(k) = \alpha_N^{b(m)}(k) \varphi_N^{(m-1)}(k)$	(13.6.11)
$E_{N,\min}^{b(m)}(k) = \lambda E_{N,\min}^{b(m)}(k-1) + e_N^{b(m)}(k) \alpha_N^{b(m)*}(k)$	(13.6.12)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) e_N^{b(m)*}(k)$	(13.6.13)
End for m	
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N^{(0)}(k)$	(13.6.14)
$e_N(k) = \alpha_N(k) \varphi_N^{(0)}(k)$	(13.6.15)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N^{(0)}(k) e_N^*(k)$	(13.6.16)
$\mathbf{t}_N^{(M)}(k+1) = \mathbf{t}_N^{(0)}(k)$, $\varphi_N^{(M)}(k+1) = \varphi_N^{(0)}(k)$	(13.6.17)
End for k	

Вычислительная сложность многоканального 1-SUSD FAEST-алгоритма равна $5MN+2N$ операциям умножения, $5MN+2N$ операциям сложения и $2M$ операциям деления, т. е. $O(NM)$.

Умножение на перестановочные матрицы в 1-SUSD алгоритмах (см. табл. 13.4—13.6) не влечет за собой выполнения дополнительных арифметических операций, поскольку для конкретных значений N_m в каналах M -канального адаптивного фильтра можно заранее вычислить произведения $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$ и определить правила перестановки элементов в векторах с помощью адресного вектора.

Многоканальный 1-SUSD стабилизированный FAEST-алгоритм, аналогичный алгоритму (см. табл. 11.4), приведен в табл. 13.7.

Таблица 13.7. Многоканальный 1-SUSD стабилизированный FAEST-алгоритм

Вычисления	Ссылки
Initialization: $E_N^{f(m)}(0) = \delta^2$, $E_N^{b(m)}(0) = \delta^2 \lambda^{-N_m}$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$,	(13.7.0)
$\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$, $m=1, \dots, M$, $\mathbf{t}_N^{(M)}(1) = \mathbf{0}_N$,	
$\varphi_N^{(M)}(1) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(13.7.1)
$e_N^{f(m)}(k) = \alpha_N^{f(m)}(k) \varphi_N^{(m)}(k)$	(13.7.2)
$\bar{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} 0 \\ \mathbf{t}_N^{(m)}(k) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{h}_N^{f(m)}(k-1) \end{bmatrix} \frac{\alpha_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k-1)}$	(13.7.3)
$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top} \bar{\mathbf{t}}_{N+1}^{(m)}(k) = \begin{bmatrix} \mathbf{q}_N^{(m)}(k) \\ \tilde{q}_N^{(m)}(k) \end{bmatrix}$	(13.7.4)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m)}(k) e_N^{f(m)*}(k)$	(13.7.5)
$E_{N,\min}^{f(m)}(k) = \lambda E_{N,\min}^{f(m)}(k-1) + e_N^{f(m)}(k) \alpha_N^{f(m)*}(k)$	(13.7.6)
$\frac{1}{\varphi_{N+1}^{(m)}(k)} = \frac{1}{\varphi_N^{(m)}(k)} + \lambda^{-1} \frac{\alpha_N^{f(m)*}(k) \alpha_N^{f(m)}(k)}{E_{N,\min}^{f(m)}(k-1)}$	(13.7.7)
$\alpha_N^{b(m)}(k) = x_m(k - N_m) - \mathbf{h}_N^{b(m)\top}(k-1) \mathbf{x}_N^{(m-1)}(k)$	(13.7.8)
$q_N^{(m)}(k) = \alpha_N^{b(m)}(k) / E_{N,\min}^{b(m)}(k-1)$	(13.7.9)
$\tilde{\alpha}_N^{b(m)}(k) = E_{N,\min}^{b(m)}(k-1) \tilde{q}_N^{(m)}(k)$	(13.7.10)
$\alpha_N^{b(1)(m)}(k) = K_1 \alpha_N^{b(m)}(k) + (1 - K_1) \tilde{\alpha}_N^{b(m)}(k)$	(13.7.11)
$\alpha_N^{b(2)(m)}(k) = K_2 \alpha_N^{b(m)}(k) + (1 - K_2) \tilde{\alpha}_N^{b(m)}(k)$	(13.7.12)
$\alpha_N^{b(5)(m)}(k) = K_5 \alpha_N^{b(m)}(k) + (1 - K_5) \tilde{\alpha}_N^{b(m)}(k)$	(13.7.13)
$t_{N+1}^{(N)(m)}(k) = K_4 q_N^{(m)}(k) + (1 - K_4) \tilde{q}_N^{(m)}(k)$	(13.7.14)
$\mathbf{t}_N^{(m-1)}(k) = \tilde{\mathbf{q}}_N^{(m)}(k) + \mathbf{h}_N^{b(m)}(k-1) t_{N+1}^{(N)(m)}(k)$	(13.7.15)

Таблица 13.7. (окончание)

Вычисления	Ссылки
$\frac{1}{\hat{\varphi}_N^{(m-1)}(k)} = 1 + \lambda^{-1} \mathbf{t}_N^{(m-1)H}(k) \mathbf{x}_N^{(m-1)}(k)$	(13.7.16)
$\frac{1}{\tilde{\varphi}_N^{(m-1)}(k)} = \frac{1}{\varphi_{N+1}^{(m)}(k)} - \lambda^{-1} \alpha_N^{b(5)(m)*}(k) \tilde{q}_N^{(m)}(k)$	(13.7.17)
$\varphi_N^{(m-1)}(k) = \left[K_3 \frac{1}{\hat{\varphi}_N^{(m-1)}(k)} + (1 - K_3) \frac{1}{\tilde{\varphi}_N^{(m-1)}(k)} \right]^{-1}$	(13.7.18)
$e_N^{b(1)(m)}(k) = \alpha_N^{b(1)(m)}(k) \tilde{\varphi}_N^{(m-1)}(k)$	(13.7.19)
$e_N^{b(2)(m)}(k) = \alpha_N^{b(2)(m)}(k) \tilde{\varphi}_N^{(m-1)}(k)$	(13.7.20)
$E_{N,\min}^{b(m)}(k) = \lambda E_N^{b(m)}(k-1) + e_N^{b(2)(m)}(k) \alpha_N^{b(2)(m)*}(k)$	(13.7.21)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \lambda^{-1} \mathbf{t}_N^{(m-1)}(k) e_N^{b(1)(m)*}(k)$	(13.7.22)
End for m	m
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N^{(0)}(k)$	(13.7.23)
$e_N(k) = \alpha_N(k) \varphi_N^{(0)}(k)$	(13.7.24)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \lambda^{-1} \mathbf{t}_N^{(0)}(k) e_N^*(k)$	(13.7.25)
$\mathbf{t}_N^{(M)}(k+1) = \mathbf{t}_N^{(0)}(k), \varphi_N^{(M)}(k+1) = \varphi_N^{(0)}(k)$	(13.7.26)
End for k	

Вычислительная сложность 1-SUSD стабилизированного FAEST-алгоритма равна $7MN + 2N$ операциям умножения, $7MN + 2N$ операциям сложения и $3M$ операциям деления, т. е. $O(NM)$. Она несколько выше, чем у 1-SUSD FTF-алгоритма и 1-SUSD FAEST-алгоритма, что является следствием дополнительных вычислительных затрат, обеспечивающих устойчивое функционирование адаптивного фильтра.

13.8. Выводы по главе

В настоящей главе рассмотрены два способа построения быстрых многоканальных RLS-алгоритмов: B-SUSD и 1-SUSD. Особенностью многоканальных 1-SUSD быстрых RLS-алгоритмов является отсутствие в их вычислительных процедурах операций обращения матриц с числом элементов $M \times M$, что делает такие алгоритмы привлекательными с точки зрения реализации. Оценка вычислительной сложности 1-SUSD алгоритмов равна $O(NM)$ арифметическим операциям. Несколько других способов построения многоканальных быстрых RLS-алгоритмов приведены в работах [147–150], а способы построения многоканальных лестничных алгоритмов — в работах [142–145, 148, 149, 152–158].

ГЛАВА 14

ДРУГИЕ РАЗНОВИДНОСТИ RLS-АЛГОРИТМОВ

В настоящей главе рассматривается ряд RLS-алгоритмов адаптивной фильтрации, которые могут быть построены с помощью приемов, описанных в главах 6–13. Это быстрые алгоритмы на основе QR-разложения, а также алгоритмы обработки нестационарных сигналов и их параллельные версии, которые могут использоваться как самостоятельно, так и в качестве части вычислительных процедур в быстрых алгоритмах аффинных проекций. Кроме того, в главе рассматриваются вопросы инициализации (задания начальных значений ВК) быстрых RLS-алгоритмов.

14.1. Введение

Многообразие RLS-алгоритмов, существующих в настоящее время, не ограничивается только алгоритмами, рассмотренными в главах 6–13. К другим алгоритмам, в частности, относятся быстрые RLS-алгоритмы на основе QR-разложения [33], а также алгоритмы обработки нестационарных сигналов [66], основанные на использовании скользящего окна [128] при оценке корреляционной матрицы входных сигналов адаптивного фильтра и ее динамической регуляризации [159]. RLS-алгоритмы для обработки нестационарных сигналов существуют и в параллельной форме, ориентированной на реализацию с помощью двух или четырех процессоров. Параллельные алгоритмы инвариантны к структуре (одноканальный или многоканальный) и числу ВК адаптивного фильтра [68]. Кроме того, вычислительные процедуры RLS-алгоритмов со скользящим окном в части линейного предсказания используются в быстрых алгоритмах аффинных проекций [160–168].

Детали получения указанных выше разновидностей RLS-алгоритмов достаточно громоздки. Для алгоритмов обработки нестационарных сигналов они сводятся к примерно двукратному повторению приемов, изложенных в главах 6–13, в случае использования скользящего окна или динамической регуляризации при оценке корреляционной матрицы; к четырехкратному повторению этих приемов в случае одновременного использования скользящего окна и динамиче-

ской регуляризации; и к однократному повторению приемов в случае параллельных алгоритмов. Поэтому в настоящей главе рассматриваются только основные приемы построения алгоритмов обработки нестационарных сигналов и процедуры вычисления лишь некоторых из таких алгоритмов. Это аналоги алгоритмов (см. табл. 6.1 и табл. 10.3). Другие разновидности перечисленных алгоритмов для многоканальных адаптивных фильтров могут быть найдены в работах [66, 68, 151, 165–175]. Особенности инициализации RLS-алгоритмов рассмотрены в работе [176]. Материалы этой главы основаны на указанных публикациях.

14.2. Быстрые алгоритмы на основе обратного QR-разложения

Быстрые RLS-алгоритмы адаптивной фильтрации (см. главы 11 и 13) могут быть также построены на основе QR-разложения матрицы входных сигналов адаптивного фильтра.

В [13] показано, что обратное QR-разложение расширенной матрицы обрабатываемых сигналов одноканального адаптивного фильтра с возрастающим окном

$$\mathbf{X}_{k(N+1)}^{(+)\text{H}}(k) = \begin{bmatrix} \lambda^{0.5(k-1)} & & \mathbf{O} \\ & \lambda^{0.5(k-2)} & \mathbf{x}_N^H(1) & d^*(1) \\ & & \ddots & \vdots \\ \mathbf{O} & & 1 & \mathbf{x}_N^H(k) & d^*(k) \end{bmatrix} \quad (14.1)$$

может быть представлено как

$$\left[\tilde{\mathbf{R}}_{N+1}^{(+)}(k) \right]^{-\text{H}} = \begin{bmatrix} \tilde{\mathbf{R}}_{N,x}^{-\text{H}}(k) & \mathbf{0}_N \\ -\mathbf{h}_N^H(k) E_{N,\min}^{-0.5}(k) & E_{N,\min}^{-0.5}(k) \end{bmatrix}. \quad (14.2)$$

Используя преобразование

$$\mathbf{P}_{N+2}(k) \begin{bmatrix} \lambda^{-0.5} \tilde{\mathbf{a}}_{N+1}^{(+)}(k) & \lambda^{-0.5} \left[\tilde{\mathbf{R}}_{N+1}^{(+)}(k-1) \right]^{-\text{H}} \\ 1 & \mathbf{0}_{N+1}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{N+1} & \left[\tilde{\mathbf{R}}_{N+1}^{(+)}(k) \right]^{-\text{H}} \\ b_{N+1}^{(+)}(k) & \mathbf{u}_{N+1}^{(+)\text{H}} \end{bmatrix} \quad (14.3)$$

для матрицы $\left[\tilde{\mathbf{R}}_{N+1}^{(+)}(k) \right]^{-\text{H}}$ и аналогичные преобразования для матриц на основе векторов $\left[\mathbf{x}_N^T(k), \mathbf{x}(k-N) \right]^T$ и $\left[x(k), \mathbf{x}_N^T(k-1) \right]^T$, можно установить следующие соотношения:

$$\left[\mathbf{u}_N^{(+)\text{H}}(k), u_N^{(+)*}(k) \right] = \left[\mathbf{u}_N^{\text{bH}}(k), u_N^{\text{b}*}(k) \right] = \left[u_N^{\text{f}*}(k), \mathbf{u}_N^{\text{fH}}(k) \right]. \quad (14.4)$$

В уравнении (14.3) переменные $\tilde{\mathbf{a}}_{N+1}^{(+)}(k)$ и $b_{N+1}^{(+)}(k)$ определяются как

$$\tilde{\mathbf{a}}_{N+1}^{(+)}(k) = \left[\tilde{\mathbf{R}}_{N+1}^{(+)}(k-1) \right]^{-\text{H}} \left[\mathbf{x}_N^T(k), d(k) \right]^T \quad (14.5)$$

и

$$b_{N+1}^{(+)}(k) = \sqrt{1 + \lambda^{-1} \tilde{\mathbf{a}}_{N+1}^H(k) \tilde{\mathbf{a}}_{N+1}^{(+)}(k)}. \quad (14.6)$$

$\mathbf{P}_{N+2}(k)$ — это матрица вращений Гивенса, см. аналогичные соотношения (7.38), (7.42) и (7.117), а верхние индексы «f» и «b» обозначают вычисления, связанные с предсказаниями вперед и назад.

В этом случае справедливо соотношение

$$\mathbf{u}_N(k) = \mathbf{u}_N^b, \quad (14.7)$$

которое позволяет определить вектор коэффициентов Калмана как

$$\mathbf{g}_N(k) = \mathbf{u}_N^b(k) / b_N(k), \quad (14.8)$$

где вектор $\mathbf{u}_N^b(k)$ и скалярная переменная $b_N(k)$ могут быть вычислены рекуррентно.

С помощью выражений (14.1)–(14.8), в работах [177, 178] были получены быстрые RLS-алгоритмы с возрастающим окном на основе обратного QR-разложения. Недостатком этих алгоритмов является наличие в них операций извлечения квадратного корня, число которых пропорционально числу ВК адаптивного фильтра N . Таким же недостатком обладают и быстрые RLS-алгоритмы на основе прямого QR-разложения [179, 180]. Одноканальный вариант быстрого RLS-алгоритма на базе обратного QR-разложения без операций квадратного корня был представлен в работе [181].

Многоканальный вариант этого алгоритма для адаптивного фильтра с возрастающим окном, требующий $2M$ операций извлечения квадратного корня на одну итерацию [66], приведен в табл. 14.1. При получении этого алгоритма использованы приемы, аналогичные (14.1)–(14.8), а также некоторые из приемов, рассмотренные в главах 11 и 13.

Вычислительная сложность алгоритма (см. табл. 14.1) равна $9MN + 2N$ операциям умножения, $6MN + 2N$ операциям сложения, $3M$ операциям деления и $2M$ операциям извлечения квадратного корня, т. е. $O(NM)$.

Исключение операций с квадратным корнем в алгоритме табл. 14.1 достигается путем масштабирования переменных. Такое масштабирование было рассмотрено в главе 9. Результирующий алгоритм без процедуры извлечения квадратного корня приведен в табл. 14.2.

Вычислительная сложность алгоритма (см. табл. 14.2) равна $8MN + 2N$ операциям умножения, $6MN + 2N$ операциям сложения и $2M$ операциям деления, т. е. $O(MN)$. Операции с квадратным корнем отсутствуют.

Анализируя переменные в (14.2.2) и (14.2.12), можно заметить, что они аналогичны переменным, вычисляемым с помощью уравнений (13.130) и (13.131). Таким образом, быстрый RLS-алгоритм на основе обратного QR-разложения без операций извлечения квадратного корня является аналогом FAEST-алгоритма, в котором вместо априорных векторов коэффициентов Калмана \mathbf{t} вычисляют

Таблица 14.1. Многоканальный быстрый RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса

Вычисления	Ссылки
Initialization: $\left[E_{N,\min}^{f(m)}(0) \right]^{-0.5} = \delta^{-1}$, $\left[E_{N,\min}^{b(m)}(0) \right]^{-0.5} = \delta^{-1} (\lambda^{N_m})^{0.5}$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$,	(14.1.0)
$\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$, $m = 1, \dots, M$, $\mathbf{u}_N^{(M)}(1) = \mathbf{0}_N$, $\mathbf{g}_N^{(M)}(1) = \mathbf{0}_N$,	
$b_N^{(M)}(1) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(14.1.1)
$\tilde{a}_{N+1}^{f(m)}(k) = \left[E_{N,\min}^{f(m)}(k-1) \right]^{-0.5} \alpha_N^{f(m)}(k)$	(14.1.2)
$b_{N+1}^{(m)}(k) = \sqrt{b_N^{(m)2}(k) + \lambda^{-1} \tilde{a}_{N+1}^{f(m)*}(k) \tilde{a}_{N+1}^{f(m)}(k)}$	(14.1.3)
$c_N^{f(m)}(k) = b_N^{(m)}(k) / b_{N+1}^{(m)}(k)$	(14.1.4)
$s_N^{f(m)}(k) = \lambda^{-0.5} \tilde{a}_{N+1}^{f(m)*}(k) / b_{N+1}^{(m)}(k)$	(14.1.5)
$\mathbf{q}_N^{f(m)}(k) = c_N^{f(m)}(k) \mathbf{u}_N^{(m)}(k) - \lambda^{-0.5} s_N^{f(m)*}(k) \mathbf{h}_N^{f(m)}(k-1) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-0.5}$	(14.1.6)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{f(m)*}(k)$	(14.1.7)
$q_N^{f(m)}(k) = \lambda^{-0.5} \left[E_{N,\min}^{f(m)}(k-1) \right]^{-0.5} s_N^{f(m)*}(k)$	(14.1.8)
$\left[E_N^{f(m)}(k) \right]^{-0.5} = \lambda^{-0.5} c_N^{f(m)}(k) \left[E_{N,\min}^{f(m)}(k-1) \right]^{-0.5}$	(14.1.9)
$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top} \begin{bmatrix} q_N^{f(m)}(k) \\ q_N^{f(m)}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_N^{(m)}(k) \\ q_N^{(m)}(k) \end{bmatrix}$	(14.1.10)
$\alpha_N^{b(m)}(k) = x_m(k - N_m) - \mathbf{h}_N^{b(m)\top}(k-1) \mathbf{x}_N^{(m-1)}(k)$	(14.1.11)
$\tilde{a}_{N+1}^{b(m)}(k) = \left[E_{N,\min}^{b(m)}(k-1) \right]^{-0.5} \alpha_N^{b(m)}(k)$	(14.1.12)
$b_N^{(m-1)}(k) = \sqrt{b_{N+1}^{(m)2}(k) - \lambda^{-1} \tilde{a}_{N+1}^{b(m)*}(k) \tilde{a}_{N+1}^{b(m)}(k)}$	(14.1.13)
$c_N^{b(m)}(k) = b_N^{(m-1)}(k) / b_{N+1}^{(m)}(k)$	(14.1.14)
$s_N^{b(m)}(k) = \lambda^{-0.5} \tilde{a}_{N+1}^{b(m)*}(k) / b_{N+1}^{(m)}(k)$	(14.1.15)
$\mathbf{q}_N^{(m-1)}(k) = \mathbf{u}_N^{(m-1)}(k) / b_N^{(m-1)}(k) + \lambda^{-0.5} \left[E_{N,\min}^{b(m)}(k-1) \right]^{-0.5} s_N^{b(m)*}(k) \mathbf{h}_N^{b(m)}(k-1) \left[c_N^{b(m)}(k) \right]^{-1}$	(14.1.16)
$\mathbf{g}_N^{(m-1)}(k) = \mathbf{u}_N^{(m-1)}(k) / b_N^{(m-1)}(k)$	(14.1.17)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k)$	(14.1.18)
$\left[E_N^{b(m)}(k) \right]^{-0.5} = \lambda^{-0.5} c_N^{b(m)}(k) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-0.5}$	(14.1.19)
End for m	
$\alpha_{N,\mathbf{x}}(k) = d(k) - \mathbf{h}_N^{\top}(k-1) \mathbf{x}_N^{(0)}(k)$	(14.1.20)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N^{(0)}(k) \alpha_N^{(0)*}(k)$	(14.1.21)
$\mathbf{g}_N^{(M)}(k+1) = \mathbf{g}_N^{(0)}(k)$, $\mathbf{u}_N^{(M)}(k+1) = \mathbf{u}_N^{(0)}(k)$, $b_N^{(M)}(k+1) = b_N^{(0)}(k)$	(14.1.22)
End for k	

Таблица 14.2. Многоканальный быстрый RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса без операций извлечения квадратного корня

Вычисления	Ссылки
Initialization: $[E_{N,\min}^{f(m)}(0)]^{-1} = \delta^{-2}$, $[E_{N,\min}^{b(m)}(0)]^{-1} = \delta^{-2} \lambda^{N_m}$, $\mathbf{h}_N^{f(m)}(0) = \mathbf{0}_N$, $\mathbf{h}_N^{b(m)}(0) = \mathbf{0}_N$, create: $\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top}$, $m = 1, \dots, M$, $\mathbf{g}_N^{(M)}(1) = \mathbf{0}_N$, $K_N^{B(M)}(1) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	(14.2.0)
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{f(m)}(k) = x_m(k) - \mathbf{h}_N^{f(m)\top}(k-1) \mathbf{x}_N^{(m)}(k)$	(14.2.1)
$K_{N+1}^{B(m)}(k) = K_N^{B(m)}(k) + \lambda^{-1} \alpha_N^{f(m)*}(k) [E_{N,\min}^{f(m)}(k-1)]^{-1} \alpha_N^{f(m)}(k)$	(14.2.2)
$\bar{c}_N^{f(m)}(k) = K_N^{B(m)}(k) / K_{N+1}^{B(m)}(k)$	(14.2.3)
$\bar{s}_N^{f(m)}(k) = \lambda^{-1} [E_{N,\min}^{f(m)}(k-1)]^{-1} \alpha_N^{f(m)*}(k) / K_{N+1}^{B(m)}(k)$	(14.2.4)
$\bar{\mathbf{q}}_N^{f(m)}(k) = \mathbf{g}_N^{(m)}(k) \bar{c}_N^{f(m)}(k) - \mathbf{h}_N^{f(m)\top}(k-1) \bar{s}_N^{f(m)*}(k)$ или (14.2.7)	(14.2.5)
$\mathbf{h}_N^{f(m)}(k) = \mathbf{h}_N^{f(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{f(m)*}(k)$	(14.2.6)
$\bar{\mathbf{q}}_N^{f(m)}(k) = \mathbf{g}_N^{(m)}(k) - \mathbf{h}_N^{f(m)\top}(k) \bar{s}_N^{f(m)*}(k)$ или (14.2.5)	(14.2.7)
$\bar{q}_N^{f(m)}(k) = \bar{s}_N^{f(m)*}(k)$	(14.2.8)
$[E_{N,\min}^{f(m)}(k)]^{-1} = \lambda^{-1} \bar{c}_N^{f(m)}(k) [E_{N,\min}^{f(m)}(k-1)]^{-1}$	(14.2.9)
$\mathbf{S}_{N+1}^{(m)} \mathbf{T}_{N+1}^{(m)\top} \begin{bmatrix} \bar{q}_N^{f(m)}(k) \\ \bar{\mathbf{q}}_N^{f(m)}(k) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{q}}_N^{b(m)}(k) \\ \bar{q}_N^{b(m)}(k) \end{bmatrix}$	(14.2.10)
$\alpha_N^{b(m)}(k) = x_m(k-N_m) - \mathbf{h}_N^{b(m)\top}(k-1) \mathbf{x}_N^{(m-1)}(k)$	(14.2.11)
$K_N^{B(m-1)}(k) = K_{N+1}^{B(m)}(k) - \lambda^{-1} \alpha_N^{b(m)*}(k) [E_{N,\min}^{b(m)}(k-1)]^{-1} \alpha_N^{b(m)}(k)$	(14.2.12)
$\bar{c}_N^{b(m)}(k) = K_N^{B(m-1)}(k) / K_{N+1}^{B(m)}(k)$	(14.2.13)
$\bar{s}_N^{b(m)}(k) = \lambda^{-1} [E_{N,\min}^{b(m)}(k-1)]^{-1} \alpha_N^{b(m)*}(k) / K_{N+1}^{B(m)}(k)$	(14.2.14)
$\mathbf{g}_N^{(m-1)}(k) = [\bar{\mathbf{q}}_N^{b(m)}(k) + \mathbf{h}_N^{b(m)\top}(k-1) \bar{s}_N^{b(m)*}(k)] [\bar{c}_N^{b(m)}(k)]^{-1}$	(14.2.15)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \mathbf{g}_N^{(m-1)}(k) \alpha_N^{b(m)*}(k)$	(14.2.16)
$[E_{N,\min}^{b(m)}(k)]^{-1} = \lambda^{-1} \bar{c}_N^{b(m)}(k) [E_{N,\min}^{b(m)}(k-1)]^{-1}$	(14.2.17)
End for m	
$\alpha_N(k) = d(k) - \mathbf{h}_N^{\top}(k-1) \mathbf{x}_N^{(0)}(k)$	(14.2.18)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N^{(0)}(k) \alpha_N^{(0)*}(k)$	(14.2.19)
$\mathbf{g}_N^{(M)}(k+1) = \mathbf{g}_N^{(0)}(k)$, $K_N^{B(M)}(k+1) = K_N^{B(0)}(k)$	(14.2.20)
End for k	

ся апостериорные векторы \mathbf{g} . Используя эти соотношения, можно построить стабилизированную версию алгоритма табл. 14.2. Такой алгоритм приведен в табл. 14.3.

Таблица 14.3. Многоканальный быстрый стабилизированный RLS-алгоритм на основе обратного QR-разложения с использованием вращений Гивенса без операций извлечения квадратного корня

Вычисления	Ссылки
Initialization: $[E_{N,\min}^{\text{f}(m)}(0)]^{-1} = \delta^{-2}$, $[E_{N,\min}^{\text{b}(m)}(0)]^{-1} = \delta^{-2} \lambda^{N_m}$, $\mathbf{h}_N^{\text{f}(m)}(0) = \mathbf{0}_N$, $\mathbf{h}_N^{\text{b}(m)}(0) = \mathbf{0}_N$, $\mathbf{S}_{N+1}^{\text{(m)}} \mathbf{T}_{N+1}^{\text{(m)T}}$, $m = 1, \dots, M$, $\mathbf{g}_N^{(M)}(1) = \mathbf{0}_N$, $K_N^{B(M)}(1) = 1$, $\mathbf{x}_N(0) = \mathbf{0}_N$, $\mathbf{h}_N(0) = \mathbf{0}_N$	(14.3.0)
For $k = 1, 2, \dots, K$	
For $m = M, M-1, \dots, 1$	
$\alpha_N^{\text{f}(m)}(k) = x_m(k) - \mathbf{h}_N^{\text{f}(m)H}(k-1) \mathbf{x}_N^{(m)}(k)$	(14.3.1)
$e_N^{\text{f}(m)}(k) = \alpha_N^{\text{f}(m)}(k) / K_N^{B(m)}(k)$	(14.3.2)
$\bar{K}_{N+1}^{B(m)}(k) = K_N^{B(m)}(k) + \lambda^{-1} \alpha_N^{\text{f}(m)*}(k) [E_{N,\min}^{\text{f}(m)}(k-1)]^{-1} \alpha_N^{\text{f}(m)}(k)$	(14.3.3)
$c_N^{\text{f}(m)}(k) = K_N^{B(m)}(k) / \bar{K}_{N+1}^{B(m)}(k)$	(14.3.4)
$s_N^{\text{f}(m)}(k) = \lambda^{-1} [E_{N,\min}^{\text{f}(m)}(k-1)]^{-1} \alpha_N^{\text{f}(m)*}(k) / \bar{K}_{N+1}^{B(m)}(k)$	(14.3.5)
$\tilde{\mathbf{q}}_N^{\text{f}(m)}(k) = \mathbf{g}_N^{(m)}(k) c_N^{\text{f}(m)}(k) - \mathbf{h}_N^{\text{f}(m)H}(k-1) s_N^{\text{f}(m)*}(k)$ или (14.3.8)	(14.3.6)
$\mathbf{h}_N^{\text{f}(m)}(k) = \mathbf{h}_N^{\text{f}(m)}(k-1) + \mathbf{g}_N^{(m)}(k) \alpha_N^{\text{f}(m)*}(k)$	(14.3.7)
$\tilde{\mathbf{q}}_N^{\text{f}(m)}(k) = \mathbf{g}_N^{(m)}(k) - \mathbf{h}_N^{\text{f}(m)H}(k) s_N^{\text{f}(m)*}(k)$ или (14.3.6)	(14.3.8)
$\tilde{q}_N^{\text{f}(m)}(k) = s_N^{\text{f}(m)*}(k)$	(14.3.9)
$\mathbf{S}_{N+1}^{\text{(m)}} \mathbf{T}_{N+1}^{\text{(m)T}} \begin{bmatrix} \tilde{q}_N^{\text{f}(m)}(k) \\ \tilde{\mathbf{q}}_N^{\text{f}(m)}(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{q}}_N^{\text{b}(m)}(k) \\ \tilde{q}_N^{\text{b}(m)}(k) \end{bmatrix}$	(14.3.10)
$E_{N,\min}^{\text{f}(m)}(k) = \lambda E_{N,\min}^{\text{f}(m)}(k-1) + e_N^{\text{f}(m)}(k) \alpha_N^{\text{f}(m)*}(k)$	(14.3.11)
$\alpha_N^{\text{b}(m)}(k) = x_m(k - N_m) - \mathbf{h}_N^{\text{b}(m)H}(k-1) \mathbf{x}_N^{(m-1)}(k)$	(14.3.12)
$\tilde{\alpha}_N^{\text{b}(m)}(k) = \lambda E_{N,\min}^{\text{b}(m)}(k-1) \tilde{q}_N^{\text{b}(m)}(k) \bar{K}_{N+1}^{B(m)}(k)$	(14.3.13)
$\alpha_N^{\text{b}(1)(m)}(k) = K_1 \alpha_N^{\text{b}(m)}(k) + (1 - K_1) \tilde{\alpha}_N^{\text{b}(m)}(k)$	(14.3.14)
$\alpha_N^{\text{b}(2)(m)}(k) = K_2 \alpha_N^{\text{b}(m)}(k) + (1 - K_2) \tilde{\alpha}_N^{\text{b}(m)}(k)$	(14.3.15)
$\alpha_N^{\text{b}(4)(m)}(k) = K_4 \alpha_N^{\text{b}(m)}(k) + (1 - K_4) \tilde{\alpha}_N^{\text{b}(m)}(k)$	(14.3.16)
$\alpha_N^{\text{b}(5)(m)}(k) = K_5 \alpha_N^{\text{b}(m)}(k) + (1 - K_5) \tilde{\alpha}_N^{\text{b}(m)}(k)$	(14.3.17)
$s_N^{\text{b}(m)}(k) = \lambda^{-1} [E_{N,\min}^{\text{b}(m)}(k-1)]^{-1} \alpha_N^{\text{b}(4)(m)*}(k) / \bar{K}_{N+1}^{B(m)}(k)$	(14.3.18)
$\tilde{\mathbf{z}}_N^{\text{b}(m)}(k) = [\tilde{\mathbf{q}}_N^{\text{b}(m)}(k) + \mathbf{h}_N^{\text{b}(m)H}(k-1) s_N^{\text{b}(m)*}(k)]$	(14.3.19)
$\hat{K}_N^{B(m-1)}(k) = 1 + \bar{K}_{N+1}^{B(m)}(k) \tilde{\mathbf{z}}_N^{\text{b}(m)H}(k) \mathbf{x}_N^{(m-1)}(k)$	(14.3.20)



Таблица 14.3. (окончание)

Вычисления	Ссылки
$\tilde{K}_N^{B(m-1)}(k) = \bar{K}_{N+1}^{B(m)}(k) - \lambda^{-1} \alpha_N^{b(5)(m)*}(k) \left[E_{N,\min}^{b(m)}(k-1) \right]^{-1} \alpha_N^{b(m)}(k)$	(14.3.21)
$K_N^{B(m-1)}(k) = K_3 \tilde{K}_N^{B(m-1)}(k) + (1 - K_3) \tilde{K}_N^{B(m-1)}(k)$	(14.3.22)
$\tilde{c}_{N,x}^{b(m)}(k) = \tilde{K}_N^{B(m-1)}(k) / \bar{K}_{N+1}^{B(m)}(k)$	(14.3.23)
$\tilde{g}_N^{(m-1)}(k) = \tilde{z}_{N,\min}^{b(m)}(k) / \tilde{c}_N^{b(m)}(k)$	(14.3.24)
$e_N^{b(2)(m)}(k) = \alpha_N^{b(2)(m)}(k) / \tilde{K}_N^{B(m-1)}(k)$	(14.3.25)
$E_{N,\min}^{b(m)}(k) = \lambda E_{N,\min}^{b(m)}(k-1) + e_N^{b(2)(m)}(k) \alpha_N^{b(2)(m)*}(k)$	(14.3.26)
$\mathbf{h}_N^{b(m)}(k) = \mathbf{h}_N^{b(m)}(k-1) + \tilde{g}_N^{(m-1)}(k) \alpha_N^{b(1)(m)*}(k)$	(14.3.27)
$c_N^{b(m)}(k) = K_N^{B(m-1)}(k) / \bar{K}_{N+1}^{B(m)}(k)$	(14.3.28)
$\mathbf{g}_N^{(M)}(k+1) = \mathbf{g}_N^{(0)}(k), K_N^{B(M)}(k+1) = K_N^{B(0)}(k)$	(14.3.32)
End for m	
$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N^{(0)}(k)$	(14.3.30)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N^{(0)}(k) \alpha_N^*(k)$	(14.3.31)
End for k	

Вычислительная сложность алгоритма (см. табл. 14.3) равна $11MN + 2N$ операциям умножения, $8MN + 2N$ операциям сложения и $7M$ операциям деления, т. е. $O(NM)$. Операции извлечения квадратного корня отсутствуют.

14.3. Регуляризация RLS-алгоритмов

Корреляционная матрица сигналов в RLS-алгоритмах иногда может быть плохо обусловленной. В [159] для одноканальных адаптивных фильтров с действительными ВК была применена динамическая регуляризация этой матрицы. Такая регуляризация может быть использована и в многоканальных адаптивных фильтрах с комплексными ВК с возрастающим или скользящим окном, т. е. она применима ко всем рассмотренным в настоящей книге RLS-алгоритмам.

Суть динамической регуляризации заключается в следующем. В случае возрастающего окна (рис. 14.1а и рис. 14.1б) корреляционная матрица сигналов многоканального адаптивного фильтра на каждой итерации модифицируется как

$$\begin{aligned} \mathbf{R}_N(k) &= \mathbf{R}_{N,p}(k) = \sum_{i=1}^k \lambda^{k-i} \left[\mathbf{x}_N(i) \mathbf{x}_N^H(i) + \xi^2 \mathbf{p}_N(i) \mathbf{p}_N^T(i) \right] = \\ &= \lambda \mathbf{R}_{N,p}(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) + \xi^2 \mathbf{p}_N(k) \mathbf{p}_N^T(k) = \mathbf{R}_{N,x}(k) + \xi^2 \mathbf{p}_N(k) \mathbf{p}_N^T(k), \end{aligned} \quad (14.9)$$

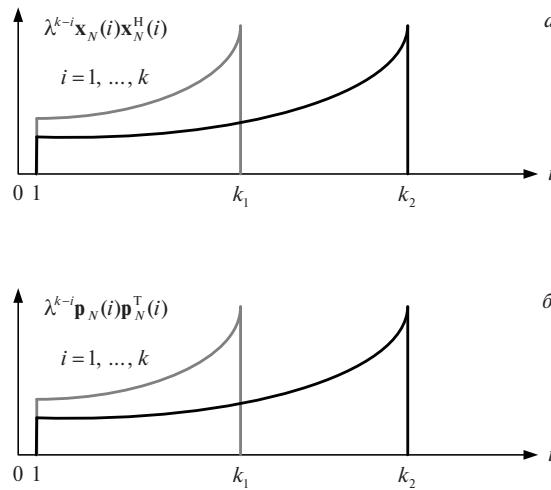


Рис. 14.1. Возрастающие окна с экспоненциальным взвешиванием: а) для составляющих корреляционной матрицы от входных сигналов адаптивного фильтра; б) для составляющих корреляционной матрицы от сигналов регуляризации

см. также уравнение (6.75).

Вектор взаимной корреляции между вектором сигналов регуляризации $\mathbf{p}_N(k)$ и требуемым сигналом равен нуль-вектору, а потому при получении регуляризованных RLS-алгоритмов не используется.

В уравнении (14.9) ξ — небольшое число, представляющее собой параметр динамической регуляризации. Для многоканального адаптивного фильтра вектор искусственных сигналов регуляризации формируется подобно вектору входных сигналов $\mathbf{x}_N(k)$ (см. главу 1) как

$$\mathbf{p}_N(k) = [\mathbf{p}_{N_1}^T(k), \mathbf{p}_{N_2}^T(k), \dots, \mathbf{p}_{N_m}^T(k), \dots, \mathbf{p}_{N_{M-1}}^T(k), \mathbf{p}_{N_M}^T(k)]^T, \quad (14.10)$$

где $\mathbf{p}_{N_m}(k) = [p_m(k), p_m(k-1), \dots, p_m(k-N_m+1), \dots, p_m(k-N_m+2), p_m(k-N_m+1)]^T$,

$$p_m(k) = \begin{cases} 0, & \text{если } 1+n_{m, \text{mod } N_m} \neq 1 \\ 1, & \text{если } 1+n_{m, \text{mod } N_m} = 1, \end{cases} \quad (14.11)$$

а $n_{m, \text{mod } N_m}$ — операция определения остатка от деления числа n_m на N_m . Значение параметра динамической регуляризации ξ^2 может быть выбрано из условия $\xi^2 \geq 0.01\sigma_x^2$ [80], подобно регуляризации NLMS- и RLS-алгоритмов (см. главы 5 и 6).

При получении регуляризованного RLS-алгоритма обращения матрицы (14.9) требуется два раза применять MIL (5.115), что приводит к примерно двукратному увеличению вычислительной сложности регуляризованных алго-



ритмов с возрастающим окном по сравнению с нерегуляризованными алгоритмами с возрастающим окном. Регуляризированный вариант RLS-алгоритма (см. табл. 6.1) представлен в табл. 14.4.

Таблица 14.4. Многоканальный регуляризированный RLS-алгоритм с возрастающим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\xi \mathbf{p}_N(0) = \mathbf{0}_N$, $\mathbf{R}_{N,p}^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N$,	(14.4.0)
$\mathbf{h}_{N,p}(0) = \mathbf{0}_N$,	
$\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}$, $m=1, \dots, M$	
For $k = 1, 2, \dots, K$	
$\mathbf{g}_{N,x}(k) = \frac{\mathbf{R}_{N,p}^{-1}(k-1) \mathbf{x}_N(k)}{\lambda + \mathbf{x}_N^H(k) \mathbf{R}_{N,p}^{-1}(k-1) \mathbf{x}_N(k)}$	(14.4.1)
$\mathbf{R}_{N,x}^{-1}(k) = \lambda^{-1} [\mathbf{R}_{N,p}^{-1}(k-1) - \mathbf{g}_{N,x}(k) \mathbf{x}_N^H(k) \mathbf{R}_{N,p}^{-1}(k-1)]$	(14.4.2)
$\mathbf{g}_{N,p}(k) = \frac{\mathbf{R}_{N,x}^{-1}(k) \xi \mathbf{p}_N(k)}{1 + \xi \mathbf{p}_N^T(k) \mathbf{R}_{N,x}^{-1}(k) \xi \mathbf{p}_N(k)}$	(14.4.3)
$\mathbf{R}_{N,p}^{-1}(k) = \mathbf{R}_{N,x}^{-1}(k) - \mathbf{g}_{N,p}(k) \xi \mathbf{p}_N^T(k) \mathbf{R}_{N,x}^{-1}(k)$	(14.4.4)
$\alpha_{N,x}(k) = d(k) - \mathbf{h}_{N,p}^H(k-1) \mathbf{x}_N(k)$	(14.4.5)
$\mathbf{h}_{N,x}(k) = \mathbf{h}_{N,p}(k-1) + \mathbf{g}_{N,x}(k) \alpha_{N,x}^*(k)$	(14.4.6)
$\alpha_{N,p}(k) = -\mathbf{h}_{N,x}^H(k) \xi \mathbf{p}_N(k)$	(14.4.7)
$\mathbf{h}_{N,p}(k) = \mathbf{h}_{N,x}(k) + \mathbf{g}_{N,p}(k) \alpha_{N,p}^*(k)$	(14.4.8)
End for k	

Оценка вычислительной сложности RLS-алгоритма (см. табл. 14.4) примерно в два раза больше вычислительной сложности RLS-алгоритма (см. табл. 6.1) без регуляризации, т. е. составляет $O(N^2)$ арифметических операций. На самом деле эта оценка несколько ниже из-за разреженной структуры векторов $\xi \mathbf{p}_N(k)$, а также из-за отсутствия множителя λ^{-1} в уравнении (14.4.4).

Пример регуляризированного LC RLS-алгоритма с двойным ограничением (см. главу 10) приведен в табл. 14.5. Оценка вычислительной сложности этого алгоритма примерно в два раза больше оценки вычислительной сложности RLS-алгоритма (см. табл. 10.3) без регуляризации.

В алгоритмах (см. табл. 14.4 и табл. 14.5) при формировании его выходного сигнала используются ВК $\mathbf{h}_{N,p}(k)$, т. е. этот сигнал формируется как $y_{N,x}(k) = \mathbf{h}_{N,p}^H(k-1) \mathbf{x}_N(k)$, а сигнал ошибки $\alpha_{N,x}(k)$ — как (14.5.11).

Таблица 14.5. Многоканальный регуляризированный LC RLS-алгоритм с двойным ограничением

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\xi \mathbf{p}_N(0) = \mathbf{0}_N$, $\mathbf{R}_{N,p}^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N$,	(14.5.0)
$\mathbf{G}_{N,p}(0) = \mathbf{R}_{N,p}^{-1}(0) \mathbf{C}_{N,p}$, $\mathbf{Q}_{N,p}(0) = \mathbf{G}_{N,p}(0) [\mathbf{C}_{N,p}^H \mathbf{G}_{N,p}(0)]^{-1}$,	
$\mathbf{h}_{N,p}(0) = \mathbf{Q}_{N,p}(0) \mathbf{f}_j$,	
$\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}$, $m=1, \dots, M$	
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_{N,x}(k)$, см., например, (14.4.1), (14.4.2)	(14.5.1)
Вычисление $\mathbf{g}_{N,p}(k)$, см., например, (14.4.3), (14.4.4)	(14.5.2)
$\mathbf{v}_{J,x}(k) = \mathbf{C}_{N,p}^H \mathbf{g}_{N,x}(k)$	(14.5.3)
$\mathbf{v}_{J,x}^H(k) = \mathbf{x}_N^H(k) \mathbf{Q}_{N,p}(k-1)$	(14.5.4)
$\mathbf{Q}'_{N,p}(k) = [\mathbf{Q}_{N,p}(k-1) - \mathbf{g}_{N,x}(k) \mathbf{v}_{J,x}^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J,x}(k) \mathbf{v}_{J,x}^H(k)}{1 - \mathbf{v}_{J,x}^H(k) \mathbf{v}_{J,x}(k)} \right]$	(14.5.5)
$\mathbf{Q}_{N,x}(k) = \mathbf{Q}'_{N,p}(k) + \mathbf{C}_{N,p} (\mathbf{C}_{N,p}^H \mathbf{C}_{N,p})^{-1} [\mathbf{I}_J - \mathbf{C}_{N,p}^H \mathbf{Q}'_{N,p}(k)]$	(14.5.6)
$\mathbf{v}_{J,p}(k) = \mathbf{C}_{N,p}^H \mathbf{g}_{N,p}(k)$	(14.5.7)
$\mathbf{v}_{J,p}^H(k) = \xi \mathbf{p}_N^T(k) \mathbf{Q}_{N,x}(k)$	(14.5.8)
$\mathbf{Q}'_{N,p}(k) = [\mathbf{Q}_{N,x}(k) - \mathbf{g}_{N,p}(k) \mathbf{v}_{J,p}^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J,p}(k) \mathbf{v}_{J,p}^H(k)}{1 - \mathbf{v}_{J,p}^H(k) \mathbf{v}_{J,p}(k)} \right]$	(14.5.9)
$\mathbf{Q}_{N,p}(k) = \mathbf{Q}'_{N,p}(k) + \mathbf{C}_{N,p} (\mathbf{C}_{N,p}^H \mathbf{C}_{N,p})^{-1} [\mathbf{I}_J - \mathbf{C}_{N,p}^H \mathbf{Q}'_{N,p}(k)]$	(14.5.10)
$\alpha_{N,x}(k) = d(k) - \mathbf{h}_{N,p}^H(k-1) \mathbf{x}_N(k)$	(14.5.11)
$\mathbf{h}'_{N,x}(k) = \mathbf{h}_{N,p}(k-1) + \mathbf{g}_{N,x}(k) \alpha_{N,x}^*(k)$	(14.5.12)
$\mathbf{h}_{N,x}(k) = \mathbf{h}'_{N,x}(k) + \mathbf{Q}_{N,x}(k) [\mathbf{f}_j - \mathbf{C}_{N,x}^H \mathbf{h}'_{N,x}(k)]$	(14.5.13)
$\alpha_{N,p}(k) = -\mathbf{h}_{N,x}^H(k) \xi \mathbf{p}_N(k)$	(14.5.14)
$\mathbf{h}'_{N,p}(k) = \mathbf{h}_{N,x}(k) + \mathbf{g}_{N,p}(k) \alpha_{N,p}^*(k)$	(14.5.15)
$\mathbf{h}_{N,p}(k) = \mathbf{h}'_{N,p}(k) + \mathbf{Q}_{N,p}(k) [\mathbf{f}_j - \mathbf{C}_{N,p}^H \mathbf{h}'_{N,p}(k)]$	(14.5.16)
End for k	

14.4. RLS-алгоритмы со скользящим окном

В рассмотренных до настоящего времени RLS- и быстрых RLS-алгоритмах предполагалось, что вычисление корреляционной матрицы сигналов осуществляется на возрастающем окне отсчетов (см. рис. 6.5 и рис. 14.1). Для обеспечения следящих свойств таких алгоритмов при обработке нестационарных сигналов обычно используется экспоненциальное взвешивание значений от-



счетов с помощью параметра $(1-0.4/N) \leq \lambda \leq 1$. Из этого неравенства следует, что минимально возможное значение параметра λ ограничено числом ВК адаптивного фильтра, а потому эффективность экспоненциального взвешивания уменьшается с увеличением числа ВК N , так как $\lambda \rightarrow 1$. В этом случае более эффективным является использование скользящего окна, длина которого (число отсчетов L , см. рис. 14.2) определяется интервалом стационарности обрабатываемых сигналов. Скользящее окно, как правило, также используется совместно с экспоненциальным взвешиванием. Оно может быть использовано во всех алгоритмах, рассмотренных в главах 6–13.

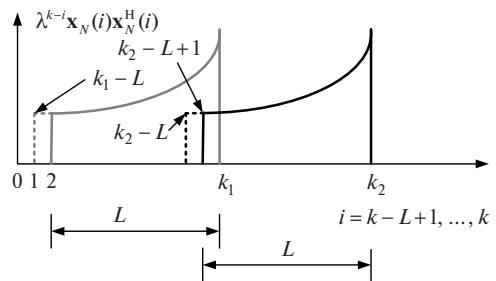


Рис. 14.2. Скользящее окно с экспоненциальным взвешиванием для составляющих корреляционной матрицы от входных сигналов адаптивного фильтра

Для многоканального адаптивного фильтра матрица $\mathbf{R}_N(k)$ и вектор $\mathbf{r}_N(k)$ определяются как

$$\begin{aligned} \mathbf{R}_N(k) = \mathbf{R}_{N,x_D}(k) = \sum_{i=k-L+1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \lambda \mathbf{R}_{N,x_D}(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) - \\ - \mu \mathbf{x}_N(k-L) \mathbf{x}_N^H(k-L) = \mathbf{R}_{N,x_U}(k) - \mu \mathbf{x}_N(k-L) \mathbf{x}_N^H(k-L) \end{aligned} \quad (14.12)$$

и

$$\begin{aligned} \mathbf{r}_N(k) = \mathbf{r}_{N,x_D}(k) = \sum_{i=k-L+1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) = \lambda \mathbf{r}_{N,x_D}(k-1) + \mathbf{x}_N(k) d^*(k) - \\ - \mu \mathbf{x}_N(k-L) d^*(k-L) = \mathbf{r}_{N,x_U}(k) - \mu \mathbf{x}_N(k-L) d^*(k-L), \end{aligned} \quad (14.13)$$

где $\mu = \lambda^L$ (не путать с шагом сходимости μ градиентных алгоритмов из глав 4 и 5). Нижний индекс U (от слова «up-dating») используется для обозначения вычислений, связанных с обновлением данных в выражениях (14.12) и (14.13) за счет сигналов с текущим отсчетом k , а индекс D (от слова «down-dating») — для обозначения вычислений, связанных с исключением старых данных, т.е. с отсчетом $(k-L)$. Исключаемые данные показаны штриховыми линиями на рис. 14.2.

Для получения RLS-алгоритма со скользящим окном требуется применять окном MIL (5.115) два раза при обращении матрицы (14.12), что обусловлено

наличием двух слагаемых в этом уравнении, зависящих от $\mathbf{x}_N(k)$ и $\mathbf{x}_N(k-L)$. Это приводит к примерно двукратному увеличению вычислительной сложности таких алгоритмов по сравнению с алгоритмами с возрастающим окном. Вариант RLS-алгоритма (см. табл. 6.1) со скользящим окном, представлен в табл. 14.6.

Таблица 14.6. Многоканальный RLS-алгоритм со скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0)=\mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1)=\mathbf{0}_N, d(0)=0, \dots,$	(14.6.0)
$d(0-L+1)=0, \mathbf{R}_{N,x_D}^{-1}(0)=\delta^{-2} \mathbf{\Lambda}_N, \mathbf{h}_{N,x_D}(0)=\mathbf{0}_N,$	
$\mathbf{\Lambda}_N=\text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}, m=1, \dots, M$	
For $k=1, 2, \dots, K$	
$\mathbf{g}_{N,x_U}(k)=\frac{\mathbf{R}_{N,x_U}^{-1}(k-1)\mathbf{x}_N(k)}{\lambda+\mathbf{x}_N^H(k)\mathbf{R}_{N,x_U}^{-1}(k-1)\mathbf{x}_N(k)}$	(14.6.1)
$\mathbf{R}_{N,x_U}^{-1}(k)=\lambda^{-1}[\mathbf{R}_{N,x_U}^{-1}(k-1)-\mathbf{g}_{N,x_U}(k)\mathbf{x}_N^H(k)\mathbf{R}_{N,x_U}^{-1}(k-1)]$	(14.6.2)
$\mathbf{g}_{N,x_D}(k)=\frac{\mathbf{R}_{N,x_D}^{-1}(k)\mathbf{x}_N(k-L)}{1-\mu\mathbf{x}_N^H(k-L)\mathbf{R}_{N,x_D}^{-1}(k)\mathbf{x}_N(k-L)}$	(14.6.3)
$\mathbf{R}_{N,x_D}^{-1}(k)=\mathbf{R}_{N,x_U}^{-1}(k)+\mu\mathbf{g}_{N,x_D}(k)\mathbf{x}_N^H(k-L)\mathbf{R}_{N,x_U}^{-1}(k)$	(14.6.4)
$\alpha_{N,x_U}(k)=d(k)-\mathbf{h}_{N,x_U}^H(k-1)\mathbf{x}_N(k)$	(14.6.5)
$\mathbf{h}_{N,x_U}(k)=\mathbf{h}_{N,x_U}(k-1)+\mathbf{g}_{N,x_U}(k)\alpha_{N,x_U}^*(k)$	(14.6.6)
$\alpha_{N,x_D}(k)=d(k-L)-\mathbf{h}_{N,x_D}^H(k)\mathbf{x}_N(k-L)$	(14.6.7)
$\mathbf{h}_{N,x_D}(k)=\mathbf{h}_{N,x_D}(k)-\mu\mathbf{g}_{N,x_D}(k)\alpha_{N,x_D}^*(k)$	(14.6.8)
End for k	

Вектор ВК адаптивного фильтра $\mathbf{h}_{N,x_D}(k)$ на каждой k -й итерации вычисляется в два приема с помощью уравнений (14.6.6) и (14.6.8), где коэффициенты Калмана $\mathbf{g}_{N,x_U}(k)$ и $\mathbf{g}_{N,x_D}(k)$ определяются как $\mathbf{g}_{N,x_U}(k)=\mathbf{R}_{N,x_U}^{-1}(k)\mathbf{x}_N(k)$ и $\mathbf{g}_{N,x_D}(k)=\mathbf{R}_{N,x_D}^{-1}(k)\mathbf{x}_N(k-L)$.

Алгоритм (см. табл. 14.6) имеет вычислительную сложность, которая, подобно алгоритму (см. табл. 14.4), примерно в два раза превышает сложность однотипного алгоритма с возрастающим окном (см. табл. 6.1).

При реализации алгоритма (см. табл. 14.6), а также других RLS-алгоритмов со скользящим окном требуется L слов памяти для хранения отсчетов сигнала $d(k)$ и $ML+\sum_{m=1}^M N_m$ слов памяти для хранения отсчетов сигналов $x_m(k)$.

Пример LC RLS-алгоритма с двойным ограничением и скользящим окном приведен в табл. 14.7.



Таблица 14.7. Многоканальный LC RLS-алгоритм с двойным ограничением и скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1) = \mathbf{0}_N, d(0) = 0, \dots,$ $d(0-L+1) = 0, \mathbf{R}_{N,x_D}^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N, \mathbf{\Gamma}_{N,x_D}(0) = \mathbf{R}_{N,x_D}^{-1}(0) \mathbf{C}_{NJ},$ $\mathbf{Q}_{N,x_D}(0) = \mathbf{\Gamma}_{N,x_D}(0) [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{N,x_D}(0)]^{-1}, \mathbf{h}_{N,x_D}(0) = \mathbf{Q}_{N,x_D}(0) \mathbf{f}_J,$ $\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}, m=1, \dots, M$	(14.7.0)
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_{N,x_U}(k)$, см., например, (14.6.1), (14.6.2)	(14.7.1)
Вычисление $\mathbf{g}_{N,x_D}(k)$, см., например, (14.6.3), (14.6.4)	(14.7.2)
$\mathbf{v}_{J,x_U}(k) = \mathbf{C}_{NJ}^H \mathbf{g}_{N,x_U}(k)$	(14.7.3)
$\mathbf{v}_{J,x_U}^H(k) = \mathbf{x}_N^H(k) \mathbf{Q}_{N,x_D}(k-1)$	(14.7.4)
$\mathbf{Q}'_{N,x_U}(k) = [\mathbf{Q}_{N,x_D}(k-1) - \mathbf{g}_{N,x_U}(k) \mathbf{v}_{J,x_U}^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J,x_U}(k) \mathbf{v}_{J,x_U}^H(k)}{1 - \mathbf{v}_{J,x_U}^H(k) \mathbf{v}_{J,x_U}(k)} \right]$	(14.7.5)
$\mathbf{Q}_{N,x_U}(k) = \mathbf{Q}'_{N,x_U}(k) + \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{N,x_U}(k)]$	(14.7.6)
$\mathbf{v}_{J,x_D}(k) = \mathbf{C}_{NJ}^H \mathbf{g}_{N,x_D}(k)$	(14.7.7)
$\mathbf{v}_{J,x_D}^H(k) = \mu \mathbf{x}_N^H(k-L) \mathbf{Q}_{N,x_U}(k)$	(14.7.8)
$\mathbf{Q}'_{N,x_D}(k) = [\mathbf{Q}_{N,x_U}(k) + \mathbf{g}_{N,x_D}(k) \mathbf{v}_{J,x_D}^H(k)] \left[\mathbf{I}_J - \frac{\mathbf{v}_{J,x_D}(k) \mathbf{v}_{J,x_D}^H(k)}{1 + \mathbf{v}_{J,x_D}^H(k) \mathbf{v}_{J,x_D}(k)} \right]$	(14.7.9)
$\mathbf{Q}_{N,x_D}(k) = \mathbf{Q}'_{N,x_D}(k) + \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{N,x_D}(k)]$	(14.7.10)
$\alpha_{N,x_U}(k) = d(k) - \mathbf{h}_{N,x_D}^H(k-1) \mathbf{x}_N(k)$	(14.7.11)
$\mathbf{h}'_{N,x_U}(k) = \mathbf{h}_{N,x_D}(k-1) + \mathbf{g}_{N,x_U}(k) \alpha_{N,x_U}^*(k)$	(14.7.12)
$\mathbf{h}_{N,x_U}(k) = \mathbf{h}'_{N,x_U}(k) + \mathbf{Q}_{N,x_U}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N,x_U}(k)]$	(14.7.13)
$\alpha_{N,x_D}(k) = d(k-L) - \mathbf{h}_{N,x_U}^H(k) \mathbf{x}_N(k-L)$	(14.7.14)
$\mathbf{h}'_{N,x_D}(k) = \mathbf{h}_{N,x_U}(k) - \mu \mathbf{g}_{N,x_D}(k) \alpha_{N,x_D}^*(k)$	(14.7.15)
$\mathbf{h}_{N,x_D}(k) = \mathbf{h}'_{N,x_D}(k) + \mathbf{Q}_{N,x_D}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N,x_D}(k)]$	(14.7.16)
End for k	

Оценка вычислительной сложности RLS-алгоритма (см. табл. 14.7) примерно в два раза больше оценки вычислительной сложности RLS-алгоритма (см. табл. 10.3).

В алгоритмах (см. табл. 14.6 и табл. 14.7) при формировании выходного сигнала используются ВК $\mathbf{h}_{N,x_D}(k)$, т.е. выходной сигнал формируется как $y_{N,x_U}(k) = \mathbf{h}_{N,x_D}^H(k-1) \mathbf{x}_N(k)$, а сигнал ошибки $\alpha_{N,x_U}(k)$ — как (14.7.11).

14.5. Одновременное использование скользящего окна и регуляризации в RLS-алгоритмах

При определении корреляционной матрицы на скользящем окне отсчетов эта матрица иногда может быть плохо обусловленной из-за того, что она оценивается на конечном числе отсчетов L . Поэтому динамическую регуляризацию корреляционной матрицы можно применить и в этом случае, для чего сигналы регуляризации, участвующие в формировании этой матрицы также взвешиваются, как показано на рис. 14.2 и рис. 14.3.

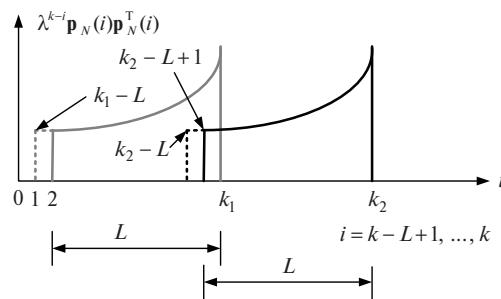


Рис. 14.3. Скользящее окно с экспоненциальным взвешиванием для составляющих корреляционной матрицы от сигналов регуляризации

Регуляризованная матрица, оцениваемая на скользящем окне, определяется как

$$\begin{aligned}
 \mathbf{R}_N(k) = \mathbf{R}_{N, \mathbf{p}_D}(k) = & \sum_{i=l-L+1}^k \lambda^{k-i} [\mathbf{x}_N(i) \mathbf{x}_N^H(i) + \xi^2 \mathbf{p}_N(i) \mathbf{p}_N^T(i)] = \lambda \mathbf{R}_{N, \mathbf{p}_D}(k-1) + \\
 & + \mathbf{x}_N(k) \mathbf{x}_N^H(k) - \mu \mathbf{x}_N(k-L) \mathbf{x}_N^H(k-L) + \xi^2 \mathbf{p}_N(k) \mathbf{p}_N^T(k) - \\
 & - \mu \xi^2 \mathbf{p}_N(k-L) \mathbf{p}_N^T(k-L) = \lambda \mathbf{R}_{N, \mathbf{x}_D}(k) - \mu \mathbf{x}_N(k-L) \mathbf{x}_N^H(k-L) + \\
 & + \xi^2 \mathbf{p}_N(k) \mathbf{p}_N^T(k) - \mu \xi^2 \mathbf{p}_N(k-L) \mathbf{p}_N^T(k-L) = \lambda \mathbf{R}_{N, \mathbf{x}_D}(k) + \xi^2 \mathbf{p}_N(k) \mathbf{p}_N^T(k) - \\
 & - \mu \xi^2 \mathbf{p}_N(k-L) \mathbf{p}_N^T(k-L) = \lambda \mathbf{R}_{N, \mathbf{p}_U}(k) - \mu \xi^2 \mathbf{p}_N(k-L) \mathbf{p}_N^T(k-L).
 \end{aligned} \tag{14.14}$$

Из уравнения (14.14) следует, что на каждой k -й итерации адаптивного алгоритма корреляционная матрица модифицируется четыре раза: два раза за счет включения в ее оценку входных сигналов адаптивного фильтра $\mathbf{x}_N(k)$ и исключения из этой оценки сигналов $\mathbf{x}_N(k-L)$ и два раза за счет включения в оценку искусственных сигналов регуляризации $\xi \mathbf{p}_N(k)$ и исключения из этой оценки сигналов $\xi \mathbf{p}_N(k-L)$.

Следствием такой модификации корреляционной матрицы является увеличение примерно в два раза вычислительной сложности регуляризированных алгоритмов со скользящим окном по сравнению с нерегуляризованными ал-

горитмами со скользящим окном и в четыре раза по сравнению с нерегуляризованными алгоритмами с возрастающим окном.

Регуляризованная версия RLS-алгоритма со скользящим окном (см. табл. 14.6) приведена в табл. 14.8, а регуляризованные версии алгоритма (см. табл. 14.7) — в табл. 14.9.

Таблица 14.8. Многоканальный регуляризованный RLS-алгоритм со скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1) = \mathbf{0}_N, \xi \mathbf{p}_N(0) = \mathbf{0}_N, \dots,$ $\xi \mathbf{p}_N(0-L+1) = \mathbf{0}_N, d(0) = 0, \dots, d(0-L+1) = 0, \mathbf{R}_{N,\mathbf{p}_D}^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N,$ $\mathbf{h}_{N,\mathbf{p}_D}(0) = \mathbf{0}_N,$ $\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_{m-1}-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}, m=1, \dots, M$	(14.8.0)
For $k = 1, 2, \dots, K$	
$\mathbf{g}_{N,\mathbf{x}_U}(k) = \frac{\mathbf{R}_{N,\mathbf{p}_D}^{-1}(k-1)\mathbf{x}_N(k)}{\lambda + \mathbf{x}_N^H(k)\mathbf{R}_{N,\mathbf{p}_D}^{-1}(k-1)\mathbf{x}_N(k)}$	(14.8.1)
$\mathbf{R}_{N,\mathbf{x}_U}^{-1}(k) = \lambda^{-1} \left[\mathbf{R}_{N,\mathbf{p}_D}^{-1}(k-1) - \mathbf{g}_{N,\mathbf{x}_U}(k)\mathbf{x}_N^H(k)\mathbf{R}_{N,\mathbf{p}_D}^{-1}(k-1) \right]$	(14.8.2)
$\mathbf{g}_{N,\mathbf{x}_D}(k) = \frac{\mathbf{R}_{N,\mathbf{x}_U}^{-1}(k)\mathbf{x}_N(k-L)}{1 - \mu \mathbf{x}_N^H(k-L)\mathbf{R}_{N,\mathbf{x}_U}^{-1}(k)\mathbf{x}_N(k-L)}$	(14.8.3)
$\mathbf{R}_{N,\mathbf{x}_D}^{-1}(k) = \mathbf{R}_{N,\mathbf{x}_U}^{-1}(k) + \mu \mathbf{g}_{N,\mathbf{x}_D}(k)\mathbf{x}_N^H(k-L)\mathbf{R}_{N,\mathbf{x}_U}^{-1}(k)$	(14.8.4)
$\mathbf{g}_{N,\mathbf{p}_U}(k) = \frac{\mathbf{R}_{N,\mathbf{x}_D}^{-1}(k)\xi \mathbf{p}_N(k)}{1 + \xi \mathbf{p}_N^T(k)\mathbf{R}_{N,\mathbf{x}_D}^{-1}(k)\xi \mathbf{p}_N(k)}$	(14.8.5)
$\mathbf{R}_{N,\mathbf{p}_U}^{-1}(k) = \mathbf{R}_{N,\mathbf{x}_D}^{-1}(k) - \mathbf{g}_{N,\mathbf{p}_U}(k)\xi \mathbf{p}_N^T(k)\mathbf{R}_{N,\mathbf{x}_D}^{-1}(k)$	(14.8.6)
$\mathbf{g}_{N,\mathbf{p}_D}(k) = \frac{\mathbf{R}_{N,\mathbf{p}_U}^{-1}(k)\xi \mathbf{p}_N(k-L)}{1 - \mu \xi \mathbf{p}_N^T(k-L)\mathbf{R}_{N,\mathbf{p}_U}^{-1}(k)\xi \mathbf{p}_N(k-L)}$	(14.8.7)
$\mathbf{R}_{N,\mathbf{p}_D}^{-1}(k) = \mathbf{R}_{N,\mathbf{p}_U}^{-1}(k) + \mu \mathbf{g}_{N,\mathbf{p}_D}(k)\xi \mathbf{p}_N^T(k-L)\mathbf{R}_{N,\mathbf{p}_U}^{-1}(k)$	(14.8.8)
$\alpha_{N,\mathbf{x}_U}(k) = d(k) - \mathbf{h}_{N,\mathbf{p}_D}^H(k-1)\mathbf{x}_N(k)$	(14.8.9)
$\mathbf{h}_{N,\mathbf{x}_U}(k) = \mathbf{h}_{N,\mathbf{x}_D}(k-1) + \mathbf{g}_{N,\mathbf{x}_U}(k)\alpha_{N,\mathbf{x}_U}^*(k)$	(14.8.10)
$\alpha_{N,\mathbf{x}_D}(k) = d(k-L) - \mathbf{h}_{N,\mathbf{x}_U}^H(k)\mathbf{x}_N(k-L)$	(14.8.11)
$\mathbf{h}_{N,\mathbf{x}_D}(k) = \mathbf{h}_{N,\mathbf{x}_U}(k) - \mu \mathbf{g}_{N,\mathbf{x}_D}(k)\alpha_{N,\mathbf{x}_D}^*(k)$	(14.8.12)
$\alpha_{N,\mathbf{p}_U}(k) = -\mathbf{h}_{N,\mathbf{x}_D}^H(k)\xi \mathbf{p}_N(k)$	(14.8.13)
$\mathbf{h}_{N,\mathbf{p}_U}(k) = \mathbf{h}_{N,\mathbf{x}_D}(k) + \mathbf{g}_{N,\mathbf{p}_U}(k)\alpha_{N,\mathbf{p}_U}^*(k)$	(14.8.14)
$\alpha_{N,\mathbf{p}_D}(k) = -\mathbf{h}_{N,\mathbf{p}_U}^H(k)\xi \mathbf{p}_N(k-L)$	(14.8.15)
$\mathbf{h}_{N,\mathbf{p}_D}(k) = \mathbf{h}_{N,\mathbf{p}_U}(k) - \mu \mathbf{g}_{N,\mathbf{p}_D}(k)\alpha_{N,\mathbf{p}_D}^*(k)$	(14.8.16)
End for k	

Таблица 14.9. Многоканальный регуляризованный LC RLS-алгоритм с двойным ограничением и скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1) = \mathbf{0}_N, \xi \mathbf{p}_N(0) = \mathbf{0}_N, \dots,$ $\xi \mathbf{p}_N(0-L+1) = \mathbf{0}_N, d(0) = 0, \dots, d(0-L+1) = 0, \mathbf{R}_{N, \mathbf{p}_D}^{-1}(0) = \delta^{-2} \mathbf{A}_N,$ $\mathbf{G}_{N, \mathbf{p}_D}(0) = \mathbf{R}_{N, \mathbf{p}_D}^{-1}(0) \mathbf{C}_{N, \mathbf{p}_D}, \mathbf{Q}_{N, \mathbf{p}_D}(0) = \mathbf{G}_{N, \mathbf{p}_D}(0) [\mathbf{C}_{N, \mathbf{p}_D}^H \mathbf{G}_{N, \mathbf{p}_D}(0)]^{-1},$ $\mathbf{h}_{N, \mathbf{p}_D}(0) = \mathbf{Q}_{N, \mathbf{p}_D}(0) \mathbf{f}_j,$ $\Lambda_N = \text{diag} \{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}, m=1, \dots, M$	(14.9.0)
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{g}_{N, \mathbf{x}_U}(k)$, см., например, (14.8.1), (14.8.2)	(14.9.1)
Вычисление $\mathbf{g}_{N, \mathbf{x}_D}(k)$, см., например, (14.8.3), (14.8.4)	(14.9.2)
Вычисление $\mathbf{g}_{N, \mathbf{p}_U}(k)$, см., например, (14.8.5), (14.8.6)	(14.9.3)
Вычисление $\mathbf{g}_{N, \mathbf{p}_D}(k)$, см., например, (14.8.7), (14.8.8)	(14.9.4)
$\mathbf{v}_{J, \mathbf{x}_U}(k) = \mathbf{C}_{N, \mathbf{x}_U}^H \mathbf{g}_{N, \mathbf{x}_U}(k)$	(14.9.5)
$\mathbf{v}_{J, \mathbf{x}_U}^H(k) = \mathbf{x}_N^H(k) \mathbf{Q}_{N, \mathbf{x}_U}(k-1)$	(14.9.6)
$\mathbf{Q}'_{N, \mathbf{x}_U}(k) = [\mathbf{Q}_{N, \mathbf{x}_U}(k-1) - \mathbf{g}_{N, \mathbf{x}_U}(k) \mathbf{v}_{J, \mathbf{x}_U}^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J, \mathbf{x}_U}(k) \mathbf{v}_{J, \mathbf{x}_U}^H(k)}{1 - \mathbf{v}_{J, \mathbf{x}_U}^H(k) \mathbf{v}_{J, \mathbf{x}_U}(k)} \right]$	(14.9.7)
$\mathbf{Q}_{N, \mathbf{x}_U}(k) = \mathbf{Q}'_{N, \mathbf{x}_U}(k) + \mathbf{C}_{N, \mathbf{x}_U} (\mathbf{C}_{N, \mathbf{x}_U}^H \mathbf{C}_{N, \mathbf{x}_U})^{-1} [\mathbf{I}_J - \mathbf{C}_{N, \mathbf{x}_U}^H \mathbf{Q}'_{N, \mathbf{x}_U}(k)]$	(14.9.8)
$\mathbf{v}_{J, \mathbf{x}_D}(k) = \mathbf{C}_{N, \mathbf{x}_D}^H \mathbf{g}_{N, \mathbf{x}_D}(k)$	(14.9.9)
$\mathbf{v}_{J, \mathbf{x}_D}^H(k) = \mu \mathbf{x}_N^H(k-L) \mathbf{Q}_{N, \mathbf{x}_D}(k)$	(14.9.10)
$\mathbf{Q}'_{N, \mathbf{x}_D}(k) = [\mathbf{Q}_{N, \mathbf{x}_D}(k) + \mathbf{g}_{N, \mathbf{x}_D}(k) \mathbf{v}_{J, \mathbf{x}_D}^H(k)] \left[\mathbf{I}_J - \frac{\mathbf{v}_{J, \mathbf{x}_D}(k) \mathbf{v}_{J, \mathbf{x}_D}^H(k)}{1 + \mathbf{v}_{J, \mathbf{x}_D}^H(k) \mathbf{v}_{J, \mathbf{x}_D}(k)} \right]$	(14.9.11)
$\mathbf{Q}_{N, \mathbf{x}_D}(k) = \mathbf{Q}'_{N, \mathbf{x}_D}(k) + \mathbf{C}_{N, \mathbf{x}_D} (\mathbf{C}_{N, \mathbf{x}_D}^H \mathbf{C}_{N, \mathbf{x}_D})^{-1} [\mathbf{I}_J - \mathbf{C}_{N, \mathbf{x}_D}^H \mathbf{Q}'_{N, \mathbf{x}_D}(k)]$	(14.9.12)
$\mathbf{v}_{J, \mathbf{p}_U}(k) = \mathbf{C}_{N, \mathbf{p}_U}^H \mathbf{g}_{N, \mathbf{p}_U}(k)$	(14.9.13)
$\mathbf{v}_{J, \mathbf{p}_U}^H(k) = \xi \mathbf{p}_N^T(k) \mathbf{Q}_{N, \mathbf{p}_U}(k)$	(14.9.14)
$\mathbf{Q}'_{N, \mathbf{p}_U}(k) = [\mathbf{Q}_{N, \mathbf{p}_U}(k) - \mathbf{g}_{N, \mathbf{p}_U}(k) \mathbf{v}_{J, \mathbf{p}_U}^H(k)] \left[\mathbf{I}_J + \frac{\mathbf{v}_{J, \mathbf{p}_U}(k) \mathbf{v}_{J, \mathbf{p}_U}^H(k)}{1 - \mathbf{v}_{J, \mathbf{p}_U}^H(k) \mathbf{v}_{J, \mathbf{p}_U}(k)} \right]$	(14.9.15)
$\mathbf{Q}_{N, \mathbf{p}_U}(k) = \mathbf{Q}'_{N, \mathbf{p}_U}(k) + \mathbf{C}_{N, \mathbf{p}_U} (\mathbf{C}_{N, \mathbf{p}_U}^H \mathbf{C}_{N, \mathbf{p}_U})^{-1} [\mathbf{I}_J - \mathbf{C}_{N, \mathbf{p}_U}^H \mathbf{Q}'_{N, \mathbf{p}_U}(k)]$	(14.9.16)
$\mathbf{v}_{J, \mathbf{p}_D}(k) = \mathbf{C}_{N, \mathbf{p}_D}^H \mathbf{g}_{N, \mathbf{p}_D}(k)$	(14.9.17)
$\mathbf{v}_{J, \mathbf{p}_D}^H(k) = \mu \xi \mathbf{p}_N^T(k-L) \mathbf{Q}_{N, \mathbf{p}_D}(k)$	(14.9.18)
$\mathbf{Q}'_{N, \mathbf{p}_D}(k) = [\mathbf{Q}_{N, \mathbf{p}_D}(k) + \mathbf{g}_{N, \mathbf{p}_D}(k) \mathbf{v}_{J, \mathbf{p}_D}^H(k)] \left[\mathbf{I}_J - \frac{\mathbf{v}_{J, \mathbf{p}_D}(k) \mathbf{v}_{J, \mathbf{p}_D}^H(k)}{1 + \mathbf{v}_{J, \mathbf{p}_D}^H(k) \mathbf{v}_{J, \mathbf{p}_D}(k)} \right]$	(14.9.19)

Таблица 14.9. (окончание)

Вычисления	Ссылки
$\mathbf{Q}_{N, \mathbf{p}_D}(k) = \mathbf{Q}'_{N, \mathbf{p}_D}(k) + \mathbf{C}_{NJ} \left(\mathbf{C}_{NJ}^H \mathbf{C}_{NJ} \right)^{-1} \left[\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{N, \mathbf{p}_D}(k) \right]$	(14.9.20)
$\alpha_{N, \mathbf{x}_U}(k) = d(k) - \mathbf{h}_{N, \mathbf{p}_D}^H(k-1) \mathbf{x}_N(k)$	(14.9.21)
$\mathbf{h}'_{N, \mathbf{x}_U}(k) = \mathbf{h}_{N, \mathbf{p}_D}(k-1) + \mathbf{g}_{N, \mathbf{x}_U}(k) \alpha_{N, \mathbf{x}_U}^*(k)$	(14.9.22)
$\mathbf{h}_{N, \mathbf{x}_U}(k) = \mathbf{h}'_{N, \mathbf{x}_U}(k) + \mathbf{Q}_{N, \mathbf{x}_U}(k) \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N, \mathbf{x}_U}(k) \right]$	(14.9.23)
$\alpha_{N, \mathbf{x}_D}(k) = d(k-L) - \mathbf{h}_{N, \mathbf{x}_U}^H(k) \mathbf{x}_N(k-L)$	(14.9.24)
$\mathbf{h}'_{N, \mathbf{x}_D}(k) = \mathbf{h}_{N, \mathbf{x}_U}(k) - \mu \mathbf{g}_{N, \mathbf{x}_D}(k) \alpha_{N, \mathbf{x}_D}^*(k)$	(14.9.25)
$\mathbf{h}_{N, \mathbf{x}_D}(k) = \mathbf{h}'_{N, \mathbf{x}_D}(k) + \mathbf{Q}_{N, \mathbf{x}_D}(k) \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N, \mathbf{x}_D}(k) \right]$	(14.9.26)
$\alpha_{N, \mathbf{p}_U}(k) = -\mathbf{h}_{N, \mathbf{x}_D}^H(k) \xi \mathbf{p}_N(k)$	(14.9.27)
$\mathbf{h}'_{N, \mathbf{p}_U}(k) = \mathbf{h}_{N, \mathbf{x}_D}(k) + \mathbf{g}_{N, \mathbf{p}_U}(k) \alpha_{N, \mathbf{p}_U}^*(k)$	(14.9.28)
$\mathbf{h}_{N, \mathbf{p}_U}(k) = \mathbf{h}'_{N, \mathbf{p}_U}(k) + \mathbf{Q}_{N, \mathbf{p}_U}(k) \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N, \mathbf{p}_U}(k) \right]$	(14.9.29)
$\alpha_{N, \mathbf{p}_D}(k) = -\mathbf{h}_{N, \mathbf{p}_U}^H(k) \xi \mathbf{p}_N(k-L)$	(14.9.30)
$\mathbf{h}'_{N, \mathbf{p}_D}(k) = \mathbf{h}_{N, \mathbf{p}_U}(k) - \mu \mathbf{g}_{N, \mathbf{p}_D}(k) \alpha_{N, \mathbf{p}_D}^*(k)$	(14.9.31)
$\mathbf{h}_{N, \mathbf{p}_D}(k) = \mathbf{h}'_{N, \mathbf{p}_D}(k) + \mathbf{Q}_{N, \mathbf{p}_D}(k) \left[\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_{N, \mathbf{p}_D}(k) \right]$	(14.9.32)
End for k	

В алгоритмах (см. табл. 14.8 и табл. 14.9) при формировании наблюдаемого выходного сигнала используются ВК $\mathbf{h}_{N, \mathbf{p}_D}(k)$, т. е. выходной сигнал формируется как $y_{N, \mathbf{x}_U}(k) = \mathbf{h}_{N, \mathbf{p}_D}^H(k-1) \mathbf{x}_N(k)$, а наблюдаемый сигнал ошибки $\alpha_{N, \mathbf{x}_U}(k)$ — как (14.9.21).

Ниже приводятся результаты моделирования, демонстрирующие эффективность применения разных способов оценки корреляционной матрицы при обработке нестационарных сигналов. Рассматривается задача идентификации трехканального линейного импульсного отклика с числом ВК в каналах N_m , равным 8, 16 и 32, при обработке различных нестационарных сигналов речи в каждом из каналов.

На рис. 14.4 сравниваются многоканальные алгоритмы с возрастающим (кривая 1) и скользящим (кривая 2) окнами без регуляризации. На рис. 14.4а можно увидеть, что ограничение 0 дБ АЧХ $|H(f)|$ аддитивного фильтра, задаваемое на выбранных частотах 1 и 2 кГц, обеспечивается обоими алгоритмами. На этом рисунке вертикальные стрелки указывают на частоты ограничений, а горизонтальная пунктирная линия обозначает уровень ограничения АЧХ. Однако в алгоритме со скользящим окном (благодаря следящим свойствам) такой параметр, как ERLE, достигает более высокого значения, чем в алгоритмах с возрастающим окном. Из рис. 14.4б видно, что в рассмотренных примерах достигаемые значения параметра ERLE для RLS-алгоритма с возрастающим

окном примерно на 20 дБ ниже, чем для RLS-алгоритма со скользящим окном. Это объясняется тем, что при обработке нестационарных сигналов во втором случае корреляционная матрица вычисляется более точно.

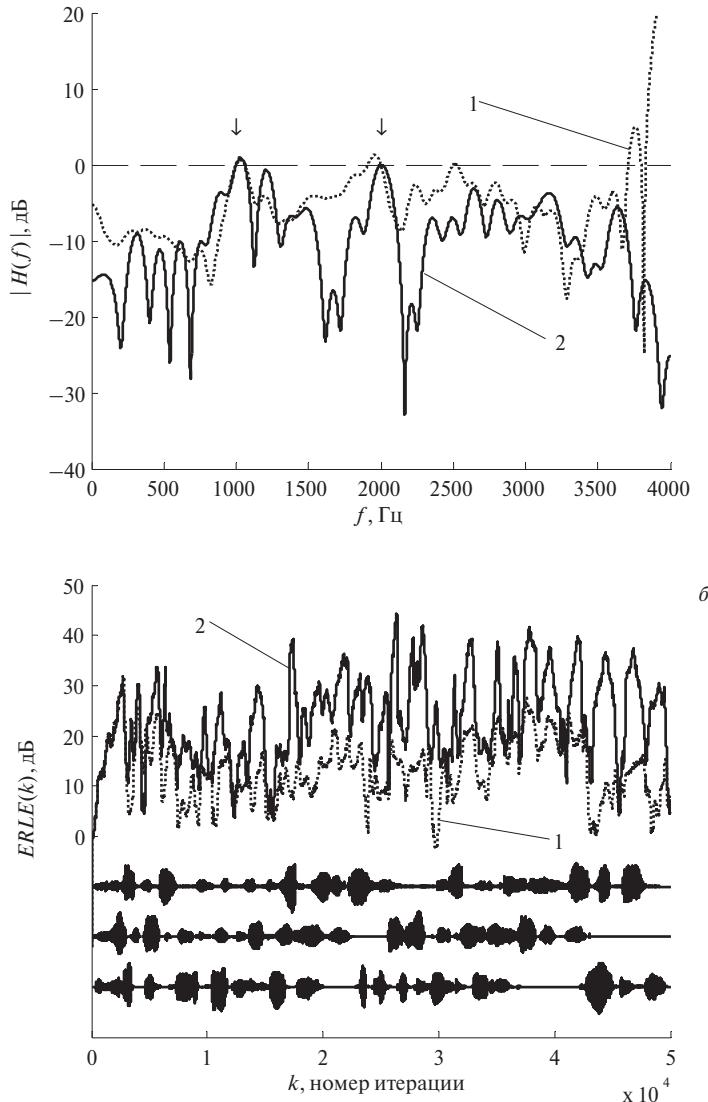


Рис. 14.4. Результаты моделирования: а) АЧХ; б) параметр ERLE

В рассмотренных примерах длина скользящего окна L равнялась 256 отсчетам, что при частоте дискретизации, равной 8 кГц, примерно соответствует 30 мс (интервалу стационарности речевых сигналов). Значения ERLE также оценивались на скользящем окне $B = 256$ как

$$ERLE(k) = 10 \lg \left(\sum_{i=k-B+1}^k d^2(i) \Big/ \sum_{i=k-B+1}^k \alpha_N^2(i) \right). \quad (14.15)$$

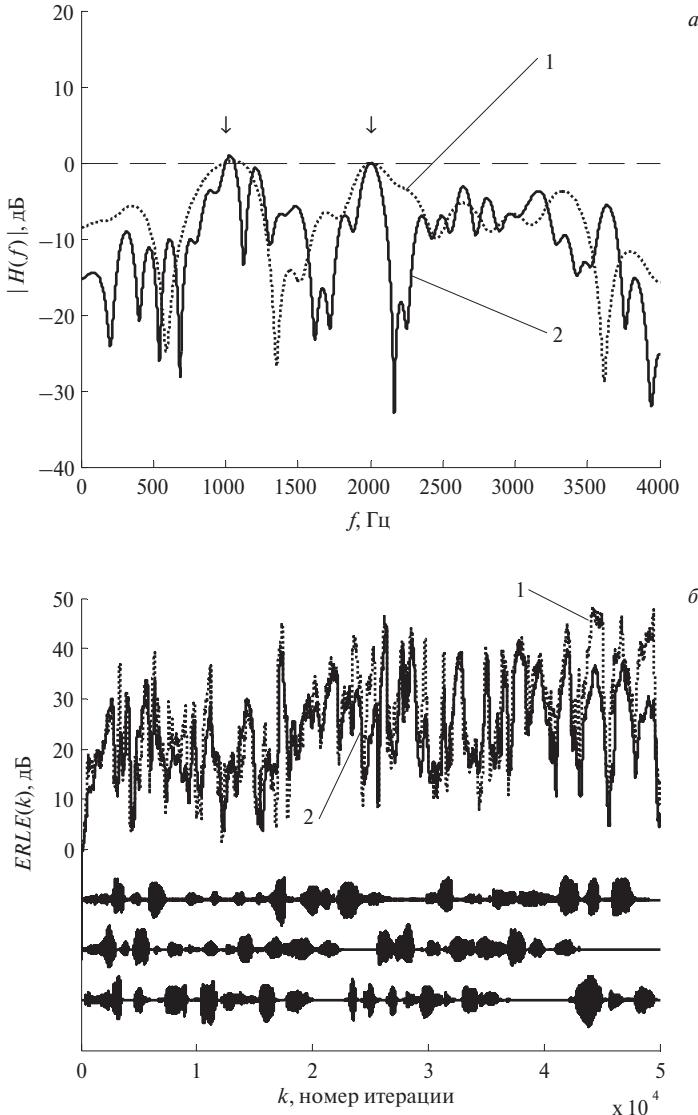


Рис. 14.5. Результаты моделирования: а) АЧХ; б) параметр ERLE

Повышение эффективности RLS-алгоритмов достигается при использовании динамической регуляризации корреляционной матрицы, оцениваемой на скользящем окне (см. рис. 14.5). Из рис. 14.5а видно, что заданные ограничения АЧХ также выполняются в случае обоих алгоритмов, а из

рис. 14.5б следует, что при использовании скользящего окна и регуляризации (кривая 1) достигаемое значение ERLE в среднем не меньше, чем при использовании только скользящего окна (кривая 2, аналогичная кривой 2 рис. 14.4б).

14.6. Параллельные RLS-алгоритмы, допускающие вычисления с помощью двух или четырех процессоров

Возможность реализации того или иного алгоритма ЦОС определяется производительностью используемых ЦСП. Производительность современных ЦСП уже позволяет эффективно реализовывать вычислительно сложные алгоритмы, к которым относятся и RLS-алгоритмы адаптивной фильтрации [43, 44]. Кроме того, современные СБИС могут содержать несколько ЦСП в одном кристалле [44]. Это позволяет создавать компактные устройства ЦОС, функционирующие на основе параллельных вычислений. RLS-алгоритмы адаптивной фильтрации также могут строиться на основе параллельных вычислительных процедур.

Наиболее привлекательным является распараллеливание вычислений, связанных с существующими потоками независимых данных. В этом случае отсутствуют дополнительные вычислительные затраты для разделения данных на параллельные потоки. В RLS-алгоритмах со скользящим окном и в регуляризованных RLS-алгоритмах с возрастающим окном имеется два ($F = 2$), а в регуляризованных RLS-алгоритмах со скользящим окном — четыре ($F = 4$) независимых потока входных данных $\mathbf{x}_N(k)$, $\mathbf{x}_N(k-L)$, $\xi\mathbf{p}_N(k)$ и $\xi\mathbf{p}_N(k-L)$, обусловленных модификацией корреляционной матрицы адаптивного фильтра. В последовательных RLS-алгоритмах эти потоки обрабатываются последовательно. Рассматриваемые далее параллельные RLS-алгоритмы основаны на возможности независимой параллельной обработки таких потоков.

Идея параллельных вычислений была высказана в работе [182] при построении RLS-алгоритмов для одноканального адаптивного фильтра со скользящим окном и действительными ВК. При получении этих алгоритмов было использовано достаточно сложное выражение для MIL, оцениваемой на скользящем окне.

Эта лемма получена путем последовательного двукратного применения обычной MIL (5.115). Если представить уравнение (14.12) как

$$\mathbf{A} = \mathbf{B} + \mathbf{x}\mathbf{x}^H - \mathbf{y}\mathbf{y}^H, \quad (14.16)$$

то результирующая лемма об обращении матрицы будет иметь вид

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1} \mathbf{x} \mathbf{x}^H \mathbf{B}^{-1} (1 - \mathbf{y}^H \mathbf{B}^{-1} \mathbf{y}) - \mathbf{B}^{-1} \mathbf{y} \mathbf{y}^H \mathbf{B}^{-1} (1 + \mathbf{x}^H \mathbf{B}^{-1} \mathbf{x})}{(1 - \mathbf{y}^H \mathbf{B}^{-1} \mathbf{y})(1 + \mathbf{x}^H \mathbf{B}^{-1} \mathbf{x}) + \mathbf{y}^H \mathbf{B}^{-1} \mathbf{x} \mathbf{x}^H \mathbf{B}^{-1} \mathbf{y}} - \\ - \frac{\mathbf{B}^{-1} \mathbf{x} \mathbf{x}^H \mathbf{B}^{-1} \mathbf{y} \mathbf{y}^H \mathbf{B}^{-1} + \mathbf{B}^{-1} \mathbf{y} \mathbf{y}^H \mathbf{B}^{-1} \mathbf{x} \mathbf{x}^H \mathbf{B}^{-1}}{(1 - \mathbf{y}^H \mathbf{B}^{-1} \mathbf{y})(1 + \mathbf{x}^H \mathbf{B}^{-1} \mathbf{x}) + \mathbf{y}^H \mathbf{B}^{-1} \mathbf{x} \mathbf{x}^H \mathbf{B}^{-1} \mathbf{y}}. \quad (14.17)$$

На основе выражения (14.17) был разработан ряд RLS- и быстрых RLS- алгоритмов для многоканальных адаптивных фильтров с неодинаковым числом комплексных ВК в каналах, включая линейно-ограниченные алгоритмы, см., например, [66]. Данные алгоритмы в случае скользящего окна или возрастающего окна и регуляризации могут быть реализованы с помощью двух ЦСП, а в случае скользящего окна и регуляризации — с помощью четырех ЦСП. Вычислительные процедуры таких алгоритмов довольно громоздки, что обусловлено сложностью выражения (14.17). Поэтому в данной книге они не приводятся.

Ниже рассматривается более простой способ описания RLS- и LC RLS- алгоритмов адаптивной фильтрации, позволяющий реализовать такие алгоритмы с помощью параллельных вычислений. При таком описании одноименные нерегуляризованные и регуляризованные алгоритмы со скользящим окном, а также регуляризованные алгоритмы с возрастающим окном можно представить с помощью одних и тех же векторно-матричных уравнений. Вычисления, связанные с независимыми потоками данных адаптивного фильтра $\mathbf{x}_N(k)$, $\mathbf{x}_N(k-L)$, $\xi \mathbf{p}_N(k)$ и $\xi \mathbf{p}_N(k-L)$, при этом выполняются параллельно, что отличает параллельные алгоритмы от последовательных алгоритмов, рассмотренных в главах 6—13, где такие потоки обрабатываются последовательно. Различия между последовательными и параллельными вычислениями в RLS-алгоритмах условно показаны на рис. 14.6—14.8.

В основе параллельных алгоритмов находится MIL (5.115) в виде

$$\mathbf{R}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1} \mathbf{C} \mathbf{A}^{-1} \mathbf{D} \mathbf{B}^{-1}, \quad (14.18)$$

где $\mathbf{A} = \mathbf{D} \mathbf{B}^{-1} \mathbf{C} + \mathbf{S}$, а \mathbf{C} и \mathbf{D} — матрицы. Для использования уравнения (14.18) при вычислении обратной корреляционной матрицы необходимо сформировать следующие матрицы:

$$\mathbf{C} = [\mathbf{y}, \mathbf{x}, \mathbf{z}, \mathbf{v}] = \mathbf{X}_{NF}(k) = [\mu^{0.5} \mathbf{x}_N(k-L), \mathbf{x}_N(k), \mu^{0.5} \xi \mathbf{p}_N(k-L), \xi \mathbf{p}_N(k)], \quad (14.19)$$

$$\mathbf{D} = \mathbf{C}^H \quad (14.20)$$

и

$$\mathbf{S} = \text{diag}\{-1, 1, -1, 1\}. \quad (14.21)$$

Число столбцов F матрицы \mathbf{X}_{NF} и их вид определяют возможные разновидности RLS-алгоритмов. Порядок столбцов в уравнении (14.19) не имеет значения. Важно лишь то, что этот порядок должен быть согласован с элементами диаго-

нальной матрицы \mathbf{S} . Номера столбцов матрицы \mathbf{C} , соответствующих векторам с индексом дискретного времени $(k-L)$, например i и j , должны соответствовать диагональным элементам матрицы \mathbf{S} , равным (-1) , т. е. $s_{ii}=-1$ и $s_{jj}=-1$.

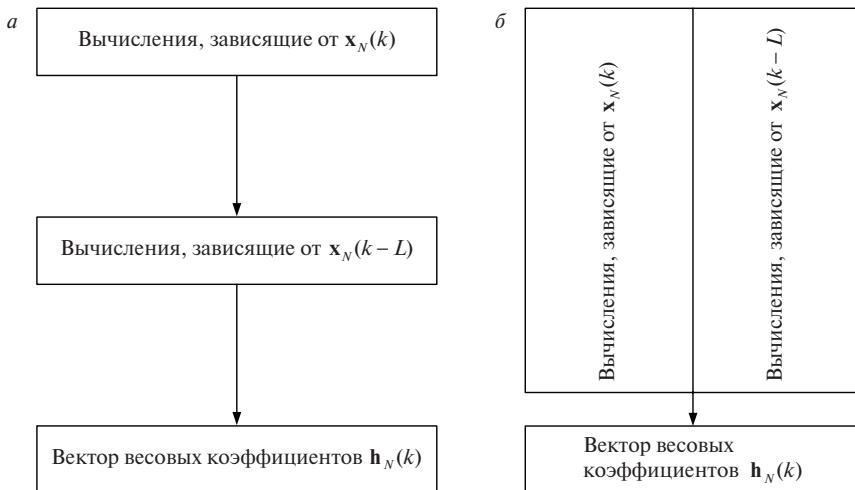


Рис. 14.6. Вычисления в RLS-алгоритмах со скользящим окном: а) последовательные; б) параллельные

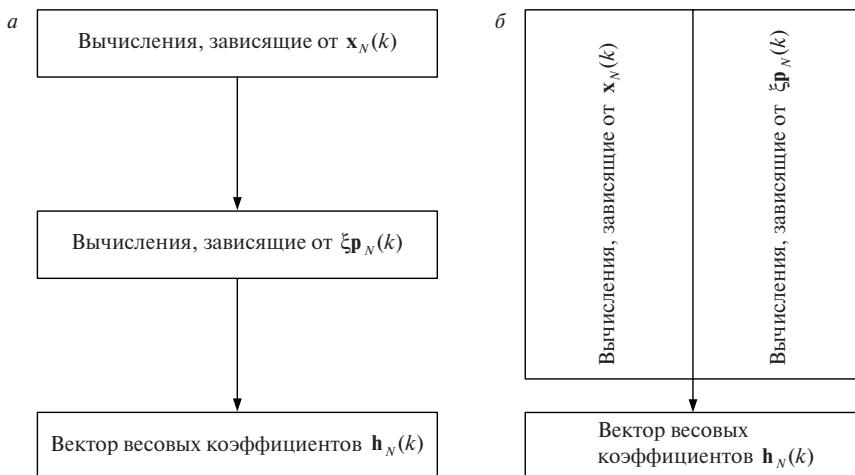


Рис. 14.7. Вычисления в регуляризованных RLS-алгоритмах с возрастающим окном: а) последовательные; б) параллельные

При использовании всех столбцов в уравнениях (14.19)–(14.21) можно получить параллельные версии регуляризованных RLS-алгоритмов со скользящим окном, при использовании первых двух столбцов — параллельные версии алгоритмов со скользящим окном без регуляризации, при использовании

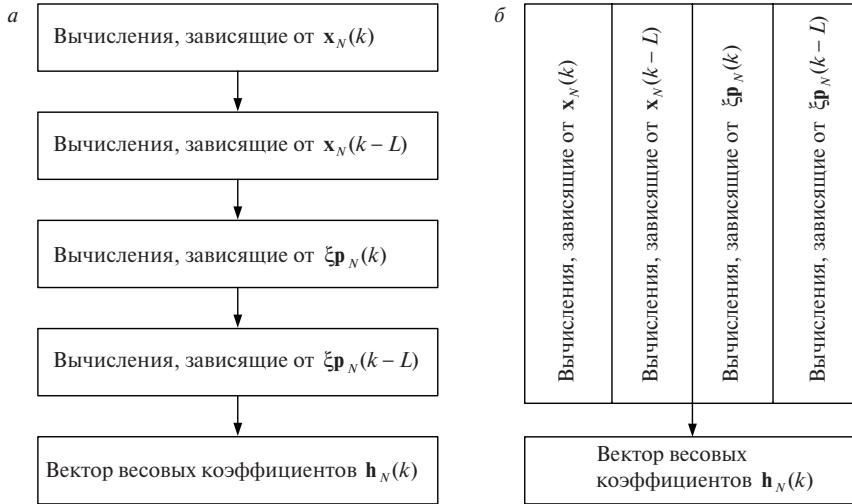


Рис. 14.8. Вычисления в регуляризованных RLS-алгоритмах со скользящим окном: а) последовательные; б) параллельные

второго и четвертого столбцов — параллельные версии регуляризованных алгоритмов с возрастающим окном. При $F=1$ и использовании второго столбца матрицы \mathbf{C} рассматриваемые в настоящей главе алгоритмы совпадают с нерегуляризованными алгоритмами с возрастающим окном.

Таким образом, используя выражение (14.19), любую из корреляционных матриц (14.9), (14.12) и (14.14) можно представить как

$$\mathbf{R}_N(k) = \lambda \mathbf{R}_N(k-1) + \mathbf{C}_{NF} \mathbf{S}_F \mathbf{C}_{NF}^H \quad (14.22)$$

и для рекурсивного обращения такой матрицы применить уравнение (14.18). Затем, используя при этом преобразования, аналогичные рассмотренным в главе 6, можно получить параллельную версию алгоритмов (см. табл. 14.4, табл. 14.6 и табл. 14.8).

Такой «универсальный» параллельный алгоритм представлен в табл. 14.10. Во многом он похож на одноименный алгоритм с возрастающим окном (см. табл. 6.1). Однако в параллельном алгоритме переменные $\mathbf{X}_{NF}(k)$, $\mathbf{G}_{NF}(k)$ — это матрицы, а не векторы. Кроме того, переменные $\mathbf{d}_F(k)$ и $\boldsymbol{\alpha}_F(k)$ являются векторами, а не скалярными величинами. Знаменатель в уравнении (14.10.1) является не скалярной величиной, а квадратной матрицей с числом элементов $F \times F$. Эта матрица обеспечивает математическую эквивалентность данного параллельного алгоритма, определяемого значением F и структурой матрицы $\mathbf{X}_{NF}(k)$, и соответствующих последовательных RLS-алгоритмов.

В алгоритме (см. табл. 14.10) вектор $\mathbf{d}_F(k)$ определяется как

$$\mathbf{d}_F(k) = \left[\mu^{0.5} d(k-L), d(k), 0, 0 \right], \quad (14.23)$$

а сигнал ошибки на выходе адаптивного фильтра — как $\alpha_{N,x}(k) = d(k) - \mathbf{h}_N^H(k-1)\mathbf{x}_N(k) = \alpha_F^{(2)}(k)$, где $\alpha_F^{(2)}(k)$ — второй элемент вектора $\alpha_F(k)$. Рассматриваемые в настоящем разделе векторы $\mathbf{d}_F(k)$ и $\alpha_F(k)$ — это вектор-строки. Матрица $\mathbf{G}_{NF}(k)$ представляет собой матрицу коэффициентов Калмана $\mathbf{G}_{NF}(k) = \mathbf{R}_N^{-1}(k)\mathbf{X}_{NF}(k)$.

Таблица 14.10. Параллельный многоканальный RLS-алгоритм со скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1) = \mathbf{0}_N, \xi \mathbf{p}_N(0) = \mathbf{0}_N, \dots,$	(14.10.0)
$\xi \mathbf{p}_N(0-L+1) = \mathbf{0}_N, d(0) = 0, \dots, d(0-L+1) = 0, \mathbf{X}_{NF}(0) = \mathbf{O}_{NF},$	
$\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N, \mathbf{h}_N(0) = \mathbf{0}_N,$	
$\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_{m-1}-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\},$	
$m=1, \dots, M$	
For $k = 1, 2, \dots, K$	
$\mathbf{G}_{NF}(k) = \mathbf{R}_N^{-1}(k-1)\mathbf{X}_{NF}(k) [\lambda \mathbf{S}_F + \mathbf{X}_{NF}^H(k)\mathbf{R}_N^{-1}(k-1)\mathbf{X}_{NF}(k)]^{-1}$	(14.10.1)
$\mathbf{R}_N^{-1}(k) = \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{G}_{NF}(k)\mathbf{X}_{NF}^H(k)\mathbf{R}_N^{-1}(k-1)]$	(14.10.2)
$\alpha_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1)\mathbf{X}_{NF}(k)$	(14.10.3)
$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{G}_{NF}(k)\alpha_F^H(k)$	(14.10.4)
End for k	

Оценка вычислительной сложности алгоритма (см. табл. 14.10) равна $O(N^2F)$ арифметическим операциям, необходимым для выполнения одной итерации алгоритма, что примерно совпадает с вычислительной сложностью одноименных последовательных (см. табл. 14.4, табл. 14.6 и табл. 14.8). В силу небольшого размера матрицы с числом элементов $F \times F$ в уравнении (14.10.1) вычислительная сложность ее обращения не оказывается на общей сложности параллельных RLS-алгоритмов при $N \gg 4$.

Например, если $F=4$, то использование стандартной процедуры обращения матриц позволяет выполнять это обращение с вычислительной сложностью 88 умножений, 47 сложений и 1 деление на одну итерацию алгоритма адаптивной фильтрации.

Использование модифицированной процедуры [183] требует для обращения матрицы 80 умножений, 64 сложения и 4 деления, а использование леммы об обращении клеточных матриц (11.51), (11.52) позволяет выполнить такое обращение за 54 умножения, 32 сложения и 4 деления на одну итерацию алгоритма адаптивной фильтрации.

Аналогичным образом получены параллельные версии линейно-ограниченных RLS-алгоритмов (см. табл. 14.5, табл. 14.7 и табл. 14.9). Эти алгоритмы приведены в табл. 14.11.

Таблица 14.11. Параллельный LC RLS-алгоритм с двойным ограничением и скользящим окном

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N, \dots, \mathbf{x}_N(0-L+1) = \mathbf{0}_N, \mathbf{p}_N(0) = \mathbf{0}_N, \dots,$ $\mathbf{p}_N(0-L+1) = \mathbf{0}_N, d(0) = 0, \dots, d(0-L+1) = 0, \mathbf{X}_{NF}(0) = \mathbf{O}_{NF},$ $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{\Lambda}_N, \mathbf{\Gamma}_{NJ}(0) = \mathbf{R}_N^{-1}(0) \mathbf{C}_{NJ}, \mathbf{Q}_{NJ}(0) = \mathbf{\Gamma}_{NJ}(0) [\mathbf{C}_{NJ}^H \mathbf{\Gamma}_{NJ}(0)]^{-1},$ $\mathbf{h}_N(0) = \mathbf{Q}_{NJ}(0) \mathbf{f}_J,$ $\mathbf{\Lambda}_N = \text{diag}\{1, \lambda, \dots, \lambda^{N_1-1}, \dots, 1, \lambda, \dots, \lambda^{N_m-1}, \dots, 1, \lambda, \dots, \lambda^{N_M-1}\}, m=1, \dots, M$	(14.11.0)
For $k = 1, 2, \dots, K$	
Вычисление $\mathbf{G}_{NF}(k)$, см., например, (14.10.1), (14.10.2)	(14.11.1)
$\mathbf{V}_{JF}(k) = \mathbf{C}_{NJ}^H \mathbf{G}_{NF}(k)$	(14.11.2)
$\mathbf{N}_{JF}^H(k) = \mathbf{X}_{NF}^H(k) \mathbf{Q}_{NJ}(k-1)$	(14.11.3)
$\mathbf{Q}'_{NJ}(k) = [\mathbf{Q}_{NJ}(k-1) - \mathbf{G}_{NF}(k) \mathbf{N}_{JF}^H(k)] \times$ $\times \left\{ \mathbf{I}_J + \mathbf{V}_{JF}(k) [\mathbf{I}_F - \mathbf{N}_{JF}^H(k) \mathbf{V}_{JF}(k)]^{-1} \mathbf{N}_{JF}^H(k) \right\}$	(14.11.4)
$\mathbf{Q}_{NJ}(k) = \mathbf{Q}'_{NJ}(k) + \mathbf{C}_{NJ} (\mathbf{C}_{NJ}^H \mathbf{C}_{NJ})^{-1} [\mathbf{I}_J - \mathbf{C}_{NJ}^H \mathbf{Q}'_{NJ}(k)]$	(14.11.5)
$\boldsymbol{\alpha}_F(k) = \mathbf{d}_F(k) - \mathbf{h}_N^H(k-1) \mathbf{X}_{NF}(k)$	(14.11.6)
$\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mathbf{G}_{NF}(k) \boldsymbol{\alpha}_F^H(k)$	(14.11.7)
$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ}(k) [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k)]$	(14.11.8)
End for k	

Другие параллельные RLS-алгоритмы могут быть найдены в работах автора настоящей книги, указанных в начале настоящей главы, а также в [184].

14.7. Особенности построения быстрых многоканальных алгоритмов аффинных проекций

Алгоритм аффинных проекций [160] является промежуточным звеном между простыми и сложными алгоритмами адаптивной фильтрации. Этот алгоритм представляет собой блочный вариант NLMS-алгоритма. В алгоритме аффинных проекций градиент определяется на основе скользящих блоков данных: матри-

цы входных сигналов и вектора ошибок на выходе адаптивного фильтра. Длительность переходного процесса в алгоритме аффинных проекций больше, чем в RLS-алгоритмах, но меньше, чем у NLMS-алгоритма. Вычислительная сложность алгоритма аффинных проекций составляет $O(N + L)$ арифметических операций. Здесь L — размер проекции (длина скользящего окна, на котором определяется градиент, выраженная числом отсчетов обрабатываемых сигналов).

В приложениях, требующих применения адаптивных фильтров с большим числом ВК, а также использующих в качестве выходного сигнала только сигнал ошибки адаптивного фильтра, широкое распространение получил быстрый алгоритм аффинных проекций (Fast Affine Projections, FAP) [161—165]. При $L \ll N$ вычислительная сложность FAP-алгоритма сравнима со сложностью NLMS-алгоритма. FAP-алгоритм в настоящее время широко применяется в адаптивных фильтрах компенсаторов сигналов акустического и электрического эха.

Известные варианты FAP-алгоритма предназначены для использования в одноканальных адаптивных фильтрах с действительными ВК. Однако существует ряд приложений, где требуются многоканальные адаптивные фильтры или фильтры с комплексными ВК. Например, для подавления сигналов акустического стереоэха требуется двухканальный адаптивный фильтр с действительными ВК, а для подавления множественных эхо-сигналов в модемах проводных каналов связи требуется многоканальный адаптивный фильтр с комплексными ВК. В обоих случаях это, как правило, адаптивные фильтры с несколькими сотнями или даже тысячами ВК. Из-за высоких требований к вычислительным ресурсам реализация даже быстрых RLS-алгоритмов для таких фильтров является проблематичной, а использование несложного NLMS-алгоритма часто неприемлемо из-за его медленной сходимости.

Для решения указанных задач в настоящем разделе рассматривается многоканальный FAP-алгоритм адаптивной фильтрации с комплексными ВК, а также применение в этом алгоритме процедур вычисления ВК фильтров и минимальной энергии ошибок линейного предсказания с помощью быстрых RLS-алгоритмов со скользящим окном без экспоненциального взвешивания. Вычисление этих переменных является частью FAP-алгоритма.

Последовательность приемов, используемых при выводе уравнений многоканального FAP-алгоритма с комплексными ВК, в основном совпадает с такими приемами для одноканального FAP-алгоритма с действительными ВК [162]. В то же время в новом алгоритме существует ряд особенностей, обусловленных наличием M каналов, неодинаковым числом ВК в каналах, комплексной арифметикой и зависимым от времени шагом сходимости $\mu(k)$. Последнее свойство требуется для модификации многоканального FAP-алгоритма за счет переменного шага сходимости [165] или остановки и продолжения работы FAP-алгоритма в случае его применения в адаптивных компенсаторах эхо-сигналов.

Обобщенная структура многоканального адаптивного фильтра показана на рис. 1.3. Алгоритм аффинных проекций для многоканального адаптивного фильтра с неодинаковым числом комплексных ВК в каналах и переменным шагом сходимости описывается уравнениями

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu(k) \mathbf{X}_{NL}(k) \boldsymbol{\epsilon}_L(k), \quad (14.24)$$

$$\boldsymbol{\epsilon}_L(k) = \left[\mathbf{X}_{NL}^H(k) \mathbf{X}_{NL}(k) + \delta^2 \mathbf{I}_L \right]^{-1} \boldsymbol{\alpha}_L^*(k), \quad (14.25)$$

$$\boldsymbol{\alpha}_L(k) = \mathbf{d}_L(k) - \left[\mathbf{h}_N^H(k-1) \mathbf{X}_{NL}(k) \right]^T, \quad (14.26)$$

где $\boldsymbol{\alpha}_L(k) = [\alpha(k), \alpha(k-1), \dots, \alpha(k-L+1)]^T$, $\boldsymbol{\epsilon}_L(k) = [\epsilon(k), \epsilon(k-1), \dots, \epsilon(k-L+1)]^T$ — векторы ошибок и нормализованных ошибок адаптивного фильтра; $\mathbf{d}_L(k) = [d(k), d(k-1), \dots, d(k-L+1)]^T$ — вектор отсчетов сигналов на входе требуемого сигнала адаптивного фильтра; $\mathbf{X}_{NL}(k) = [\mathbf{x}_N(k), \mathbf{x}_N(k-1), \dots, \mathbf{x}_N(k-L+1)]$. Структура матрицы $\mathbf{X}_{NL}(k)$ приведена на рис. 14.9.

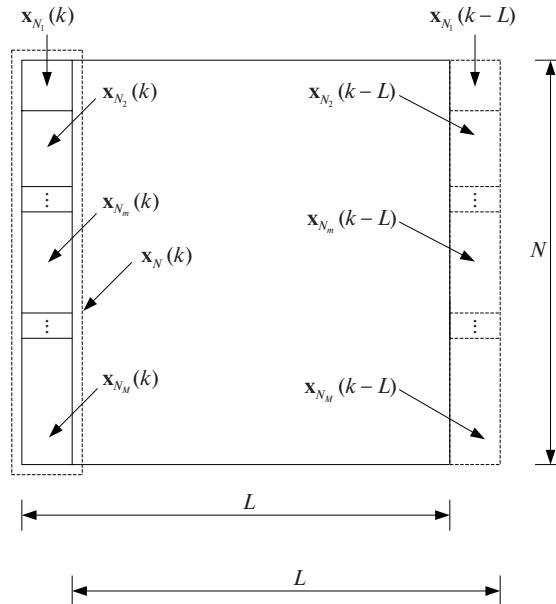


Рис. 14.9. Структура матрицы $\mathbf{X}_{NL}(k)$

Алгоритм (14.24)–(14.26) отличается от алгоритма [160] наличием переменного шага сходимости, комплексной арифметикой и структурой векторов $\mathbf{x}_N(k)$ и $\mathbf{h}_N(k)$, в то время как последовательность вычислений в обоих алгоритмах одинакова. Отличия в вычислениях многоканальных и одноканальных алгоритмов возникают в их быстрых версиях. Ниже рассматриваются основные шаги получения многоканального FAP-алгоритма с переменным шагом сходимости.

В алгоритме аффинных проекций требуется примерно $2NL + NL^2$ арифметических операций для выполнения вычислений в уравнениях (14.24), (14.25) и для вычисления матрицы $\mathbf{X}_{NL}^H(k)\mathbf{X}_{NL}(k)$ в уравнении (14.26), а также $O(L^3)$ операций для обращения этой матрицы. В FAP-алгоритме уравнения (14.24)–(14.26) заменяются эквивалентными уравнениями со сложностью, примерно равной $2N$ арифметическим операциям, т. е. сравнимой со сложностью NLMS-алгоритма.

Первым шагом такой замены является запись уравнения (14.26) в виде

$$\alpha_L(k) = \begin{bmatrix} \alpha(k) \\ \bar{\mathbf{e}}_{L-1}(k-1) \end{bmatrix}, \quad (14.27)$$

где $\mathbf{e}_L(k-1) = \mathbf{h}_N^H(k-1)\mathbf{X}_{NL}(k-1)$ [162]. Здесь и далее запись типа $\bar{\mathbf{e}}_{L-1}$ обозначает вектор, состоящий из первых $L-1$ элементов вектора \mathbf{e}_L , т. е. элементов с номерами $1, \dots, L-1$. При использовании переменного шага сходимости $\mu(k)$ уравнение (14.27) эквивалентно уравнению [165]

$$\alpha_L(k) = \begin{bmatrix} \alpha(k) \\ [1 - \mu(k-1)] \bar{\alpha}_{L-1}(k-1) \end{bmatrix}, \quad (14.28)$$

которое может быть использовано вместо уравнения (14.27). Для вычислительной процедуры (14.28) требуется всего $N+L$ арифметических операций, в то время как для процедуры (14.26) требуется $NL+L$ операций.

Следующим шагом является получение уравнения для вычисления значений ошибок на выходе адаптивного фильтра $\alpha(k)$, в котором не используется вектор ВК $\mathbf{h}_N(k)$. Это достигается путем замены уравнения (14.24) с вычислительной сложностью, примерно равной NL арифметическим операциям, уравнением со сложностью N арифметических операций для вычисления другого вектора $\hat{\mathbf{h}}_N(k)$.

В случае переменного шага сходимости $\mu(k)$ уравнение (14.24) может быть заменено эквивалентным уравнением

$$\mathbf{h}_N(k) = \hat{\mathbf{h}}_N(k) + \bar{\mathbf{X}}_{N(L-1)}(k) \bar{\mathbf{p}}_{L-1}(k). \quad (14.29)$$

Текущее значение ошибки на выходе адаптивного фильтра в уравнении (14.28) определяется как

$$\alpha(k) = \hat{\alpha}(k) - \bar{\mathbf{p}}_{L-1}^H(k-1) \mathbf{r}_{L-1}(k). \quad (14.30)$$

В уравнении (14.29) выражение $\bar{\mathbf{X}}_{N(L-1)}(k)$ обозначает матрицу, состоящую из первых $L-1$ столбцов матрицы $\mathbf{X}_{NL}(k)$, т. е. столбцов с номерами $1, \dots, L-1$, а вектор $\hat{\mathbf{h}}_N(k)$ определяется как

$$\hat{\mathbf{h}}_N(k) = \hat{\mathbf{h}}_N(k-1) + \mathbf{x}_N(k-L+1) p_{L-1}(k), \quad (14.31)$$

$$\hat{\mathbf{h}}_N(k) = \left[\hat{\mathbf{h}}_1^T(k), \dots, \hat{\mathbf{h}}_m^T(k), \dots, \hat{\mathbf{h}}_M^T(k) \right]^T. \quad (14.32)$$

Вычислительная сложность уравнения (14.31) равна N арифметическим операциям. Переменная $p_{L-1}(k)$ в этом уравнении является последним элементом вектора $\mathbf{p}_L(k)$, определяемого в случае переменного шага сходимости как

$$\mathbf{p}_L(k) = \mu(k) \mathbf{\epsilon}_L(k) + \begin{bmatrix} 0 \\ \bar{\mathbf{p}}_{L-1}(k-1) \end{bmatrix}. \quad (14.33)$$

Переменная $\hat{\alpha}(k)$ в уравнении (14.30) определяется как

$$\hat{\alpha}(k) = d(k) - \hat{\mathbf{h}}_N^H(k-1) \mathbf{x}_N(k), \quad (14.34)$$

а вектор $\mathbf{r}_{L-1}(k)$ — как

$$\begin{aligned} \mathbf{r}_{L-1}(k) = & \bar{\mathbf{X}}_{N(L-1)}^H(k-1) \mathbf{x}_N(k) = \mathbf{r}_{L-1}(k-1) + \\ & + \sum_{m=1}^M \left[x_m(k) \mathbf{x}_{L-1,m}^*(k-1) - x_m(k-N_m) \mathbf{x}_{L-1,m}^*(k-N_m-1) \right], \end{aligned} \quad (14.35)$$

где $\mathbf{x}_{L-1,m}(k-1) = [x_m(k-1), x_m(k-2), \dots, x_m(k-L+1)]^T$ и $\mathbf{x}_{L-1,m}(k-N_m-1) = [x_m(k-N_m-1), x_m(k-N_m-2), \dots, x_m(k-N_m-L+1)]^T$.

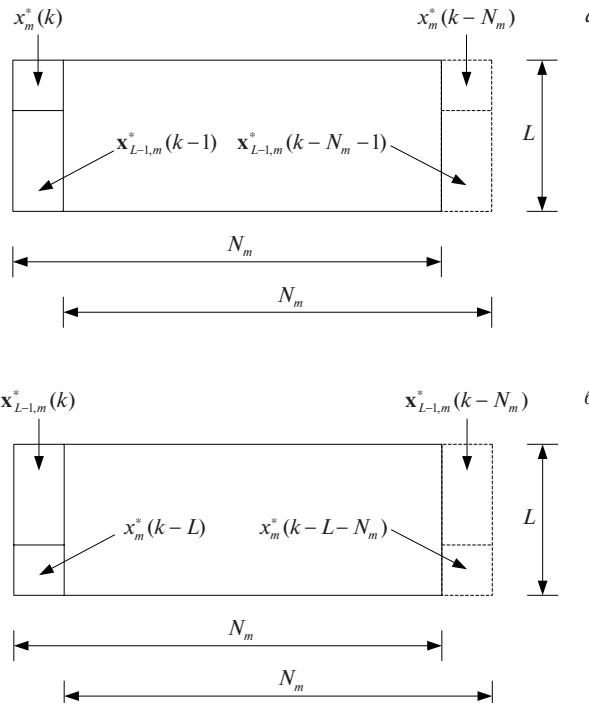


Рис. 14.10. Структура данных: а) при вычислении вектора $\mathbf{r}_{L-1}(k)$ и ошибок линейного предсказания $\alpha_{L-1,m}^{\text{fU}}(k), \alpha_{L-1,m}^{\text{fD}}(k)$; б) при вычислении ошибок линейного предсказания $\alpha_{L-1,m}^{\text{bU}}(k), \alpha_{L-1,m}^{\text{bD}}(k)$

Уравнение (14.35) является одним из отличий многоканального FAP-алгоритма от одноканального FAP-алгоритма. Из этого уравнения следует, что

вектор $\mathbf{r}_{L-1}(k)$ в случае M -канального адаптивного фильтра равен сумме M составляющих, каждая из которых вычисляется на своем скользящем окне длиной в N_m отсчетов (рис. 14.10). Значения N_m могут быть разными, что определяется решаемой задачей.

Другая особенность многоканального FAP-алгоритма связана с вычислением матрицы $\mathbf{R}_L(k)$ в уравнении (14.25). Из этого уравнения следует, что в многоканальном адаптивном фильтре эта матрица вычисляется как

$$\begin{aligned}\mathbf{R}_L(k) &= \mathbf{X}_{NL}^H(k) \mathbf{X}_{NL}(k) + \delta^2 \mathbf{I}_L = \sum_{m=1}^M \mathbf{X}_{N_m L}^H(k) \mathbf{X}_{N_m L}(k) + \delta^2 \mathbf{I}_L = \\ &= \sum_{m=1}^M \sum_{i=0}^{N_m-1} \mathbf{x}_{L,m}^*(k-i) \mathbf{x}_{L,m}^T(k-i) + \delta^2 \mathbf{I}_L = \sum_{m=1}^M \mathbf{R}_{L,m}(k) + \delta^2 \mathbf{I}_L,\end{aligned}\quad (14.36)$$

где $\mathbf{X}_{N_m L}(k) = [\mathbf{x}_{N_m}(k), \mathbf{x}_{N_m}(k-1), \dots, \mathbf{x}_{N_m}(k-L+1)]^T$ и $\mathbf{x}_{L,m}(k) = [x_m(k), x_m(k-1), \dots, x_m(k-L+1)]^T$. Таким образом, матрица $\mathbf{R}_L(k)$ также равна сумме матриц для сигналов отдельных каналов адаптивного фильтра, каждая из которых вычисляется на своем скользящем окне длиной в N_m отсчетов.

Для эффективного вычисления $\boldsymbol{\epsilon}_L(k)$ в FAP-алгоритмах используется представление $\mathbf{R}_L(k)$ в виде клеточной матрицы. Для обращения такой матрицы в уравнении (14.25) применяются леммы об обращении клеточных матриц (11.51), (11.52) и теория линейного предсказания по критерию наименьших квадратов (см. глава 11).

В результате, для вычисления $\boldsymbol{\epsilon}_L(k)$ вместо уравнения (14.25) используется последовательность уравнений

$$\boldsymbol{\epsilon}_L(k) = \begin{bmatrix} 0 \\ \mathbf{s}_{L-1}(k-1) \end{bmatrix} + \frac{1}{E_{L-1,\min}^f(k)} \mathbf{a}_L(k) \mathbf{a}_L^H(k) \boldsymbol{\alpha}_L^*(k), \quad (14.37)$$

$$\begin{bmatrix} \mathbf{z}_{L-1}(k) \\ 0 \end{bmatrix} = \boldsymbol{\epsilon}_L(k) - \frac{1}{E_{L-1,\min}^b(k)} \mathbf{b}_L(k) \mathbf{b}_L^H(k) \boldsymbol{\alpha}_L^*(k), \quad (14.38)$$

$$\mathbf{s}_{L-1}(k) = [1 - \mu(k)] \mathbf{z}_{L-1}(k). \quad (14.39)$$

В уравнениях (14.37) и (14.38) переменные $E_{L-1,\min}^f(k)$ и $E_{L-1,\min}^b(k)$ — это энергии ошибок линейного предсказания, а $\mathbf{a}_L(k)$ и $\mathbf{b}_L(k)$ — векторы ВК фильтров линейного предсказания комплексно сопряженных значений входных сигналов адаптивного фильтра. Поскольку матрица $\mathbf{R}_L(k)$ состоит из суммы матриц $\mathbf{R}_{L,m}(k)$, то для вычисления $E_{L-1,\min}^f(k)$, $E_{L-1,\min}^b(k)$, $\mathbf{a}_L(k)$ и $\mathbf{b}_L(k)$ необходимо последовательно M раз использовать процедуру одноканального линейного предсказания. Кроме того, так как матрицы $\mathbf{R}_{L,m}(k)$ определяются на скользящих окнах длиной в N_m отсчетов входных сигналов адаптивного фильтра $x_m(k)$, то и переменные $E_{L-1,\min}^f(k)$, $E_{L-1,\min}^b(k)$, $\mathbf{a}_L(k)$ и $\mathbf{b}_L(k)$ также вычисляются на аналогичных

скользящих окнах. Эти вычисления могут быть выполнены на основе процедур линейного предсказания быстрых одноканальных RLS-алгоритмов со скользящим окном [66, 184] без экспоненциального взвешивания, т. е. при $\lambda=1$.

Вычислительная процедура полученного многоканального FAP-алгоритма с комплексными ВК приведена в табл. 14.12.

Таблица 14.12. Многоканальный FAP-алгоритм

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\hat{\mathbf{h}}_N(0) = \mathbf{0}_N$, $\mathbf{X}_{NL}(0) = \mathbf{0}_{NL}$, $\boldsymbol{\alpha}_L(0) = \mathbf{0}_L$, $\mathbf{p}_L(0) = \mathbf{0}_L$, $\mu(0) = \mu$, $\mathbf{r}_{L-1}(k) = \mathbf{0}_{L-1}$, $\mathbf{s}_{L-1}(k) = \mathbf{0}_{L-1}$	(14.12.0)
For $k = 1, 2, \dots, K$	
$\mathbf{r}_{L-1}(k) = \mathbf{r}_{L-1}(k-1) + \sum_{m=1}^M [x_m(k) \mathbf{x}_{L-1,m}^*(k-1) - x_m(k-N_m) \mathbf{x}_{L-1,m}^*(k-N_m-1)]$	(14.12.1)
$\hat{\alpha}(k) = d(k) - \hat{\mathbf{h}}_N^H(k-1) \mathbf{x}_N(k)$	(14.12.2)
$\alpha(k) = \hat{\alpha}(k) - \bar{\mathbf{p}}_{L-1}^H(k-1) \mathbf{r}_{L-1}(k)$	(14.12.3)
$\boldsymbol{\alpha}_L(k) = \begin{bmatrix} \alpha(k) \\ [1-\mu(k-1)] \bar{\boldsymbol{\alpha}}_{L-1}(k-1) \end{bmatrix}$	(14.12.4)
For $m = 1, 2, \dots, M$	
Вычисление $E_{L-1,\min}^f(k)$, $E_{L-1,\min}^b(k)$, $\mathbf{h}_{L-1}^f(k)$, $\mathbf{h}_{L-1}^b(k)$	(14.12.5)
End for m	
$\boldsymbol{\varepsilon}_L(k) = \begin{bmatrix} 0 \\ \mathbf{s}_{L-1}(k-1) \end{bmatrix} + \frac{1}{E_{L-1,\min}^f(k)} \begin{bmatrix} 1 \\ -\mathbf{h}_{L-1}^f(k) \end{bmatrix} \begin{bmatrix} 1, -\mathbf{h}_{L-1}^{fH}(k) \end{bmatrix} \boldsymbol{\alpha}_L^*(k)$	(14.12.6)
$\begin{bmatrix} \mathbf{z}_{L-1}(k) \\ 0 \end{bmatrix} = \boldsymbol{\varepsilon}_L(k) - \frac{1}{E_{L-1,\min}^b(k)} \begin{bmatrix} 1 \\ -\mathbf{h}_{L-1}^b(k) \end{bmatrix} \begin{bmatrix} 1, -\mathbf{h}_{L-1}^{bH}(k) \end{bmatrix} \boldsymbol{\alpha}_L^*(k)$	(14.12.7)
$\mathbf{s}_{L-1}(k) = [1-\mu(k)] \mathbf{z}_{L-1}(k)$	(14.12.8)
$\mathbf{p}_L(k) = \mu(k) \boldsymbol{\varepsilon}_L(k) + \begin{bmatrix} 0 \\ \bar{\mathbf{p}}_{L-1}(k-1) \end{bmatrix}$	(14.12.9)
$\hat{\mathbf{h}}_N(k) = \hat{\mathbf{h}}_N(k-1) + \mathbf{x}_N(k-L+1) p_{L-1}(k)$	(14.12.10)
End for k	

Пример использования быстрых RLS-алгоритмов со скользящим окном в шаге (14.12.5) алгоритма (см. табл. 14.12) приведен в табл. 14.13. Здесь верхние индексы U и D обозначают вычисления, связанные с добавлением и исключением данных в скользящих окнах (см. также RLS-алгоритмы в разделе 14.4).

Табл. 14.13 дает представление о модификации быстрых RLS-алгоритмов при их использовании в качестве шага (14.12.5) вычислений, представленных в табл. 14.12. При этом в FAP-алгоритме требуется дополнительная инициализация переменных, см. уравнение (14.13.0), используемых в процедурах линей-

Таблица 14.13. Вычисление переменных $E_{\min}^f(k)$, $E_{\min}^b(k)$, $\mathbf{h}_{L-1}^f(k)$ и $\mathbf{h}_{L-1}^b(k)$ с помощью быстрого FK-алгоритма

Вычисления	Ссылки
Additional initialization: $\mathbf{h}_{L-1}^f(0) = \mathbf{0}_{L-1}$, $\mathbf{h}_{L-1}^b(0) = \mathbf{0}_{L-1}$, $E_{L-1,\min}^f(0) = \delta^2$, $E^b(0) = \delta^2$, $\mathbf{g}_{L-1,m}^U(0) = \mathbf{0}_{L-1}$, $\mathbf{g}_{L-1,m}^D(0) = \mathbf{0}_{L-1}$, $m = 1, \dots, M$	(14.13.0)
For $m = 1, 2, \dots, M$	
$\alpha_{L-1,m}^{fU}(k) = x_m^*(k) - \mathbf{h}_{L-1}^{fH}(k-1) \mathbf{x}_{L-1,m}^*(k-1)$	(14.13.1)
$\alpha_{L-1,m}^{bU}(k) = x_m^*(k-L) - \mathbf{h}_{L-1}^{bH}(k-1) \mathbf{x}_{L-1,m}^*(k)$	(14.13.2)
$\mathbf{h}_{L-1}^f(k) = \mathbf{h}_{L-1}^f(k-1) + \mathbf{g}_{L-1}^U(k-1) \alpha_{L-1,m}^{fU*}(k)$	(14.13.3)
$e_{L-1,m}^{fU}(k) = x_m^*(k) - \mathbf{h}_{L-1}^{fUH}(k) \mathbf{x}_{L-1,m}^*(k-1)$	(14.13.4)
$E_{L-1,\min}^{fU}(k) = E_{L-1,\min}^f(k-1) + e_{L-1,m}^{fU}(k) \alpha_{L-1,m}^{fU*}(k)$	(14.13.5)
$\begin{bmatrix} \mathbf{q}_{L-1,m}^U(k) \\ q_{L-1,m}^U(k) \end{bmatrix} = \begin{bmatrix} 1 \\ -\mathbf{h}_{L-1}^f(k) \end{bmatrix} \frac{e_{L-1,m}^{fU}(k)}{E_{L-1,\min}^{fU}(k)} + \begin{bmatrix} 0 \\ \mathbf{g}_{L-1,m}^U(k) \end{bmatrix}$	(14.13.6)
$\mathbf{g}_{L-1,m}^U(k) = \frac{\mathbf{q}_{L-1,m}^U(k) + \mathbf{h}_{L-1}^b(k-1) q_{L-1,m}^U(k)}{1 - \alpha_{L-1,m}^{bU*}(k) q_{L-1,m}^U(k)}$	(14.13.7)
$\mathbf{h}_{L-1,m}^{bU}(k) = \mathbf{h}_{L-1,m}^b(k-1) + \mathbf{g}_{L-1,m}^U(k) \alpha_{L-1,m}^{bU*}(k)$	(14.13.8)
$e_{L-1,m}^{bU}(k) = x_m^*(k-L) - \mathbf{h}_{L-1}^{bUH}(k) \mathbf{x}_{L-1,m}^*(k)$	(14.13.9)
$E^{bU}(k) = E^b(k-1) + e_{L-1,m}^{bU}(k) \alpha_{L-1,m}^{bU*}(k)$	(14.13.10)
$\alpha_{L-1,m}^{fD}(k) = x_m^*(k-N_m) - \mathbf{h}_{L-1,m}^{fUH}(k) \mathbf{x}_{L-1,m}^*(k-1-N_m)$	(14.13.11)
$\alpha_{L-1,m}^{bD}(k) = x_m^*(k-L-N_m) - \mathbf{h}_{L-1,m}^{bUH}(k) \mathbf{x}_{L-1,m}^*(k-N_m)$	(14.13.12)
$\mathbf{h}_{L-1}^f(k) = \mathbf{h}_{L-1}^f(k-1) - \mathbf{g}_{L-1,m}^D(k-1) \alpha_{L-1,m}^{fD*}(k)$	(14.13.13)
$e_{L-1,m}^{fD}(k) = x_m^*(k-N_m) - \mathbf{h}_{L-1,m}^{fH}(k) \mathbf{x}_{L-1,m}^*(k-1-N_m)$	(14.13.14)
$E_{L-1,\min}^f(k) = E_{L-1,\min}^{fU}(k) - e_{L-1,m}^{fD*}(k) \alpha_{L-1,m}^{fD*}(k)$	(14.13.15)
$\begin{bmatrix} \mathbf{q}_{L-1,m}^D(k) \\ q_{L-1,m}^D(k) \end{bmatrix} = \begin{bmatrix} 1 \\ -\mathbf{h}_{L-1,m}^{fU}(k) \end{bmatrix} \frac{e_{L-1,m}^{fD}(k)}{E_{L-1,\min}^f(k)} + \begin{bmatrix} 0 \\ \mathbf{g}_{L-1,m}^D(k-1) \end{bmatrix}$	(14.13.16)
$\mathbf{g}_{L-1,m}^D(k) = \frac{\mathbf{q}_{L-1,m}^D(k) + \mathbf{h}_{L-1,m}^b(k) q_{L-1,m}^D(k)}{1 + \alpha_{L-1,m}^{bD*}(k) q_{L-1,m}^D(k)}$	(14.13.17)
$\mathbf{h}_{L-1,m}^{bD}(k) = \mathbf{h}_{L-1,m}^b(k) - \mathbf{g}_{L-1,m}^D(k) \alpha_{L-1,m}^{bD*}(k)$	(14.13.18)
$e_{L-1,m}^{bD}(k) = x_m^*(k-L-N_m) - \mathbf{h}_{L-1}^{bH}(k) \mathbf{x}_{L-1,m}^*(k-N_m)$	(14.13.19)
$E_{L-1,\min}^b(k) = E_{L-1,\min}^{bU}(k) - e_{L-1,m}^{bD}(k) \alpha_{L-1,m}^{bD*}(k)$	(14.13.20)
End for m	

ного предсказания, а линейное предсказание осуществляется для комплексно-сопряженных значений входных сигналов адаптивного фильтра. Кроме того,

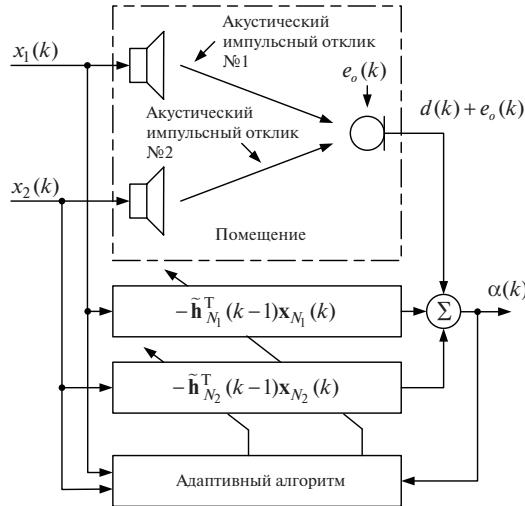


Рис. 14.11. Подавление сигналов двухканального акустического эха

как уже упоминалось, в вычислениях отсутствует множитель экспоненциального взвешивания обрабатываемых сигналов.

В многоканальном FAP-алгоритме вектор $\mathbf{r}_{L-1}(k)$ и матрица $\mathbf{R}_L(k)$ определяются суммой составляющих, число которых равно числу каналов многоканального адаптивного фильтра. Поэтому для вычисления матрицы $\mathbf{R}_L^{-1}(k)$ последовательно M раз используется одноканальная процедура линейного предсказания, см. уравнения (14.13.1)–(14.13.21).

В случае переменного шага сходимости в уравнениях (14.28), (14.33) и (14.39) на текущей итерации алгоритма k используются два различных значения шага сходимости: текущее значение $\mu(k)$ и значение $\mu(k-1)$ с предыдущей итерации.

Совокупность отмеченных особенностей отличает рассмотренный FAP-алгоритм от других известных вариантов этого алгоритма.

Вычислительная сложность многоканального FAP-алгоритма равна

$$C \approx 2ML|_{(14.12.1)} + N|_{(4.12.2)} + L|_{(4.12.3)} + L|_{(4.12.4)} + O(ML)|_{(4.12.5)} + \\ + 2L|_{(4.12.6)} + 2L|_{(4.12.7)} + L|_{(4.12.8)} + L|_{(4.12.9)} + N|_{(4.12.10)} = 2N + O(ML) + 8L \quad (14.40)$$

арифметическим операциям (сложениям с умножениями). В уравнении (14.40) цифры в нижних индексах указывают на номера уравнений алгоритма в табл. 14.12, а значение $O(ML)$ определяется используемой процедурой линейного предсказания и находится в переделах $10ML \dots 20ML$ [66, 184].

Результаты моделирования многоканального FAP-алгоритма в задаче подавления двухканального акустического эха (рис. 14.11) приведены на рис. 14.12а и рис. 14.12б. Они подтверждают работоспособность этого алгоритма и демонстрируют его основные свойства. Подавление акустического эха — это одна

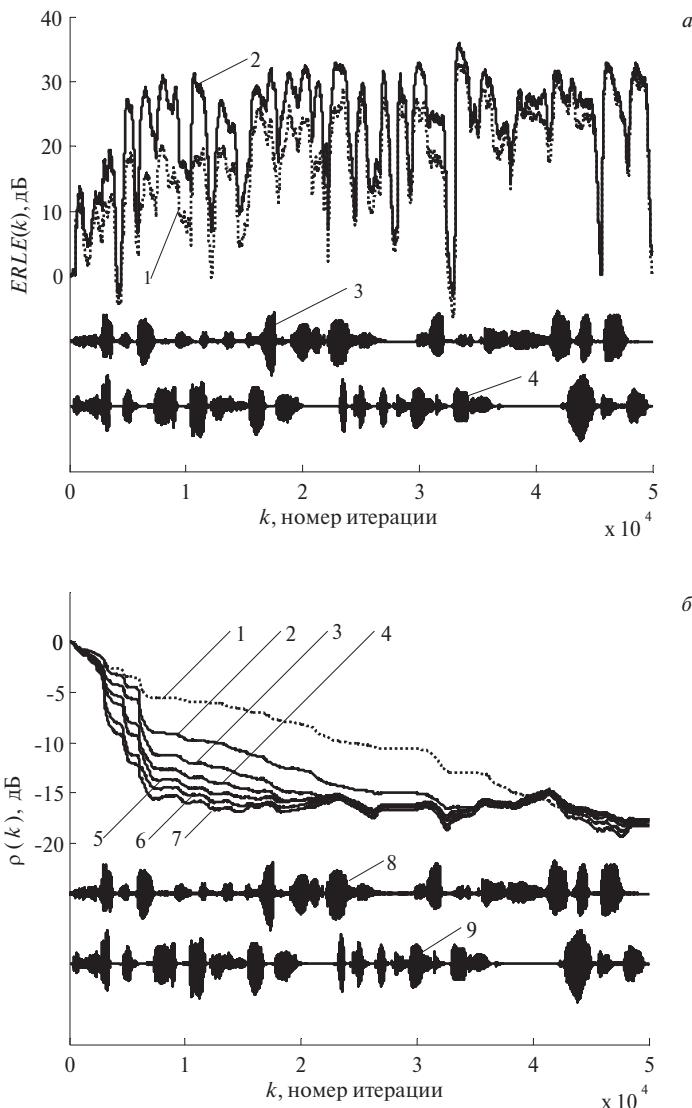


Рис. 14.12. Результаты моделирования: а) параметр $ERLE$; б) рассогласование из задач, где такой алгоритм может быть использован. Выходным сигналом адаптивного фильтра в такой задаче является сигнал ошибки.

Подавление акустического эха относится к классу задач идентификации объекта (определения неизвестного импульсного отклика в данном случае). В экспериментах различные акустические импульсные отклики \mathbf{w}_{N_1} и \mathbf{w}_{N_2} , обозначенные на рис. 14.11 как №1 и №2, моделировались с помощью метода зеркальных отображений [51], а число ВК в каждом из каналов равнялось 1024. Эхо-сигнал моделировался как

$$d(k) = \mathbf{w}_N^T \mathbf{x}_N(k), \quad (14.41)$$

где $\mathbf{w}_N = \left[\mathbf{w}_{N_1}^T, \mathbf{w}_{N_2}^T \right]^T$. В качестве сигналов $x_1(k)$ и $x_2(k)$ использовались две различные речевые последовательности (нестационарные сигналы), дискретизированные на частоте 8 кГц. Эти сигналы условно показаны в нижней части рис. 14.12а и рис. 14.12б. К эхо-сигналу добавлялся микрофонный шум $e_o(k)$ с уровнем примерно -30 дБ относительно среднего значения эхо-сигнала на длительности эксперимента. Параметр регуляризации выбирался из условия $\delta^2 \approx 30\sigma_x^2$, где σ_x^2 — дисперсия сигналов $x_1(k)$ и $x_2(k)$. Такое значение параметра регуляризации [162] обеспечивает устойчивое функционирование процедур линейного предсказания, используемых в FAP-алгоритмах. Указанный уровень шума примерно равен уровню микрофонных шумов в задачах подавления сигналов акустического эха. В других приложениях адаптивных фильтров, например в задачах подавления электрического эха в проводных каналах связи, уровень аддитивного шума на входе сигнала $d(k)$ определяется параметрами канала связи [48, 49]. При моделировании алгоритма использовался постоянный шаг сходимости $\mu(k) = 1$.

Рис. 14.12а демонстрирует зависимость параметра ERLE (14.15) в процессе адаптации. На этом рисунке: 1 — NLMS-алгоритм; 2 — FAP-алгоритм, $L=64$; 3 — сигнал $x_1(k)$; 4 — сигнал $x_2(k)$. Из этого рисунка следует, что FAP-алгоритм обеспечивает не только меньшее время переходного процесса по сравнению с NLMS-алгоритмом, но и несколько более высокое значение ERLE в установившемся режиме. Это свидетельствует о более высоком качестве многоканального FAP-алгоритма по сравнению с многоканальным NLMS-алгоритмом.

На рис. 14.12б представлен другой показатель качества адаптивного фильтра в задачах идентификации, к которым принадлежит и задача подавления эхо-сигналов. Это рассогласование (1.8). По времени достижения установившегося значения параметра $\rho(k)$ можно судить о длительности переходного процесса адаптивного фильтра. Так как в FAP-алгоритме $\mathbf{h}_N(k)$ не вычисляется в явном виде, то в уравнении (1.8) используются значения $\mathbf{h}_N(k)$, определяемые с помощью уравнения (14.29). На рис. 14.12б: 1 — NLMS-алгоритм; 2 — FAP-алгоритм, $L=2$; 3 — FAP-алгоритм, $L=4$; 4 — FAP-алгоритм, $L=8$; 5 — FAP-алгоритм, $L=16$; 6 — FAP-алгоритм, $L=32$; 7 — FAP-алгоритм, $L=64$, $L=128$; 8 — сигнал $x_1(k)$; 9 — сигнал $x_2(k)$.

Основные свойства многоканального FAP-алгоритма совпадают со свойствами одноканального FAP-алгоритма, отмеченными в [162]. В многоканальном FAP-алгоритме длительность переходного процесса уменьшается с увеличением L (см. рис. 14.12б). Увеличение $L > 30$ практически не сказывается на длительности переходного процесса FAP-алгоритма. Поэтому при больших значениях N (больше 1000) и небольших (в пределах 30...100) значениях L вычислительная сложность FAP-алгоритма становится примерно равной сложно-

сти NLMS-алгоритма. Например, при $M=2$, $N_m=1024$ и $L=64$ вычислительная сложность FAP-алгоритма определяется согласно уравнению (14.40) как $C \approx 6144$ арифметических операций (при $O(ML)=12ML$), что лишь в 1.5 раза превышает вычислительную сложность простейшего NLMS-алгоритма ($C \approx 2N=4096$ арифметических операций). При этом длительность переходного процесса в FAP-алгоритме примерно в 3...5 раз меньше, чем в NLMS-алгоритме (см. рис. 14.12б). Для сравнения, вычислительная сложность самого простого быстрого многоканального RLS-алгоритма (FTF-алгоритма с возрастающим окном) в этом случае равняется $C \approx 5MN+2N=25464$ арифметическим операциям. Это примерно в 4.1 раза больше сложности FAP-алгоритма при указанном значении L и в 6.2 раза больше сложности NLMS-алгоритма.

Сравнительные характеристики вычислительной сложности NLMS-алгоритма, FAP-алгоритма при $L=64$ и FTF-алгоритма при различных значениях N и M представлены на рис. 14.13. Из этого рисунка следует, что при заданном общем числе ВК адаптивного фильтра N увеличение числа каналов на единицу незначительно увеличивает вычислительную сложность многоканального FAP-алгоритма, если N велико. При этом примерно сохраняется отмеченное ранее соотношение между вычислительной сложностью этих алгоритмов при больших значениях N . На рис. 14.13 крайние точки на графиках соответствуют значениям $N=32$ и $N=2048$. Из рис. 14.13 также следует, что вычислительная сложность быстрого RLS-алгоритма растет значительно быстрее с увеличением

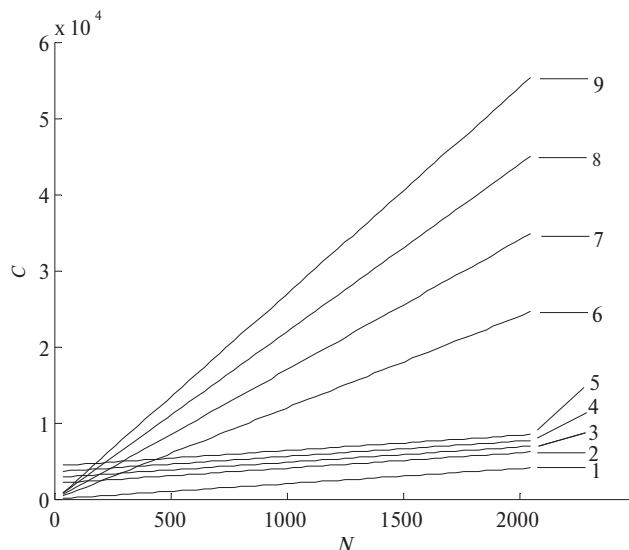


Рис. 14.13. Вычислительная сложность многоканальных алгоритмов: 1 — NLMS-алгоритм; 2 — FAP-алгоритм, $M=2$; 3 — FAP-алгоритм, $M=3$; 4 — FAP-алгоритм, $M=4$; 5 — FAP-алгоритм, $M=5$; 6 — быстрый RLS-алгоритм, $M=2$; 7 — быстрый RLS-алгоритм, $M=3$; 8 — быстрый RLS-алгоритм, $M=4$; 9 — быстрый RLS-алгоритм, $M=5$

числа каналов и общего числа ВК, чем сложность FAP-алгоритма при заданном значении L . Начиная с $N \approx 200$, становятся очевидными вычислительные преимущества FAP-алгоритма по сравнению с быстрыми RLS-алгоритмами.

В алгоритмах аффинных проекций также могут быть использованы линейные ограничения (см. главы 3, 5 и 10). В отношении алгоритмов аффинных проекций известна только версия одноканального линейно-ограниченного алгоритма с действительными ВК [185].

Ниже рассматривается многоканальный линейно-ограниченный алгоритм аффинных проекций с неодинаковым числом комплексных ВК в каналах адаптивного фильтра. С учетом уравнений (14.24)–(14.26) и [185], многоканальный линейно-ограниченный алгоритм аффинных проекций может быть представлен набором следующих уравнений:

$$\mathbf{h}_N(k) = \mathbf{P}_N [\mathbf{h}_N(k-1) + \mu(k) \mathbf{X}_{NL}(k) \boldsymbol{\epsilon}_L(k)] + \mathbf{a}_N, \quad (14.42)$$

$$\boldsymbol{\epsilon}_L(k) = [\mathbf{X}_{NL}^H(k) \mathbf{P}_N \mathbf{X}_{NL}(k) + \delta \mathbf{I}_L]^{-1} \boldsymbol{\alpha}_L^*(k), \quad (14.43)$$

$$\mathbf{P}_N = \mathbf{I}_N - \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{C}_{NJ}^H, \quad (14.44)$$

$$\mathbf{a}_N = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1} \mathbf{f}_J. \quad (14.45)$$

Из-за матрицы \mathbf{P}_N оценка вычислительной сложности линейно-ограниченного алгоритма аффинных проекций равна $O(N^2)$ арифметическим операциям. Умножение переменных на эту матрицу не позволяет получить такой алгоритм с вычислительной сложностью $O(N)$. Однако, используя ряд преобразований, аналогичных рассмотренным в настоящей главе и в главе 10, можно уменьшить вычислительную сложность алгоритма [185], которая примерно равна $2N^2 + (N+1)L^2 + (2N+1)L + O(L^3)$ умножений со сложением и $2N + L^2$ сложений, где $O(L^3)$ — вычислительная сложность обращения матрицы в уравнении (14.25).

Первым шагом уменьшения вычислительной сложности линейно-ограниченного алгоритма аффинных проекций является эквивалентное преобразование уравнения (14.42) к виду

$$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k)], \quad (14.46)$$

где $\mathbf{h}'_N(k) = \mathbf{h}_N(k-1) + \mu(k) \mathbf{X}_{NL}(k) \boldsymbol{\epsilon}_L(k)$, а $\mathbf{Q}_{NJ} = \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{C}_{NJ}]^{-1}$. Вычислительная сложность уравнения (14.46) примерно равна $2NJ$ умножениям со сложениями и $N+L$ сложениям. Таким образом, использование уравнения (14.46) вместо уравнения (14.42) заменяет операцию с вычислительной сложностью N^2 умножений со сложением на операцию со сложностью всего $2NJ$ таких операций. Вектор $\boldsymbol{\alpha}_L(k)$ можно определять с помощью уравнения (14.28).

Уравнение (14.43) получено на основании соотношения $\mathbf{P}_N = \mathbf{P}_N^H \mathbf{P}_N$, см. уравнение (5.200). В этом случае матрицу

$$\mathbf{R}_L(k) = \mathbf{X}_{NL}^H(k) \mathbf{P}_N \mathbf{X}_{NL}(k) \quad (14.47)$$

в уравнении (14.43) можно вычислять как

$$\mathbf{R}_L(k) = \begin{cases} \mathbf{R}_{2:L, 2:L}(k) = \mathbf{R}_{1:L-1, 1:L-1}(k-1) \\ \mathbf{R}_{1, 1:L}(k) = \tilde{\mathbf{x}}_N^H(k) \tilde{\mathbf{x}}_N(k-l), \quad \text{при } l=0:L-1, \\ \mathbf{R}_{2:L, 1}(k) = \tilde{\mathbf{x}}_N^H(k-l) \tilde{\mathbf{x}}_N(k), \quad \text{при } l=1:L-1 \end{cases} \quad (14.48)$$

где $\tilde{\mathbf{x}}_N(k) = \mathbf{P}_N \tilde{\mathbf{x}}_N(k)$. Таким образом, вместо вычисления полной матрицы (14.47), требующего NL^2 арифметических операций, можно вычислять лишь элементы первой строки или первого столбца, для чего требуется всего $N(L-1)$ арифметических операций.

Так как элементы столбцов матрицы $\tilde{\mathbf{X}}_{NL}(k)$ не удовлетворяют условию инвариантности к сдвигу, то использовать теорию линейного предсказания для получения обратной матрицы (подобно FAP-алгоритмам) в уравнении (14.43) не представляется возможным. Обращение этой матрицы можно осуществлять различными способами. Например, можно рекурсивно использовать лемму об обращении клеточных матриц (11.52). Вычислительная сложность такого обращения с учетом структуры матрицы равна $2L^3/3$ арифметическим операциям.

Полученный таким образом линейно-ограниченный алгоритм аффинных проекций приведен в табл. 14.14.

Таблица 14.14. Многоканальный линейно-ограниченный алгоритм аффинных проекций

Вычисления	Ссылки
Initialization: $\mathbf{x}_N(0) = \mathbf{0}_N$, $\hat{\mathbf{h}}_N(0) = \mathbf{0}_N$, $\mathbf{X}_{NL}(0) = \mathbf{0}_{NL}$, $\boldsymbol{\alpha}_L(0) = \mathbf{0}_L$, $\mathbf{p}_L(0) = \mathbf{0}_L$, $\mu(0) = \mu$, $\mathbf{h}_N(0) = \mathbf{Q}_{NJ} \mathbf{f}_J$	(14.14.0)
For $k = 1, 2, \dots, K$	
$\alpha(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k)$	(14.14.1)
$\boldsymbol{\alpha}_L(k) = \begin{bmatrix} \alpha(k) \\ [1-\mu(k-1)] \bar{\boldsymbol{\alpha}}_{L-1}(k-1) \end{bmatrix}$	(14.14.2)
$\boldsymbol{\epsilon}_L(k) = [\mathbf{R}_L(k) + \delta \mathbf{I}_L]^{-1} \boldsymbol{\alpha}_L^*(k)$	(14.14.3)
$\mathbf{p}_L(k) = \mu(k) \boldsymbol{\epsilon}_L(k) + \begin{bmatrix} 0 \\ \bar{\mathbf{p}}_{L-1}(k-1) \end{bmatrix}$	(14.14.4)
$\hat{\mathbf{h}}_N(k) = \hat{\mathbf{h}}_N(k-1) + \mathbf{x}_N(k-L+1) p_{L-1}(k)$	(14.14.5)
$\mathbf{h}'_N(k) = \hat{\mathbf{h}}_N(k) + \bar{\mathbf{X}}_{N(L-1)}(k) \bar{\mathbf{p}}_{L-1}(k)$	(14.14.6)
$\mathbf{h}_N(k) = \mathbf{h}'_N(k) + \mathbf{Q}_{NJ} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}'_N(k)]$	(14.14.7)
End for k	

В FAP-алгоритмах вектор $\mathbf{h}'_N(k)$ (14.14.6) непосредственно не вычисляется, что в конечном итоге позволяет получить такой алгоритм с вычислительной сложностью $O(N+L)$ арифметических операций. В линейно-ограниченном алгоритме аффинных проекций вектор $\mathbf{h}'_N(k)$ присутствует в явном виде, см. уравнение (14.14.7), что и вызывает необходимость вычисления этого вектора.

Сложность рассмотренного линейно-ограниченного алгоритма аффинных проекций равна $N^2+2N(L+J)+O(L^3)$ умножениям со сложениями, $(L-1)$ умножениям и $N+L^2+J$ сложениям, что примерно на $N^2+(N+1)L^2$ операций умножения со сложением меньше, чем сложность алгоритма [185], при условии, что вычисления обратной матрицы в уравнении (14.14.3) и в [185] выполняется одинаковым способом.

Результаты моделирования алгоритма (см. табл. 14.14) при наличии двух линейных ограничений (0 дБ АЧХ на частотах 1 и 2 кГц) в условиях, аналогичных рис. 14.12, приведены на рис. 14.14. Видно, что многоканальный линейно-ограниченный AP-алгоритм (см. рис. 14.14), подобно FAP-алгоритму без ограничений (см. рис. 14.12б), демонстрирует превосходство над линейно-ограниченным NLMS-алгоритмом с точки зрения длительности переходного процесса.

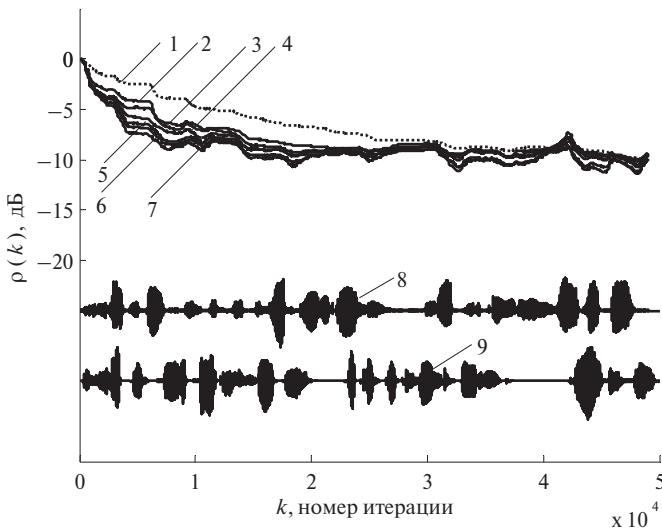


Рис. 14.14. Результаты моделирования линейно-ограниченного алгоритма аффинных проекций

Таким образом, рассмотренные многоканальные FAP-алгоритм и линейно-ограниченный алгоритм аффинных проекций могут быть использованы в адаптивных фильтрах с большим числом ВК при решении различных задач адаптивной фильтрации.

При любом значении $L > 1$, одинаковых значениях N_m и μ длительность переходного процесса в многоканальном FAP-алгоритме меньше, чем в многоканальном NLMS-алгоритме. Она заметно меньше при $L > 30$. Вычислительные преимущества FAP-алгоритма становятся заметными, если $N \gg L$.

14.8. Инициализация RLS-алгоритмов

Разрабатывать и проверять алгоритмы адаптивной фильтрации невозможно без наличия определенного критерия правильности их функционирования. Одним из таких очевидных критериев может служить сравнение поведения адаптивных фильтров с одинаковым числом ВК, использующих разные алгоритмы, при решении одинаковых задач и обработке одинаковых сигналов. Поскольку RLS-алгоритмы (в пределах своих классов) представляют собой тождественные преобразования определенных математических соотношений, то эти алгоритмы должны быть математически эквивалентными друг другу.

В литературных источниках инициализация (задание начальных условий) различных адаптивных алгоритмов часто осуществляется по-разному. С одной стороны, это не противоречит работоспособности алгоритмов, но с другой — не позволяет корректно их сравнивать даже в одинаковых условиях, так как из-за различной инициализации алгоритмы не обеспечивают одинаковых показателей качества, а значит, не являются полностью математически эквивалентными. Это затрудняет проверку алгоритмов в процессе их разработки и при решении прикладных задач.

В настоящем разделе рассматривается способ инициализации RLS-алгоритмов, обеспечивающий их корректное сравнение.

В RLS-алгоритмах с возрастающим окном обратная корреляционная матрица сигналов вычисляется с помощью MIL как (6.1.2), в алгоритмах с возрастающим окном и регуляризацией — как (14.4.2), (14.4.4), в алгоритмах со скользящим окном — как (14.6.2), (14.6.4) и в алгоритмах со скользящим окном и регуляризацией — как (14.8.2), (14.8.4), (14.8.6), (14.8.8). Эти уравнения требуют инициализации обратной корреляционной матрицы $\mathbf{R}_N^{-1}(0)$. Инициализация обычно осуществляется как $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{I}_N$, см. табл. 6.1. Здесь $\delta^2 \geq 0.01\sigma_x^2$ [80] — параметр начальной регуляризации, σ_x^2 — дисперсия входных сигналов адаптивного фильтра.

Подобная инициализация не может быть выполнена в быстрых RLS-алгоритмах, так как в таких алгоритмах инициализируется не обратная корреляционная матрица, а скалярные величины — энергии ошибок линейного предсказания. Как следствие, RLS- и быстрые RLS-алгоритмы становятся неэквивалентными, несмотря на то, что они являются результатом эквивалентных преобразований.



Если воспользоваться приемом, рассмотренным в [186] для обеспечения «мягкого» рестарта одного из RLS-алгоритмов, то можно получить условия эквивалентной инициализации для всех видов RLS-алгоритмов. Для этого при получении вектора ВК $\mathbf{h}_N(k)$ одноканального адаптивного фильтра вместо минимизации целевой функции (6.58), представляющей собой энергию апостериорных ошибок моделирования сигнала $d(k)$ выходным сигналом адаптивного фильтра $y(k)$, минимизируется целевая функция

$$E_{N,\min}(k) = \sum_{i=p}^k \lambda^{k-i} |d(i) - \mathbf{h}_N^H(k) \mathbf{x}_N(i)|^2 + \delta^2 \mathbf{h}_N^H(k) \Lambda_N^{-1}(k) \mathbf{h}_N(k). \quad (14.49)$$

Здесь $\Lambda_N^{-1}(k) = \text{diag}\{\lambda^k, \lambda^{k-1}, \dots, \lambda^{k-(N-1)}\}$ — диагональная матрица, $p=1$ — в случае возрастающего окна и $p=k-L+1$ — в случае скользящего окна. Следуя приемам, рассмотренным в главе 6, можно показать, что минимизация целевой функции (14.49) позволяет определить вектор ВК адаптивного фильтра как

$$\mathbf{h}_N(k) = [\mathbf{R}(k) + \delta^2 \Lambda_N^{-1}(k)]^{-1} \mathbf{r}_N(k). \quad (14.50)$$

Таким образом, матрица $\delta^{-2} \Lambda_N(0) = \delta^{-2} \text{diag}\{\lambda^0, \lambda^1, \dots, \lambda^{N-1}\}$ может быть использована в качестве начального значения обратной корреляционной матрицы при $k=0$. При такой инициализации можно получить начальные значения энергий линейного предсказания в быстрых RLS-алгоритмах, с помощью которых обеспечивается вычисление векторов ВК адаптивного фильтра, полностью соответствующее уравнению (14.50).

Вектор ВК можно также получить с помощью алгоритмов на основе прямого или обратного QR-разложения матрицы входных сигналов адаптивного фильтра (см. главы 7—9). Такие алгоритмы могут содержать или не содержать операции извлечения квадратного корня. RLS-алгоритмы на основе обратного QR-разложения без операций извлечения квадратного корня инициализируются как $\delta^{-2} \Lambda_N(0)$, а алгоритмы с операциями извлечения квадратного корня — как $\delta^{-1} \Lambda_N^{0.5}(0)$. RLS-алгоритмы на основе прямого QR-разложения без операций извлечения квадратного корня инициализируются как $\delta^2 \Lambda_N^{-1}(0)$, а с операциями извлечения квадратного корня — как $\delta \Lambda_N^{-0.5}(0)$. При такой инициализации все RLS-алгоритмы полностью эквивалентны друг другу.

Обычно в литературных источниках энергии ошибок линейного предсказания вперед и назад в быстрых RLS-алгоритмах инициализируются одинаковыми значениями δ^2 . Однако такая инициализация при экспоненциальном взвешивании сигналов не эквивалентна ни инициализации $\mathbf{R}_N^{-1}(0) = \delta^{-2} \mathbf{I}_N$, ни $\mathbf{R}_N^{-1}(0) = \delta^{-2} \Lambda_N$, а потому не позволяет сравнивать алгоритмы друг с другом. Для эквивалентной инициализации быстрых RLS-алгоритмов энергию ошибок линейного предсказания вперед, подобно уравнению (14.49), необходимо определять как

$$\begin{aligned}
E_{N,\min}^f(k) &= \mathbf{a}_{N+1}^H(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) \right\} \mathbf{a}_{N+1}(k) + \delta^2 \mathbf{a}_{N+1}^H(k) \Lambda_{N+1}^{-1}(k) \mathbf{a}_{N+1}(k) = \\
&= \sum_{i=p}^k \lambda^{k-i} |x(i) - \mathbf{h}_N^{fH}(k) \mathbf{x}_N(i-1)|^2 + \delta^2 [1, -\mathbf{h}_N^{fH}(k)] \begin{bmatrix} \lambda^k & \mathbf{0}_N^T \\ \mathbf{0}_N & \Lambda_N^{-1}(k-1) \end{bmatrix} \times \\
&\times \begin{bmatrix} 1 \\ -\mathbf{h}_N^f(k) \end{bmatrix} = \left\{ \sum_{i=p}^k \lambda^{k-i} x(i) x^*(i) + \delta^2 \lambda^k \right\} - \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N^H(i-1) x(i) \right\} \mathbf{h}_N^f(k) - \\
&- \mathbf{h}_N^{fH}(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N(i-1) x^*(i) \right\} + \\
&+ \mathbf{h}_N^{fH}(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) + \delta^2 \Lambda_N^{-1}(k-1) \right\} \mathbf{h}_N^f(k).
\end{aligned} \tag{14.51}$$

Из уравнения (14.51) следует, что при $k=0$, $\mathbf{x}_{N+1}(0)=\mathbf{0}_{N+1}$ и $\mathbf{h}_N^f(0)=\mathbf{0}_N$ энергия ошибок линейного предсказания вперед принимает значение

$$E_{N,\min}^f(0) = \delta^2. \tag{14.52}$$

Величина (14.52) используется в качестве начального значения при инициализации одноканальных вариантов быстрых RLS-алгоритмов.

Энергия ошибок линейного предсказания назад определяется как

$$\begin{aligned}
E_{N,\min}^b(k) &= \mathbf{b}_{N+1}^H(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) \right\} \mathbf{b}_{N+1}(k) + \delta^2 \mathbf{b}_{N+1}^H(k) \Lambda_{N+1}^{-1}(k) \mathbf{b}_{N+1}(k) = \\
&= \sum_{i=p}^k \lambda^{k-i} |x(i-N) - \mathbf{h}_N^{bH}(k) \mathbf{x}_N(i)|^2 + \delta^2 [-\mathbf{h}_N^{bH}(k), -1] \begin{bmatrix} \Lambda_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & \lambda^{k-N} \end{bmatrix} \times \\
&\times \begin{bmatrix} -\mathbf{h}_N^b(k) \\ 1 \end{bmatrix} = \left\{ \sum_{i=p}^k \lambda^{k-i} x(i-N) x^*(i-N) + \delta^2 \lambda^{k-N} \right\} - \\
&- \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N^H(i) x(i-N) \right\} \mathbf{h}_N^b(k) - \mathbf{h}_N^{bH}(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N(i) x^*(i-N) \right\} + \\
&+ \mathbf{h}_N^{bH}(k) \left\{ \sum_{i=p}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) + \delta^2 \Lambda_N^{-1}(k) \right\} \mathbf{h}_N^b(k).
\end{aligned} \tag{14.53}$$

Из уравнения (14.53) следует, что при $k=0$, $\mathbf{x}_{N+1}(0)=\mathbf{0}_{N+1}$ и $\mathbf{h}_N^b(0)=\mathbf{0}_N$ значение $E_{N,\min}^b(0)$ определяется как

$$E_{N,\min}^b(0) = \delta^2 \lambda^{-N}. \tag{14.54}$$

Величина (14.54) принимается в качестве начального значения энергии ошибок предсказания назад при инициализации быстрых RLS-алгоритмов.

В быстрых RLS-алгоритмах на основе обратного QR-разложения с операциями извлечения квадратного корня в процессе работы вычисляются переменные $[E_{N,\min}^f(k)]^{-0.5}$ и $[E_{N,\min}^b(k)]^{-0.5}$. Поэтому эти переменные инициализируются соответственно как $[E_{N,\min}^f(0)]^{-0.5} = \delta^{-1}$ и $[E_{N,\min}^b(0)]^{-0.5} = \delta^{-1} \lambda^{0.5N}$. В быстрых



RLS-алгоритмах на основе обратного QR-разложения без операций извлечения квадратного корня энергии ошибок линейного предсказания инициализируются как $[E_{N,\min}^f(0)]^{-1} = \delta^{-2}$ и $[E_{N,\min}^b(0)]^{-1} = \delta^{-2}\lambda^N$.

В быстрых лестничных RLS-алгоритмах (см. главу 12) также вычисляются энергии ошибок линейного предсказания, лишь с тем отличием, что эти энергии определяются для всех фильтров порядка $n=1, \dots, N$. Из уравнения (14.52) следует, что начальное значение энергии ошибок предсказания вперед не зависит от порядка фильтра, поэтому инициализация этих ошибок осуществляется как $E_{n,\min}^f(0) = \delta^2$ для всех $n=1, \dots, N$. В то же время из уравнения (14.54) следует, что энергии ошибок линейного предсказания назад необходимо инициализировать как $E_{n,\min}^b(0) = \delta^2\lambda^{-(n-1)}$, т.е. использовать разные значения для разных $n=1, \dots, N$. В лестничных алгоритмах на основе QR-разложения без операций извлечения квадратного корня инициализация энергии ошибок линейного предсказания осуществляется аналогичным образом. В таких же алгоритмах с операциями извлечения квадратного корня вычисляются параметры $[E_{n,\min}^f(k)]^{0.5}$ и $[E_{n,\min}^b(k)]^{0.5}$, поэтому в качестве их начальных значений принимаются значения $[E_{n,\min}^f(0)]^{0.5} = \delta$ и $[E_{n,\min}^b(0)]^{0.5} = \delta\lambda^{-0.5(n-1)}$.

Среди лестничных алгоритмов имеется также нормализованный алгоритм с обратными связями на основе апостериорных ошибок линейного предсказания (см. табл. 12.7), в котором инициализируются только значения энергии ошибок линейного предсказания при $n=1$. Из приведенных выше соотношений следует, что в этом случае $E_{0,\min}^f(0) = E_{0,\min}^b(0) = \delta^2$.

Рассмотренный способ инициализации одноканальных RLS- и быстрых RLS-алгоритмов распространяется и на алгоритмы для многоканальных адаптивных фильтров, которые в общем случае могут содержать неодинаковое число ВК N_m в каждом из $m=1, \dots, M$ каналов (см. рис. 1.3). Корреляционная матрица многоканального адаптивного фильтра определяется уравнением (3.14), а диагональная матрица $\Lambda_N^{-1}(k)$ — как

$$\Lambda_N^{-1}(k) = \text{diag} \left\{ \lambda^k, \lambda^{k-1}, \dots, \lambda^{k-(N_1-1)}, \dots, \lambda^k, \lambda^{k-1}, \dots, \lambda^{k-(N_m-1)}, \dots, \lambda^k, \lambda^{k-1}, \dots, \lambda^{k-(N_M-1)} \right\}. \quad (14.55)$$

Следовательно, при инициализации многоканальных RLS-алгоритмов с вычислительной сложностью $O(N^2)$ необходимо использовать диагональную матрицу

$$\Lambda_N(0) = \text{diag} \left\{ \lambda^0, \lambda^1, \dots, \lambda^{N_1-1}, \dots, \lambda^0, \lambda^1, \dots, \lambda^{N_m-1}, \dots, \lambda^0, \lambda^1, \dots, \lambda^{N_M-1} \right\}. \quad (14.56)$$

Выполнив преобразования, подобные (14.51) и (14.53), можно показать, что в быстрых RLS-алгоритмах для многоканальных адаптивных фильтров энергии ошибок линейного предсказания для каждого из $m=1, \dots, M$ каналов определяются как $E_{N_m,\min}^f(0) = \delta^2$ и $E_{N_m,\min}^b(0) = \delta^2\lambda^{-N_m}$. Эти энергии соответствующим

образом модифицируются в многоканальных быстрых лестничных алгоритмах и в алгоритмах на основе QR-разложения.

Рассмотренные способы инициализации также справедливы для линейно-ограниченных RLS-алгоритмов, рассмотренных в главе 10, и для параллельных версий RLS-алгоритмов, рассмотренных в настоящей главе.

На рис. 14.15 приведены результаты сравнительного моделирования в арифметике с плавающей точкой, демонстрирующие эквивалентность RLS-алгоритмов друг другу при использовании предложенной инициализации. Рассматривалась задача идентификации одноканального линейного фильтра при $N=32$.

При одинаковых числах ВК идентифицируемого импульсного отклика и адаптивного фильтра, а также при отсутствии аддитивного шума на входе сигнала $d(k)$ значение параметра ERLE в установившемся режиме ограничено лишь ошибками округления в процессе работы алгоритмов (кривые 1—5). Из рис. 14.15 следует, что в данном эксперименте значения ERLE в установившемся режиме для разных алгоритмов отличаются незначительно в силу разных ошибок округления в вычислениях, так как алгоритмы используют различные математически эквивалентные процедуры, содержащие различное число арифметических операций.

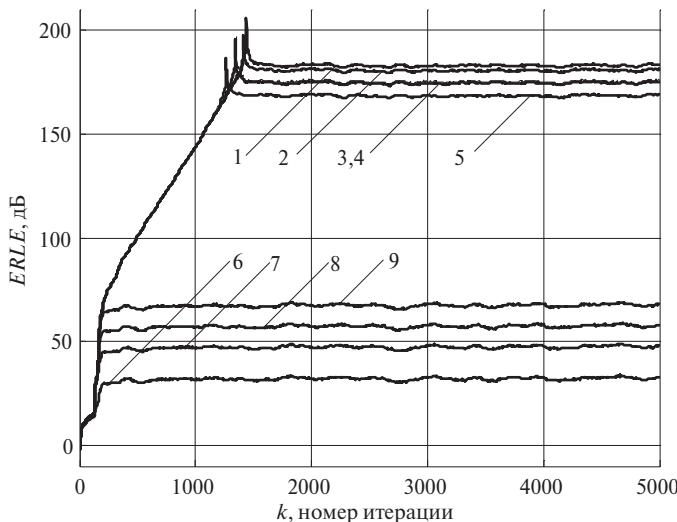


Рис. 14.15. Результаты моделирования: 1 — RLS-алгоритм на основе прямого QR-разложения (с операциями извлечения квадратного корня); 2 — лестничный алгоритм на основе априорных и апостериорных ошибок без обратных связей; 3 — RLS-алгоритм; 4 — RLS-алгоритм на основе прямого QR-разложения (с операциями извлечения квадратного корня); 5 — FK-алгоритм; 6 — ОСШ = 30 дБ; 7 — ОСШ = 40 дБ; 8 — ОСШ = 50 дБ; 9 — ОСШ = 60 дБ



На практике такие большие значения ERLE не достигаются из-за наличия на входе сигнала $d(k)$ аддитивного шума (природа которого может носить разный характер в зависимости от применения адаптивного фильтра). Для любого из приведенных на рис. 14.15 значений отношения сигнала-шум (ОСШ) все рассматриваемые адаптивные фильтры демонстрируют на каждой итерации одинаковое поведение в терминах параметра ERLE (кривые 6—9). В этом случае шум нарушает процесс адаптации, что приводит к смещению вектора ВК от оптимального значения, определяемого уравнением (14.50), и, как следствие, сигнал $d(k)$ «недостаточно точно» моделируется. При этом ошибки моделирования значительно превышают ошибки округления в вычислениях и на каждой итерации можно наблюдать одинаковые значения параметра ERLE, что свидетельствует об эквивалентности сравниваемых в эксперименте алгоритмов.

Таким образом, представленный способ инициализации обеспечивает математическую эквивалентность различных RLS-алгоритмов друг другу. Эквивалентность демонстрируется наблюдением на каждой итерации одинаковых сигналов на выходе адаптивного фильтра и других показателей качества его работы в случае, если сравниваемые фильтры имеют одинаковый порядок и обрабатывают одинаковые входные сигналы. Данный способ инициализации позволяет корректно сравнивать RLS-алгоритмы при решении различных прикладных задач.

14.9. Выводы по главе

В настоящей главе рассмотрены быстрые RLS-алгоритмы на основе обратного QR-разложения для многоканальных адаптивных фильтров с неодинаковым числом ВК в каналах, алгоритмы обработки нестационарных сигналов, основанные на оценке корреляционной матрицы сигналов адаптивного фильтра на скользящем окне отсчетов и динамической регуляризации этой матрицы, параллельные версии этих алгоритмов, а также рассмотрено применение RLS-алгоритмов со скользящим окном в качестве части вычислительных процедур в FAP-алгоритмах. Кроме того, обсуждены вопросы инициализации быстрых RLS-алгоритмов.

ГЛАВА 15

ПРИМЕНЕНИЕ АДАПТИВНЫХ АЛГОРИТМОВ

В настоящей главе рассматриваются стандартные и нестандартные средства моделирования адаптивных фильтров и их приложений на языке программирования MATLAB, а также в средах разработки LabVIEW и SystemVue. Кроме того, рассматриваются прикладные библиотеки адаптивных фильтров, реализованные на современных цифровых сигнальных процессорах отечественного и зарубежного производства.

15.1. Введение

Многообразие существующих на сегодняшний день алгоритмов адаптивной фильтрации (алгоритмов вычисления ВК адаптивного фильтра и его выходного сигнала), часть из которых рассмотрена в настоящей книге, условно показано на рис. 15.1. Адаптивные алгоритмы вытекают из теории оптимальной винеровской фильтрации. Прообразами большинства этих алгоритмов служат алгоритм Ньютона, алгоритм наискорейшего спуска и алгоритм по критерию наименьших квадратов. К простым относятся разновидности градиентных LMS- и NLMS-алгоритмов, а к сложным — разновидности RLS-алгоритмов на базе MIL и QR-разложения. Промежуточный класс алгоритмов образуют алгоритмы аффинных проекций.

Наличие разнообразных алгоритмов адаптивной фильтрации дает возможность разработчикам радиоэлектронной аппаратуры принимать компромиссные решения между вычислительной сложностью и эффективностью адаптивных фильтров при их разработке или использовании. Простые градиентные алгоритмы, преимущественно ориентированы на аппаратную реализацию адаптивных фильтров. Сложные RLS-алгоритмы и алгоритмы аффинных проекций, включая их быстрые версии, в основном ориентированы на программную реализацию. Несмотря на сложность, RLS-алгоритмы являются наиболее эффективными, а потому также представляют практический интерес наряду с простыми градиентными алгоритмами.

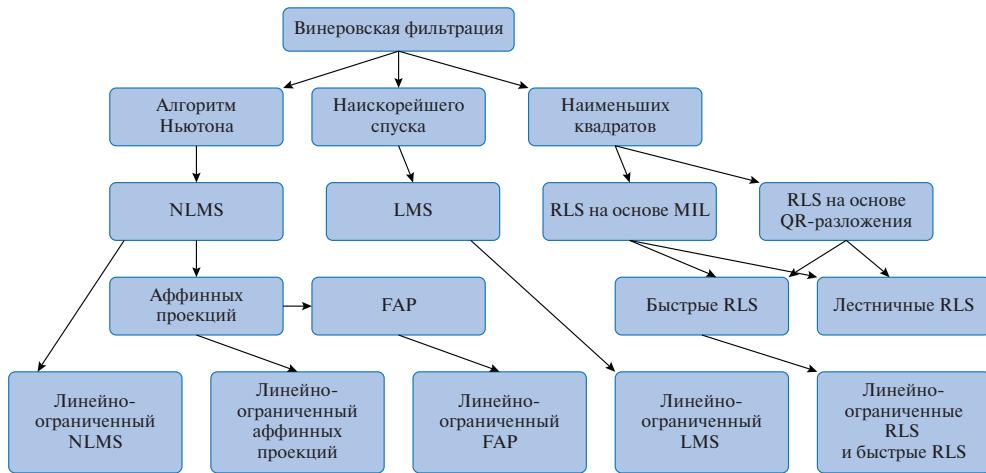


Рис. 15.1. Многообразие алгоритмов адаптивной фильтрации сигналов

Большая часть книги посвящена RLS-алгоритмам (рис. 15.2), основные разновидности которых приведены на рис. 15.3.

Многообразие RLS-алгоритмов (см. рис. 15.2) образуется за счет использования различных оконных функций при оценке корреляционной матрицы входных сигналов адаптивного фильтра и ее динамической регуляризации, а также структурнонезависимой параллельной реализации с помощью одного, двух или четырех ЦСП (см. главу 14). Это многообразие применимо ко всем разновидностям RLS-алгоритмов (см. рис. 15.3 и главы 6–11), включая лестничные адаптивные алгоритмы без линейных ограничений (см. главу 12).

Рис. 15.1–15.3 свидетельствуют о значительных успехах в области адаптивной обработки сигналов, достигнутых на протяжении всей истории ее развития. Эти достижения не могли пройти мимо производителей современных

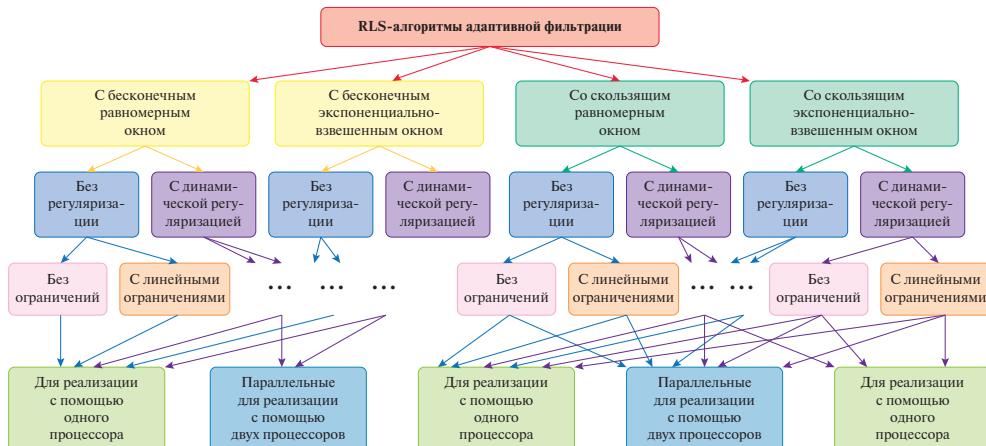


Рис. 15.2. Многообразие RLS-алгоритмов адаптивной фильтрации сигналов



Рис. 15.3. Разновидности RLS-алгоритмов адаптивной фильтрации сигналов

средств программирования и моделирования, таких как MATLAB, LabVIEW и SystemVue [187–189], а также производителей современных ЦСП [190–193], которые реализовали в своих прикладных библиотеках некоторые разновидности адаптивных фильтров и их приложений. Ряд программных реализаций адаптивных фильтров представлен в работах [43, 194–206]. Вопросы моделирования адаптивных фильтров освещаются в некоторых современных книгах по ЦОС [27, 207–209].

Ниже рассматриваются адаптивные фильтры, представленные в указанных и ряде других источников информации.

15.2. Моделирование адаптивных фильтров с помощью приложений DSP System Toolbox языка MATLAB

Язык MATLAB [210] на сегодняшний день является основным программным инструментом для моделирования алгоритмов, устройств и систем, использующих ЦОС. Его лидирующая роль подтверждена и тем, что такие средства разработки, как LabVIEW [211] и SystemVue [212, 213], поддерживают возможность совместной работы с языком MATLAB. Кроме того, в этих программных продуктах содержатся урезанные ядра языка MATLAB (MathScript в LabVIEW и MathLang в SystemVue), в которых реализованы основной синтаксис и набор базовых функций, позволяющих проводить совместное моделирование программ на языке MATLAB в среде LabVIEW или SystemVue без установки языка MATLAB на персональном компьютере.

Поэтому неудивительно, что в стандартном приложении DSP System Toolbox языка MATLAB, где сосредоточены алгоритмы ЦОС, также имеется около тридцати различных разновидностей адаптивных алгоритмов (см. табл. 15.1) [27], реализованных в виде объектов. Эти объекты могут быть использованы для моделирования различных адаптивных приложений. В частности, они использованы в адаптивных эквалайзерах стандартного приложения Communication Toolbox.

Таблица 15.1. Объекты адаптивных алгоритмов языка MATLAB

Конструктор	Алгоритм
Семейство LMS-алгоритмов	
adaptfilt.adjlms	Adjoint LMS-алгоритм (сопряженный)
adaptfilt.blms	Блочный LMS-алгоритм
adaptfilt.blmsfft	Блочный LMS-алгоритм в частотной области с применением БПФ
adaptfilt.dlms	Delayed LMS-алгоритм (с задержкой обновления ВК адаптивного фильтра)
adaptfilt.filtxlms	Filtered-x LMS-алгоритм (с дополнительной фильтрацией выходного сигнала фильтра перед формированием сигнала ошибки)
adaptfilt.lms	LMS-алгоритм (классический)
adaptfilt.nlms	NLMS-алгоритм
adaptfilt.sd	Sign-Data LMS-алгоритм с использованием для адаптации только знаков данных, содержащихся в линии задержки адаптивного фильтра (вместо $x_N(k)$ используется $\text{sign}\{x_N(k)\}$)
adaptfilt.se	Sign-Error LMS-алгоритм с использованием для адаптации только знака сигнала ошибки (вместо $\alpha(k)$ используется $\text{sign}\{\alpha(k)\}$)
adaptfilt.ss	Sign-Sign LMS-алгоритм с использованием для адаптации только знаков сигнала ошибки и данных, содержащихся в линии задержки адаптивного фильтра (комбинация двух предыдущих вариантов)
Семейство RLS-алгоритмов	
adaptfilt.ftf	Быстрый RLS-алгоритм (FTF)
adaptfilt.hrls	RLS-алгоритм с использованием преобразования Хаусхолдера
adaptfilt.hswrls	RLS-алгоритм со скользящим окном и использованием преобразования Хаусхолдера
adaptfilt.qrdrls	RLS-алгоритм с использованием QR-разложения
adaptfilt.rls	RLS-алгоритм (классический, на базе MIL)
adaptfilt.swfft	Быстрый RLS-алгоритм со скользящим окном
adaptfilt.swrls	RLS-алгоритм со скользящим окном

Таблица 15.1. (окончание)

Конструктор	Алгоритм
	Алгоритмы аффинных проекций
adaptfilt.ap	Алгоритм аффинных проекций с прямым обращением матрицы
adaptfilt.apru	Алгоритм аффинных проекций с рекурсивным обновлением обратной матрицы
adaptfilt.bap	Блочный алгоритм аффинных проекций
	Алгоритмы адаптации в частотной области
adaptfilt.fdaf	LMS-алгоритм в частотной области (на базе БПФ) с индивидуальной нормировкой размера шага в каждом частотном канале
adaptfilt.pbfdaf	Нормированный LMS-алгоритм в частотной области (на базе БПФ), использующий деление сигнала на блоки малого размера во временной области
adaptfilt.pbufdaf	NLMS-алгоритм в частотной области (на базе БПФ), использующий деление сигнала на блоки малого размера во временной области, без ограничения длины эквивалентной импульсной характеристики
adaptfilt.tdafdct	Адаптивный фильтр, использующий LMS-алгоритм после предварительного применения к входному сигналу дискретного косинусного преобразования
adaptfilt.tdafdft	Адаптивный фильтр, использующий LMS-алгоритм после предварительного применения к входному сигналу дискретного преобразования Фурье
adaptfilt.ufdaf	Нормированный LMS-алгоритм в частотной области (на базе БПФ), без ограничения длины эквивалентной импульсной характеристики
	Адаптивные алгоритмы для лестничных фильтров
adaptfilt.gal	Gradient-Adaptive Lattice LMS (алгоритм адаптации для лестничного фильтра, основанный на градиентном спуске)
adaptfilt.lsl	Лестничный RLS-алгоритм
adaptfilt.qrdlsl	Лестничный RLS-алгоритм на базе QR-разложения

Для создания объекта адаптивного фильтра служит функция `adaptfilt`. При ее вызове необходимо указать требуемый алгоритм адаптивной фильтрации из табл. 15.1 как

```
ha = adaptfilt.algorithm(...).
```

Набор входных параметров конструктора, который указывается в скобках, зависит от используемого алгоритма.

Обработка сигналов адаптивным фильтром осуществляется с помощью функции `[y, e] = filter(ha, x, d)`, где `ha` — объект адаптивного фильтра,

x — вектор отсчетов входного сигнала, d — вектор отсчетов требуемого сигнала. Результатами работы функции являются векторы отсчетов выходного сигнала y и сигнала ошибки фильтрации e . Более подробное описание использования объектов адаптивных фильтров в языке MATLAB и реализации с их помощью таких приложений адаптивной фильтрации, как идентификация неизвестной системы, линейное предсказание, шумоподавление и выравнивание каналов связи, приведены в [27].

На рис. 15.4 показаны два примера использования объектов адаптивных алгоритмов в языке MATLAB при решении задачи идентификации импульсного отклика КИХ-фильтра с 32 ВК. Рис. 15.4 отражает одно из рассмотренных в главах 5 и 6 сравнительных свойств LMS- и RLS-алгоритмов, а именно: более короткий переходной процесс (см. кривую «Сигнал ошибки») в случае последнего алгоритма. На рисунках «Требуемый сигнал» — это сигнал адаптивного фильтра $d(k)$, «Выходной сигнал» — сигнал адаптивного фильтра $y(k)$, «Сигнал ошибки» — сигнал $\alpha(k)$, «КИХ-фильтр» — ВК идентифицируемого импульсного отклика w_n и «Адаптивный фильтр» — ВК адаптивного фильтра h_n на последней 500-й итерации, где $n=1, \dots, N$ — номера ВК, а $N=32$.

Скрипт для получения графиков на рис. 15.4а приведен ниже.

```
randn('state',100);
x = randn(1,500); % Входной сигнал фильтра
b = fir1(31,0.5); % Идентифицируемый КИХ-фильтр
n = 0.1*randn(1,500); % Аддитивный шум
d = filter(b,1,x)+n; % Требуемый сигнал
mu = 0.008; % Шаг сходимости LMS-алгоритма
ha = adaptfilt.lms(32,mu);
[y,e] = filter(ha,x,d);
subplot(2,1,1); plot(1:500,[d;y;e]);
title('Идентификация КИХ-фильтра');
legend('Требуемый сигнал','Выходной сигнал','Сигнал ошибки');
xlabel('Номер итерации'); ylabel('Значение сигнала');
subplot(2,1,2); stem([b,ha.coefficients.']);
legend('КИХ-фильтр','Адаптивный фильтр');
xlabel('Номер весового коэффициента'); ylabel('Значение весового коэффициента'); grid on;
```

Этот скрипт дает представление о порядке использования объектов (см. табл. 15.1). Скрипты для получения аналогичных рисунков при использовании других алгоритмов адаптивной фильтрации, реализованных на языке MATLAB, приведены в «Справочной системе» приложения DSP System Toolbox.

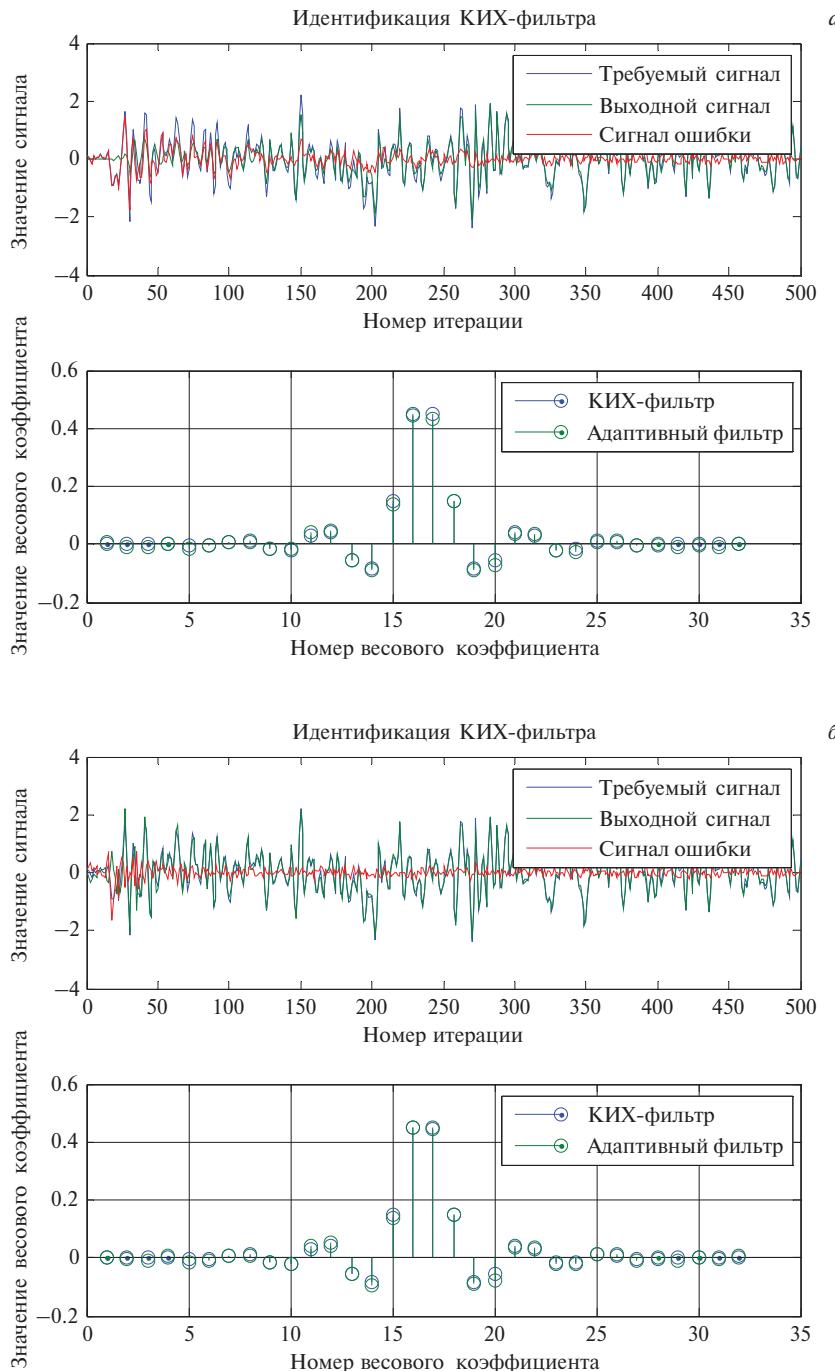


Рис. 15.4. Моделирование адаптивных фильтров: а) на базе LMS-алгоритма; б) на базе RLS-алгоритма

Несмотря на большую популярность и широкое распространение языка MATLAB, реализация в нем адаптивных алгоритмов в виде объектов имеет ряд недостатков, ограничивающих использование этих алгоритмов при моделировании некоторых приложений [27]. Этих недостатков лишена рассматриваемая далее библиотека алгоритмов адаптивной фильтрации [43, 195, 196].

15.3. Нестандартные программные средства для моделирования адаптивных фильтров на языке MATLAB

Библиотека [43, 195, 196] включает практически все разновидности алгоритмов адаптивной фильтрации, показанные на рис. 15.1—15.3, для одноканальных и многоканальных адаптивных фильтров с действительными и комплексными ВК. Все графики, приведенные в настоящей книге, получены путем моделирования на языке MATLAB с использованием элементов этой библиотеки.

Все алгоритмы библиотеки реализованы в виде программ, а не объектов или функций. Такая реализация удобна при исследовании свойств алгоритмов или проверке их работоспособности, так как в этом случае имеется доступ ко всем «внутренним» переменным алгоритма. Ее также удобно использовать при переносе алгоритмов на язык Assembler интересуемого ЦСП, так как алгоритмы библиотеки запрограммированы на поитерационную, а не векторную обработку сигналов.

Большая часть разработанных адаптивных алгоритмов интегрирована в графический интерфейс пользователя (ГИП), показанный на рис. 15.5.

С помощью этого приложения можно исследовать свойства интересуемого алгоритма адаптивной фильтрации, функционирующего под воздействием внутренних или внешних тестовых сигналов. В последнем случае используются сигналы, передаваемые в виде файлов данных. ГИП позволяет выбирать интересующий пользователя алгоритм, определять число каналов адаптивного фильтра и тип арифметики (действительная или комплексная), а также задавать параметры, зависящие от выбранного алгоритма. Входные/выходные сигналы моделирования могут быть сохранены в файлах для использования в качестве тестовых векторов при переносе адаптивных алгоритмов на интересуемые языки программирования и вычислительные платформы.

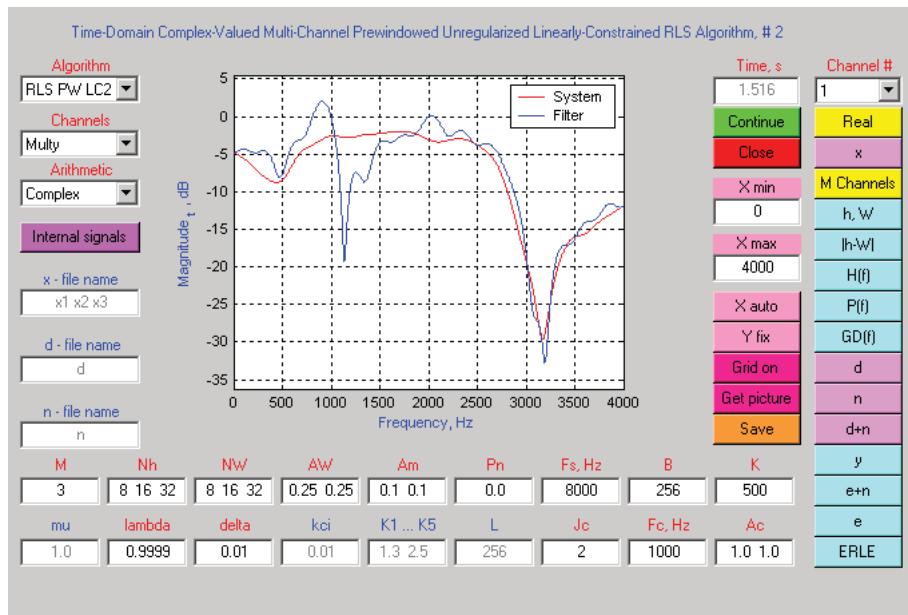


Рис. 15.5. ГИП алгоритмов адаптивной фильтрации

Ряд алгоритмов адаптивной фильтрации на языке MATLAB также реализован в библиотеке «FREE-ASPT for MATLAB» [214]. Однако бесплатно распространяемая демо-версия этой библиотеки может быть использована только для моделирования адаптивных фильтров с ограниченным числом ВК. Кроме того, программный код этих алгоритмов является закрытым, что не позволяетзнакомиться с особенностями реализации алгоритмов и является неудобным при исследовании их свойств.

15.4. Моделирование приложений адаптивной фильтрации с помощью Adaptive Filter Toolkit среды разработки LabVIEW

Другим средством, широко используемым сегодня для моделирования и создания различных приложений, среди которых и приложения ЦОС, является среда разработки LabVIEW [211]. Существует два основных отличия LabVIEW от большинства других языков программирования.

Во-первых, LabVIEW реализует концепцию графического программирования языка G, поэтому исходный код представляет собой блок-диаграмму, т.е. соединенные друг с другом блоки (пиктограммы) элементов языка. Несмотря на такой подход, в языке G используются те же конструкции и ме-

тоды программирования, что и в других языках: типы данных, циклы, переменные, рекурсия, обработка событий и объектно-ориентированное программирование.

Во-вторых, LabVIEW поддерживает выполнение кода, написанного на языке G, в режиме потока данных (потоковое программирование), в то время как традиционные текстовые языки (например С и С++) обеспечивают выполнение кода в виде последовательности команд. В основе языков потокового программирования (таких как G, Agilent VEE, Microsoft Visual Programming Language и Apple Quartz Composer) лежит концепция потока данных, который определяет последовательность выполнения функциональных узлов программы.

Прибор, реализуемый средствами LabVIEW, состоит из двух основных компонент: лицевой панели и блок-диаграммы, между которыми существует однозначная взаимосвязь. Лицевая панель представляет собой ГИП и служит для ввода/вывода цифровых, логических и графических данных. Блок-диаграмма содержит графический код лицевой панели (блоки ее элементов), а также другие элементы программы в соответствии с реализуемым алгоритмом.

Графические программы в LabVIEW называются виртуальными приборами (Virtual Instruments, VI). Каждый блок графической программы выполняется тогда, когда на всех его входах присутствуют требуемые данные, после чего выходные данные блока передаются на другие блоки. Используемое в LabVIEW потоковое программирование позволяет автоматически распараллеливать выполнение кода. В отличие от языков с последовательным выполнением команд, графические языки изначально содержат в себе информацию о том, какие части кода следует выполнять параллельно. Параллелизм становится все более важным способом увеличения производительности программ, которые не могут эффективно работать на последних моделях многоядерных процессоров по причине последовательного выполнения кода.

Помимо поддержки многопоточности на многоядерных системах, среда LabVIEW позволяет создавать приложения для параллельных вычислений на ПЛИС, в которых выполнение независимых участков кода организуется абсолютно независимыми участками микросхемы, не влияющими на производительность друг друга.

Указанные особенности, а также большой набор различных блоков, имеющихся сегодня в LabVIEW, позволяют создавать графические программы как для моделирования, так и для обработки сигналов, поступающих в реальном времени от внешних объектов.

Блоки для создания адаптивных КИХ-фильтров и некоторые из их приложений сосредоточены в Adaptive Filter Toolkit (см. табл. 15.2).

Таблица 15.2. Блоки адаптивных фильтров приложения Adaptive Filter Toolkit

Алгоритм	Ком- плексная арифме- тика	Приложения			
		Шумо- очистка	Компен- сация эхо- сигналов	Активное подавление шума	Выравнива- ние каналов связи
LMS	+	+	+	+	+
NLMS	+	+	+	+	+
Sign-Error LMS	—	+	+	+	—
Блочный LMS в частотной обла- сти (на базе БПФ)	—	+	+	—	—
Filtered-x LMS	—	—	—	+	—
RLS	+	—	—	—	+
QR RLS	+	—	—	—	+
FTF	+	+	+	—	+

Adaptive Filter Toolkit и модуль LabVIEW FPGA позволяют генерировать код для ПЛИС (Field-Programmable Gate Array, FPGA) и создавать на них прототипы адаптивных LMS КИХ-фильтров в целочисленной арифметике. Такие адаптивные фильтры могут обрабатывать сигналы с частотой дискретизации до нескольких сотен килогерц, что зависит от числа ВК адаптивного фильтра, числа разрядов входных и внутренних данных, типа используемой ПЛИС и ее тактовой частоты.

Проектирование адаптивных фильтров под ПЛИС осуществляется в три этапа: моделирование на хост-машине в арифметике с плавающей точкой, моделирование в арифметике с фиксированной точкой и собственно генерация FPGA-кода.

Моделирование в арифметике с плавающей точкой позволяет определить параметры адаптивного алгоритма и диапазон возможных значений его ВК, что требуется для определения точности представления ВК в арифметике с фиксированной точкой или в конечном итоге в целочисленной арифметике в FPGA.

С помощью моделирования в арифметике с фиксированной точкой можно проверить, удовлетворяют ли параметры целочисленного представления данных при реализации адаптивного фильтра в FPGA (рис. 15.6). На этом этапе можно принимать компромиссные решения между числом разрядов арифметических операций (а значит, и требуемыми ресурсами для FPGA-реализации) и точностью работы адаптивного фильтра.



В случае удовлетворительных результатов моделирования полученные параметры можно сохранить и создать FPGA-код адаптивного фильтра, использующего эти параметры.

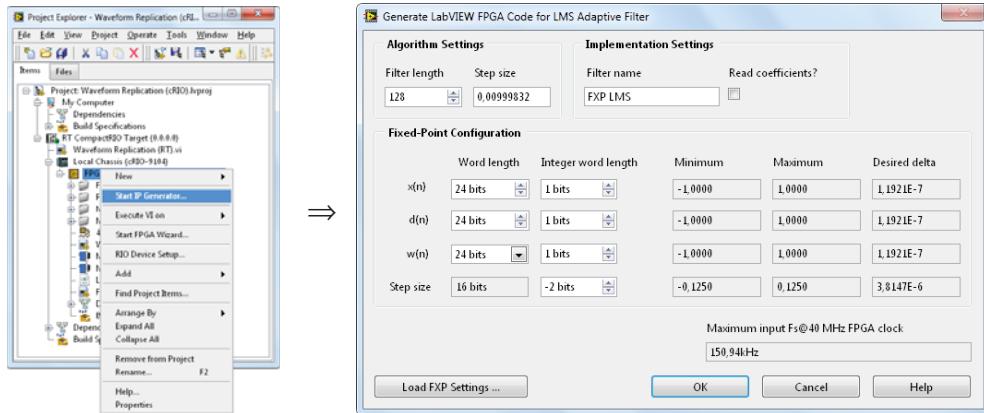


Рис. 15.6. Параметры для целочисленной FPGA-реализации адаптивного фильтра

Как следует из примера (см. рис. 15.6), адаптивный фильтр на базе LMS-алгоритма с числом ВК $N = 128$, с 24-разрядным представлением входных данных и ВК, а также 16-разрядным представлением шага сходимости может обрабатывать входные сигналы с частотой дискретизации 150.94 кГц при использовании ПЛИС Virtex-II (3000000 вентиляй), работающей на тактовой частоте 40 МГц. Такой фильтр требует примерно 3% вентиляй от их общего числа в указанной ПЛИС.

При компиляции кода может появиться возможность умножить тактовую частоту и получить, например, 80 МГц вместо 40 МГц. При этом частота дискретизации обрабатываемых сигналов также удвоится.

При проектировании FPGA-код компилируется в VHDL-код. Компилятор оптимизирует VHDL-код таким образом, чтобы программа максимально эффективно выполнялась на ПЛИС. Параметрами оптимизации являются число задействованных логических ячеек, скорость выполнения программы, задействованная оперативная память на ПЛИС и т. д. Результаты оптимизации, т. е. задействованные ресурсы ПЛИС, выводятся в виде отчета.

В приложении Adaptive Filter Toolkit проект адаптивного фильтра на ПЛИС, в частности, используется при решении прикладной задачи шумоподавления в реальном масштабе времени. Примеры решения с помощью Adaptive Filter Toolkit других прикладных задач, таких как выделение электрокардиограммы зародыша на фоне сердцебиения матери, адаптивная компенсация эхо-сигналов, адаптивная идентификация системы, адаптивное выравнивание канала связи,

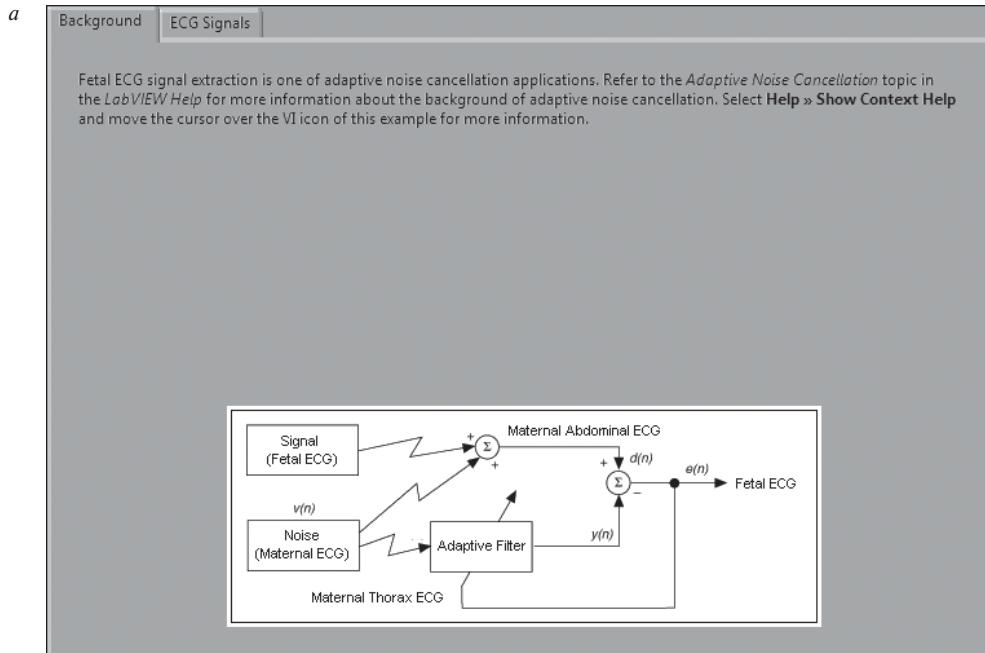


Рис. 15.7. Шумоподавление: а) принцип; б) виртуальный прибор



адаптивное обратное управление, адаптивное линейное предсказание, активное подавление шума, и ряд других приведены в «Справочной системе» среды LabVIEW. Результаты выполнения этих примеров показаны на рис. 15.7—15.13.

К стандартным бесплатно распространяемым адаптивным приложениям LabVIEW также относится приложение LMS Adaptive Filter Lab Demo, с помощью которого можно исследовать адаптивный фильтр на базе LMS-алгоритма при решении задачи идентификации линейной системы. Это же приложение позволяет проводить сравнительное моделирование адаптивного фильтра, использующего разные варианты LMS-алгоритма: классический LMS-алгоритм, блочный LMS-алгоритм, Sing-Error LMS-алгоритм и NLMS-алгоритм. Примеры результатов использования этого приложения показаны на рис. 15.14—15.16, а рис. 15.17 дает представление о том, как выглядят графические программы в среде LabVIEW.

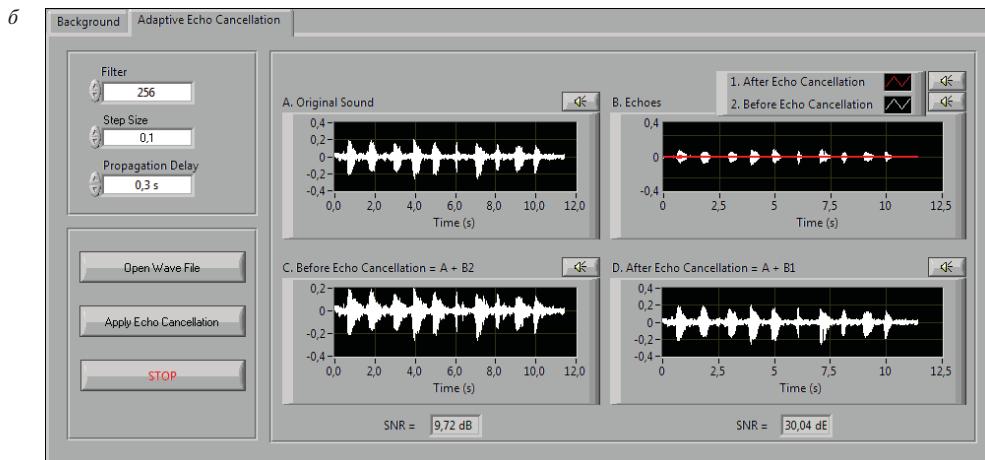
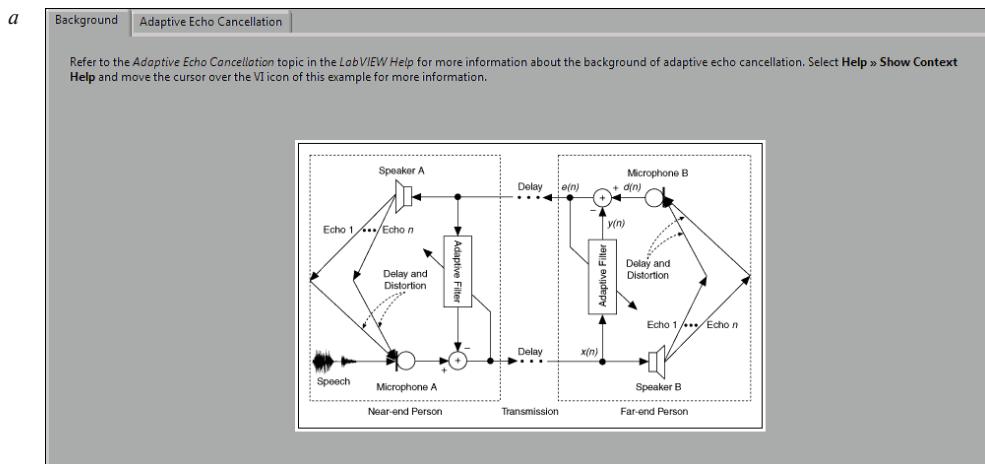


Рис. 15.8. Адаптивная компенсация эхо-сигналов: а) принцип; б) виртуальный прибор

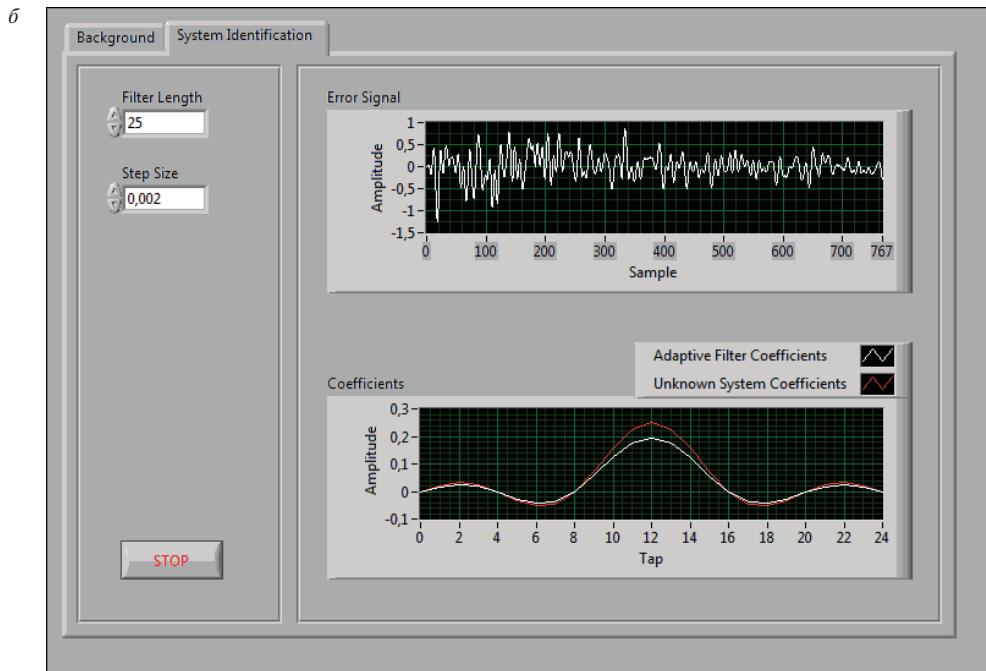
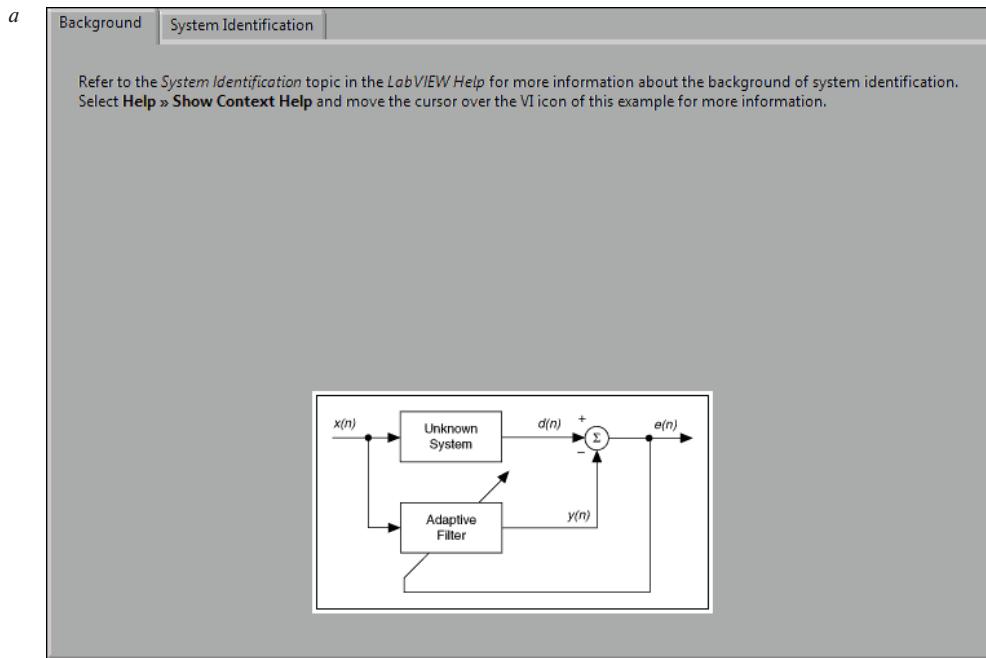


Рис. 15.9. Адаптивная идентификация системы: а) принцип; б) виртуальный прибор

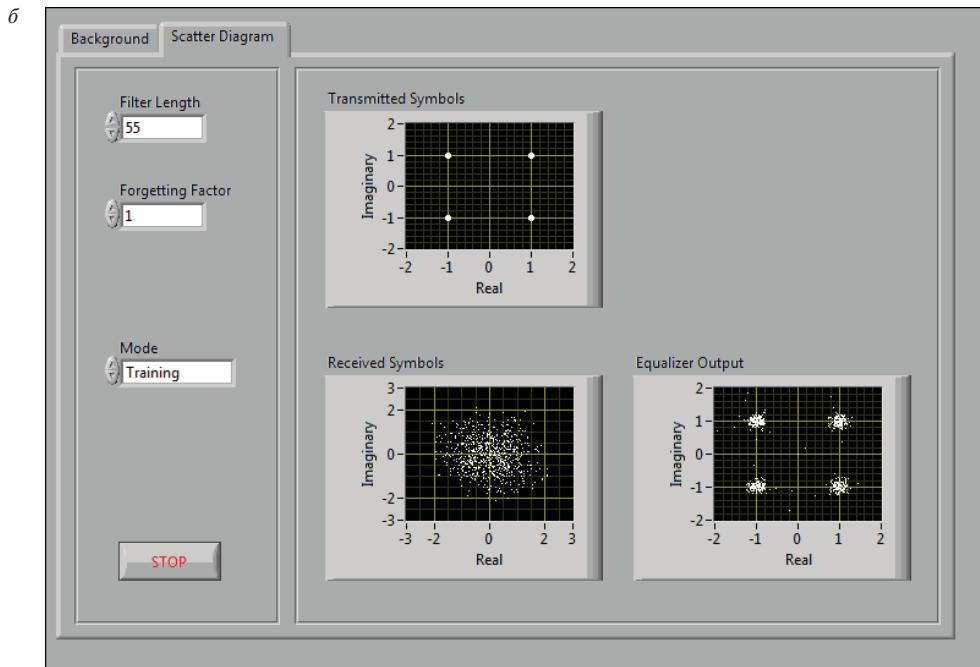
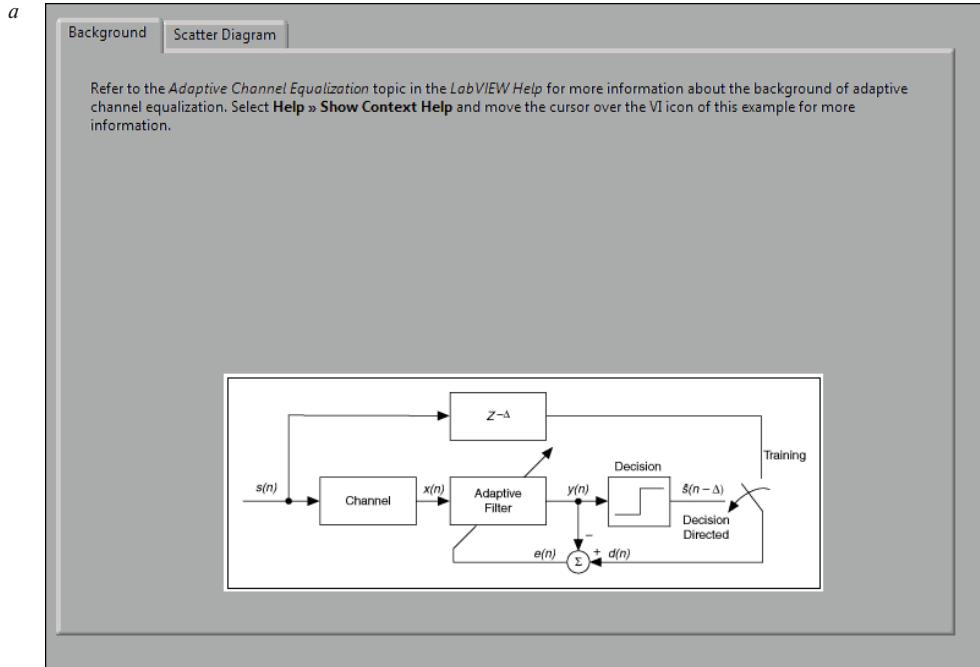


Рис. 15.10. Адаптивное выравнивание канала связи: а) принцип; б) виртуальный прибор

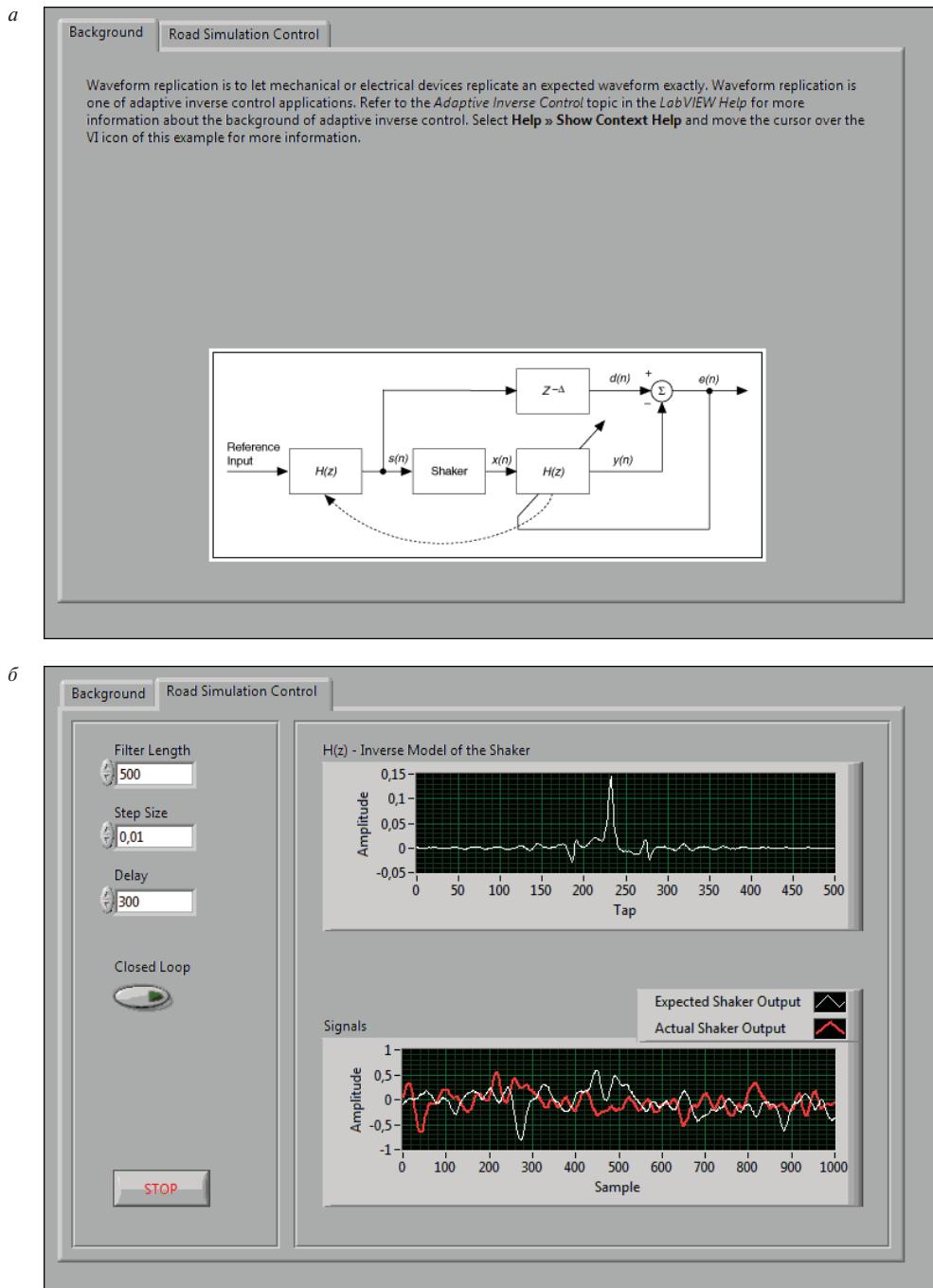
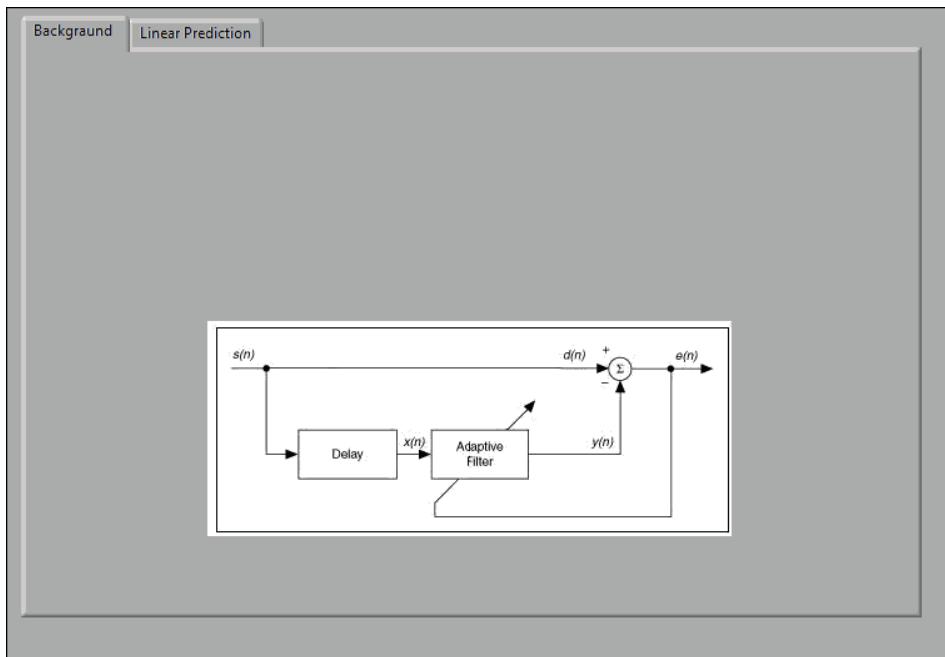


Рис. 15.11. Адаптивное обратное управление: а) принцип; б) виртуальный прибор



а



б

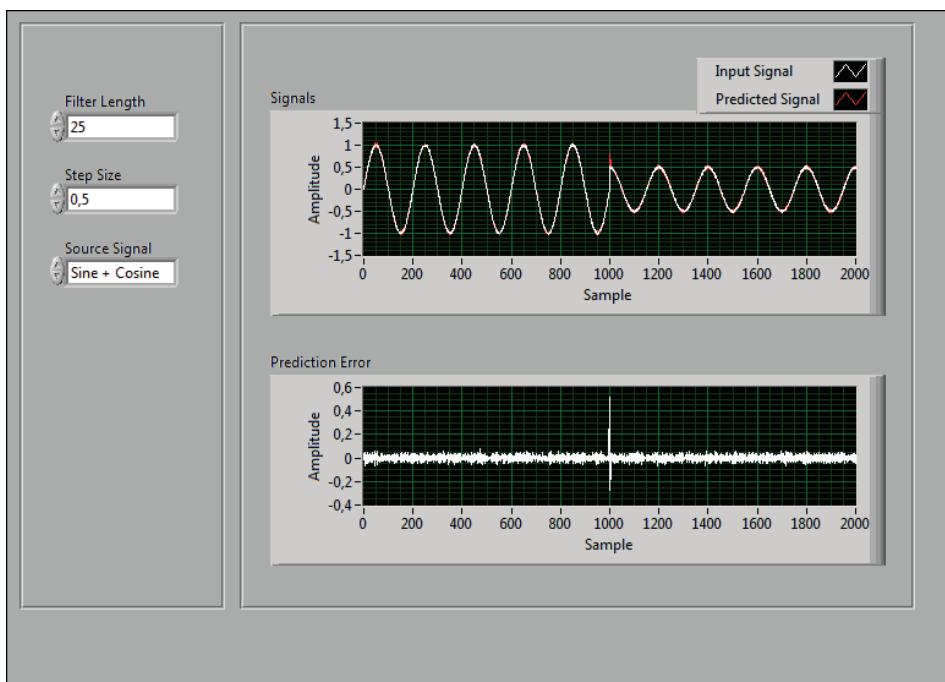


Рис. 15.12. Адаптивное линейное предсказание: а) принцип; б) виртуальный прибор

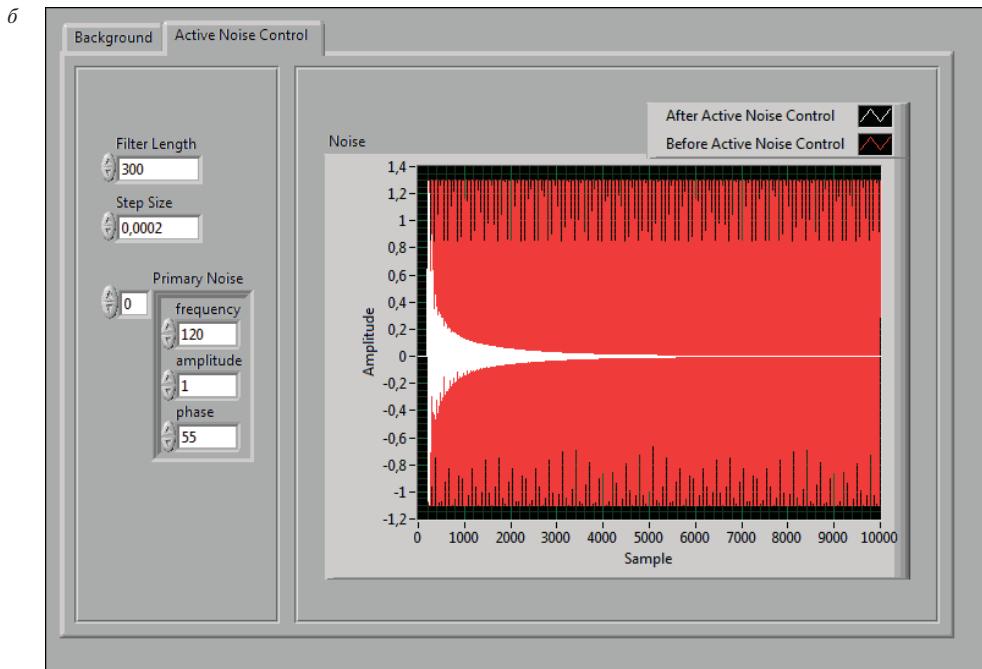
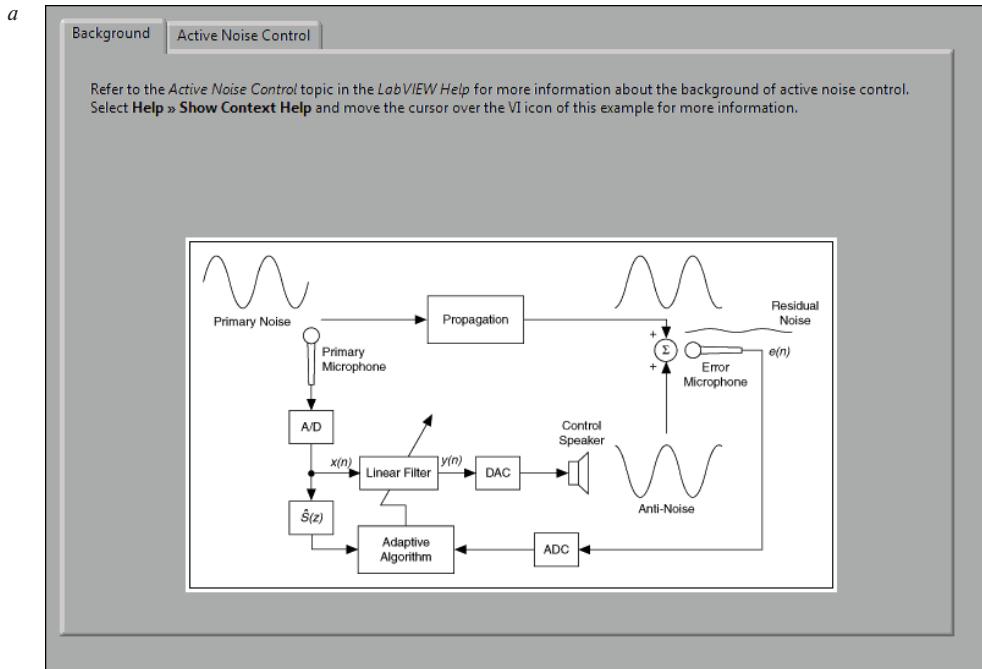


Рис. 15.13. Активное подавление шума: а) принцип; б) виртуальный прибор

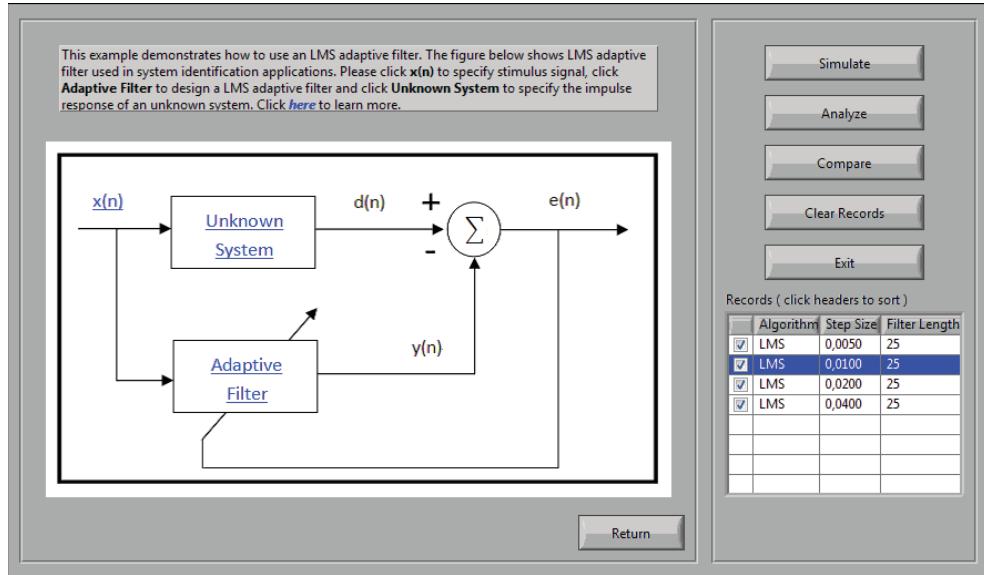


Рис. 15.14. LMS Adaptive Filter Lab Demo: принцип

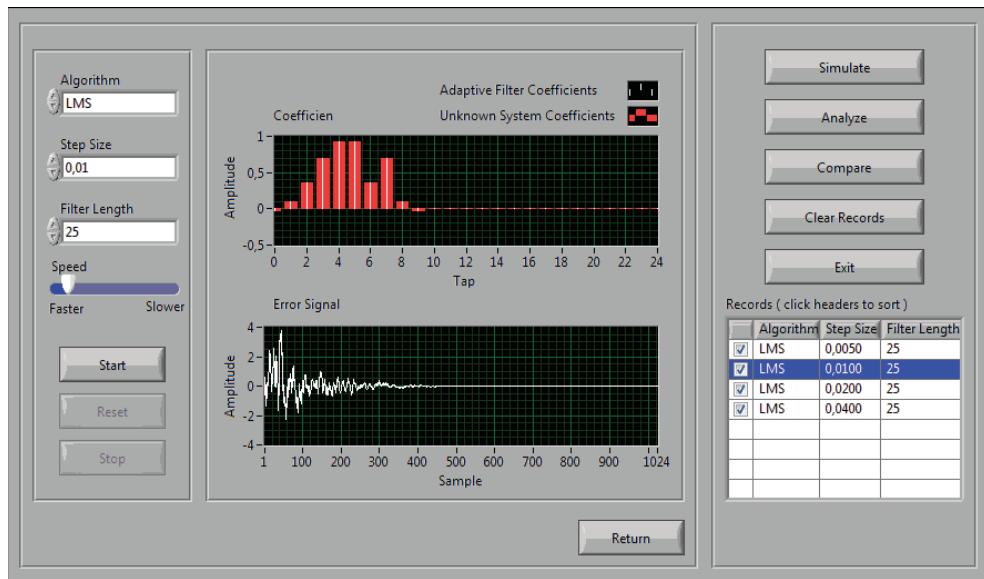


Рис. 15.15. LMS Adaptive Filter Lab Demo: переходный процесс

Таким образом, среда разработки LabVIEW имеет библиотеку основных разновидностей алгоритмов адаптивной фильтрации (см. табл. 15.2), с помощью которых можно проводить исследование различных адаптивных устройств, а также обрабатывать сигналы, поступающие посредством оборудования компании National Instruments с физических объектов окружающего мира.

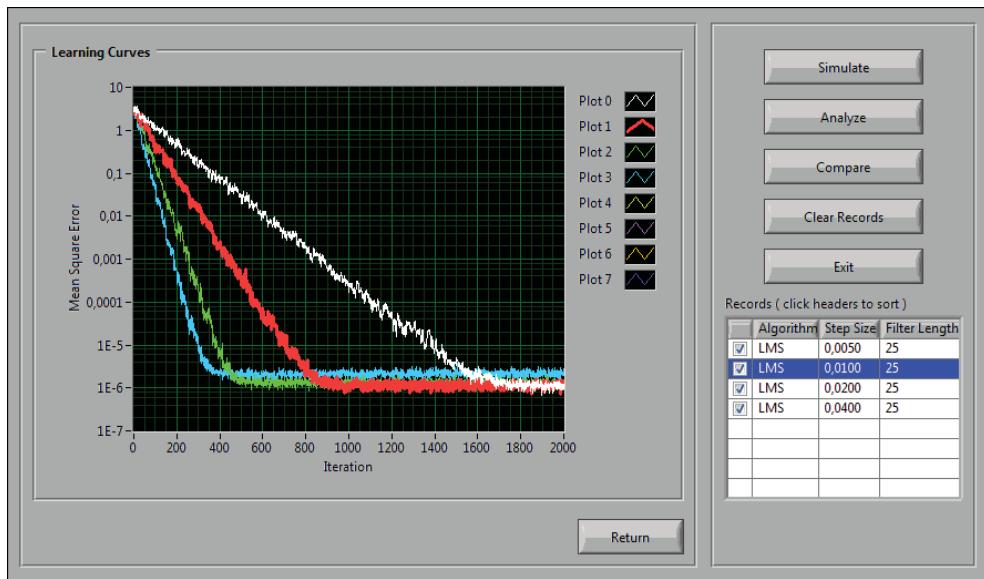


Рис. 15.16. LMS Adaptive Filter Lab Demo: обучающая кривая

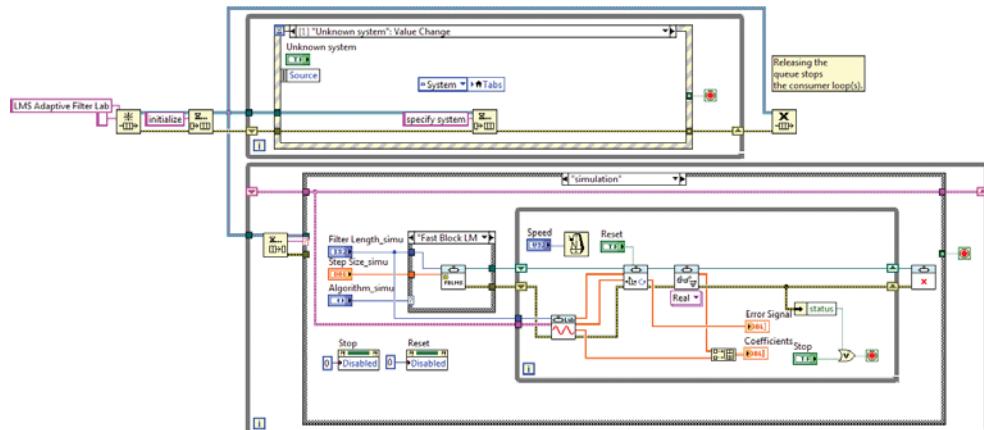


Рис. 15.17. LMS Adaptive Filter Lab Demo: блок-диаграмма

15.5. Нестандартные программные средства для моделирования адаптивных фильтров в среде разработки LabVIEW

К широкому спектру достоинств среды разработки LabVIEW следует отнести и возможность для пользователя дополнять графические программы тек-

стовыми программами. Эту возможность удобно использовать при программировании математических формул или при использовании отлаженных программ на других языках. Поэтому в LabVIEW реализованы механизмы добавления в блок-диаграмму различного текстового кода, включая код языка MATLAB.

В настоящем разделе приведены краткие сведения о нестандартных программных реализациях адаптивных фильтров [202–206], полученных путем совместного использования LabVIEW и MATLAB с применением элементов прикладной библиотеки адаптивной фильтрации, рассмотренной в разделе 15.3.

На рис. 15.18 показан внешний вид виртуального прибора [203] для исследования свойств адаптивного фильтра с двумя ВК, вычисляемыми с помощью алгоритмов Ньютона, наискорейшего спуска, LMS и RLS (см. главы 4–6) в задаче линейного предсказания (рис. 3.10), детали которой приведены в [20].

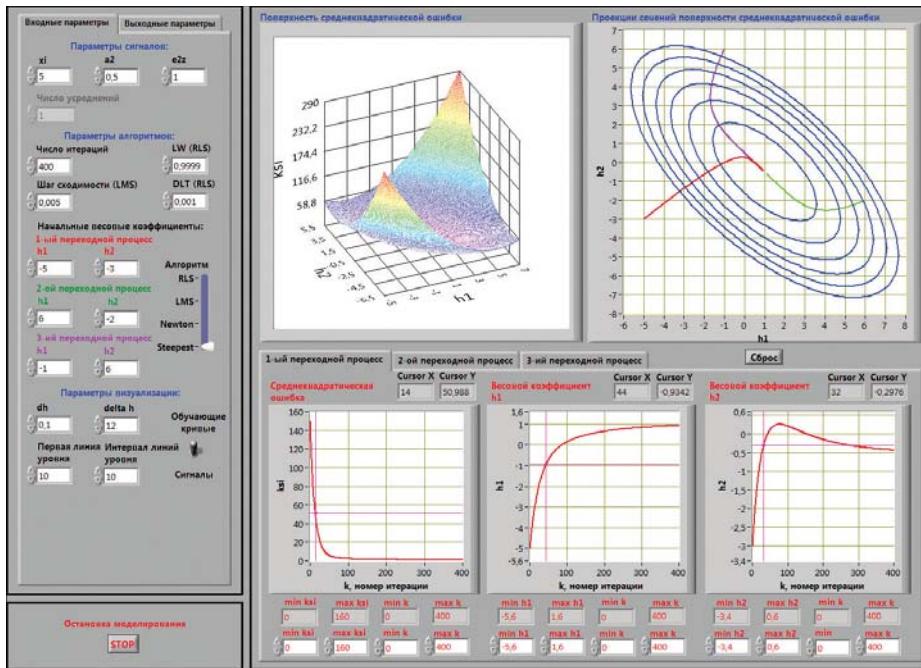


Рис. 15.18. Виртуальный прибор для исследования свойств адаптивного фильтра с двумя ВК, вычисляемыми с помощью алгоритмов Ньютона, наискорейшего спуска, LMS и RLS в задаче линейного предсказания

В качестве основного параметра для разных адаптивных алгоритмов задается требуемое отношение собственных чисел корреляционной матрицы адаптивного фильтра. В соответствии с этим параметром вырабатывается входной сигнал. Одновременно исследуется поведение трех адаптивных фильтров, вычисляющих ВК с помощью одного и того же выбранного алгоритма и обра-

батывающих одни и те же сигналы. Различие между адаптивными фильтрами заключается в значениях начальных ВК, которые можно независимо задавать.

С помощью виртуального прибора можно исследовать свойства корреляционной матрицы адаптивного фильтра, исследовать характер поверхности среднеквадратической ошибки и ее срезов (линий уровня) в зависимости от отношения собственных чисел корреляционной матрицы, а также оценить постоянные времени переходных процессов.

Число ВК в адаптивных фильтрах прибора (см. рис. 15.18) равно двум. Выбор в качестве объекта исследования адаптивного фильтра с двумя ВК обусловлен возможностью визуализации в этом случае трехмерной поверхности среднеквадратической ошибки и двумерных графиков ее линий уровня с целью исследования зависимости их формы от отношения собственных чисел корреляционной матрицы, а также визуализации траектории движения ВК в процессе работы адаптивного фильтра.

В то же время в большинстве практических приложений обычно используются адаптивные фильтры с числом ВК, отличным от двух. Для исследования адаптивных фильтров на базе LMS- и RLS-алгоритмов с произвольным числом ВК применяется виртуальный прибор (рис. 15.19) [206].

В этом приборе также вырабатывается входной сигнал адаптивного фильтра с наперед заданным отношением максимального и минимального собственных чисел корреляционной матрицы. С помощью адаптивного фильтра решается задача идентификации линейной системы (импульсного отклика КИХ-фильтра). В качестве идентифицируемого импульсного отклика могут быть использованы табличные значения регламентируемых стандартом G.168 [52] импульсных откликов эхо-путей в проводных телефонных сетях, вычисленный с помощью метода зеркальных отображений [51] акустический импульсный отклик прямоугольного помещения [215] или произвольный импульсный отклик, введенный вручную.

Численными результатами моделирования являются значения относительной избыточной среднеквадратической ошибки (см. главы 5 и 6), значения ВК адаптивного фильтра на последней итерации работы алгоритма и экспериментально оцененное отношение максимального и минимального собственных чисел корреляционной матрицы. В качестве графических результатов моделирования в виртуальном приборе (см. рис. 15.19) выводятся графики входных/выходных сигналов адаптивного фильтра, среднеквадратической ошибки и ряд других.

Еще одним примером нестандартных виртуальных приборов является прибор для исследования трехлучевых линейных эквидистантных адаптивных антенных решеток (AAP) [202, 204] (рис. 15.20). В нем реализованы различные

линейно-ограниченные алгоритмы адаптивной фильтрации (LMS, NLMS, RLS и аффинных проекций). Прибор позволяет использовать «слепые» версии указанных алгоритмов, т. е. не требующих опорного сигнала в явном виде. Детали этих алгоритмов приведены в работах [216–219].

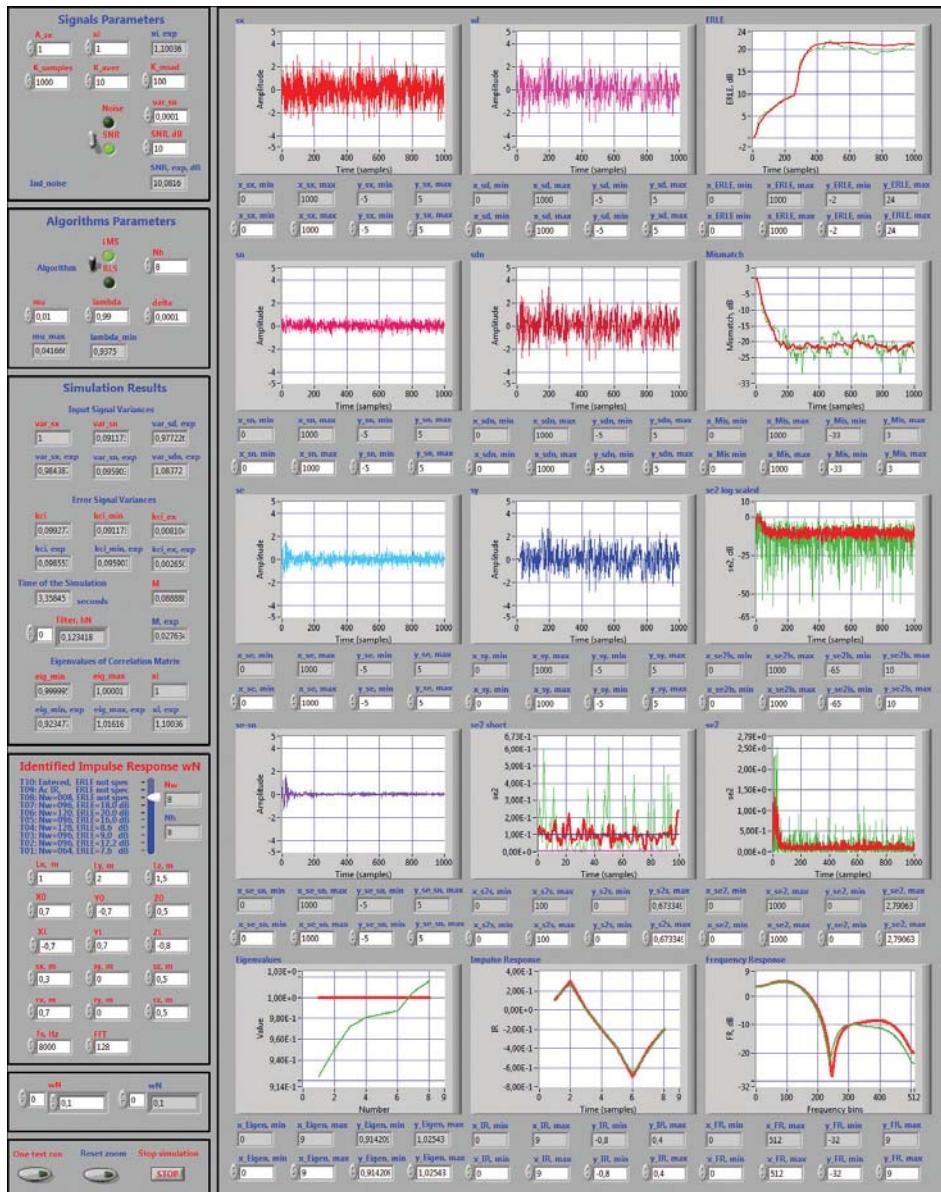


Рис. 15.19. Виртуальный прибор для исследования свойств адаптивного фильтра с произвольным числом ВК, вычисляемыми с помощью LMS- и RLS-алгоритмов, в задаче идентификации линейной системы

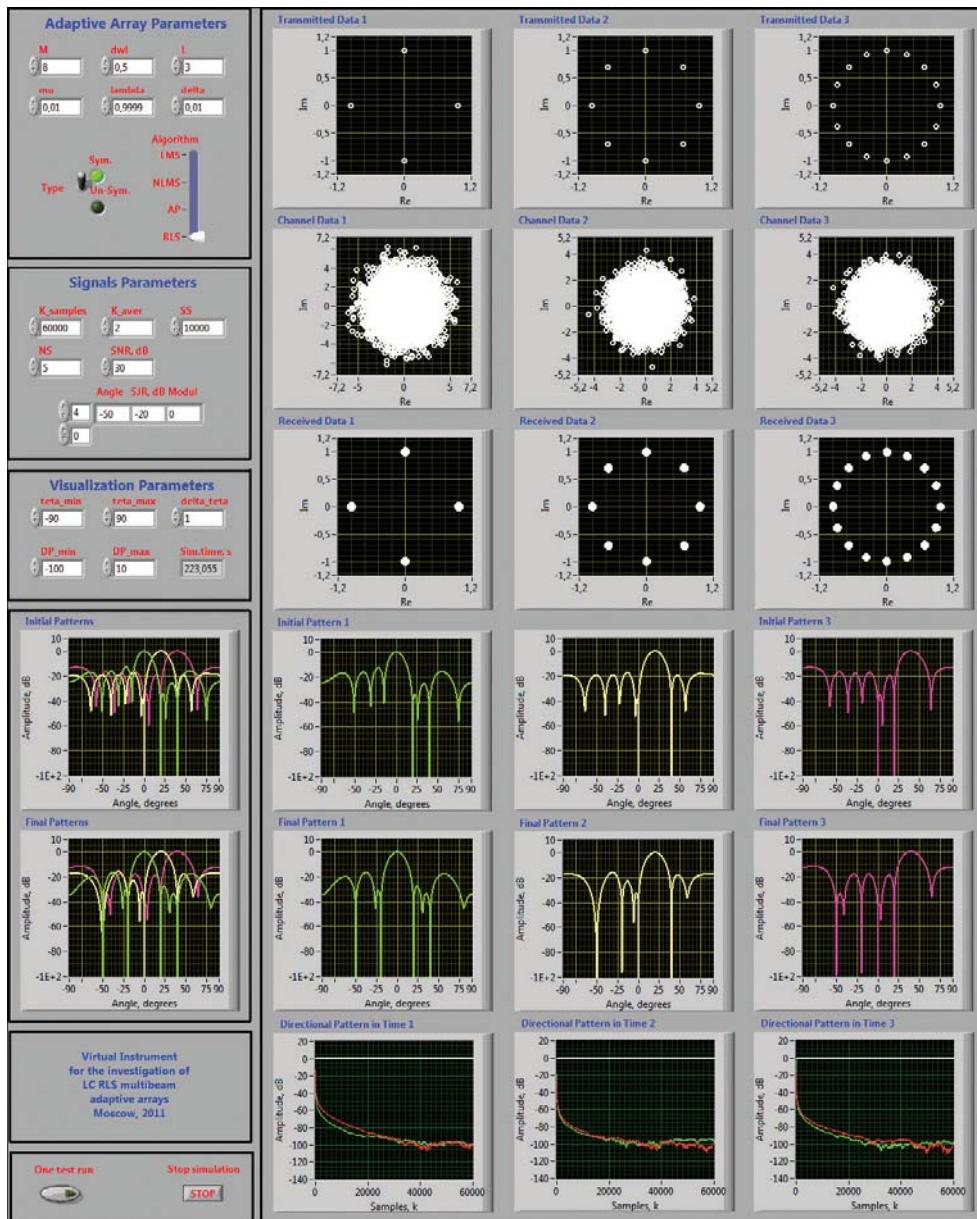


Рис. 15.20. Виртуальный прибор для исследования многолучевой ААР

В качестве результатов моделирования выводятся созвездия принятых решеткой PSK-сигналов, созвездия на выходе антенной решетки при отсутствии и при использовании адаптации, а также ДН ААР до и после адаптации как раздельно для каждого луча многолучевой ААР, так и совместно для трех лучей.

Аналогичные виртуальные приборы адаптивного эхокомпенсатора и эквализера канала связи представлены в работе [205].

Таким образом, рассмотренные виртуальные приборы демонстрируют возможности построения в среде LabVIEW различных нестандартных адаптивных приложений, в том числе с использованием отлаженных программ на языке MATLAB.

15.6. Моделирование приложений адаптивной фильтрации с помощью Adaptive Equalization Library среды разработки SystemVue

Подобно LabVIEW и графическому приложению Simulink языка MATLAB среда разработки SystemVue [212, 213] также является средой графического программирования. Она позволяет примерно вдвое сократить время проектирования на физическом уровне различных алгоритмов и систем. SystemVue представляет собой простую в использовании среду разработки с передовыми технологиями моделирования, с возможностью подключения к аппаратной реализации и с возможностью проведения испытаний.

Основные особенности среды SystemVue включают:

- расширенное моделирование потока данных, т.е. обработку сигналов с различными скоростями, частотами и с учетом реальных эффектов;
- обширный набор точных аналоговых моделей;
- сотни пополняемых библиотек, которые экономят время при работе с функциями обработки сигналов, радиочастотными блоками, функциями с фиксированной точкой и блоками различных стандартов связи;
- поддержку моделей на основе X-параметров для совместного моделирования аналоговой и цифровой частей устройства;
- возможность легкого переключения между блоками, написанными на языке C++, реализованными в т-коде или Verilog/VHDL;
- поддержку языков программирования C, MATLAB;
- автоматическую генерацию VHDL/Verilog, позволяющую быстро изготавливать опытные образцы на основе ПЛИС.

Среди многообразия систем, которые можно исследовать и проектировать с помощью SystemVue важное место занимают различные системы связи, неотъемлемым элементом оборудования которых является эквалайзер. Поэтому библиотека Algorithm Design Library [220] среды SystemVue содержит приложение под названием Adaptive Equalizers с набором алгоритмов адаптивной фильтрации и их составных частей.

И хотя эти алгоритмы разработаны в первую очередь для построения и моделирования эквалайзеров каналов связи, их также можно использовать и в дру-

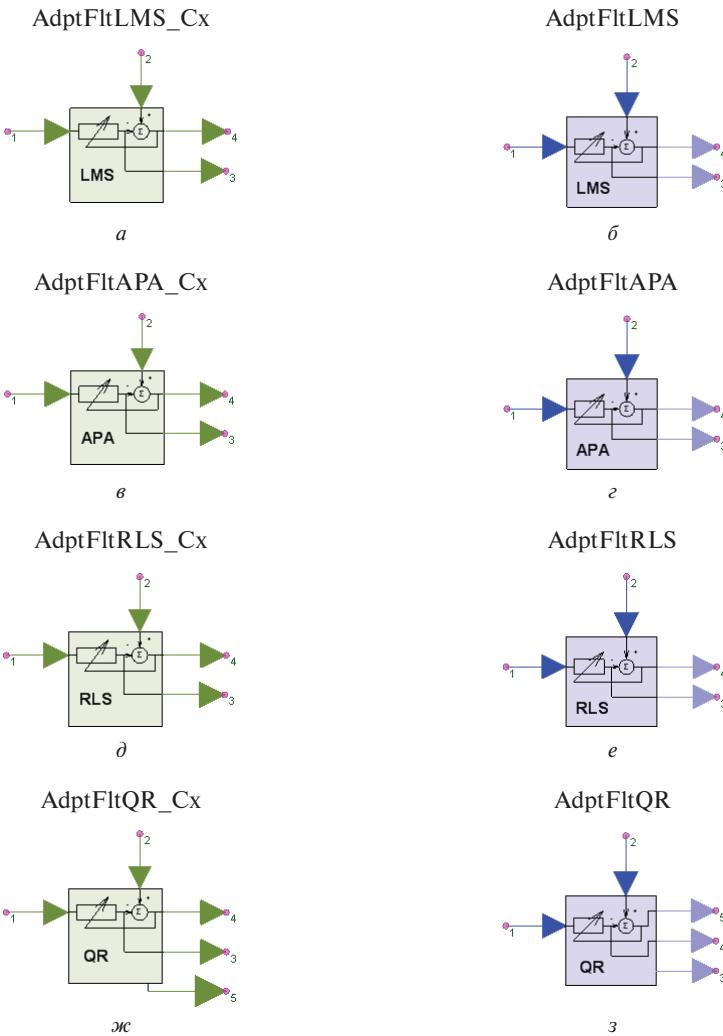


Рис. 15.21. Адаптивные фильтры приложения Adaptive Equalizers: а) LMS-алгоритм с комплексными ВК; б) LMS-алгоритм с действительными ВК; в) алгоритм аффинных проекций с комплексными ВК; г) алгоритм аффинных проекций с действительными ВК; д) RLS-алгоритм с комплексными ВК; е) RLS-алгоритм с действительными ВК; ж) RLS-алгоритм на базе QR-разложения с комплексными ВК; з) RLS-алгоритм на базе QR-разложения с действительными ВК

гих приложениях. В библиотеку Adaptive Equalizers входят LMS-алгоритм, алгоритм аффинных проекций, RLS-алгоритм на базе MIL и RLS-алгоритм на базе QR-разложения. Алгоритмы существуют в двух разных формах: с действительной и с комплексной арифметикой (рис. 15.21) в формате с плавающей точкой. Путем изменения параметров LMS-алгоритм может быть переключен в NLMS-

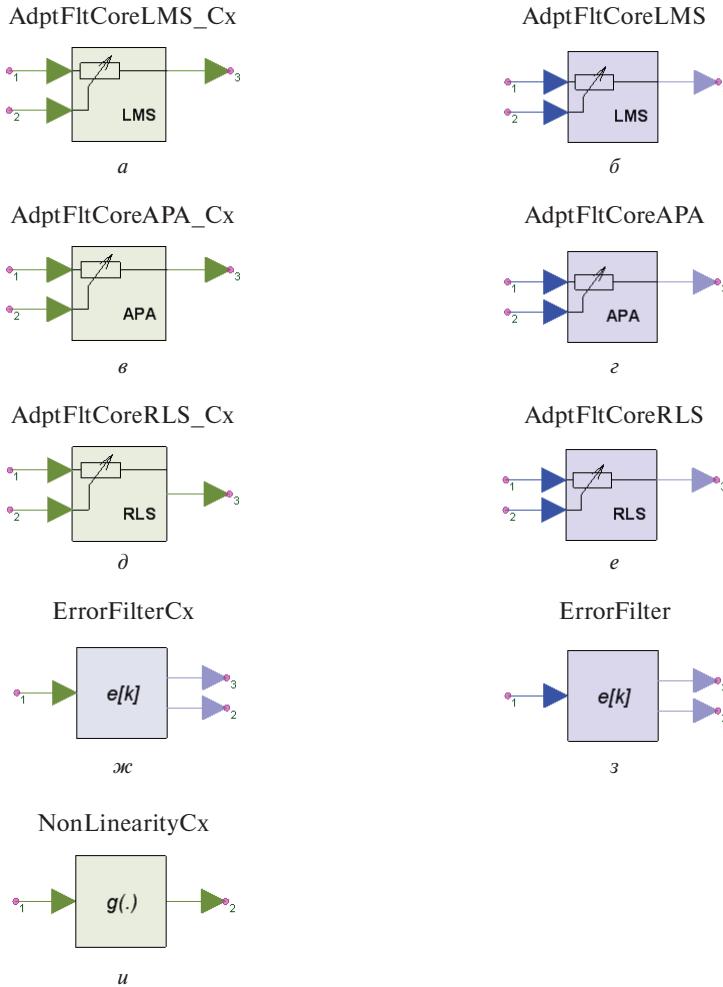


Рис. 15.22. Фильтры с переменными ВК приложения Adaptive Equalizers и другие вспомогательные блоки: а) LMS-алгоритм с комплексными ВК; б) LMS-алгоритм с действительными ВК; в) алгоритм аффинных проекций с комплексными ВК; г) алгоритм аффинных проекций с действительными ВК; д) RLS-алгоритм с комплексными ВК; е) RLS-алгоритм с действительными ВК; ж) измерение комплексной ошибки; з) измерение действительной ошибки; и) нелинейная комплексная функция

алгоритм. Отличительной особенностью элементов приложения Adaptive Equalizers является то, что они поддерживают структуры адаптивных КИХ- и БИХ-фильтров. Детальное описание блоков Adaptive Equalizers приведено в [220].

Блоки составных «частей» алгоритмов LMS, аффинных проекций и RLS, показаны на рис. 15.22. Блоки на рис. 15.22а—е представляют собой фильтры с переменными ВК, блоки на рис. 15.22ж и рис. 15.22з предназначены для изме-

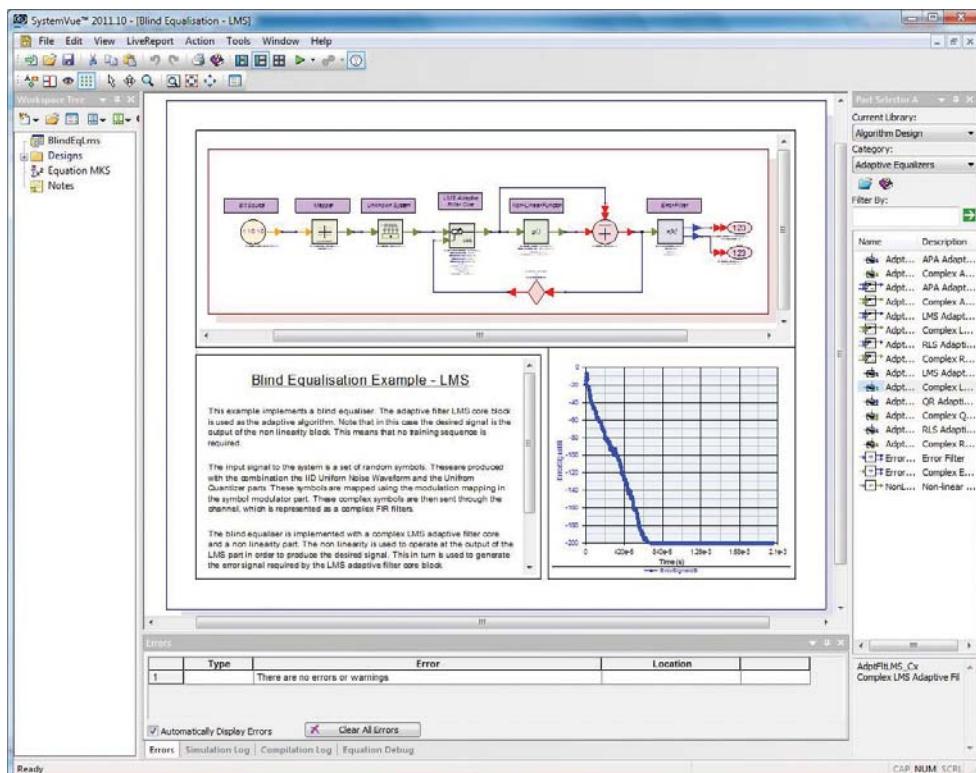


Рис. 15.23. «Слепой» эквалайзер, использующий адаптивный фильтр на базе LMS-алгоритма (пример BlindEqLms.wsv)

рения «мгновенной мощности» сигнала путем вычисления квадрата его амплитуды и сглаживания результатов, а блок на рис. 15.22и, реализует нелинейную функцию, используемую при построении «слепых» алгоритмов адаптивной фильтрации [45].

Различные примеры решения прикладных задач с помощью элементов Algorithm Design Library (адаптивных фильтров на базе алгоритма аффинных проекций, LMS-, MIL RLS-, QR RLS-алгоритмов с комплексными и действительными ВК) приведены в «Справочной системе» языка SystemVue, а результаты решения некоторых из этих задач приведены ниже.

На рис. 15.23 показаны элементы основного окна языка SystemVue при решении задачи выравнивания канала связи с помощью «слепого» эквалайзера, использующего адаптивный фильтр на базе LMS-алгоритма. Здесь эквалайзер включен в систему связи, состоящую из источника информации, модулятора, канала связи, собственно адаптивного эквалайзера и фильтра ошибок.

На рис. 15.24 показан эквалайзер с обратной связью (Decision Feedback Equalizer), использующий адаптивный фильтр на базе LMS-алгоритма.

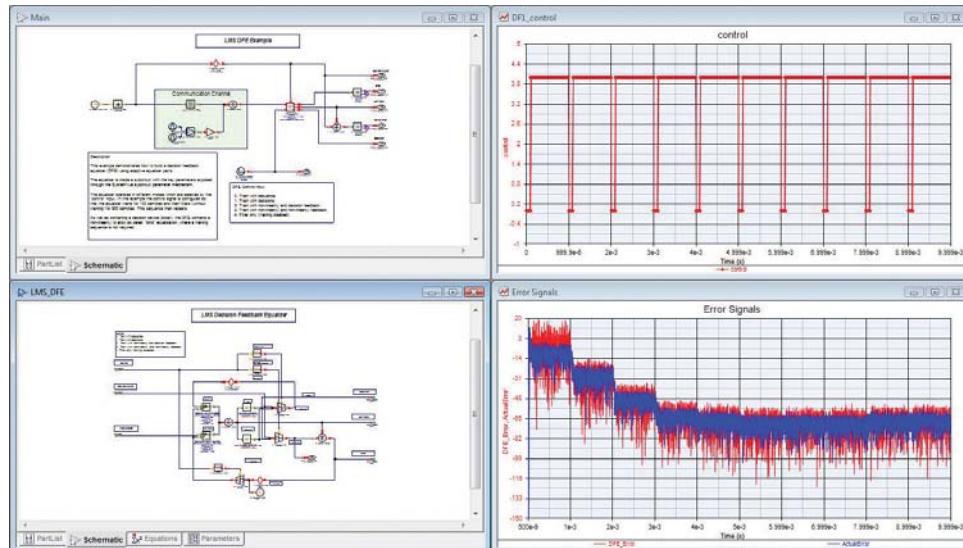


Рис. 15.24. Эквалайзер с обратной связью (Decision Feedback Equalizer), использующий адаптивный фильтр на базе LMS-алгоритма (пример LMSDFDE.wsv)

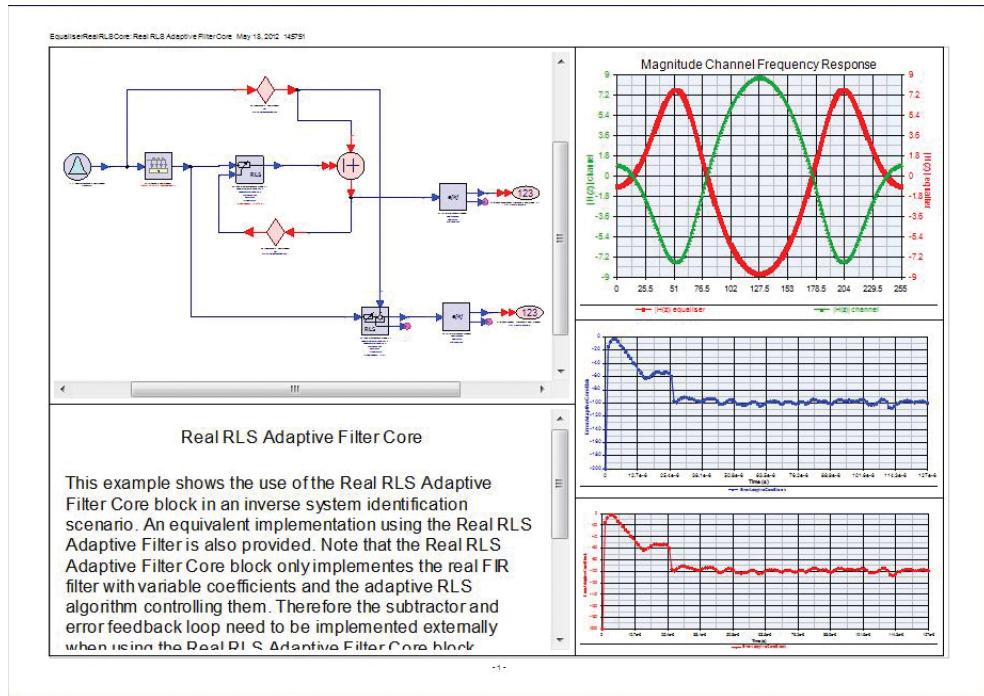


Рис. 15.25. Обратная идентификация с помощью адаптивного фильтра на базе RLS-алгоритма с действительными ВК (пример EqualiserRealRLS.wsv)

На рис. 15.25 показан пример решения задачи обратной идентификации с помощью адаптивного фильтра на базе MIL RLS-алгоритма, а на рис. 15.26 — пример решения задачи прямой идентификации с помощью адаптивного фильтра на базе алгоритма аффинных проекций.

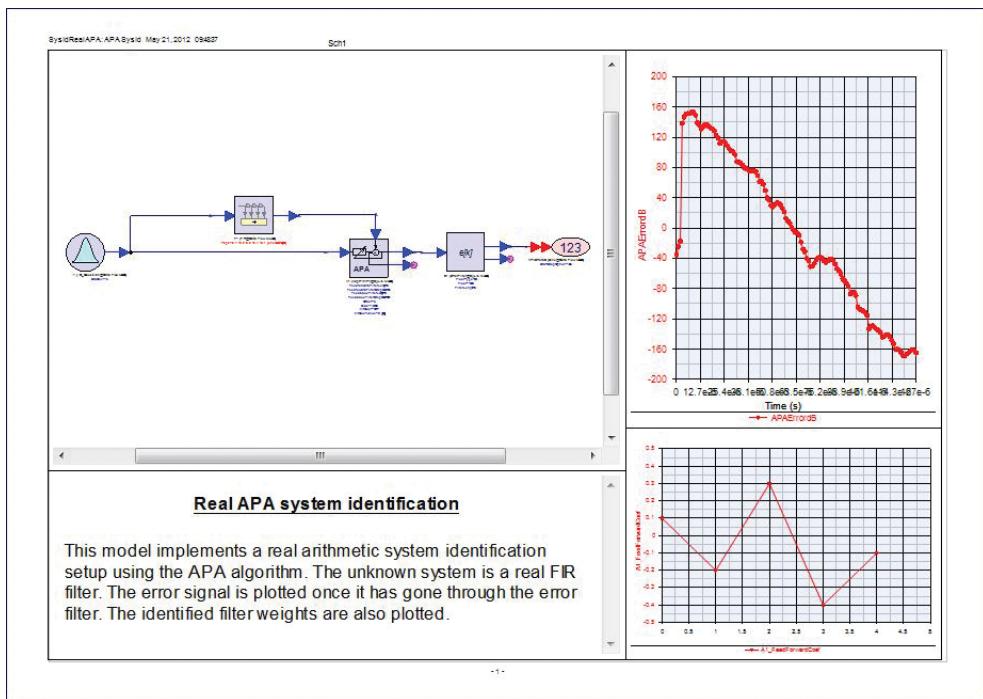


Рис. 15.26. Прямая идентификация с помощью адаптивного фильтра на базе алгоритма аффинных проекций с действительными ВК (пример SysIdRealAPA.wsv)

Рис. 15.23—15.26 дают представление о том, как выглядят программы в среде SystemVue при решении прикладных задач, которые не ограничиваются только адаптивными эквалайзерами.

15.7. Библиотека адаптивной фильтрации для ЦСП отечественной серии «Мультикор»

Рассмотренные выше средства моделирования позволяют на этапе проектирования выполнять исследования адаптивных фильтров, а также устройств и систем на их основе. Если результаты моделирования являются удовлетворительными, то далее требуется адаптивный фильтр реализовать. Реализация может осуществляться двумя способами: аппаратно, когда требуются высокие скоро-

сти обработки сигналов, и программно, когда требования к скорости обработки сигналов сравнительно невысоки.

Детали аппаратной реализации адаптивных фильтров в технической литературе отсутствуют. Что же касается программной реализации, то большинство производителей сегодня наполняют прикладные библиотеки своих ЦСП в том числе и алгоритмами адаптивной фильтрации.

Ниже приводится описание библиотеки адаптивной фильтрации (рис. 15.27), которая является частью прикладной библиотеки программ для ЦСП отечественной серии «Мультикор» [43, 44], разработанных ОАО НПЦ «ЭЛ-ВИС» [193].

ЦСП серии «Мультикор» — это однокристальные программируемые многочиповые «системы на кристалле» на базе сложных функциональных блоков платформы «МУЛЬТИКОР». Они сочетают в себе качества двух классов приборов: микроконтроллеров и ЦСП, что позволяет решать в рамках ограниченных габаритов одновременно обе задачи: управления и высокоточной обработки сигналов. Основные разновидности выпускаемых на сегодняшний день ЦСП и их основные параметры приведены в табл. 15.3.

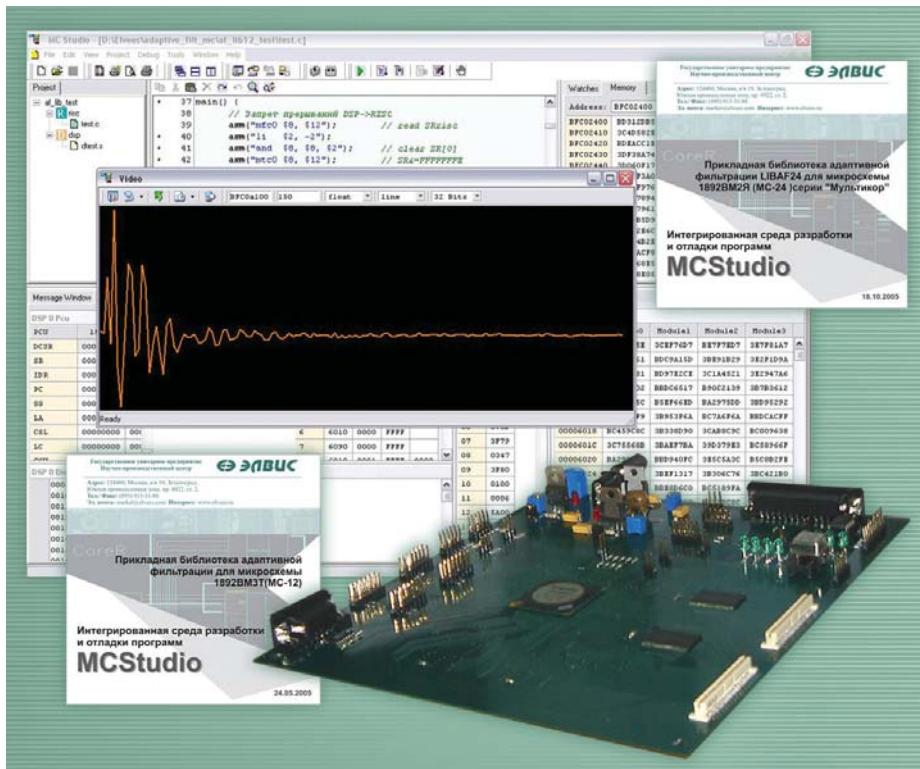


Рис. 15.27. Прикладные библиотеки алгоритмов адаптивной фильтрации для СБИС ЦСП серии «Мультикор»

Таблица 15.3. Информация о микросхемах серии «Мультикор»

Микросхема	1892ВМ3Т	1892ВМ2Я	1892ВМ5Я	NVCom-01
Технология изгото- ления, мкм	0.25	0.25	0.25	0.13
Микропроцессорная архитектура	2 про- цессора: RISCore32 + ELcore-14	2 про- цессора: RISCore32 + ELcore-24	3 процессора: RISCore32 + 2 xELcore-26	3 процессора: RISCore32 + 2 xDELCore-30
Рабочая тактовая частота, МГц	80	80	100	300
Пиковая производи- тельность, MFLOPs, 32-битных операций с плавающей точкой (IEEE 754)	240	480	1200	3600

В качестве процессорных блоков в СБИС серии «Мультикор» используются процессорные RISC-ядра с архитектурой MIPS32 (RISCore32), выполняющие функции центрального процессора системы, и высокопроизводительные ядра процессоров-акселераторов (ELcore-14, ELcore-24, xELcore-26, xDELCore-30) для ЦОС с плавающей и фиксированной точкой.

Оценки вычислительной сложности реализации некоторых алгоритмов адаптивной фильтрации на языке Assembler ЦСП серии «Мультикор» (1892ВМ3Т и 1892ВМ2Я) приведены на рис. 15.28. Для разработки программного кода использовалась среда MCStudio™ СБИС серии «Мультикор» [221, 222].

На рис. 15.28а кривые 1, 2 — это комплексный RLS-алгоритм со скользящим окном (одноканальный и многоканальный), СБИС 1892ВМ3Т; кривые 3, 4 — действительный RLS-алгоритм со скользящим окном (одноканальный и многоканальный), СБИС 1892ВМ3Т; кривые 5, 6 — комплексный RLS-алгоритм со скользящим окном (одноканальный и многоканальный), СБИС 1892ВМ2Я; кривые 7, 8 — действительный RLS-алгоритм со скользящим окном (одноканальный и многоканальный), СБИС 1892ВМ2Я; кривые 9, 10 — комплексный RLS-алгоритм с возрастающим окном (одноканальный и многоканальный), СБИС 1892ВМ3Т; кривые 11, 12 — действительный RLS-алгоритм с возрастающим окном (одноканальный и многоканальный), СБИС 1892ВМ3Т; кривые 13, 14 — комплексный RLS-алгоритм с возрастающим окном (одноканальный и многоканальный), СБИС 1892ВМ2Я; кривые 15, 16 — действительный RLS-алгоритм со скользящим окном (одноканальный и многоканальный), СБИС 1892ВМ2Я. Обозначение «действительный алгоритм» означает алгоритм для адаптивного фильтра с действительными ВК, а «комплексный алгоритм» — алгоритм для адаптивного фильтра с комплексными ВК. Из этого рисунка следует, что вычислительная сложность алгоритмов со скользящим окном примерно в 2

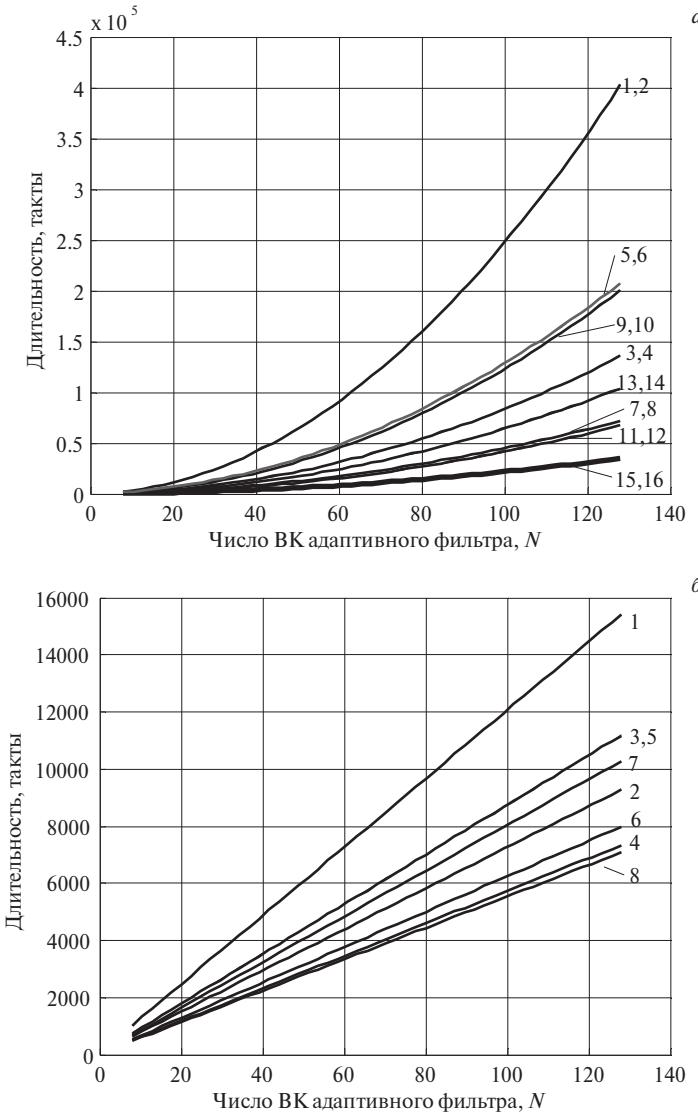


Рис. 15.28. Вычислительная сложность адаптивных алгоритмов: а) MIL RLS;
б) лестничных

раза больше сложности алгоритмов с возрастающим окном, что согласуется с материалами, рассмотренными в главе 14.

На рис. 15.28б приведены оценки вычислительной сложности по результатам реализации [199] некоторых лестничных алгоритмов на основе СБИС 1892ВМ3Т. Здесь линия 1 — это алгоритм на основе априорных и апостериорных ошибок (комплексный, без обратных связей), линия 2 — алгоритм на основе априорных и апостериорных ошибок (действительный, без обратных связей), линия 3 — алгоритм на основе априорных и апостериорных ошибок (комплекс-

ный, с обратными связями), линия 4 — алгоритм на основе априорных и апостериорных ошибок (действительный, с обратными связями), линия 5 — алгоритм на основе априорных ошибок (комплексный, без обратных связей), линия 6 — алгоритм на основе априорных ошибок (действительный, без обратных связей), линия 7 — алгоритм на основе априорных ошибок (комплексный, с обратными связями), линия 8 — алгоритм на основе априорных ошибок (действительный, с обратными связями). Из этого рисунка следует, что вычислительная сложность лестничных алгоритмов является линейной, что согласуется с материалами, рассмотренными в главе 12.

В прикладную библиотеку также входит ряд простых градиентных алгоритмов (разновидности LMS- и NLMS-алгоритмов). Все элементы библиотеки реализованы в формате вычислений с плавающей точкой с одинарной точностью. Функции для СБИС 1892ВМ3Т реализованы в режиме Single Instruction Single Data (SISD, скалярный режим), а для СБИС 1892ВМ2Я реализованы в режиме 2SISD. Полный перечень функций этой библиотеки приведен в [193] в разделе «Программное обеспечение. Прикладные библиотеки».

В [223] представлены результаты разработки оптимального кода функции адаптивного фильтра-декиматора с применением цифровой гребенчатой фильтрации для СБИС 1892ВМ2Я. Стандартная функция реализации LMS-алгоритма для этого ЦСП требует $1.5N_h + 20$ тактов на один отсчет входных-выходных сигналов, где N_h — число ВК адаптивного фильтра. Применение цифровой гребенчатой фильтрации прореживает импульсную характеристику адаптивного фильтра нулевыми отсчетами в R раз, дополняя алгоритм адаптивной фильтрации сглаживающим фильтром с числом ВК J . При такой реализации адаптивного фильтра общие вычислительные затраты сокращаются до $1.5\frac{N_h}{R} + J + 28$ тактов процессора.

Используя технические характеристики ЦСП серии «Мультикор» (см. табл. 15.3), с помощью представленных графиков можно определить частоту дискретизации, на которой адаптивные фильтры могут обрабатывать сигналы.

15.8. Приложения адаптивной фильтрации для цифровых сигнальных процессоров компаний Texas Instruments, Analog Devices и Freescale Semiconductor

Небольшое число адаптивных фильтров также реализовано и в прикладных библиотеках ЦСП зарубежных производителей. Параметры этих фильтров рассматриваются ниже.

Одним из лидирующих зарубежных производителей на рынке современных ЦСП сегодня является компания Texas Instruments. Среди ЦСП этой компании можно выделить две наиболее актуальные на сегодняшний день серии. Это серия TMS320C5000™ Ultra Low-Power DSPs, представляющая собой малопотребляющие 16-разрядные ЦСП с фиксированной точкой и с тактовой частотой до 300 МГц, а также серия TMS320C6000™ High Performance DSPs, представляющая собой высокопроизводительные ЦСП с плавающей и фиксированной точкой и тактовой частотой до 1.2 ГГц, на которой строится широкий ряд гибридных процессоров и систем на кристалле, а также многоядерных ЦСП. В прикладных библиотеках [224–226] для указанных процессоров реализованы лишь несколько разновидностей LMS-алгоритмов.

Так, в прикладной библиотеке TMS320C55x DSP Library [224] реализованы два типа LMS-алгоритмов в формате с фиксированной точкой (16 бит). Это Delayed LMS-алгоритм, размер кода которого составляет $V = 122$ байта. Для выполнения этого кода требуется $C = N_x(5 + 2N_h)$ тактов ЦСП на одну итерацию алгоритма. Здесь N_x — число отсчетов входных/выходных данных. Другой алгоритм — это быстрый Delayed LMS-алгоритм, размер кода которого составляет $V = 322$ байта. Для выполнения этого кода требуется $C = \frac{N_x}{2}(26 + 3N_h)$ тактов ЦСП на одну итерацию алгоритма.

В прикладной библиотеке TMS320C64x DSP Library [225] реализован классический LMS-алгоритм в формате с фиксированной точкой (16 бит), размер кода которого составляет $V = 148$ байт. Для выполнения этого кода требуется $C = \frac{3N_h}{4} + 16$ тактов ЦСП на одну итерацию алгоритма.

В прикладной библиотеке TMS320C67x DSP Library [226] реализованы две разновидности классического LMS-алгоритма в формате с плавающей точкой с одинарной и двойной точностью. Алгоритм с одинарной точностью с размером кода $V = 1367$ байт требует $C = (N_h + 35)N_x + 21$ тактов на одну итерацию. Алгоритм с одинарной точностью, размер кода которого составляет $V = 640$ байт, требует $C = (4N_h + 47)N_x + 27$ тактов ЦСП на одну итерацию.

Другой известный зарубежный производитель ЦСП — компания Analog Devices [191]. Она выпускает ЦСП серий ADSP-21xx, SHARC, TigerSHARC, Blackfin и SigmaDSP. Серия ADSP-21xx представляет собой ЦСП с тактовой частотой до 160 МГц. Процессоры этого семейства на данный момент можно считать устаревшими, однако они до сих пор находят широкое применение в ряде приложений. Серия SHARC — это семейство ЦСП с плавающей точкой, ориентированных на те сферы применения, в которых плавающая точка является необходимой, включая высококачественные системы обработки звука, медицинское диагностическое оборудование и ряд других. Серия TigerSHARC — наиболее высокопроизводительные ЦСП с фиксированной и плавающей

точкой, со специальной поддержкой на уровне архитектуры встраивания в многопроцессорные системы обработки сигналов. Семейство Blackfin является на сегодняшний день наиболее популярным и перспективным семейством ЦСП фирмы Analog Devices. Это 16/32-разрядные ЦСП с фиксированной точкой, изначально ориентированные на применение в портативных, энергоэффективных и высокопроизводительных устройствах массового потребления с поддержкой обработки мультимедийных данных. Серия SigmaDSP является узкоспециализированной, ориентированной на простую и быструю разработку систем обработки аудио.

Прикладная библиотека процессоров ADSP-21xx [227] содержит ряд функций адаптивных фильтров на базе алгоритмов, представленных в табл. 15.4. Алгоритмы выполняются в 16-разрядной арифметике с фиксированной точкой.

Таблица 15.4. Адаптивные фильтры прикладной библиотеки ADSP-21xx

Алгоритм	Число тактов на итерацию	Использование памяти		
		Объем програм- мы (слов в памя- ти программ)	Объем данных (слов в памя- ти программ)	Объем данных (слов в памя- ти данных)
LMS	$3N+8$	29	N	N
Leaky LMS	$3N+8$	30	N	N
NLMS	$3N+16$	40	N	N
Sing-Error LMS	$3N+8$	29	N	N
Sing-Data LMS	$4N+8$	32	N	N
Sing-Sing LMS	$4N+8$	31	N	N
RLS	$3N^2+9N$	89	$2N$	N^2+2N

Из табл. 15.4 следует, что знаковые алгоритмы, подразумевающие упрощения аппаратной реализации адаптивных фильтров, при программной реализации в последних двух случаях требуют больше тактов, чем традиционные адаптивные алгоритмы, что обусловлено дополнительными операциями определения знаков переменных.

В работе [228] был рассмотрен вариант использования нестандартного фильтра-декиматора в составе функций ЦОС ЦСП TigerSHARC при реализации многоскоростного узкополосного адаптивного фильтра на базе LMS-алгоритма. В этом фильтре удалось понизить вычислительную сложность стандартной реализации фильтра-декиматора из прикладной библиотеки ЦСП TigerSHARC [229] и понизить общие вычислительные затраты на реализацию узкополосного адаптивного фильтра с предварительной декимацией, которые выполняются за $C = 9 + \frac{N_x}{v} \left(\frac{N_h}{2} + 27 \right) + \frac{N_x}{v} \left(\frac{N_h}{2} + 18 \right)$ тактов на одну итерацию ра-



боты этого фильтра, где v — коэффициент децимации. Этот результат свидетельствует о вычислительной эффективности предложенной в работе [228] реализации фильтра-дециматора.

Зарубежная компания Freescale Semiconductor также является производителем ряда современных ЦСП. Среди ЦСП этой компании можно выделить серию DSP56300. Эта серия представляет собой 24-разрядные ЦСП с фиксированной точкой и тактовой частотой до 275 МГц. В прикладной библиотеке [230] для указанных процессоров реализованы две разновидности адаптивных фильтров: на базе классического LMS-алгоритма и Delayed LMS-алгоритма. В классическом LMS-алгоритме размер кода составляет $V=15$ слов (24-разрядных), для выполнения которого требуется $C=3N_h + 16$ тактов ЦСП на одну итерацию алгоритма. В Delayed LMS-алгоритме размер кода составляет $V=13$ слов (24-разрядных), для выполнения которого требуется $C=3N_h + 12$ тактов ЦСП на одну итерацию алгоритма.

Адаптивные алгоритмы в прикладных библиотеках других ЦСП компаний Texas Instruments, Analog Devices и Freescale Semiconductor отсутствуют.

Используя технические характеристики отмеченных ЦСП, можно определить частоту дискретизации, на которой рассмотренные адаптивные фильтры могут обрабатывать сигналы, а также оценить запас ресурсов для решения других прикладных задач.

15.9. Выводы по главе

В настоящей главе рассмотрены основные элементы стандартных и нестандартных приложений адаптивной фильтрации на языках программирования MATLAB, LabVIEW и SystemVue, элементы прикладных библиотек адаптивной фильтрации современных ЦСП отечественной компании ОАО НПЦ «ЭЛВИС», а также зарубежных компаний Texas Instruments, Analog Devices и Freescale Semiconductor. Эта глава может служить путеводителем при выборе средства моделирования и разработки адаптивных фильтров, а также их приложений. При выборе вычислительной платформы для реализации адаптивного фильтра можно воспользоваться материалами, приведенными в [231]. Здесь рассматриваются архитектурные особенности и сравнительный анализ ЦСП отечественного и зарубежного производства.

Пользуясь приведенными в настоящей книге теоретическими сведениями и подробными вычислительными процедурами адаптивных фильтров, многие ее читатели, вероятно, смогут самостоятельно создать свои библиотеки алгоритмов адаптивной фильтрации на интересующих их языках программирования и вычислительных платформах.

Заключение

В настоящей книге дано общее представление об адаптивных фильтрах, определены требования к ним, а также рассмотрены примеры применения таких фильтров.

Теоретические вопросы, связанные с основами адаптивной фильтрации сигналов, сосредоточены на описании корреляционной матрицы, на анализе свойств ее собственных чисел и собственных векторов, а также на анализе среднеквадратической целевой функции и ее градиента. В рамках этих вопросов также рассмотрена винеровская фильтрация сигналов.

В книге рассмотрены такие алгоритмы адаптивной фильтрации, как алгоритмы Ньютона и наискорейшего спуска, имеющие в основном теоретическое значение, так как в их уравнениях используются статистические характеристики обрабатываемых сигналов, которые на практике обычно не всегда известны. Однако эти алгоритмы могут быть использованы в качестве эталонов для сравнения с другими алгоритмами адаптивной фильтрации, в основе которых лежит оценка статистических характеристик обрабатываемых сигналов в процессе функционирования адаптивного фильтра.

В качестве алгоритмов, нашедших широкое применение на практике, в книге рассмотрены LMS-алгоритм, его нормализованная версия — NLMS-алгоритм и RLS-алгоритм. Градиентные LMS- и NLMS-алгоритмы сегодня по-прежнему являются популярными среди инженеров в силу их низкой вычислительной сложности (наименьшей среди всех алгоритмов адаптивной фильтрации) и простоты описывающих их математических выражений. Существуют также версии таких алгоритмов, в которых с целью их дальнейшего упрощения в формулах вычисления ВК вместо значений ошибки или элементов вектора входных сигналов используются их знаки. Эти алгоритмы ориентированы преимущественно на аппаратную реализацию, так как указанная модификация позволяет уменьшить вычислительную сложность и упростить архитектуру адаптивного фильтра. Поскольку в этих алгоритмах используется грубая оценка градиента целевой функции, то такие алгоритмы характеризуются более длинным переходным процессом по сравнению с немодифицированными алгоритмами. Упрощенные алгоритмы в книге не рассматривались.

Популярными сегодня являются и градиентные алгоритмы с субполосной фильтрацией обрабатываемых сигналов [32]. Существуют две основные разновидности таких алгоритмов: на базе процедур БПФ и на базе гребенок фильтров. Центральной идеей субполосной адаптивной фильтрации является то, что при ее использовании уменьшается разброс собственных чисел корреляционной матрицы в каждой из разделенных частотных полос, что позволяет для обработки сигналов в этих полосах использовать отдельные адаптивные фильтры



на базе градиентных алгоритмов с разными шагами сходимости. Поэтому при обработке сигналов с большим разбросом собственных чисел корреляционной матрицы, субполосные адаптивные фильтры демонстрируют большую скорость сходимости по сравнению с адаптивными фильтрами на базе LMS- и NLMS-алгоритмов. Субполосные адаптивные фильтры также характеризуются низкой вычислительной сложностью, что достигается за счет блочной обработки сигналов и использования БПФ, а также более коротких (с меньшим числом ВК) адаптивных фильтров в полосах. Недостатком субполосных адаптивных фильтров является задержка выходных сигналов, обусловленная блочной обработкой, а также высокие требования к фильтрам, осуществляющим разделение обрабатываемых сигналов на полосы. Субполостные адаптивные фильтры в книге не рассматривались.

Существует большое число RLS-алгоритмов адаптивной фильтрации на базе QR-разложения матрицы входных сигналов адаптивного фильтра [33]. Эти алгоритмы характеризуются высокой устойчивостью. Однако их недостатком является наличие операций деления и извлечения квадратного корня, число которых пропорционально числу ВК адаптивного фильтра. Эти операции в QR RLS-алгоритмах можно исключить путем масштабирования ряда переменных. QR RLS-алгоритмы, подобно RLS-алгоритму на базе MIL, характеризуются квадратичной вычислительной сложностью.

Эта сложность уменьшается до линейной в быстрых RLS-алгоритмах. Такие алгоритмы строятся на базе леммы об обращении клеточных матриц, в результате использования которой удается получить процедуру вычисления векторов коэффициентов Калмана, не требующую операции обращения корреляционной матрицы, что и приводит к линейной вычислительной сложности адаптивных алгоритмов. Вычислительная сложность быстрых RLS-алгоритмов в разы превышает сложность простых градиентных алгоритмов, но при этом гораздо меньше сложности RLS-алгоритмов на базе MIL и QR-разложения. Это позволяет использовать быстрые RLS-алгоритмы в адаптивных фильтрах с большим числом ВК. К быстрым RLS-алгоритмам также относятся лестничные адаптивные алгоритмы, получившие свое название из-за структуры вычислений переменных, напоминающей лестницу и решетку.

Промежуточными между LMS- и RLS-алгоритмами являются алгоритмы аффинных проекций. Они представляют собой разновидность блочного NLMS-алгоритма со скользящим прямоугольным окном, который обновляет ВК адаптивного фильтра на каждом отсчете обрабатываемых сигналов. При обработке нестационарных сигналов алгоритм аффинных проекций демонстрирует эффективность (в терминах длительности переходного процесса), близкую к RLS-алгоритмам. Быстрая версия алгоритма аффинных проекций (FAP) при использовании в адаптивных фильтрах с большим числом ВК характеризуется линейной вычислительной сложностью, близкой к сложности NLMS-

алгоритма. Это обусловило большую популярность FAP-алгоритма в задачах эхокомпенсации, при решении которых требуются адаптивные фильтры с большим числом ВК (сотни, тысячи).

Известно, что если вместо КИХ-фильтров с постоянными ВК использовать БИХ-фильтры, то число ВК можно существенно уменьшить, что заметно снижает требования к вычислительным ресурсам при реализации фильтров. Это в равной мере относится и к адаптивным БИХ-фильтрам [16]. Однако сегодня адаптивные БИХ-фильтры не нашли ни широкого практического использования, ни большого теоретического интереса, что обусловлено не столько их потенциальной неустойчивостью, сколько многоэкстремальным характером целевой функции, который не гарантирует нахождение ее глобального минимума. Адаптивные БИХ-фильтры в книге не рассматривались.

В то же время, в книге освещены некоторые вопросы адаптивной фильтрации нестационарных сигналов и дано описание стандартных и нестандартных приложений для моделирования адаптивных фильтров на таких языках программирования как MATLAB, LabVIEW и SystemVue, широко применяемых сегодня при исследовании алгоритмов ЦОС. Кроме того, приведены доступные из литературных источников сведения о реализации адаптивных фильтров на современных ЦСП отечественного и зарубежного производства.

В настоящей книге не отражены вопросы, касающиеся квантования обрабатываемых сигналов и результатов вычислений при реализации адаптивных фильтров в целочисленной арифметике. Некоторые подходы к их освещению могут быть найдены в работах [20, 22, 29].

Таким образом, можно заключить, что адаптивная фильтрация сигналов является современным научно-техническое направлением в ЦОС. Ее круг вопросов гораздо шире, чем вопросы, которые удалось рассмотреть в рамках настоящей книги. Сегодня по разным оценкам в мире насчитывается более 10 тысяч публикаций, отражающих вопросы теории и практики адаптивной обработки сигналов, что подчеркивает неугасаемый интерес к этому научно-техническому направлению и его важность для научно-технического прогресса.

Несмотря на это, специализированных книг по адаптивной обработке сигналов, изданных за всю историю ее развития, сегодня в мире лишь около тридцати штук. Эти книги в основном англоязычные. Поэтому издание настоящей книги является скромным вкладом автора в рассмотренное научно-техническое направление и первым систематическим изложением теории адаптивной фильтрации на русском языке.

Список литературы

1. Монзинго Р.А., Миллер Т.У. Адаптивные антенные решетки. Введение в теорию. Пер. с англ. Челпанова В. Г., Лексаченко В. А. — М.: Радио и связь, 1986. — 448 с.
2. Коуэн К.Ф.Н., Грант П. М. Адаптивные фильтры. Пер. с англ. Лихацкой Н. Н., Ряковского С. М. — М.: Мир, 1988. — 392 с.
3. Уидроу Б., Стирнз С.Д. Адаптивная обработка сигналов. Пер. с англ. под ред. Шахгильдяна В. В. — М.: Радио и связь, 1989. — 440 с.
4. Журавлев А.К., Лукошкин А.П., Поддубный С.С. Обработка сигналов в адаптивных антенных решетках. — Л.: Издательство Ленинградского университета, 1983. — 240 с.
5. Пистолькорс А.А, Литвинов О. С. Введение в теорию адаптивных антенн. — М.: Наука, 1991. — 200 с.
6. Hudson J. E. Adaptive array principles. — England, Loughborough: Peter Peregrinus Ltd., 1981. — 253 p.
7. Honig M. L., Messerschmitt D. G. Adaptive filters: structures, algorithms and applications. — MA, Hingham: Kluwer Academic Publishers, 1984. — 337 p.
8. 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.
9. Alexander S. T. Adaptive signal processing. Theory and applications. — New York: Springer Verlag, 1986. — 179 p.
10. Ljung L., Soderstrom T. Theory and practice of recursive identification. — Cambridge, Massachusetts, London, England: MIT Press, 1986. — 529 p.
11. Treicher J. R., Johnson C. R., Larimore M. G. Theory and design of adaptive filters. — New York: John Wiley and Sons, Inc., 1987. — 342 p.
12. Orfanidis S. J. Optimum signal processing. — New York: McGraw-Hill, Inc., 1988. — 590 p.
13. Kalouptsidis N., Theodoridis S., Eds. Adaptive system identification and signal processing algorithms. — NJ, Englewood Cliffs: Prentice-Hall, 1993. — 560 p.
14. Clarkson P. M. Optimum and adaptive signal processing. — London: CRC Press, 1993. — 529 p.
15. Zelniker G., Taylor F. J. Advanced digital signal processing: theory and applications. — New York: Marcel Dekker, Inc., 1994. — 666 p.
16. Regalia P. A. Adaptive IIR filtering in signal processing and control. — New York: Marcel Dekker, Inc., 1995. — 678 p.
17. Macchi O. Adaptive processing. The least mean squares approach with application in transmission. — New York: John Wiley and Sons, Inc., 1995. — 456 p.
18. Farhang-Boroujeny B. Adaptive filters theory and applications. — New York: John Wiley & Sons, 1999. — 548 p.

19. Bellanger M. G., Adaptive digital filters. Second edition. — New York, Marcel Dekker, Inc., 2001. — 451 p.
20. Haykin S. Adaptive filter theory (4-th edition). — Prentice Hall, 2001. — 936 p.
21. Benvenuto N., Cherubini G. Algorithms for communication systems and their applications. — NJ, Hoboken: John Wiley and Sons, Inc., 2002. — 1285 p.
22. Sayed A. H. Fundamentals of adaptive filtering. — NJ, Hoboken: John Wiley and Sons, Inc., 2003. — 1125 p.
23. Ратынский М. В. Адаптация и сверхразрешение в антенных решетках. — М.: 2003. — Радио и связь. — 200 с.
24. Benesty J., Huang Y. (Eds). Adaptive signal processing: applications to real-world problems. — Berlin, Heidelberg, New York, Springer-Verlag, 2003. — 356 p.
25. Manolakis D. G., Ingle V. K., Kogon S. M. Statistical and adaptive signal processing. — Boston: Artech House, 2005. — 796 p.
26. Тараканов А. Н., Хрящев В. В., Приоров А. Л. Адаптивная цифровая обработка сигналов: учебное пособие. — Ярославль: ЯрГУ, 2001. — 134 с.
27. Сергиенко А. Б. Цифровая обработка сигналов: учебн. пособие. 3-е изд. — С.Пб.: БХВ-Петербург, 2011. — 768 с.
28. Ogunfunmi T. Adaptive nonlinear system identification: the Volterra and Wiener model approaches. — Springer Science + Business Media, LCC. — 2007. — 230 p.
29. Diniz P. S. R. Adaptive filtering algorithms and practical implementation. Third edition. — New York, Springer Science + Business Media, 2008. — 627 p.
30. Sayed A. H. Adaptive filters. — NJ, Hoboken: John Wiley and Sons, Inc., 2008. — 785 p.
31. Mandic D. P., Goh V. S. L. Complex valued nonlinear adaptive filters. — UK, Chichester, 2009. — 324 p.
32. Lee K.-A., Gan W.-S., Kuo S. M. Subband Adaptive Filtering: Theory and Implementation. — UK, West Sussex: John Wiley and Sons, Ltd., 2009. — 324 p.
33. Apolinario J. A., Ed. QRD-RLS Adaptive Filtering. — Springer, 2009. — 356 p.
34. Джиган В. И. Адаптивные фильтры и их приложения в радиотехнике и связи. Часть 1 // Современная электроника. — 2009. — №9. — С. 56—63.
35. Джиган В. И. Адаптивные фильтры и их приложения в радиотехнике и связи. Часть 2 // Современная электроника. — 2010. — №1. — С. 72—77.
36. Джиган В. И. Адаптивные фильтры и их приложения в радиотехнике и связи. Часть 3 // Современная электроника. — 2010. — №2. — С. 70—77.
37. Makino S. Acoustic echo cancellation // IEEE Signal Processing Magazine. — 1997. — Vol. 14. — №5. — P. 39—41.
38. Messerschmitt D. Echo cancellation in speech and data transmission // IEEE Journal on Selected Areas in Communications. — 1984. — Vol. 2. — №2. — P. 283—297.
39. Qureshi S. Adaptive equalization // IEEE Communications Magazine. — 1982. — Vol. 20. — №2. — P. 9—16.



40. Qureshi S. Adaptive equalization // Proceedings of the IEEE. — 1985. — Vol. 73. — №9. — P. 1349—1387.
41. Cioffi J. M. (Stanford University, CA, USA). Частное сообщение, сделанное в Samsung Advanced Institute of Technology. — Ю. Корея, 1996.
42. Djigan V. I. Recursive least squares — an idea whose time has come // Proceedings of the 7-th International Workshop on Spectral Methods and Multirate Signal Processing (Moscow, September 1—2, 2007). — Moscow, 2007. — P. 255—260.
43. Джиган В. И. Прикладная библиотека адаптивных алгоритмов // Электроника: Наука, Технологии, Бизнес. — 2006. — №1. — С. 60—65.
44. Солохина Т., Александров Ю., Петричкович Я. Сигнальные контроллеры компаний «ЭЛВИС»: первая линейка отечественных DSP // Электроника: Наука, Технология, Бизнес. — 2005. — №7. — С. 70—77.
45. Treichler J., Larimore M. New processing techniques based on the constant modulus adaptive algorithm // IEEE Trans. Acoustics, Speech and Signal Processing. — 1985. — Vol. 33. — №2. — P. 420—431.
46. Widrow B. Thinking about thinking: the discovery of the LMS algorithm — DSP history // IEEE Signal Processing Magazine. — 2005. — Vol. 22. — №1. — P. 100—106.
47. Джиган В. И., Солохина Т. В., Петричкович Я. Я. Подавление электрического эха на базе контроллеров «МУЛЬТИКОР» // Электроника: Наука, Технологии, Бизнес. — 2004. — №8. — С. 26—33.
48. Chen W. Y. Simulation techniques and standards development for digital subscriber line systems. — IN: Indianapolis, Macmillan Technical Publishing, 1998. — 644 p.
49. Starr T., Cioffi J. M., Silverman P. J. Understanding digital subscriber line technology. — NJ: Englewood Cliffs, Prentice Hall, 1999. — 474 p.
50. Huntly H. R. Transmission design of intertoll telephone trunks // Bell System Technical Journal. — 1953. — Vol. 32. — Sept. — P. 1019—1036.
51. Allen J. B., Berkley D. A. Image method for efficiently simulating small-room acoustics // The Journal Acoustic Society of America. — 1979. — Vol. 65. — №4. — P. 943—950.
52. Digital network echo cancellers // ITU-T Recommendation G.168. Series G: Transmission systems and media, digital systems and networks. International telephone connections and circuits — Apparatus associated with long-distance telephone circuits. 04/2000. — Geneva. — 2001. — 116 p.
53. Deller J. R., Proakis J. G., Hansen G. H. L. Discrete-time processing of speech signals. — Upper Saddle River, NJ: Prentice Hall, 1993. — 908 p.
54. Chen Y., Le-Ngoc T., Champagne B., Xu C. Recursive least squares constant modulus algorithm for blind adaptive array // IEEE Trans. Signal Processing. — 2004. — Vol. 52. — №5. — P. 1452—1456.

55. Плетнева И.Д., Джиган В.И. Моделирование обработки сигналов в цифровых антенных решетках // Исследования в области цифровых систем связи (Межвузовский сборник). — М.: Изд. МИЭТ, 2007. — С. 36—43.
56. Widrow B. Hoff M. E. Adaptive switching circuits // IRE WESCON Convention Record. — 1960. — Part 4. — P. 96—104.
57. Widrow B., McCool J., Ball M. The complex LMS algorithm // Proceedings of the IEEE. — 1975. — Vol. 63. — №4 — P. 719—720.
58. Frost O. L. An algorithm for linearly constrained adaptive array processing // Proceedings of the IEEE. — 1972. — Vol. 60. — №8. — P. 926—935.
59. Su Y.-L. A complex algorithm for linearly constrained adaptive arrays // IEEE Antennas and Propagation. — 1983. — Vol. 31. — №4. — P. 676—678.
60. Reed I. S. Brief history of adaptive arrays // Proceedings of the Military Communication Conference. — 1985. — Vol. 2,3. — P. 515—518.
61. Widrow B., Mantey P. E., Griffiths L. J., Goode B. B. Adaptive antenna systems // Proceedings of the IEEE. — 1967. — Vol. 55. — №12. — P. 2143—2159.
62. Becker F. K., Holzman L. N., Lucky R. W., Port E. Automatic equalization for digital communication // Proceedings of the IEEE. — 1965. — Vol. 52. — №1. — P. 96—97.
63. Lucky R. W. The adaptive equalizer // IEEE Signal Processing Magazine. — 2006. — Vol. 23. — №3. — P. 104—107.
64. Dentino M., McCool J., Widrow B. Adaptive filtering in frequency domain // Proceedings of the IEEE. — 1978. — Vol. 66. — №12. — P. 1658—1659.
65. Shynk J. J. Frequency-domain and multirate adaptive filtering // IEEE Signal Processing Magazine. — 1992. — Vol. 9. — №1. — P. 14—37.
66. Джиган В.И. Многоканальные RLS- и быстрые RLS-алгоритмы адаптивной фильтрации // Успехи современной радиоэлектроники. — 2004. — №11. — С. 48—77.
67. Джиган В.И. Многообразие лестничных RLS-алгоритмов адаптивной фильтрации // Цифровая обработка сигналов. — 2005. — №3. — С. 2—12.
68. Джиган В.И. Параллельные вычисления в RLS алгоритмах адаптивной фильтрации // Вестник МГТУ им. Н.Э. Баумана. Серия Приборостроение. — 2006. — №1. — С. 30—49.
69. Лавреньев М.А., Шабат Б. В. Методы теории функции комплексного переменного. — М.: Наука. Главная редакция физико-математической литературы, 1987. — 588 с.
70. Brandwood D. H. A complex gradient operator and its applications in adaptive array theory // IEE Proceedings. Parts F and H. — 1983. — Vol. 130. — №1. — P. 11—16.
71. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. — М.: Наука. Главная редакция физико-математической литературы, 1984. — 320 с.



72. Белоусов И. В. Матрицы и определители. — Кишинев: Институт прикладной физики Академии наук Республики Молдова, 2006. — 101 с.
73. Widrow B. Adaptive filters I: fundamentals. — Stanford Electronics Laboratories Technical Report 6764-6. — December 1966. — 58 p.
74. Widrow B., McCool J. M., Larimore M. G., Johnson C. R. Stationary and nonstationary characteristics of the LMS adaptive filter // Proceedings of the IEEE. — 1976. — Vol. 64. — №8. — P. 1151—1162.
75. Horowitz L., Senne K. Performance advantage of complex LMS for controlling narrow-band adaptive arrays // IEEE Trans. Acoustics, Speech and Signal Processing. — 1981. — Vol. 29. — №3. — P. 722—736.
76. Feuer A., Weinstein E. Convergence analysis of LMS filters with uncorrelated Gaussian data // IEEE Trans. Acoustics, Speech and Signal Processing. — 1985. — Vol. 33. — №1. — P. 222—230.
77. Reed I. S. On moment theorem for complex Gaussian processes // // IEEE Trans. Information Theory. — 1962. — Vol. 8. — №3. — P. 194—195.
78. Nagomo J. I., Noda A. A. learning method for system identification // IEEE Trans. Automatic Control. — 1967. — Vol. 12. — №3. — P. 282—287.
79. Bitmead R. R., Anderson B. D. O. Performance of adaptive estimation algorithms in independent random environments // IEEE Trans. Automatic Control. — 1980. — Vol. 25. — №3. — P. 788—794.
80. Hubing N. E., Alexander S. T. Statistical analysis of the soft constrained initialization of recursive least squares algorithms // Proceedings of the International Conference on Acoustic Speech and Signal Processing. — 1991. — P. 1277—1280.
81. Hsia T. C. Convergence analysis of LMS and NLMS adaptive algorithms. // Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. 983. P. 667—670.
82. Douglas S. C., Meng T. H.-Y. Normalized data nonlinearities for LMS adaptation // IEEE Trans. Signal Processing. — 1994. — Vol. 42. — №6. — P. 1352—1365.
83. Slock D. T. M. On the convergence behavior of the LMS and the normalized LMS algorithms // IEEE Trans. Signal Processing. — 1994. — Vol. 41. — №9. — P. 2811—2825.
84. Harris R. W., Chabries D. M., Bishop F. A. A variable step (VS) adaptive filter algorithm // IEEE Trans. Acoust., Speech, Signal Processing. — 1986. — Vol. 34. — №2. — P. 309—316.
85. Kwong R. H., Johnston E. W. A variable step size LMS algorithm // IEEE Trans. Signal Processing. — 1992. — Vol. 40. — №7. — P. 1633—1642.
86. Mathews V. J., Xie Z. A stochastic gradient adaptive filter with gradient adaptive step size // IEEE Trans. Signal Processing. — 1993. — Vol. 41. — №6. — P. 2075—2087.

87. Aboulnasr T., Mayyas K. A robust variable step-size LMS type algorithm: analysis and simulation // IEEE Trans. Signal Processing. — 1997. — Vol. 45. — №3. — P. 631—639.
88. Pazaitis D. I., Constantinides A. G. A novel kurtosis driven variable step-size adaptive algorithm // IEEE Trans. Signal Processing. — 1999. — Vol. 47. — №3. — P. 864—872.
89. Apolinario J. A., Werner S., Diniz P. S. R., Laakso T. I. Constrained normalized adaptive filters for CDMA mobile communication // Proceedings of the 9-th European Signal Processing Conference. — Island of Rhodes, Greece, 1998. — P. 2053—2056.
90. Lee K.-A., Gan W.-S., Kuo S. M. Subband adaptive filtering: theory and implementation. — New York: John Wiley and Sons, Inc., 2009. — 344 p.
91. Линович А. Ю., Витязев В. В., Субполосная адаптивная фильтрация в задачах обратного моделирования // Цифровая обработка сигналов. — 2004. — №4. — С. 41—48.
92. Витязев С. В., Витязев В. В. Методы синтеза узкополосного адаптивного КИХ-фильтра на основе многоскоростной обработки // Цифровая обработка сигналов. — 2007. — №4. — С. 15—20.
93. Soo J.-S., Pang K. K. Multidelay block frequency domain adaptive filter // IEEE Transactions Signal Processing. — 1990. — Vol. 38. — №2. — P. 373—376.
94. Тихонов В. И. Статистическая радиотехника. 2-е изд., перераб. и доп. — М.: Радио и связь, 1982. — 624 с.
95. Slock 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.
96. Benallal A., Gilliore A. A new method to stabilize fast RLS algorithm based on the first-order model of the propagation of numerical errors // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — 1988. — Vol. 5. — P. 1373—1376.
97. Yang B. A note on error propagation analysis of recursive least-squares algorithms // IEEE Trans. Signal Processing. — 1994. — Vol. 42. — №12. — P. 3523—3525.
98. Eleftheriou E., Falconer D. D. Tracking properties and steady-state performance of RLS adaptive filter algorithms // IEEE Trans. Acoust., Speech, Signal Processing. — 1986. — Vol. 34. — №5. — P. 1097—1110.
99. Голуб Дж., Ван Лоун Ч. Матричные вычисления. Пер. с англ. под ред. Воеvodина В. В. — М.: Мир, 1999. — 548 с.
100. Leung H., Haykin S. Stability of recursive QRD-LS algorithms using finite-precision systolic array implementation // IEEE Trans. Acoustics, Speech and Signal Processing. — 1989. — Vol. 37. — №5. — P. 760—763.



101. McWhirter J. G. Recursive least-squares minimization using a systolic array // Proceedings of the SPIE Intern. Sic. Opt. Eng. — 1983. — Vol. 431. — P. 105—112.
102. Alexander S. T., Ghirnikar A. L. A method for recursive least squares filtering based upon an inverse QR decomposition // IEEE Trans. Signal Processing. — 1993. — Vol. 41. — №1. — P. 20—30.
103. Pan C.-T., Plemmons R. J. Least squares modification with inverse factorization: parallel implications // Journal of Computational and Applied Mathematics. — 1989. — Vol. 27. — P. 109—127.
104. Rontogiannis A. A., Theodoridis S. On inverse factorization adaptive least-squares algorithms // Signal Processing. — 1996. — Vol. 52. — P. 35—47.
105. Rader C. M., Steinhardt A. O. Hyperbolic Householder transformations // IEEE Trans. Acoustics, Speech and Signal Processing. — 1986. — Vol. 34. — №6. — P. 1589—1602.
106. Hsieh S. F., Liu K. J. R., Yao K. A unified square-root-free approach for QRD based recursive least squares estimation // IEEE Trans. Signal Processing. — 1993. — Vol. 41. — №3. — P. 1405—1409.
107. Prouder I. K. Fast time-series adaptive-filtering algorithm based on the QRD inverse-updates method // IEE Proceedings: Vision, Image and Signal Processing. — 1994. — Vol. 141. — №5. — C. 325—333.
108. Gill P. E., Golub G. H., Murray W., Sanders M. A. Methods for modifying matrix factorization // Math. Comput. — 1974. — Vol. 25. — P. 505—535.
109. Семушин И. В. Численные методы: учебное пособие для вузов. — Ульяновск: УлГТУ. — 2006. — 178 с.
110. Балашов Е. П., Негода В. Н., Пузанков Д. В. и др. Информационные системы: Табличная обработка информации. Под ред. Балашова Е. П. и Смолова В. Б. — Л.: Энергоатомиздат. Ленинград. отд-ние, 1985. — 184 с.
111. Байков В. Д., Смолов В. Б. Специализированные процессоры: Итерационные алгоритмы и структуры. — М.: Радио и связь, 1985. — 288 с.
112. Wang L.-K., Schulte M. J. Decimal floating-point division using Newton-Raphson iteration // Proceedings of the 15-th IEEE International Conference on Application-Specific Systems, Architectures and Processors. — 2004. — P. 84—95.
113. Steinhardt A. O. Householder transforms in signal processing // IEEE Acoustics, Speech, and Signal Processing Magazine. — 1988. — Vol. 5. — №3. — P. 4—12.
114. Hsieh S. F., Liu K. J. R., Yao K. A unified SQRT-free rank-1 up/down-dating approach for recursive least-squares problems // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — Toronto, Canada. — 1991. — Vol. 2. — P. 1017—1020.
115. Gotze J. Schwiegelshohn U. A square root and division free Givens rotation for solving least squares problems on systolic arrays // SIAM J. Sci. Stat. Comput. — 1991. — Vol. 12. — №4. — P. 800—807.

116. Frantzeskakis E. N., Liu K. J. R. A class of square root and division free algorithms and architectures for QRD-based adaptive signal processing // IEEE Trans. Signal Processing. — 1994. — Vol. 42. — №9. — P. 2455—2469.
117. Takao K., Fujita M., Nishi T. An adaptive antenna array under directional constraint // IEEE Trans. Antennas and Propagation. — 1976. — Vol. 24. — №5. — P. 662—669.
118. Applebaum S., Chapman D. Adaptive arrays with main beam constraints // IEEE Trans. Antennas and Propagation. — 1976. — Vol. 24. — №5. — P. 650—662.
119. Er M., Cantoni A. On an adaptive antenna array under directional constraint // IEEE Trans. Acoustics, Speech and Signal Processing. — 1985. — Vol. 33. — №5. — P. 1326—1328.
120. Ward D. B. Technique for broadband correlated interference rejection in microphone arrays // IEEE Trans. Speech and Audio Processing. — 1998. — Vol. 6. — №4. — P. 414—417.
121. Chen S.-J., Chang C.-Y. Adaptive linearly constrained inverse QRD-RLS beam-forming algorithm for moving jammers suppression // IEEE Trans. Antennas and Propagation. — 2002. — Vol. 50. — №4. — P. 676—678.
122. 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.
123. Resende L. S., Romano J. M. T., Bellanger M. G. Adaptive split transversal filtering: a linearly-constrained approach // Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium. — 2000. — P. 213—217.
124. Proakis J. G. Adaptive digital filters for equalization of telephone channels // IEEE Trans. Audio Electroacoustics. — 1970. — Vol. 18. — №2. — P. 195—200.
125. Falconer D., Ljung L. Application of fast Kalman estimation to adaptive equalization // IEEE Trans. Communications. — 1978. — Vol. 26. — №10. — P. 1439—1446.
126. Carayannis G., Manolakis D., Kalouptsidis N. A fast sequential algorithm for least-squares filtering and prediction // IEEE Trans. Acoustics, Speech, and Signal Processing. — 1983. — Vol. 31. — №6. — P. 1394—1402.
127. Cioffi J. M., Kailath T. Fast, recursive-least squares transversal filters for adaptive filtering // IEEE Trans. Acoustics, Speech, and Signal Processing. — 1984. — Vol. 32. — №2. — P. 304—337.
128. Cioffi J. M., Kailath T. Windowed fast transversal filters adaptive algorithms with normalization // IEEE Trans. Acoustics, Speech, and Signal Processing. — 1985. — Vol. 33. — №3. — P. 607—625.
129. Carayannis G., Manolakis D., Kalouptsidis N. A unified view of parametric processing algorithms for prewindowed signals // Signal Processing. — 1986. — Vol. 10. — P. 335—368.



130. Schutze H., Ren Z. Numerical characteristics of fast recursive least squares transversal adaptation algorithms — a comparative study // *Signal Processing*. — 1992. — Vol. 27. — P. 317—331.
131. Sayed A. H., Kailath T. A state-space approach to adaptive RLS filtering // *IEEE Signal Processing Magazine*. — 1994. — Vol. 11. — №3. — P. 18—60.
132. Glentis G.-O., Berberidis K., Theodoridis S. Efficient least squares adaptive algorithms for FIR transversal filtering // *IEEE Signal Processing Magazine*. — 1999. — Vol. 16. — №4. — P. 13—41.
133. Зинчук В. М., Сосулин Ю. Г., Лимарев А. Е., Мухин Н. П. Адаптивная цифровая фильтрация шумоподобных сигналов в радиотехнических системах // *Цифровая обработка сигналов*. — 2000. — №1. — С. 5—18.
134. Сергиенко А. Б. Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB // *Exponenta Pro: математика в приложениях*. — 2003. — №1. — С. 18—28.
135. Botto J.-L., Moustakides G. V. Stabilizing the fast Kalman algorithms // *IEEE Trans. Acoustics, Speech, and Signal Processing*. — 1989. — Vol. 37. — №9. — P. 1342—1348.
136. Lee D., Morf M., Friedlander B. Recursive least squares ladder estimation algorithms // *IEEE Trans. Acoustics, Speech, and Signal Processing*. — 1981. — Vol. 29. — №3. — P. 627—641.
137. Honig M. L., Messerschmitt D. G. Convergence properties of an adaptive digital lattice filter // *IEEE Trans. Acoustics, Speech and Signal Processing*. — 1981. — Vol. 29. — №3. — P. 642—653.
138. Porat B., Friedlander B., Morf M. Square root covariance ladder algorithms // *IEEE Trans. Automatic Control*. — 1982. — Vol. 27. — №4. — P. 813—829.
139. Porat B. Kailath T. Normalized lattice algorithms for least-squares FIR system identification // *IEEE Trans. Acoustics, Speech, and Signal Processing*. — 1983. — Vol. 31. — №13. — P. 122—128.
140. Lev-Ari H., Kailath T., Cioffi J. Least-squares adaptive lattice and transversal filters: a unified geometric theory // *IEEE Trans. Information Theory*. — 1984. — Vol. 30. — №2. — P. 222—236.
141. Ling F., Manolakis D., Proakis J. Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients // *IEEE Trans. Acoustics, Speech, and Signal Processing*. — 1986. — Vol. 34. — №4. — P. 837—845.
142. Ling F., Proakis J. A generalized multichannel least squares lattice algorithm based on sequential processing stages // *IEEE Trans. Acoustics, Speech and Signal Processing*. — 1984. — Vol. 32. — №2. — P. 381—389.
143. Lev-Ari H. Modular architectures for adaptive multichannel lattice algorithms // *IEEE Trans. Acoustics, Speech and Signal Processing*. — 1987. — Vol. 35. — №4. — P. 543—552.

144. Lewis P. S. QR-based algorithms for multichannel adaptive least squares lattice filters // IEEE Trans. Acoustics, Speech and Signal Processing. — 1990. — Vol. 38. — №3. — P. 421—432.
145. Zhao K., Ling F., Proakis J. G. Multichannel Givens lattice adaptive algorithm // Proceedings of the International Conference on Acoustic Speech and Signal Processing. — 1991. — P. 1849—1852.
146. Ling F. Givens rotation based least squares lattice and related algorithms // IEEE Trans. Signal Processing. — 1991. — Vol. 39. — №7. — P. 1541—1551.
147. Glentis G. O., Kalouptsidis N. Efficient order recursive algorithms for multichannel LS filtering // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — 1990. — Vol. 3. — P. 1429—1432.
148. Glentis G.-O.A., Kalouptsidis N. Efficient order recursive algorithms for multichannel least squares filtering // IEEE Trans. Signal Processing. — 1992. — Vol. 40. — №6. — P. 1354—1374.
149. Glentis G.-O.A., Kalouptsidis N. Fast adaptive algorithms for multichannel filtering and system identification // IEEE Trans. Signal Processing. — 1992. — Vol. 40. — №10. — P. 2433—2458.
150. Glentis G. O., Kalouptsidis N. Efficient multichannel FIR filtering using a single step versatile order recursive algorithm // Signal Processing. — 1994. — Vol. 37. — №1. — P. 437—462.
151. Джиган 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—99.
152. Glentis G., Kalouptsidis N. A highly modular adaptive lattice algorithm for multichannel least squares filtering // Signal Processing. — 1995. — Vol. 36. — №1. — P. 47—55.
153. Glentis G. O., Gerez S. H. Very high speed least squares adaptive multichannel filtering and system identification. — Mierlo, Netherlands // Proceedings of the ProRISC/IEEE Benelux Workshop on Circuits, Systems and Signal Processing — 1995. — P. 123—132.
154. Glentis G. O., Slump C. H. A highly modular normalized adaptive lattice algorithm for multichannel least squares filtering // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — 1995. — Vol. 2. — P. 1420—1423.
155. Gomes G., Barroso V. Multichannel lattice filtering based on parallel scalar modules // Proceedings of the 2-th Conference on Telecommunications. — Sesimbra, Portugal. — 1999. — P. 151—155.
156. Gomes G., Barroso V. QR-RLS adaptation of modular multichannel adaptive filters // Proceedings of the 4-th Conference on Telecommunications. — Aveiro, Portugal. — 2003. — P. 135—138.



157. Gomes G., Barroso V. A CORDIC-based QR-RLS multichannel lattice filter // European Signal Processing Conference. — Pozgan, Poland. — 2007. — P. 1043—1047.
158. Gomes G., Barroso V. Array-Based QR-RLS Multichannel Lattice Filtering // IEEE Trans. Signal Processing. — 2008. — Vol. 56. — №8. — P. 3510—3522.
159. Gay S. L. Dynamically regularized fast RLS with application to echo cancellation // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — 1996. — P. 957—960.
160. Ozeki K., Umeda K. An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties // Trans. IECE Japan. — 1984. — Vol. J67A. — №2. — P. 126—132.
161. Tanaka M., Kaneda Y., Makino S. Reduction of high order projection algorithm // Proceedings of the EICS Fall Conference. — Tokio, Japan. — 1993. — P. A-101.
162. Gay S. L. A fast converging, low complexity adaptive filtering algorithm // Proc. of the 3-rd International Workshop on Acoustic Echo Control. — Plestin les Greves, France. — 1993. — P. 223—226.
163. Gay S. L., Tavathia S. The fast affine projection algorithm // Proceedings of the International Conference on Acoustic, Speech and Signal Processing. — Detroit, Michigan, US. — 1995. — Vol. 5. — P. 3023—3026.
164. Tanaka M., Kaneda Y., Makino S., Kojima J. A fast projection algorithm for adaptive filtering // IEICE Transactions. Fundamentals. — 1995. — V. E78A. — №10. — P. 1355—1361.
165. Djigan V. I. Improved fast affine projection algorithm with gradient adaptive step-size // Proceedings of the 3-rd International Conference on Antennas, Radiocommunication Systems & Means (ICARSM-97) (Voronezh Construction Bureau of Antenna Design, 26—29 May 1997). — Voronezh, 1997. — Vol. 3. — P. 23—32.
166. Джиган В. И. Быстрый многоканальный алгоритм аффинных проекций с комплексными коэффициентами для адаптивной фильтрации // Доклады 5-й Международной конференции «Цифровая обработка сигналов и ее применения (DSPA-2003)» (Российская академия наук: Институт проблем управления им. В. А. Трапезникова, 12—14 марта 2003 г.). — Москва, 2003. — Том 1. — С. 39—44.
167. Джиган В. И. Уменьшение вычислительных затрат в линейно-ограниченном алгоритме аффинных проекций // Труды 60-й научной сессии, посвященной Дню Радио (Московский технический университет связи и информатики, 17 — 19 мая 2005 г.). — Москва, 2005. — Том 1. — С. 99—102.
168. Джиган В. И. Особенности построения быстрого алгоритма аффинных проекций для многоканальной адаптивной фильтрации // Радиотехника и электроника. — 2005. — №11. — С. 1391—1399.

169. Джиган В. И. Многоканальные линейно-ограниченные RLS алгоритмы со скользящим окном // Радиотехника. — 2005. — №10. — С. 32—36.
170. Джиган В. И. Об использовании параллельных вычислений в лестничных адаптивных алгоритмах // Телекоммуникации. — 2005. — №12. — С. 2—9.
171. Джиган В. И. Параллельные линейно-ограниченные RLS алгоритмы адаптивной фильтрации // Вестник Московского государственного авиационного института (технического университета). — 2005. — №3. — С. 81—92.
172. Джиган В. И. Алгоритмы и структуры лестничных адаптивных фильтров на основе QR-разложения со скользящим окном и динамической регуляризацией корреляционной матрицы // Труды учебных заведений связи. — 2005. — №173. — С. 130—141.
173. Джиган В. И. Простое представление параллельных линейно-ограниченных алгоритмов адаптивной фильтрации // Вестник Рязанской государственной радиотехнической академии. — 2005. — №2. — С. 33—41.
174. Djigan V.I. Lattice RLS for nonstationary signal processing // Radioengineering: Proceedings of Czech and Slovak Technical Universities and URSI Committers. — 2006. — Vol. 15. — №1. — P. 25—31.
175. Djigan V.I. Multichannel parallelizable sliding window RLS and fast RLS algorithms with linear constraints // Signal Processing (Elsevier). — 2006. — Vol. 86. — №4. — P. 776—791.
176. Джиган В. И. Условия эквивалентности рекурсивных алгоритмов адаптивной фильтрации по критерию наименьших квадратов // Телекоммуникации. — 2006. — №6. — С. 6—11.
177. Rontogiannis A. A., Theodoridis S. New fast QR decomposition least squares adaptive algorithms // IEEE Trans. Signal Processing. — 1998. — Vol. 46. — №8. — P. 2113—2121.
178. Rontogiannis A. A., Theodoridis S. Multichannel fast QRD-LS adaptive filtering: new technique and algorithms // IEEE Trans. Signal Processing. — 1998. — Vol. 46. — №11. — P. 2862—2876.
179. Prouder I. K., McWhirter J. G., Shepherd T. J. Computationally efficient QR decomposition approach to least squares adaptive filtering // IEE Proceedings: F. — 1991. — Vol. 138. — №4. — С. 341—353.
180. Terre M., Bellanger M. A fast least squares QRD-based algorithm for complex data // IEEE Trans. Signal Processing. — 1994. — Vol. 42. — №11. — P. 3272—3273.
181. Prouder I. K. Fast time-series adaptive-filtering algorithm based on the QRD inverse-updates method // IEE Proceedings: Vision, Image and Signal Processing. — 1994. — Vol. 141. — №5. — С. 325—333.
182. 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.



183. Mazuch T., Kozanek J. New recurrent algorithm for a matrix inversion // Journal of Computational and Applied Mathematics. — 2001. — Vol. 136. — №1,2. — P. 219—226.
184. Джиган В. И. Алгоритмы адаптивной фильтрации нестационарных сигналов // Диссертация на соискание ученой степени доктора технических наук. — Государственное унитарное предприятие города Москвы Научно-производственный центр «Электронные вычислительно-информационные системы». — Москва, 2006. — 342 с.
185. De Campos M. R. L., Apolinario J. A. The constrained affine projection algorithm — development and convergence issues // Proceedings of the First Balkan Conference on Signal Processing, Communications, Circuits, and Systems. — Istanbul, May 2000. — 4 p.
186. Lin D. W. On digital implementation of the fast Kalman algorithms // IEEE Trans. Acoustics, Speech and Signal Processing. — 1984. — Vol. 32. — №5. — P. 998—1005.
187. <http://www.mathworks.com/> Интернет-сайт компании MathWorks.
188. <http://www.ni.com/> Интернет-сайт компании National Instruments.
189. <http://www.agilent.com/> Интернет-сайт компании Agilent Technologies.
190. <http://www.ti.com/> Интернет-сайт компании Texas Instruments.
191. <http://www.analog.com/> Интернет-сайт компании Analog Devices.
192. <http://www.freescale.com/> Интернет-сайт компании Freescale Semiconductor.
193. <http://www.multicore.ru/> Интернет-сайт ОАО НПЦ «ЭЛВИС».
194. Hayes M. H., Treichler J. Adaptive filtering (best of the web) // IEEE Signal Processing Magazine. — 2008. — Vol. 25. — № 6. — 169—172.
195. Джиган В. И. Библиотека алгоритмов адаптивной фильтрации // Доклады 6-й Международной конференции «Цифровая обработка сигналов и ее применения (DSPA-2004)» (Российская академия наук: Институт проблем управления им. В. А. Трапезникова, 31 марта — 2 апреля 2004 г.). — Москва, 2004. — Том 1. — С. 89—94.
196. Джиган В. И. Разработка алгоритмов адаптивной фильтрации с применением среды «MATLAB» // Труды 2-й Всероссийской научной конференции «Проектирование инженерных и научных приложений в среде MATLAB» (Российская академия наук: Институт проблем управления им. В. А. Трапезникова, 25—26 мая 2004 г.). — Москва, 2004. — С. 1689—1696 (CD).
197. Джиган В. И. Оценка эффективности реализации алгоритмов адаптивной фильтрации на базе «систем на кристалле» серии «МУЛЬТИКОР» // Материалы докладов Всероссийской научно-технической конференции «Проблемы разработки перспективных микроэлектронных систем (МЭС-2005)» (Российская академия наук: Институт проблем проектирования в микроэлектронике, 11—14 октября 2005 г.). — Москва, 2005. — С. 453—460.

198. Джиган В. И., Коплович Е. А. Особенности реализации алгоритмов адаптивной фильтрации на базе микропроцессорных БИС серии «МУЛЬТИКОР» // Труды 1-й Международной научно-технической конференции «Современные проблемы оптимизации в инженерных приложениях (IWOPE-2005)» (Ярославский государственный университет, 17–24 декабря 2005 г.). — Ярославль, 2005. — Том 2. — С. 262–270.
199. Джиган В. И., Коплович Е. А. Лестничные адаптивные фильтры на основе СБИС сигнального контроллера 1892ВМ3Т серии «Мультикор» // Вопросы радиоэлектроники. Общетехническая серия. — 2006. — №2. — С. 126–136.
200. Джиган В. И. Адаптивные антенные решетки на базе СБИС серии «Мультикор» // Вопросы радиоэлектроники. Серия «Электронная вычислительная техника». — 2008. — №3. — С. 57–69.
201. Гончаров А. А., Хребтов А. Ю., Джиган В. И. Опыт реализации параллельных алгоритмов адаптивной фильтрации на базе СБИС серии «Мультикор» // Материалы 16-й Международной конференции «Информационные средства и технологии» (Московский энергетический институт — технический университет, 21–23 октября 2008 г.). — Москва, 2008. — Том 1. — С. 137–143.
202. Джиган В. И. Виртуальная адаптивная антenna решетка в среде LabVIEW// Материалы 7-й Международной научно-практической конференции «Образовательные, научные и инженерные приложения в среде LabVIEW и технологии National Instruments». (Российский университет дружбы народов, 28–29 ноября 2008 г.). — Москва, 2008. — Том. 2. — С. 52–58.
203. Богданов И. П., Джиган В. И. Виртуальный инструмент для исследования свойств алгоритмов адаптивной фильтрации // Материалы 9-й Международной научно-практической конференции «Образовательные, научные и инженерные приложения в среде LabVIEW и технологии National Instruments». (Российский университет дружбы народов, 3–4 декабря 2010 г.). — Москва, 2010. — С. 260–263.
204. Джиган В. И. Виртуальная многолучевая антenna решетка на основе RLS-алгоритма адаптивной фильтрации // Материалы 10-й Международной научно-практической конференции «Образовательные, научные и инженерные приложения в среде LabVIEW и технологии National Instruments». (Московский технический университет связи и информатики, 8–9 декабря 2011 г.). — Москва, 2011. — С. 5–8.
205. Белиовская Л., Джиган М., Джиган О. Совместное использование пакетов LabVIEW и MATLAB в задачах эхокомпенсации и выравнивания каналов связи // Современная электроника. — 2007. — №3. — С. 56–60.
206. Джиган О. В., Джиган М. В. Виртуальный инструмент для исследования свойств адаптивных фильтров в среде LabVIEW // Материалы 10-й Международной научно-практической конференции «Образовательные, науч-



- ные и инженерные приложения в среде LabVIEW и технологии National Instruments». (Московский технический университет связи и информатики, 8—9 декабря 2011 г.). — Москва, 2011. — С. 9—12.
207. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB: учебн. пособие. — С.Пб.: БХВ-Петербург, 2008. — 816 с.
208. Солонина А. И., Цифровая обработка сигналов. Моделирование в Simulink: учебн. пособие. — С.Пб.: БХВ-Петербург, 2008. — 432 с.
209. Кехтарнаваз Н., Ким Н. Цифровая обработка сигналов на системном уровне с использованием LabVIEW/ Пер. с англ. под ред. В. К. Макухи. — М.: Издательский дом «Додэка-XXI», 2007. — 304 с.
210. Ануфриев И., Смирнов А., Смирнова Е. MATLAB 7. Наиболее полное руководство. — СПб.: «БХВ-Петербург», 2005. — 1102 с.
211. Трэвис Д., Кринг Д. LabVIEW для всех. 4-е издание, переработанное и дополненное. — М.: ДМК Пресс, 2011. — 904 с.
212. Sigale D. Digital communication systems using SystemVue. — Charles River Media, 2006. — 350 р.
213. Загидуллин Р.Ш., Карутин С. Н., Стешенко В. Б. SystemView. Системотехническое моделирование устройств обработки сигналов. — М.: Горячая Линия — Телеком, 2005. — 296 с.
214. <http://www.dspalgorithms.com> Интернет-сайт компании DSP Algorithms.
215. Белиовская Л., Джиган М., Джиган О. Вычисление импульсных откликов акустических и электрических каналов // Современная электроника. — 2007. — №9. — С. 42—44.
216. Джиган В. И. Многолучевая адаптивная антенная решетка // Известия ЮФУ. Технические науки. — 2012. — №2. — С. 23—29.
217. Djigan V. I. Joint use of constant modulus and least squares criteria in linearly-constrained communication arrays // Radioengineering: Proceedings of Czech and Slovak Technical Universities and URSI Committers. — 2007. — Vol. 16. — №4. — Р. 88—95.
218. Джиган В. И. Плетнева И. Д. Линейно-ограниченный нормализованный алгоритм по критерию наименьшего среднеквадратичного отклонения для цифровой адаптивной антенной решетки // Информационные технологии. — 2008. — №10. — С. 68—74.
219. Джиган В. И., Плетнева И. Д. Применение СМ-алгоритма аффинных проекций с линейными ограничениями для адаптивной фильтрации сигналов в антенной решетке // Антенны. — 2008. — №10. — С. 14—24.
220. Algorithm design library. SystemVue 211.10. — Agilent Technologies, 2011. — 574 р.
221. Александров Ю. Н., Никольский В. Ф., Кучинский А. С. и др. Библиотека прикладных функций в составе среды MCStudio™ для разработки про-

- граммного обеспечения «систем на кристалле» серии МУЛЬТИКОР // Материалы докладов Всероссийской научно-технической конференции «Проблемы разработки перспективных микроэлектронных систем (МЭС-2005)» (Российская академия наук: Институт проблем проектирования в микроэлектронике, 11–14 октября 2005 г.). — Москва, 2005. — С. 516–522.
222. Синицын В.В., Косырев С.А., Никольский В.Ф. Среда разработки программного обеспечения для «систем на кристалле» серии «МУЛЬТИКОР» MCStudio_Lnx // Материалы докладов Всероссийской научно-технической конференции «Проблемы разработки перспективных микроэлектронных систем (МЭС-2005)» (Российская академия наук: Институт проблем проектирования в микроэлектронике, 11–14 октября 2005 г.). — Москва, 2005. — С. 406–411.
223. Витязев С.В. Адаптивный гребенчатый фильтр на процессорах «МУЛЬТИКОР» // Вопросы Радиоэлектроники. Серия «Электронная вычислительная техника». — 2008. — №3. — С. 121–126.
224. TMS320C55x DSP library. Programmer's reference. SPRU422J. — Texas Instrument. — January 2009.
225. TMS320C64x DSP library. Programmer's reference. SPRU565B. — Texas Instrument. — October 2003.
226. TMS320C67x DSP library. Programmer's reference. SPRU657C. — Texas Instrument. — June 2010.
227. ADSP-21000 family. Application handbook. Volume 1. — Analog Devices. — 1994. — 333 р.
228. Витязев С. В. Программная реализация цифрового фильтра-дециматора на цифровых сигнальных процессорах TigerSHARC ADSP-TS101 // Международной конференции «Цифровая обработка сигналов и ее применения (DSPA-2003)» (Российская академия наук: Институт проблем управления им. В.А. Трапезникова, 31 марта — 2 апреля 2010 г.). — Москва, 2010. — Том 2. — С. 259–261.
229. VisualDSP++. C/C++ compiler and library manual for TigerSHARC processors. Part Number 80-000336-03. — Analog Devices. — 2004.
230. 24-Bit digital signal processors DSP56300 family manual. DSP56300FM. — Free-scale Semiconductor. — April 2005.
231. Витязев С. В. Цифровые процессы обработки сигналов. — Рязань: Изд. РГРТУ, 2012. — 115 С.

Предметный указатель

- А**
- Автокорреляционная функция 32
Авторегрессионный процесс 97, 142, 205
Адаптация 21
Адаптивная обработка сигналов 21
— фильтрация узкополосных сигналов 57
— шумоочистка 20, 56
Адаптивные антенные решетки 16, 27, 51
— компенсаторы сигналов
акустического эха 20, 28, 57
— компенсаторы сигналов
электрического эха 20, 41
— компенсаторы шумов 57
— эквалайзеры (выравниватели)
электрических характеристик каналов связи 20
Адаптивный фильтр 22
— на основе апостериорных ошибок 340
— на основе априорных ошибок 340
Аддитивный шум 41
Акустический импульсный отклик 486
Акустическое эхо 20, 28, 40, 88, 298, 451
Алгоритм аффинных проекций 419, 443
Алгоритм наискорейшего спуска 110, 118, 120
Алгоритм Ньютона 17, 110, 112, 114, 123, 131, 166
Алгоритм стохастического градиентного поиска 135
Алгоритм Фроста 284
Алгоритмическая сложность 30, 183
Алгоритмы на основе критерия наименьших квадратов 33
Алгоритмы обработки нестационарных сигналов 419
Амплитудно-частотная характеристика 41
Аналитическая функция 64
Аналого-цифровые и цифро-аналоговые преобразователи 39
Апостериорная ошибка 26
Априорная ошибка 26
Арифметика действительных чисел 204, 229, 243, 259
Арифметика комплексных чисел 204, 231, 243, 259
Арифметика с плавающей точкой 261, 308, 462, 474
- Б**
- Безусловная винеровская фильтрация сигналов 86
Белый шум 33
Блочные градиентные алгоритмы 140
Быстрые алгоритмы на основе обратного QR-разложения 420
Быстрые лестничные RLS-алгоритмы 324
Быстрые RLS-алгоритмы
адаптивной фильтрации 307
Быстрый алгоритм аффинных проекций 444
Быстрый алгоритм Калмана 315

**В**

Вектор взаимной корреляции 72
Вектор коэффициентов
 Калмана 199
Вектор-столбец 26
Вектор значений ограничеваемого
 параметра 103
Верхняя треугольная квадратная
 матрица 220
Весовые коэффициенты 120
Взаимокорреляционная
 функция 74
Взвешенные наименьшие
 квадраты 31
Винеровская фильтрация 86
Возрастающее окно 325, 420
Вращения Гивенса 218
Выходной сигнал адаптивного
 фильтра 22, 57
Вычислительная сложность
 алгоритма 28

Г

Геометрическая прогрессия 113,
 170
Гибридные или дифференциальные
 схемы 38
Гиперокружность 97
Гиперпараболоид 104,
Гиперплоскость 92, 176, 193
Гиперэллипсы 91, 96, 102
Главные оси 92
Градиентные алгоритмы 30
Градиентные алгоритмы
 с переменным шагом
 сходимости 171
Градиентный спуск 209
Графики линий постоянного
 уровня 91, 102

Д

Дельта-функция 46
Диагональная матрица 26
Диаграмма направленности 51
Дифференцирование функции
 комплексной переменной 60
Длительность одной итерации
 адаптивного алгоритма 23
Длительность переходного
 процесса 23
Дополнительная
 среднеквадратическая
 ошибка 146, 154

Е

Единичная матрица 26
Единичный вектор 26

З

Знаменатель геометрической
 прогрессии 114, 170, 203

И

Импульсный отклик 40
Индекс дискретного времени или
 номер отсчета обрабатываемого
 сигнала 23
Инициализация 137
Интерфейс пользователя 471
Итерационные процедуры 29, 118

К

Квадрат мгновенной ошибки 32
Квадратичная функция
 ошибки 22
Квадратная матрица 25, 76
Комплексная огибающая 31
Комплексное сопряжение 205
Комплексное число 48, 60



Комплексный градиент 59
 Корреляционная матрица 72
 Коэффициент ослабления эхосигнала 36
 Коэффициент отражения 327
 Критерий наименьших квадратов 29, 50
 Критерий постоянства модуля информационных символов 31

Л

Лемма об обращении матриц 156
 Лестничный RLS-алгоритм в пространстве состояний с использованием вращений Гивенса без операций извлечения квадратного корня 372
 — на основе апостериорных ошибок линейного предсказания 342
 — на основе априорных и апостериорных ошибок 339
 — на основе априорных ошибок линейного предсказания 345
 — с использованием вращений Гивенса без операций извлечения квадратного корня 372
 — с использованием вращений Гивенса с операциями извлечения квадратного корня 367
 — с обратными связями на основе апостериорных ошибок линейного предсказания 350
 — с обратными связями на основе априорных и апостериорных ошибок линейного предсказания 346

— с обратными связями на основе априорных ошибок линейного предсказания 347
 Линейная вычислительная сложность 30, 293
 Линейное ограничение 51
 Линейное предсказание 57
 Линейное предсказание вперед 300
 Линейное предсказание назад 304
 Линейно-ограниченная винеровская фильтрация 102
 Линейно-ограниченная оптимизация 103
 Линейно-ограниченная рекурсивная адаптивная фильтрация сигналов по критерию наименьших квадратов 284
 Линейно-ограниченные LMS-алгоритмы 59, 173
 Линейно-ограниченный алгоритм аффинных проекций 455
 Линейно-ограниченный RLS-алгоритм 287
 Линейный сумматор 72

М

Масштабирующий множитель динамического шага сходимости 44, 167
 Матрица 25
 Матрица Вандермонда 81
 Матрица вращения Гивенса 221, 228, 231
 Матрица ортонормированная 84
 Матрица унитарная 84
 Матрица Хаусхолдера 263
 Межсимвольная интерференция 45



- Метод наименьших квадратов 182
Минимум целевой функции 22,
104
Мнимальная единица 50
Многоканальный адаптивный
КИХ-фильтр 27
Многоканальный быстрый
стабилизированный
RLS-алгоритм на основе
обратного QR-разложения с
использованием вращений
Гивенса без операций
извлечения квадратного
корня 424
Многоканальный быстрый
RLS-алгоритм на основе
обратного QR-разложения с
использованием вращений
Гивенса 422
Многоканальный быстрый
RLS-алгоритм на основе
обратного QR-разложения с
использованием вращений
Гивенса без операций
извлечения квадратного
корня 423
Многоканальный линейно-
ограниченный алгоритм
аффинных проекций 455
Многоканальный
регуляризованный LC
RLS-алгоритм с двойным
ограничением 428
Многоканальный
регуляризованный RLS-
алгоритм с возрастающим
окном 427
Многоканальный
регуляризованный RLS-
алгоритм со скользящим окном
433
- Многоканальный 1-SUSD
стабилизированный FAEST-
алгоритм 417
Многоканальный 1-SUSD FAEST-
алгоритм 415
Многоканальный 1-SUSD FK-
алгоритм 411
Многоканальный 1-SUSD FTF-
алгоритм 414
Многоканальный B-SUSD FAEST-
алгоритм 396
Многоканальный B-SUSD FK-
алгоритм 393
Многоканальный B-SUSD FTF-
алгоритм 394
Многоканальный FAP-
алгоритм 444, 451
Многоканальный LC RLS-алгоритм
с двойным ограничением и
скользящим окном 431
Множители Лагранжа 103
- Н**
Наименьшие квадраты 31
Направляющие векторы 91
Неравенство Шварца 69
Несмешенная система
координат 90
Нижняя треугольная матрица 225
Норма вектора 69
Нормализованный адаптивный
фильтр на основе апостериорных
ошибок 359
Нормализованный алгоритм по
критерию наименьшего среднего
квадрата 41
Нормализованный лестничный
алгоритм 350
Нормализованный лестничный
RLS-алгоритм с обратными
связями на основе



апостериорных ошибок линейного предсказания 358
 Нормированные собственные векторы 83
 Нулевой вектор 26
 Нулевые матрицы 26

О

Обратная идентификация объекта 44
 Обратная корреляционная матрица 75, 111
 Обратное отношение ошибок 314
 Обратное QR-разложение 219
 Обратные подстановки 219
 Обучающая кривая 111
 Одновременное использование скользящего окна и регуляризации в RLS-алгоритмах 432
 Одноканальный адаптивный фильтр 23
 Оператор комплексного градиента 61
 Определитель матрицы 76
 Ортогональность собственных векторов 82
 Ортонормированные векторы 83
 Относительное значение дополнительной среднеквадратической ошибки 146
 Отношение Релея 82
 Отношение сигнал–шум 463

П

Параллельные вычисления 241
 Параллельные RLS-алгоритмы 438
 Перестановочные матрицы 378
 Переходный процесс 23

Период дискретизации 23
 Повернутая и смещенная система координат 89
 Поверхность среднеквадратической ошибки 89
 Положительно определенная матрица 76
 Положительно полуопределенная матрица 76
 Постоянные времени 114
 Предсказание вперед 299
 Предсказание назад 299
 Преобразование Карунена–Лоева 133
 Преобразование Хаусхолдера 261, 265
 Преобразование частоты 108
 Прикладная библиотека 464
 Прикладная библиотека адаптивных фильтров 485
 Принцип ортогональности 88
 Проекционная матрица 176
 Пространственные фильтры 21
 Прямая идентификация неизвестной линейной системы 33
 Прямое QR-разложение 205, 219
 Прямоугольная матрица 25

Р

Разброс собственных чисел 97
 Размер проекции 444
 Рассогласование 36
 Регуляризация RLS-алгоритмов 425
 Рекурсивное вычисление отношения ошибок 313
 Рекурсивное вычисление ошибок линейного предсказания 330



Рекурсивное вычисление ошибок моделирования требуемого сигнала адаптивного фильтра 334

Рекурсивное вычисление энергий ошибок линейного предсказания 325

Рекурсивные фильтры 24

Рекурсивный метод наименьших квадратов 182

С

Свойство инвариантности к сдвигу 299

Сигнал ошибки 23, 33

Сигнал электрического эха 39

Синфазная и квадратурная составляющие комплексной огибающей 107

Скалярное произведение 69

Скользящее окно 42, 419, 428, 432

Смешенная система координат 89

Собственные векторы 78

Собственные числа 78

Соотношение между апостериорными и априорными ошибками 312

Среднеквадратическая ошибка 26, 86, 89, 93, 99

Стабилизированный FAEST-алгоритм 322

Стационарная точка 68

Структура фильтра 22, 24

Т

Теория функции комплексного переменного 63

Теплицева матрица 75

Трансверсальные фильтры 24

Транспонирование 25

Требуемый сигнал 23

Тренировочная последовательность символов 46, 204

У

Унимодальная квадратичная функция 26, 182

Унитарная матрица 84

Уравнение Винера–Хопфа 87

Уравнение переменного шага сходимости 171

Условия Коши–Римана дифференцируемости функции комплексного переменного 65

Условная оптимизация 103

Установившийся режим 36

Ф

Фильтр с изменяемыми параметрами 22

Фильтры с бесконечной импульсной характеристикой (БИХ) 24

Фильтры с конечной импульсной характеристикой (КИХ) 24

Х

Характеристическое уравнение 78, 94, 100

Ц

Целевая функция 22, 28, 31, 61

Цифровая обработка сигналов 51

Ч

Частота дискретизации обрабатываемых сигналов 21

Число арифметических операций 28

III

Шаг сходимости 44, 111
Шумоочистка сигналов 53

Э

Эквалайзеры 20
Эллиптический параболоид 89

Энергия ошибок линейного
предсказания 460
Эрмитова матрица 75
Эрмитово сопряжение 25
Эхо-компенсация 37
Эхо-сигнал 36

Производство книг на заказ
Издательство «Техносфера»
тел.: (495) 234-01-10
e-mail: knigi@technosphera.ru

Реклама в книгах:

- модульная
- статьи

**Подробная информация о книгах на сайте
www.technosphera.ru**

**Джиган В.И.
Адаптивная фильтрация сигналов:
теория и алгоритмы**

Компьютерная верстка – С.С. Бегунов

Корректор – О.Ч. Кохановская

Дизайн книжных серий – С.Ю. Биричев

Дизайн – А.А. Давыдова

Выпускающий редактор – С.Ю. Артемова

Ответственный за выпуск – О.А. Казанцева

Формат 70 x 100/16. Печать офсетная.

Гарнитура Ньютон

Печ.л.33. Тираж 1000 экз. Зак. №

Бумага офсет. №1, плотность 65 г/м²

Издательство «Техносфера»

Москва, ул. Краснопролетарская, д.16, стр.2

Отпечатано в ГУП Чувашской Республики
«ИПК «Чувашия» Мининформполитики Чувашии
428019, Чувашская Республика,
г. Чебоксары, пр. И. Яковлева, 13