

Алгоритмы адаптивной фильтрации: особенности реализации в MATLAB

Обзор посвящен цифровым адаптивным фильтрам, параметры которых автоматически подстраиваются под статистические свойства обрабатываемого сигнала. Это позволяет создавать системы обработки сигналов, успешно функционирующие в присутствии шумов и помех с неизвестными заранее свойствами.

Приведены теоретические положения, лежащие в основе наиболее распространенных алгоритмов адаптивной фильтрации — наименьших квадратов, рекурсивного наименьших квадратов и Калмана.

Моделирование процесса адаптивной обработки сигнала можно реализовать с использованием MATLAB: в пакете расширения Filter Design и в наборе блоков DSP Blockset среды моделирования Simulink имеются функции и блоки, реализующие наиболее распространенные алгоритмы адаптивной фильтрации. Примеры решаемых с помощью адаптивных фильтров задач: идентификация систем, подавление шума, компенсация искажений сигнала, формирование диаграмм направленности антенных решеток.

Устройства, в которых используются адаптивные фильтры: модемы, сотовые телефоны, системы цифровой и аналоговой связи, системы обработки речи. Статья адресована специалистам в области телекоммуникаций и радиотехники, а также всем, кто заинтересован в применении адаптивных фильтров в своих разработках.

1. Введение

При поиске оптимальных алгоритмов обработки сигнала неизбежно приходится опираться на некоторые *статистические модели* сигналов и шумов. Чаще всего при формировании этих моделей используются концепции линейности, стационарности и нормальности (гауссовости). Однако перечисленные принципы далеко не всегда выполняются на практике, а от адекватности выбранной модели в значительной мере зависит качество приема сигнала. Возможным решением проблемы является использование адаптивных фильтров, которые позволяют системе подстраиваться под статистические параметры входного сигнала, не требуя при этом задания каких-либо моделей.

Появившись в конце 1950-х годов, адаптивные фильтры прошли большой путь, превратившись из экзотической технологии, применявшейся преимущественно в военных целях, в «ширпотреб», без которого сейчас была бы немислима работа модемов, сотовых телефонов и многого другого. Существует большое количество адаптивных алгоритмов, различающихся вычислительной сложностью, особенностями поведения, используемыми исходными данными и структурами самих адаптирующихся систем. В статье рассматриваются несколько основных алгоритмов, более подробные сведения можно найти в [1, 2, 3, 4].

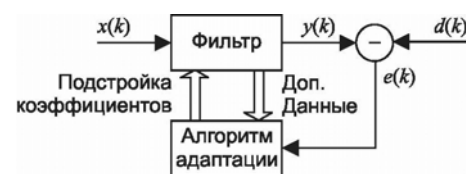
2. Базовая идея адаптивной обработки сигнала

Общая структура адаптивного фильтра показана на рис. 1. Входной дискретный сигнал $x(k)$ обрабатывается дискретным фильтром, в результате чего получается выходной сигнал $y(k)$. Этот выходной сигнал сравнивается с *образцовым* сигналом $d(k)$, разность между ними образует сигнал ошибки $e(k)$. Задача адаптивного фильтра — минимизировать ошибку воспроизведения образцового сигнала. С этой целью блок адаптации после обработки каждого отсчета анализирует сигнал ошибки и дополнительные данные, поступающие из фильтра, используя результаты этого анализа для подстройки параметров (коэффициентов) фильтра.

Возможен и иной вариант адаптации, при котором образцовый сигнал не используется. Такой режим работы называется *слепой адаптацией* (blind adaptation) или *обучением без учителя* (unsupervised learning). Разумеется, в этом случае требуется некоторая информация о структуре полезного входного сигнала (например, знание типа и параметров используемой модуляции). Очевидно, что слепая адаптация является более сложной вычислительной задачей, чем адаптация с использованием образцового сигнала. В статье эти алгоритмы рассматриваться не будут. Некоторые сведения о них можно найти в [4, 5].

Может показаться, что алгоритмы с использованием образцового сигнала лишены практического смысла, поскольку выходной сигнал должен быть заранее известен. Однако есть целый ряд практических задач, при решении которых образцовый сигнал оказывается доступен. Подробнее об этом пойдет речь в разделе 8. Следует отметить, что в ряде случаев при этом полезным сигналом является не выходной сигнал фильтра, а *сигнал ошибки*, то есть разность между образцовым сигналом и выходным сигналом адаптивного фильтра.

В качестве фильтра в структуре, показанной на рис. 1, чаще всего используется нерекурсивный цифровой фильтр. Одним из главных достоинств этого варианта является то, что нерекурсивный фильтр является устойчивым при любых значениях коэффициентов. Однако следует помнить, что алгоритм адаптации в любом случае вносит в систему обратную связь, вследствие чего адаптивная система в целом может стать неустойчивой.



▲ Рис. 1. Общая структура адаптивного фильтра.

Существуют адаптивные алгоритмы и для рекурсивных фильтров, однако при их разработке возникают серьезные проблемы, прежде всего связанные с устойчивостью, поэтому такие фильтры не получили широкого распространения. Некоторые теоретические сведения на эту тему содержатся в [2]. Еще один класс адаптивных систем — *нейронные сети* (neural networks), которые в определен-

ной степени моделируют работу нервной системы живых организмов.

Далее будут рассмотрены три адаптивных алгоритма с использованием образцового сигнала, часто применяемых на практике в различных системах обработки информации. Для упрощения математических выкладок предположим, что сигналы и фильтры являются вещественными. Однако результирующие формулы легко обобщаются на случай комплексных сигналов и фильтров.

3. Оптимальный фильтр Винера

Прежде чем рассматривать собственно алгоритмы адаптации, необходимо определить те оптимальные параметры фильтра, к которым эти алгоритмы должны стремиться. Подход к задаче оптимальной фильтрации может быть как статистическим, так и детерминированным. Сначала рассмотрим статистический вариант.

Пусть входной дискретный случайный сигнал $x(k)$ обрабатывается нерекурсивным дискретным фильтром порядка N , коэффициенты которого могут быть представлены вектор-столбцом $\mathbf{w} = [w_0, w_1, \dots, w_N]^T$. Выходной сигнал фильтра равен

$$y(k) = \mathbf{u}^T(k)\mathbf{w}, \quad (3.1)$$

где $\mathbf{u}(k) = [x(k), x(k-1), \dots, x(k-N)]^T$ — вектор-столбец содержимого линии задержки фильтра на k -м шаге.

Кроме того, имеется образцовый (также случайный) сигнал $d(k)$. Ошибка воспроизведения образцового сигнала равна

$$e(k) = d(k) - y(k) = d(k) - \mathbf{u}^T(k)\mathbf{w}. \quad (3.2)$$

Необходимо найти такие коэффициенты фильтра \mathbf{w} , которые обеспечивают максимальную близость выходного сигнала фильтра к образцовому, то есть минимизируют ошибку $e(k)$. Поскольку $e(k)$ также является случайным процессом, в качестве меры ее величины разумно принять средний квадрат. Таким образом, оптимизируемый функционал выглядит так:

$$J(\mathbf{w}) = \overline{e^2(k)} \rightarrow \min.$$

Квадрат ошибки равен

$$\begin{aligned} e^2(k) &= (d(k) - \mathbf{u}^T(k)\mathbf{w})^2 = \\ &= d^2(k) - 2d(k)\mathbf{u}^T(k)\mathbf{w} + \mathbf{w}^T\mathbf{u}(k)\mathbf{u}^T(k)\mathbf{w}. \end{aligned}$$

Статистически усредняя это выражение, получаем следующее:

$$J(\mathbf{w}) = \overline{e^2(k)} = \overline{d^2(k)} - \overline{2d(k)\mathbf{u}^T(k)\mathbf{w}} + \overline{\mathbf{w}^T\mathbf{u}(k)\mathbf{u}^T(k)\mathbf{w}}. \quad (3.3)$$

Входящие в полученную формулу усредненные величины имеют следующий смысл:

- $\overline{d^2(k)} = \sigma_d^2$ — средний квадрат образцового сигнала;

- $\overline{d(k)\mathbf{u}^T(k)} = \mathbf{p}^T$ — транспонированный вектор-столбец \mathbf{p} взаимных корреляций между k -м отсчетом образцового сигнала и со-

держимым линией задержки фильтра. Если рассматриваемые случайные процессы $x(t)$ и $d(k)$ являются совместно стационарными, вектор взаимных корреляций не зависит от номера шага k ;

- $\mathbf{u}(k)\mathbf{u}^T(k) = \mathbf{R}$ — корреляционная матрица сигнала, имеющая размер $(N+1) \times (N+1)$. Для стационарного случайного процесса корреляционная матрица имеет вид матрицы Тейлора, то есть на ее диагоналях стоят одинаковые величины:

$$\mathbf{R} = \begin{bmatrix} R_x(0) & R_x(1) & R_x(2) & \dots & R_x(N) \\ R_x(1) & R_x(0) & R_x(1) & \dots & R_x(N-1) \\ R_x(2) & R_x(1) & R_x(0) & \dots & R_x(N-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_x(N) & R_x(N-1) & R_x(N-2) & \dots & R_x(0) \end{bmatrix},$$

здесь $R_x(m) = \overline{x(k)x(k-m)}$ — корреляционная функция (КФ) случайного процесса $\{x(k)\}$.

С учетом введенных обозначений (3.3) принимает следующий вид:

$$J(\mathbf{w}) = \sigma_d^2 - 2\mathbf{p}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}\mathbf{w}. \quad (3.4)$$

Данное выражение представляет собой квадратичную форму относительно \mathbf{w} и поэтому при невырожденной матрице \mathbf{R} имеет единственный минимум, для нахождения которого необходимо приравнять нулю вектор градиента:

$$\mathbf{grad} J(\mathbf{w}) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} = 0.$$

Отсюда получаем искомое решение для оптимальных коэффициентов фильтра:

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p}. \quad (3.5)$$

Такой фильтр называется *фильтром Винера*. Подстановка (3.5) в (3.4) дает минимально достижимую дисперсию сигнала ошибки:

$$\overline{e^2(k)}_{\min} = \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p}. \quad (3.6)$$

Несложно также показать, что $\overline{e(k)y(k)} = 0$ и $\overline{e(k)x(k)} = 0$, то есть что сигнал ошибки для фильтра Винера некоррелирован со входным и выходным сигналами фильтра.

В качестве примера рассчитаем с помощью MATLAB оптимальный фильтр для коррекции искажений, вносимых в передаваемый сигнал $x_0(k)$ каналом связи, имеющим четырехэлементную импульсную характеристику следующего вида: $\mathbf{h} = [-2, -4, 6, 3]$. Отсчеты передаваемого сигнала будем считать дискретным белым шумом с нулевым средним значением и единичной дисперсией. В этом случае корреляционная функция сигнала, прошедшего через канал связи, будет совпадать с корреляционной функцией импульсной характеристики канала: $\mathbf{r} = [65, 2, -24, -6]$. Корреляционная матрица входного сигнала строится как матрица Тейлора на основе данной корреляционной функции:

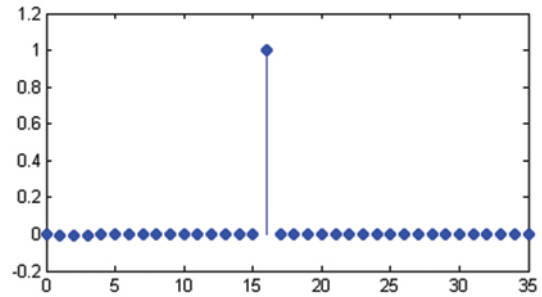
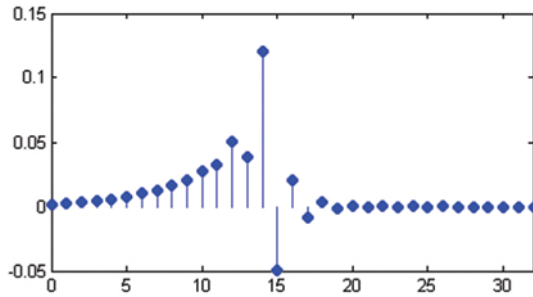


Рис. 2. Импульсные характеристики корректирующего фильтра (слева) и скорректированного канала связи (справа).

$$\mathbf{R} = \begin{bmatrix} 65 & 2 & -24 & -6 & 0 & \dots & 0 \\ 2 & 65 & 2 & -24 & -6 & \dots & 0 \\ -24 & 2 & 65 & 2 & -24 & \dots & 0 \\ -6 & -24 & 2 & 65 & 2 & \dots & 0 \\ 0 & -6 & -24 & 2 & 65 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 65 \end{bmatrix}$$

Восстановление переданного сигнала неизбежно требует внесения некоторой временной задержки, поэтому образцовый сигнал представляет собой задержанную копию переданного: $d(k) = x_0(k - \Delta k)$. В линии задержки фильтра на k -м шаге находятся отсчеты искаженного сигнала с номерами $k, k-1, k-2, \dots, k-N$, где N — порядок фильтра. Каждый из этих отсчетов представляет собой линейную комбинацию отсчетов переданного сигнала:

$$x(k) = \sum_{n=-\infty}^{\infty} x_0(n)h(k-n). \quad (3.7)$$

Поскольку исходный сигнал считается белым шумом, то при вычислении m -го элемента вектора \mathbf{p} результат усреднения будет отличен от нуля только для одного слагаемого (3.7):

$$\begin{aligned} p_m &= \overline{x(k-m)d(k)} = \\ &= \overline{\sum_{n=-\infty}^{\infty} x_0(n)h(k-m-n)x_0(k-\Delta k)} = \\ &= h(\Delta k - m). \end{aligned}$$

Таким образом, вектор \mathbf{p} содержит *перевернутую* импульсную характеристику канала, при необходимости обрезанную или дополненную нулями с одной или двух сторон.

Ниже приводится код MATLAB-программы, реализующей расчет оптимального фильтра:

```
h = [-2 -4 6 3]; % импульсная характеристика канала связи
N = 32; % порядок рассчитываемого фильтра
r = xcorr(h, N);
r = r(N+1:end); % односторонняя КФ импульсной характеристики канала
R = toeplitz(r); % корреляционная матрица искаженного сигнала
p = zeros(N+1, 1);
k0 = round((N-length(h))/2);
p(k0:k0+length(h)-1) = fliplr(h); % вектор взаимных корреляций
w = R \ p; % коэффициенты оптимального фильтра
subplot(1,2,1)
impz(w) % график импульсной характеристики рассчитанного фильтра
subplot(1,2,2)
impz(conv(h, w)) % график импульсной характеристики скорректированного канала
```

Код составлен так, чтобы можно было варьировать вид импульсной характеристики канала и порядок рассчитываемого фильтра.

На рис. 2 приведены графики импульсной характеристики корректирующего фильтра и сквозной импульсной характеристики скорректированного канала связи. Видно, что сквозная характеристика близка к единичному импульсу; это говорит о хорошем качестве коррекции. Столь качественная коррекция оказалась возможна потому, что частотная характеристика канала не имеет глубоких провалов.

4. Алгоритм LMS

Один из наиболее распространенных адаптивных алгоритмов основан на поиске минимума целевой функции (3.3) методом наискорейшего спуска. При использовании данного способа оптимизации вектор коэффициентов фильтра $\mathbf{w}(k)$ должен рекурсивно обновляться следующим образом:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \frac{\mu}{2} \text{grad}J(\mathbf{w}(k)) = \\ &= \mathbf{w}(k) + \mu \mathbf{p} - \mu \mathbf{R} \mathbf{w}(k), \end{aligned} \quad (4.1)$$

где μ — положительный коэффициент, называемый размером шага. Подробный анализ сходимости данного процесса приведен, например, в [4]. Показано, что алгоритм сходится, если $0 < \mu < 2/\lambda_{\max}$, где λ_{\max} — максимальное собственное число корреляционной матрицы \mathbf{R} . Скорость сходимости при этом зависит от разброса собственных чисел корреляционной матрицы \mathbf{R} — чем меньше отношение $\lambda_{\max}/\lambda_{\min}$, тем быстрее сходится итерационный процесс.

Однако для расчета градиента необходимо знать значения матрицы \mathbf{R} и вектора \mathbf{p} . На практике могут быть доступны лишь *оценки* этих значений, получаемые по входным данным. Простейшими такими оценками являются *мгновенные* значения корреляционной матрицы и вектора взаимных корреляций, получаемые без какого-либо усреднения:

$$\begin{aligned} \hat{\mathbf{R}}(k) &= \mathbf{u}(k)\mathbf{u}^T(k), \\ \hat{\mathbf{p}}(k) &= d(k)\mathbf{u}(k). \end{aligned}$$

При использовании данных оценок формула (4.1) принимает следующий вид:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu d(k)\mathbf{u}(k) - \\ &\quad - \mu \mathbf{u}(k)\mathbf{u}^T(k)\mathbf{w}(k) = \\ &= \mathbf{w}(k) + \mu \mathbf{u}(k)(d(k) - \mathbf{u}^T(k)\mathbf{w}(k)). \end{aligned} \quad (4.2)$$

Выражение, стоящее в скобках, согласно (3.2), представляет собой разность между образцовым сигналом и выходным сигналом фильтра на k -м шаге, то есть ошибку фильтрации $e(k)$. С учетом этого выражение для рекурсивного обновления коэффициентов фильтра оказывается очень простым:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k) \mathbf{u}(k). \quad (4.3)$$

Алгоритм адаптивной фильтрации, основанный на формуле (4.3), получил название LMS (Least Mean Square, метод наименьших квадратов). Можно получить ту же формулу и несколько иным образом: используя вместо градиента *статистически усредненного* квадрата ошибки $e^2(k)$ градиент его *мгновенного значения* $e^2(k)$.

Анализ сходимости алгоритма LMS показывает [4], что верхняя граница для размера шага μ в данном случае является меньшей, чем при использовании истинных значений градиента. Эта граница примерно равна

$$\mu_{\max} \approx \frac{2}{\sum_k \lambda_k} = \frac{2}{\text{trace}(\mathbf{R})} = \frac{2}{(N+1)\sigma_x^2}, \quad (4.4)$$

где λ_k — собственные числа корреляционной матрицы \mathbf{R} , а σ_x^2 — средний квадрат входного сигнала фильтра.

Основным достоинством алгоритма LMS является предельная вычислительная простота — для подстройки коэффициентов фильтра на каждом шаге нужно выполнить $N+1$ пар операций «умножение–сложение». Платой за простоту является медленная сходимость и повышенная (по сравнению с минимально достижимым значением (3.6)) дисперсия ошибки в установившемся режиме — коэффициенты фильтра всегда флуктуируют вокруг оптимальных значений (3.5), что и увеличивает уровень выходного шума.

Существует большое число модификаций алгоритма LMS [3], направленных на ускорение сходимости либо на уменьшение числа арифметических операций. Ускорение сходимости может быть достигнуто за счет улучшения используемой оценки градиента, а также за счет преобразования входного сигнала с целью сделать его отсчеты некоррелированными. Уменьшение вычислительной сложности может быть достигнуто, в частности, за счет использования в (4.3) не самих сигнала ошибки и содержимого линии задержки фильтра, а лишь их *знаков*. Это позволяет полностью избавиться от операций умножения при обновлении коэффициентов фильтра. В целом следует отметить, что требования ускорения сходимости и сокращения вычислительных затрат являются противоречивыми.

5. Детерминированная задача оптимизации

Рассматривая статистическую задачу оптимизации, мы считали входной сигнал *случайным процессом* и минимизировали

средний квадрат ошибки воспроизведения образцового сигнала. Однако возможен и иной подход, не использующий статистические методы.

Пусть, как и раньше, обработке подвергается последовательность, состоящая из K отсчетов $x(k)$, коэффициенты нерекурсивного фильтра образуют вектор-столбец \mathbf{w} , а отсчеты образцового сигнала равны $d(k)$. Выходной сигнал фильтра определяется формулой (3.1), а ошибка воспроизведения образцового сигнала — формулой (3.2). Теперь оптимизационная задача формулируется так: нужно отыскать такие коэффициенты фильтра \mathbf{w} , чтобы норма ошибки воспроизведения образцового сигнала была минимальной:

$$J(\mathbf{w}) = \sum_{k=0}^{K-1} |e(k)|^2 \rightarrow \min. \quad (5.1)$$

Для решения задачи в выражениях (3.1) и (3.2) необходимо перейти к матричной записи вдоль координаты k , получив формулы для векторов-столбцов выходного сигнала \mathbf{y} и для ошибки воспроизведения входного сигнала \mathbf{e} :

$$\begin{aligned} \mathbf{y} &= \mathbf{U}^T \mathbf{w}, \\ \mathbf{e} &= \mathbf{d} - \mathbf{U}^T \mathbf{w}. \end{aligned} \quad (5.2)$$

Здесь \mathbf{d} — вектор-столбец отсчетов образцового сигнала, а \mathbf{U} — матрица, столбцы которой представляют собой содержимое линии задержки фильтра на разных тактах:

$$\mathbf{U} = [\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(K-1)].$$

Выражение (5.1) для нормы ошибки можно переписать в матричном виде следующим образом:

$$J(\mathbf{w}) = \mathbf{e}^T \mathbf{e} \rightarrow \min. \quad (5.3)$$

Подставив (5.2) в (5.3), имеем

$$\begin{aligned} J(\mathbf{w}) &= (\mathbf{d} - \mathbf{U}^T \mathbf{w})^T (\mathbf{d} - \mathbf{U}^T \mathbf{w}) = \\ &= \mathbf{d}^T \mathbf{d} - \mathbf{w}^T \mathbf{U} \mathbf{d} - \mathbf{d}^T \mathbf{U}^T \mathbf{w} + \mathbf{w}^T \mathbf{U} \mathbf{U}^T \mathbf{w}. \end{aligned}$$

Для нахождения минимума необходимо вычислить градиент данного функционала и приравнять его нулю:

$$\mathbf{grad} J(\mathbf{w}) = -2\mathbf{U} \mathbf{d} + 2\mathbf{U} \mathbf{U}^T \mathbf{w} = 0.$$

Отсюда легко получается искомое оптимальное решение:

$$\mathbf{w} = (\mathbf{U} \mathbf{U}^T)^{-1} \mathbf{U} \mathbf{d}. \quad (5.4)$$

В формуле (5.4) прослеживается близкое родство с формулой (3.5), описывающей оптимальный в статистическом смысле фильтр Винера. Действительно, если учесть, что $(\mathbf{U} \mathbf{U}^T)^{-1} / K$ дает оценку корреляционной матрицы сигнала, полученную по одной реализации сигнала путем временного усреднения, а $\mathbf{U} \mathbf{d} / K$ является аналогичной оценкой взаимных корреляций между образцовым сигналом и содержимым линии задержки фильтра, то формулы (3.5) и (5.4) совпадут.

6. Алгоритм RLS

В принципе, в процессе приема сигнала можно на каждом очередном шаге пересчитать коэффициенты фильтра непосредственно по формуле (5.4), однако это связано с неоправданно большими вычислительными затратами. Действительно, размер матрицы \mathbf{U} постоянно увеличивается и, кроме того, необходимо каждый раз заново вычислять обратную матрицу $(\mathbf{U}\mathbf{U}^T)^{-1}$.

Сократить вычислительные затраты можно, если заметить, что на каждом шаге к матрице \mathbf{U} добавляется лишь один новый столбец, а к вектору \mathbf{d} — один новый элемент. Это дает возможность организовать вычисления *рекурсивно*. Соответствующий алгоритм называется *рекурсивным методом наименьших квадратов* (Recursive Least Square, RLS).

Подробный вывод формул, описывающих алгоритм RLS, можно найти, например, в [2, 4], здесь же приведем лишь базовые идеи. При использовании алгоритма RLS производится рекурсивное обновление оценки обратной корреляционной матрицы $\mathbf{P} = (\mathbf{U}\mathbf{U}^T)^{-1}$, а вывод формул основывается на следующем матричном тождестве:

$$(\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{D})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}, \quad (6.1)$$

где \mathbf{A} и \mathbf{C} — квадратные невырожденные матрицы (необязательно одинаковых размеров), а \mathbf{B} и \mathbf{D} — матрицы совместимых размеров.

Применение формулы (6.1) для рекурсивного обновления обратной корреляционной матрицы \mathbf{P} в сочетании с исходной формулой (5.4) для коэффициентов оптимального фильтра дает следующую последовательность шагов адаптивного алгоритма RLS.

1. При поступлении новых входных данных $\mathbf{u}(k)$ производится фильтрация сигнала с использованием текущих коэффициентов фильтра $\mathbf{w}(k-1)$ и вычисление величины ошибки воспроизведения образцового сигнала:

$$y(k) = \mathbf{u}^T(k)\mathbf{w}(k-1), \\ e(k) = d(k) - y(k).$$

2. Рассчитывается вектор-столбец *коэффициентов усиления* (следует отметить, что знаменатель дроби в следующих двух формулах является скаляром, а не матрицей):

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{u}(k)}{1 + \mathbf{u}^T(k)\mathbf{P}(k-1)\mathbf{u}(k)}. \quad (6.2)$$

3. Производится обновление оценки обратной корреляционной матрицы сигнала:

$$\mathbf{P}(k) = \mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{u}^T(k)\mathbf{P}(k-1). \quad (6.3)$$

4. Наконец, производится обновление коэффициентов фильтра:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{K}(k)e(k).$$

Начальное значение вектора \mathbf{w} обычно принимается нулевым, а в качестве исходной

оценки матрицы \mathbf{P} используется диагональная матрица вида $\mathbf{C}\mathbf{I}/\sigma_x^2$, где $C \gg 1$ (в [2] рекомендуется $C \geq 100$).

В формулах (5.1) и (5.3) значениям ошибки на всех временных тактах придается одинаковый вес. В результате, если статистические свойства входного сигнала со временем изменяются, это приводит к ухудшению качества фильтрации. Чтобы дать фильтру возможность отслеживать нестационарный входной сигнал, можно применить в (5.1) *экспоненциальное забывание*, при котором вес прошлых значений сигнала ошибки экспоненциально уменьшается:

$$J(\mathbf{w}) = \sum_{k=0}^{K-1} \lambda^{K-1-k} |e(k)|^2, \quad 0 < \lambda \leq 1.$$

При использовании экспоненциального забывания формулы (6.2) и (6.3) принимают следующий вид:

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{u}(k)}{\lambda + \mathbf{u}^T(k)\mathbf{P}(k-1)\mathbf{u}(k)},$$

$$\mathbf{P}(k) = \frac{1}{\lambda} (\mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{u}^T(k)\mathbf{P}(k-1)).$$

Главным достоинством алгоритма RLS является быстрая сходимость. Однако достигается это за счет значительно более высокой (по сравнению с алгоритмом LMS) вычислительной сложности. Согласно [2] при оптимальной организации вычислений для обновления коэффициентов фильтра на каждом такте требуется $(2.5N^2 + 4N)$ пар операций «умножение–сложение».

7. Алгоритм Калмана

Прежде чем рассматривать применение алгоритма Калмана для решения задачи адаптивной фильтрации, напомним формулировку задачи фильтрации случайного процесса с известными динамическими свойствами, для решения которой фильтр Калмана изначально предназначен. Цель фильтра Калмана — минимизировать дисперсию оценки векторного случайного процесса $\mathbf{x}(k)$, изменяющегося во времени следующим образом:

$$\mathbf{x}(k+1) = \mathbf{\Phi}(k)\mathbf{x}(k) + \mathbf{v}(k),$$

где $\mathbf{\Phi}(k)$ — матрица перехода, $\mathbf{v}(k)$ — случайный вектор (*шум процесса*), имеющий нормальное распределение с корреляционной матрицей $\mathbf{Q}_p(k)$.

Для наблюдения доступен линейно преобразованный процесс $\mathbf{y}(k)$, к которому добавляется шум наблюдения:

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k),$$

где $\mathbf{H}(k)$ — матрица наблюдения, $\mathbf{w}(k)$ — шум наблюдения, представляющий собой случайный вектор, имеющий нормальное распределение с корреляционной матрицей $\mathbf{Q}_m(k)$.

Поиск алгоритма для рекурсивного обновления оценки процесса $\hat{\mathbf{x}}(k)$ дает следующую последовательность формул:

• $\hat{\mathbf{y}}(k) = \mathbf{C}(k)\mathbf{\Phi}(k)\hat{\mathbf{x}}(k-1)$ — прогнозируемое значение наблюдаемого сигнала;

• $e(k) = y(k) - \hat{y}(k)$ — невязка между прогнозируемым и реально наблюдаемым значениями;

• $\mathbf{K}(k) = \mathbf{P}(k-1)\mathbf{C}^T(k) \times (\mathbf{C}(k)\mathbf{P}(k-1)\mathbf{C}^T(k) + \mathbf{Q}_M(k))^{-1}$ — калмановский коэффициент усиления;

• $\hat{\mathbf{x}}(k) = \mathbf{\Phi}(k)\hat{\mathbf{x}}(k-1) + \mathbf{K}(k)\mathbf{e}(k)$ — обновление оценки процесса $\mathbf{x}(k)$;

• $\mathbf{P}(k) = \mathbf{\Phi}(k)[\mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{C}(k)\mathbf{P}(k-1)]\mathbf{\Phi}^T(k) + \mathbf{Q}_M(k)$ — обновление оценки корреляционной матрицы ошибок фильтрации.

При использовании фильтра Калмана для решения задачи адаптивной фильтрации отслеживаемым процессом является *вектор коэффициентов оптимального фильтра* \mathbf{w} . Предполагается, что детерминированных изменений коэффициентов не происходит, поэтому матрица перехода $\mathbf{\Phi}$ является единичной: $\mathbf{\Phi}(k) = \mathbf{I}$. В качестве матрицы наблюдения выступает вектор содержимого линии задержки фильтра $\mathbf{u}(k)$. Таким образом, выходной сигнал фильтра представляет собой прогнозируемое значение наблюдаемого сигнала, а в качестве самого наблюдаемого сигнала выступает образцовый сигнал адаптивного фильтра $d(k)$. Шум наблюдения в данном случае является ошибкой воспроизведения образцового сигнала, а матрица \mathbf{Q}_M превращается в скалярный параметр — средний квадрат сигнала ошибки. Как отмечается в [4], величина этого параметра слабо влияет на поведение алгоритма. Рекомендуемые значения — $(0.001...0.01)\sigma_d^2$.

Если фильтруется стационарный случайный процесс, коэффициенты оптимального фильтра являются постоянными и можно принять $\mathbf{Q}_p = 0$. Чтобы дать фильтру возможность отслеживать медленные изменения статистики входного сигнала, в качестве \mathbf{Q}_p может использоваться диагональная матрица.

В результате приведенные выше формулы принимают следующий вид:

• $y(k) = \mathbf{u}^T(k)\hat{\mathbf{w}}(k-1)$ — выходной сигнал фильтра (прогнозируемое значение образцового сигнала);

• $e(k) = d(k) - y(k)$ — ошибка фильтрации;

• $\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{u}(k)}{\mathbf{u}^T(k)\mathbf{P}(k-1)\mathbf{u}(k) + \mathbf{Q}_M}$ — калмановский коэффициент усиления;

• $\hat{\mathbf{w}}(k) = \hat{\mathbf{w}}(k-1) + \mathbf{K}(k)e(k)$ — обновление оценки коэффициентов фильтра $\mathbf{w}(k)$;

• $\mathbf{P}(k) = \mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{u}^T(k)\mathbf{P}(k-1) + \mathbf{Q}_p$ — обновление оценки корреляционной матрицы ошибок оценивания.

Начальное значение вектора \mathbf{w} обычно принимается нулевым, а в качестве исходной оценки матрицы \mathbf{P} используется диагональная матрица вида $\mathbf{C}\mathbf{I}$.

Сравнивая формулы, описывающие алгоритмы RLS и Калмана, легко заметить их сходство. Вычисли-

тельная сложность и качественные параметры двух алгоритмов также оказываются весьма близкими. Разница заключается лишь в исходных посылах, использовавшихся при выводе формул, и в трактовке параметров алгоритмов. В некоторых источниках (см., например, [5]) алгоритмы RLS и Калмана применительно к адаптивной фильтрации отождествляются.

8. Применение адаптивных фильтров

Адаптивные фильтры в настоящее время нашли применение во многих радиотехнических и телекоммуникационных системах. В данном разделе кратко рассматриваются лишь некоторые из возможных областей их использования.

8.1. Идентификация систем

Все способы использования адаптивных фильтров так или иначе сводятся к решению задачи *идентификации*, то есть определения характеристик некоторой системы. Возможны два варианта идентификации — *прямая* и *обратная*. В первом случае адаптивный фильтр включается *параллельно* с исследуемой системой (рис. 3, а). Входной сигнал является общим для исследуемой системы и адаптивного фильтра, а выходной сигнал системы служит для адаптивного фильтра образцовым сигналом. В процессе адаптации временные и частотные характеристики фильтра будут стремиться к соответствующим характеристикам исследуемой системы.

При обратной идентификации адаптивный фильтр включается *последовательно* с исследуемой системой (рис. 3, б). Выходной сигнал системы поступает на вход адаптивного фильтра, а входной сигнал системы является образцом для адаптивного фильтра. Таким образом, фильтр стремится компенсировать влияние системы и восстановить исходный сигнал, устранив внесенные системой искажения.

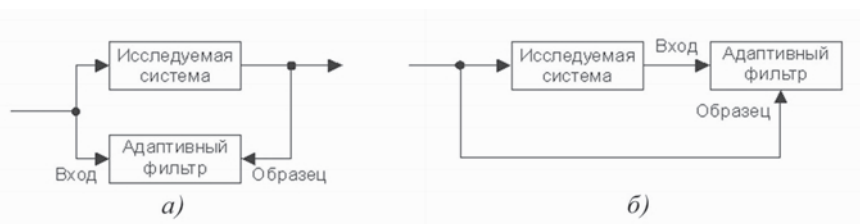
В MATLAB прямая идентификация нестационарной системы с помощью адаптивного фильтра производится в демонстрационной программе **adaptkalmndemo** из пакета расширения Filter Design, а также в модели **kalmnsce** из набора блоков DSP Blockset среды моделирования Simulink.

Теперь перейдем от обобщенных схем к рассмотрению более конкретных вариантов.

8.2. Подавление шума

Пусть необходимо обеспечить пилота самолета или, скажем, водителя трактора системой речевой связи. При этом воспринимае-

■ **Рис. 3.** Идентификация систем с помощью адаптивного фильтра: а — прямая, б — обратная.



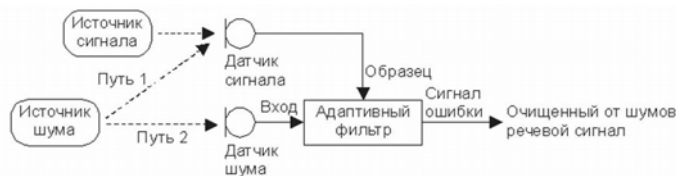


Рис. 4. Подавление шума с помощью адаптивного фильтра.

мый микрофоном речевой сигнал неизбежно окажется сильно зашумленным звуками работающего двигателя и т. п. Избавиться от этих шумов нельзя, но можно получить *образец* шумового сигнала, установив второй микрофон в непосредственной близости от двигателя (или иного источника шумов). Разумеется, этот шум нельзя просто вычесть из речевого сигнала, поскольку по дороге до двух микрофонов шум следует *разными путями* и, следовательно, претерпевает *разные искажения* (рис. 4). Однако шумовые случайные процессы, воспринимаемые двумя микрофонами, будут *коррелированными*, так как они происходят из общего источника. В то же время очевидно, что шумовой сигнал не коррелирован с полезным речевым сигналом.

С помощью адаптивного фильтра в данном случае решается задача прямой идентификации (см. рис. 3, а) преобразований шума на пути от шумового микрофона к сигнальному. Входным сигналом адаптивного фильтра является шумовой сигнал от дополнительного микрофона, а в качестве образцового сигнала используется сигнально-шумовая смесь, воспринимаемая основным микрофоном.

Адаптивный фильтр стремится преобразовать входной сигнал так, чтобы сделать его как можно ближе к образцовому. Поскольку со входным сигналом фильтра коррелирована лишь шумовая составляющая образцового сигнала, в установившемся режиме на выходе фильтра будет получаться оценка шума, присутствующего в образцовом сигнале. Сигнал ошибки, рассчитываемый как разность между образцовым сигналом и выходным сигналом адаптивного фильтра, будет в этом случае представлять собой очищенный от шума речевой сигнал.

Данному варианту использования адаптивных фильтров посвящены несколько демонстрационных программ, имеющихся в составе MATLAB. Это программа **adaptrlsdemo** из пакета расширения Filter Design, а также модели **lmsdemo** и **rlsdemo** из набора блоков DSP Blockset среды моделирования Simulink.

8.3. Выравнивание канала связи

При передаче по каналу связи информационный сигнал неизбежно претерпевает некоторые искажения. В системах цифровой связи эти искажения могут привести к возникновению ошибок при приеме цифровых данных. Для снижения вероятности ошибок необходимо компенсировать влияние канала связи, то есть решить задачу обратной идентификации (см. рис. 3, б). В частотной области компенсация вносимых каналом искажений означает *выравнивание* (equalization) его частотной характеристики, по-

этому фильтры, выполняющие такое выравнивание, получили название *эквалайзеров* (equalizer).

При использовании адаптивного фильтра в качестве эквалайзера возникает проблема получения образцового сигнала. Эта проблема решается путем передачи специального настроечного сигнала перед началом передачи данных. В качестве такого настроечного сигнала обычно используется псевдослучайная последовательность символов. Алгоритм формирования этого сигнала известен приемной стороне, поэтому образцовый сигнал может быть сгенерирован автономно и использован для обучения адаптивного фильтра. Этот режим работы называется *режимом обучения* (training mode) (рис. 5).

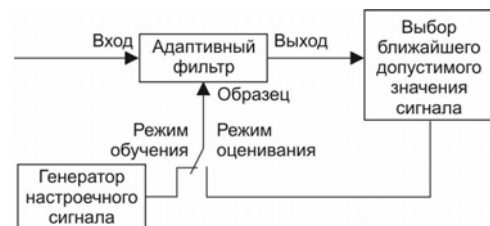


Рис. 5. Выравнивание канала связи с помощью адаптивного фильтра.

После окончания настроечного сигнала начинается собственно передача данных. Приемник при этом переключается в другой режим, называемый *режимом оценивания* (decision-directed mode). В этом режиме для получения образцового сигнала используется тот факт, что множество возможных значений сигнала в системе цифровой связи является конечным. После приема очередного временного такта ищется ближайшее к принятому сигналу допустимое значение. Оно используется в качестве образцового сигнала, а разность между этим значением и принятым сигналом дает сигнал ошибки, используемый для адаптации. На рис. 6 это иллюстрируется применительно к 16-позиционной квадратурной манипуляции.

Если после настройки эквалайзера, произведенной в режиме обучения, уровень шумов на выходе фильтра такой, что ближайшая допустимая точка в большинстве случаев оказывается правильной (то есть если вероятность ошибки мала), адаптивный алгоритм сохраняет стабильность.

В некоторых, в частности, многопользовательских системах связи передавать настроечный сигнал не представляется возможным. В этом случае может использоваться *слепая* (blind) адаптация, рассмотрение ко-

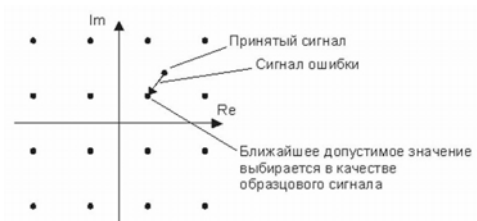


Рис. 6. Формирование образцового сигнала и сигнала ошибки в режиме оценивания.

торой выходит за рамки данной статьи. Некоторые алгоритмы слепой адаптации приведены в [4, 5].

Адаптивному выравниванию канала связи посвящена демонстрационная модель **Insadeq** из набора блоков DSP Blockset среды моделирования Simulink.

8.4. Эхоподавление

Данная технология, так же как и выравнивание канала связи, широко используется в современных модемах. Скоростные модемы для телефонных линий связи работают в *дуплексном режиме*, то есть передают и принимают данные одновременно, при этом для передачи и приема используется одна и та же полоса частот. Однако сигнал собственного передатчика в данном случае неизбежно просачивается в приемник, мешая работе последнего. Просачивающийся сигнал может распространяться разными путями, приобретая при этом неизвестные заранее искажения.

Подавить эхо-сигнал можно с помощью адаптивного фильтра. При этом решается задача прямой идентификации тракта распространения эха (рис. 7). На вход адаптивного фильтра поступает сигнал передатчика модема, а в качестве образцового сигнала используется принимаемый сигнал, содержащий эхо. Адаптивный фильтр формирует оценку эхо-сигнала, а сигнал ошибки представляет собой принимаемый сигнал, очищенный от эха.

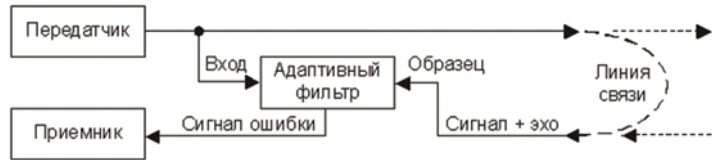
Для правильной работы системы эхоподавления необходимо, чтобы передаваемый и принимаемый сигналы были некоррелированы. Поэтому входные данные, поступающие в модем для передачи, прежде всего подвергаются *скремблированию* (scrambling), то есть преобразуются в псевдослучайный битовый поток. При этом два взаимодействующих модема используют *разные* скремблеры, что и обеспечивает некоррелированность.

Эхоподавление, осуществляемое согласно схеме рис. 7, используется во всех современных модемах.

9. Функции адаптивной фильтрации пакета Filter Design

В пакете расширения Filter Design, начиная с версии 2.1, входящей в поставку MATLAB 6.1, содержатся функции, реализующие следующие алгоритмы адаптивной фильтрации:

- **adaptlms** — алгоритм LMS в классическом виде;
- **adaptnlms** — нормированный алгоритм LMS (расчет шага m на каждом такте осуществляется автоматически по формуле (4.4), исходя из энергии фрагмента сигнала, содержащегося в линии задержки фильтра);
- **adaptsd** — алгоритм LMS, для адаптации используется только знак данных, содержащихся в линии задержки фильтра (sign-data);
- **adaptse** — алгоритм LMS, для адаптации используется только знак сигнала ошибки (sign-error);
- **adaptss** — алгоритм LMS, для адаптации используются только знаки сигнала ошибки



и данных, содержащихся в линии задержки фильтра (sign-sign);

- **adaptlms** — алгоритм RLS, в том числе с экспоненциальным забыванием данных;
- **adaptkalman** — алгоритм Калмана.

Все шесть функций имеют одинаковый синтаксис вызова:

```
[y, e, s] = adaptlms(x, d, s);
[y, e, s] = adaptnlms(x, d, s);
[y, e, s] = adaptsd(x, d, s);
[y, e, s] = adaptse(x, d, s);
[y, e, s] = adaptss(x, d, s);
[y, e, s] = adaptrls(x, d, s);
[y, e, s] = adaptkalman(x, d, s);
```

Здесь \mathbf{x} — вектор отсчетов входного сигнала, \mathbf{d} — вектор отсчетов образцового сигнала, \mathbf{s} — структура с параметрами алгоритма и начальным состоянием фильтра (см. далее). Три возвращаемых результата имеют следующий смысл: \mathbf{y} — вектор отсчетов выходного сигнала, \mathbf{e} — вектор отсчетов сигнала ошибки, \mathbf{s} — результирующие значения параметров и конечное состояние фильтра.

Структура \mathbf{s} хранит текущее состояние фильтра и содержит параметры, набор которых зависит от конкретного алгоритма. Назначение полей структуры и соответствие входных параметров *init*-функций полям возвращаемой структуры \mathbf{s} приведено в табл. 1. Доступ к начальному и конечному состояниям фильтра позволяет организовать блочную или даже поотсчетную обработку исходных данных.

В качестве примера реализуем адаптивную коррекцию искажений, вносимых в сигнал тем же каналом связи, что использовался в примере расчета фильтра Винера. При этом сравним качество работы четырех версий алгоритма LMS — классического варианта и трех вариантов с использованием знаковых преобразований. Чтобы более наглядно показать качество компенсации, в виде входного сигнала используем четырехуровневый цифровой сигнал, с равной вероятностью принимающий значения $-3, -1, 1$ и 3 . Код MATLAB-программы приведен ниже.

```
h = [-2 -4 6 3]; % импульсная характеристика канала связи
% генерация сигнала
NX = 4000; % число отсчетов
levels = [-3 -1 1 3]; % возможные уровни исходного сигнала
x = randsrc(NX, 1, levels); % случайный сигнал
y = conv(x, h); % искаженный сигнал
N = 32; % порядок адаптивного фильтра
mu = 1/var(levels, 1)/sum(h.^2)/(N+1); % размер шага для LMS
s_lms = initlms(zeros(N+1,1), mu);
s_lms_se = initse(zeros(N+1,1), mu/10);
s_lms_sd = initsd(zeros(N+1,1), mu);
s_lms_ss = initss(zeros(N+1,1), mu);
y = y(round(N/2)+1:end); % задержка фильтрации
[x, y] = eqtflength(x, y); % выравнивание длин векторов x и y
```

▲ Рис. 7. Эхоподавление, осуществляемое с помощью адаптивного фильтра.


```
% адаптивная фильтрация
[x1_lms, e_lms, s_lms] = adaptlms(y(1:1000), x(1:1000), s_lms);
[x1_lms_se, e_lms_se, s_lms_se] = adaptse(y, x, s_lms_se);
[x1_lms_sd, e_lms_sd, s_lms_sd] = adaptsd(y, x, s_lms_sd);
[x1_lms_ss, e_lms_ss, s_lms_ss] = adaptss(y, x, s_lms_ss);
% вывод графиков
subplot(4,2,1), plot(e_lms), title("LMS")
subplot(4,2,2), plot(x1_lms, '.')
subplot(4,2,3), plot(e_lms_se), title("LMS - sign error")
subplot(4,2,4), plot(x1_lms_se, '.')
subplot(4,2,5), plot(e_lms_sd), title("LMS - sign data")
subplot(4,2,6), plot(x1_lms_sd, '.')
subplot(4,2,7), plot(e_lms_ss), title("LMS - sign sign")
subplot(4,2,8), plot(x1_lms_ss, '.')
```

На рис. 8 показаны результаты работы программы. В левом столбце — зависимость сигнала ошибки от номера шага, в правом — график выходного сигнала фильтра. Видно, что использование знаковых преобразований замедляет сходимость LMS-алгоритма (пришлось даже увеличить число отображаемых итераций с 1000 до 4000). В наименьшей сте-

пени сходимость замедляется при знаковом преобразовании сигнала ошибки (sign-error), в наибольшей — при одновременном знаковом преобразовании сигнала ошибки и содержимого линии задержки фильтра (sign-sign). Кроме того, предельное значение μ для знаковых вариантов сильно зависит от масштаба уровней образцового и входного сигналов. Так, в данном примере пришлось в 10 раз уменьшить значение μ для варианта со знаковым преобразованием сигнала ошибки.

10. Блоки адаптивной фильтрации из набора DSP Blockset

В состав библиотеки **Filtering/Adaptive Filters** набора блоков DSP Blockset среды моделирования Simulink входят три блока адаптивных фильтров:

- **nLMS** — алгоритм LMS, в том числе нормированный;
- **RLS** — алгоритм RLS, в том числе с экспоненциальным забыванием данных;
- **Kalman** — алгоритм Калмана.

Каждый из блоков по умолчанию имеет два входа (**In** — входной сигнал, **Error** — сигнал ошибки) и два выхода (**Out** — выходной сигнал, **Taps** — вектор коэффициентов фильтра). В отличие от функций пакета Filter Design вместо образцового сигнала на вход блока подается *сигнал ошибки*, который должен формироваться внешними цепями. Это позволяет использовать блоки более гибко.

Параметры блоков в целом аналогичны полям структуры **s** в функциях пакета Filter Design, основные отличия заключаются в следующем:

- порядок фильтра задается явно — в виде числовой константы;
- для LMS не поддерживается вариант с утечкой, а также варианты со знаковым преобразованием содержимого линии задержки фильтра (знаковое преобразование сигнала ошибки легко произвести во внешней цепи);
- начальные значения обратной корреляционной матрицы в алгоритме RLS и корреляционной матрицы сигнала ошибки в алгоритме Калмана считаются диагональными и задаются путем указания скалярного параметра;
- в алгоритме Калмана корреляционная матрица шума процесса считается диагональной и задается в виде скалярного параметра — дисперсии.

Среди параметров всех блоков имеется флажок **Adapt input**. При его установке в блоке появляется третий вход **Adapt**. Этот вход позволяет включать и выключать адаптацию во время работы модели. При отключенной адаптации блок работает как фильтр с постоянными параметрами.

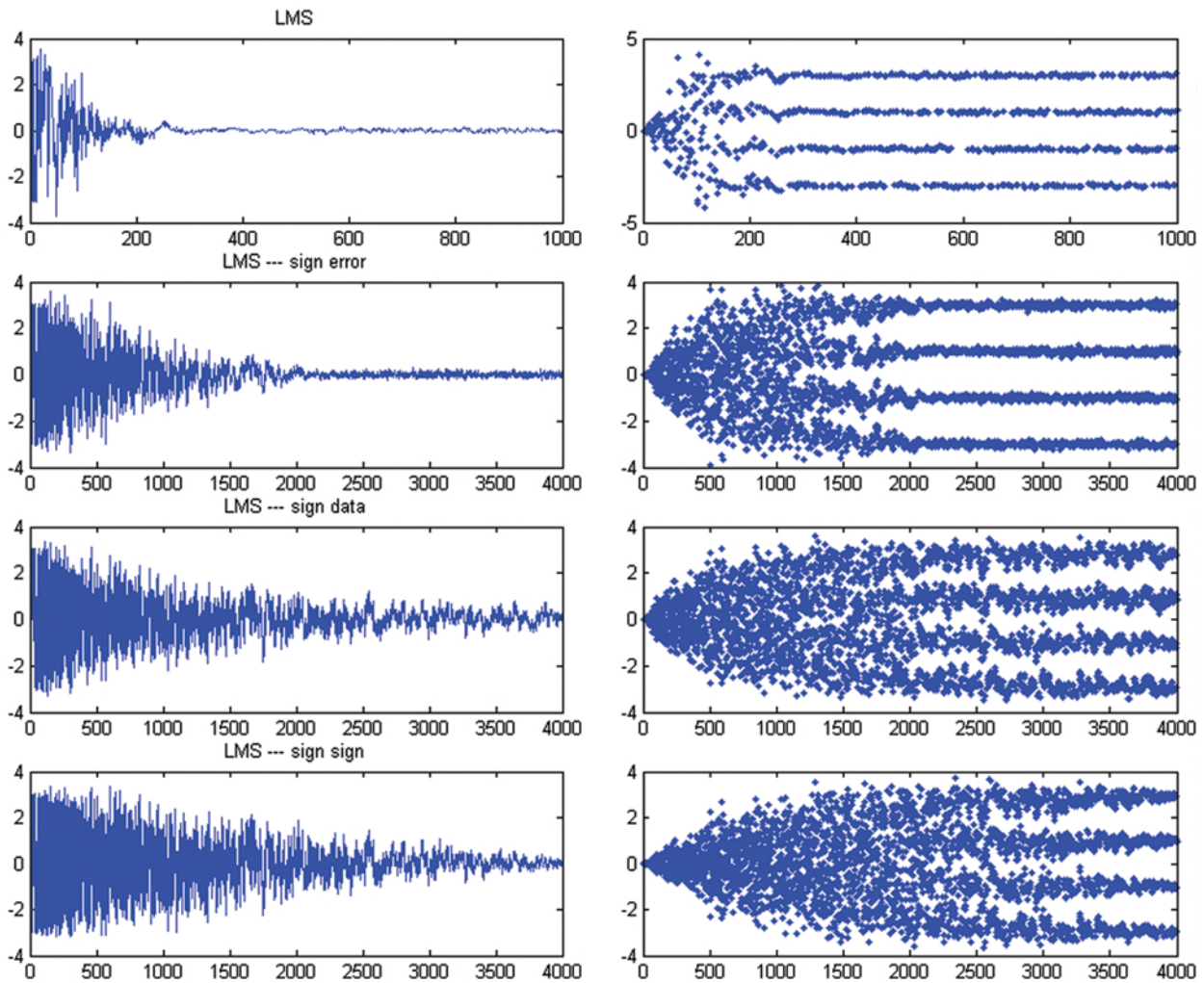
В качестве примера на рис. 9 показана модель, позволяющая сравнить качество выравнивания канала связи адаптивными фильтрами, использующими алгоритмы LMS, RLS и Калмана. Исходный сигнал и канал связи — те же, что и в примере использования функ-

Таблица 1

Назначение полей структуры **s**

Поле структуры	Параметр функции init*	Назначение	Имеет смысл для
coeffs	w0	Вектор коэффициентов фильтра. По длине этого вектора определяется порядок фильтра	Всех алгоритмов
states	zi	Вектор содержимого линии задержки фильтра	Всех алгоритмов
iter	(только для чтения)	Число выполненных итераций	Всех алгоритмов
step	mu	Размер шага μ	Всех разновидностей LMS
leakage	lf	Коэффициент утечки (в диапазоне 0...1, по умолчанию равен 1)	Всех разновидностей LMS
offset	offset	Константа, добавляемая к знаменателю формулы (4.4) для ограничения роста μ при малом уровне сигнала	Нормированного LMS
invcov	p0	Оценка обратной корреляционной матрицы входного сигнала	RLS
lambda	lambda	Коэффициент забывания (в диапазоне 0...1)	RLS
alg	alg	Вариант RLS-алгоритма: 'direct' - стандартный, 'sqrt' - обладающий большей численной устойчивостью вариант с использованием QR-разложения матриц	RLS
gain	(только для чтения)	Рассчитанный на последней выполненной итерации вектор коэффициентов усиления	RLS и алгоритма Калмана
errcov	k0	Оценка корреляционной матрицы ошибок отслеживания вектора коэффициентов фильтра	алгоритма Калмана
measvar	qm	Дисперсия шума наблюдения, то есть ожидаемая дисперсия сигнала ошибки	алгоритма Калмана
procov	qp	Корреляционная матрица ожидаемых флуктуаций коэффициентов фильтра	алгоритма Калмана

Заполнение полей структуры **s** может быть произведено вручную или с помощью функций **init***, где * — имя адаптивного алгоритма:
 s = initlms(w0, mu, zi, lf);
 s = initnlms(w0, mu, zi, lf, offset);
 s = initsd(w0, mu, zi, lf);
 s = initse(w0, mu, zi, lf);
 s = initss(w0, mu, zi, lf);
 s = initrls(w0, p0, lambda, zi, alg);
 s = initkalman(w0, k0, qm, qp, zi);



ций пакета Filter Design.

Трехканальный осциллограф (рис. 10), демонстрирующий сигнал ошибки для трех алгоритмов адаптации, показывает, что LMS-алгоритм сходится медленнее и дает большие остаточные шумы, в то время как алгоритмы RLS и Калмана обеспечивают сходные показатели качества.

11. Заключение

Функции пакета Filter Design и блоки набора DSP Blockset позволяют реализовать три наиболее популярных алгоритма адаптивной фильтрации и могут использоваться при расчете и моделировании разнообразных систем передачи информации. Их возможности достаточны для использования в ознакомительных и учебных целях. Однако при попытке применить функции и блоки для профессионального моделирования реальных телекоммуникационных систем выявляется ряд недостатков и ограничений, присущих их текущей реализации:

- все функции адаптивной фильтрации полностью написаны на языке MATLAB в виде цикла по отсчетам сигналов, поэтому скорость их работы невелика. Это особенно относится к более сложным в вычислительном отношении алгоритмам Калмана и RLS;
- как показывает синтаксис вызова фун-

кций, образцовый сигнал должен быть известен заранее, что делает невозможным применение данных функций в режиме оценивания (см. раздел 8.3 и рис. 5, 6);

• функции и блоки на каждом временном такте реализуют сдвиг данных в линии задержки фильтра на один отсчет. Это затрудняет применение функций в тех ситуациях, когда данные обновляются иным образом (сюда относятся, в частности, *дробные* (fractionally-spaced) эквалайзеры [2, 5]);

▲ Рис. 8. Пример использования функций адаптивной фильтрации из пакета Filter Design.

■ Рис. 9. Пример использования блоков адаптивных фильтров из набора DSP Blockset.

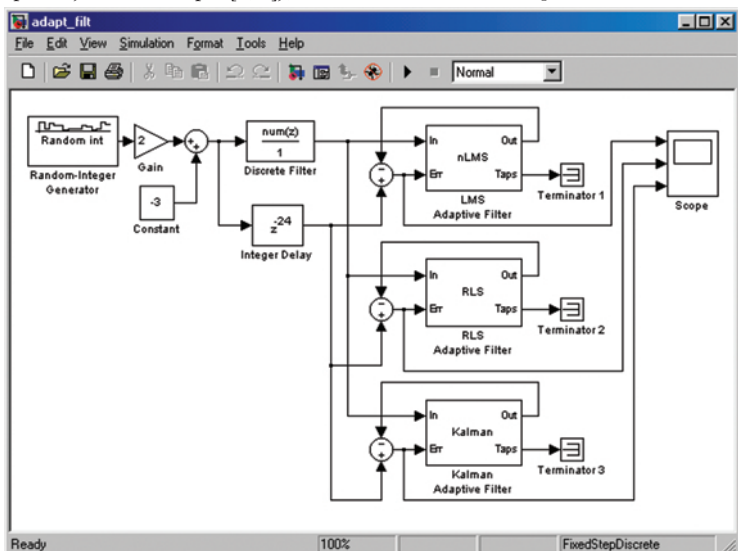
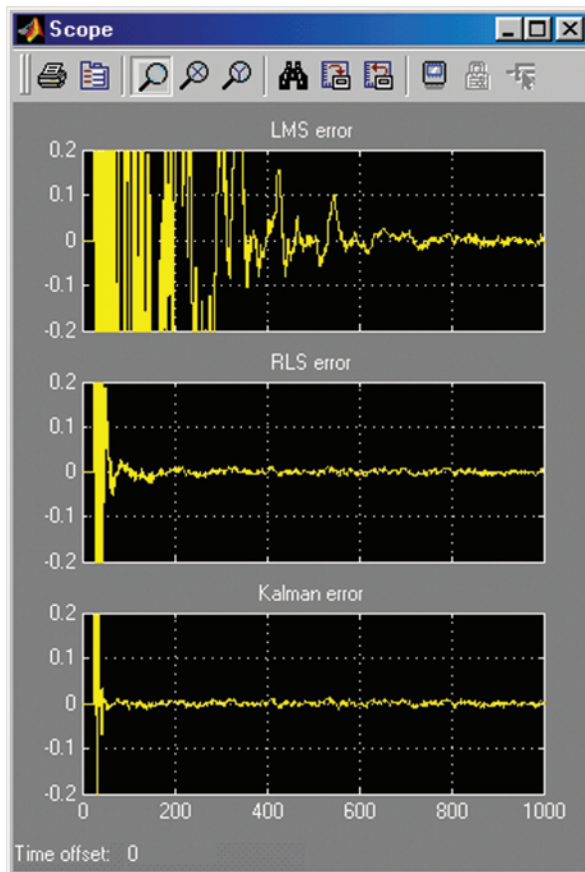


Рис. 10. Сигнал ошибки для трех алгоритмов адаптации.



• в пакете MATLAB и библиотеках Simulink, ориентированных на обработку сигналов, отсутствует классическая реализация алгоритма Калмана, предназначенная для оценивания значений случайного процесса.

Хочется надеяться, что в будущих версиях компонентов MATLAB разработчики устранят перечисленные недостатки.

Литература

1. Уидроу Б., Стирнз С. Д. Адаптивная обработка сигналов. — М.: Радио и связь, 1989. — 440 с.
2. Адаптивные фильтры: Пер. с англ. / Под ред. К. Ф. Н. Коуэна и П. М. Гранта. — М.: Мир, 1988. — 392 с.
3. Glentis G.-O., Berberidis K., Theodoridis S. Efficient Least Squares Adaptive Algorithms for FIR Transversal Filtering // IEEE Signal Processing Magazine. — 1999. — V. 16. — N. 4. — P. 13–41.
4. Haykin S. Adaptive Filter Theory, 4th edition. — Prentice Hall, 2002. — 936 p.
5. Прокис Дж. Цифровая связь: Пер. с англ. / Под ред. Д. Д. Кловского. — М.: Радио и связь, 2000. — 800 с.

Новости портала Exponenta.ru

С 1 сентября 2002 года на образовательном математическом сайте Exponenta.ru бесплатно предоставляются домены третьего уровня и хостинг авторам некоммерческих сайтов естественнонаучной тематики и сайтов о научном программном обеспечении. Цель проекта — собрать вокруг Exponenta.ru действительно интересные для научной и образовательной аудитории авторские сайты, которые сейчас разбросаны по разным уголкам российского Интернета.

Сейчас размещено уже 11 авторских сайтов, в частности:

<http://iglin.exponenta.ru/> — сайт содержит интерактивные учебные пособия по различным математическим и техническим курсам (задачи линейного программирования, обработка массива данных, вариационное исчисление, сопротивление материалов). Часть этих пособий — исполняемые файлы, которые можно переписать из архива и работать с ними на локальном компьютере. Другая часть — активные Web-страницы, загружаемые через Интернет. Они предназначены для совместной работы с системой MATLAB, установленной на компьютере пользователя. Материалы сайта могут быть полезны преподавателям для использования в учебном процессе, студентам для подготовки к экзаменам и всем пользователям MATLAB. Автор сайта — С. П. Иглин.

<http://ukrgap.exponenta.ru/> — сайт Украинской группы пользователей GAP (Groups, Algorithms and Programming). GAP — свободно распространяемая система компьютерной алгебры, первоначально созданная для исследований по комбинаторной теории групп, а к настоящему времени охватывающая и многие другие разделы алгебры. На страницах ресурса, помимо дистрибутива, размещены описание системы, обзор ее возможностей, различные материалы по работе с GAP, ссылки на ресурсы по GAP, TeX, Linux, адреса сайтов математических журналов, институтов, издательств. Работает рассылка новостей. Автор сайта — А. Б. Коновалов, председатель Украинской группы пользователей GAP.

<http://zmn.exponenta.ru/> — сайт, посвященный системе TeX. На сайте для свободного использования размещен собранный автором дистрибутив системы TeX под платформы MSDOS, Win3.1, OS/2, Win9*, WinME, включающий русифицированные Plain-TeX, AMS-TeX, LaTeX 2.09, LaTeX 2e. Весь комплект умещается на 8 дискетах и снабжен документацией и дополнительными стилевыми пакетами. На сайте приведено подробное описание процесса установки и использования данной версии TeX. Автор сайта — М. Н. Зарипов.

М. В. Бессонова