Recursive least-squares sequence estimation

by F. Gozzo

A family of adaptive communication receivers based on recursive least-squares sequence estimation (RLSSE) algorithms is proposed which provides performance comparable to that of conventional linear receivers, but with reduced complexity and less sensitivity to channel mismatch. A software-implemented version of the linear member of the family is shown to have performance equivalent to that of standard transversal equalizers under ideal conditions, yet offers a drastic improvement in white Gaussian noise mismatch environments. An analogous performance improvement for several test channels is also shown for software-implemented versions of the constrained (nonlinear) members of the family over decision feedback equalizers. Another advantage of the RLSSE family of receivers may be its ease of implementation, since it should be possible to combine the functions of channel estimation and sequence estimation on the same chip.

Introduction

Inherent in every communication system are channels which link the transmitter and receiver. These channels include telephone lines used in voice and modem applications, underwater channels used in acoustic applications, read/write channels used in magnetic storage devices, and atmospheric channels used in radar, satellite, and other wireless communication systems. Although their physical media and propagation characteristics vary greatly, these channels typically share three fundamental problems which plague the majority of high-speed communication systems: intersymbol interference, noise, and channel mismatch.

- *Intersymbol interference*

The majority of practical communication systems are adequately modeled as linear systems in which the received signal represents the convolution of the transmitted sequence with the channel impulse response. Because of several elements, including faster-than-Nyquist signaling and band-limited channels, the time dispersion created by this convolution causes a received pulse to be spread in time. Hence, two (or more) adjacent symbols can interfere with one another; this can be intuitively understood by analogy with echoes over telephone lines. The phenomenon, which is commonly known as intersymbol interference (ISI), may cause severe degradation in system performance unless the receiver can unravel or *deconvolve* the received signal.

There are many well-known receivers which mitigate the effects of ISI. These make use of schemes that range from high-complexity algorithms based on maximum-likelihood sequence estimation (MLSE) to low-complexity linear and decision feedback equalizers (DFEs). The latter have been and continue to be the backbone of modern receivers

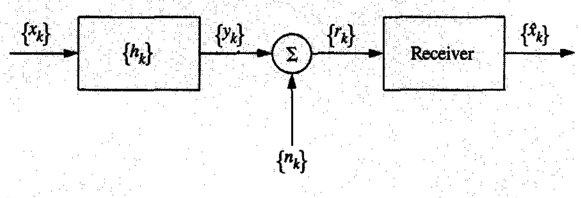

Figure 1

Discrete-time equivalent communication system model.

[1–3]. For all these schemes, though, mitigating their complexity and/or suboptimum nature, particularly in *real-time* applications where low complexity is crucial, remains an open challenge.

• *Noise*

Although the performance loss due to ISI is typically the predominant factor in higher-data-rate systems [1], noise remains a key contributor to performance degradation. Of course, there are many types of random disturbances, such as thermal noise attributed to a receiver's front end, active jammers found in military systems, and impulsive noise due to switching. However, since thermal noise plagues all practical receivers, one cannot ignore this common problem.

By itself, thermal noise can cause appreciable performance degradation; coupled with severe ISI, as found in channels with spectral nulls, noise can be crippling, particularly to linear receivers which more or less invert the channel and thus enhance the noise. Although nonlinear methods have been sought in these cases, the DFE is perhaps the only nonlinear receiver which can satisfy a low-complexity constraint while adequately treating these channels. Thus, expanding the number of low-complexity alternatives to treat noisy channels with severe ISI is still a key area of research.

• *Channel mismatch*

Generally, a receiver must have at least implicit information regarding the spectral characteristics of the channel as well as the statistics of the corrupting noise. Unfortunately, these quantities are often unknown and possibly time-varying. Therefore, a receiver must continuously track the changing channel and noise statistics. If the receiver is not able to provide this feature accurately, the receiver is said to operate in the presence of *channel mismatch*. Under ideal training conditions and with sufficient processor power, continuous updates could be performed. However, since realistic cases are limited by complexity and by assorted training/processing errors, some degree of channel mismatch is inevitable.

With respect to the three fundamental problems posed above, the objective of this paper is to *propose a robust, low-complexity scheme for sequence estimation in the presence of noise, intersymbol interference, and mismatched channel conditions*. The paper first formulates the problem of interest and then provides a brief yet important review of a related receiver, as well as the theoretical development of the common least-squares algorithms. Theoretical and empirical performance analyses are then provided for a new family of receivers, followed by implementation considerations and a summary of the new results.

Problem formulation

As shown in **Figure 1**, we restrict attention to the common *discrete-time equivalent communication system model*. In the figure, an uncoded pulse-amplitude-modulated (PAM) binary data symbol, x_k, is transmitted across a linear time-invariant (LTI) channel filter[1] with impulse response $\{h_k\}$. The filter output y_k is then corrupted by additive white Gaussian noise (AWGN), denoted as n_k, which has a zero mean and a variance of σ_n^2. Thus, the kth received symbol, r_k, can be expressed as

$$\begin{aligned} r_k &= y_k + n_k \\ &= \sum_{i=-\infty}^{\infty} h_i x_{k-i} + n_k, \end{aligned} \tag{1}$$

and the goal of the receiver is to efficiently process this received signal to generate $\hat{x}_k$, a reliable estimate of the transmitted signal.

To address this challenging problem, a family of adaptive communication receivers is proposed which is based on variants of the common recursive least-squares (RLS) algorithm. These new algorithms, which we refer to collectively as recursive least-squares sequence estimation (RLSSE) algorithms, offer the performance and low complexity enjoyed by traditional receivers, yet are insensitive to practical phenomena such as noise mismatch, which have been shown to plague both linear and DFE receivers [4]. In addition, we show that RLSSE receivers can be implemented in a very efficient manner, since the same basic algorithm can be used for both the channel and sequence estimation tasks [5].

Kalman filter equalizer

The basic RLSSE structure described herein is related to the Kalman filter equalizer (KFE) originally proposed by

[1] Although the problem of interest is aimed at time-varying channels, we approach it by assuming that the sampling rate is much greater than the channel dynamics.

Lawrence and Kaufman [6], who utilized a discrete Kalman filter to estimate a binary input sequence transmitted over a dispersive finite impulse response (FIR) channel. They also treated the channel estimation problem by extending the state vector to include the channel tap coefficients. This (nonlinear) extended Kalman filter (EKF) approach falls into the category of blind equalization, since no training sequence is exploited.

There have been several other efforts which follow the KFE approach for sequence estimation. Mark [7] briefly discusses the initialization of the state vector, as well as a method of avoiding the EKF approach in [6] by estimating the channel tap gains via a pseudo decision feedback approach. Kleibanov et al. [8] also investigated the Kalman equalizer, but examined its convergence properties further and showed the advantages of increasing the filter dimension beyond the length of the channel filter. Luvison and Pirani [9, 10] also proposed a scheme which is based on that of [6], but additionally discussed topics such as carrier recovery, timing extraction, and limitations of the KFE under real-time constraints.

An in-depth summary of the KFE approach as well as its practical considerations can be found in the work of Benedetto and Biglieri [11], who extended the theory to cover correlated data, and also provided several examples to demonstrate the sensitivity of the KFE approach to channel mismatch. Mulgrew and Cowan also presented a detailed state-space analysis of the KFE for signaling over minimum and nonminimum phase channels [12]. By exploiting Wiener filter theory, they bounded the performance of the KFE and showed that the convergence and MSE performance of FIR transversal equalizers and KFEs are roughly the same. They also found that the KFE was typically of lower order, although this does not necessarily constitute lower overall complexity.

Prasad and Pathak [13] developed a similar analysis; a state-space approach was exploited. However, they used Mendel's smoothing approach from seismology [14], claiming that it yields marginally better performance under a low-complexity constraint than the Lawrence-Kaufman approach [6]. More recently, Yurtseven and Kumar [15] have revisited the work of [6] and applied a stochastic Newton algorithm for the estimation of the extended state vector. Although they claimed to have improved on the EKF approach of [6], their approach is also not guaranteed to converge. Similar efforts for improving the EKF channel estimator can also be found in the work of Delle Mese and Corsini [16].

• *Limitations of the KFE approach*

Although the KFE approach described above is theoretically sound and may be useful in certain applications, KFE receivers share two fundamental limitations. First, one must have an accurate estimate of the noise statistics in order to fully exploit the Kalman filter structure.[2] This not only increases system complexity (since a noise variance estimator must be implemented), but also renders KFE receivers susceptible to noise mismatch. Second and more important is the fact that the Kalman filter is inherently a *linear* filter; hence, its performance over channels with spectral nulls is severely degraded by noise enhancement. In fact, the MSE performance of all KFE receivers is (asymptotically) no better than that of conventional linear equalizers—simple linear filters—optimized for an MSE criterion. This fundamental limitation, coupled with the noise mismatch problem, precludes the use of existing KFE schemes over many practical channels.

• *Motivation behind the RLSSE approach*

The RLSSE approach has been motivated by three practical needs: low complexity, reduced sensitivity to channel mismatch, and adequate performance over channels with spectral nulls. It differs from the KFE approach in several fundamental respects. First, the error criterion used in the RLSSE receivers is based on a least-squares criterion rather than a least-mean-square criterion. Since the least-squares approach makes no assumptions regarding the statistics of the noise, we will see that the RLSSE algorithms are impervious to white Gaussian noise mismatch, and that they are significantly less complex, because an estimate of the noise variance is not needed.

The second fundamental difference is that one class of the proposed RLSSE algorithms is *nonlinear*. Therefore, while the simplest (linear) RLSSE algorithm is comparable in performance to linear equalizers—and hence KFE receivers—the *constrained* RLSSE algorithm can cope with channels with spectral nulls. Finally, and perhaps most important from a practical standpoint, we show that the RLSSE algorithms lead to an extremely efficient implementation, since the channel estimation and sequence estimation tasks have a duality that is fully exploited by the RLSSE approach [5].

Least-squares sequence estimation

The new RLSSE algorithm introduced in this section falls in the class of channel estimation-based receivers, as shown in **Figure 2**. This class of receivers requires an *explicit* channel estimate to decode the transmitted sequence, and has been found to be generally more robust to mismatch conditions than channel-equalization-based receivers such as the linear, DFE, and neural-network-based receivers.

Before we describe the RLSSE algorithm, it is worthwhile to briefly review the standard RLS algorithm

[2] A method of estimating the noise statistics was briefly discussed in [12]; however, the complexity and accuracy of such a process remain an open issue.

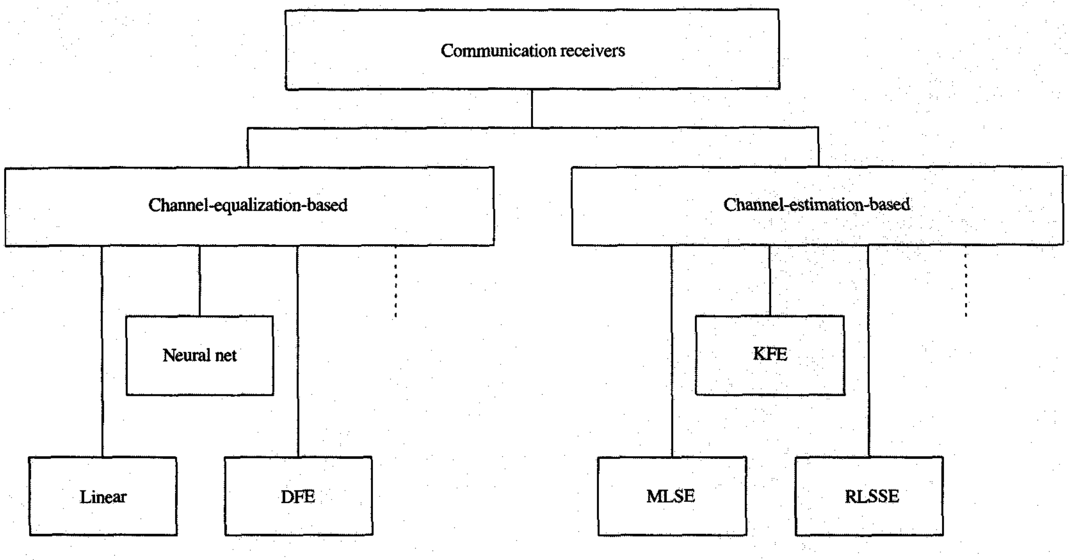

Figure 2

Partition of selected communication receivers. The RLSSE approach falls under the category of channel-estimation-based receivers.

and its utility in channel estimation. As we contrast the channel and sequence estimation problems and note their strong duality, the motivation behind the RLSSE approach becomes apparent.

- *RLS algorithm and channel estimation*

The application of the recursive least-squares (RLS) algorithm to adaptive filtering problems has had considerable attention in the communications literature because of its fast convergence properties and insensitivity to the channel's eigenvalue spread [2, 17–22]. It has served well in initializing and tracking the taps of linear and decision feedback equalizers, and has also been extremely useful in the task of channel estimation, which is crucial in any MLSE receiver.

Although the development of the least-squares estimator is now well known, we review it briefly here so that the RLSSE development can exploit this familiar derivation. We derive both the batch and recursive forms of the least-squares channel estimator in order to expose the features which will help us predict the performance of the RLSSE family of receivers. In addition, in order to deal with possible variations in the coefficients of the channel model, an exponential weighting is also employed in the least-squares performance index.[3]

[3] The channel estimation approach here and in Appendix A follows that found in [2] and [19], but has been extended to treat the more generic IIR channel model.

Batch least-squares channel estimator (BLSCE)

In our analysis, we assume a slowly time-varying channel filter which can be described by a finite-order difference equation given by

$$y_k = \sum_{i=1}^{M} a_i y_{k-i} + \sum_{i=0}^{L-1} b_i x_{k-i}, \tag{2}$$

where x_k and y_k represent the noise-free channel input and output, respectively, and M and L represent the order of the model. Thus, it is clear that y_k can be recursively computed solely on the basis of M past outputs and L inputs, and our development also holds for the special FIR case (i.e., $M = 0$).

We can define the coefficients of the actual channel model in (2) by $\mathbf{a}$ and $\mathbf{b}$, which are of dimension $M \times 1$ and $L \times 1$, respectively, and are given by

$$\mathbf{a} = [a_1\, a_2 \cdots a_M]^{\mathrm{T}},$$

$$\mathbf{b} = [b_0\, b_1 \cdots b_{L-1}]^{\mathrm{T}}, \tag{3}$$

where $[\bullet]^{\mathrm{T}}$ denotes transposition. For notational simplicity, an $(M + L) \times 1$ *coefficient vector* is introduced and defined as

$$\mathbf{c} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}. \tag{4}$$

Since the vector $\mathbf{c}$ completely specifies the actual channel, the channel estimation problem can be posed as finding an estimate $\bar{\mathbf{c}}$ which is closest to $\mathbf{c}$ in a least-squares sense.[4] The actual channel and its estimate at time k are denoted by $\mathbf{c}_k$ and $\bar{\mathbf{c}}_k$, respectively.

Assume that a *known* training sequence of $N + 1$ symbols is transmitted, resulting in the $N + 1$ input–output observable pairs,

$$\{(x_0, r_0)(x_1, r_1) \cdots (x_N, r_N)\}. \tag{5}$$

To ensure a unique solution, let $N > n$, where n is defined as $n = \max(L, M)$. Let the vector $\mathbf{q}_k$ be defined as the $(M + L) \times 1$ *training vector*[5]

$$\mathbf{q}_k = [r_{k-1}\ r_{k-2} \cdots r_{k-M}\ x_k\ x_{k-1} \cdots x_{k-(L-1)}]^{\mathrm{T}}. \tag{6}$$

We define the *estimation error* e_k as the difference between the actual channel output (i.e., the desired response) and the estimated channel output:

$$e_k = r_k - \mathbf{q}_k^{\mathrm{T}}\bar{\mathbf{c}}_k. \tag{7}$$

We can use the vector form of the above equation, i.e.,

$$\mathbf{e}_N = \mathbf{r}_N - \mathbf{Q}_N\,\bar{\mathbf{c}}_N, \tag{8}$$

where $\mathbf{r}_N$ and $\mathbf{e}_N$ are $(N - n + 1) \times 1$ *observable* and *error* vectors, respectively, and $\mathbf{Q}_N$ is an $(N - n + 1) \times (M + L)$ *training matrix*. These are given by

$$\mathbf{r}_N = [r_n\ r_{n+1} \cdots r_N]^{\mathrm{T}},$$

$$\mathbf{e}_N = [e_n\ e_{n+1} \cdots e_N]^{\mathrm{T}},$$

$$\mathbf{Q}_N = \begin{bmatrix} \mathbf{q}_n^{\mathrm{T}} \\ \mathbf{q}_{n+1}^{\mathrm{T}} \\ \vdots \\ \mathbf{q}_N^{\mathrm{T}} \end{bmatrix}. \tag{9}$$

The cost, J, for a weighted least-squares performance index can be expressed as

$$J = \sum_{k=n}^{N} w_k e_k^2, \tag{10}$$

where w_k is defined as an exponential weighting term,

$$w_k = \gamma^{(N-k)} \qquad 0 < \gamma \le 1, n \le k \le N. \tag{11}$$

To express our cost equation in matrix form, we define a weighting matrix $\mathbf{W}_N$, whose elements are given by

$$\mathbf{W}_N(i, j) = \begin{cases} w_{n+i-1} & i = j, \\ 0 & i \ne j. \end{cases} \tag{12}$$

Thus our cost function can be formulated as

$$\begin{aligned} J &= \mathbf{e}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{e}_N \\ &= [\mathbf{r}_N - \mathbf{Q}_N\bar{\mathbf{c}}_N]^{\mathrm{T}}\mathbf{W}_N[\mathbf{r}_N - \mathbf{Q}_N\bar{\mathbf{c}}_N] \\ &= \mathbf{r}_N^T\mathbf{W}_N\mathbf{r}_N - \bar{\mathbf{c}}_N^{\mathrm{T}}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N - \mathbf{r}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N\bar{\mathbf{c}}_N + \bar{\mathbf{c}}_N^{\mathrm{T}}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N\bar{\mathbf{c}}_N \\ &= \mathbf{r}_N^T\mathbf{W}_N\mathbf{r}_N - 2\bar{\mathbf{c}}_N^{\mathrm{T}}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N + \bar{\mathbf{c}}_N^{\mathrm{T}}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N\bar{\mathbf{c}}_N. \end{aligned} \tag{13}$$

Now, by using the gradient operator to minimize (13) with respect to $\bar{\mathbf{c}}_N$, we obtain

$$\nabla J = -2\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N + 2\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N\bar{\mathbf{c}}_N = \mathbf{0}, \tag{14}$$

where $\mathbf{0}$ is the all-zero vector. Solving for $\bar{\mathbf{c}}_N$ yields the optimal weighted least-squares estimate,

$$\bar{\mathbf{c}}_N^{LS} = [\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N]^{-1}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N. \tag{15}$$

With each new received signal, we can update the channel estimate by simply updating the observable vector $\mathbf{r}$ and the training matrix $\mathbf{Q}$, and computing the new channel estimate by (15). The update of $\mathbf{r}$ can be accomplished via

$$\mathbf{r}_k^{\mathrm{T}} = \mathbf{r}_{k-1}^{\mathrm{T}}\mathbf{S} + r_k\mathbf{g}, \tag{16}$$

where we have introduced an *initialization vector* $\mathbf{g}$, which is defined by

$$\mathbf{g} = [0\ 0 \cdots 0\ 0\ 1]. \tag{17}$$

The *shift matrix* $\mathbf{S}$ is given by

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & & 0 & 0 \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & 0 & 0 \\ 0 & 0 & & 1 & 0 \end{bmatrix}. \tag{18}$$

A block diagram[6] of the BLSCE structure is shown in **Figure 3**. Note that while the continuous update of the observable vector $\mathbf{r}$ is explicitly shown, the continuous update of the training matrix $\mathbf{Q}$ is not shown.

Several observations can now be made regarding the BLSCE structure which will be useful for comparing with the batch sequence estimator development.

BLSCE observations

- *Linear estimator* — Since all operations on the observable vector are linear operations, we see that the BLSCE structure is clearly a linear estimator.

[4] Real-valued (soft) estimates of a vector $\mathbf{v}$ are denoted as $\bar{\mathbf{v}}$, and $\hat{\mathbf{v}}$ denotes a binary (hard) estimate. Also, unless denoted otherwise by context, lowercase bold symbols represent vectors, while uppercase bold symbols represent matrices.

[5] In theory, $\mathbf{q}_k$ should be the ideal training vector,

$$\mathbf{q}_k = [y_{k-1}\ y_{k-2} \cdots y_{k-M}\ x_k\ x_{k-1} \cdots x_{k-(L-1)}]^{\mathrm{T}}.$$

But for IIR channels ($M > 0$), the first M elements in this ideal version require knowledge of the unobservable (noise-free) channel output, so the more practical definition above is used. Although the implications of this are not significant at moderate signal-to-noise ratio (SNR), the practical definition generally leads to biased channel estimates for IIR channel filters that are based on an output-error formulation [23].

[6] Solid lines in subsequent figures represent scalar values, while dashed lines represent vector quantities.

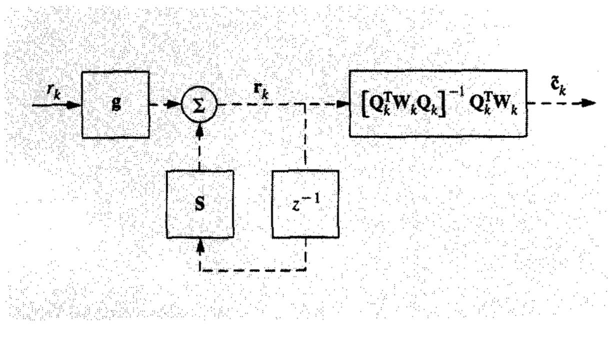

Figure 3

Batch least-squares channel estimator (BLSCE). The term z^{-1} represents a unit delay.

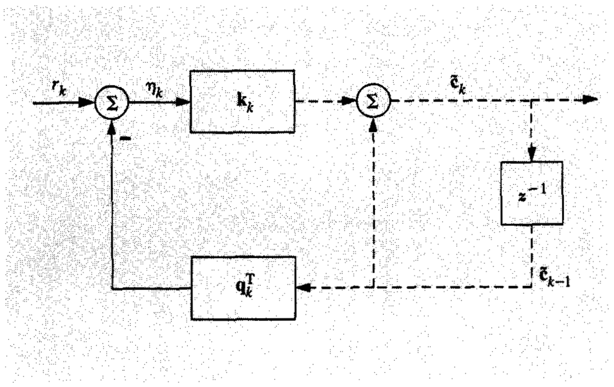

Figure 4

Recursive least-squares channel estimator (RLSCE).

- *Time-varying* — Since $\mathbf{q}_k^{\mathrm{T}}$, and hence $\mathbf{Q}_k^{\mathrm{T}}$, are changing with each new training symbol, the filter shown in the diagram is a time-varying structure. Note further that this is true whether or not the channel is time-variant.
- *Sensitive to training matrix* — By noting the inverse operation required in this structure, we see that the estimator performance hinges upon the invertibility of $[\mathbf{Q}_k^{\mathrm{T}}\mathbf{W}_k\mathbf{Q}_k]$. However, since we are at liberty to design the training sequence *a priori*, a carefully planned training sequence usually avoids any significant problems.
- *High complexity* — Since the BLSCE filter is time-variant, the inverse operation shown in the diagram must be performed with each new sample. This is, of course, highly inefficient, and so recursive forms are desirable.

Recursive least-squares channel estimator (RLSCE)

The recursive implementation of (15) is embodied in **Figure 4**. The figure illustrates a linear recursive estimator which forces the estimate to take the form

$$\tilde{\mathbf{c}}_k = \tilde{\mathbf{c}}_{k-1} + \mathbf{k}_k \eta_k , \qquad (19)$$

where η_k represents the estimation error and $\mathbf{k}_k$ is a gain vector. Thus the essence of the recursive solution lies in finding the gain vector (derived in Appendix A) which yields the least-squares estimate. The key equations in the RLSCE algorithm are summarized in **Table 1**. Note that the algorithm requires an initialization of $\tilde{\mathbf{c}}_k$ by an initial guess, as well as a starting point for the correlation matrix inverse, $\mathbf{P}_k^{-1} = \mathbf{Q}_k^{\mathrm{T}}\mathbf{W}_k\mathbf{Q}_k$, to remedy its possibly ill-conditioned nature [2]. Also, the intermediate variables $\mathbf{U}_k$ and $\mathbf{v}_k$ have been introduced to reduce computations.[7]

- *Duality between channel and sequence estimation*

Theoretically, the receiver has no knowledge of whether a signal $\{x_k\}$ was passed through a linear filter with impulse response $\{h_k\}$, or whether $\{h_k\}$ was passed through a filter with impulse response $\{x_k\}$. The channel output is simply the noise-corrupted convolution of the input and channel impulse response, i.e.,

$$r_k = x_k * h_k + n_k . \qquad (20)$$

By posing both problems in a system identification framework, we can describe the two problems as *dual* adaptive filtering tasks. That is, the channel estimation process uses a known information sequence to identify an unknown channel, while the sequence estimation process uses a "known" impulse response to identify an unknown sequence. This duality is shown in **Figures 5** and **6**. The inherent similarity between channel and sequence estimation is clearly evident from (20) and the figures. Several observations worth noting are the following:

- Both problems are inverse problems (more specifically, deconvolution problems) which are known to be *ill-posed*; i.e., their solutions for nontrivial cases are not necessarily unique.
- In general, accurate knowledge of $\{h_k\}$ is required to estimate $\{x_k\}$, and vice versa.
- Accurate estimation of an arbitrary $\{h_k\}$ requires the use of a known sequence $\{x_k\}$ which is *persistently exciting*; i.e., the training signal must have sufficient energy in the spectral bands of interest to excite the unknown channel, and vice versa.

Although both problems share these similarities, the fact remains that $\{h_k\}$ is specified over a field of reals, and $\{x_k\}$

[7] An intermediate variable is introduced in [19] to reduce computation. That variable was factored to yield $\mathbf{U}_k$ and $\mathbf{v}_k$, resulting in a further reduction (at the expense of memory).

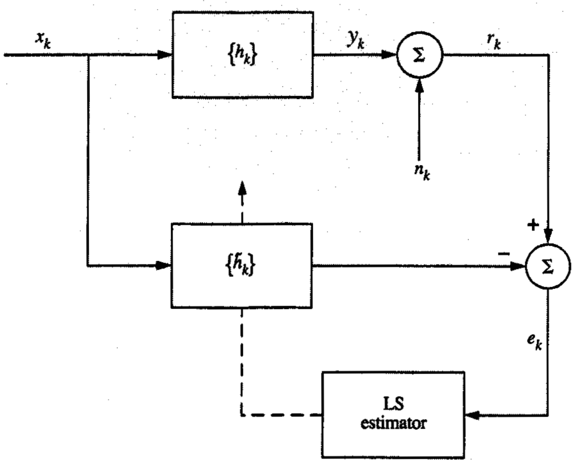

Figure 5

Training a channel estimator. A known training sequence $\{x_k\}$ is used to estimate the unknown sequence $\{h_k\}$.

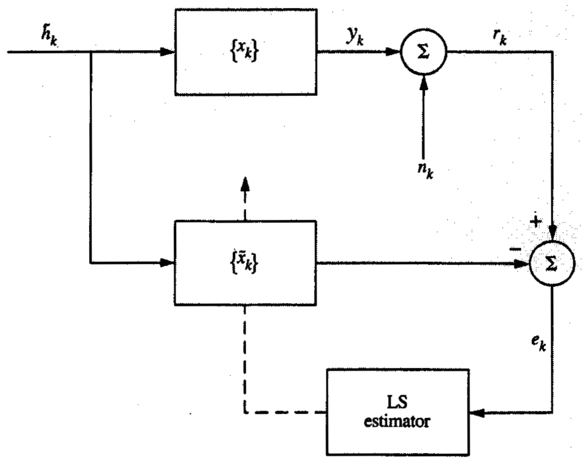

Figure 6

Training a sequence estimator. A "known" training sequence $\{\tilde{h}_k\}$ is used to estimate the unknown (soft) sequence $\{\tilde{x}_k\}$.

Table 1 Recursive least-squares channel estimator (RLSCE) algorithm.

No.	*Procedure*	$\pm$	$\times$	$\div$	$\gtrless$	*I/O*	*Storage*	*Comment*
0	$k = 0$, $\bar{\mathbf{c}}_0 = \mathbf{c}_{\text{init}}$, $\mathbf{P}_0 = \varepsilon^{-1}\mathbf{I}$	–	–	–	–	–	–	Initialization
1	r_k, $k = k + 1$	0	0	0	0	1	1	Receive signal
2	$\mathbf{q}_k$	0	0	0	0	N	N	Training vector
3	$\mathbf{U}_k = \gamma^{-1}\mathbf{P}_{k-1}$	0	$0.5(N^2 + N)$	0	0	$0.5(N^2 + N)$	N^2	Intermediate step
4	$\mathbf{v}_k = \mathbf{U}_k\mathbf{q}_k$	N^2	N^2	0	0	1	N	Intermediate step
5	$\mathbf{k}_k = \dfrac{\mathbf{v}_k}{1 + \mathbf{q}_k^{\mathrm{T}}\mathbf{v}_k}$	$N + 1$	N	N	0	1	N	Gain vector
6	$\eta_k = r_k - \mathbf{q}_k^{\mathrm{T}}\bar{\mathbf{c}}_{k-1}$	$N + 1$	N	0	0	1	1	Innovation term
7	$\mathbf{P}_k = \mathbf{U}_k - \mathbf{k}_k\mathbf{v}_k^{\mathrm{T}}$	$0.5(N^2 + N)$	$0.5(N^2 + N)$	0	0	$0.5(N^2 + N)$	N^2	Correlation matrix inverse
8	$\bar{\mathbf{c}}_k = \bar{\mathbf{c}}_{k-1} + \mathbf{k}_k\eta_k$	N	N	0	0	0	N	Channel estimate
	Total	$1.5N^2 + 3.5N + 2$	$2N^2 + 4N$	N	0	$N^2 + 2N + 4$	$2N^2 + 4N + 2$	

Notes:
1. $\bar{\mathbf{c}}_k = [\tilde{a}_1 \cdots \tilde{a}_M \; \tilde{b}_0 \cdots \tilde{b}_{L-1}]^{\mathrm{T}}$, $\mathbf{q}_k = [y_{k-1} \cdots y_{k-M} \, x_k \cdots x_{k-(L-1)}]^{\mathrm{T}}$.
2. N is the order of the estimator; i.e., $N \triangleq M + L$.
3. $\mathbf{c}_{\text{init}}$ is initial estimate; ε is small positive constant; γ is exponential-weighting.
4. I/O represents memory input/output with no arithmetic operations performed.
5. Step 3 not required if $\gamma = 1.0$.

is over a binary field. Thus, while channel estimation deals with the estimation of a *finite* number of *real-valued* parameters, sequence estimation[8] deals with the estimation of an *infinite* number of *binary-valued* parameters.

Batch least-squares sequence estimator (BLSSE)

The duality described above suggests that by making several basic substitutions, the same estimator structure can be used, and so we now revisit the batch channel estimator and perform the appropriate modifications.

First, we need an appropriate model for the unknown information sequence. Since we are assuming that $\{x_k\}$ is an uncoded[9] sequence, the z-transform of an arbitrary L-length message sequence can be given by a simple FIR model, viz.,

$$X(z) = \sum_{i=0}^{L-1} x_i z^{-1}, \tag{21}$$

where the length of the message sequence is at least as long as the channel response to enable a unique solution. Thus, by using the same naming convention followed in the channel estimation development, we find that the coefficient vector and its soft estimate are now given by

$$\mathbf{c}_k = [x_{k-(L-1)}\ x_{k-(L-1)+1}\ \cdots\ x_{k-1}\ x_k]^{\mathrm{T}},$$
$$\bar{\mathbf{c}}_k = [\bar{x}_{k-(L-1)}\ \bar{x}_{k-(L-1)+1}\ \cdots\ \bar{x}_{k-1}\ \bar{x}_k]^{\mathrm{T}}. \tag{22}$$

Next, we must reexamine the topic of input–output observable pairs. Recall that the channel estimator development required the use of a known training sequence. The same applies for sequence estimation. However, in this case the "known" training signal is the estimate of the channel's impulse response. Thus, the training vector is now given by the $(L \times 1)$ vector

$$\mathbf{q}_k = [\bar{h}_{L-1}\ \bar{h}_{L-2}\ \cdots\ \bar{h}_1\ \bar{h}_0]^{\mathrm{T}}. \tag{23}$$

The estimated samples of the impulse response are determined by

$$\bar{h}_k = \sum_{i=1}^{M} \bar{a}_i h_{k-i} + \sum_{i=0}^{L-1} \bar{b}_i \delta_{k-i}, \tag{24}$$

where δ_{k-i} is the Dirac delta function, and the parameters $\bar{a}_i$ and $\bar{b}_i$ are found by the channel estimator. Note that the training matrix also changes with the new $\mathbf{q}_k$, as defined in (9). These key variables and their counterparts in the channel estimation process are summarized in **Table 2**.

By exploiting the duality shown in Table 2 and applying the new variables directly in the solution for the least-squares channel estimate (15),

$$\bar{\mathbf{c}}_k = [\mathbf{Q}_k^{\mathrm{T}}\mathbf{W}_k\mathbf{Q}_k]^{-1}\mathbf{Q}_k^{\mathrm{T}}\mathbf{W}_k\mathbf{r}_k, \tag{15}$$

we obtain the *unconstrained* solution for the least-squares sequence estimate, i.e., the *real-valued* sequence estimate which is closest to the actual transmitted sequence in a least-squares sense. Recall, however, that the transmitted information need not be of finite length, and it is, furthermore, a binary-valued sequence. To accommodate this, we first extract the first element of $\bar{\mathbf{c}}_k$ by filtering the parameter vector,

$$\bar{x}_{k-(L-1)} = \mathbf{f}\bar{\mathbf{c}}_k, \tag{25}$$

where the *pick-off vector* $\mathbf{f}$ is defined by

$$\mathbf{f} = [1\ 0 \cdots 0]. \tag{26}$$

This soft decision can now be passed through a simple threshold device to yield the hard decision $\hat{x}_{k-(L-1)}$, i.e.,

$$\hat{x}_{k-(L-1)} = \mathrm{sgn}\,\{\bar{x}_{k-(L-1)}\} = \begin{cases} +1 & \bar{x}_{k-(L-1)} \geq 0, \\ -1 & \bar{x}_{k-(L-1)} < 0. \end{cases} \tag{27}$$

The process described would continue by simply shifting the elements of $\mathbf{r}_k$ as was done in the BLSCE structure, and recomputing the least-squares solution given by (15). A block diagram of the BLSSE receiver is shown in **Figure 7.**

BLSSE observations

- *Identical in structure to batch channel estimator* — Preceding the pick-off vector **f** shown in the figure, the BLSSE and BLSCE structures are identical in form; hence, by simply redefining the contents of the training vector, one can use the same architecture to estimate both the channel and unknown sequence.
- *Linear estimator* — As was the case for the BLSCE structure, the *soft* estimate is a linear function of the observable vector; hence, the BLSSE estimator—up to the threshold device—is a linear receiver.
- *Time-invariant receiver for time-invariant channels* — Recall that the batch channel estimator was time-varying, even for stationary channels. If the channel filter varies with time, the BLSSE receiver is also time-varying. However, if the channel is time-invariant, the training vector $\mathbf{q}_k^{\mathrm{T}}$, and hence $\mathbf{Q}_k^{\mathrm{T}}$, of the BLSSE receiver does not change with each new training symbol. Therefore, once an initial burst of symbols has been sent, the filter reaches steady state and hence becomes stationary.
- *Sensitivity to training matrix* — The BLSSE filter is also sensitive to the invertibility of $[\mathbf{Q}_k^{\mathrm{T}}\mathbf{W}_k\mathbf{Q}_k]$. In contrast to the channel-estimation problem, though, the training

[8] From a classical standpoint [24], it can be argued that what is commonly known as *sequence estimation* is not an estimation problem at all, but a detection problem. We resist this argument, since posing the problem in an estimation framework leads to an elegant duality which we exploit.

[9] This model could be easily modified for exploiting sequences which have been intentionally (or unintentionally) coded by a linear filter, such as a transmission filter or linear code used in bandwidth-shaping schemes.

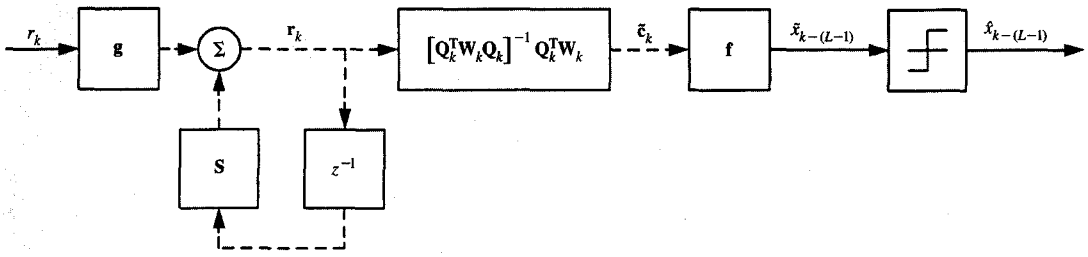

Figure 7

Batch least-squares sequence estimator (BLSSE).

Table 2 Duality between channel and sequence estimation.

	Channel estimation	*Sequence estimation*
Signal model	$H(z) = \sum_{i=0}^{L-1} h_i z^{-i}$	$X(z) = \sum_{i=0}^{L-1} x_i z^{-i}$
Coefficient vector $\mathbf{c}_k$	$[h_0\ h_1 \cdots h_{L-2}\ h_{L-1}]^T$	$[x_{k-(L-1)}\ x_{k-(L-1)+1} \cdots x_k]^T$
Training vector $\mathbf{q}_k$	$[x_k\ x_{k-1} \cdots x_{k-(L-1)}]^T$	$[\bar{h}_{L-1}\ \bar{h}_{L-2} \cdots \bar{h}_1\ \bar{h}_0]^T$

Note: An FIR channel model is shown, since this is much more prevalent in practice. The duality also exists for an IIR channel model, although it is not as obvious as that shown for the FIR model.

matrix $\mathbf{Q}_k$ cannot be specified by the designer, since it now represents the channel coefficients.

- *Moderate complexity* — Since the BLSSE filter is time-invariant for LTI channels, the matrix $[\mathbf{Q}_k^T\mathbf{W}_k\mathbf{Q}_k]^{-1}\mathbf{Q}_k^T\mathbf{W}_k$ must be calculated just once; hence, the batch form itself is not as inefficient as the BLSCE structure. However, even though the BLSSE structure may be more efficient than its BLSCE counterpart under time-invariant conditions, the ability to handle time-varying channels still necessitates a recursive form.

Recursive least-squares sequence estimator (RLSSE)
By substituting the parameter definitions detailed in Table 2, the recursive form of the least-squares sequence estimator follows the derivation of the recursive channel estimator. The only difference is found at the end of each iteration, where the following must occur in order to accommodate the shift and threshold operations, which are *not* found in conventional RLS algorithms:

- *Pick off and constrain soft estimate.* Prior to shifting the elements of the coefficient vector, we must first extract an estimate of the "oldest" symbol, i.e., $\tilde{x}_{k-(L-1)}$. Analytically, we can do so by filtering the parameter vector by the pick-off vector **f** defined in (26). Once this soft estimate has been removed from the coefficient vector, we constrain it and obtain the final estimate.
- *Shift and initialize* $\tilde{\mathbf{c}}_k$. By noting the form of the parameter vector, it is clear that after each iteration, a new initial estimate of the parameter vector can be found by simply shifting the elements to the left by using the shifting matrix S, already defined in (18). In addition, while all RLS algorithms require an initial estimate of the coefficient vector at start-up, our shifted RLS scheme requires a continuous initialization of the current (rightmost) symbol of the coefficient vector. This can be accomplished by augmenting the shifted coefficient vector by the quantity $\bar{x}_k\mathbf{g}$, where $\bar{x}_k$ is an initial guess of the current symbol. Note that $\bar{x}_k = 0$ for our equally likely binary model, but we leave this important step in the algorithm for generality and for possible extensions to coded schemes. Note that the shift matrix is a linear operation which is useful for the purpose of analysis, but in practice this is a trivial memory shift which simply

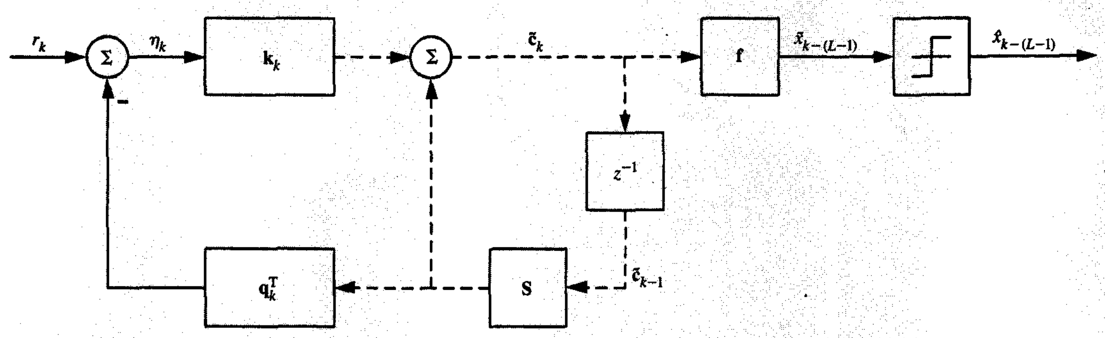

Figure 8

The RLSSE(L,0) receiver.

propagates a particular vector element from right to left. Thus, for a filter of length five, the unconstrained parameter vector and its shifted/initialized version could be generally described by

$$\tilde{\mathbf{c}}_k^{\mathrm{T}} = [\tilde{c}_1 \ \tilde{c}_2 \ \tilde{c}_3 \ \tilde{c}_4 \ \tilde{c}_5],$$

$$\tilde{\mathbf{c}}_k^{\mathrm{T}}\mathbf{S} + \tilde{x}_k\mathbf{g} = [\tilde{c}_2 \ \tilde{c}_3 \ \tilde{c}_4 \ \tilde{c}_5 \ \tilde{x}_k]. \tag{28}$$

- *Shift and initialize P_k*. The inverse covariance matrix must also be shifted (up and to the left) to remain synchronized with the coefficient vector. This can be accomplished by filtering the covariance matrix by the transformation $\mathbf{S}^{\mathrm{T}}\mathbf{PS}$, where **S** is the previously defined shifting matrix. It should be noted that this covariance shift effectively shifts the elements of $\mathbf{k}_k$ as well. After the shift, the "new" elements of $\mathbf{P}_k$ must also be initialized. We can satisfy this initialization by augmenting the shifted covariance matrix above with the matrix $\varepsilon^{-1}\mathbf{gI}$, whose entries are all zero except for the lower right element, which has the value ε^{-1}. For example, shifting and initializing an arbitrary 3 × 3 matrix **P** would appear as

$$\mathbf{P} = \begin{bmatrix} P(1,1) & P(2,1) & P(3,1) \\ P(1,2) & P(2,2) & P(3,2) \\ P(1,3) & P(2,3) & P(3,3) \end{bmatrix},$$

$$\mathbf{S}^{\mathrm{T}}\mathbf{PS} + \varepsilon^{-1}\mathbf{gI} = \begin{bmatrix} P(2,2) & P(3,2) & 0 \\ P(2,3) & P(3,3) & 0 \\ 0 & 0 & \varepsilon^{-1} \end{bmatrix}. \tag{29}$$

A block diagram of the linear RLSSE receiver is shown in **Figure 8**. For reasons which become apparent in the next section, this receiver is referred to as the RLSSE(L, 0) receiver.

Constraining the estimator—the RLSSE(L, D) algorithm

Although the optimum (maximum-likelihood) receiver for binary signaling is well known and has been shown to be nonlinear (see for instance [25]), the complexity of these MLSE schemes typically precludes a realizable implementation (e.g., Viterbi decoders). Other suboptimum methods have evolved, such as reduced-state methods [26, 27] or reduced-search methods such as the M-algorithm [28]. Still, the moderate to high complexity of these suboptimum MLSE schemes has prevented their widespread use.

Indeed, the DFE is undoubtedly the most popular nonlinear receiver which exhibits both low complexity and reasonable performance. Furthermore, since the DFE is merely a nonlinear variant of the linear feedback equalizer, we are motivated to seek a nonlinear variant of the linear RLSSE estimator in an effort to improve its performance over channels with severe ISI.

Toward that end, consider the role of the past sequence estimate $\tilde{\mathbf{c}}_{k-1}$ shown in Figure 8. During each iteration, this vector acts as an initial guess which is corrected on the basis of the error and gain vector. Note, however, that $\tilde{\mathbf{c}}_{k-1}$ represents a *soft* estimate of the true coefficient vector $\mathbf{c}_k$. Hence, we already know that it is, in general, not correct. It thus seems intuitive to constrain these soft estimates before they are used in the next iteration. If these decisions are correct, $\tilde{\mathbf{c}}_k$ stands a good chance of being correct as well. However, since some of these elements have been in the RLSSE decoding window for only

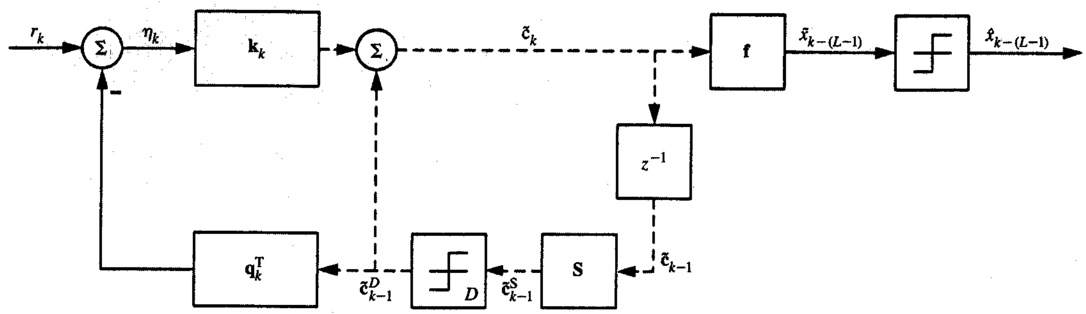

Figure 9

The RLSSE(L, D) receiver.

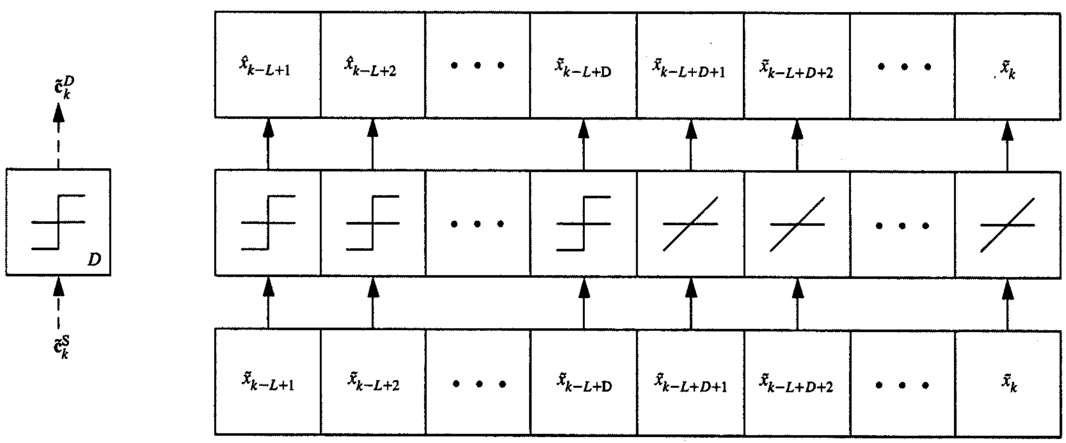

Figure 10

Expanded view of the D-threshold device for binary data transmission.

a few samples and are, furthermore, noise-corrupted, an incorrect decision is certainly possible. The leftmost element of $\tilde{\mathbf{c}}_{k-1}$, however, can be considered the most *mature* estimate; i.e., since it has propagated through the full length of the RLSSE window, it seems reasonable to assume that, after some point, it has been equalized enough so that constraining it may improve the overall performance.

In this spirit, we extend the linear RLSSE receiver to a nonlinear structure which we designate the RLSSE(L, D) receiver, as shown in **Figure 9**.

In the figure, all elements are identical to those found in Figure 8 except for the new D-threshold device which has been inserted in the feedback path. The D in the lower right corner of the device signifies that only the leftmost D symbols are constrained. Thus, the RLSSE$(L, 0)$ is the

Table 3 The RLSSE(L, D) algorithm.

No.	*Procedure*	$\pm$	$\times$	$\div$	$\gtrless$	*I/O*	*Storage*	*Comment*
0	$k = 0,\ \tilde{\mathbf{c}}_0 = \mathbf{c}_{\text{init}},$ $\mathbf{P}_0 = \varepsilon^{-1}\mathbf{I}$	–	–	–	–	–	–	Initialization
1	$r_k,\ k = k + 1$	0	0	0	0	1	1	Receive signal
2	$\mathbf{q}_k$	0	0	0	0	L	L	Training vector
3	$\mathbf{U}_k = \gamma^{-1}\mathbf{P}_{k-1}$	0	$0.5(L^2 + L)$	0	0	$0.5(L^2 + L)$	L^2	Intermediate step
4	$\mathbf{v}_k = \mathbf{U}_k\mathbf{q}_k$	L^2	L^2	0	0	1	L	Intermediate step
5	$\mathbf{k}_k = \dfrac{\mathbf{v}_k}{1 + \mathbf{q}_k^{\mathrm{T}}\mathbf{v}_k}$	$L + 1$	L	L	0	1	L	Gain vector
6	$\eta_k = r_k - \mathbf{q}_k^{\mathrm{T}}\tilde{\mathbf{c}}_{k-1}$	$L + 1$	L	0	0	1	1	Innovation term
7	$\mathbf{P}_k = \mathbf{U}_k - \mathbf{k}_k\mathbf{v}_k^{\mathrm{T}}$	$0.5(L^2 + L)$	$0.5(L^2 + L)$	0	0	$0.5(L^2 + L)$	L^2	Correlation matrix inverse
8	$\tilde{\mathbf{c}}_k = \tilde{\mathbf{c}}^D_{k-1} + \mathbf{k}_k\eta_k$	L	L	0	0	0	L	Sequence estimate
9	$\hat{x}_{k-(L-1)} = \mathrm{sgn}(\mathbf{f}\tilde{\mathbf{c}}_k)$	0	0	0	1	0	0	Pickoff & constrain
10	$\tilde{\mathbf{c}}^S_{k-1} = \tilde{\mathbf{c}}^{\mathrm{T}}_{k-1}\mathbf{S} + \tilde{x}_k\mathbf{g}$	0	0	0	0	L	0	Shift & initialize $\tilde{\mathbf{c}}_k$
11	$\tilde{\mathbf{c}}^D_{k-1} = \mathrm{sgn}_D(\tilde{\mathbf{c}}^S_{k-1})$	0	0	0	D	0	0	Constrain D symbols
12	$\mathbf{P}_k = \mathbf{S}^{\mathrm{T}}\mathbf{P}_k\mathbf{S} + \varepsilon^{-1}\mathbf{g}\mathbf{I}$	0	0	0	0	L^2	0	Shift & initialize $\mathbf{P}_k$
	Total	$1.5L^2 + 3.5L + 2$	$2L^2 + 4L$	L	$D + 1$	$2L^2 + 3L + 4$	$2L^2 + 4L + 2$	

Notes:
1. $\tilde{\mathbf{c}}_k = [\tilde{x}_{k-(L-1)}\ \tilde{x}_{k-(L-1)+1} \cdots \tilde{x}_{k-1}\ \tilde{x}_k]^{\mathrm{T}}$, $\mathbf{q}_k = [\tilde{h}_{L-1}\ \tilde{h}_{L-2} \cdots \tilde{h}_1\ \tilde{h}_0]^{\mathrm{T}}$.
2. $\mathbf{c}_{\text{init}}$ is initial estimate; ε is small positive constant; γ is exponential-weighting; $\tilde{x}_k$ is soft estimate of current symbol.
3. Steps 1–8 are identical to the RLSCE algorithm of Table 1.
4. Step 3 is not required if $\gamma = 1.0$.
5. For time-invariant channels, Steps 2–7 and Step 12 are not necessary after the gain vector converges.

linear receiver already discussed, while, at the other extreme, the RLSSE(L, L) receiver uses a fully constrained estimate. An expanded view of the D-threshold device is shown in **Figure 10**, and a summary of the RLSSE(L, D) algorithm is given in **Table 3**.

Observations on RLSSE(L, D) receiver

- *Varying degrees of nonlinearity* — By noting once again that the final soft estimate is linear in the observable vector, we see that for $D = 0$, the RLSSE is a linear estimator up to the threshold device. However, one can also increase D up to the length of the RLSSE window, in order to increase the degree of nonlinearity.
- *Same core algorithm as RLSCE* — It is important to note that the processing required by Steps 0–8 of the RLSSE algorithm is identical to that found in the RLSCE algorithm—only the definition of $\mathbf{q}_k$ has changed. Furthermore, Steps 0–8 represent virtually all of the significant processing. This significant fact suggests that an RLSSE receiver can exploit the *same* core architecture for both the channel and sequence estimation tasks, as we have successfully accomplished in our tests.
- *Moderate to low complexity* — As shown in the table, the RLSSE algorithm can handle time-varying channels, since $\mathbf{q}_k$ need only be updated with the new channel estimate. From the complexity analysis shown in the table, we see that the worst-case complexity is $O(N^2)$ for time-varying channels. If, however, the channel is time-invariant, the gain vector rapidly converges to its steady-state value. Therefore, the burdensome task of computing $\mathbf{P}_k$ is not necessary, and so Steps 2–7 and Step 12 can be avoided, resulting in an $O(N)$ algorithm.
- *Synchronous operation* — After the initial burst of L symbols has been received, the algorithm will synchronously generate estimates. Thus, the overhead

of buffer management required by some receivers is avoided.

Theoretical performance analysis

An informative theoretical analysis of the nonlinear RLSSE(L, D) receiver does not seem feasible without unrealistic assumptions; hence, computer simulations have been used to establish the performance of the generic RLSSE(L, D) receiver. However, by assuming stationary and ergodic conditions, the asymptotic performance of the linear RLSSE$(L, 0)$ structure can easily be determined, as is now shown.

- *Perfect channel knowledge*

If one assumes that the channel estimator generated a perfect estimate of the channel impulse response, the mean square error (MSE) performance of the linear RLSSE$(L, 0)$ structure can be readily ascertained. Since the least-squares and least-mean-square error are asymptotically identical in AWGN [29, 30], the asymptotic performance of the RLSSE$(L, 0)$ structure with perfect channel knowledge is that of all linear receivers which were optimized for an MSE criterion, viz.,

$$J_{\mathrm{RLSSE}(\infty,0)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{N_0}{|H(e^{j\omega})|^2 + N_0} \, d\omega, \tag{30}$$

where $|H(e^{j\omega})|^2$ represents the discrete power spectrum of the channel filter.

- *White Gaussian noise mismatch*

If the noise statistics during training differ from those during decoding, a noise mismatch condition is present, as shown in **Figure 11**.

However, recall that the RLSSE approach does not depend on the noise statistics, implicitly or explicitly. It is strictly based on the channel estimate. Therefore, as long as the training sequence is sufficiently long that the channel estimate converges to an unbiased solution, the performance of the RLSSE(L, D) algorithm during decoding is independent of the noise variance during training. Hence, the RLSSE(L, D) receiver is insensitive to white Gaussian noise mismatch, and the MSE performance given in (30) is valid for the special case of noise mismatch. This insensitivity to noise mismatch is a significant advantage which is not offered by linear or decision feedback equalizers[10] [4].

- *Arbitrary channel mismatch*

We can apply the same argument to the case of arbitrary channel mismatch, shown in **Figure 12**, as long as we constrain the noise to be white and Gaussian. Therefore, by using the arbitrary channel mismatch formula derived in [4], and constraining the noise to be white, we find that the asymptotic performance of the RLSSE$(L, 0)$ receiver is given by

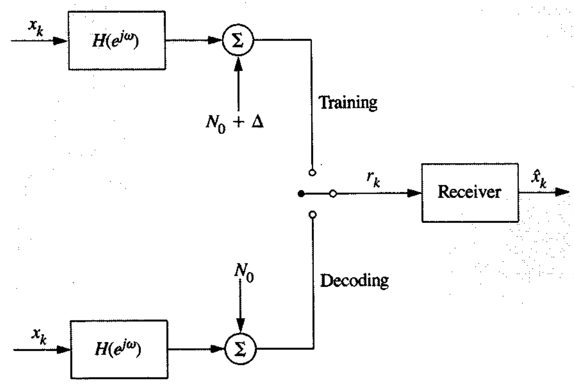

Figure 11

White Gaussian noise mismatch: the condition in which a receiver is trained/optimized for a particular noise level, but decodes in a different noise level environment.

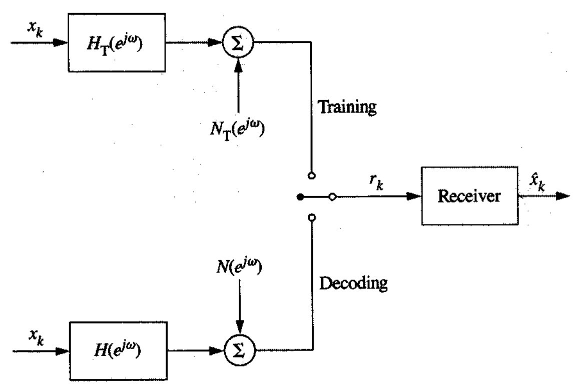

Figure 12

Arbitrary channel mismatch: the condition in which a receiver is trained for a particular channel filter and noise level, but decodes in another environment.

[10] F. Gozzo and J. B. Anderson, "The Impact of Noise Mismatch on Linear and Decision Feedback Equalizers," *IEEE Trans. Commun.*, submitted August 20, 1991.

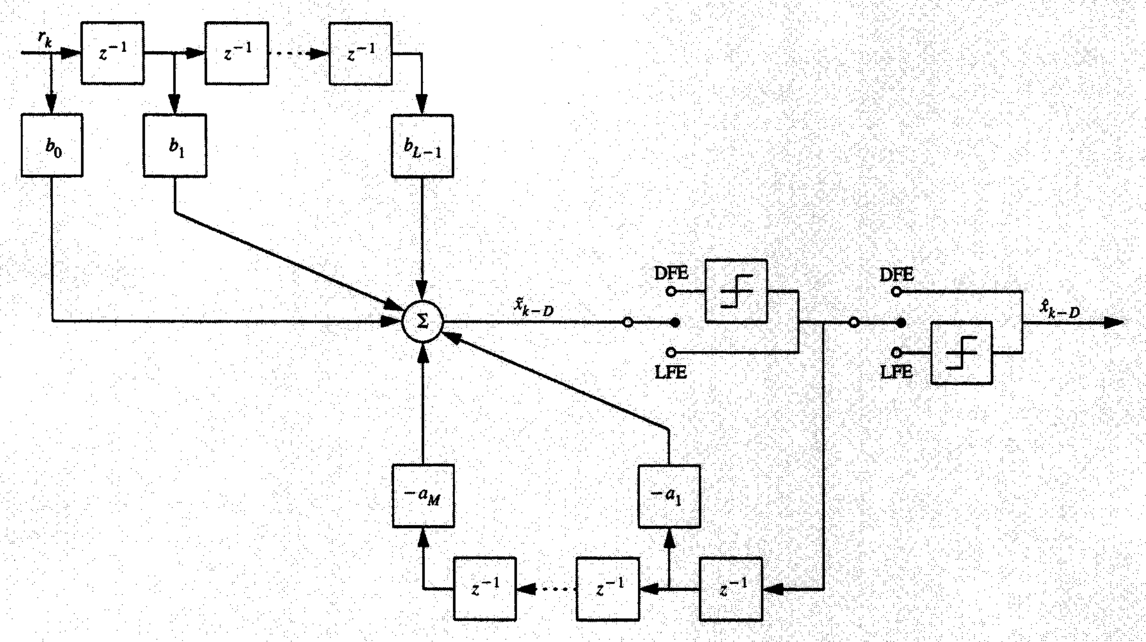

Figure 13

Block diagram of a generic equalizer. Feedforward and feedback coefficients are generally different from channel coefficients introduced previously.

Table 4 Channels used in noise mismatch tests.

Channel	$H(z)$	χ	*Comment*
A	$\dfrac{\sqrt{1-a^2}}{1-az^{-1}}$	$\dfrac{(1+a)^2}{(1-a)^2}$	IIR model of simple low-pass channel.
B	$\sqrt{1/1.951}\,(0.06 - 0.07z^{-1} + 0.1z^{-2} - 0.3z^{-3} - 0.7z^{-4} + z^{-5} + 0.5z^{-6} + 0.0z^{-7} - 0.3z^{-8} + 0.05z^{-9} + 0.1z^{-10})$	~24	FIR model of typical data-quality telephone line.
C	$\sqrt{1/2}\,(1 + z^{-1})$	∞	Two-way model with spectral null. Common in radio channel applications.

$$J_{\mathrm{RLSSE}(\infty,0)} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\frac{N_0}{|H(e^{j\omega})|^2 + N_0} + \frac{\left|[|H(e^{j\omega})|^2 + N_0]H_{\mathrm{T}}(e^{j\omega}) - [|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0]H(e^{j\omega})\right|^2}{[|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0]^2\,[|H(e^{j\omega})|^2 + N_0]}\right] d\omega, \tag{31}$$

where $H_{\mathrm{T}}(e^{j\omega})$ represents the discrete Fourier transform of the channel filter during training.

- *Performance comparison with linear/DFE receivers*

By exploiting the arbitrary mismatch results from [4], we can directly compare the performance of the RLSSE$(L, 0)$ receiver to linear and DFE receivers in the presence of channel mismatch in white Gaussian noise. The equalization schemes we address can be depicted by the generic equalizer shown in **Figure 13**. For the linear equalizer, the two switches shown are in the down position; hence, the threshold device is not in the feedback loop. For notational convenience, we refer to this linear structure as LFE(L, M), where L and M represent the number of taps in the feedforward and feedback filters, respectively. In addition, since the marginal performance advantage offered by recursive linear equalizers is typically outweighed by their potential instability [2], we restrict our analysis of linear equalizers to the nonrecursive case, i.e., LFE$(L, 0)$.

By flipping both switches up, the nonlinear threshold device is moved into the feedback loop, thereby creating the decision feedback equalizer which we refer to as DFE(L, M). Now, the feedforward and feedback filters can be interpreted as a linear equalizer and canceler, respectively [1]. That is, the received signal is first processed by the Lth-order feedforward filter which attempts to equalize the precursors (samples before the peak sample) of the channel impulse response. Once the precursors have been equalized, the Mth-order recursive filter mitigates the effect of the postcursors (samples after the peak sample) [31]. Thus,

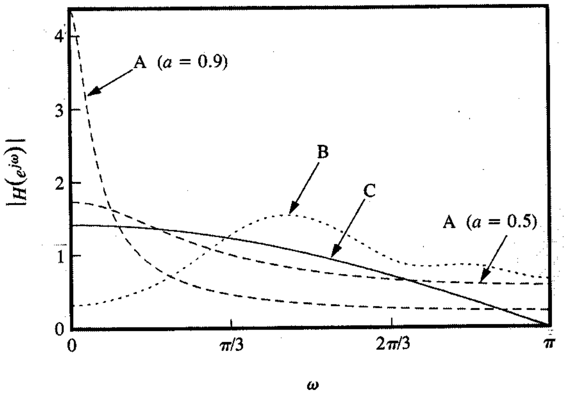

Figure 14

Magnitude spectra of channels used in noise mismatch tests.

Table 5 Channels used in arbitrary mismatch tests.

Channel	$H(z)$
A	$\dfrac{\sqrt{1 - a^2}}{1 - az^{-1}}$
C	$\sqrt{1 - a^2} + az^{-1}$

$$J_{\mathrm{LFE}(\infty,0)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{N_0}{|H(e^{j\omega})|^2 + N_0} + \frac{\left|[|H(e^{j\omega})|^2 + N_0] H_{\mathrm{T}}(e^{j\omega}) - [|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0 + \Delta] H(e^{j\omega})\right|^2}{[|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0 + \Delta]^2 [|H(e^{j\omega})|^2 + N_0]} \right] d\omega \tag{32}$$

and

$$J_{\mathrm{DFE}(\infty,\infty)} = \exp\left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln \left[\frac{N_0}{|H(e^{j\omega})|^2 + N_0} + \frac{\left|[|H(e^{j\omega})|^2 + N_0] H_{\mathrm{T}}(e^{j\omega}) - [|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0 + \Delta] H(e^{j\omega})\right|^2}{[|H_{\mathrm{T}}(e^{j\omega})|^2 + N_0 + \Delta]^2 [|H(e^{j\omega})|^2 + N_0]} \right] d\omega \right\}. \tag{33}$$

Note that by comparing (31) and (32), it is clear that $J_{\mathrm{RLSSE}(\infty,0)} \le J_{\mathrm{LFE}(\infty,0)}$ is always valid for the case of white Gaussian noise mismatch:

$$J_{\mathrm{RLSSE}(\infty,0)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{N_0}{|H(e^{j\omega})|^2 + N_0} d\omega \le \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{N_0}{|H(e^{j\omega})|^2 + N_0} + \frac{\Delta^2 |H(e^{j\omega})|^2}{[|H(e^{j\omega})|^2 + N_0 + \Delta]^2 [|H(e^{j\omega})|^2 + N_0]} \right] d\omega = J_{\mathrm{LFE}(\infty,0)}. \tag{34}$$

This relation is also valid under most (but not all) arbitrary channel mismatch cases. [11]

[11] The exceptions we found occurred in those situations where, for a given level of filter mismatch, the transversal equalizer performed better *with* noise mismatch than without it. Intuitively, the filter and noise mismatch can sometimes mitigate each other [4].

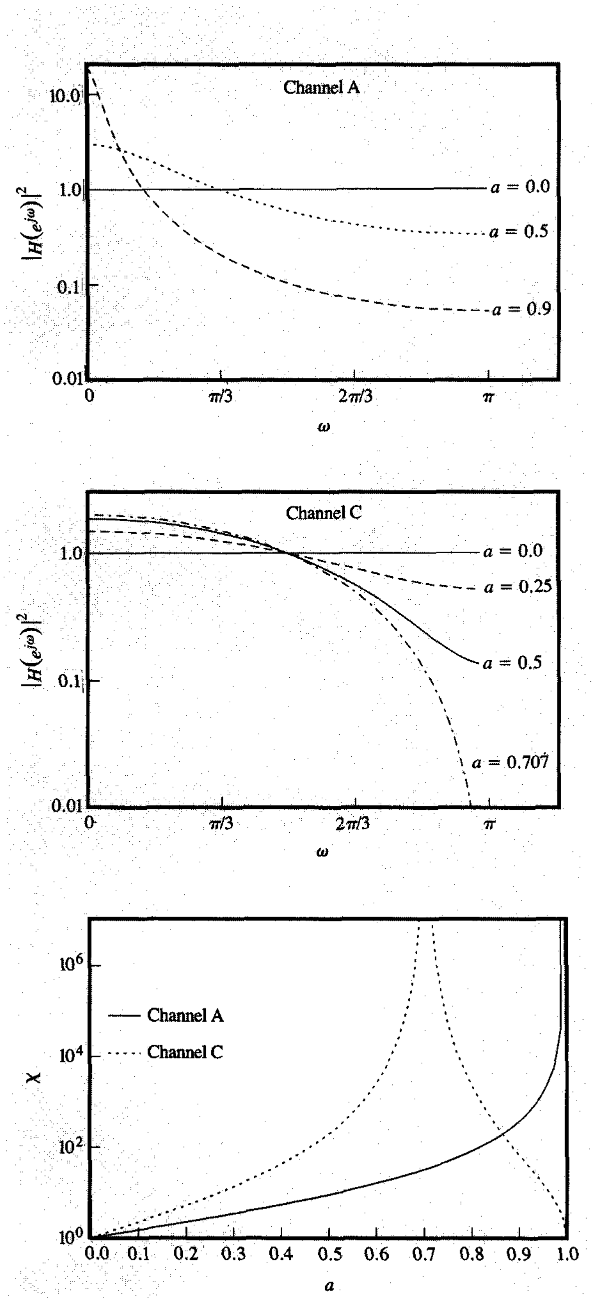

Figure 15

Magnitude spectra and approximate eigenvalue spread χ of channels used in arbitrary mismatch tests.

Empirical test results

We have resorted to simulation in order to further evaluate the MSE and P_b performance of the RLSSE(L, D) receiver under perfect channel knowledge and in various degrees of noise and channel filter mismatch. The transfer functions and magnitude spectra of these test channels are tabulated and plotted in **Table 4** and **Figure 14**, respectively, for the noise mismatch tests and in **Table 5** and **Figure 15**, respectively, for the arbitrary mismatch tests. Note that Channel C exhibits nulls, while Channel A does not have any nulls over the stable[12] regions of interest. Figure 15 also plots the approximate eigenvalue spread χ as a function of the test channel parameter a, where χ is defined as

$$\chi = \frac{\max(|H(e^{j\omega})|^2)}{\min(|H(e^{j\omega})|^2)} \tag{35}$$

and has been shown to be a useful tool in the study of arbitrary channel mismatch [4].

- *Test description*

For each of the test channels, the MSE and error probability P_b of LFE(31, 0), DFE(16, 15) and RLSSE(31, 0) receivers were measured. Since the LFE and DFE receivers required 31 taps to validate asymptotic MSE predictions, we used 31 taps for the linear RLSSE$(L, 0)$ receiver as well (for the MSE test only). Note, however, that the RLSSE$(L, 0)$ requires roughly L_c taps to achieve its optimum performance, where L_c is the length of the channel filter. This becomes evident when we examine the sensitivity of the RLSSE$(L, 0)$ performance as a function of L.

All receivers were trained by an RLS algorithm with a pseudorandom binary (± 1) data sequence until their respective tap vectors converged. The trained receivers then detected a pseudorandom binary data sequence until at least 100 bit errors occurred. All RLSSE receivers utilized the steady-state gain vector after 100 symbols; hence, the complexity of all receivers shown is $O(N)$. Finally, note that the RLSSE receivers invoked the *same* RLS procedure for both channel and sequence estimation.

- *Validation of theoretical MSE predictions*

The MSE performance comparison of the receivers, assuming perfect channel knowledge, is shown in **Figure 16**, while the analogous plots for the case of noise mismatch are shown in **Figure 17**. As expected, MSE results for channel filter mismatch were identical to the LFE results previously shown in [4], so for brevity they are not shown (P_b results are shown for channel filter mismatch, as described below). Theoretical MSE curves for the RLSSE$(\infty, 0)$, LFE$(\infty, 0)$ and DFE(∞, ∞) receivers were calculated by numerically integrating Equations (31), (32), and (33), respectively. In addition, the theoretical curve for the AWGN channel $[|H(e^{j\omega})|^2 = 1]$ was plotted

[12] Channel A does exhibit a null at $\omega = \pi$ when $a = 1$, but it becomes unstable since the pole is on the unit circle.

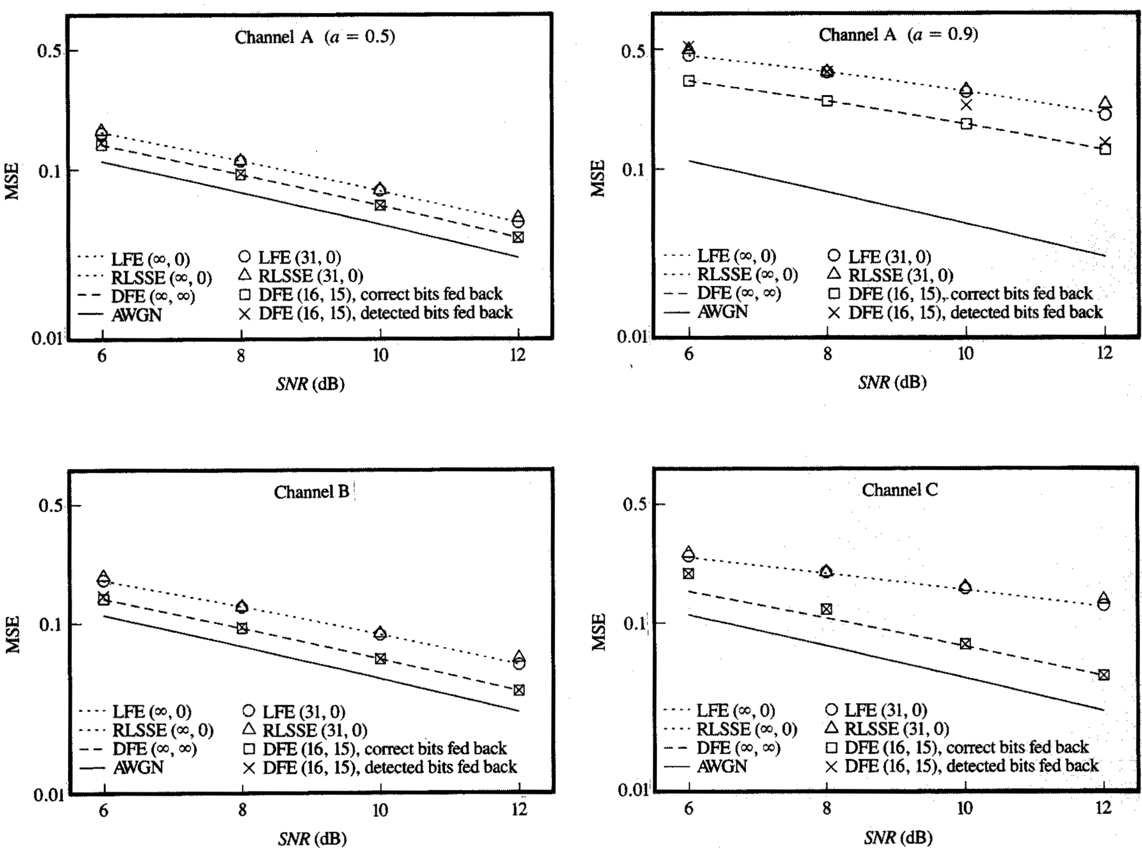

Figure 16

MSE performance of RLSSE(L,0) assuming perfect channel estimates: theoretical predictions vs. test results. The RLSSE(∞,0), LFE(∞,0) and DFE(∞,∞) curves were calculated by numerically integrating Equations (31), (32), and (33), respectively. The symbols represent test measurements. Note that the RLSSE(∞,0) and LFE(∞,0) curves (dotted lines) are identical.

as a benchmark. For ease of interpretation, the plots are labeled SNR_{TRAIN} and SNR_{RUN}, denoting the signal-to-noise ratio during train and run modes, respectively. Note that $SNR_{\mathrm{TRAIN}} = -10 \log[2(N_0 + \Delta)]$ and $SNR_{\mathrm{RUN}} = -10 \log(2N_0)$.

- *Probability of bit error*

Bit error rates were also measured during testing.[13] Representative results from these tests are shown in **Figure 18** for the case of perfect channel knowledge and in **Figure 19** for the channel filter mismatch case. By inspecting theoretical and test results, several observations can be made:

- For all test cases, the measured MSE for the linear RLSSE(L, 0) receiver closely approximated the asymptotic predictions.
- The empirical results support the theoretical prediction that $J_{\mathrm{RLSSE}(\infty,0)} \le J_{\mathrm{LFE}(\infty,0)}$ in white Gaussian noise mismatch, with equality only under the idealistic case of no noise mismatch. In fact, the linear RLSSE(31,0) receiver also outperformed the DFE in several of the noise mismatch tests.
- The linear RLSSE is relatively robust. Recall that the LFE and DFE receivers required an accurate estimate of the "peak sample" of the channel impulse response; i.e., they were trained with the optimal delay for each channel. No such delay was used for the linear RLSSE

[13] A theoretical approximation of P_b is introduced in Appendix B.

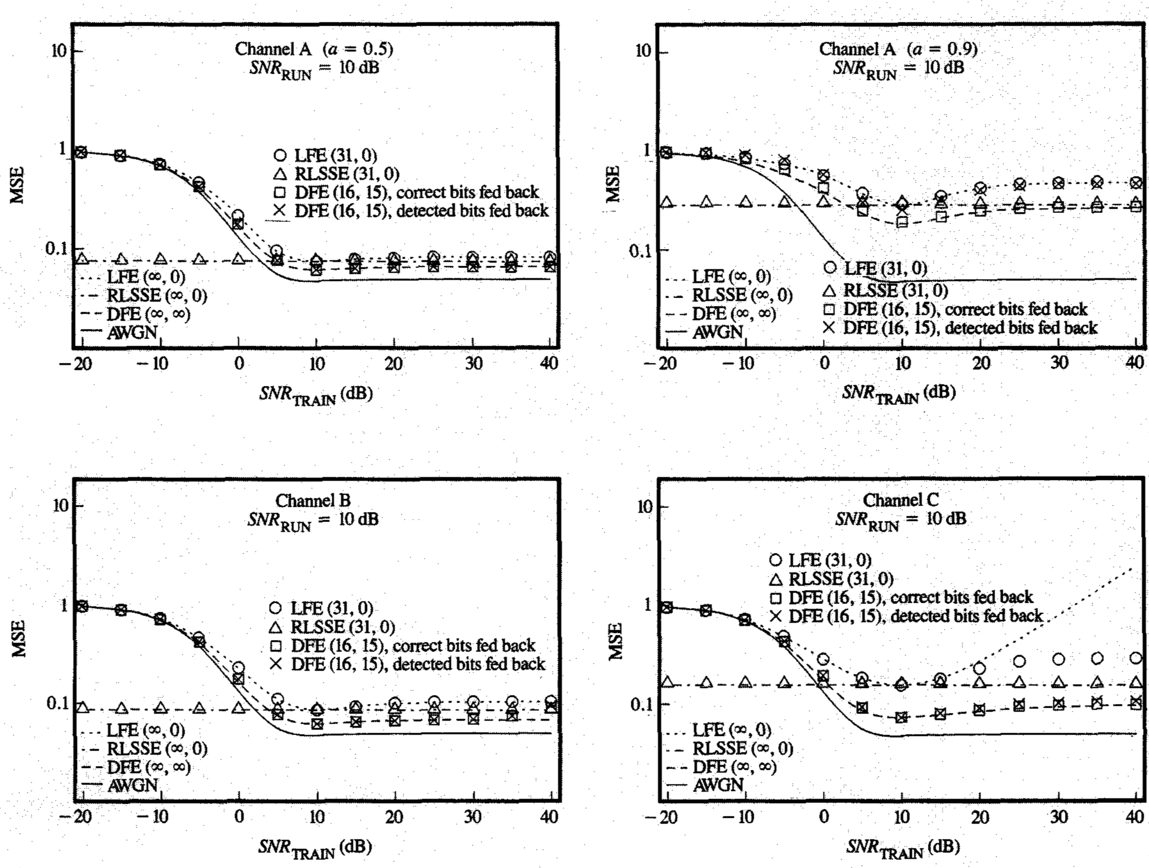

Figure 17

MSE performance of RLSSE(L,0) in white noise mismatch: theoretical predictions vs. test results. The RLSSE(∞,0), LFE(∞,0), and DFE(∞,∞) curves were calculated by numerically integrating Equations (31), (32), and (33), respectively. The symbols represent test measurements. The AWGN curves apply only to LFE and DFE receivers; they show the best MSE possible for either equalizer in the presence of noise mismatch.

receiver. In fact, the linear RLSSE receivers used in all tests were identical, including the initialization of γ and ε.

- For some channels, the performance of the fully constrained nonlinear RLSSE receiver was better than or comparable to the DFE performance. For those channels in which the RLSSE(L,D) could not perform well, it seems that a large number of precursors reduced the effectiveness of the constrained algorithm. This problem was found to be less severe as the constraint parameter D was decreased. However, this led to performance comparable to that of the RLSSE(L,0) receiver. It was also found that modifying parameters γ and ε could improve on the performance of the fully constrained receiver.

Implementation considerations

A number of implementation considerations have been investigated throughout the various tests. Several of the key findings are now summarized.

- *Parameter sensitivity*

There are four basic parameters which must be specified for the generic RLSSE(L,D) receiver—L, D, γ, and ε^{-1} We have found that the filter length L must be at least equal to the channel length in order to achieve adequate performance. Although higher-length filters may improve performance in some cases, the gain is not always appreciable. In fact, there were mismatch scenarios in

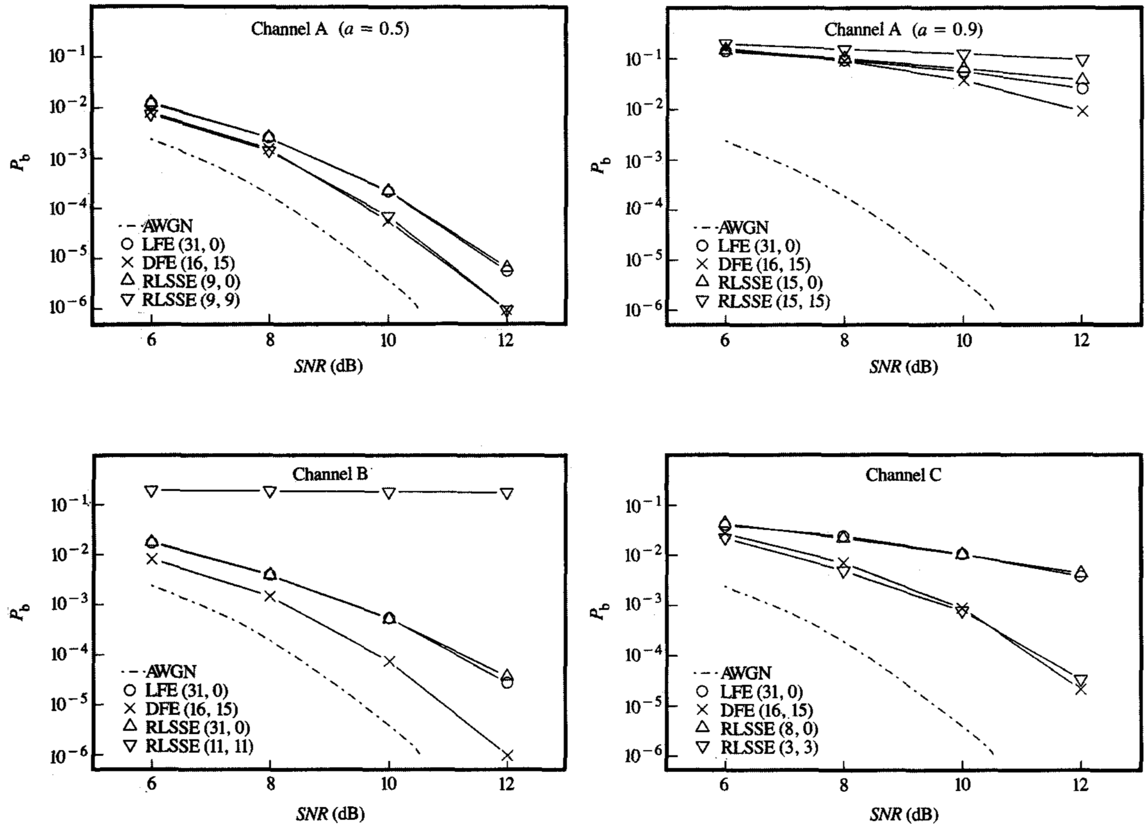

Figure 18

Bit error probability of RLSSE(L, D) assuming perfect channel knowledge. DFE uses actual decision feedback vs. correct decision feedback assumed in theoretical predictions.

which *increasing* the receiver complexity actually *degraded* performance. This *complexity inversion phenomenon*, which was also found to exist in linear, DFE, and MLSE receivers, is further described in [4]. Test results which show the sensitivity of the linear receiver to the length of the RLSSE window L are summarized in **Figure 20**.

From the tests, we have found that the linear receiver is quite robust with regard to γ and ε. In fact, we used constant values of $\gamma = 1.0$ and $\varepsilon^{-1} = 15$ for all test channels. Other values outside these nominal values were also used, but there was no appreciable gain or loss in performance. It is also clear that the channel characteristics (e.g., degree of nonminimum phase) play a key role in the optimal selection of D. Although the optimization of D was not theoretically addressed, we found that the extreme values $D = 0$ and $D = L$ usually led to optimum performance.

- *Exploiting IIR channel models*

Although the RLSSE development is based on a finite-length decoding window, the channel estimator need not be constrained to an FIR model. In fact, for Channel A, which is a one-pole IIR channel, we found it useful to identify the channel with an IIR model and truncate the model to a suitable length before presenting it to the RLSSE decoder. The advantage of this method (for this channel) is rapid convergence, which occurs because only one parameter must be estimated. A disadvantage of this approach is the biased estimate which occurs at extremely

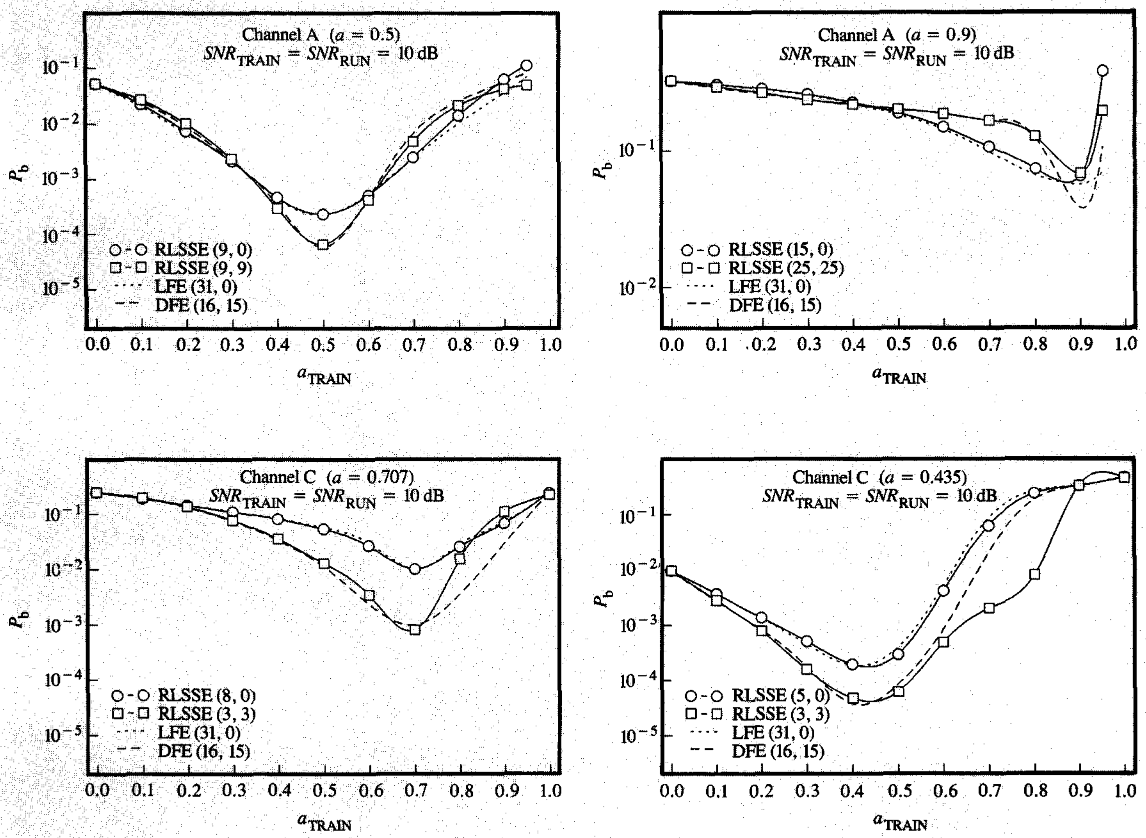

Figure 19

Bit error probability of RLSSE(L,D) in the presence of channel filter mismatch. DFE uses actual decision feedback vs. correct decision feedback assumed in theoretical predictions.

low SNR levels. A comparison of the FIR- and the IIR-trained receivers is shown in **Figure 21**. Note that since the FIR approach yields unbiased estimates, the performance is insensitive to noise mismatch. The IIR approach resulted in a biased estimate; hence, the receiver was in a channel mismatch condition at low SNR.

- *Steady-state implementation*

The generic RLSSE(L,D) receiver shown in Figure 9 has complexity $O(L^2)$, where L is the width of the RLSSE window. This quadratic complexity is quite promising in light of the exponential complexity, i.e., $O(2^{L-1})$, exhibited by most MLSE schemes. However, more efficient RLSSE implementations are possible.

If the channel impulse response is stationary, or, for that matter, if only block-adaptive training is used, the gain vector converges to a steady-state solution. The important control-theoretic proof of this convergence, shown in [11] for the KFE receiver, can be adopted here under the restriction of stationarity and ergodicity. In fact, this convergence was found to be extremely rapid—roughly three to four times the channel length. It was also found that the gain vector is not a function of D; hence, both linear and nonlinear schemes can use the same steady-state gain vector. **Figure 22** illustrates the fast convergence of the gain vector for the RLSSE(3,0) and the fully constrained RLSSE(3,3) receiver over Channel C.

- *Stability*

The numerical stability of both conventional and fast RLS algorithms has had considerable attention in the literature. This potential instability has been attributed primarily to

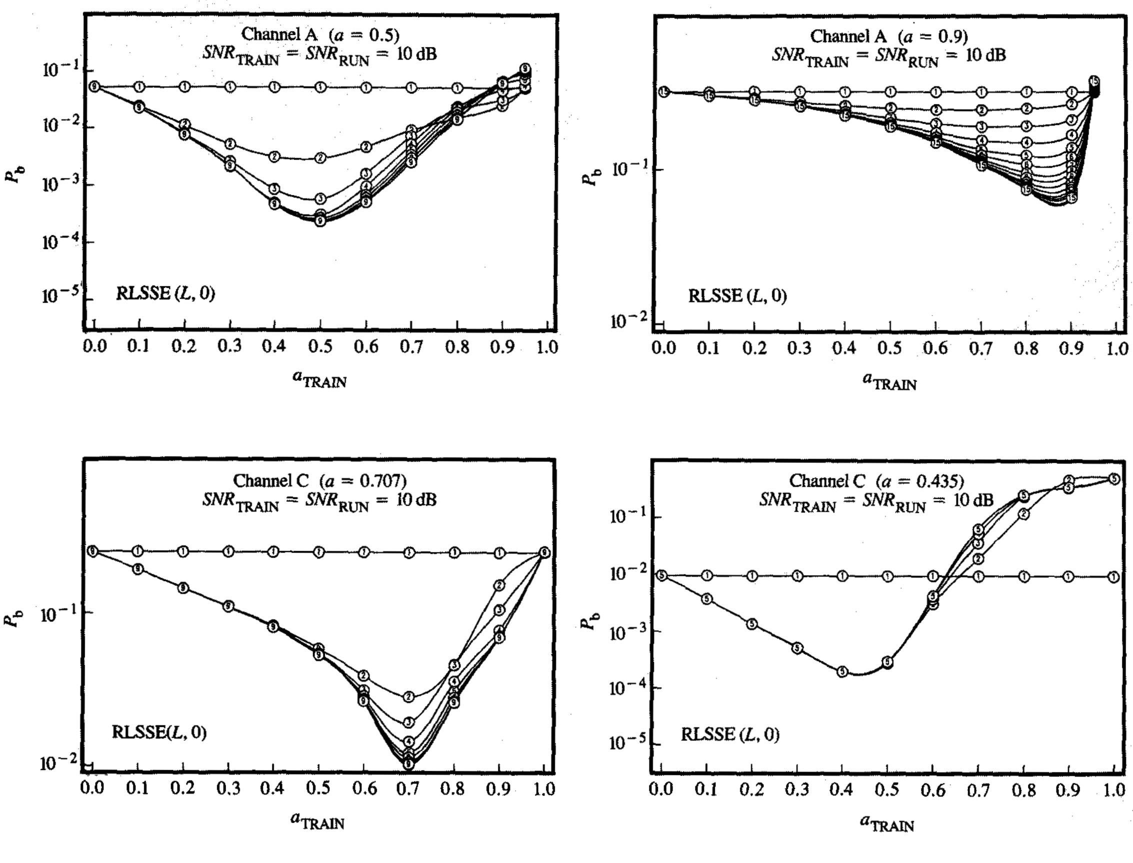

Figure 20

Impact of complexity on performance of RLSSE(L,0) receiver in the presence of channel filter mismatch.

the propagation of numerical imprecision when updating $\mathbf{P}_k$ [32–34]. As noted in the RLSSE development, the $\mathbf{P}_k$ matrix is partially reinitialized with every iteration. Hence, its stability—with respect to numerical precision—is ensured irrespective of the exponential weighting, since the value of any element of $\mathbf{P}_k$ can be calculated as a finite sum of operations on the initialization matrix $\varepsilon^{-1}\mathbf{gI}$.

- *Numerical precision*

The numerical precision shown in the simulation results was based on 64-bit floating-point arithmetic. Although we did not investigate the effects of precision below 32 bits, we feel that the impact of this should not be significant, because of the inherent stability created by constant reinitialization of $\mathbf{P}_k$. Certainly, more precise RLS schemes such as square-root factorizations [35] could be implemented if available.

- *Using variants of the RLS algorithm*

If the channel impulse response is time-varying and requires continuous channel estimation (in a decision-directed mode), the gain vector does not reach a steady state; hence, $\mathbf{P}_k$ must be calculated on a continuous basis. Fast RLS algorithms [36–38] have been introduced to reduce the complexity of the standard RLS algorithm from $O(N^2)$ to $O(N)$. Although these schemes have typically been plagued by instability, it is believed that the RLSSE receiver can exploit these fast RLS algorithms and guarantee stability because of the continuous reinitialization of $\mathbf{P}_k$ described in (29). It should be noted

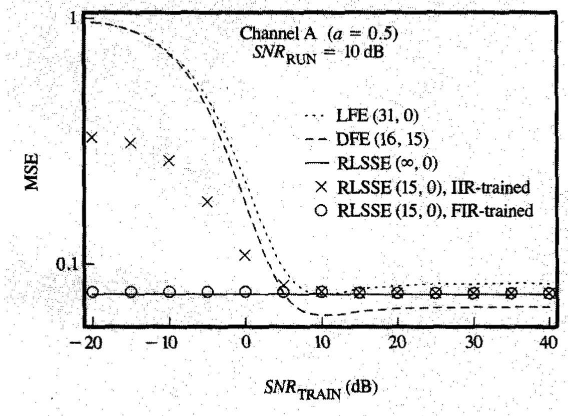

Figure 21

Impact of FIR vs. IIR channel model on RLSSE(L,0) performance.

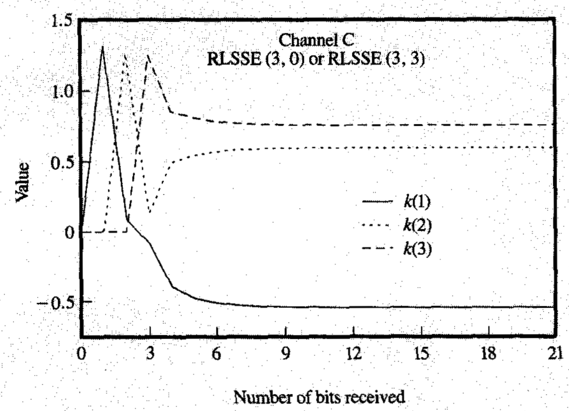

Figure 22

Convergence of RLSSE gain vector $\mathbf{k}_k$.

that employing a fast RLS algorithm in our RLSSE receiver would be extremely advantageous, since the complexity would be $O(N)$ for both time-varying and time-invariant channels.

Several variants which were tested with some success included reinitializing the entire covariance matrix during each iteration, guessing at the newest (unequalized) received symbol, and adaptively switching to a steady-state mode once the gain vector converges.

- *Combined channel/sequence estimator*

Since the core RLS algorithm is identical for both the channel estimator and the sequence estimator, the same software can be used for both. In fact, the receiver used in our tests invoked the same software procedure. Furthermore, since there have been many advances in hardware implementations of RLS algorithms, we expect that combining the channel and sequence estimator in a single chip could be achieved if the RLSSE algorithm were to be used.

Summary

We have proposed a new family of algorithms based on recursive least-squares estimation for the reception of digital signals over channels with intersymbol interference (ISI) and white Gaussian noise. For comparison, the asymptotic predicted performance of the linear version is shown in **Table 6** along with the asymptotic performance of linear and decision feedback equalizers. The analogous predictions for the case of channel filter mismatch in white Gaussian noise are shown in **Table 7.**

Theoretical predictions and test results have indicated that the performance of a software-implemented version of the RLSSE(L, 0) receiver is comparable to that of standard linear equalizers under no-mismatch (ideal) conditions. When noise mismatch was present, the performance far exceeded that of LFE receivers as well as the DFE receiver in some severe cases of mismatch.

While the linear and DFE receivers required tedious optimization of the training delay, no such delay was necessary for the software-implemented version of the linear RLSSE(L, 0) receiver. Its robustness with respect to the algorithm parameters—L, ε, and γ—was very good.

For several channels with minimal precursors, the software-implemented version of the fully constrained RLSSE(L,L) receiver was comparable to a DFE receiver in performance under ideal conditions, and substantially better in the presence of noise mismatch over those channels. The optimization of D, however, must be further addressed for broad applicability of the nonlinear RLSSE(L,D) receiver. Promising areas of research include joint optimization of the parameters (L, D, ε, and γ), incorporating a delay in the channel-estimation process, and incorporating prefilters to mitigate precursors.

The same core algorithm was used for both training and decoding, suggesting that it should be possible to implement a hardwired RLSSE receiver in an efficient manner. Since incorporating both channel estimator and decoder on a single chip should be possible with RLSSE receivers, the use of the RLSSE family of algorithms should be beneficial for applications such as hand-held communications, multimedia, wireless networks, and other size/weight/cost-constrained systems.

Table 6 Theoretical MSE performance of RLSSE, LFE, and DFE in white Gaussian noise mismatch.

Receiver	*Mean square error*
RLSSE(∞, 0)	$\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0}\, d\omega$
LFE(∞, 0)	$\frac{1}{2\pi}\int_{-\pi}^{\pi} \left[\frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0} + \frac{\Delta^2\lvert H(e^{j\omega})\rvert^2}{[\lvert H(e^{j\omega})\rvert^2 + N_0 + \Delta]^2[\lvert H(e^{j\omega})\rvert^2 + N_0]}\right] d\omega$
DFE(∞, ∞)	$\exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left[\frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0} + \frac{\Delta^2\lvert H(e^{j\omega})\rvert^2}{[\lvert H(e^{j\omega})\rvert^2 + N_0 + \Delta]^2[\lvert H(e^{j\omega})\rvert^2 + N_0]}\right] d\omega\right\}$

Assumptions:
1. Source is uncorrelated sequence with unit power.
2. Noise is AWGN with variance $N_0 + \Delta$ during training and N_0 during decoding.
3. MSE for DFE(∞, ∞) receiver assumes correct past decisions.

Table 7 Theoretical MSE performance of RLSSE, LFE, and DFE in channel filter and white Gaussian noise mismatch.

Receiver	*Mean square error*
RLSSE(∞, 0)	$\frac{1}{2\pi}\int_{-\pi}^{\pi} \left[\frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0} + \frac{\lvert[\lvert H(e^{j\omega})\rvert^2 + N_0]H_T(e^{j\omega}) - [\lvert H_T(e^{j\omega})\rvert^2 + N_0]H(e^{j\omega})\rvert^2}{[\lvert H_T(e^{j\omega})\rvert^2 + N_0]^2[\lvert H(e^{j\omega})\rvert^2 + N_0]}\right] d\omega$
LFE(∞, 0)	$\frac{1}{2\pi}\int_{-\pi}^{\pi} \left[\frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0} + \frac{\lvert[\lvert H(e^{j\omega})\rvert^2 + N_0]H_T(e^{j\omega}) - [\lvert H_T(e^{j\omega})\rvert^2 + N_0 + \Delta]H(e^{j\omega})\rvert^2}{[\lvert H_T(e^{j\omega})\rvert^2 + N_0 + \Delta]^2[\lvert H(e^{j\omega})\rvert^2 + N_0]}\right] d\omega$
DFE(∞, ∞)	$\exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left[\frac{N_0}{\lvert H(e^{j\omega})\rvert^2 + N_0} + \frac{\lvert[\lvert H(e^{j\omega})\rvert^2 + N_0]H_T(e^{j\omega}) - [\lvert H_T(e^{j\omega})\rvert^2 + N_0 + \Delta]H(e^{j\omega})\rvert^2}{[\lvert H_T(e^{j\omega})\rvert^2 + N_0 + \Delta]^2[\lvert H(e^{j\omega})\rvert^2 + N_0]}\right] d\omega\right\}$

Assumptions:
1. Source is uncorrelated sequence with unit power.
2. Noise is AWGN with variance $N_0 + \Delta$ during training and N_0 during decoding. Channel filter spectrum is $H_T(e^{j\omega})$ and $H(e^{j\omega})$ during training and decoding, respectively.
3. MSE for DFE(∞, ∞) receiver assumes correct past decisions.

Appendix A: Derivation of the RLS algorithm

The RLS algorithm can be derived by use of the batch least-squares estimator, repeated here:

$$\tilde{\mathbf{c}}_N^{LS} = [\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N]^{-1}\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N . \qquad (15)$$

First, we exploit the diagonal nature of the weighting matrix in (15), $\mathbf{W}_N$, to obtain the recursive form,

$$[\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N] = \sum_{k=n}^{N} \mathbf{q}_k\gamma^{N-k}\mathbf{q}_k^{\mathrm{T}}$$

$$= \sum_{k=n}^{N-1} \mathbf{q}_k\gamma^{N-k}\mathbf{q}_k^{\mathrm{T}} + \mathbf{q}_N\gamma^{N-N}\mathbf{q}_N^{\mathrm{T}}$$

$$= \sum_{k=n}^{N-1} \mathbf{q}_k\gamma\gamma^{N-k-1}\mathbf{q}_k^{\mathrm{T}} + \mathbf{q}_N\mathbf{q}_N^{\mathrm{T}}$$

$$= \gamma\mathbf{Q}_{N-1}^{\mathrm{T}}\mathbf{W}_{N-1}\mathbf{Q}_{N-1} + \mathbf{q}_N\mathbf{q}_N^{\mathrm{T}} . \qquad (\mathrm{A1})$$

For notational convenience[14], we define the inverse term in (15) as $\mathbf{P}_N$,

$$\mathbf{P}_N = [\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{Q}_N]^{-1}, \qquad (\mathrm{A2})$$

and by using this simplification in (A1), we yield

[14] In addition to notational convenience, $\mathbf{P}_N^{-1} = \mathbf{Q}_N^{T}\mathbf{W}_N\mathbf{Q}_N$ represents the time-averaged autocorrelation matrix. Furthermore, as is mentioned by Proakis [2] and elaborated by Bierman [35], this matrix is in general non-Toeplitz and may be ill-conditioned.

$$\mathbf{P}_N^{-1} = \gamma \mathbf{P}_{N-1}^{-1} + \mathbf{q}_N \mathbf{q}_N^{\mathrm{T}}. \tag{A3}$$

To remove the inverse operator, a simplified version of the matrix inversion lemma [19],

$$(\mathbf{A} + \mathbf{BC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(1 + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1} \tag{A4}$$

is applied by making the substitutions

$$\mathbf{A} = \gamma \mathbf{P}_{N-1}^{-1},$$

$$\mathbf{B} = \mathbf{q}_N,$$

$$\mathbf{C} = \mathbf{q}_N^{\mathrm{T}}. \tag{A5}$$

After these substitutions, the following recursive equation is found for $\mathbf{P}_N$:

$$\mathbf{P}_N = \gamma^{-1}\mathbf{P}_{N-1} - \frac{\gamma^{-1}\mathbf{P}_{N-1}\mathbf{q}_N}{[1 + \gamma^{-1}\mathbf{q}_N^{\mathrm{T}}\mathbf{P}_{N-1}\mathbf{q}_N]} \mathbf{q}_N^{\mathrm{T}} \gamma^{-1}\mathbf{P}_{N-1}. \tag{A6}$$

A further simplification results from defining the *gain vector*,

$$\mathbf{k}_N = \frac{\gamma^{-1}\mathbf{P}_{N-1}\mathbf{q}_N}{[1 + \gamma^{-1}\mathbf{q}_N^{\mathrm{T}}\mathbf{P}_{N-1}\mathbf{q}_N]}, \tag{A7}$$

so that we can rewrite $\mathbf{P}_N$ as

$$\mathbf{P}_N = \gamma^{-1}[\mathbf{I} - \mathbf{k}_N\mathbf{q}_N^{\mathrm{T}}]\mathbf{P}_{N-1}, \tag{A8}$$

where $\mathbf{I}$ is the appropriately dimensioned identity matrix.

Now consider the nonbracketed factor in (15). This term, which represents the time-averaged cross-correlation vector, can be rewritten as

$$\mathbf{Q}_N^{\mathrm{T}}\mathbf{W}_N\mathbf{r}_N = \gamma \mathbf{Q}_{N-1}^{\mathrm{T}}\mathbf{W}_{N-1}\mathbf{r}_{N-1} + \mathbf{q}_N r_N. \tag{A9}$$

Finally, by calculating the *innovation term*[15],

$$\eta_N = r_N - \mathbf{q}_N^{\mathrm{T}}\tilde{\mathbf{c}}_{N-1}, \tag{A10}$$

and substituting these results in (15), we obtain the recursive weighted least-squares estimator for $\mathbf{c}$,

$$\tilde{\mathbf{c}}_N = \tilde{\mathbf{c}}_{N-1} + \mathbf{k}_N\eta_N. \tag{A11}$$

Appendix B: Approximating the probability of bit error

The MSE analyses and measurements described in this paper and in [4] led to satisfying conclusions, since both the theoretical and practical results were in close agreement. Recall, however, that the P_{b} performance curves shown so far were based only on test measurements. While test measurements are indeed the only means of accurately assessing P_{b} for arbitrary channels and mismatch conditions, it is important to provide a theoretical justification for the P_{b} results. In this appendix, an approximation of P_{b} for the LFE, DFE, and linear RLSSE receivers is obtained as a function of their respective MSE performance in arbitrary channel mismatch conditions.

Under practical (finite-complexity) conditions, the output of any ISI receiver generally contains residual ISI which is *not* Gaussian. Even with this residual ISI, it is possible and useful to determine the SNR of the (unconstrained) receiver output [2, 39].

If the total MSE at the output of the receiver (prior to the detection device) is denoted by J, it is clear that the output signal-to-noise ratio $\mathrm{SNR}_{\mathrm{out}}$ is given [2] by

$$\mathrm{SNR}_{\mathrm{out}} = \frac{1 - J}{J}. \tag{A12}$$

Now, assume that the unconstrained output is passed through a simple threshold device.[16] If we assume that both the residual ISI and the noise components of the unconstrained output are Gaussian, the error probability can be approximated by

$$P_{\mathrm{b}} \approx \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{\mathrm{SNR}_{\mathrm{out}}}{2}}\right)$$

$$= \frac{1}{2}\,\mathrm{erfc}\left[\sqrt{\frac{1}{2}\left(\frac{1 - J}{J}\right)}\right], \tag{A13}$$

where

$$\mathrm{erfc}(\beta) \triangleq \frac{2}{\sqrt{\pi}} \int_{\beta}^{\infty} e^{-t^2}\, dt. \tag{A14}$$

Note that Equation (A13) can be used for the LFE, DFE, or linear RLSSE receiver by simply replacing J in (A13) with the appropriate formula in Table 7.

The theoretical P_{b} approximation shown above was compared against the P_{b} measured via Monte Carlo simulation for several test channels under various degrees of mismatch. These comparisons are shown in **Figure 23**. As can be seen, the theoretical approximation supports the experimental data quite well in many regions of interest. Furthermore, there are regions in which the fit could be tightened by increasing the receiver complexity.

Acknowledgment

The author is grateful to Prof. John B. Anderson of Rensselaer Polytechnic Institute, who provided a valuable critique of early versions of the manuscript.

[15] Note that the innovation term here represents the *a priori* estimation error, since it is based on the previous channel estimate, whereas the batch least-squares development utilized the *a posteriori* estimation error $e(k)$, which was based on a current channel estimate.

[16] If the soft output is followed by an *optimal* decision device (i.e., a sequence estimator) which attempts to eliminate the residual ISI, the analysis would follow an error-state approach to bound the error event probability [39, 40], which cannot make direct use of the MSE formulas of Table 7.

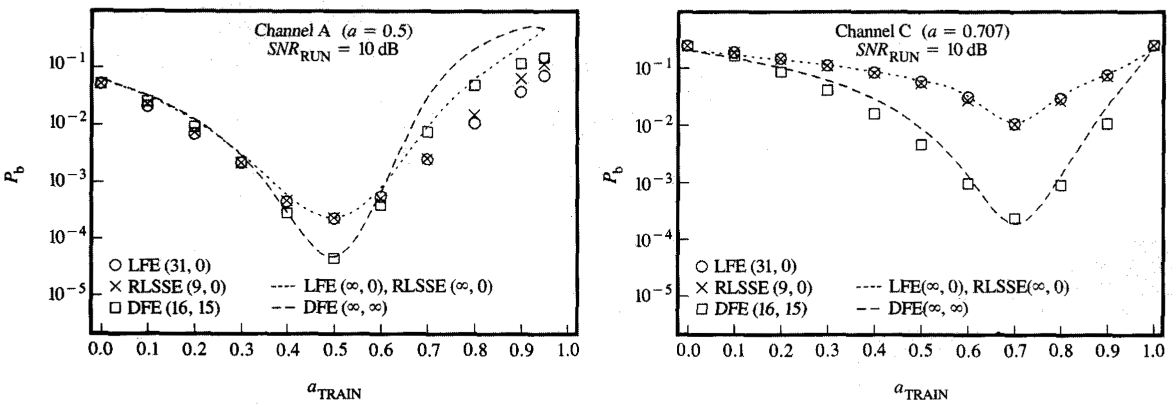

Figure 23

P_b performance in channel filter mismatch: theoretical vs. test results (Channels A, C). RLSSE(∞,0), LFE(∞,0), and DFE(∞,∞) curves were calculated by first numerically integrating the appropriate equation of Table 7 to find J, and then applying (A13). Symbols represent test measurements.

References

1. S. Qureshi, "Adaptive Equalization," *Proc. IEEE* **73,** No. 9, 1349–1387 (September 1985).
2. J. G. Proakis, *Digital Communications*, McGraw-Hill Book Co., Inc., New York, 1989.
3. C. A. Belfiore and J. H. Park, "Decision Feedback Equalization," *Proc. IEEE* **67,** 1143–1156 (August 1979).
4. F. Gozzo, "Robust Sequence Estimation in the Presence of Channel Mismatch," Ph.D. Dissertation, Rensselaer Polytechnic Institute, Troy, NY, May 1992.
5. F. Gozzo, "System and Method of Robust Sequence Estimation in the Presence of Channel Mismatch," U.S. Patent Pending, No. EN992026, April 10, 1992.
6. R. E. Lawrence and H. Kaufman, "The Kalman Filter for the Equalization of a Digital Communication Channel," *IEEE Trans. Commun.* **COM-19,** No. 6, 1137–1141 (December 1971).
7. J. W. Mark, "A Note on the Modified Kalman Filter for Channel Equalization," *Proc. IEEE* **61,** 481–482 (April 1973).
8. S. B. Kleibanov, V. B. Privalskii, and I. V. Time, "Kalman Filter for Equalization of Digital Communications Channel," *Automat. Remote Control* **35,** Pt. 1, 1097–1102 (1974) (English translation).
9. A. Luvison and G. Pirani, "A State-Variable Approach to the Equalization of Multilevel Digital Signals," *CSELT Rapporti Tecnici*, No. 3, 3–22 (April 1974).
10. A. Luvison and G. Pirani, "Design and Performance of an Adaptive Kalman Receiver for Synchronous Data Transmission," *IEEE Trans. Aero. Elec. Syst.* **AES-15,** No. 5, 635–648 (September 1979).
11. S. Benedetto and E. Biglieri, "On Linear Receivers for Digital Transmission Systems," *IEEE Trans. Commun.* **COM-22,** No. 9, 1205–1214 (September 1974).
12. B. Mulgrew and C. F. N. Cowan, "An Adaptive Kalman Equalizer: Structure and Performance," *IEEE Trans. Acoust., Speech, & Signal Proc.* **35,** No. 12, 1727–1735 (December 1987).
13. S. Prasad and S. S. Pathak, "An Approach to Recursive Equalization via White Sequence Estimation Techniques," *IEEE Trans. Commun.* **COM-35,** No. 10, 1037–1040 (October 1987).
14. J. M. Mendel, "White-Noise Estimators for Seismic Data Processing in Oil Exploration," *IEEE Trans. Automat. Contr.* **AC-22,** 694–706 (October 1977).
15. M. K. Yurtseven and M. M. Kumar, "An Adaptive FIR Kalman Equalizer with Improved Convergence Properties," *Proc. Ann. Pitts. Conf. Computer Sim.*, Vol. 21, Pt. 2, 507–513 (1990).
16. E. Delle Mese and G. Corsini, "Adaptive Kalman Filter Equalizer," *Electron. Lett.* **16,** No. 14, 547–549 (July 3, 1980).
17. G. W. Davidson, D. D. Falconer, and A. U. H. Sheikh, "An Investigation of Block-Adaptive Decision Feedback Equalization for Frequency Selective Fading Channels," *Can. J. Elec. Comp. Eng.* **13,** No. 3–4, 106–111 (1988).
18. E. Eleftheriou and D. D. Falconer, "Tracking Properties and Steady-State Performance of RLS Adaptive Filter Algorithms," *IEEE Trans. Acoust., Speech, & Signal Proc.* **ASSP-34,** No. 5, 1097–1109 (October 1986).
19. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1986.
20. J. G. Proakis and F. Ling, "Recursive Least Squares Algorithms for Adaptive Equalization of Time-Variant Multipath Channels," *Proc. IEEE ICC*, Amsterdam, Netherlands, May 1984, pp. 1250–1254.
21. K. Zhou, J. G. Proakis, and F. Ling, "Decision Feedback Equalization of Time-Dispersive Channels with Coded Modulation," *IEEE Trans. Commun.* **38,** No. 1, 18–24 (January 1990).
22. R. A. Ziegler and J. M. Cioffi, "A Comparison of Least Squares and Gradient Adaptive Equalization for Multipath Fading in Wideband Digital Mobile Radio," *Proc. IEEE GLOBECOM*, 1989, pp. 102–106.
23. J. J. Shynk, "Adaptive IIR Filtering," *IEEE ASSP Magazine*, pp. 4–21 (April 1989).

24. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, John Wiley & Sons, New York, 1968.
25. G. Ungerboeck, "Nonlinear Equalization of Binary Signals in Gaussian Noise," *IEEE Trans. Commun.* **19,** No. 6, 1128–1137 (December 1971).
26. P. R. Chevillat and E. Eleftheriou, "Decoding of Trellis-Encoded Signals in Presence of Intersymbol Interference and Noise," *IEEE Trans. Commun.* **37,** No. 7, 669–676 (July 1989).
27. V. M. Eyuboglu and S. Qureshi, "Reduced-State Sequence Estimation for Coded Modulation on ISI Channels," *J. Select. Areas Commun.* **7,** No. 6 (August 1989).
28. J. B. Anderson, "Limited Search Trellis Decoding of Convolutional Codes," *IEEE Trans. Info. Theory* **35,** No. 5, 944–955 (September 1989).
29. A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
30. H. W. Sorenson, *Parameter Estimation: Principles and Problems*, Marcel Dekker, New York, 1980.
31. J. Salz, "Optimum Mean-Square Decision Feedback Equalization," *Bell Syst. Tech. J.* **52,** No. 8, 1341-1373 (October 1973).
32. S. Chennakeshu, A. Narasimhan, and J. B. Anderson, "Decision Feedback Equalization for Digital Cellular Radio," *Proc. IEEE ICC*, 1990.
33. J. M. Cioffi, "Limited Precision Effects in Adaptive Filtering," *IEEE Trans. Circuits & Syst.* **CAS-34,** No. 7, 821–833 (July 1987).
34. L. Ljung and S. Ljung, "Error Propagation Properties of Recursive Least-Squares Adaptation Algorithms," *Automatica* **21,** 157–167 (1985).
35. G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, Inc., New York, 1977.
36. S. H. Ardalan and L. J. Faber, "A Fast ARMA Transversal RLS Filter Algorithm," *IEEE Trans. Acoust., Speech, & Signal Proc.* **36,** No. 3, 349–358 (March 1988).
37. J. M. Cioffi and T. Kailath, "Fast Recursive-Least Squares Transversal Filters for Adaptive Filtering," *IEEE Trans. Acoust., Speech, & Signal Proc.* **ASSP-32,** 304–337 (April 1984).
38. D. T. M. Slock and T. Kailath, "Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering," *IEEE Trans. Signal Proc.* **39,** No. 1, 92–114 (January 1991).
39. J. W. Modestino and V. M. Eyuboglu, "Integrated Multielement Receiver Structures for Spatially Distributed Interference Channels," *IEEE Trans. Info. Theory* **IT-32,** No. 2, 195–219 (March 1986).
40. D. Divsalar, "Performance of Mismatched Receivers on Bandlimited Channels," Ph.D. Dissertation, University of California at Los Angeles, 1978.

Received May 4, 1993; accepted for publication December 2, 1993

Francesco Gozzo *IBM Federal Systems Company, 1801 Route 17C, Owego, New York 13827 (GOZZO at OWGVM3, gozzo@vnet.ibm.com).* Dr. Gozzo received B.S., M.E., and Ph.D. degrees in electrical engineering from the Rensselaer Polytechnic Institute, Troy, New York, in 1984, 1985, and 1992, respectively. In 1985, he joined the IBM Federal Systems Company at the Owego facility, where he has held key technical roles in various research and development programs for military applications—including programs on radar, sonar, communication, and tactical weapons systems. In 1989, he received an IBM Resident Study Fellow Award, through which he pursued his doctorate in the area of robust sequence estimation. His current interests are in the areas of optical character recognition, adaptive systems, and intelligent vehicle highway systems. Dr. Gozzo is a member of the IBM Owego Technical Advisory Council; he has received other honors, including the Best Paper Award for the IEEE Southern Tier Technical Conference (1987) and the IBM Technical Leadership Program (1986). He has published numerous papers in areas such as image compression, adaptive equalization, robust estimation, and fuzzy sets, and has several patents pending. He is actively involved in the IEEE, and has served as an Adjunct Lecturer at the State University of New York at Binghamton.