# Chapter 4

# On the Robustness of LMS Filters

Babak Hassibi

Department of Electrical Engineering
California Institute of Technology, Pasadena, CA 91125

**Abstract**

Classical methods in estimation theory (such as maximum-likelihood, maximum entropy and least-mean-squares) require exact models and a priori knowledge of the statistics of the exogenous signals. In many applications, however, one is faced with model uncertainties and lack of statistical information. Therefore, the introduction of the LMS algorithm by Widrow and Hoff in 1960 came as a significant development for a broad range of engineering applications since the LMS adaptive filter requires no advance knowledge of the signal statistics. The LMS algorithm was originally conceived as an approximate recursive solution to a stochastic least-squares problem. Although exact recursive solutions can be obtained (these are the recurisive least-squares (RLS) algorithm and its variants), they are computationally more complex, as well as less robust to disturbance variation. For a long time, a satisfactory explanation for the robustness properties of the LMS filter was lacking.

In the early 1980's researchers in control theory began to bring robustness issues into the center stage of control system design. The ensuing theory, now referred to as $H^\infty$, had the objective of designing controllers that were less sensitive to modeling errors and disturbance variation. Following the success of $H^\infty$ control, a parallel theory for $H^\infty$ estimation was also developed. Due to the similarity between the objectives of adaptive filtering and $H^\infty$ estimation one may suspect a connection between the two. In this chapter, we make this connection explicit by showing that the LMS algorithm is $H^\infty$-optimal, a statement that will be made precise. This result has several ramifications. First, it furnishes LMS with a rigorous explanation of its inherent robustness. Second, it provides a framework for studying the robustness of other well known adaptive filtering algorithms. Third, it allows for certain generalizations and extensions. Fourth, and finally, it points to an inherent connection between "adaptation" and "robustness", a connection that is well worthy of further scrutiny. We will review all these, and their various interconnections, in this chapter.

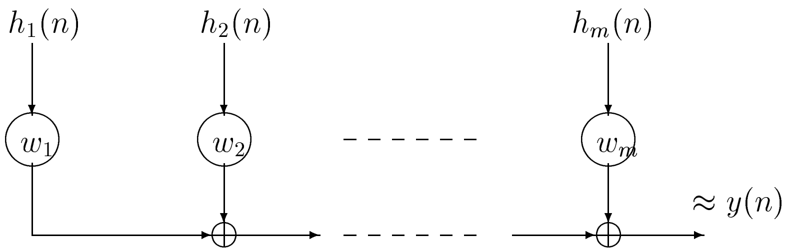


Figure 4.1: The basic adaptive filtering model.

## 4.1 Introduction

A central problem in adaptive signal processing is the following: given a set of input-vector and output-scalar pairs[1] $\{(\mathbf{h}(n), y(n))\}$ find a model that "well represents" this set of data. Well represents is often meant in a predictive fashion, i.e., if the model is presented with a *new* input vector $\mathbf{h}$ then it should be able to provide a good prediction of the output $y$. Among all possible models, arguably the simplest one that can be conceived of is linear, i.e., one for which we assume

$$y(n) \approx \mathbf{h}(n)^T \mathbf{w}, \tag{4.1}$$

for some fixed so-called *weight vector*, $\mathbf{w}$ (see Fig. 4.1). Despite its apparent simplicity, the linear model has broad implications and applies to many different problems and applications. Most often the crucial step in writing the model (4.1) is in determining the input-output pairs $(\mathbf{h}(n), y(n))$ and in determining the nature of the approximation "$\approx$". For example, if we are presented with scalar input-output sequences, or *time series*, $u(n)$ and $y(n)$ then a possible model could be

$$y(n) \approx w_1 u(n) + w_2 u(n-1) + \ldots + w_m u(n-m+1). \tag{4.2}$$

In this case, the input vector (also called the *regressor*) is $\mathbf{h}(n) = \begin{bmatrix} u(n) & u(n-1) & \ldots & u(n-m+1) \end{bmatrix}^T$ and the weight vector $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \ldots & w_m \end{bmatrix}^T$. Models of the form (4.2) are referred to as *finite-impulse-response* (FIR) filters and clearly the crucial parameter that determines the model is the *length* of the filter, $m$ (see Fig. 4.2). There are other examples that can be given, such as infinite-impulse-response

[1] In this chapter we will assume, for simplicity, that the output is a scalar. The more general problem of a vector output can be handled without much further difficulty.

(IIR) filters, however, we shall not go into the details here. We only mention in passing that the model (4.1) is also of central importance because it can be regarded as the linearization of more general nonlinear models (such as neural networks) around a suitable operating point.

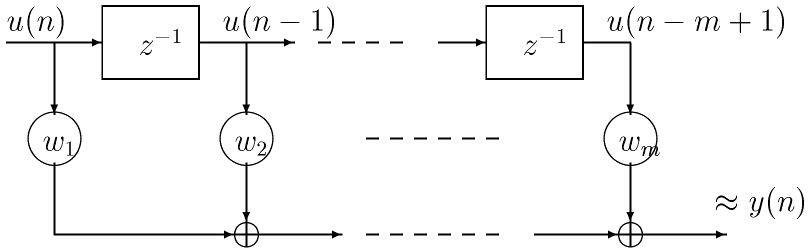


Figure 4.2: An FIR adaptive filtering model.

In any event, once the model (4.1) has been constructed, the problem is to determine the best weight vector that describes the relationship between the inputs $\{\mathbf{h}(n)\}$ and the outputs $\{y(n)\}$. The question, of course, is in what sense do we mean "best"? A reasonable choice is to have $\mathbf{h}(n)^T\mathbf{w}$ match $y(n)$ in a least-mean-squares sense, i.e., to choose $\mathbf{w}$ according to the criterion

$$\min_{\mathbf{w}} E\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2, \tag{4.3}$$

where $E$ denotes expectation. This requires us to assume that the $\mathbf{h}(n)$ and $y(n)$ are random, in fact that they are (vector-valued) *random processes* (since they are random time series). In order to evaluate the expectation in (4.3) we need to know what the statistics of $\mathbf{h}(n)$ and $y(n)$ are. Here the standard assumption is that $\mathbf{h}(n)$ and $y(n)$ are *jointly wide-sense stationary* random processes with second-order statistics:

$$E\mathbf{h}(n)\mathbf{h}(n)^T = \mathbf{R} \ , \ E\mathbf{h}(n)y(n) = \mathbf{p} \ , \ Ey(n)^2 = r_y. \tag{4.4}$$

Here $\mathbf{R}$ is the autocorrelation matrix of $\mathbf{h}(n)$ and $\mathbf{p}$ is the cross-correlation vector between $\mathbf{h}(n)$ and $y(n)$. Note that the wide-sense stationarity assumption implies that $\mathbf{R}$, $\mathbf{p}$ and $r_y$ are time-independent. In terms of these vectors and matrices, the mean-square-error can be evaluated to be $r_y - \mathbf{p}^T\mathbf{w} - \mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}$, and hence the choice of $\mathbf{w}$ that minimizes this error is

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}. \tag{4.5}$$

For historical reasons, $\mathbf{w}_o$ is called the *Wiener solution*. [1, 2].

Computing the Wiener solution requires knowledge of the second-order statistics of the signals involved. But in adaptive signal processing applications this is not available. (If indeed it were, then there would be no need to provide the data $\{(\mathbf{h}(n), y(n))\}$!) The whole premise is to adapt to the statistics of the signals involved.

### 4.1.1 The LMS Algorithm

Therefore the pioneering work of Widrow and Hoff came as a breakthrough since it provided a recursive way of approximately solving (4.3) without knowledge of the statistics of the signals involved [3, 1, 2]. Since the statistics of the signals are not known, the expectation in (4.3) cannot be explicitly performed. Nor can the gradient of the cost function be computed. However, the key observation of Widrow and Hoff was that using the instantaneous value of the squared error, $\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2$, rather than its unknown mean, one can come up with an estimate of the gradient function via differentiation with respect to $\mathbf{w}$. This so-called instantaneous gradient is given by $-\mathbf{h}(n)\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)$, and so the algorithm updates estimates of the weight vector along the negative direction of the instantaneous gradient:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mu\mathbf{h}(n)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right), \tag{4.6}$$

where $\hat{\mathbf{w}}(n)$ denotes the estimate of the weight vector at time $n$, and $\mu > 0$ is the so-called *learning rate* or *step size*.

This is the celebrated LMS (least-mean-squares) adaptive algorithm. It plays a central role in adaptive filter theory and, along with its variants, is by far the most widely used adaptive algorithm. The backpropagation algorithm for learning in neural networks, for example, is essentially an extension of LMS to the nonlinear setting since weight updates are also performed in the direction of the negative instantaneous gradient [4, 5]. Choice of the learning rate $\mu$ has a profound influence on the performance, convergence rate, etc., of the LMS filter. Although we shall later say a few things about the choice of $\mu$, it is an extensive topic in itself and will be studied in much greater detail in other chapters of this volume.

The LMS algorithm, as just described, presents a simple approximate solution to (4.3). Nonetheless, one may question the validity of approximating the true gradient by its instantaneous value and therefore question the performance of the recursion (4.6). However, LMS has withstood the test of time, and extensive experience over the past four decades has demonstrated that, over a surprisingly wide range of applications (from biomedical to wireless communications, say), it yields good performance. This may come as a surprise, especially because of the simplicity of the algorithm and the approximations used in its derivation.

### 4.1.2 The RLS Algorithm

It is therefore be natural to ask whether better approximations to the Wiener solution can be found? One plausible approximation is to estimate the unknown statistics $\mathbf{R}$ and $\mathbf{p}$ from the observed data. Assuming we have $N$ data points $\{(\mathbf{h}(n), y(n))\}_{n=1}^{N}$, reasonable estimates can be obtained by simple averaging: $\hat{\mathbf{R}} = \hat{\mathbf{R}}(N) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}(n)\mathbf{h}(n)^T$ and $\hat{\mathbf{p}} = \hat{\mathbf{p}}(N) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}(n)y(n)$. Thus a solution is given by

$$\hat{\mathbf{w}} = \hat{\mathbf{w}}(N) = \left(\sum_{n=1}^{N} \mathbf{h}(n)\mathbf{h}(n)^T\right)^{-1} \sum_{n=1}^{N} \mathbf{h}(n)y(n). \tag{4.7}$$

Of course, of interest is whether $\hat{\mathbf{w}}(N)$ converges to $\mathbf{w}_o$ as $N \to \infty$. When the $\mathbf{h}(n)$ and $y(n)$ are each iid (independent and identically distributed), by virtue of the law of large numbers, $\hat{\mathbf{R}}(N) \to \mathbf{R}$ and $\hat{\mathbf{p}}(N) \to \mathbf{p}$, so that $\hat{\mathbf{w}}(N) \to \mathbf{w}_o$. However, indepence is too strong an assumption for $\mathbf{h}(n)$ and $y(n)$ and is rarely met in practice. (It is. for example, not true of the FIR model (4.2)). Fortunately, the law of large numbers holds under much milder conditions, such as $\mathbf{h}(n)$ and $\mathbf{h}(n+k)$ having vanishing autocorrelation as $k \to \infty$. (This and similar conditions are referred to as *mixing conditions* [6].) Therefore in almost all practical applications it follows that $\hat{\mathbf{w}}(N) \to \mathbf{w}_o$.

Another plausible approximation can be obtained by replacing the expectation in the cost function of (4.3) with its estimated mean based on the data. Thus, we can propose the criterion

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T \mathbf{w}\right)^2. \tag{4.8}$$

Note that, compared to (4.3) which was a *stochastic* least-squares problem, (4.8) is a *deterministic* least-squares problem. Thus, there is no need to assume random processes or to take expectations. The problem (4.8) can be readily solved via a straightforward differentiation, or completion-of-squares. The re-assuring, and not altogether unexpected, result is that the solution to (4.8) is also given by (4.7). Thus, replacing the unknown statistics $\mathbf{R}$ and $\mathbf{p}$ by their data-based averages and replacing the mean-squared error by its data-based average are essentially the same thing and lead to the same solution (4.7).

By all accounts, the solution of (4.7) appears to be a better approximation for solving (4.3) than that of the LMS filter. However, compared to the LMS algorithm which yields a recursive solution, it has the drawback that one needs access to the entire data set in order to compute the solution. Although this is not an issue in some applications (such as system identification), it is crucial in many applications, such as control and communications, where certain decisions must be made in real-time, and so depend on the current estimate of the weight vector. In such applications a recursive solution is a must.

Fortunately, the situation is easily remedied. Note that the solution to (4.8) at time $m \leq i \leq N$ (obtained by setting the upper indeces in the sums to $i$) is given by

$$\hat{\mathbf{w}}(i) = \left(\sum_{n=1}^{i} \mathbf{h}(n)\mathbf{h}(n)^T\right)^{-1} \sum_{n=1}^{i} \mathbf{h}(n)y(n), \tag{4.9}$$

where we have assumed that the matrix appearing in the brackets is invertible (which, incidentally is also why we need $i \geq m$). It is convenient to define the matrix $\mathbf{P}(i) = \left(\sum_{n=1}^{i} \mathbf{h}(n)\mathbf{h}(n)^T\right)^{-1}$ so that

$$\begin{aligned}
\hat{\mathbf{w}}(i) &= \mathbf{P}(i)\sum_{n=1}^{i} \mathbf{h}(n)y(n) \\
&= \left(\mathbf{P}(i-1)^{-1} + \mathbf{h}(i)\mathbf{h}(i)^T\right)^{-1} \left(\sum_{n=1}^{i-1} \mathbf{h}(n)y(n) + \mathbf{h}(i)y(i)\right) \\
&= \left(\mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{h}(i)\mathbf{h}(i)^T\mathbf{P}(i-1)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}\right) \left(\sum_{n=1}^{i-1} \mathbf{h}(n)y(n) + \mathbf{h}(i)y(i)\right) \\
&= \hat{\mathbf{w}}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{h}(i)\mathbf{h}(i)^T}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}\hat{\mathbf{w}}(i-1) + \mathbf{P}(i-1)\mathbf{h}(i)y(i) \\
&\quad -\mathbf{P}(i-1)\mathbf{h}(i)\frac{\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)y(i)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)},
\end{aligned}$$

where in the third step we used the matrix inversion lemma $(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$. Now the last expression readily yields

$$\hat{\mathbf{w}}(i) = \hat{\mathbf{w}}(i-1) + \frac{\mathbf{P}(i-1)\mathbf{h}(i)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}\left(y(i) - \mathbf{h}(i)^T\hat{\mathbf{w}}(i)\right), \tag{4.10}$$

which is the recursion we were pursuing. All that is further needed is a recursion for $P(i)$. But the matrix inversion lemma can again be used to obtain

$$\mathbf{P}(i) = \mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{h}(i)\mathbf{h}(i)^T\mathbf{P}(i-1)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}\mathbf{h}(i)^T\mathbf{P}(i-1). \tag{4.11}$$

Together the recursions (4.10-4.11) constitute what is known as the *recursive least-squares* (RLS) algorithm. It has a long history and its inception goes back to Gauss and Legendre. Due to its similarity to a differential equation first studied by Riccati, the recursion (4.11) is referred to as a *Riccati* recursion. (For an interesting review of these see [7]). As with all recursions, the recursions (4.10-4.11) must be initialized. Here they are initialized with their values at time $m$, i.e., $\hat{\mathbf{w}}(m)$ and $\mathbf{P}(m)$, since this is the first value for which the matrix $\mathbf{P}(m)$ can be well-defined.

### 4.1.3 Some Stochastic Interpretations

The RLS algorithm just described, at each time instant $i$, gives the *exact* solution to the deterministic least-squares problem $\min_{\mathbf{w}} \sum_{n=1}^{i} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2$. We also saw that, under some mild mixing conditions, it converges to the Wiener solution, which is the optimal solution to the stochastic least-squares problem (4.3). However, one may wonder whether the estimate provided by the RLS algorithm at any time instant $i$, and not just its limiting value, has a stochastic interpretation in its own right. It turns out that this is indeed the case.

To this end, recall from (4.1) that the linear model we have assumed is *approximate*. However, we can always make it accurate by adding an appropriate disturbance signal $v(n)$, so that

$$y(n) = \mathbf{h}(n)^T\mathbf{w} + v(n). \tag{4.12}$$

Of course, a more complete description of the model would require us to say something about the properties of $v(n)$. If we are interested in having a stochastic model, then $v(n)$ is assumed to be a stochastic process. In general, however, having $v(n)$ "small" (in some sense) is what constitutes a good model.

#### ML Estimators

A common assumption for the disturbance signal $v(n)$ is that it is a zero-mean, stationary white Gaussian process with variance $\sigma^2$, which is independent of both $\mathbf{w}$ and the $\mathbf{h}(n)$. In other words, each of the $v(n)$ are independent zero-mean $\sigma^2$-variance Gaussian random variables. In this case it is straightforward to write down the conditional probability density function

$$p\left(y(1), \ldots, y(N) \,|\mathbf{w}, \mathbf{h}(1), \ldots, \mathbf{h}(N)\,\right) = \frac{1}{\sqrt{(2\pi\sigma^2)^N}} \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2\right), \tag{4.13}$$

since, conditioned on $\mathbf{w}$ and the $\mathbf{h}(n)$, the $y(n)$ are independent Gaussian with mean $\mathbf{h}(n)^T\mathbf{w}$ and variance $\sigma^2$. The above conditional density is often referred to as the *likelihood function*. Any estimator that maximizes it according to the criterion

$$\max_{\mathbf{w}} p\left(y(1),\ldots,y(N)\,|\mathbf{w},\mathbf{h}(1),\ldots,\mathbf{h}(N)\right), \tag{4.14}$$

is appropriately called a *maximum likelihood* (ML) estimator. Inspection of the formula in (4.13) clearly shows that solving (4.8) maximizes the exponent in (4.13) and so yields the ML estimator. In other words, the RLS algorithm recursively computes the ML solution.

Some remarks on the stochastic assumptions that lead to the above observation are in order. Most importantly, they differ from the stochastic assumptions we made when obtaining the Wiener solution in two ways. First, we require that the disturbance sequence be iid Gaussian (which we did not need for the Wiener solution). And second, we do not need any stochastic assumption on the input vectors $\mathbf{h}(n)$ (as we did in the Wiener case) since we treat them as known and condition on their values.

### MAP Estimators and Regularized Least-Squares

It is also possible to allow the weight vector to be random. This allows us to write the *a posteriori* conditional probability as follows

$$p\left(\mathbf{w}\,|(y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)\right) = \frac{p\left(y(1),\ldots,y(N)\,|\mathbf{w},\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)p(\mathbf{w}\,|\mathbf{h}(1),\ldots,\mathbf{h}(N))}{p\left(y(1),\ldots,y(N)\,|\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)} \tag{4.15}$$

where we have used Bayes' rule. Any estimator that maximizes the above a posteriori probability is referred to as a MAP (maximum a posteriori) estimator. Since the denominator is independent of $\mathbf{w}$, and since we assume that $\mathbf{w}$ is independent of the regressor vectors $\mathbf{h}(n)$—so that $p(\mathbf{w}\,|\mathbf{h}(1),\ldots,\mathbf{h}(N)) = p(\mathbf{w})$—it follows that MAP estimators satisfy the criterion

$$\max_{\mathbf{w}} p\left(y(1),\ldots,y(N)\,|\mathbf{w},\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)p(\mathbf{w}). \tag{4.16}$$

To obtain an explicit solution we need to assume a certain model for $\mathbf{w}$ and, again, the standard one is that it is a zero-mean Gaussian random vector with covariance matrix $E\mathbf{w}\mathbf{w}^T = \Pi_0$, independent of all the other signals involved. In this case, we have

$$p\left(y(1),\ldots,y(N)\,|\mathbf{w},\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)p(\mathbf{w}) = \frac{1}{K}\exp\left(-\frac{1}{2\sigma^2}\left[\mathbf{w}^T\sigma^2\Pi_0^{-1}\mathbf{w} + \sum_{n=1}^{N}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right]\right), \tag{4.17}$$

where $K = \sqrt{(2\pi)^{N+m}\sigma^{2N}\det\Pi_0}$. Therefore the MAP estimator is one that solves the following so-called *regularized* least-squares problem

$$\min_{\mathbf{w}}\left[\mathbf{w}^T\sigma^2\Pi_0^{-1}\mathbf{w} + \sum_{n=1}^{N}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right]. \tag{4.18}$$

This is identical to the least-squares problem (4.8), except for the regularization term (often called the prior) $\mathbf{w}^T\sigma^2\Pi_0^{-1}\mathbf{w}$. The solution is identical to that of RLS (4.10-4.11) except that now we should initialize the recursions with

$$\hat{\mathbf{w}}(0) = 0 \quad\text{and}\quad \mathbf{P}(0) = \sigma^{-2}\Pi_0.$$

In fact, this makes the regularized solution more convenient to use than the non-regularized one, for which we could only start the recurions from time $m$ and needed to explicitly compute$\hat{\mathbf{w}}(m)$ and $\mathbf{P}(m)$.

### Least-Mean-Squares Estimation

Let us return to the orginal least-mean-squares criterion (4.3), but now apply the above stochastic assumptions. Thus, using $\hat{\mathbf{w}}$ to denote our estimate of the weight vector and $\mathbf{w}$ to denote its true unknown value, the mean-square-error (4.3) becomes

$$E\left(y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}\right)^2 = E\left(v(n)+\mathbf{h}(n)^T(\mathbf{w}-\hat{\mathbf{w}})\right)^2 = \sigma^2 + \mathbf{h}(n)^T E(\mathbf{w}-\hat{\mathbf{w}})(\mathbf{w}-\hat{\mathbf{w}})^T\mathbf{h}(n),$$

where in the second step we used the fact that $v(n)$ is independent of $\mathbf{w}$ and where we have used our assumption that $\mathbf{h}(n)$ is deterministic to pull it outside the expectation. This implies that the criterion becomes that of minimizing $E\|\mathbf{h}(n)^T\mathbf{w} - \mathbf{h}(n)^T\hat{\mathbf{w}}\|^2$. It is often customary to define the "uncorrupted" output $\mathbf{h}(n)^T\mathbf{w}$ as the *desired* signal $d(n) = \mathbf{h}(n)^T\mathbf{w}$. In this case, the criterion is to choose $\mathbf{w}$ so as to best match the desired signal $d(n)$ in a least-mean-squares sense.

In any event, at time $n$, the optimal estimate of the desired signal is given by the conditional mean [8]

$$\hat{d}(n) = E\left[\mathbf{h}(n)^T\mathbf{w}\,|y(1),\ldots,y(n)\right] = \mathbf{h}(n)^T E\left[\mathbf{w}\,|y(1),\ldots,y(n)\right].$$

This implies that, irrespective of the input vector $\mathbf{h}(n)$, we may define the optimal estimate of the weight vector at time $n$ as

$$\hat{\mathbf{w}}(n) = E\left[\mathbf{w}\,|y(1),\ldots,y(n)\right]. \tag{4.19}$$

The fact that the optimal estimate of $\mathbf{w}$ here does not depend on $\mathbf{h}(n)$ is significant and follows from the linearity of the conditional mean. We remark that this is not necessarily true of other estimation criteria—more on this later.

When $v(n)$ is a stationary white Gaussian process, it is well known that the conditional mean is given by the MAP estimator [9]. Thus the solution to the regularized RLS problem (4.18) yields the least-mean-squares solution. When $v(n)$ is not Gaussian, however, the conditional mean does not coincide with the MAP estimator and the solution is not given by (4.18).

If we insist that our estimator be linear in the observations, i.e., that $\hat{d}(n)$ be a linear function of $\{y(1),\ldots,y(n)\}$, then it turns out that the optimal estimator depends only on the first and second-order statistics (the mean and covariance functions) of the signals involved. In this case the optimal estimator is known as the *linear least-mean-squares error* (LLMSE) estimator. If $v(n)$ is a zero-mean stationary white process, $Ev(n) = 0$ and $Ev(n)v(k) = \sigma^2\delta_{nk}$, where $\delta_{nk}$ represents the Kronecker delta function, and $\mathbf{w}$ is zero-mean, $\Pi_0$-variance and uncorrelated with the $v(n)$, then it turns out that the LLMSE is given by the solution to the regularized least-squares problem (4.18). This makes sense since in the Gaussian case the least-mean-squares solution is linear and given by (4.18), whereas the LMMSE estimator depends only on the first and second order statistics. Therefore the LLMSE estimator should coincide with the Gaussian MAP estimator with the same first and second order statistics, which is given by (4.18).

To summarize, the RLS algorithm yields the least-mean-squares estimator when the $v(n)$ and $\mathbf{w}$ are independent zero-mean Gaussian random variables. It yields the linear least-mean-squares estimator under the milder assumption that $v(n)$ and $\mathbf{w}$ are uncorrelated zero-mean random variables.

### 4.1.4 Some Questions

In the last few sections we have provided the RLS algorithm with a plethora of properties and optimality criteria. However, at this stage, for the LMS algorithm all we have provided is a heuristic argument for it being an approximation to the Wiener solution.

We have argued that, under some mild mixing conditions, the RLS solution converges to the optimal Wiener solution. The convergence of the RLS algorithm can be seen from the fact that, as $n \to \infty$, the matrix

$\mathbf{P}(i) = \left(\sum_{n=1}^{i} \mathbf{h}(n)\mathbf{h}(n)^T\right)^{-1}$ approaches zero[2] and therefore so does the *gain vector* $\frac{\mathbf{P}(i-1)\mathbf{h}(i)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}$ in (4.10), implying that $\hat{\mathbf{w}}(i-1) \to \hat{\mathbf{w}}(i) \to \mathbf{w}_o$. The LMS algorithm (4.6), on the other hand, has no chance of convergence since the gain vector $\mu\mathbf{h}(n)$ is non-zero for all time, meaning that, as long as there is a non-zero error signal $y(n) - \mathbf{h}(n)^T\mathbf{w}$ (which is always the case when we have a disturbance signal $v(n)$), the value of $\hat{\mathbf{w}}(i-1)$ can never approach $\hat{\mathbf{w}}(i)$.[3]

We have also shown that under a Gaussian disturbance model, and assuming that the input vectors are deterministic, that, depending on whether we consider a regularizing term or not, RLS recursively yields the ML and MAP estimates of the weight vector. It also yields the least-mean-squares solution under the Gaussian assumption and the linear least-mean-squares solution under the assumption of a zero-mean white disturbance signal. For LMS, on the other hand, we have no such stochastic interpretations.

With all that has been said it appears that RLS should be *the* choice for adaptive filtering. Nonetheless, a survey of the applications of adaptive filtering over the last few decades reveals that the LMS algorithm and its variants are more widely used than RLS and its variants. It is therefore natural to ask why this has been the case. Apart from the performance and optimality issues just discussed, there are other criteria that determine the applicability of a certain algorithm or methodology for different practical problems. These may be listed as follows.

1. *Simplicity*. Simpler solutions are often preferred in practice and the LMS algorithm is certainly simple. However, the RLS algorithm is not really all that complex. It has a structure quite similar to that of LMS—the weight vector is updated according to the error signal along the direction of a certain gain vector. The only difference is that computing the gain vector in RLS requires the propagation of a Riccati recursion.

2. *Computational complexity.* Algorithms that require less computations are preferred in practice. The LMS algorithm clearly requires $O(m)$ computations per iteration. The RLS algorithm, as depicted in (4.10-4.11), requires $O(m^2)$ computations per iteration, essentially because the Riccati recursion (4.11) requires a matrix-vector product, as well as a vector-vector outerproduct. However, in many applications the input vectors possess certain structure. The most common of these is the time series

[2]This requires that for any scalar $\Delta > 0$ there exist a time instant $i$ such that $\sum_{n=1}^{i} \mathbf{h}(n)\mathbf{h}(n)^T > \Delta I_m$, where the latter inequality is in the sense of positive definite matrices. This condition is referred to as *persistence of excitation* and is a very reasonable assumption.

[3]We should mention that the above argument holds when the LMS algorithm has a constant step size $\mu > 0$. There also exist variants of LMS with a time varying step size $\mu(i) > 0$. If $\mu(i) \to 0$, then the LMS algorithm will converge. However, we will not be considering vanishing step sizes in this chapter.

structure $\mathbf{h}(n)^T = \left[ \begin{array}{cccc} u(n) & u(n-1) & \ldots & u(n-m+1) \end{array} \right]$ of (4.2) where there is a great deal of redundancy between $\mathbf{h}(n)$ and $\mathbf{h}(n-1)$. There exist several clever techniques to exploit this redundancy and thereby reduce the computations to $O(m)$ per iteration [10]. These are generally referred to *fast* RLS algorithms.

3. *Numerical stability.* Imprecisions and round-off errors are unavoidable in the numerical implementation of any algorithm. Algorithms that suffer from numerical instability in the face of such errors are not suitable in practice. If the learning rate is not too large, so that the weight vector estimates do not diverge, then it can be shown that the LMS algorithm is numerically stable. In other words, round-off errors cannot lead to divergence and other problems. With RLS, if implemented according to (4.10-4.11), numerical instability can be an issue since round-off errors and finite-precision can lead to the matrix $P(i)$ of the Riccati recursion (4.11) losing its positive definiteness. (When this happens, RLS may not yield meaningful estimates.) The problem of losing positive definiteness of the Riccati variable has been known for a long time, especially in the context of Kalman filtering, and has been resolved by employing certain so-called *square-root* algorithms [11, 8]. Rather than propagate the Riccati variable $P(i)$, these algorithms propagate its square-root factor, i.e., an $m \times m$ matrix $P(i)^{1/2}$, such that $P(i)^{1/2}P(i)^{T/2} = P(i)$. With this trick, loss of positivity is no longer an issue. Moreover, square-root algorithms make extensive use of unitary transformations which, since they do not change the norm of vectors, are the most numerically stable matrix operations that can be performed.

   In their most general form, square-root algorithms for RLS require $O(m^2)$ operations. Under the time series structure $\mathbf{h}(n)^T = \left[ \begin{array}{cccc} u(n) & u(n-1) & \ldots & u(n-m+1) \end{array} \right]$, there exist fast square-root algorithms that require $O(m)$ computations. However, these employ $J$-unitary, rather than unitary, transformations and so their numerical stability is still an area of active research.

In view of the above comments it follows that modifications of the RLS algorithm exist that, in terms of simplicity, computational complexity, and numerical stability are comparable to the LMS algorithm. When we add to these the optimaility properties of RLS which LMS lacks, we are still left with the question of why LMS is more widely used in adaptive filtering applications. One possible explanation could be that RLS is less well known, but this certainly cannot be true since it is much older than LMS (going back to Gauss and Legendre). In fact, the celebrated Kalman filter (of which RLS is a special case) was discovered at about the same time as the discovery of LMS [12].

## 4.2 The Question of Robustness

The answer to why LMS is more often used in adaptive filtering can be found in the assumptions required for the various optimality properties of RLS. When the inputs and outputs are stationary random processes, the Wiener solution (4.5) is meaningful and the RLS algorithm converges to it. However, what if the signals involved are not stationary? Although the Wiener solution is no longer meaningful, under a persistence of excitation assumption, the RLS algorithm still converges—but what does it converge to? Moreover, what if the unknown weight vector $\mathbf{w}$, which we have so far assumed to be constant, itself varies with time? The RLS algorithm (4.10-4.11) has a vanishing gain vector and so cannot track a time-varying $\mathbf{w}$.[4] The LMS algorithm, with its non-vanishing gain vector, on the other hand, may be able to perform such tracking.

The stochastic optimality properties of being the ML and/or MAP estimators require that the additive noise term $v(n)$ be zero-mean, white and Gaussian. The optimality property of being the linear least-mean-squares estimator requires that $v(n)$ be zero-mean and white. But what if the $v(n)$ are not Gaussian? What if they are not white? In many adaptive filtering applications the additive noise term includes modeling errors: the *true* model may be an IIR filter, and so by assuming the FIR model (4.2) we are neglecting the tail of the filter and relegating it to $v(n)$. Or the model may have some nonlinearities which we have ignored and included in $v(n)$. In any event, with the inclusion of modeling errors, the $v(n)$ are no longer Gaussian, or white for that matter, and the stochastic optimality properties do not hold. So what happens to these algorithms and what happens to their performance?

Of course, no matter what the statistics and distributions of the noise term, the RLS algorithm always is optimal in the sense of minimizing the least-squares cost (4.8). The reason being that this is a deterministic cost. Now this very well may be a reasonable optimality property to have. However, insofar as deterministic least-squares costs go, it may also be reasonable to consider the *weighted* least-squares criterion:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} q_n \left(y(n) - \mathbf{h}(n)^T \mathbf{w}\right)^2 , \tag{4.20}$$

where $\{q_n > 0\}$ is a set of weights. More generally, we can consider the weighted least-squares problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{N} \left(y(n) - \mathbf{h}(n)^T \mathbf{w}\right) q_{nk} \left(y(k) - \mathbf{h}(k)^T \mathbf{w}\right) , \tag{4.21}$$

[4] There exist certain variations of the RLS algorithm, such as the exponentially-windowed RLS algorithm that have a non-vanishing gain vector and so may be suitable for tracking. More on these later.

where $\mathbf{Q} = [q_{nk}]$ is an $N \times N$ positive definite matrix. This latter cost, especially, can be obtained by first applying the error sequence $\{y(n) - \mathbf{h}(n)^T\mathbf{w}\}$ to a linear filter, and then applying the output of the filter to a standard least-squares problem of the form (4.8).

We thus conclude that there is nothing special about the least-squares criterion (4.8). Which of the three alternatives (4.8), (4.20) and (4.21) one should use depends on the application at hand.

### 4.2.1 Robustness and Adaptation

The above discussions should bother us on two counts. First, we are left with the question of what the performance of RLS will be when the stochastic assumptions are not met? In other words, how *robust* is RLS with respect to modeling errors and lack of statistical knowledge of the exogenous signals? Second, if indeed we assume certain stochastic assumptions (such as Gaussianity, whiteness, etc.) then the problem of adaptive filtering reduces to a statistical estimation problem. This really takes the wind out of the sails of adaptive filtering. Instead of being an independent field in itself, it is relegated to being a subset of statistical estimation theory. More importantly, the word *adaptive* becomes vacuous. What are we adapting to? Nothing really, since we have assumed perfect stochastic models for all the signals involved.

In reality adaptive algorithms should be able to operate under different stochastic assumptions, and to be able to tolerate different types of modeling errors. In other words, they should be able to *adapt* to the stochastic, or otherwise, environment that they are in. This clearly implies that adaptive algorithms must be *robust* to variations of the system model and underlying statistical assumptions.

Therefore adaptation is much more related to robustness with respect to statistical variation than it is to optimality with respect to a specific statistical model. This is a very important point that is not nearly as well recognized as it should be. Of course, due to their robustness properties, adaptive algorithms will be conservative and will not perform as well as the optimal algorithm for any particular statistical model. However, we do expect them to perform reasonably well over a wide range of statistical models and environments. Indeed very much like the way the LMS algorithm performs in different environments.

In order to begin to quantify what we mean by robustness, it is helpful to pose the basic robustness question for any estimation algorithm:

*Is it possible that small disturbances and modeling errors may lead to large estimation errors?*

Note that in the above question we have made no reference to the statistics of the disturbances. All we are asking for is that, if the disturbances and modeling errors are small, then the estimation errors be small *no matter what the statistics of the disturbances may be.* In other words, as long as we set up a model that reasonably describes our data, i.e., the disturbances and modeling errors are small, then a robust estimator will guarantee that the estimation errors will be small.

The above coments imply that any approach to robust estimation requires a notion of *largeness* and *smallness* for the signals involved. For this there exist many possibilities. For example, one can consider the peak of the absolute value of the signals as one such measure. In control-theoretic jargon this is referred to as $l^1$ theory [13]. A perhaps more physical measure, widely used in practice, and one that allows for more analytic tractability is the *energy* of the signal. This is what leads to $H^\infty$ theory.

### 4.2.2 The $H^\infty$ Approach

The first systematic study of robustness, within the framework described above, was done in the context of control theory and was introduced by Zames in 1980 [14]. Zames' $H^\infty$ theory was concerned, not with estimation problems but, with the design of controllers that were robust with respect to model uncertainty and lack of statistical knowledge on the exogenous signals. $H^\infty$ control theory can be regarded as the outgrowth and extension of classical linear-quadratic-Gaussian (LQG) control theory, developed in the 1950's and 1960's, which assumed perfect models and complete statistical knowledge [15]. Indeed the development of $H^\infty$ theory in the 1980's and 1990's is considered as one of the significant achievements in control theory.

Now in the $H^\infty$ context, a robust estimator is one for which disturbances with small energy lead to estimation errors with small energy. Therefore the natural object to study is the *energy gain* from the disturbances to the estimation errors. In particular, since we are interested in having small estimation error energies for *all* small disturbance energies, we need to focus on the *worst-case* energy gain. This is what is referred to as the $H^\infty$ norm.[5]

**Definition 1 ($H^\infty$ norm).** *Suppose an estimator maps the disturbance sequence $\{v(n)\}$ to the estimation*

[5] The norm defined here is really what is known as a 2-induced norm. (Since it is the maximum of the ratio of energies or 2-norms.) For historical reasons, in the control theory literature the 2-induced norm is referred to as the $H^\infty$ norm. Here $H$ stands for Hardy space, the space of all causal and stable functions of a complex variable, i.e., the space of all functions analytic outside the unit circle. The superscript $\infty$ refers to the fact that $H^\infty$ is the space of functions analytic outside the unit circle and with finite magnitude on the unit circle [16]. In any event, the terminology $H^\infty$ norm is a misnomer since, strictly speaking, the 2-induced norm becomes an $\infty$ norm (in the frequency domain) only when we consider infinite-horizon linear time-invariant (LTI) systems, which is the context in which $H^\infty$ control was originally introduced. (As is well known the maximum energy gain of any stable LTI system with transfer function $K(z)$ is given by $\max_\omega \left|K(e^{j\omega})\right|^2$). Nontheless, we too shall be guilty of using this loose terminology.

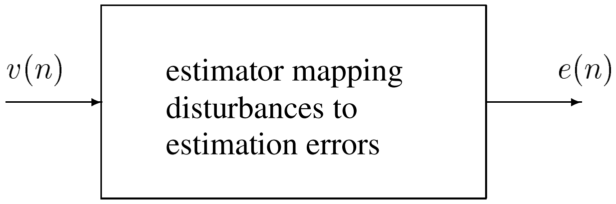


Figure 4.3: The $H^\infty$ norm is the maximum energy gain from the disturbances to the estimation errors.

*error sequence* $\{e(n)\}$. *Then the squared* $H^\infty$ *norm of the estimator is defined as*

$$\gamma^2 = \sup_{\{v(n)\}\in l^2} \frac{\|e\|^2}{\|v\|^2}, \tag{4.22}$$

*where* $\|v\|^2 = \sum_n v(n)^2$, $\|e\|^2 = \sum_n e(n)^2$, $l^2$ *denotes the space of square-summable sequences, and* sup *refers to supremum.*

In $H^\infty$ estimation we seek the estimator that minimizes the $H^\infty$ norm.

**Problem 1 (Optimal $H^\infty$ estimation problem).** *Find an estimator that minimizes the* $H^\infty$ *(or* 2*-induced) norm from the disturbances* $\{v(n)\}$ *to the estimation errors, i.e.,*

$$\inf_{\textit{all estimators}} \sup_{\{v(n)\}\in l^2} \frac{\|e\|^2}{\|v\|^2}, \tag{4.23}$$

*where* inf *refers to infimum. Moreover, find the resulting optimal value* $\gamma_{opt} = \inf \gamma$.

The *minimax* nature of $H^\infty$-optimal estimation is evident from the above problem formulation. $H^\infty$ estimation is essentially a game problem: nature (the opponent) has access to the unknown disturbance $\{v(n)\}$ and chooses it to maximize the energy gain in (4.23), whereas we have choice of the estimator to minimize it.

$H^\infty$-optimal estimators safeguard against the *worst-case* disturbance that maximizes the energy gain to the estimation errors. Since this worst-case disturbance is a deterministic sequence, such estimators do not require any statistical assumptions. Moreover, since the infimization in (4.23) is taken over all possible disturbances, the resulting estimator will be robust with respect to disturbance variation. It can, on the other hand, be quite conservative.

We should mention that $H^\infty$ theory is a very rich area and that there exist many different approaches to solving it. The original methods were operator- and function-theoretic and made use of interpolation theory

[17, 18], but there also exist state-space [19], circuit-theoretic [20] and game-theoretic [21] approaches. We shall not go into any of these here, though the interested reader may consult any of the aforementioned books for details. We only mention in passing that in [22] an approach is developed based on indefinite quadratic forms and indefinite matric spaces, where the connections to the classical theories of stochastic and determinsitic least-squares are made explicit. Some of our discussions below are motivated by this approach.

### 4.2.3 A First Attack

Let us begin to consider how we can apply the $H^\infty$ approach to the adaptive filtering problem at hand. Suppose we are given the input-output pairs $\{(\mathbf{h}(n), y(n))\}_{n=1}^N$ and we would like to estimate the unknown weight vector $\mathbf{w}$. How should we set up the problem?

First let us look at the disturbances. Clearly there are two unknowns. The weight vector itself and the unknown additive noise

$$v(n) = y(n) - \mathbf{h}(n)^T \mathbf{w}.$$

We can therefore define the energy of the disturbances as[6]

$$\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} v(n)^2 = \mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2, \tag{4.24}$$

where $\mu > 0$ is a constant that adjusts the relative importance of our uncertainty about the weight vector and our uncertainty about the additive noise. As we shall see below, it is no coincidence that we have used the same notation for this constant as for the learning rate in LMS. Note that, since the outputs $y(n)$ are observed, the only uncertainty is due to the unknown weight vector $\mathbf{w}$.

Now what about the estimation error? One plausible choice is the following: Suppose after observing the $N$ data points $\{(\mathbf{h}(n), y(n))\}_{n=1}^N$ we are presented with a *new* input vector $\mathbf{h}(N+1)$ and are interested in our ability to *predict* the resulting desired output $d(N+1) = \mathbf{h}(N+1)^T\mathbf{w}$. Therefore a reasonable estimation error is $\hat{d}(N+1) - \mathbf{h}(N+1)^T\mathbf{w}$, where $\hat{d}(N+1)$ denotes our estimate.[7]

[6] If we have an initial estimate of the weight vector $\hat{\mathbf{w}}(0)$, say, then it is more appropriate to define the disturbance energy as

$$\mu^{-1}\left(\mathbf{w} - \hat{\mathbf{w}}(0)\right)^T\left(\mathbf{w} - \hat{\mathbf{w}}(0)\right) + \sum_{n=1}^{N} v(n)^2.$$

However, without loss of generality we shall assume that $\hat{\mathbf{w}}(0) = 0$.

[7] We remark that in the prediction problem under consideration $\hat{d}(N+1)$ is allowed to be a function of $\mathbf{h}(N+1)$, but not of

Thus, we are confronted with the following problem:

$$\min_{\hat{d}(N+1)} \max_{\mathbf{w}} \frac{\left(\hat{d}(N+1) - \mathbf{h}(N+1)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}. \tag{4.25}$$

To facilitate solving this problem, let us first look at the suboptimal problem of guaranteeing that the maximum energy gain is bounded by the value $\gamma^2$. In other words, for all possible $\mathbf{w}$, we would like to have

$$\frac{\left(\hat{d}(N+1) - \mathbf{h}(N+1)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2} < \gamma^2,$$

or, equivalently,

$$\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2 - \gamma^{-2}\left(\hat{d}(N+1) - \mathbf{h}(N+1)^T\mathbf{w}\right)^2 > 0. \tag{4.26}$$

Note that, due to the minus sign on the last term, this is an indefinite quadratic form in the unknown $\mathbf{w}$. Defining

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}(1) & \dots & \mathbf{h}(N) \end{bmatrix}, \quad \mathbf{h} = \mathbf{h}(N+1), \quad \hat{d} = \hat{d}(N+1)$$

allows us to rewrite (4.26) as

$$\begin{bmatrix} \mathbf{w}^T & \mathbf{y}^T & \hat{d} \end{bmatrix} \begin{bmatrix} \mu^{-1}I + \mathbf{H}\mathbf{H}^T - \gamma^{-2}\mathbf{h}\mathbf{h}^T & -\mathbf{H} & \gamma^{-2}\mathbf{h} \\ -\mathbf{H}^T & I & 0 \\ \gamma^{-2}\mathbf{h}^T & 0 & -\gamma^{-2} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{y} \\ \hat{d} \end{bmatrix} > 0. \tag{4.27}$$

Now the above indefinite quadratic form is positive for all $\mathbf{w}$ if, and only if,

1. It has a minimum in the unknown variable $\mathbf{w}$ (otherwise its value could approach $-\infty$).

2. $\hat{d}$ can be chosen such that the value at the minimum is positive.

The condition for having a minimum can be readily found from computing the Jacobian (or second deriva-

$y(N+1)$.

tive) with respect to $\mathbf{w}$ and insisting that it be positive definite, i.e.,

$$\mu^{-1}I + \mathbf{H}\mathbf{H}^T - \gamma^{-2}\mathbf{h}\mathbf{h}^T > 0.$$

Some simple algebra shows that this is equivalent to

$$\gamma^2 > \mathbf{h}^T \left[\mu I + \mathbf{H}\mathbf{H}^T\right]^{-1} \mathbf{h}. \tag{4.28}$$

From this we conclude that the optimal value of $\gamma$ is

$$\gamma_{opt}^2 = \mathbf{h}(N+1)^T \left[\mu I + \sum_{n=1}^{N} \mathbf{h}(n)\mathbf{h}(n)^T\right]^{-1} \mathbf{h}(N+1). \tag{4.29}$$

Some further algebra shows that, once the minimization over $\mathbf{w}$ has been done, the value of the quadratic form at its minimum is

$$\begin{bmatrix} \mathbf{y}^T & \hat{d} \end{bmatrix} \left( \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 \end{bmatrix} + \frac{1}{\mu} \begin{bmatrix} \mathbf{H}^T \\ \mathbf{h}^T \end{bmatrix} \begin{bmatrix} \mathbf{H} & \mathbf{h} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{y} \\ \hat{d} \end{bmatrix}.$$

A further completion of squares shows this to be

$$\mathbf{y}^T \left(I + \frac{1}{\mu}\mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{y} + \frac{\left(\hat{d} - \frac{1}{\mu}\mathbf{h}^T\mathbf{H}\left(\frac{1}{\mu}I + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{y}\right)^2}{-\gamma^2 + \mathbf{h}^T\left[\mu I + \mathbf{H}\mathbf{H}^T\right]^{-1}\mathbf{h}}. \tag{4.30}$$

Note that, due to (4.28), the second term in the above is always non-positive. Therefore it is clear that one choice that guarantees the value at the minimum to be positive is:

$$\hat{d} = \frac{1}{\mu}\mathbf{h}^T\mathbf{H}\left(\frac{1}{\mu}I + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{y} = \mathbf{h}^T\left(\frac{1}{\mu}I + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}\mathbf{y}.$$

But this is nothing but the estimate obtained from the solution to the regularized least-squares problem (4.18) with $\Pi_0 = \mu I$:

$$\hat{d}(N+1) = \mathbf{h}(N+1)^T \underbrace{\left(\mu I + \sum_{n=1}^{N}\mathbf{h}(n)\mathbf{h}(n)^T\right)^{-1} \sum_{n=1}^{N}\mathbf{h}(n)y(n)}_{\hat{\mathbf{w}}_{ls}} \; ! \tag{4.31}$$

In other words, after all the trouble of defining robustness with respect to disturbance variation and introducing the $H^\infty$ estimation problem, it turns out that the optimal solution is still given by the regularized least-squares solution (4.18). A strange predicament indeed. So was all this worth while? Let us probe further.

### 4.2.4 A Prediction Problem

What we have shown above is that if we are given the data $\{(\mathbf{h}(n), y(n))\}_{n=1}^N$ then, for any new input vector $\mathbf{h}(N+1)$, the best predictor in the $H^\infty$ sense is given by the regularized least-squares estimate. Of course the regularized solution we have presented is off-line—we need all the data before we can predict the output for $\mathbf{h}(N+1)$. As mentioned earlier, in many applications, real-time constraints are crucial and at any time $n$ we will need to predict the output at time $n+1$ using our past observations. In other words, we need an optimal solution that is recursive.

For the least-squares problem we saw that a recursive solution was readily available via the RLS algorithm. In other words, at each time $i$ the solution to the (regularized) RLS algorithm solves the following problem:

$$\min_{\mathbf{w}} \sum_{n=1}^{i} \left(y(n) - \mathbf{h}(n)^T \mathbf{w}\right)^2 .$$

Since the off-line $H^\infty$-optimal solution is given by the least-squares solution, it appears plausible that the on-line recursive least-squares solution should recursively give the $H^\infty$-optimal solution. But is this true?

What is certainly true is that, for each time $i$, the RLS solution solves the following minimax problem

$$\min_{\hat{d}(i+1)} \max_{\mathbf{w}} \frac{\left(\hat{d}(i+1) - \mathbf{h}(i+1)^T \mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{i} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2},$$

and also, since future values of the disturbance have no effect on the prediction error at time $i$:

$$\min_{\hat{d}(i+1)} \max_{\mathbf{w}} \frac{\left(\hat{d}(i+1) - \mathbf{h}(i+1)^T \mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}.$$

From these arguments, we conclude that the RLS algorithm recursively solves the following minimax estimation problem:

$$\min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \left[ \max_{\mathbf{w}} \frac{\left(\hat{d}(1) - \mathbf{h}(1)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2} + \ldots \max_{\mathbf{w}} \frac{\left(\hat{d}(N+1) - \mathbf{h}(N+1)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}. \right] \tag{4.32}$$

But this is not quite the problem we would like to solve. What we would like to solve is the following:

$$\min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \max_{\mathbf{w}} \frac{\sum_{n=1}^{N+1} \left(\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}, \tag{4.33}$$

i.e., we would like to minimize the maximum energy gain from the disturbances to the prediction errors $\{\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}\}$. The problem with (4.32) is that at each time $i$ we have a different worst-case disturbance sequence (since we have a different maximization at each time). This does not make sense since we are allowed to only have one worst-case disturbance sequence. Moreover, due to the obvious inequality

$$\max\left[a_1 + a_2 \ldots a_N\right] \leq \max a_1 + \max a_2 + \ldots + \max a_N,$$

it follows that the optimal value of (4.33) is lower than that of (4.32). In other words, we may be able to do better than the RLS algorithm. In fact, it turns out that this is the case.

To solve the $H^\infty$ problem (4.33) we may recourse to general $H^\infty$ solutions. However, this will require us to develop quite a bit of theory. Fortunately, to solve (4.33) we do not need to bring to bear the full force of $H^\infty$ techniques. Using some hindsight one can give a rather simple first principles derivation of the solution—which is what we will set about to do.

### The Value of $\gamma_{opt}$

We first argue that

$$\gamma_{opt}^2 = \min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \max_{\mathbf{w}} \frac{\sum_{n=1}^{N+1} \left(\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}$$

cannot be less than one. To this end, suppose, without loss of generality, that the initial guess of the estimator is $\hat{\mathbf{w}}(0) = 0$. Then one may conceive of a disturbance sequence $v(n)$ that yields an output signal that is zero for all times and therefore coincides with the output expected from $\hat{\mathbf{w}}(0) = 0$. Thus,

$$v(n) = -\mathbf{h}(n)^T\mathbf{w} \quad \text{and} \quad y(n) = \mathbf{h}(n)^T\mathbf{w} + v(n) = 0.$$

In this case, any permissible estimator will not change its estimate of $\mathbf{w}$ so that $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(0) = 0$ for all $n$.[8] Moreover, the prediction error becomes

$$\hat{d}(n+1) - \mathbf{h}(n+1)^T\mathbf{w} = 0 - \mathbf{h}(n+1)^T\mathbf{w} = v(n),$$

and so the energy gain

$$\frac{\|v\|^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \|v\|^2} = \frac{\sum_{n=1}^{N+1}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2}.$$

Let us now assume that the $\{\mathbf{h}(n)\}$ have the property that

$$\lim_{N\to\infty}\sum_{n=1}^{N}\mathbf{h}(n)^T\mathbf{h}(n) = \infty. \tag{4.34}$$

Note that this is a weaker condition than the earlier-mentioned *persistently exciting* condition which required the matrix $\lim_{N\to\infty}\sum_{n=1}^{N}\mathbf{h}(n)\mathbf{h}(n)^T$ go to infinity in *all* directions. Therefore we shall condition (4.34) just *exciting*.

In any event, when the $\{\mathbf{h}(n)\}$ are exciting, for any $\epsilon > 0$, it is always possible to find a weight vector $\mathbf{w}$ and an integer $N$ such that $\sum_{n=1}^{N}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2 \geq \frac{\mathbf{w}^T\mathbf{w}}{\epsilon\mu}$. With these choices we have

$$\frac{\sum_{n=1}^{N+1}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2} \geq \frac{\sum_{n=1}^{N}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\left(\mathbf{h}(n)^T\mathbf{w}\right)^2} \geq 1 - \epsilon,$$

which can be made arbitrarily close to one. We thus conclude that $\gamma_{opt} \geq 1$.

This implies that for all estimators, in the worst-case, the prediction error energy can be no less than the disturbance energy. In other words, in the worst-case, it is not possible to obtain disturbance attenuation.

The question now is whether a worst-case energy gain of unity is achievable. In other words, is $\gamma_{opt} = 1$? And if so, what is the optimal estimator?

[8] Note that if an estimator changes its estimate of $\mathbf{w}$ when confronted with an all zero output ($y(n) = 0$, for all $n$) then in the disturbance free case ($\mathbf{w} = 0$ and $v(n) = 0$, for all $n$) the denominator of (4.33) will be zero, but the numerator non-zero, which makes the enregy gain infinite. This is clearly not permissible.

### 4.2.5 The LMS Algorithm

Although this volume is devoted to the LMS algorithm, we have spent most of our time studying the RLS algorithm and considering robustness issues. It is now time to return to the LMS algorithm (4.6). If we define the estimation error of the weight vector as $\tilde{\mathbf{w}}(n) = \mathbf{w} - \hat{\mathbf{w}}(n)$, then it is straightforward to see that

$$\mu^{-1/2}\tilde{\mathbf{w}}(n) = \mu^{-1/2}\left[\tilde{\mathbf{w}}(n-1) - \mu\mathbf{h}(n)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right)\right]. \tag{4.35}$$

(The reason for pre-multiplying by $\mu^{-1/2}$ will become clear in a moment.) Moreover, we have

$$v(n) = \left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right) - \mathbf{h}(n)^T\tilde{\mathbf{w}}(n-1). \tag{4.36}$$

Squaring both sides of (4.35) and (4.36) and subtracting the results yields

$$\mu^{-1}\left|\tilde{\mathbf{w}}(n)\right|^2 - v(n)^2 = \mu^{-1}\left|\tilde{\mathbf{w}}(n-1)\right|^2 - \left(\mathbf{h}(n)^T\tilde{\mathbf{w}}(n-1)\right)^2 - \left(1 - \mu\left|\mathbf{h}(n)\right|^2\right)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right)^2, \tag{4.37}$$

where for any row vector $\mathbf{a}$ we have defined $|\mathbf{a}|^2 = \mathbf{a}^T\mathbf{a}$.

If we now add up all the equations (4.37) from time $n = 1$ to time $n = N + 1$, all but the first and last terms of $\mu^{-1}\left|\tilde{\mathbf{w}}(n)\right|^2$ cancel out and we are left with

$$\begin{aligned}\mu^{-1}\left|\tilde{\mathbf{w}}(N+1)\right|^2 - \sum_{n=1}^{N+1} v(n)^2 \quad &= \quad \mu^{-1}\left|\tilde{\mathbf{w}}(0)\right|^2 - \sum_{n=1}^{N+1}\left(\mathbf{h}(n)^T\tilde{\mathbf{w}}(n-1)\right)^2 - \\ &\quad \sum_{n=1}^{N+1}\left(1 - \mu\left|\mathbf{h}(n)\right|^2\right)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right)^2.\end{aligned}$$

The latter equation can be rewritten as

$$\frac{\mu^{-1}\left|\tilde{\mathbf{w}}(N+1)\right|^2 + \sum_{n=1}^{N+1}\left(\mathbf{h}(n)^T\tilde{\mathbf{w}}(n-1)\right)^2 + \sum_{n=1}^{N+1}\left(1 - \mu\left|\mathbf{h}(n)\right|^2\right)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right)^2}{\mu^{-1}\left|\tilde{\mathbf{w}}(0)\right|^2 + \sum_{n=1}^{N+1} v(n)^2} = 1. \tag{4.38}$$

Note that the second term in the numerator is just the energy of the prediction error $\mathbf{h}(n)^T\tilde{\mathbf{w}}(n-1) = \mathbf{h}(n)^T\mathbf{w} - \hat{d}(n)$. Moreover, if we assume that

$$\mu \leq \frac{1}{\mathbf{h}(n)^T\mathbf{h}(n)}, \quad \forall n \tag{4.39}$$

then the third term in the numerator of (4.38) is non-negative and so we have

$$\frac{\sum_{n=1}^{N+1}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{N+1}v(n)^2}\leq 1, \tag{4.40}$$

where we have used the fact that $|\tilde{\mathbf{w}}(0)|^2=\mathbf{w}^T\mathbf{w}$.

The result of (4.40) is significant since it shows that, if the learning rate satisfies the bound (4.39), then LMS guarantees an energy gain no greater than one. Since we previously argued that $\gamma_{opt}\geq 1$ this implies that LMS is $H^\infty$-optimal!

### The Condition on the Learning Rate

We have shown that if the learning rate satisfies (4.39) then for prediction errors LMS is $H^\infty$-optimal and achieves $\gamma_{opt}=1$. But what if (4.39) is not satisfied? Is it still true that $\gamma_{opt}=1$?

To answer this question, suppose that $\gamma_{opt}=1$ so that for any time $i$ there exists an estimator such that for all disturbances

$$\frac{\sum_{n=1}^{i+1}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{i}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}\leq 1.$$

This is equivalent to the existence of an estimator such that

$$\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{i}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2-\sum_{n=1}^{i+1}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\geq 0, \tag{4.41}$$

for all possible values of $\mathbf{w}$. This is an indefinite quadratic form in $\mathbf{w}$ and as mentioned earlier it can only be positive for all $\mathbf{w}$ if it has a minimum over $\mathbf{w}$. To check whether we have a minimum we twice differentiate to compute the Jacobian matrix and check for its positivity, which yields

$$\mu^{-1}I+\sum_{n=1}^{i}\mathbf{h}(n)\mathbf{h}(n)^T-\sum_{n=1}^{i+1}\mathbf{h}(n)\mathbf{h}(n)^T=\mu^{-1}I-\mathbf{h}(i+1)\mathbf{h}(i+1)^T\geq 0.$$

Since the above Jacobian matrix has only one non-unity eigenvalue, this latter condition is simply

$$\mu^{-1}-\mathbf{h}(i+1)^T\mathbf{h}(i+1)\geq 0.$$

But since $i$ was an arbitrary time instant, this is precisely the condition (4.39).

We have thus shown that if $\gamma_{opt} = 1$ then (4.39) must hold. Therefore it follows that if (4.39) does not, then $\gamma_{opt} \neq 1$, which implies $\gamma_{opt} > 1$.

### 4.2.6 $H^\infty$-Optimality of the LMS Algorithm

We can now summarize the above results.

**Theorem 1.** *Consider the input-vector output-pair $\{\mathbf{h}(n), y(n)\}$, related via the equation*

$$y(n) = \mathbf{h}(n)^T \mathbf{w} + v(n), \quad n \geq 0$$

*where $\mathbf{w}$ is an unknown weight vector, $v(n)$ is an unknown disturbance sequence and the input vectors $\mathbf{h}(n)$ are exciting. Consider the maximum energy gain from the disturbances $\mathbf{w}$ and $v(n)$ to the prediction error $\tilde{d}(n) = \hat{d}(n) - \mathbf{h}(n)^T \mathbf{w}$, i.e.,*

$$\gamma^2 = \max_{\mathbf{w}, v(\cdot) \in l^2} \frac{\left\|\tilde{d}\right\|^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \|v\|^2}.$$

1. *If $\mu \leq \inf_n \frac{1}{\mathbf{h}(n)^T\mathbf{h}(n)}$, then the minimum value of $\gamma^2$ is given by*

$$\gamma_{opt}^2 = 1,$$

*and an $H^\infty$-optimal predictor is given by $\hat{d}(n) = \mathbf{h}(n)^T \hat{\mathbf{w}}(n-1)$, where $\hat{\mathbf{w}}(n)$ is found from the LMS algorithm (4.6).*

2. *If $\mu > \inf_n \frac{1}{\mathbf{h}(n)^T\mathbf{h}(n)}$, then*

$$\gamma_{opt}^2 = 1 + \sup_n \lambda_{max} \left[ \left( \mu I + \sum_{i=0}^{n} \mathbf{h}(n)\mathbf{h}(n)^T \right)^{-1} \left( \mathbf{h}(n+1)\mathbf{h}(n+1)^T - \mu I \right) \right],$$

*where $\lambda_{max}(\mathbf{A})$ is the largest eigenvalue of $\mathbf{A}$. An $H^\infty$-optimal predictor is given by $\hat{d}(n) = \mathbf{h}(n)^T \hat{\mathbf{w}}(n-1)$, where now $\hat{\mathbf{w}}(n)$ is found from the recursion*

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{P}(n+1)\mathbf{h}(n) \left( y(n) - \mathbf{h}(n)^T \hat{\mathbf{w}}(n-1) \right), \quad \hat{\mathbf{w}}(-1) \tag{4.42}$$

*and where* $\mathbf{P}(n)$ *satisfies the Riccati recursion*

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \frac{\mathbf{P}(n)\mathbf{h}(n)\mathbf{h}(n)^T\mathbf{P}(n)}{\frac{\gamma_{opt}^2}{\gamma_{opt}^2-1} + \mathbf{h}(n)^T\mathbf{P}(n)\mathbf{h}(n)}, \quad P(0) = \mu I. \tag{4.43}$$

We should remark that the proof for part 1 of the above theorem has already been given. Proving part 2 requires knowledge of $H^\infty$ theory and so we omit it and refer the interested reader to [22]. Nonetheless, we have included the statement of part 2 for completeness and only mention that (4.42) is identical to the LMS algorithm (4.6) except that the learning rate $\mu I$ has been replaced by the Riccati variable $\mathbf{P}(n)$.

Theorem 1 solves a long standing issue of finding a rigorous basis for the LMS algorithm. Moreover, it confirms the robustness of the algorithm and gives theoretical justification for its widespread use in adaptive filtering. More to the point, the LMS algorithm is widely used not because it is an approximate least-squares solution (the exact solution, RLS, is readily available), nor because it is simple, computationally efficient or numerically stable (RLS can be made competitive to LMS on all these counts), but rather because it is an algorithm that is *robust with respect to disturbance variation*, a property that RLS, for example, cannot boast of. In fact, for prediction errors, and in the $H^\infty$ setting, it is the optimal (hence most robust) algorithm that there is.

Since it is a robust algorithm, LMS exhibits reasonable to good performance over a wide range of environments and operating conditions. However, it cannot hope to compete with algorithms that know the exact statistics of the signals involved and are optimized for them. The point is that LMS will invariably perform within reason no matter what the statistics and modeling errors are.

Finally, we mention that the LMS algorithm can be viewed as providing a *contractive* mapping from the disturbances to the prediction errors. (This is true since the prediction error energy is always less than the disturbance energy.) This property turns out to have significant implications for studying the stability of a wide class of adaptive algorithms. The idea is to represent any adaptive algorithm as the feedback connection of the LMS algorithm and a secondary system and to apply the *small-gain-theorem* from control theory [23, 24]. Since the LMS algorithm is a contraction, the loop gain is equal to the gain of the secondary system, and so stability is guaranteed if this gain is less than unity. This approach to stability analysis is expounded upon in [25].

### 4.2.7 Non-Uniqueness of the $H^\infty$-Optimal Predictor

In the statement of Theorem 1 we were careful to mention that LMS is *an* $H^\infty$-optimal estimator, since we have not yet determined whether the $H^\infty$ problem has a unique solution or not. Let us now explore this issue.

We are interested in determing all predictors that yield

$$\frac{\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{i}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}\leq 1,$$

for all $\mathbf{w}$ and all times $i$. This is equivalent to the inequality

$$\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{i}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\\ y(n)-\mathbf{h}(n)^T\mathbf{w}\end{bmatrix}^T\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\\ y(n)-\mathbf{h}(n)^T\mathbf{w}\end{bmatrix}\geq 0.$$

As mentioned several times earlier, the above indefinite quadratic form is non-negative for all $\mathbf{w}$ if, and only if, it has a minimum over $\mathbf{w}$ and the $\hat{d}(n)$ can be chosen such that the value at the minimum is non-negative. Due to our condition on the learning rate (4.39), we always have a minimum over $\mathbf{w}$. Minimizing over $\mathbf{w}$, the value at the minimum is

$$\sum_{n=1}^{i}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\\ y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\end{bmatrix}^T\begin{bmatrix}-1+\mu|\mathbf{h}(n)|^2 & \mu|\mathbf{h}(n)|^2\\ \mu|\mathbf{h}(n)|^2 & 1+\mu|\mathbf{h}(n)|^2\end{bmatrix}^{-1}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\\ y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\end{bmatrix}, \tag{4.44}$$

where $\hat{\mathbf{w}}(n)$ satisfies the recursion

$$\hat{\mathbf{w}}(n)=\hat{\mathbf{w}}(n-1)+\mu\mathbf{h}(n)\left(y(n)-\hat{d}(n)\right),\quad \hat{\mathbf{w}}(0)=0. \tag{4.45}$$

(For a proof of this result and a more general discussion of the minimization of such indefinite quadratic forms see [22], Theorem 3.4.2 and Lemmas 3.4.1 to 3.4.3.)

Performing an LDU (lower-diagonal-upper) factorization of the center matrix in (4.44) allows us to rewrite it as

$$-\sum_{n=1}^{i}\frac{(\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1))^2}{1-\mu|\mathbf{h}(n)|^2}+\sum_{n=1}^{i}\left((y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1))-\mu|\mathbf{h}(n)|^2(y(n)-\hat{d}(n))\right)^2\geq 0. \tag{4.46}$$

Note that, due to the learning rate constraint (4.39), the first summation in the above expression is non-positive. Clearly, one choice that renders (4.46) non-negative is $\hat{d}(n) = \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)$. Plugging this back into (4.45) readily gives the LMS algorithm. However, is this the only choice that renders (4.46) non-negative?

Obviously not. As long as the sequences

$$\left\{\frac{\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)}{\sqrt{1-\mu|\mathbf{h}(n)|^2}}\right\}_{n=1}^{i} \quad \text{and} \quad \left\{\left(y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\right)-\mu|\mathbf{h}(n)|^2\left(y(n)-\hat{d}(n)\right)\right\}_{n=1}^{i}$$

are related by a *strictly causal contractive mapping*, then (4.46) holds.[9]

We thus have shown the following result.

**Theorem 2.** *Consider the setting of Theorem 1 and assume that* $\mu \leq \inf_n \frac{1}{|\mathbf{h}(n)|^2}$. *Then all* $H^\infty$*-optimal predictors (that achieve* $\gamma_{opt}=1$*) are given by*

$$\begin{aligned}\hat{d}(n) \;=\; & \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)+\sqrt{1-\mu|\mathbf{h}(n)|^2}\,\times \\ & f_n\Big(\left(y(n-1)-\mathbf{h}(n-1)^T\hat{\mathbf{w}}(n-2)\right)-\mu|\mathbf{h}(n-1)|^2\left(y(n-1)-\hat{d}(n-1)\right), \\ & \left(y(n-2)-\mathbf{h}(n-2)^T\hat{\mathbf{w}}(n-3)\right)-\mu|\mathbf{h}(n-2)|^2\left(y(n-2)-\hat{d}(n-2)\right),\ldots\Big)\end{aligned} \tag{4.47}$$

*where* $f_n(\cdot,\cdot,\ldots)$ *is a strictly causal contractive mapping, and where* $\hat{\mathbf{w}}(n)$ *satisfies the recursion (4.45).*

An illustration of the parametrization of Theorem 2 is given in Fig. 4.4.

The simplest strictly causal contraction is $f_n(\cdot,\cdot,\ldots)=0$ for all $n$, which gives the LMS algorithm. Another simple causal contraction is the identity map $f_n(a(n-1),a(n-2),\ldots)=a(n-1)$. This gives

$$\hat{d}(n)=\sqrt{1-\mu|\mathbf{h}(n)|^2}\left(y(n-1)-\mathbf{h}(n-1)^T\hat{\mathbf{w}}(n-2)\right)-\mu|\mathbf{h}(n-1)|^2\left(y(n-1)-\hat{d}(n-1)\right),$$

which, upon plugging into (4.45) and performing simplifications yields

$$\begin{cases}\hat{d}(n) = \mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)+\sqrt{1-\mu|\mathbf{h}(n)|^2}\left(y(n-1)-\mathbf{h}(n-1)^T\hat{\mathbf{w}}(n-1)\right) \\ \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1)+\mu\mathbf{h}(n)\left[y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)-\sqrt{1-\mu|\mathbf{h}(n)|^2}\left(y(n-1)-\mathbf{h}(n-1)^T\hat{\mathbf{w}}(n-1)\right)\right] \\ \hat{\mathbf{w}}(0) = 0.\end{cases} \tag{4.48}$$

[9] Two sequences $\{a(n)\}$ and $\{b(n)\}$ are said to be related via a strictly causal contraction if, and only if, $a(i)=f_i(b(i-1),b(i-2),\ldots)$ and $\sum_{n=1}^{i}(a(n))^2 \leq \sum_{n=1}^{i}(b(n))^2$ for all $i$.

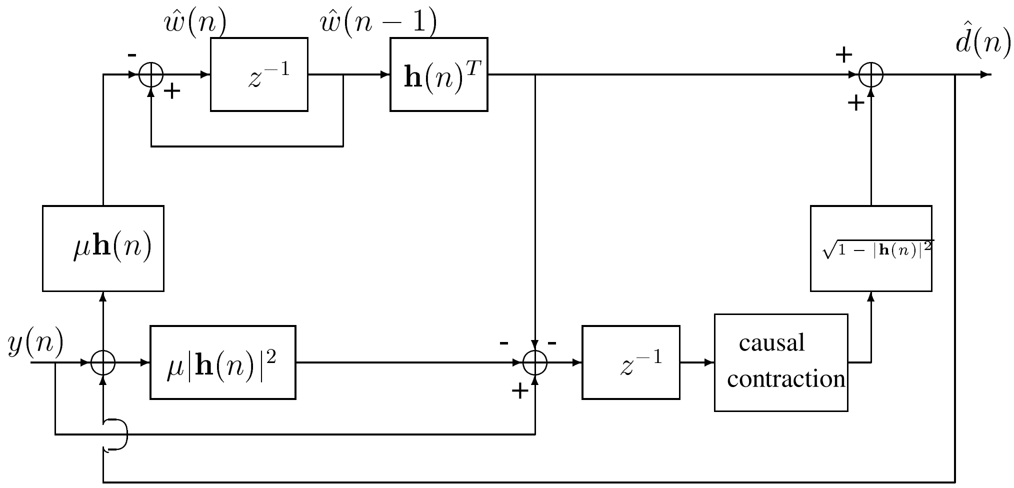


Figure 4.4: All $H^\infty$-optimal adaptive filters.

The above adaptive algorithm (4.48), as well as the LMS algorithm (4.6), are two different $H^\infty$-optimal adaptive filters. Of course, as suggested by Theorem 2 there are an infinite number of such algorithms, since there an infinite number of possibilities for the strictly causal contraction $f_n(\cdot,\cdot,\ldots)$.

In retrospect, it is perhaps not surprising that $H^\infty$-optimal filters are non-unique. The reason is that the $H^\infty$ criterion only guarantees a certain performance for the *worst-case* disturbance. It does not really specify what the performance should, or will, be with respect to other disturbances.

A natural question then is whether some of these $H^\infty$-optimal filters are preferable over other ones. In other words, although all the resulting filters are optimal from a robustness point of view, is it possible that they may have drastic differences with respect to other reasonable criteria, such as average, rather than worst-case, performance or response to white Gaussian noise? It turns out that there is indeed a great deal of performance variation among the family of $H^\infty$-optimal filters with respect to criteria other than robustness. It can be shown, for example, that the filter (4.48) has particularly poor average performance [26].

In fact, the non-uniqueness of the $H^\infty$-optimal filters has led many researchers to attempt to optimize other criteria over the family of these filters. We may refer to the super-optimal criterion, as well as to the mixed $H^2/H^\infty$ criterion that attempts to find the filter with the best average performance among all those that guarantee a prescribed worst-case bound [26].

However, we shall not go into any of these here. Instead we will be interested in the question of whether there is anything special about the LMS algorithm. Or whether it is just an arbitrary member of the family of $H^\infty$-optimal filters, not worthy of any further distinction. To answer this question we will now turn our attention to finding a stochastic interpretation of the LMS algorithm.

## 4.3 A Stochastic Interpretation

Recall that, even though the RLS algorithm can be considered as an algorithm that minimizes the deterministic quadratic forms (4.8) or (4.18), under suitable Gaussian assumptions on the signals involved, it also yields the ML or MAP estimates. The reason being that the deterministic quadratic form that RLS minimizes can be considered to be the (negative of the) exponent of a suitably chosen Gaussian probability density function (cf. (4.13)).

Now the LMS algorithm is related to the deterministic quadratic form

$$\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{i} \begin{bmatrix} \hat{d}(n) - \mathbf{h}(n)^T\mathbf{w} \\ y(n) - \mathbf{h}(n)^T\mathbf{w} \end{bmatrix}^T \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{d}(n) - \mathbf{h}(n)^T\mathbf{w} \\ y(n) - \mathbf{h}(n)^T\mathbf{w} \end{bmatrix}. \tag{4.49}$$

Indeed, referring to our derivation of all $H^\infty$-optimal filters in section 4.2.7, we first minimized the above quadratic form to obtain the quadratic form (4.44), or equivalently (4.46). Inspection of (4.46) shows that the choice $\hat{d}(n) = \mathbf{h}(n)^T\hat{\mathbf{w}}$, which leads to the LMS algorithm, *recursively maximizes* this quadratic form.

In other words, LMS performs the following optimization

$$\max_{\hat{d}(1),\ldots,\hat{d}(i)} \min_{\mathbf{w}} \left( \mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{i} \begin{bmatrix} \hat{d}(n) - \mathbf{h}(n)^T\mathbf{w} \\ y(n) - \mathbf{h}(n)^T\mathbf{w} \end{bmatrix}^T \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{d}(n) - \mathbf{h}(n)^T\mathbf{w} \\ y(n) - \mathbf{h}(n)^T\mathbf{w} \end{bmatrix} \right), \tag{4.50}$$

where the maximization is done *recursively*, i.e., $\hat{d}(1) = 0$, $\hat{d}(2)$ depends only on $y(1)$, $\hat{d}(3)$ depends only on $y(1), y(2)$, and so on.

Now at first sight it appears that (4.50) cannot be related to a stochastic problem, since the quadratic form is indefinite and so cannot be the exponent of a Gaussian probability density function. Moreover, we have a minimization over $\mathbf{w}$, but a maximization over the $\hat{d}(n)$. Nonetheless, if we write the quadratic form in (4.50) as

$$\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{i} \left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2 - \sum_{n=1}^{i} \left(\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2 \triangleq J,$$

then we can identify the first two terms as the (negative of the) exponent of a Gaussian density. More specifically, using (4.17), we can write

$$\frac{e^{-J}}{(2\pi)^{(i+m)/2}} = p\left(y(1),\ldots,y(N)\,|\mathbf{w},\mathbf{h}(1),\ldots,\mathbf{h}(N)\right) p(\mathbf{w}) \exp\left(\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right),$$

where we have assumed that $\mathbf{w}$ and the $v(n)$ are zero-mean independent Gaussian random variables with variance $\mu I$ and unity, respectively. Therefore we may also write

$$\begin{aligned}\frac{e^{-J}}{(2\pi)^{(i+m)/2} p\left(y(1),\ldots,y(N)\,|\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)} &= p\left(\mathbf{w}\,|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)\times \\ &\quad \exp\left(\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right). \qquad (4.51)\end{aligned}$$

Now it is not hard to show that for any $p\times p$ matrix $\mathbf{A}>0$

$$\begin{aligned}\int_{-\infty}^{\infty}\exp\left(-\begin{bmatrix}\mathbf{a}^* & \mathbf{b}^*\end{bmatrix}\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^* & \mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}\right)d\mathbf{a}d\mathbf{b} &= \sqrt{(2\pi\det\mathbf{A}^{-1})^p}\times \\ &\quad \exp\left(-\min_{\mathbf{a}}\begin{bmatrix}\mathbf{a}^* & \mathbf{b}^*\end{bmatrix}\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^* & \mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}\right).\end{aligned}$$

Therefore integrating both sides of (4.51) over $\mathbf{w}$ yields

$$\frac{\sqrt{(2\pi\det\mathbf{A}^{-1})^m}e^{-\min_{\mathbf{w}}J}}{(2\pi)^{(i+m)/2} p\left(y(1),\ldots,y(N)\,|\mathbf{h}(1),\ldots,\mathbf{h}(N)\right)} = E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)}\exp\left(\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right), \qquad (4.52)$$

where $\mathbf{A}=\mu^{-1}I+\sum_{n=1}^{i}\mathbf{h}(n)\mathbf{h}(n)^T$. Since the LHS of (4.52) depends on the $\hat{d}(n)$ only through $J$ we conclude from (4.50) that the LMS algorithm recursively solves the following problem

$$\min_{\hat{d}(1),\ldots,\hat{d}(i)} E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)}\exp\left(\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right). \qquad (4.53)$$

### 4.3.1 Risk-Sensitive Optimality

We can now formalize the result we have obtained in the following theorem.

**Theorem 3.** *Consider the model*

$$y(n)=\mathbf{h}(n)^T\mathbf{w}+v(n),\quad n>0$$

*where* $\mathbf{w}$ *and the* $v(n)$ *are zero-mean independent Gaussian random variables with variance* $\mu I$ *and unity, respectively. Assume further that (4.39) holds. Then the LMS algorithm (4.6) recursively solves the problem*

$$\min_{\hat{d}(1),\ldots,\hat{d}(i)} E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)} \exp\left(\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right),$$

*where* $\hat{d}(1)=0$*, and* $\hat{d}(n)$ *can depend only on past values of the observation,* $y(n-1)$*,* $y(n-2)$*,* $\ldots$*.*

*Moreover, if the* $\mathbf{h}(n)$ *are exciting, there exists no estimator that renders the cost function*

$$E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)} \exp\left(\gamma^{-2}\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right),$$

*finite for any* $\gamma < 1$*.*

In other words, under a standard Gaussian assumption on the disturbances, rather than minimize the mean-square prediction error,

$$E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)} \sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2, \tag{4.54}$$

which is what the RLS algorithm does, the LMS algorithm minimizes the *mean-exponential-square prediction error* (4.53).

The exponential quadratic cost (4.53) was first introduced in the control theory context by Jacobson [27]. It has also been championed by Whittle in statistics who calls it the *risk-sensitive* criterion [28]. In fact, Whittle introduces a whole family of cost functions parametrized by the positive scalar $\gamma$:

$$E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)} \exp\left(\gamma^{-2}\sum_{n=1}^{i}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2\right).$$

Whittle refers to estimators that minimize this criterion as *risk-averse*. The reason is that, compared to the mean-square criterion (4.54), the risk-sensitive criterion puts much larger penalty (in fact, exponentially larger penalty) on large values of the prediction error. In other words, the criterion is more concerned with the occasional occurrence of large values of prediction error, rather than with the frequent occurrence of moderate values of error. Some further intuition regarding the criterion (4.53) can be obtained by expanding the exponential function and noting that the criterion penalizes *all* the moments of the prediction error, and not just the second moment.

The smaller the parameter $\gamma$ is, the more risk-averse the estimator, since the criterion has a stronger exponential function. However, it turns out that $\gamma$ cannot be made arbitrarily small and there exists a critical value for $\gamma$ (called $\gamma_{opt}$) below which there exists no estimator that renders the risk-sensitive cost finite. In our problem, the critical value is $\gamma_{opt} = 1$, since if $\gamma < 1$ were possible it would mean that an $H^\infty$ estimator with $\gamma < 1$ is possible, which as we know is not the case. This is essentially the second statement of Theorem 3.

We should remark that the risk-sensitive optimality of the LMS algorithm goes very nicely with the robustness properties we have described earlier. Essentially, not tolerating large values of error is a way of saying that the algorithm is robust. In any event, the risk-sensitive optimality of LMS is a very interesting property that is not shared by any other of the $H^\infty$-optimal filters of Theorem 2. It is important for two reasons: first, it gives a non-obvious stochastic interpretation to the LMS algorithm and, second, it further emphasizes its special nature.

## 4.4 Nonstationary Signals and Tracking

We have seen that the LMS algorithm outperforms the RLS algorithm when we have nonstationary signals and need to track time-variations of the weight vector $\mathbf{w}$. At first sight, one may argue that comparison of the tracking abilities of these two algorithms is not fair since the LMS algorithm (4.6) has a constant gain vector, whereas the RLS algorithm (4.10-4.11) has a vanishing-to-zero gain vector and so can have no hope of tracking a time-varying $\mathbf{w}$. However, the comparison is fair if we consider that both algorithms deal with the same time-invariant model

$$y(n) = \mathbf{h}(n)^T\mathbf{w} + v(n),$$

with the only difference being that the RLS algorithm assumes that the disturbance sequence $v(n)$ is stationary and white (also Gaussian) and finds the linear-least-mean-squares (also least-mean-squares) estimate, whereas the LMS algorithm assumes only that the disturbance sequence is unknown and finds the $H^\infty$-optimal estimate. The point is that the RLS algorithm explicitly uses the fact that $\mathbf{w}$ is constant and so forces the gain vector to go to zero. The LMS algorithm, on the other hand, by virtue of its robustness to modeling errors and disturbance variation, safeguards us against a time-varying $\mathbf{w}$ by enforcing a non-zero gain vector.

### 4.4.1 Exponential-Windowing

The fact that the vanishing-to-zero gain vector of RLS leads to poor tracking performance is well-recognized in the literature and so various modifications to RLS that circumvent this shortcoming have been proposed. The most common one is to use an *exponential window* and to replace the deterministic cost function (4.18) with

$$\min_{\mathbf{w}} \left[ \mathbf{w}^T \sigma^2 \Pi_0^{-1} \mathbf{w} + \sum_{n=1}^{N} \lambda^{-n} \left( y(n) - \mathbf{h}(n)^T \mathbf{w} \right)^2 \right], \tag{4.55}$$

where $0 < \lambda < 1$ is referred to as a *forgetting* factor. Note that $\lambda^{-n}$ in the above cost function gives larger weight to the more recent observations and so in solving (4.55) we tend to "forget" the earlier values of the observation and thereby track the time-variations of $\mathbf{w}$. Clearly, the smaller $\lambda$, the more rapid the time-variation of $\mathbf{w}$ that we assume.

Now the same technique (of differentiation, say) that we used to solve (4.8) and (4.18) can be used to solve (4.55) and leads to the following weight update

$$\hat{\mathbf{w}}(i) = \hat{\mathbf{w}}(i-1) + \frac{\mathbf{P}(i-1)\mathbf{h}(i)}{1 + \mathbf{h}(i)^T \mathbf{P}(i-1)\mathbf{h}(i)} \left( y(i) - \mathbf{h}(i)^T \hat{\mathbf{w}}(i) \right), \quad \hat{\mathbf{w}}(0) = 0 \tag{4.56}$$

which is the recursion we were pursuing. $P(i)$ itself satisfies a Riccati recursion which can be computed to be

$$\mathbf{P}(i) = \lambda^{-1} \left( \mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{h}(i)\mathbf{h}(i)^T \mathbf{P}(i-1)}{1 + \mathbf{h}(i)^T \mathbf{P}(i-1)\mathbf{h}(i)} \mathbf{h}(i)^T \mathbf{P}(i-1) \right), \quad \mathbf{P}(0) = \sigma^{-2} \Pi_0. \tag{4.57}$$

The reason why, unlike (4.10), the gain vector in (4.56) does not go to zero is that, due to the factor $\lambda^{-1} > 1$ in (4.57), the matrix $P(i)$ does not go to zero.

Therefore using an exponential window in (4.55) alleviates the problem of a vanishing-to-zero-gain vector and improves the tracking performance. However, it is also possible to apply the exponential window to the $H^\infty$ setting and to obtain a robust version of (4.56-4.57). All one needs to do is replace problem (4.33) with

$$\gamma_{opt}^2 = \min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \max_{\mathbf{w}} \frac{\sum_{n=1}^{N+1} \lambda^{-n} \left( \hat{d}(n) - \mathbf{h}(n)^T \mathbf{w} \right)^2}{\mu^{-1} \mathbf{w}^T \mathbf{w} + \sum_{n=1}^{N} \lambda^{-n} \left( y(n) - \mathbf{h}(n)^T \mathbf{w} \right)^2}. \tag{4.58}$$

It turns out that finding an explicit formula for $\gamma_{opt}$ in (4.58) is not possible. However, in [29] section

11.3.1, the following bound for $\gamma_{opt}$ is obtained: Let

$$\bar{h} \triangleq \sup_i \mathbf{h}(i)^T \mathbf{h}(i) \quad , \quad \mathbf{R}_\lambda(i) \triangleq \sum_{j=1}^{i} \lambda^{i-j} \mathbf{h}(j)\mathbf{h}(j)^T \tag{4.59}$$

then

$$\gamma_{opt}^2 \leq \sup_i \frac{\bar{h} + \underline{\sigma}\left(\mathbf{R}_\lambda(i)\right)}{\lambda^i/\mu + \underline{\sigma}\left(\mathbf{R}_\lambda(i)\right)}, \tag{4.60}$$

where $\underline{\sigma}(\cdot)$ represesnts the minimum singular value of its argument. Moreover, it is shown that, for any $\gamma > \gamma_{opt}$, an $H^\infty$ predictor that achieves a worst-case energy gain of $\gamma^2$ is given by $\hat{d}(i) = \mathbf{h}(i)^T \hat{\mathbf{w}}(i-1)$, where $\hat{\mathbf{w}}(i-1)$ satisfies the recursion

$$\hat{\mathbf{w}}(i) = \hat{\mathbf{w}}(i-1) + \frac{\mathbf{P}(i-1)\mathbf{h}(i)}{1 + \mathbf{h}(i)^T \mathbf{P}(i-1)\mathbf{h}(i)} \left(y(i) - \mathbf{h}(i)^T \hat{\mathbf{w}}(i)\right), \quad \hat{\mathbf{w}}(0) = 0 \tag{4.61}$$

where now $\mathbf{P}(i-1)$ satisfies the recursion

$$\mathbf{P}(i)^{-1} = \lambda \mathbf{P}(i)^{-1} + \lambda \mathbf{h}(i)\mathbf{h}(i)^T - \gamma^{-2}\mathbf{h}(i+1)\mathbf{h}(i+1)^T, \quad \mathbf{P}(0)^{-1} = \mu^{-1} I - \gamma^{-2}\mathbf{h}(1)\mathbf{h}(1)^T. \tag{4.62}$$

The above estimator is one of many possible level-$\gamma$ estimators. But, as with LMS, it has the distinction of being the risk-sensitive-optimal solution.

### 4.4.2 General Time-Variation

The exponential-windowing scheme just described is really an ad-hoc way of dealing with a time-varying $\mathbf{w}$. In effect, what we are doing is assuming a constant $\mathbf{w}$, but assigning (exponentially) higher weight to more recent observations. A more fundamental approach would be to introduce a time-varying weight vector into the model directly. In other words, we should assume that the observed sequence is given by

$$y(n) = \mathbf{h}(n)^T \mathbf{w}(n) + v(n), \quad n > 0. \tag{4.63}$$

The question that then arises is how to describe the time-variation of $\mathbf{w}(n)$. A reasonable assumption is that $\mathbf{w}(n)$ itself satisfies the recursion

$$\mathbf{w}(n+1) = \alpha \mathbf{w}(n) + \sqrt{\mu(1-\alpha^2)}\,\mathbf{u}(n), \quad n > 0 \tag{4.64}$$

where $0 < \alpha < 1$ and $\mathbf{u}(n)$ is a disturbance vector often referred to as *process noise*. The reason for the for the coefficient $\sqrt{\mu(1-\alpha^2)}$ is that, if we assume that

$$E\mathbf{w}(0)\mathbf{w}(0)^T = \mu I \quad , \quad E\mathbf{u}(n)\mathbf{u}(m)^T = I\delta_{mn} \tag{4.65}$$

where $\delta_{mn}$ is the Kronecker delta, then the covariance matrix of the weight vector $\mathbf{w}(n)$ is constant for all time

$$E\mathbf{w}(n)\mathbf{w}(n)^T = \mu I, \quad n > 0.$$

In other words, even though the weight vector is time-varying its covariance matrix is constant for all time. The parameter $\alpha$ clearly determines the rate of the time-variation of $\mathbf{w}(i)$. The smaller it is, the faster the time-variation.

Under the assumptions (4.63-4.65), the linear least-mean-squares predictor of the weight vector$\hat{\mathbf{w}}(i|i-1) \stackrel{\Delta}{=} \hat{\mathbf{w}}(i-1)$ is given by the Kalman filter recursion

$$\hat{\mathbf{w}}(i) = \alpha\hat{\mathbf{w}}(i-1) + \frac{\alpha\mathbf{P}(i-1)\mathbf{h}(i)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}\left(y(i) - \mathbf{h}(i)^T\hat{\mathbf{w}}(i)\right), \quad \hat{\mathbf{w}}(0) = 0 \tag{4.66}$$

where now $\mathbf{P}(i-1)$ satisfies the Riccati recursion

$$\mathbf{P}(i) = \alpha^2\mathbf{P}(i-1) + \mu(1-\alpha^2)I - \frac{\alpha^2\mathbf{P}(i-1)\mathbf{h}(i)\mathbf{h}(i)^T\mathbf{P}(i-1)}{1+\mathbf{h}(i)^T\mathbf{P}(i-1)\mathbf{h}(i)}, \quad \mathbf{P}(0) = \mu I. \tag{4.67}$$

(For the details pertaining to the derivation of the above equations see, e.g., [8, 2].) Note that, due to the term $\mu(1-\alpha^2)I$ in the Riccati recursion, $\mathbf{P}(i)$ is positive definite for all times and so the gain vector in (4.66) cannot go to zero. Therefore our introduction of the time-varying model (4.64) automatically leads to an algorithm capable of tracking $\mathbf{w}(i)$.

### The Leaky LMS Algorithm

In the $H^\infty$ setting, the disturbance signal $\mathbf{u}(n)$ in (4.64) is assumed to be unknown and the predicted values of the uncorrupted output $\mathbf{h}(n)^T\mathbf{w}(n)$ are determined via the criterion

$$\gamma_{opt}^2 = \min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \max_{\mathbf{w},\mathbf{u}(\cdot)\in l^2} \frac{\sum_{n=1}^{N+1}\left(\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}(n)\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\mathbf{u}(n)^T\mathbf{u}(n) + \sum_{n=1}^{N}\left(y(n) - \mathbf{h}(n)^T\mathbf{w}(n)\right)^2}. \tag{4.68}$$

Note now that we have three disturbances: $\mathbf{w}$, the $\mathbf{u}(n)$ and the $v(n)$, which is why we have three terms in the denominator of the above energy gain.

Using $H^\infty$ theory one can show the following result.

**Theorem 4.** *Consider the model (4.63-4.64) and assume that $\mu \leq \frac{1}{\mathbf{h}(i)^T\mathbf{h}(i)}$ for all $i$. Then the optimal prediction error energy gain $\gamma^2_{opt}$, found via solving (4.68), satisfies*

$$\mu\left[\sup_i \mathbf{h}(i)^T\mathbf{h}(i)\right] \leq \gamma^2_{opt} \leq \bar{\gamma}^2 \leq 1, \tag{4.69}$$

*where $\bar{\gamma}^2 \leq 1$ is the infimum, over all $i$, of the largest positive solution to the quadratic equation*

$$\bar{\gamma}^4 - \frac{\mu\left(|\mathbf{h}(i)|^2 + |\mathbf{h}(i+1)|^2 - \alpha^2\mu(\mathbf{h}(i+1)^T\mathbf{h}(i))^2\right)}{(1+\mu|\mathbf{h}(i+1)|^2)}\bar{\gamma}^2 + \frac{\mu^2\left(|\mathbf{h}(i)|^2|\mathbf{h}(i+1)|^2 - \alpha^2(\mathbf{h}(i+1)^T\mathbf{h}(i))^2\right)}{(1+\mu|\mathbf{h}(i+1)|^2)} = 0,$$

*where we have used the notation $|\mathbf{a}|^2 = \mathbf{a}^T\mathbf{a}$.*

Note that the above theorem implies that the optimal energy gain can be less than one.[10] Although it is possible to give the expression for an arbitrary level-$\gamma$ predictor, we shall not do so here. Rather we will give an interesting, though slightly suboptimal, predictor that achieves $\gamma = 1$.

To this end, a simple variation to the LMS algorithm that has been proposed for tracking applications is the so-called *leaky LMS algorithm*,

$$\hat{\mathbf{w}}(i) = \alpha\hat{\mathbf{w}}(i-1) + \alpha\mu\mathbf{h}(i)\left(y(i) - \mathbf{h}(i)\hat{\mathbf{w}}(i-1)\right), \quad \hat{\mathbf{w}}(0) = 0. \tag{4.70}$$

Note that, compared to the LMS algorithm (4.6), the leaky LMS algorithm attenuates the weight vector estimate by the factor $0 < \alpha < 1$. This allows for the algorithm to "forget" earlier data and allows better tracking.

Let us now study the consequences of algorithm (4.70) to the time-varying model (4.63-4.64). Now if we define the prediction error of the weight vector at time $i+1$ using the observations up to time $i$ as $\tilde{\mathbf{w}}(i) = \mathbf{w}(i+1) - \hat{\mathbf{w}}(i-1)$, then subtracting (4.70) from (4.64) allows us to write

$$\tilde{\mathbf{w}}(i) = \alpha\left(I - \mu\mathbf{h}(i)\mathbf{h}(i)^T\right)\tilde{\mathbf{w}}(i-1) - \alpha\mu\mathbf{h}(i)v(i) + \sqrt{(1-\alpha^2)\mu}\mathbf{u}(i), \tag{4.71}$$

where we have made use of the fact that $y(i) - \hat{\mathbf{w}}(i-1) = v(i) + \tilde{\mathbf{w}}(i-1)$. This now allows us to write the

[10] The reason for this is the existence of the exponential decay factor $0 < \alpha < 1$.

mapping from the variables $\{\mu^{-1/2}\tilde{\mathbf{w}}(i-1), v(i), \mathbf{u}(i)\}$ to the variables $\{\mu^{-1/2}\tilde{\mathbf{w}}(i), \mathbf{h}(i)^T\tilde{\mathbf{w}}(i-1)\}$ as

$$\begin{bmatrix} \mu^{-1/2}\tilde{\mathbf{w}}(i) \\ \mathbf{h}(i)^T\tilde{\mathbf{w}}(i-1) \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha\left(I-\mu\mathbf{h}(i)\mathbf{h}(i)^T\right) & -\alpha\mu^{1/2}\mathbf{h}(i) & \sqrt{1-\alpha^2}I \\ \mu^{1/2}\mathbf{h}(i)^T & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mu^{-1/2}\tilde{\mathbf{w}}(i-1) \\ v(i) \\ \mathbf{u}(i) \end{bmatrix}. \tag{4.72}$$

We will now show that the mapping $\mathbf{A}$ is a contraction. Indeed, from the above equation it follows that

$$I - \mathbf{A}\mathbf{A}^* = \begin{bmatrix} \alpha^2\mu\left(1-\mu|\mathbf{h}(i)|^2\right)\mathbf{h}(i)\mathbf{h}(i)^T & -\alpha\mu^{1/2}\left(1-\mu|\mathbf{h}(i)|^2\right)\mathbf{h}(i) \\ -\alpha\mu^{1/2}\left(1-\mu|\mathbf{h}(i)|^2\right)\mathbf{h}(i)^T & 1-\mu|\mathbf{h}(i)|^2 \end{bmatrix} \geq 0,$$

where the last inequality follows from the fact that the block $(2,2)$ entry satisfies $1-\mu|\mathbf{h}(i)|^2 \geq 0$ and its Schur complement is

$$\alpha^2\mu\left(1-\mu|\mathbf{h}(i)|^2\right)\mathbf{h}(i)\mathbf{h}(i)^T - \alpha^2\mu\left(1-\mu|\mathbf{h}(i)|^2\right)\mathbf{h}(i)\mathbf{h}(i)^T = 0.$$

The fact that $\mathbf{A}$ is a contraction implies that the norm of the output variables is less than the norm of the input variables, i.e.,

$$\mu^{-1}|\tilde{\mathbf{w}}(i)|^2 + \left(\mathbf{h}(i)^T\tilde{\mathbf{w}}(i-1)\right)^2 \leq \mu^{-1}|\tilde{\mathbf{w}}(i-1)|^2 + v(i)^2 + |\mathbf{u}(i)|^2. \tag{4.73}$$

Adding all of the above equations from time $i=1$ to time $i=N$ implies that

$$\mu^{-1}|\tilde{\mathbf{w}}(N)|^2 + \sum_{n=1}^{N}\left(\mathbf{h}(i)^T\tilde{\mathbf{w}}(n-1)\right)^2 \leq \mu^{-1}|\mathbf{w}(0)|^2 + \sum_{n=1}^{N}|\mathbf{u}(i)|^2 + \sum_{n=1}^{N}v(i)^2, \tag{4.74}$$

from which we infer

$$\frac{\sum_{n=1}^{N}\left(\mathbf{h}(i)^T\tilde{\mathbf{w}}(n-1)\right)^2}{\mu^{-1}|\mathbf{w}(0)|^2 + \sum_{n=1}^{N}|\mathbf{u}(i)|^2 + \sum_{n=1}^{N}v(i)^2} \leq 1, \tag{4.75}$$

for all possible disturbances $\mathbf{w}(0)$, $\mathbf{u}(i)$ and $\mathbf{v}(i)$. In other words, we have shown that the leaky LMS algorithm guarantees a worst-case prediction error energy gain of unity.

**Theorem 5.** *Consider the model (4.63-4.64) and assume that* $\mu \leq \frac{1}{\mathbf{h}(i)^T\mathbf{h}(i)}$ *for all* $i$. *Then leaky LMS* *algorithm (4.70) guarantees*

$$\frac{\sum_{n=1}^{N}\left(\mathbf{h}(i)^T\tilde{\mathbf{w}}(n-1)\right)^2}{\mu^{-1}|\mathbf{w}(0)|^2+\sum_{n=1}^{N}|\mathbf{u}(i)|^2+\sum_{n=1}^{N}v(i)^2}\leq 1,$$

*for all possible disturbances* $\mathbf{w}(0)$, $\mathbf{u}(i)$ *and* $\mathbf{v}(i)$.

*In particular, if* $\mu = \frac{1}{\mathbf{h}(i)^T\mathbf{h}(i)}$ *then the leaky LMS algorithm is* $H^\infty$*-optimal, since in this case*

$$\min_{\hat{d}(1),\ldots,\hat{d}(N+1)}\ \max_{\mathbf{w},\mathbf{u}(\cdot)\in l^2}\frac{\sum_{n=1}^{N+1}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}(n)\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{N}\mathbf{u}(n)^T\mathbf{u}(n)+\sum_{n=1}^{N}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}(n)\right)^2}=1.$$

*Finally, if* $\mathbf{w}(0)$ *and the* $\mathbf{u}(i)$ *and* $v(i)$ *are independent zero-mean Gaussian random variables with variances* $\mu I$*, I and 1, respectively, then the leaky LMS algorithm is risk-senstive optimal in the sense that it solves the problem*

$$\min_{\hat{d}(1),\ldots,\hat{d}(N+1)} E_{|y(1),\ldots,y(N),\mathbf{h}(1),\ldots,\mathbf{h}(N)}\exp\left(\sum_{n=1}^{N}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}(n)\right)^2\right).$$

The first statement of the above theorem follows from the arguments preceding the theorem. The second statement follows from (4.69). The third statement can be proven using an argument similar to the one that we presented for the risk-sensitive optimality of the LMS algorithm.

In any event, Theorem 5 demsonstrates the robustness of the leaky LMS algorithm (4.70) for applications where the unknown weight vector varies with time. It also demonstrates the robustness of the LMS algorithm itself for such applications, provided the time variation of the weight vector is slow, i.e., $\alpha \approx 1$, since in this case there is little difference between the LMS algorithm (4.6) and its leaky version (4.70).

## 4.5 Further Remarks

In addition to yielding a new interpretation for the LMS algorithm and providing it with a rigorous basis, the results described so far have lent themselves to various generalizations and to several new results. We close this chapter by listing some of these.

### 4.5.1 Filtered Errors and the Normalized LMS Algorithm

In this chapter we have focused on prediction errors and predicting the uncorrupted output $\mathbf{h}(n)^T\mathbf{w}$. It is also possible to look at so-called *filtered* errors

$$\mathbf{h}(n)^T\mathbf{w} - \mathbf{h}(n)^T\hat{\mathbf{w}}(n), \tag{4.76}$$

i.e., on the error in estimating the uncorrupted output $\mathbf{h}(n)^T\mathbf{w}$ using observations up to *and including* the current time instant $n$. In this case the $H^\infty$-optimal algorithm turns out to be the normalized LMS algorithm

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \frac{\mu}{1+\mu|\mathbf{h}(n)|^2}\mathbf{h}(n)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(i-1)\right), \tag{4.77}$$

which is a commonly used variant to the LMS algorithm. It turns out that optimal energy gain is $\gamma_{opt}^2 = 1$ and that this is true, *irrespective of the learning rate* $\mu$. Results such as the non-uniqueness of the $H^\infty$-optimal estimators, the risk-sensitive optimality, tracking properties, etc., all extend to the normalized LMS algorithm. For a proof of these results the reader may refer to [30].

### 4.5.2 LMS with Time-Varying Learning Rate

In many applications, one uses the LMS algorithm with a time-varying step-size (or learning rate), i.e.,

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mu(n)\mathbf{h}(n)\left(y(n) - \mathbf{h}(n)^T\hat{\mathbf{w}}(i-1)\right). \tag{4.78}$$

In this case, it is straightforward to show that if the vectors $\mu(n)^{1/2}\mathbf{h}(n)$ are exciting, and $\mu(n)\mathbf{h}(n)^T\mathbf{h}(n) \leq 1$ for all $n$, then the above LMS algorithm with time-varying step-size solves the following minimax problem

$$\min_{\hat{d}(1),\ldots,\hat{d}(N+1)} \max_{\mathbf{w}} \frac{\sum_{n=1}^{N+1}\mu(n)\left(\hat{d}(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2}{\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\mu(n)\left(y(n) - \mathbf{h}(n)^T\mathbf{w}\right)^2} = 1. \tag{4.79}$$

### 4.5.3 $H^\infty$ Norm Bounds for the RLS Algorithm

In order to compare the robustness of least-squares algorithms (such as RLS) with $H^\infty$-optimal algorithms (such as LMS) it is useful to obtain $H^\infty$ norm bounds for these algorithms. In [31] this is done for the RLS and Kalman filtering algorithms. In particular, it is shown that for the RLS algorithm

$$\left(\sqrt{1+\mu\bar{h}^2}-1\right)^2 \;\leq\; \max_{\mathbf{w}} \frac{\sum_{n=1}^{N+1}\left(\hat{d}(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2}{\mu^{-1}\mathbf{w}^T\mathbf{w}+\sum_{n=1}^{N}\left(y(n)-\mathbf{h}(n)^T\mathbf{w}\right)^2} \;\leq\; \left(\sqrt{1+\mu\bar{h}^2}+1\right)^2, \quad (4.80)$$

where $\bar{h}^2 = \sup_n \mathbf{h}(n)^T\mathbf{h}(n)$. Thus, unlike the LMS algorithm where the optimal energy gain was independent of $\mu$ and the $\mathbf{h}(n)$, for RLS it highly depends on these. Moreover, for large values of $\mu$, the upper and lower bounds in (4.80) grow as $\sqrt{\mu}$. This is reminiscent of the robustness properties of LMS, where the learning rate had to be small enought to guarantee $H^\infty$-optimality. More importantly, it shows that the unregularized least-squares problem (4.8) (corresponding to $\mu = \infty$) can be highly non-robust with respect to prediction errors.

### 4.5.4 Mixed $H^2/H^\infty$ Problems

As mentioned in section 4.2.7 the $H^\infty$-optimal predictors are highly non-unique. It therefore may be expected that the different $H^\infty$-optimal filters in this family have very different performances with respect to other criteria. For example, the LMS algorithm was shown to be risk-sensitive optimal under suitable Gaussian assumptions on the disturbances. One other possibility is to search for the $H^\infty$-optimal filter that has the least expected prediction error energy. Such an estimator would, in a sense, achieve the best of both worlds: among all estimators that achieve a certain level of robustness, it would have the best average performance. Estimators that have such a property are referred to as mixed $H^2/H^\infty$-optimal. Finding the mixed $H^2/H^\infty$-optimal solution for general estimation problems is an important open problem. However, for the adaptive filtering problem we have been studying, it turns out that the solution can be found and is given below [32]

$$\min_{\hat{d}(1),\ldots,\hat{d}(N)} \sum_{n=1}^{N}\left(\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}_{rls}(n-1)\right)^2, \quad \text{subject to}$$

$$\sum_{n=1}^{n}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\\ y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\end{bmatrix}^T\begin{bmatrix}-1+\mu|\mathbf{h}(n)|^2 & \mu|\mathbf{h}(n)|^2\\ \mu|\mathbf{h}(n)|^2 & 1+\mu|\mathbf{h}(n)|^2\end{bmatrix}^{-1}\begin{bmatrix}\hat{d}(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\\ y(n)-\mathbf{h}(n)^T\hat{\mathbf{w}}(n-1)\end{bmatrix}\geq 0 \quad (4.81)$$

where $\hat{\mathbf{w}}_{rls}(n)$ denotes the RLS estimate of the weight vector. The above problem is a quadratic program and can be readily solved.

### 4.5.5 A Time-Domain Feedback Analysis

Using some of the ideas presented here, a time-domain feedback analysis of recursive adaptive schemes, including gradient-based and Gauss-Netwon filters, has been developed [25] for both the FIR and IIR contexts. The analysis highlights an intrinsic feedback structure in terms of a feedforward lossless or contractive map (based on the LMS algorithm) and a feedback memoryless of dynamic map. The structure lends itself to analysis via energy arguments and via standard tools in system theory, such as the small gain theorem [23, 24]. It further suggests choices for the learning rates in order to enforce robust performance (along the lines of $H^\infty$ theory), as well as to improve the convergence speed of the adaptive algorithms.

### 4.5.6 Nonlinear Problems

The results presented in this chapter are for linear adaptive filters and can be somewhat generalized to nonlinear adaptive filters (such as neural networks) if one linearizes these models around some suitable operating point. Using this approach, it can be shown (see [33]) that, for nonlinear problems, instantaneous-gradient-based methods (such as backpropagation [4]) are locally $H^\infty$-optimal. This means that if the initial estimate of the weight vector is close enough to its true value, and if the disturbances are small enough, then the maximum energy gain from the disturbances to the output prediction errors is arbitrarily close to one. Global $H^\infty$-optimal filters can also be found in the nonlinear case, but they have the drawback of being infinite-dimensional (see [34]).

## 4.6 Conclusion

In this chapter we showed that the LMS algorithm is $H^\infty$-optimal. This result solves a long standing issue of finding a rigorous basis for the LMS algorithm, and also confirms its robustness. We have argued that, compared to exact least-squares solutions, the wide use of the LMS algorithm over a broad range of applications is best explained by its robustness to modeling errors and disturbance variation, rather than its simplicity, computational efficiency, or numerical stability (for all of which exact least-squares solutions can be designed to be competitive). We further obtained a stochastic interpretation of the LMS algorithm in terms of risk-sensitive optimality: under a Gaussian assumption on the disturbances, LMS minimizes the expected exponential of the prediction error energy. We also demonstrated the ability of the LMS algorithm to work in non-stationary environments that require tracking the unknown weight vector, and especially showed this for the leaky-LMS algorithm.

It is appropriate to conclude this chapter by emphasizing two important points. First, that, even though there has been only relatively recent interest in $H^\infty$ estimation theory, there has existed an $H^\infty$-optimal estimation algorithm that has been widely used in practice for four decades. The robustness properties, and various statistical interpretations, of the LMS algorithm that we have discussed in this chapter go far beyond the original intentions of the inventors of this algorithm. And so it is a testament to the insight and intuition of Widrow and Hoff that their algorithm possesses such remarkable properties. Second, that the results presented here clearly indicate the strong connection between adaptive filtering and robust estimation. Over the last four decades most of the contributions to adaptive filtering have been coming from the field of statistical estimation theory. Therefore further scrutiny of the connection to robust estimation is certainly worthwhile and may lead to many fruitful developments.

# Bibliography


[1] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[2] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ, 2002.

[3] B. Widrow and M. Hoff, "Adaptive switching circuits," *IRE WESCON Conv. Rec., Pt. 4*, pp. 96–104, 1960.

[4] D. Rumelhart and J. McLelland, *Parallel Distributed Processing: Explorations in Microstructure of Cognition*. MIT Press, Cambridge, MA, 1986.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Englewood Cliffs, NJ, 1998.

[6] P. Billingsley, *Probability and Measure*. Wiley-Interscience, 1995.

[7] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Transactions on Information Theory*, vol. 20, pp. 146–181, 1974.

[8] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, Englewood Cliffs, NJ, 2000.

[9] H. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, 2001.

[10] A. Sayed and T. Kailath, "A state-space approach to RLS adaptive filtering," *IEEE Signal Processing Magazine*, vol. 11, pp. 18–60, 1994.

[11] J. Potter and R. Stern, "Statistical filtering of space navigation measuerments," *Proceedings of the AIAA Guidance and Control Conference*, 1963.

[12] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering Transactions of the ASME, Series D*, vol. 82, no. 11, pp. 35–45, 1960.

[13] M. Dahleh and J. Pearson, “$l_1$-optimal compensators for continuous-time systems,” *IEEE Transactions on Automatic Control*, vol. 32, no. 10, pp. 889–895, 1987.

[14] G. Zames, “Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms and approximate inverses,” *IEEE Transactions on Automatic Control*, vol. 26, pp. 301–320, 1981.

[15] A. Saberi, P. Sannuti, and B. Chen, *$H^2$ Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[16] P. Duren, *Theory of $H^p$ Spaces*. Dover, 2000.

[17] B. Francis, *A Course of $H^\infty$ Control Theory*. Springer-Verlag, 1987.

[18] A. Feintuch, *Robust Control Theory in Hilbert Space*. Springer-Verlag, 1998.

[19] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, “State-space solutions to standard $H^2$ and $H^\infty$ control problems,” *IEEE Transactions on Automatic Control*, vol. 34, pp. 831–847, 1989.

[20] H. Kimura, *Chain-Scattering Approach to $H^\infty$ Control*. Birkhauser, Boston, 1997.

[21] T. Basar and P. Bernhard, *$H^\infty$-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhauser, Boston, 1991.

[22] B. Hassibi, A. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control: A Unified Approach to $H^2$ and $H^\infty$ Theories*. SIAM, 1999.

[23] H. Khalil, *Nonlinear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 2001.

[24] M. Vidyasagar, *Nonlinear System Analysis*. SIAM, 2002.

[25] A. Sayed and M. Rupp, “Error-energy bounds for adaptive gradient algorithms,” *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1982–89, 1996.

[26] B. Halder, B. Hassibi, and T. Kailath, “Mixed $H^2/H^\infty$ estimation: Preliminary analytic characterization and a numerical solution,” *Proceedings of the 13th World Congress International Federation of Automatic Control. Vol.J. Identification II, Discrete Event Systems*, pp. 37–42, 1997.

[27] D. Jacobson, “Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic games,” *IEEE Transactions on Automatic Control*, vol. 18, pp. 124–131, 1973.

[28] P. Whittle, *Risk-Sensitive Optimal Control*. John Wiley & Sons, 1990.

[29] B. Hassibi, *Indefinite Metric Spaces in Estimation, Control and Adaptive Filtering*. PhD thesis, Stanford University, 1996.

[30] B. Hassibi, A. Sayed, and T. Kailath, "$H^\infty$-optimality of the LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 267–280, 1996.

[31] B. Hassibi and T. Kailath, "$H^\infty$ bounds for least-squares estimators," *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 309–314, 2001.

[32] B. Hassibi and T. Kailath, "On adaptive filtering with combined least-mean-squares and $H^\infty$h criteria," *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1570–74, 1998.

[33] B. Hassibi, A. Sayed, and T. Kailath, "$H^\infty$-optimality criteria for LMS and backpropagation," *Advances in Neural Information Processing Systems*, vol. 6, pp. 351–359, 1994.

[34] B. Hassibi and T. Kailath, "$H^\infty$-optimal training algorithms and their relation to backpropagations," *Advances in Neural Information Processing Systems*, vol. 7, pp. 191–99, 1995.