Power vs. Performance Tradeoffs for Reduced Resolution LMS Adaptive Filters

Riten Gupta and Alfred O. Hero
EECS Department
University of Michigan

Abstract

Low power implementation of digital adaptive filters for channel equalization and interference cancelling is an important aspect of wireless communications transceiver design. In this paper we provide a low power design methodology for the widely used finite precision LMS adaptive cahnnel equalizer. We first derive expressions for the increase in mean square error of the digital adaptive filter when the total power dissipated is decreased by reducing both the number of data bits and filter coefficient bits. We then derive a constraint on the data and coefficient wordlengths to avoid the so called stopping, or slowdown, phenomenon. We also obtain expressions for the power-optimal bit-allocation factor which determines the optimal proportion of bits to allocate to the data, the remainder being allocated to the coefficients. Numerical studies are presented for an exponential memory ISI channel with Gaussian training sequence.

Keywords

Roundoff and Quantization Analysis, Low-Power Channel Equalization, Optimal Bit Allocation

EDICS numbers: 2.6.4, 2.8.1

This research was supported in part by the Department of Defense Research & Engineering (DDR&E) Multidisciplinary University Research Initiative (MURI) on Low Energy Electronics Design for Mobile Platforms and managed by the Army Research Office (ARO) under grant ARO DAAH04-96-1-0337. Corresponding author: Riten Gupta, robby@eecs.umich.edu, (734)-764-4110 (voice), (734)-763-8041 (fax). 1301 Beal Ave. #4301, Ann Arbor, MI 48109-2122

List of Figures

Figure 14 *Quantizer model.*

Figure 15 *Regions defined in (A-3).*

Figure 16 P_T *vs.* B_T *for different power relations with* $\eta_a \approx 1.4\ mW$ *and* $\eta_t \approx 6.8\ mW$.

I. Introduction

In a battery powered wireless receiver, the adaptive equalization function consumes a significant portion of total battery power. For example the SINCGARS combat radio used by the US Army consumes on the average 7 Watts in receive mode of which more than 1 Watt (over 14 % of total power) is consumed by the channel equalizer [1]. Therefore the channel equalization function is a prime target for power reduction in these handsets. There have been many digital hardware design strategies proposed for power reduction including: reduction of supply voltage, reduction of clock speed and data rate, parallelization and pipelining of operations, using sign-magnitude arithmetic, and differential encoding of data [2], [3]. Another technique, which is the springboard for this paper, is the reduction of the number of bits used to represent the data and control variables in the digital circuit [4]. The bit width reduction strategy is very highly leveraged since it reduces the power dissipation everywhere in the data and control flow paths. This strategy is also very versatile since it can be applied to any hardware architecture and can be easily adjusted in real time by dynamically switching-off bus lines and register bits. However, bit width reduction generally entails a degradation in algorithm performance, as measured by adaptive algorithm convergence rate, steady state mean square error (MSE), and subsequent probability of symbol error. This paper provides an analysis of MSE degradation vs power reduction, due to reducing coefficient and data wordlengths, for a widespread class of adaptive algorithms popularly known as the LMS algorithm.

The LMS algorithm was introduced by Widrow [5] and is one of the most common adaptation algorithms used in practical systems such as adaptive channel equalizers [6], adaptive beamforming antenna arrays [7], and adaptive interference cancellation systems [8]. In channel equalization, the LMS algorithm adapts the filter coefficients of an FIR filter driven by the received data based on an error signal formed by subtracting a training signal from the filtered output. In [9] it was shown that uniform scalar quantization of the error signal is close to optimal for the case of Gaussian inputs. The finite precision LMS algorithm can be viewed as an infinite precision LMS algorithm implemented with separate uniform scalar quantizers in the data path and in the filter coefficient paths. In this paper we study the case where the quantizers can have different resolutions. To obtain significant power reduction, it may be beneficial to apply different quantizer resolutions during the transient acquisition phase and the steady state tracking phase of the algorithm. For example, Nicol has implemented incorporated "dynamic precision tuning" of filter coefficients in a low power 128-tap adaptive modem [10]. However, in this paper we focus on steady state analysis of finite precision effects when the coefficient and data wordlengths are distinct but fixed over time.

Since quantization is a non-invertible and non-differentiable operation an exact statistical analysis of the finite precision LMS is very difficult. Several methods of approximate analysis have been proposed. Alexander [11] and Caraiscos and Liu [12] performed first and second order statistical analyses, respectively, of the real-valued finite precision LMS algorithm under the assumption of a linear white noise model for the quantization error. Under similar assumptions, Cioffi [**?**] performed an analysis of the finite precision LMS adaptive echo canceler, which was later extended to block LMS by Cioffi and Ho [13]. The linear white noise model does not account for the stopping phenomenon which occurs when the finite precision LMS weight update falls below the least significant bit (LSB) of the coefficient quantizer. Bermudez and Bershad [14], showed that the stopping phenomenon is actually a slowdown phenomenon which does not stop the algorithm's updating, but causes a severe increase in convergence time. In [15], [16] they bypassed the linear white noise assumption and derived an accurate recursive approximation for an LMS algorithm with quantized weight update and infinite precision data, which they called quantized LMS. Their work showed that the white noise assumption is invalid for this type of algorithm, as the stopping phenomenon always occurs. In addition, the steady state MSE of quantized LMS is very close to that of the corresponding infinite precision algorithm, and differs greatly from the theoretical value given in [12]. Unfortunately, their analysis does not easily extend to the finite precision LMS algorithm where both coefficients and data are represented with finite wordlength. While the linear white noise assumption is admittedly a coarse approximation, there appears to be no practical alternative to the approach of Caraiscos and Liu [12] for analysis of the effects of coefficient resolution and data resolution on MSE. Fortunately, in many operating regimes of practical interest, it has been demonstrated that this approach gives useful and accurate performance predictions [11], [13], [?], [17], [?]. In fact, unlike the quantized LMS algorithm of [15], [16], the finite precision LMS algorithm can operate without experiencing the stopping phenomenon. It is in this operating region that the model used in [12] gives accurate results. Accordingly, we adopt the white noise approach for our analysis of finite precision LMS.

While our results are directly applicable to the generic finite precision LMS algorithm in a variety of applications, for concreteness, we concentrate on the case of channel equalization with training. We first present an asymptotic bound which relates power dissipation to register length for samples of a Gaussian random process. We then derive a formula for the power-per-iteration of the finite precision power-of-two step size LMS algorithm under fixed point complex arithmetic, cyclic updating of the data stack, and table lookup implementation of multiplication. Next we obtain analytical expressions for the increase in steady state mean square error (MSE) due to finite precision in the absence of the stopping phenomenon which generalizes the formulas of Caraiscos and Liu [12] to the case of complex data and coefficients. After this we present a simple constraint on the data wordlength, coefficient wordlength, and adaptation step size, μ, which prevents the stopping phenomenon. We then derive a pair of optimal bit-allocation factors which minimize the amount of increase in MSE subject to two constraints: 1) a total bit-width constraint (total bit budget); and 2) a total power consumption constraint (total power budget). We conclude that assigning more bits to coefficients than data is necessary to avoid the stopping phenomenon and in general gives better steady state MSE performance under a total power budget constraint. This conclusion mirrors a similar result reported by Eweda *etal* [17] for the finite precision sign LMS algorithm. Finally we specialize to equalization of a single pole IIR channel.

II. Register Length and Power

It is well known that the power consumed by the operation of loading successive time samples of a random real-valued sequence into a binary B-bit register is proportional to the average number of bit flips per unit time in the register [18]. While for a uniform white sequence the average number of bit flips is $B/2$, in general this average can be much less than $B/2$ for a correlated sequence. This is because for correlated sequences most significant bits (MSB's) have lower probability of transitioning than least significant bits (LSB's). Several models for the power consumption of register loading have been proposed [18]. We propose to use the following approximate upper bound (derived in Appendix A) on power consumption for a B-bit register, and a zero mean wide-sense stationary Gaussian random sequence:

$$P_B \quad \leq \quad B\eta \cdot \left[1 - \frac{1}{2}\mathrm{erf}\left(\left[2^B\sqrt{2R(0) - 2R(1)}\right]^{-1}\right)\right] \tag{1}$$

where η is the power dissipation-per-bit, which depends on factors such as switching load capacitance and supply voltage, and $R(\tau)$ is the autocorrelation function of the random sequence. Note that the bound (1) is conservative: if we constrain the right hand side of (1) to some maximum tolerable power dissipation, P_{max} say, then a circuit design which uses register bit width $B = B_{max}$ which solves the equation $P_{max} = B\eta \cdot [1 - \frac{1}{2}\mathrm{erf}([2^B\sqrt{2R(0) - 2R(1)}]^{-1})]$ is guaranteed to consume less power than P_{max}.

A plot of P_{max} versus B is given in Figure 1 for an AR(1) sequence with real pole located at a_1. For $|a_1| < 0.8$ the power dissipation increases almost linearly as a function of B. Note from Figure 1 that for more highly correlated sequences stored in registers with few bits, the power increase due to increasing bit width is less severe. This is due to the fact that only the LSB's flip with high probability in two successive samples of a highly correlated sequence, while for an uncorrelated uniform sequence, all bits have equal probability (0.5) of flipping.

III. Finite Precision Adaptive Channel Equalization

Figure 2 shows the block diagram of an adaptive equalization system with two different quantizers, denoted Q_d and Q_c, applied to the complex data and to the complex filter coefficients of an adaptive p-tap FIR filter. Here s_k is a complex (baseband) training signal, y_k is a scaled version of s_k, x_k is the complex received (baseband) data, and $\hat{y}_k$ is a linear estimate of y_k given the p samples $\underline{x}_k = [x_k, \ldots, x_{k-p+1}]^T$ and filter coefficients $\underline{w}_k = [w_{0,k}, \ldots, w_{p-1,k}]^H$. The scaling factor a represents a right shift of one or more bits and is used to prevent overflow of the quantized weight vector.

The quantizers Q_d and Q_c are allocated B_d bits plus sign and B_c bits plus sign, respectively, to the real and imaginary parts of their inputs. We assume that the input sequences have been scaled to lie between -1 and $+1$. As in [12] we assume that the effect of the operators Q_d and Q_c is to add to the input a complex white noise of variance $2\sigma_d^2 = \frac{1}{6}2^{-2B_d}$ for Q_d and $2\sigma_c^2 = \frac{1}{6}2^{-2B_d}$ for Q_c. The standard LMS algorithm adapts the filter coefficients in an attempt to minimize the quadratic surface specified by the mean square error: $E[|y_k - \hat{y}_k|^2]$ [5], [19]. The finite precision LMS algorithm, implements a recursive filter coefficient update of the form

$$\underline{w}'_{k+1} = \underline{w}'_k + Q_c\left(\mu\ \underline{x}'_k e'^*_k\right) \tag{2}$$

where

$$e'_k = y'_k - Q_d\left(\underline{w}'^H_k \underline{x}'_k\right). \tag{3}$$

is the quantized error signal. Here μ is the adaptive gain parameter which controls the convergence properties of the algorithm. We assume that μ is chosen to be a power of two so that multiplication by μ is implemented by right shifts. We have used primed symbols to represent quantized values. That is, $\underline{x}'_k = Q_d(\underline{x}_k)$, $y'_k = Q_d(y_k)$, and $\underline{w}'_k = Q_c(\underline{w}_k)$ is the quantized weight vector used in the algorithm. Also, * denotes complex conjugation. We assume that in the computation of the inner product in (3), the partial sums are themselves quantized. Therefore, $Q_d(\underline{w}'^H_k \underline{x}'_k) = \sum_{i=0}^{p-1} Q_d(w'_{i,k} x'_{k-i})$ and the noise added to the inner product has variance $2p\sigma_d^2$.

A. Power Consumption of LMS Algorithm

The total power per iteration of the finite precision LMS algorithm is determined by power dissipation of shift, add, multiply, memory load, and memory store operations. This depends on the specific circuit implementation of the FIR filter and control circuitry. The bulk of the power dissipation usually comes from add and multiply operations. In Appendix B, we derive the following formula for power-per-iteration of LMS, considering only adds and multiplies, under the assumption that $\mu = 2^{-q}$ for $q \in \{0, \ldots, B_c - 1\}$:

$$P_T = 4p[(2B_d + B_c + 2)\eta_a + (3B_d + B_c)\eta_t]. \tag{4}$$

This expression is linear in the number of bits B_d and B_c and assumes fixed point complex arithmetic, cyclic updating of the data stack $[x_k, \ldots, x_{k-p+1}]^T$ by overwriting x_{k-p+1} with x_{k+1}, and multiplication using table lookup as opposed to adding partial products. In (4), we have defined generic power coefficients η_a and η_t representing power consumption per add per bit and power consumption per table lookup operation per bit, respectively.

B. Performance of LMS Algorithm

The performance of the LMS adaptive algorithm is typically characterized by two quantities: the speed of convergence and the excess MSE [5], [6], [21]. We assume that s_k and x_k are both wide sense stationary random sequences. Similar to [12], we make the following high resolution assumptions:

> The quantization errors of quantizers Q_d and Q_c are zero mean, white, with variances $2\sigma_d^2$ and $2\sigma_c^2$, respectively. Furthermore, these errors are independent of the quantizer inputs. The process x_k is circular Gaussian. The step size μ is greater than 0, and B_c, $B_d \geq 1$. Finally, we assume that the sequence s_k has been properly scaled to lie within the unit circle. (5)

In addition, we assume that the stopping phenomenon does not occur. Later, we derive a constraint on B_c, B_d, and μ which ensures that the stopping phenomenon will not occur.

B.1 Mean Convergence

Define the $p \times p$ covariance matrix $R_{x'} = E[\underline{x}'_k \underline{x}'^H_k]$ of the quantized data and let this matrix have real non-negative eigenvalues $\{\lambda'_i\}_{i=1}^p$. Further define the cross correlation vector $R_{x'y'} = E[\underline{x}'_k y'^*_k]$. Assume the gain parameter μ satisfies the condition

$$0 < |1 - \mu\lambda'_i| < 1, \qquad i = 1, \ldots, p. \tag{6}$$

In Appendix C we show that the filter coefficients of the finite precision LMS algorithm converge in the mean to a set of optimal weights, $\underline{w}'^o$.

$$\lim_{k\to\infty} E[\underline{w}'_k] = \underline{w}'^o \tag{7}$$

where

$$\underline{w}'^o = R_{x'}^{-1} R_{x'y'}.$$

When the finite precision LMS algorithm converges, the MSE converges as a decaying exponential with the $1/e$ time constant of the slowest mode equal to $\tau_{3dB} = 1/(-\max_i \ln(|1-\mu\lambda'_i|))$, called the adaptation time constant. Note that the speed of convergence generally increases as μ increases.

B.2 Steady State Weight Error Covariance

In Appendix C we derive an expression for the steady state quantized weight error covariance. Let $R_x = E[\underline{x}_k \underline{x}_k^H]$, $R_{xy} = E[\underline{x}_k y_k^*]$, and let $P_k = E[(\underline{w}'_k - \underline{w}_k)(\underline{w}'_k - \underline{w}_k)^H]$ where $\underline{w}_k$ is the weight vector at time k of the infinite precision LMS algorithm. Then assuming P_k converges as $k \to \infty$, it converges to the steady state covariance matrix

$$P_k = P = p\sigma_c^2(2I - \mu R_x)^{-1} + \frac{\sigma_c^2}{\mu} R_x^{-1}. \tag{8}$$

B.3 Excess Mean Square Error

Under the assumptions (5), the following expression for the steady state mean square error is derived in Appendix C:

$$\xi = E[|s_k - \hat{s}'_k|^2] = \xi_{min} + \xi_{excess} + \xi_q \tag{9}$$

where

$$\xi_{min} = \frac{1}{a^2}\left(\sigma_y^2 - R_{xy}^H R_x^{-1} R_{xy}\right)$$

is the optimal mean-square error of the Wiener filter for the known channel,

$$\xi_{excess} = \frac{1}{2}\mu \mathrm{tr}(R_x)\xi_{min}$$

is the excess MSE due to misadjustment,

$$\xi_q = \alpha_c\ 2^{-2B_c} + \alpha_d\ 2^{-2B_d} \tag{10}$$

where

$$\alpha_c = \frac{p}{12\mu a^2}, \qquad \alpha_d = \frac{\|\underline{w}^o\|^2 + p}{6a^2}$$

and $\underline{w}^o = R_x^{-1}R_{xy}$ is the optimal Wiener weight vector for the standard infinite precision complex LMS algorithm.

The term ξ_q is the excess MSE due to quantization of the data and filter coefficients. The first term in the expression (10) is the excess MSE due only to quantization of the filter coefficients while the second term represents the MSE due to quantization of the data. Note that for small μ, the term α_c dominates the excess MSE due to quantization unless B_c is made large. This implies that for small step sizes, a high resolution is required for the filter coefficients. Also worth noting is that ξ_q increases in p at a linear rate, decreases in μ at an inverse linear rate, and decreases in B_d and B_c at an exponential rate. Therefore, the total number of bits allocated gives more leverage over excess MSE than any other of the design parameters.

With these relations the increase in MSE due to finite precision, ξ_q, can be plotted as a function of B_d and B_c. A plot of the increase in MSE is given in Figs. 3 and 4 for the case of white s_k and for x_k generated by passing s_k through a single pole IIR filter ($p = 2$) with pole at $a_1 = 0.8$ and with LMS gain parameter $\mu = 1/4$ and scaling factor $a = 1/8$. The vertical plane on the surface plot and the thick line on the contour plot divide the B_c, B_d plane into two regions. In the next section we will show that the value ξ_q is valid only in the region to the left of the vertical plane in Figure 3 (right of the thick line in Figure 4) and the other region should be avoided when choosing B_c and B_d.

B.4 Stopping (Slowdown) Phenomenon

The stopping phenomenon occurs when one or more components of the input to the quantizer Q_c in (2) falls below the LSB of Q_c [12], [15], [16], [20]. Stated mathematically, this condition is

$$|\mu x'_{k-i} e'^{*}_{k}| < \frac{\Delta_c}{2} \tag{11}$$

where $i \in \{0, \ldots, p-1\}$ and $\Delta_c = 2^{-B_c}$ is the granularity of the coefficient quantizer. During the initial stages of adaptation, before condition (11) is satisfied, the finite precision LMS algorithm behavior does not differ significantly from the inifinite precision algorithm [15], [16]. When condition (11) is met at iteration k, the ith component of the vector $\underline{w}'_k$ does not get updated and the finite precision algorithm's learning curve begins to diverge from that of the infinite precision algorithm. To determine the point at which this divergence occurs (the slowdown point), one could square both sides of (11) and apply the independence assumption [12] to conclude that the divergence, or slowdown, occurs when

$$\xi'_k = E[|e'_k|^2] < \frac{2^{-2(B_c+1)}}{\mu^2 E[|x'_{k-i}|^2]}.$$

However, experiments have shown that this inequality in general does not accurately predict the slowdown point. To get a more accurate result, we assume that slowdown begins when

$$P\left(|\mu x'_{k-i} e'^{*}_{k}| < \frac{\Delta_c}{2}\right) > 1-\epsilon \tag{12}$$

where ϵ is a small positive number. Since the finite precision algorithm's behavior does not differ significantly from the infinite precision algorithm before slowdown, the slowdown condition (12) becomes

$$P\left(|\mu x_{k-i} e^{*}_{k}| < \frac{\Delta_c}{2}\right) > 1-\epsilon \tag{13}$$

where $e_k = y_k - \hat{y}_k$ is the error signal in the infinite precision algorithm. Now define $g_k = x_{k-i} e^*_k$ and assume for simplicity that g_k is Gaussian with mean zero and variance $\sigma_x^2 \sigma_{e,k}^2$ where $\sigma_x^2 = E[|x_k|^2]$ and $\sigma_{e,k}^2 = E[|e_k|^2]$ is the mean square error of the infinite precision algorithm. Before slowdown we have $\sigma_{e,k}^2 \approx \xi'_k$. Then using (13), slowdown will begin when

$$\operatorname{erf}\left(\frac{\Delta_c}{2\mu\sigma_x\sqrt{2\xi'_k}}\right) = 1-\epsilon.$$

This is equivalent to

$$\xi'_k = \frac{2^{-2B_c+2q-3}}{\sigma_x^2[\operatorname{erf}^{-1}(1-\epsilon)]^2} = \xi'_{slow} \tag{14}$$

where $q = -\log_2 \mu$.

Now from the definition of e'_k, we have $\xi'_\infty = a^2\xi + 2\sigma_d^2$ in the absence of slowdown. Therefore, assuming slowdown does not occur, for a fixed B_d

$$\xi'_\infty > a^2(\xi_{min} + \xi_{excess} + \xi_q|_{B_c=\infty}) + 2\sigma_d^2.$$

Assuming μ is small,

$$\begin{aligned}\xi'_\infty &> a^2\xi_{min} + 2\sigma_d^2(\|\underline{w}^o\|^2 + p + 1)\\ &= \xi'_{floor}.\end{aligned} \tag{15}$$

Figure 5 shows a sample learning curve of a finite precision channel equalizer which experiences slowdown along with the values ξ'_{slow} and ξ'_{floor}.

Since the finite and infinite precision algorithms agree closely before slowdown, slowdown can be prevented by choosing B_c such that

$$\xi'_{slow} < \xi'_{floor}.$$

This is equivalent to

$$2^{-2B_c+2q-3} < \sigma_x^2[\mathrm{erf}^{-1}(1-\epsilon)]^2[a^2\xi_{min} + 2\sigma_d^2(\|\underline{w}^o\|^2 + p + 1)].$$

Therefore, a sufficient condition for preventing slowdown is

$$B_c > B_d + \nu \tag{16}$$

where

$$\nu = q - \frac{3}{2} - \frac{1}{2}\log_2\left(\frac{\sigma_x^2}{6}(\|\underline{w}^o\|^2 + p + 1)[\mathrm{erf}^{-1}(1-\epsilon)]^2\right).$$

IV. Optimal Bit Allocation Strategies

We present expressions for the optimum allocation of bits to data versus filter coefficients under two constraints: total number of bits and total power consumption.

Assume that there are a total of B_T bits plus 2 sign bits which are available to allocate between data and coefficients, i.e. $B_T = B_d + B_c$. Further define the data bit allocation factor $\rho = B_d/B_T$. Then we have the obvious relations

$$B_d = \rho B_T, \qquad\qquad B_c = (1-\rho)B_T. \tag{17}$$

A. Total Bit Budget Constraint

Under a constraint on B_T the objective becomes to minimize the increase ξ_q in MSE with respect to ρ. Graphically, this is the same as minimizing ξ_q along the diagonal line $B_T = B_d + B_c$ of slope -1 in the B_c, B_d plane. shown on Fig. 4. Note that the region to the left of the thick line in Figure 4 must be avoided. It is shown in Appendix D that ξ_q is convex as a function of ρ with a single minimum occurring at the point $\rho = \rho^*$

$$\begin{aligned}\rho^* &= \frac{1}{4B_T}\log_2\left(\frac{\alpha_d}{\alpha_c}\right) + \frac{1}{2}\\ &= \frac{1}{4B_T}\log_2\left(2\mu\frac{\|\underline{w}^o\|^2 + p}{p}\right) + \frac{1}{2},\end{aligned} \tag{18}$$

Observe that as B_T increases, ρ^* approaches the standard textbook allocation of $1/2$. However, for low B_T, the standard allocation is suboptimal.

Note that condition (16) is equivalent to

$$\rho < \frac{1}{2} - \frac{\nu}{2B_T} = \rho_{slow}. \tag{19}$$

Therefore, since ξ_q is convex as a function of ρ, the optimal value of ρ is

$$\rho^B = \min\{\rho^*, \rho_{slow}\}.$$

B. Total Power Budget Constraint

Under a constraint on total power budget, P_T, we can use (4) and (17) to re-express the total combined number of bits B_T as a function of ρ and P_T

$$B_T = \frac{P_T - 8p\eta_a}{4p[\rho(\eta_a + 2\eta_t) + \eta_a + \eta_t]} \tag{20}$$

In Appendix D we show that for $B_T \geq 2$, ξ_q is once again a convex function of ρ with unique minimum at $\rho = \rho^{**}$

$$\rho^{**} = \frac{P_T - 8p\eta_a + 2p(\eta_a + \eta_t)\log_2\left[\frac{\alpha_d}{\alpha_c} \cdot \frac{\eta_a + \eta_t}{2\eta_a + 3\eta_t}\right]}{2(P_T - 8p\eta_a) - 2p(\eta_a + 2\eta_t)\log_2\left[\frac{\alpha_d}{\alpha_c} \cdot \frac{\eta_a + \eta_t}{2\eta_a + 3\eta_t}\right]}. \tag{21}$$

Similar to the bit budget constraint, ρ^{**} converges to the standard $1/2$ allocation as P_T becomes large, while for low P_T, the standard allocation is suboptimal.

Again applying constraint (16), the optimal allocation is

$$\rho^P = \min\{\rho^{**}, \rho_{slow}\}$$

where ρ_{slow} is calculated using (19) and (20).

V. Numerical Example

Here we briefly consider the case of an IIR channel with a single pole at $a_1 = 0.8$, a Gaussian signal s_k of unit variance, noiseless training sequence $y_k = as_k$, and a 2-tap LMS filter with gain coefficient $\mu = 1/4$ and scaling factor $a = 1/8$. For convenience, the gain control (see Figure 2) is set to a constant of $1/5.5$. This corresponds to a rather severe exponential memory channel with intersymbol interference (ISI) extending over 5 to 10 data samples.

The first step in designing the finite precision equalizer is to choose ϵ. Although an analytical solution is not yet available, we expect ϵ to be dependent on p more heavily than any other design parameter. In particular, as p becomes large, slowdown is not likely to occur if only a few taps remain unchanged. To reflect this phenomenon, ϵ must decrease as p increases. For this example ($p = 2$), ϵ was chosen to be

0.025. Figure 6 shows the graph of ξ'_k and ξ'_{slow}, calculated using (14) with $\epsilon = 0.025$, for two cases in which B_c, B_d, and μ differ. Observe that the estimate ξ'_{slow} is accurate for both cases.

Having chosen ϵ, we next calculate $\nu = 1.89$ and (16) becomes $B_c \geq B_d + 2$. Figure 7 shows the learning curves for several finite precision equalizers satisfying (16) along with the predicted value of the steady state MSE given by (9).

To determine the power coefficients η_a and η_t for this example, several adders and multipliers with varying bit-width were simulated using the Epoch CAD package. The energy consumption of each adder and multiplier was determined and by using linear least squares fits to this data, the adder energy per bit and multiplier energy per bit were obtained. These figures were then multiplied by the assumed clock cycle of 50 MHz to give the coefficients $\eta_a \approx 1.4$ mW and $\eta_t \approx 6.8$ mW. The simulated multipliers were shift-add (partial product) multipliers. Although our analysis considers table lookup multipliers, it is clear from Figure 16 that the chosen value of η_t gives a good power approximation for either multiplier.

Figure 8 shows the B_T-constrained data bit allocation factor ρ^* as a function of B_T as well as the maximum allowable allocation satisfying (16), ρ_{slow}. It is clear that for all B_T, $\rho_{slow} < \rho^*$ and therefore the optimal allocation factor is $\rho^B = \rho_{slow}$. Figure 9 shows the P_T-constrained data bit allocation factor ρ^{**} as a function of P_T as well ρ_{slow}. Again it is clear that for all P_T of interest, the optimal allocation factor is $\rho^P = \rho_{slow}$. Finally, Figure 10 shows the resultant MSE, ξ, as a function of P_T using the optimal allocation $\rho = \rho_{slow}$ plotted on a log scale.

While these results show that the bit allocation ρ_{slow} is always optimal for this example, it should be remembered that this is not always the case. For example, consider the design of a finite precision LMS equalizer with the following parameters: $p = 128$, $\|\underline{w}^o\|^2 = 1$, $a = 1$, $\mu = 1/8$, $\sigma_s^2 = 0.01$, $\sigma_x^2 = 0.09$. Figures 11 and 12 show that for this case the optimal bit allocation factors are ρ^* and ρ^{**}. Figure 13 shows the MSE as a function of P_T for this case with $\rho = \rho_{slow}$ and $\rho = \rho^{**}$. Observe that as expected, the MSE with ρ^{**} is less.

VI. Conclusion

We have presented a design methodology by which low power implementation of LMS adaptive channel equalizers can be achieved. We have derived expressions for optimal bit allocation under combined register length constraints and total power constraints while avoiding the stopping, or slowdown, phenomenon. These expressions can easily be specialized to a specific hardware implementation for computation of the number of bits to allocate to data and filter coefficients. A general conclusion is that the standard design strategy of allocating an equal number of bits to the data and filter coefficients is optimal only as the power or register length constraints become very large. Furthermore, this 50% allocation can yield undesired slowdown in adaptation. For typical LMS implementations, it is optimal to allocate more bits to the coefficients than to the data.

Appendix A
Derivation of (1)

Consider a zero-mean wide-sense stationary Gaussian random sequence x_k, with autocorrelation sequence $R(\tau) = E[x_k x_{k+\tau}]$, quantized to B bits and loaded into a B-bit register, whose maximum and minimum values are +1 and -1, respectively. The quantizer cells are shown in Figure 14.

Let n_k denote the number of bits that flip when loading the value x_{k+1} into the register. Then

$$E[n_k] = \sum_{i=1}^{B} i\ P(n_k = i) = \sum_{i=0}^{B} P(n_k > i).$$

Next we note that

$$\sum_{i=1}^{B} P(n_k = i) \leq \sum_{i=0}^{B} P(n_k > i) \leq \sum_{i=1}^{B} B \cdot P(n_k = i).$$

Therefore,

$$P(n_k > 0) \leq E[n_k] \leq B \cdot P(n_k > 0). \tag{A-1}$$

Now $P(n_k > 0)$ is equal to the probability that at least one bit flips when loading x_{k+1} into the register. This is equal to one minus the probability that no bits flip. Therefore

$$P(n_k > 0) = 1 - P(n_k = 0). \tag{A-2}$$

No bits in the register will flip if x_k and x_{k+1} lie in the same quantizer cell. The granularity of the quantizer is $\Delta = 2^{-B+1}$. Define

$$\begin{aligned} S &= \bigcup_{m=-\infty}^{\infty} \{(x_k, x_{k+1}) \mid x_k, x_{k+1} \in (\Delta(m-1/2), \Delta(m+1/2))\} \\ T &= \{(x_k, x_{k+1}) \mid x_k, x_{k+1} \in (-\infty, 3\Delta/2 - 1) \cup (1 - 3\Delta/2, +\infty)\} \\ U &= \left\{(x_k, x_{k+1}) \mid |x_k - x_{k+1}| \leq \Delta/\sqrt{2}\right\}. \end{aligned} \tag{A-3}$$

These regions are shown in Figure 15. Then $P(n_k = 0) = P(S \cup T)$ and

$$P(S) \leq P(n_k = 0) \leq P(S) + P(T). \tag{A-4}$$

For $P(S)$ we have,

$$P(S) = \sum_{m=-\infty}^{\infty} \int_{\Delta(m-1/2)}^{\Delta(m+1/2)} \int_{\Delta(m-1/2)}^{\Delta(m+1/2)} f_{x_k, x_{k+1}}(x, y)\ dxdy.$$

For each integer, m, let $x_m = y_m = \Delta \cdot m$. By the Mean Value Theorem, for each m, there exist $\bar{x}_m, \bar{y}_m$ such that

$$\begin{aligned} P(S) &= \sum_{m=-\infty}^{\infty} f_{x_k, x_{k+1}}(\bar{x}_m, \bar{y}_m)\Delta^2 \\ &= \sum_{m=-\infty}^{\infty} f_{x_k, x_{k+1}}(x_m, y_m)\Delta^2 + \sum_{m=-\infty}^{\infty} \left[f_{x_k, x_{k+1}}(\bar{x}_m, \bar{y}_m) - f_{x_k, x_{k+1}}(x_m, y_m)\right] \Delta^2 \end{aligned} \tag{A-5}$$

Next we note that

$$P(U) = \sum_{m=-\infty}^{\infty} \left[\int_{x=(m-1)\Delta}^{m\Delta} \int_{y=(2m-1)\Delta-x}^{x+\Delta} f_{x_k,x_{k+1}}(x,y)\ dxdy + \int_{x=m\Delta}^{(m+1)\Delta} \int_{y=x-\Delta}^{-x+(2m+1)\Delta} f_{x_k,x_{k+1}}(x,y)\ dxdy \right].$$

Again by the Mean Value Theorem, there exist x'_m, y'_m such that

$$\begin{aligned} P(U) &= \sum_{m=-\infty}^{\infty} f_{x_k,x_{k+1}}(x'_m, y'_m)(\Delta\sqrt{2})^2 \\ &= \sum_{m=-\infty}^{\infty} f_{x_k,x_{k+1}}(x_m, y_m) \cdot 2\Delta^2 + \sum_{m=-\infty}^{\infty} \left[f_{x_k,x_{k+1}}(x'_m, y'_m) - f_{x_k,x_{k+1}}(x_m, y_m) \right] \cdot 2\Delta^2 . \end{aligned} \tag{A-6}$$

From (A-5) and (A-6),

$$P(S) = \frac{P(U)}{2} + \delta \tag{A-7}$$

where $\delta = \Delta^2 \sum_{m=-\infty}^{\infty} \left[f_{x_k,x_{k+1}}(\bar{x}_m, \bar{y}_m) - f_{x_k,x_{k+1}}(x'_m, y'_m) \right]$. Now $P(T)$ can be bounded using the Chebyshev inequality:

$$\begin{aligned} P(T) &= P\left(|x_k| \geq 1 - 3\Delta/2, |x_{k+1}| \geq 1 - 3\Delta/2\right) \\ &\leq \gamma R(0) \end{aligned} \tag{A-8}$$

where $\gamma = (1 - 3\Delta/2)^{-2}$.

From (A-2), (A-4), (A-7), and (A-8)

$$1 - \frac{P(U)}{2} - (\delta + \gamma R(0)) \leq P(n_k > 0) \leq 1 - \frac{P(U)}{2} - \delta.$$

As, $x_k - x_{k+1}$ is Gaussian with mean zero and variance $2R(0) - 2R(1)$, we have

$$P(U) = \operatorname{erf}\left(\left[2^B \sqrt{2R(0) - 2R(1)} \right]^{-1} \right).$$

Next with η the power consumption per bit flip, we have: $P_B = \eta E[n_k]$, where P_B is the average power consumed during each register load. Using this and (A-1), we obtain the following two bounds on P_B:

$$\begin{aligned} P_B &\leq B\eta \cdot \left[1 - \frac{1}{2} \operatorname{erf}\left(\left[2^B \sqrt{2R(0) - 2R(1)} \right]^{-1} \right) - \delta \right] \\ P_B &\geq \eta \cdot \left[1 - \frac{1}{2} \operatorname{erf}\left(\left[2^B \sqrt{2R(0) - 2R(1)} \right]^{-1} \right) - (\delta + \gamma R(0)) \right]. \end{aligned}$$

Finally, since δ is bounded by the maximum variation in f and since $\delta \to 0$ and $\gamma \to 1$ as $\Delta \to 0$, the approximate bound (1) holds.

Appendix B
Derivation of (4)

We consider a hardware implementation of the finite precision LMS algorithm in which all data are stored in $B_d + 1$ bits and all coefficients in $B_c + 1$ bits. We assume that all right shifted quantities are rounded correctly (away from zero) and that right shifting consumes negligible energy compared to additions and multiplications.

Two multipliers are used: one with B_c-bit multiplier and B_d-bit multiplicand, and one with B_d-bit multiplier and multiplicand. Only magnitudes are multiplied and determination of input magnitude and output sign is assumed to consume negligible energy. The multipliers are assumed to be direct table lookup multipliers, each implemented by use of a ROM indexed by the magnitudes of the multiplier and multiplicand, which stores the product (magnitude) quantized to $B_1 + B_2$ bits where B_1 and B_2 are the number of bits used for the multiplier and multiplicand [23]. Altough the power consumed by such a multiplier is proportional to $2^{B_1+B_2}$, we use a simplified formula and assume the power consumed during a (real) multiplication is $P_{B_1\times B_2,real} = \eta_t(B_1 + B_2)$ where η_t is the power per bit of a table lookup operation.

A complex multiplication of $a + bj$ stored in B_1 bits plus sign by $c + dj$ stored in B_2 bits plus sign requires four real B_1-bit by B_2-bit multiplications and two additions of $B_1 + B_2 + 1$ bits. Therefore, the power consumed by such a complex multiplication is $P_{B_1\times B_2,complex} = 4\eta_t(B_1 + B_2) + 2\eta_a(B_1 + B_2 + 1)$ where η_a is the power per bit of a real adder.

In the LMS update formula given by (2) and (3), the calculation of the inner product $\underline{w}_k'^H \underline{x}_k'$ requires p complex multiplications of numbers stored in B_c bits plus sign by numbers stored in B_d bits plus sign and $p - 1$ complex $(B_d + 1)$-bit additions. Subtracting this inner product from y_k' requires an additional complex addition of $B_d + 1$ bits. Next, multiplying the conjugate of this quantity (e_k') by $\underline{x}_k'$ we have p complex multiplications in which both multiplier and multiplicand are stored in B_d bits plus sign. Since multiplication by μ requires only a shift, we are left with the addition by $\underline{w}_k'$. This operation requires p complex additions of B_c+1 bits. Therefore, the total power consumed during one iteration of the complex finite precision LMS algorithm is

$$P_T \quad = \quad 4p\eta_t(3B_d + B_c) + 2p\eta_a(3B_d + B_c + 2) + 2p\eta_a(B_d + B_c + 2)$$

which is equivalent to (4).

We can derive a similar formula for multiplication using partial product accumulation. In such a multiplier, each real B_1-bit by B_2-bit multiplication requires approximately B_1 additions of B_2 bits [24] and therefore has a power consumption of $B_1B_2\eta_a$. The total real operations are the same as with the table lookup multiplier. This gives the following relation for power per iteration of LMS using partial product accumulation multiplication:

$$P_T = 4p\eta_a(B_d^2 + B_dB_c + 2B_d + B_c + 2). \tag{A-9}$$

Using equations (4) and (A-9) along with (17), we can plot P_T as a function of B_T for both power relations. These plots are shown in Figure 16 for $\rho = 1/2$, $\eta_a \approx 1.4$ mW, $\eta_t \approx 6.8$ mW, and $p = 2$. Note that the formula derived using partial product accumulation is quadratic in B_T while the formula derived using table lookup multiplication is linear in B_T.

Appendix C
Derivation of (7), (8) and (10)

Consider the following mean square error cost function:

$$\xi'_k = E[|e'_k|^2]. \tag{A-10}$$

where e'_k is given in (3). Using the definition of Q_d and e'_k

$$e'_k = y'_k - \underline{w}'^H_k \underline{x}'_k + n^*_k$$

where $n^*_k = \underline{w}'^H_k \underline{x}'_k - Q_d(\underline{w}'^H_k \underline{x}'_k)$. As in [12] we approximate n_k as zero-mean, white, with variance $2p\sigma_d^2$ and uncorrelated with $\underline{w}'^H_k \underline{x}'_k$ and y'_k. Therefore, we have

$$\xi'_k = E[|e'_k|^2] = E[|y'_k - \underline{w}'^H_k \underline{x}'_k|^2] + 2p\sigma_d^2.$$

Taking the gradient of this cost function with respect to $\underline{w}'_k$ we get

$$\frac{d\xi'_k}{d\underline{w}'_k} = 2R_{x'}\underline{w}'_k - 2R_{x'y'}.$$

Setting this to zero and solving for $\underline{w}'_k$ we obtain the optimal weight vector (8).

Now, from (2) and (3), we see that the finite precision LMS update is of the form

$$\underline{w}'_{k+1} = \underline{w}'_k + \mu\underline{x}'_k(y'^*_k - \underline{x}'^H_k \underline{w}'_k + n_k) + \underline{\sigma}_k$$

where n_k is defined above and $\underline{\sigma}_k$ is a zero-mean complex white noise vector whose components have variance $2\sigma_c^2$. Using the method of [21] we obtain (7) and (6). First define

$$\delta\underline{w}'_{k+1} = \underline{w}'_{k+1} - \underline{w}'_k = \mu\underline{x}'_k e'^*_k + \underline{\sigma}_k,$$

and

$$\epsilon'_k = y'_k - \underline{w}'^{oH}\underline{x}'_k, \qquad \tilde{\underline{w}}'_k = \underline{w}'_k - \underline{w}'^o.$$

Then we have

$$\begin{aligned} e'_k &= \epsilon'_k + \underline{w}'^{oH}\underline{x}'_k - \underline{w}'^H_k \underline{x}'_k + n^*_k \\ &= \epsilon'_k - \tilde{\underline{w}}'^H_k \underline{x}'_k + n^*_k. \end{aligned}$$

Next use, $\tilde{\underline{w}}'_{k+1} - \tilde{\underline{w}}'_k = \delta \underline{w}'_{k+1}$ to obtain

$$\tilde{\underline{w}}'_{k+1} = (I - \mu \underline{x}'_k \underline{x}'^H_k)\tilde{\underline{w}}'_k + \mu \underline{x}'_k \epsilon'^*_k + \mu \underline{x}'_k n_k + \underline{\sigma}_k.$$

Similar to [21], we assume that $\underline{x}'_k$ and ϵ'_k are independent. Using this and the fact that n_k and $\underline{\sigma}_k$ are zero-mean, we see that

$$E[\tilde{\underline{w}}'_{k+1} | \underline{x}'_1, \ldots, \underline{x}'_{k-1}, \epsilon'_1, \ldots, \epsilon'_{k-1}] = \tilde{\underline{w}}'_k - \mu R_{x'} \tilde{\underline{w}}'_k$$

and taking expectations,

$$E[\tilde{\underline{w}}'_{k+1}] = (I - \mu R_{x'}) E[\tilde{\underline{w}}'_k]$$

which is completely analogous to the standard unquantized LMS algorithm. Using the same analysis as in [21] (Sec. 5.2), we conclude that $E[\underline{w}'_k] \to \underline{w}'^o$ as long as condition (6) is met.

To derive (8) and (10) we first write the weight update formula for the infinite precision LMS algorithm

$$\underline{w}_{k+1} = \underline{w}_k + \mu \; \underline{x}_k \left(y^*_k - \underline{x}^H_k \underline{w}_k \right). \tag{A-11}$$

We define

$$\sigma^2_d = \frac{1}{12} 2^{-2B_d}, \qquad \sigma^2_c = \frac{1}{12} 2^{-2B_c}. \tag{A-12}$$

These are the variances of the quantization noises associated with quantizers having B_d and B_c bits (plus sign), respectively. We use primed symbols to represent quantized values and define the following quantities

$$\begin{aligned} \underline{x}'_k &= \underline{x}_k + \underline{\alpha}_k \\ y'_k &= y_k + \beta_k \\ \underline{w}'_k &= \underline{w}_k + \underline{\rho}_k. \end{aligned} \tag{A-13}$$

The components of the vector $\underline{\alpha}_k$ and β_k are complex numbers whose real and imaginary parts are assumed uncorrelated and have variances σ^2_d. Therefore, the variance of β_k and of each component of $\underline{\alpha}_k$ is $2\sigma^2_d$.

Thus we have the representation for the quantized value of the filter output

$$\hat{y}'_k = Q_d(\underline{w}'^H_k \underline{x}'_k) = \underline{w}^H_k \underline{x}_k + \underline{\rho}^H_k \underline{x}_k + \underline{w}^H_k \underline{\alpha}_k + \eta_k \tag{A-14}$$

where η_k is a complex quantization noise with variance $2p\,\sigma^2_d$. This variance arises as a result of quantizing the individual products of the inner product $\underline{w}'^H_k \underline{x}'_k$ and then summing them [12]. As in [12], we have ignored the noise product term $\underline{\rho}^H_k \underline{\alpha}_k$ in (A-14), as its power is of order $\sigma^2_d \sigma^2_c$.

The total error is now

$$\begin{aligned} s_k - \hat{s}'_k &= \frac{1}{a}(y_k - \underline{w}^H_k \underline{x}_k) - \\ &\quad \frac{1}{a}(\underline{\rho}^H_k \underline{x}_k + \underline{w}^H_k \underline{\alpha}_k + \eta_k) \end{aligned} \tag{A-15}$$

The first term on the right hand side of (A-15) is the error of the infinite precision algorithm. The second term is the error due to quantization and will be denoted e_q. Under the hypothesis (5), these terms are uncorrelated and the MSE for the finite precision LMS algorithm is

$$\begin{aligned} \xi &= \frac{1}{a^2} E[|y_k - \underline{w}_k^H \underline{x}_k|^2] + \\ & \frac{1}{a^2} E[|\underline{\rho}_k^H \underline{x}_k + \underline{w}_k^H \underline{\alpha}_k + \eta_k|^2]. \end{aligned} \tag{A-16}$$

The first term on the right hand side of (A-16) is the MSE of the infinite precision LMS algorithm and is equal to $\xi_{min} + \xi_{excess}$ [5], [21]. The second term is the excess MSE due to quantization,

$$\xi_q = E[|e_q|^2].$$

Under the assumptions (5), $\underline{\alpha}_k$, β_k, $\underline{\rho}_k$, and η_k are all uncorrelated and

$$\xi_q = \frac{1}{a^2} \left(E\left[|\underline{\rho}_k^H \underline{x}_k|^2\right] + E\left[|\underline{w}_k^H \underline{\alpha}_k|^2\right] + E\left[|\eta_k|^2\right] \right). \tag{A-17}$$

The term, $E[|\eta_k|^2]$ is equal to $2p\ \sigma_d^2$. For the term $E[|\underline{w}_k^H \underline{\alpha}_k|^2]$ we obtain

$$E[|\underline{w}_k^H \underline{\alpha}_k|^2] = 2\sigma_d^2\ E[\|\underline{w}_k\|^2].$$

Note that this differs from the real case studied in [12] by a factor of 2. In the steady state, this becomes

$$E[|\underline{w}_k^H \underline{\alpha}_k|^2] = 2\sigma_d^2\ \left(\|\underline{w}^o\|^2 + \frac{1}{2} p\ \mu\ \xi_{min} \right). \tag{A-18}$$

Finally, for the first term in (A-17)

$$E[|\underline{\rho}_k^H \underline{x}_k|^2] = \mathrm{tr}\left(E\left[\underline{\rho}_k \underline{\rho}_k^H\right] R_x \right). \tag{A-19}$$

It remains to find an expression for $E[\underline{\rho}_k \underline{\rho}_k^H]$. Let $e_k = y_k - \underline{w}_k^H \underline{x}_k$ and define $\zeta_k = \beta_k - \eta_k$. Note that the variance of ζ_k is $2(1+p)\sigma_d^2$. Then we have

$$e'_k = y'_k - \hat{y}'_k = e_k - \underline{\rho}_k^H \underline{x}_k - \underline{w}_k^H \underline{\alpha}_k + \zeta_k$$

The finite precision LMS update equations (2), (3) can now be written

$$\begin{aligned} \underline{w}'_{k+1} &= \underline{w}'_k + Q_c\left[\mu\ \underline{x}'_k \left(y'_k - Q_d[\underline{w}'^H_k \underline{x}'_k]\right)^*\right] \\ &= \underline{w}'_k + Q_c[\mu\ \underline{x}'_k e'^*_k] \\ &= \underline{w}'_k + \mu(\underline{x}_k e^*_k - \underline{x}_k \underline{x}_k^H \underline{\rho}_k - \\ & \quad \underline{x}_k \underline{\alpha}_k^H \underline{w}_k + \underline{x}_k \zeta^*_k + \underline{\alpha}_k e^*_k) + \underline{\sigma}_k \end{aligned}$$

where $\underline{\sigma}_k$ is a quantization noise of variance $2\sigma_c^2$ and all second-order noise terms, e.g. $\underline{\rho}_k^H \underline{\alpha}_h$ have been ignored.

Using (A-11) and (A-13), $\underline{\rho}_k$ can be seen to obey the recursion

$$\underline{\rho}_{k+1} = F_k \underline{\rho}_k + \underline{b}_k \tag{A-20}$$

where

$$F_k = I - \mu \underline{x}_k \underline{x}_k^H$$

and

$$\underline{b}_k = \mu(-\underline{x}_k \underline{\alpha}_k^H \underline{w}_k + \underline{x}_k \zeta_k^* + \underline{\alpha}_k e_k^*) + \underline{\sigma}_k \tag{A-21}$$

Next define $P_k = E[\underline{\rho}_k \underline{\rho}_k^H]$ and $Q_k = E[\underline{b}_k \underline{b}_k^H]$. Then from (A-20) and the assumptions (5)

$$P_{k+1} = E[F_k P_k F_k] + Q_k. \tag{A-22}$$

Using (A-21), we get

$$\begin{aligned} Q_k &= 2\mu^2 \left(\sigma_d^2 \ E[\|\underline{w}_k\|^2] R_x + (1+p)\sigma_d^2 \ R_x + \right. \\ &\quad \left. E[|e_k|^2]\sigma_d^2 I\right) + 2\sigma_c^2 I \end{aligned}$$

In the steady state, this becomes

$$\begin{aligned} Q_k &= 2\mu^2 \left(\sigma_d^2 \ R_x \|\underline{w}^o\|^2 + (1+p)\sigma_d^2 \ R_x + \right. \\ &\quad \left. \xi_{min}\sigma_d^2 I\right) + 2\sigma_c^2 I \end{aligned} \tag{A-23}$$

Dropping the time subscripts and writing R_x as R, the i,jth element of $E[FPF]$ is

$$\begin{aligned} E[FPF]_{ij} &= E\left[\sum_k \sum_l (\delta_{ik} - \mu x_i x_k^*)(\delta_{lj} - \mu x_l x_j^*) P_{kl}\right] \\ &= P_{ij} - \mu(PR)_{ij} - \mu(RP)_{ij} + \\ &\quad \mu^2 \sum_k \sum_l E[x_i x_j^* x_k^* x_l] P_{kl} \end{aligned}$$

Now under the assumption that $\underline{x}_k$ is a circular Gaussian random vector [22],

$$E[x_i x_j^* x_k^* x_l] = R_{ij} R_{lk} + R_{ik} R_{lj}.$$

Therefore,

$$\begin{aligned} E[FPF]_{ij} &= P_{ij} - \mu(PR)_{ij} - \mu(RP)_{ij} + \\ &\quad \mu^2 R_{ij} \mathrm{tr}(RP) + \mu^2 (RPR)_{ij}. \end{aligned} \tag{A-24}$$

While in [12], the last term is neglected, here we take it into account. From (A-22) and (A-24) we obtain the recursion

$$\begin{aligned} P_{k+1} &= P_k - \mu(P_k R + R P_k) + \mu^2 R \ \mathrm{tr}(R P_k) \\ &\quad + \mu^2 R P_k R + Q_k \end{aligned}$$

In the steady state, $P_{k+1} = P_k$ and this becomes

$$\mu(P_k R + R P_k) = \mu^2 R \ \mathrm{tr}(R P_k) + \mu^2 R P_k R + Q_k. \tag{A-25}$$

As a candidate solution, assume that the matrix $P = P_k$ has the same eigenvectors as R. We will verify that this is a valid assumption below. Then $RP = PR$ and

$$P = (2\mu I - \mu^2 R)^{-1}(\mu^2 \mathrm{tr}(RP) I + 2\mu^2 q I + r R^{-1}) \tag{A-26}$$

where $q = \sigma_d^2(\|\underline{w}^o\|^2 + 1 + p)$ and $r = 2\mu^2 \xi_{min} \sigma_d^2 + 2\sigma_c^2$. This gives

$$PR = (2\mu I - \mu^2 R)^{-1}(\mu^2 \mathrm{tr}(RP) R + 2\mu^2 q R + r I).$$

Taking the trace and solving for $\mathrm{tr}(RP)$ we get

$$\mathrm{tr}(RP) \quad = \quad \frac{\mathrm{tr}[(2\mu I - \mu^2 R)^{-1}(2\mu^2 q R + r I)]}{1 - \mathrm{tr}[(2\mu I - \mu^2 R)^{-1}(\mu^2 R)]}. \tag{A-27}$$

Then, if $\{\lambda_i\}_{i=1}^p$ are the eigenvalues of R, we have, from (A-19), and (A-27)

$$\mathrm{tr}(RP) = E[|\underline{\rho}_k^H \underline{x}_k|^2] = \frac{\sum_{i=1}^p \frac{2\mu^2 q \lambda_i + r}{2\mu - \mu^2 \lambda_i}}{1 - \sum_{i=1}^p \frac{\mu^2 \lambda_i}{2\mu - \mu^2 \lambda_i}}. \tag{A-28}$$

It follows from (A-26) that the eigenvalues $\{\delta_k\}_{k=1}^p$ of P are

$$\delta_k = \frac{\mu \mathrm{tr}(RP) + 2\mu q}{2 - \mu \lambda_k} + \frac{r}{\lambda_k(2\mu - \mu^2 \lambda_k)}$$

and therefore

$$\begin{aligned} P \quad &= \quad (\mu \mathrm{tr}(RP) + 2\mu q)(2I - \mu R)^{-1} + \\ &\quad\quad r(2\mu R - \mu^2 R^2)^{-1}. \end{aligned} \tag{A-29}$$

Substitution of (A-28) into (A-29) yields an expression for P which satisfies (A-25). This verifies that the candidate solution is indeed the steady state solution.

Next we specialize to the case $\mu \ll 1$. In this case, (A-28) gives

$$\begin{aligned} \mathrm{tr}(RP) \quad &= \quad \frac{\mu \mathrm{tr}[(2qR + 2\xi_{min}\sigma_d^2 I)(2I - \mu R)^{-1}]}{1 - \mu \mathrm{tr}[R(2I - \mu R)^{-1}]} + \frac{p\sigma_c^2}{\mu} \\ &\approx \quad \frac{p\sigma_c^2}{\mu} \end{aligned} \tag{A-30}$$

and (A-29) along with (A-30) gives

$$\begin{aligned} P \quad &= \quad p\sigma_c^2(2I - \mu R)^{-1} + 2\mu q(2I - \mu R)^{-1} + \\ &\quad\quad 2\mu \xi_{min} \sigma_d^2 (2R - \mu R^2)^{-1} + \\ &\quad\quad 2\sigma_c^2(2\mu R - \mu^2 R^2)^{-1} \\ &\approx \quad p\sigma_c^2(2I - \mu R)^{-1} + \frac{\sigma_c^2}{\mu} R^{-1}. \end{aligned} \tag{A-31}$$

Now, using (A-17), (A-18), and (A-30) we get

$$\xi_q = \frac{1}{a^2}\left[\frac{p\,\sigma_c^2}{\mu} + 2\sigma_d^2\,(\|\underline{w}^o\|^2 + p)\right] \tag{A-32}$$

from which, by using (A-12), we obtain (10).

Appendix D
Derivation of (18) and (21)

Using equations (10) and (17) we can write the excess MSE due to quantization as

$$\xi_q = \alpha_c 2^{-2(1-\rho)B_T} + \alpha_d 2^{-2\rho B_T}. \tag{A-33}$$

Differentiating this equation with respect to ρ we get

$$\frac{d\xi_q}{d\rho} \quad = \quad 2\ln 2\; B_T\;\left(\alpha_c 2^{-2(1-\rho)B_T} - \alpha_d 2^{-2\rho B_T}\right). \tag{A-34}$$

Differentiating again we get

$$\frac{d^2\xi_q}{d\rho^2} \quad = \quad (2\ln 2\; B_T)^2\;\xi_q > 0. \tag{A-35}$$

Equation (A-35) shows that for $\rho \in [0,1]$, $\frac{d^2\xi_q}{d\rho^2} > 0$ and therefore, ξ_q is a convex function of ρ. Now, setting $\frac{d\xi_q}{d\rho}$ in (A-34) equal to zero at $\rho = \rho^*$, we have

$$\frac{\alpha_d}{\alpha_c} = 2^{4\rho^* B_T - 2B_T}. \tag{A-36}$$

Equation (A-36) leads directly to (18).

To derive (21) we first define the following constants

$$A = P_T - 8p\eta_a, \qquad B = 8p(\eta_a + \eta_t), \qquad C = 2p\eta_a. \tag{A-37}$$

Then from (20) we have

$$B_T = \frac{A}{B\rho + C} \tag{A-38}$$

Differentiating (A-33) with respect to ρ gives

$$\begin{aligned}\frac{d\xi_q}{d\rho} \quad = \quad & \ln 2\;\alpha_c\;2^{-2(1-\rho)B_T}\cdot\frac{d}{d\rho}[-2(1-\rho)B_T] \quad + \\ & \ln 2\;\alpha_d\;2^{-2\rho\;B_T}\cdot\frac{d}{d\rho}[-2\rho\;B_T]\end{aligned} \tag{A-39}$$

Differentiating again and using (A-38), we have

$$\begin{aligned}\frac{d^2\xi_q}{d\rho^2} \quad = \quad & \ln 2\;\alpha_c\;2^{-2(1-\rho)B_T}\left[-\frac{4AB(B+C)}{(B\rho+C)^3} + \right. \\ & \left. \ln 2\cdot\frac{4A^2(B+C)^2}{(B\rho+C)^4}\right] + \\ & \ln 2\;\alpha_d\;2^{-2\rho B_T}\left[\frac{4ABC}{(B\rho+C)^3} + \right. \\ & \left. \ln 2\cdot\frac{4A^2C^2}{(B\rho+C)^4}\right].\end{aligned} \tag{A-40}$$

Now, ξ_q is convex if $\frac{d^2\xi_q}{d\rho^2}$ is positive. From (A-40), it is clear that $\frac{d^2\xi_q}{d\rho^2}$ will be positive if the following condition is met:

$$\ln 2 \cdot \frac{4A^2(B+C)^2}{(B\rho+C)^4} > \frac{4AB(B+C)}{(B\rho+C)^3}.$$

Using (A-37) and (A-38), this condition will be satisfied if and only if the following condition is satisfied:

$$B_T > \frac{B}{\ln 2(B+C)} = \frac{1}{\ln 2}\left[1 - \frac{\eta_a + \eta_t}{2\eta_a + 3\eta_t}\right] \tag{A-41}$$

The term on the right hand side of (A-41) is clearly less than $1/\ln 2$. This means that $\frac{d^2\xi_q}{d\rho^2}$ will be positive if $B_T > 1/\ln 2$. This condition is clearly true under the assumption $B_T \geq 2$. Therefore, ξ_q is once again a convex function of ρ. To solve for ρ^{**}, we set $\frac{d\xi_q}{d\rho}$ equal to zero. Using (A-39) we have

$$\alpha_c\ 2^{-2(1-\rho^{**})\cdot\frac{A}{B\rho^{**}+C}} \cdot \frac{2A(B+C)}{(B\rho^{**}+C)^2} \quad = \quad \alpha_d\ 2^{-2\rho^{**}\cdot\frac{A}{B\rho^{**}+C}} \cdot \frac{2AC}{(B\rho^{**}+C)^2}.$$

This implies

$$\rho^{**} = \frac{2A + C\log_2\left(\frac{\alpha_d}{\alpha_c}\cdot\frac{C}{B+C}\right)}{4A - B\log_2\left(\frac{\alpha_d}{\alpha_c}\cdot\frac{C}{B+C}\right)} \tag{A-42}$$

Equations (A-37) and (A-42) lead to (21).

References

[1] J. McChesney, *Notes for MURI Seminar on Low Energy Electronics Design for Mobile Platforms, The University of Michigan*, Oct. 1997.

[2] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *IEEE Proceedings*, vol. 77, pp. 1879–1895, Dec. 1989.

[3] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *IEEE Proceedings*, vol. 83, pp. 498–523, April 1995.

[4] A. Peled and B. Liu, *Digital Signal Processing: Theory, Design, and Implementation*, Wiley, New York, 1976.

[5] B. Widrow and S. D. Stearns, *Adaptive signal processing*, Prentice Hall, Englewood Cliffs NJ, 1985.

[6] J. R. Treichler, C. R. Johnson, and M. G. Larimore, *Theory and Design of Adaptive Filters*, Wiley, New York, 1987.

[7] B. Widrow and *et al*, "Adaptive antenna systems," *IEEE Proceedings*, vol. 55, pp. 2143–2159, 1967.

[8] W. F. Gabriel, "Adaptive arrays – an introduction," *IEEE Proceedings*, vol. 64, pp. 239–272, Feb. 1976.

[9] S. Douglas and T.-Y. Meng, "Optimum error quantization for lms adaptation," in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, May 1991.

[10] C. J. Nicol, P. Larsson, K. Azadet, and J. H. O'Neill, "A low-power 128-tap digital adaptive equalizer for broadband modems," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1777–1789, Nov. 1997.

[11] S. T. Alexander, "Transient weight misadjustment properties for the finite precision LMS algorithm," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-35, no. 9, pp. 1250–1258, Sep. 1987.

[12] C. Caraiscos and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-32, no. 1, pp. 34–41, Feb. 1984.

[13] J. M. Cioffi and M. Ho, "A finite precision analysis of the block-gradient adaptive data-driven echo canceller," *IEEE Trans. on Communications*, vol. 40, no. 5, pp. 940–946, May 1992.

[14] N. J. Bershad and J. C. M. Bermudez, "New insights on the transient and steady-state behavior of the quantized LMS algorithm," *IEEE Transactions on Signal Processing*, vol. SP-44, no. 10, pp. 2623–2625, Oct. 1996.

[15] J. C. M. Bermudez and N. J. Bershad, "A nonlinear analytical model for the quantized LMS algorithm – the arbitrary step size case," *IEEE Transactions on Signal Processing*, vol. SP-44, no. 5, pp. 1175–1183, May 1996.

[16] N. J. Bershad and J. C. M. Bermudez, "A nonlinear analytical model for the quantized LMS algorithm – the power-of-two step size case," *IEEE Transactions on Signal Processing*, vol. SP-44, no. 11, pp. 2895–2900, Nov. 1996.

[17] E. Eweda, N. R. Yousef, and S. H. El-Ramly, "Effect of finite wordlength on the performance of an adaptive filter equipped with the signed regressor algorithm," in *Procc IEEE Global Telecommunications Conf.*, volume 2, pp. 1325–1329, London, UK, 1996.

[18] D. Singh, J. M. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal, and T. J. Mozden, "Power conscious CAD tools and methodologies: a perspective," *IEEE Proceedings*, vol. 83, pp. 5701–594, April 1995.

[19] B. Widrow, J. McCool, and M. Ball, "Complex LMS algorithm," *IEEE Proceedings*, vol. 63, pp. 719–720, 1975.

[20] R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the design of gradient algorithms for digitally implemented adaptive filters," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 125–136, Mar. 1973.

[21] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[22] P. Stoica and R. Moses, *Introduction to Spectral Analysis*, Prentice Hall, Upper Saddle River, NJ, 1997.

[23] T. A. Brubaker and J. C. Becker, "Multiplication using logarithms implemented with read only memory," *IEEE Trans. on Computers*, vol. C-24, pp. 761–765, 1975.

[24] R. H. Katz, *Contemporary Logic Design*, Benjamin-Cummings, 1994.

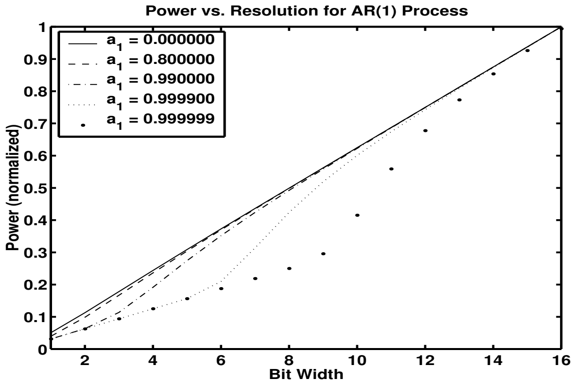

Fig. 1. *Power versus bit width B as function of AR parameter a_1 for loading a B-bit register with successive samples of an AR(1) process. Curves are normalized relative to power P_{16} consumed for a white sequence in a 16 bit register. In the plot the bit width is reduced from right to left by eliminating LSB's*

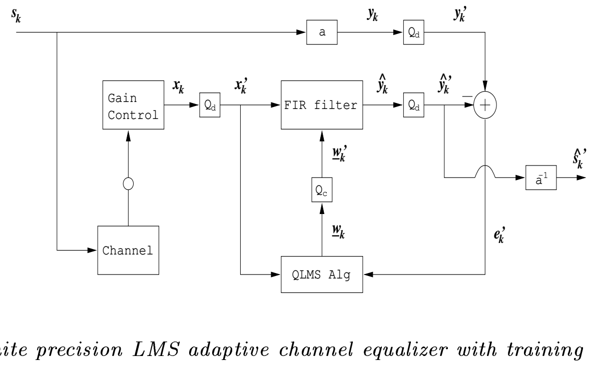

Fig. 2. *Finite precision LMS adaptive channel equalizer with training sequence s_k.*

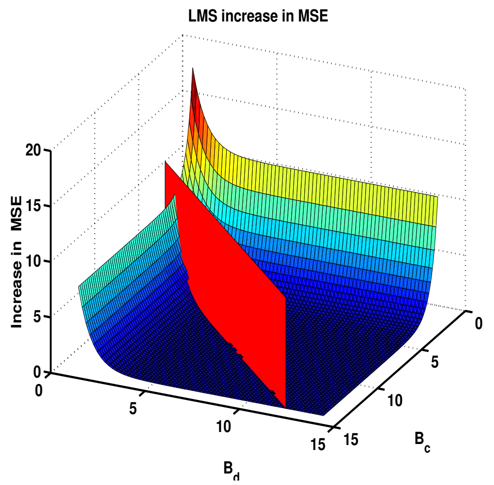

Fig. 3. *Excess MSE due to finite precision as a function of B_d and B_c for single pole IIR channel.*

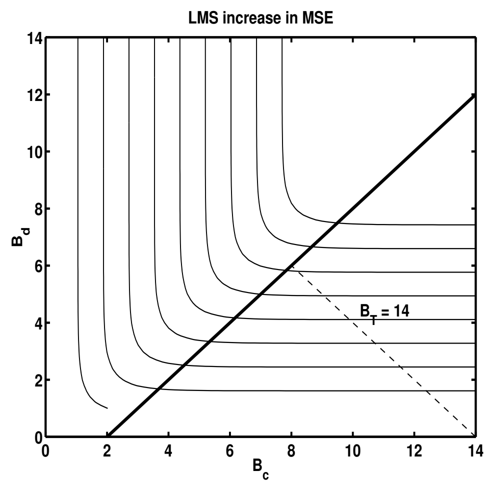

Fig. 4. *Excess MSE due to finite precision as a function of B_d and B_c for single pole IIR channel.*

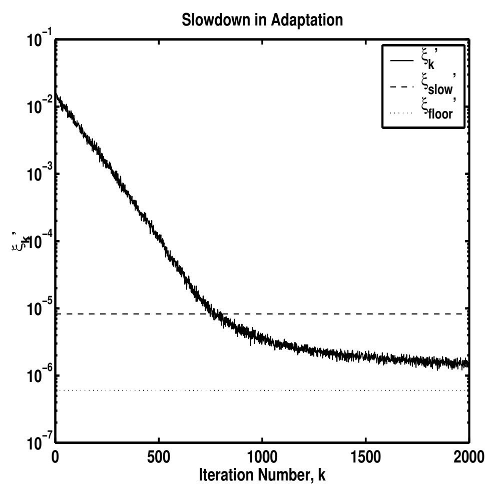

Fig. 5. *Example of slowdown.*

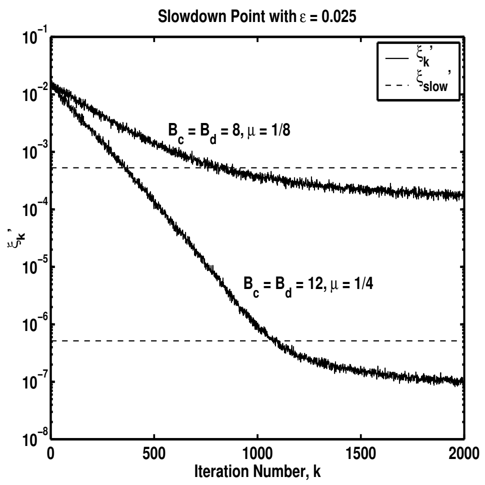

Fig. 6. *Slowdown point.*

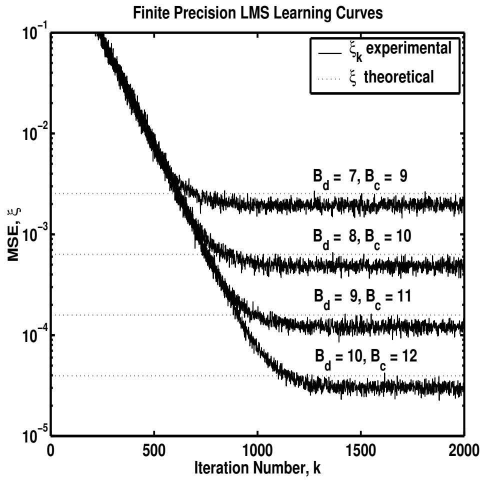

Fig. 7. *Learning curves of LMS filters equalizing IIR channel with Gaussian input without slowdown.*

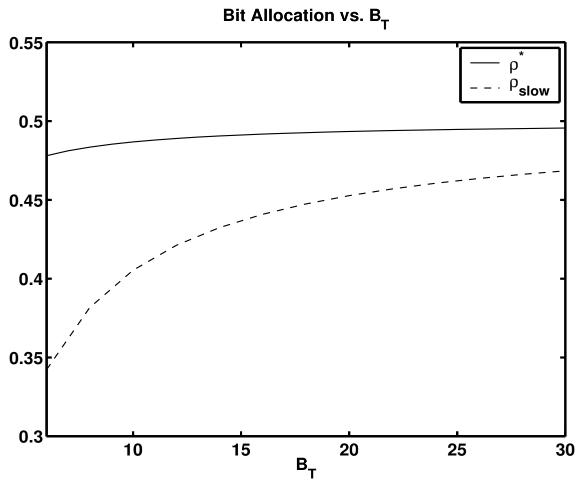

Fig. 8. *IIR channel with Gaussian input. Data bit allocation factors under B_T constraint as functions of B_T.*

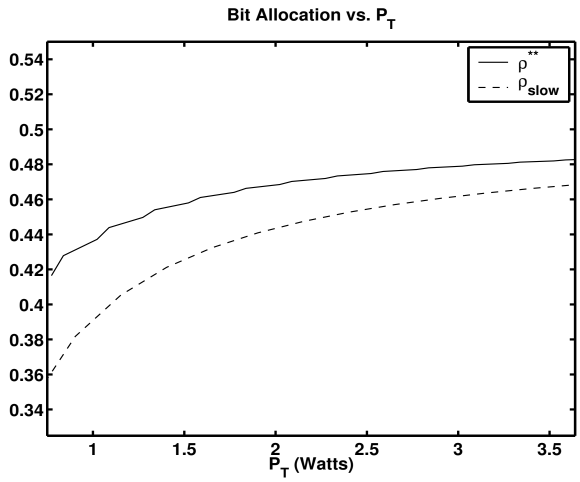

Fig. 9. *IIR channel with Gaussian input. Data bit allocation factors under P_T constraint as functions of P_T.*

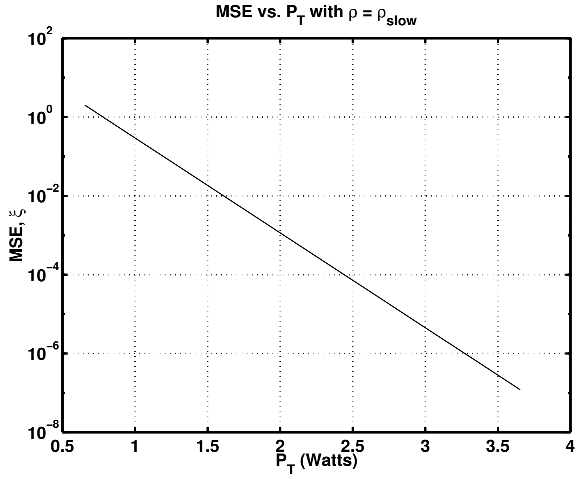

Fig. 10. *IIR channel with Gaussian input. MSE as function of P_T for $\rho = \rho_{slow}$.*

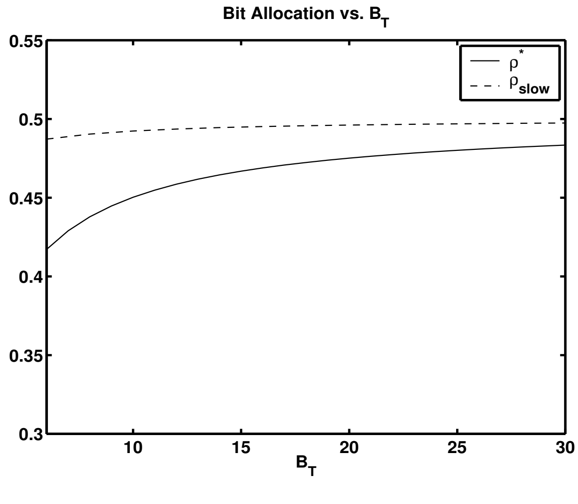

Fig. 11. *128-tap LMS channel equalizer. Data bit allocation factors under B_T constraint as functions of B_T.*

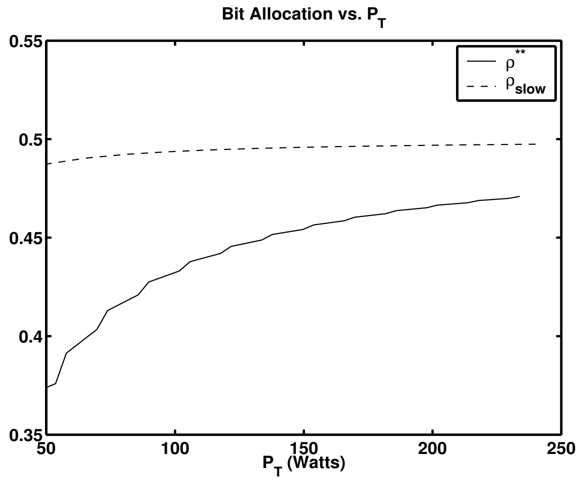

Fig. 12. *128-tap LMS channel equalizer. Data bit allocation factors under P_T constraint as functions of P_T.*

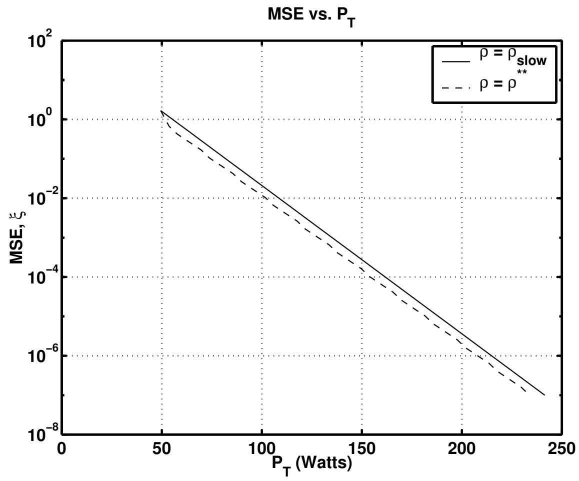

Fig. 13. *128-tap LMS channel equalizer. MSE as function of P_T for $\rho = \rho_{slow}$.*

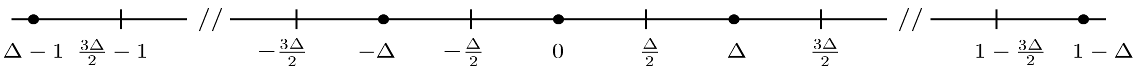

Fig. 14. *Quantizer model.*

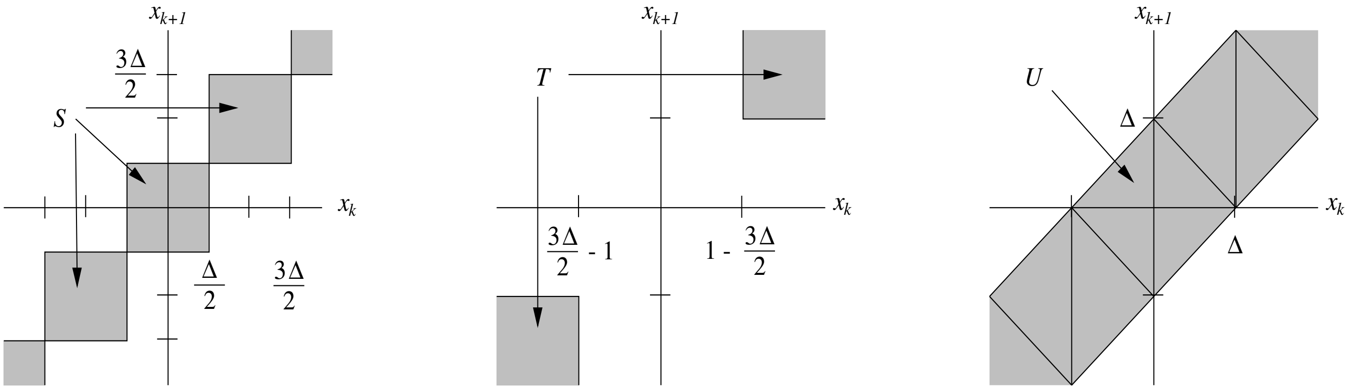

Fig. 15. *Regions defined in (A-3).*

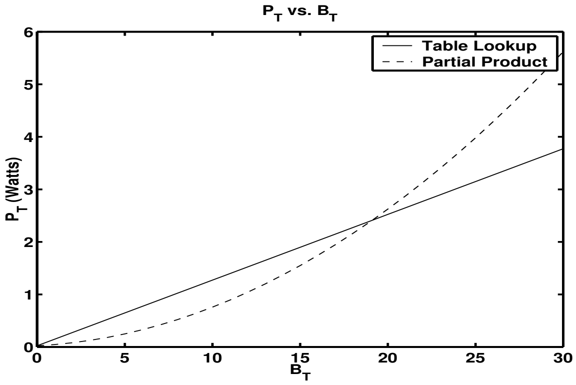

Fig. 16. *P_T vs. B_T for different power relations with $\eta_a \approx 1.4$ mW and $\eta_t \approx 6.8$ mW.*