S-38.220
Licentiate Course on Signal Processing in Communications,
FALL - 97

Fixed-point Implementation of Multiuser Algorithms

Mika Jäsberg

Nokia Telecommunications
Radio Access Systems/New Radio Systems

Email:mika.jasberg@ntc.nokia.com

Date: 20.11.1997

Abstract

The purpose of this paper is to present quantization issues in implementing a fixed-point multistage receiver in CDMA cellular system. Also some finite wordlength effects are studied in the case of minimum mean square error (MMSE) receiver performance in both fading and non-fading conditions. The performance comparison between MMSE receiver and the digital matched receiver (DMF), for a given number of bits per sample, is also evaluated. Current receiver analysis has neglected the effect of quantization, which can degrade system performance and decrease overall system capacity.

1.0 Introduction

A Code Division Multiple Access (CDMA) cellular system has been proposed for use as a next generation cellular system. The goal is to improve system capacity while achieving additional benefits of digital system such as improved privacy and error correction. One of the drawbacks of CDMA is that traditional receivers are susceptible to the near-far problem, which occurs when one received signal is much stronger than another. Much research work has been done to develop receiver structures that are near-far resistant. Different approaches can be divided in two classes: single user receivers and multiuser receivers.

In receivers where only single user is demodulated (mobile station in downlink in cellular system), techniques based on adaptive filtering can be used to combat interference and near-far problem. In multiuser receiver, all of the desired users are demodulated at the same location (base station in uplink direction in cellular system). A multiuser receiver uses all of the information about the received signal from all users to combat interference and near-far problem.

It has been developed different optimum multiuser receivers which outperforms the conventional single-user detectors as far as near-far problem is considered. The drawback is that optimum multiuser receiver is hopelessly complex, precluding the possibility of a physical implementation. For that reason so called sub-optimal detectors, that would retain much of the performance improvement possible with optimum receivers, but at a greatly reduced complexity, have been widely investigated.

Two classes of sub-optimal multiuser receivers that have been recently investigated are decorrelators and interference cancellers. The decorrelators are based on multiplying the vector of received decision variables by the inverse of the matrix of code sequence cross-correlations, thus removing the effect of interfering users. The interference cancellers can cancel interference either successively or in parallel using multiple stages. Also some hybrid approaches exist.

The main approach in this paper is to present the results of quantization and fixed-point implementation in multiuser algorithms. A DSP implementation of a practical receiver will require a DSP chip with a fewer number of bits than the computer chips used in simulation of a receiver performance. Therefore, DSP implementation performs poorer than the simulation results predict. Also a fixed-point implementation is favored over floating-point implementation, due to high processing requirements necessitates by the high chip rate. This further degrades performance because of the limited dynamic range available with fixed-point arithmetic. This paper is organized in a following way: In chapter 2 analysis of multiuser receiver is investigated: MUI (Multiple Access Interference) for different stages is calculated. In chapter 3 different quantization effects are presented. In chapter 4 effects of quantization in multiuser algorithms are presented and in chapter 5 some performance issues are presented as far as MMSE and DMF are considered. Finally, in chapter 6 some conclusions are presented.

2. Multistage receiver

2.1 Introduction

It has been developed two main classes of suboptimal receivers, linear receivers, largely based on decorrelators, and nonlinear receivers which are based on interference cancellation. Now we concentrate on multistage interference canceller (IC).

The idea in IC is to subtract the interference from received signal, thereby leaving the residual signal as essentially a single user signal in the presence of channel noise. If all of the MAI interference can be perfectly canceled, the performance of the receiver will be identical to that of a conventional receiver in a single user system and the receiver will have a near-far resistance of one. In practice, the interference cannot be canceled perfectly and the efficiency will be less than one.

In multistage receiver estimates are made simultaneously for all interferers, and then subtracted from the received signal .This is repeated in multiple stages to gain better estimates of the interference. A block diagram of a multistage receiver, which has two stages is shown in figure 1.

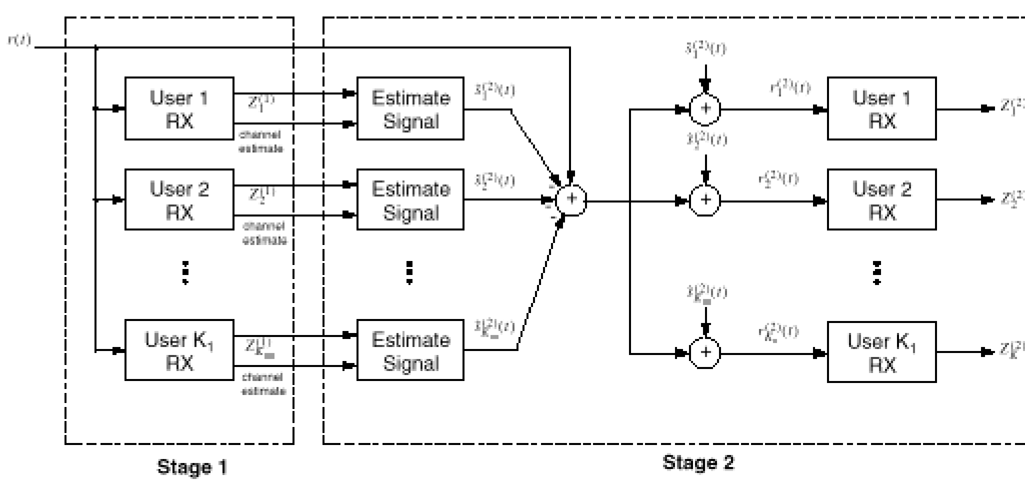


Fig. 1. Two-stage multistage receiver

The first stage is the traditional multiuser receiver, which is a bank of single-user receivers. The only difference is that a bit decision is not made at this point. At the beginning of the second stage, an estimate is made of each user's received signal. These estimates are then subtracted from the received signal. The desired user's signal is then added to this residual signal and the signal is again passed to single-user receiver. The complexity per demodulated symbol is shown to be linear with respect to the total number of users K while maintaining a performance comparable to the optimum multiuser receiver for most practical cases.

Any CDMA system will have a mix of synchronized users and unsynchronized users. Both user types will act as a multiple access interference (MAI) to the desired user. For one stage receivers (i.e. no interference cancellation) there will be no difference in the effect caused by unsynchronized and synchronized users. However, for stages 2 and beyond, interference cancellation can be performed for the synchronized users, but not for the unsynchronized users. The unsynchronized interference will affect the BER in two ways. First, the uncancelled interference will appear directly at the output of the last stage of the receiver, and will likely dominate if the power is significantly greater than the residual power of the synchronized users after cancellation. The unsyncronized interference will also appear at the each decision stage in the receiver, degrading the estimation of the synchronized interference. This will increase the noise in the cancellation process and increase the BER in the output.

2.2 Receiver algorithms

There are K_1 synchronized users and K_2 unsyncronized users in the system, with a total of $K = K_1 + K_2$ users. Any arbitrary user k's received signal is represented by

$$s_k(t) = \sqrt{2P_k}\, b_k(t-\tau_k) a_k(t-\tau_k) \cos\{\omega_c(t-\tau_k)+\varphi_k\} \tag{2.1}$$

where k is the number of the user, P_k is the power of the signal, $b_k(t-\tau_k)$ is the data signal, $a_k(t-\tau_k)$ represents the spreading signal, and $\cos\{\omega_k(t-\tau_k)+\varphi_k\}$ represents the modulating waveform.

The data signal $b_k(t)$ is given by

$$b_k(t) = \sum_{i=-\infty}^{\infty} b_{k,i}\, p_T(t-iT) \tag{2.2}$$

where $b_{k,i} \in \{+1,1\}$ is an infinite sequence of data bits and $p_T(t)$ is a rectangular pulse with unity amplitude and duration T. The spreading code $a_k(t)$ is given by

$$a_k(t) = \sum_{i=-\infty}^{\infty} a_{k,i}\, p_{T_c}(t-iT_c) \tag{2.3}$$

where the chip values are given by $a_{k,i} \in \{+1,1\}$ and T_c is the chip duration.

When a single propagation path is present, the received signal at the base station is given by

$$r(t) = n(t) + \sum_{k=1}^{K} s_k(t - \tau_k) \tag{2.4}$$

where n(t) is additive white Gaussian noise with two sided power spectral density $\frac{N_0}{2}$.

The first stage in a receiver is a bank of correlation receivers. Each of these receivers recovers the data bit by correlating the received signal with the spreading signal of user k, to form a decision statistic

$$Z_{k,i}^{(1)} = \int_{iT+\tau_k}^{(i+1)T+\tau_k} r(t) a_k(t - \tau_k) \cos(\omega_c t + \varphi_k) dt \tag{2.5}$$

The next step, for each stage s+1, is to make an estimate $\hat{s}_k(t)$ of each synchronized user k's received signal $s_k(t)$, which is as follows

$$\hat{s}_k^{(s+1)}(t) = \frac{2}{T} a_k(t - \tau_k) \cos(\omega_c + \varphi_k) \sum_{i=-\infty}^{\infty} Z_{k,i}^{(s)} p_T(t - iT) \tag{2.6}$$

The interference cancellation is performed by subtracting all estimates from the received signal, and then adding back the desired user before detection by the next receiver stage. New received signal at each stage s for each user j, $1 \le j \le K_1$ is given by

$$r_j^{(s)}(t) = n(t) + s_j(t) + \sum_{k=1, k \ne j}^{K_1} \left[s_k(t - \tau_k) - \hat{s}_k^{(s)}(t - \tau_k) \right] + \sum_{k=K_1+1}^{K_2} s_k(t - \tau_k) \tag{2.7}$$

The first term is the Gaussian noise, the second term is the desired user's signal, the third term is the residual after the interference estimates have been subtracted out, and the final term is the MAI due to the unsynchronized users.

The received signal for each user j given by (2.7) is then passed to the next bank of correlation receivers .A new decision statistic is formed at stage s during bit i by correlating (2.7) with user j's spreading code. This new decision statistic is then passed to next stage until final stage of the receiver has been reached. At the last stage, an estimate is made of the desired user d's transmitted bit. The estimate of data bit i from user d is determined as follows:

$$\hat{b}_{d,i} = \left\{ 1 \Leftrightarrow Z_{d,i}^{(s)} \ge 0, -1 \Leftrightarrow Z_{d,i}^{(s)} \le 0 \right. \tag{2.8}$$

The next step is to develop BER based on the decision statistic. The improved Gaussian approximation uses the mean and variance of the variance of MAI to determine BER.

A probability of error can be shown to be

$$P_{b,d}^{(s)} = \frac{2}{3} Q\left(\sqrt{\frac{P_d T^2}{2\left(\mu_\Psi^s + \frac{N_0 T}{4}\right)}}\right) + \frac{1}{6} Q\left(\sqrt{\frac{P_d T^2}{2\left(\mu_\Psi^s + \sqrt{3}\sigma_\Psi^s + \frac{N_0 T}{4}\right)}}\right) + \frac{1}{6} Q\left(\sqrt{\frac{P_d T^2}{2\left(\mu_\Psi - \sqrt{3}\sigma_\psi^s + \frac{N_0 T^2}{4}\right)}}\right) \quad (2.9)$$

where μ_Ψ is the mean of the conditional variance of the total MAI and σ_Ψ is the variance of the conditional variance of the total MAI. Exact formulas can be found in (Cam).

Now we should define and interference MAI for different stages of the receiver. The interference characteristics of the MAI due to synchronized and unsynchronized users will be the same at the first stage, since there is no interference cancellation at this point. Since interference cancellation is not performed on unsyncronized interference at later stages, the unsynchronized interference at later stages will be the same as stage 1. The interference caused by user k to user j during bit interval i is

$$I_{k,j}^{(1)} = \sqrt{\frac{P_k}{2}} \cos\left(\varphi_k - \varphi_j\right)\left\{\int_{iT+\tau_j}^{iT+\tau_k} b_{k,i-1} a_k\left(t-\tau_k\right) a_j\left(t-\tau_j\right)dt + \int_{iT+\tau_k}^{(i+1)T+\varsigma_j} b_{k,i} a_k\left(t-\tau_k\right) a_j\left(t-\tau_j\right)dt\right\} \quad (2.10)$$

To compute the total multiple access interference, we need to know the mean and variance of the power of the interference. Since there is no interference cancellation at this point, these are simply the mean and the variance of each individual power P_k . After the calculation of μ_Ψ and σ_Ψ BER is calculated via (2.9).

The MAI for stages 2 or higher is different compared to stage 1 due to fact that the MAI of synchronized users will change due to interference cancellation. It can be written the interference for a synchronized user j caused by a synchronized user k at stages 2 and higher as follows

$$I_{k,j}^{(s+1)} = I_{k,j}^{(1)} + \hat{I}_{k,j}^{(s+1)} \quad (2.17)$$

where the second term represents the MAI caused by the estimates $\hat{s}_k^{(s+1)}(t)$. In other words, the MAI at stage s+1 is given by the original MAI minus the estimate of the original MAI. This interference is given by

$$I_{k,j}^{(s+1)} = \cos\left(\varphi_k - \varphi_j\right)\left\{\left[\sqrt{\frac{P_k}{2}} b_{k,i-1} - \frac{Z_{k,i-1}^{(s)}}{T}\right]\int_{iT+\tau_j}^{iT+\tau_k} a_k\left(t-\tau_k\right) a_j\left(t-\tau_j\right)dt + \left[\sqrt{\frac{P_k}{2}} b_{k,i} - \frac{Z_{k,i}^{(s)}}{T}\right]\int_{iT+\tau_k}^{(i+1)T+\tau_j} a_k\left(t-\tau_k\right) a_j\left(t-\tau_j\right)dt\right\} \quad (2.18)$$

3.0 Quantization effects

When performing a theoretical analysis, numbers can be represented as real or complex numbers with infinite precision. In a DSP chip, a number must be represented with some finite numerical precision. The process of converting a continuos amplitude signal into discrete amplitude signal is known as quantization. In the process of quantizing any given number, there are two main effects that determine the precision with which that number can be discretely represented: the number of bits used for quantization and whether fixed- or floating-point representation is used .There are additional techniques that can be used to improve the accuracy or quantized signal, including companding (compression and expansion) and non-uniform quantization. These techniques are useful if the probability distribution of the signal is known and it is non-uniform.

The use of quantized signals can cause additional errors in DSP system. In a mathematical operation such as addition or multiplication, it is possible for the result to be larger than either of the inputs. Since the result must be quantized to the number of bits used in the rest of the system, the result must be either truncated or rounded to the nearest level. Also, if the magnitude of the result of a mathematical operation becomes too large, overflow (or underflow) will result. This can cause serious errors, and scaling is sometimes used to ensure that this does not occur.

The operation of the uniform quantizer is straightforward and deterministic. The analysis of this device is anyway complex, due to the nonlinearity of the quantization process. There have been two different basis in this area: a deterministic examination of the output based on a deterministic input and approximating the quantizer with a uniform noise source based on a stochastic description of the input signal. It have to remember, that when the noise model is dependent on the input signal, then studying the effects of quantization in a given receiver depends on the set of inputs.

3.1 Fixed-point and floating-point implementation

The main characteristic of digital arithmetic is the limited (usually fixed) number of digits to represent numbers. This constraint leads to finite numerical precision in computations, which leads to roundoff errors and nonlinear effects in the performance of the system under consideration. The main errors in digital system are as follows:

- ADC quantization noise, which results from representing the samples of the input data, x(n), by only a small number of bits
- overflow errors, which result from additions or accumulation of partial results in a limited register length
- product round-off errors, caused when the output, y(n), and results of internal arithmetic operations are rounded (or truncated) to the permissible wordlength
- coefficient quantization errors, caused by representing the sub-system coefficients by a finite number of bits

The representation of numbers in a fixed-point format is a generalization of the familiar digital representation of a number as a string of digits with a decimal point. In this notation, the digits to the left of the decimal point represent the integer part of the number and, and the digits to the right of the decimal point represent the fractional part of the number. Thus a real number can be represented as

$$X = \left(b_{-A}, b_{-A+1}, \ldots, b_{-1}, b_0, b_1, \ldots, b_{B-1}, b_B\right) = \sum_{i=-A}^{B} b_i r^{-i} \quad (3.1)$$

where $0 \le b_i \le (r-1)$ and b_i represents the digit, r is the radix of the base, A is the number of integer digits and B is the number of fractional digits. The most important representation in digital processing systems is the binary representation. In this case r=2 and the digits $\{b_i\}$ are called binary digits or bits and take the values {0,1}. The binary digit b_{-A} is called the most significant bit (MSB) of the number, and the binary digit b_B is called the least significant bit (LSB).

By using an n-bit integer format (A=n-1, B=0), we can represent unsigned integers with magnitude in the range 0 to $2^n - 1$. If we use the fraction format (A=0, B=n-1), with a binary point between b_0 and b_1, that permits numbers in the range from 0 to $1 - 2^{-n}$. Any integer or mixed number can be represented in a fraction format by factoring out the term r^A in (3.1).

Here our attention is on the binary fraction format because mixed numbers are difficult to multiply and the number of bits representing an integer cannot be reduced by truncation or rounding. There are three different ways to represent negative numbers. On the other hand, the format for positive numbers is the same for all of the representations, namely

$$X = 0.b_1 b_2 \cdots b_B = \sum_{i=1}^{B} b_i \cdot 2^{-i}, X \ge 0 \quad (3.2)$$

The MSB b_0 is set to zero to represent the positive sign. Now a negative fraction

$$X = -0.b_1 b_2 \cdots b_B = -\sum_{i=1}^{B} b_i \cdot 2^{-i} \quad (3.3)$$

can be represented using one of the three following formats.

In sign-magnitude format MSB is set to 1 to represent the negative sign

$$X_{SM} = 1.b_1 b_2 \cdots b_B \qquad for X \le 0 \quad (3.4)$$

In one's-complement format the negative numbers are represented as

$$X_{1C} = 1.\bar{b}_1 \bar{b}_2 \cdots \bar{b}_B \qquad X \le 0 \quad (3.5)$$

where $\overline{b_i} = 1 - b_i$ is the complement of b_i. Thus if X is a positive number, the corresponding negative number is determined by complementing (changing 1's to 0's and 0's to 1's) all the bits.

The two's-complement format is formed by complementing the positive number and adding one LSB. Thus

$$X_{2C} = 1.\overline{b}_1\overline{b}_2\cdots\overline{b}_B + 00\cdots01 \qquad X < 0 \tag{3.6}$$

where + represents modulo-2 addition that ignores any carry generated from the sign bit. For example, the number -3/8 is obtained by complementing 0011 (3/8) to obtain 1100 and then adding 0001. This yields 1101, which represents -3/8 in two's complement format. Most fixed-point digital signal processors use two's complement arithmetic. A benefit of using two's complement notation is that, if two numbers addition do not cause overflow, then the result is correct even if an intermediate sum cause overflow. This can be illustrated by an example. Assume that 4+4=8 is added to -2-2=-4 to get 8-4=4. In two's complement notation we add 0100+0100=1000 (overflow) to 1110+1110=1100 to get 1000+1100=0100, which is indeed correct.

The basic arithmetic operations of addition and multiplication depend on the format used. For one's complement and two's complement formats, addition is carried out by adding the numbers bit by bit. The formats differ only how a carry bit affects the MSB. In two's complement format, if the carry is present in MSB, it is dropped. On the other hand, in one's complement format it is carried around to LSB. Addition in sign-magnitude format is more complex and involves sign checks, complementing and generation of carry. On the other hand, direct multiplication of two sign-magnitude numbers is relatively straightforward, whereas a special algorithm is usually employed for one's complement and two's complement multiplication.

A fixed point representation of numbers allows us to cover a range of numbers $x_{\max} - x_{\min}$ with a resolution

$$\nabla = \frac{x_{\max} - x_{\min}}{m - 1} \tag{3.7}$$

where $m = 2^b$ is the number of levels and b is the number of bits. A basic characteristic of fixed point representation is that the resolution is fixed. Also ∇ increases in direct proportion to an increase in the dynamic range. It is also possible that very large numbers cannot be represented in the set range, and that very small numbers will be quantized to zero. A floating point representation may be employed as a means for covering a large dynamic range. The binary floating-point representation commonly used in practice consists of mantissa M, which is the fractional part of the number and falls in the range $0.5 \le M < 1$, multiplied by the exponential factor 2^E, where exponent E is either a positive or negative integer. Hence a number can be represented as

$$X = M \cdot 2^{E} \tag{3.8}$$

The mantissa requires a sign bit for representing positive and negative numbers, and the exponent requires an additional sign bit. Since the mantissa is a signed fraction, we may use any of the methods described above. For example, the number 3/8 is represented by the following mantissa and exponent

$$M = 0.110000$$
$$E = 101$$

where the leftmost bit in the exponent represents the sign bit. If the two numbers are multiplied, the mantissas are multiplied and the exponents are added. For example to multiply numbers 5 and 3/8 happens as follows

$$X_1 X_2 = M_1 M_2 \cdot 2^{E_1 - E_2} = (0.011110) \cdot 2^{010} = (0.0111100) \cdot 2^{001}$$

If two floating point numbers must be added, exponent should be equal. This can be make sure by shifting the mantissa of the smaller number to the right and compensating by increasing the corresponding exponent. It should be observed that the shifting operation required to equalize results in general in loss of precision. Also overflow occurs in the multiplication of the two floating-point numbers when the sum of the exponent exceeds the dynamic range of the fixed point representation of the exponent.

In comparing a fixed-point representation with a floating-point representation ,with a same number of total bits, it is apparent that the floating-point representation allows to cover a larger dynamic range by varying the resolution across range. The distance between two successive numbers increases as the numbers increase in size. This variable resolution results in a larger dynamic range. If we wish to cover the same dynamic with both fixed-point and floating-point representation, the floating-point provides finer resolution for smaller numbers but coarser resolution for the larger numbers. In contrast, the fixed-point representation provides a uniform resolution throughout the range of numbers. While floating-point representation is desirable due to its numerical accuracy, the hardware tends to be slower and more expensive compared to fixed-point representation. In table 1 is a comparison of fixed-point fractional, fixed-point integer and floating point implementations.

Feature	**Fixed-point fractional**	**Fixed-point integer**	**Floating point**

overflow;addition	yes	yes	unlikely
overflow; multiplication	no	yes	unlikely
roundoff error;addition	no	no	yes
roundoff error; multiplication	yes	no	yes
dynamic range	modest	modest	high
hardware implementation	simple	simple	difficult,slow

Table 1. Comparison of binary implementations

When can overflow occur? When adding operands with different signs, overflow cannot occur. On the other hand, when subtracting operands with the same sign, overflow cannot occur. In multiplication the number of digits of product is considerably higher than the number of bits in the multiplicand or multiplier. Hence multiply must cope with overflow, since we normally want a B-bit product as the result of multiplying two B-bit numbers. Usually the product is either truncated or rounded. As a result we have a truncation or round-off error in the B least significant bits. In table 2 are different cases when overflow might happen.

Operation	Operand A	Operand B	Result
A+B	≥ 0	≥ 0	< 0
A+B	< 0	< 0	≥ 0
A-B	≥ 0	< 0	< 0
A-B	< 0	≥ 0	≥ 0

Table 2. Overflow conditions

3.2 Errors resulting from rounding and truncation

The most basic quantizer is shown in Fig. 2, where x is the input, x_e is the quantized output with 3 bits. The quantization error is given by

$$e = q(x) = x_e - x \qquad (3.9)$$

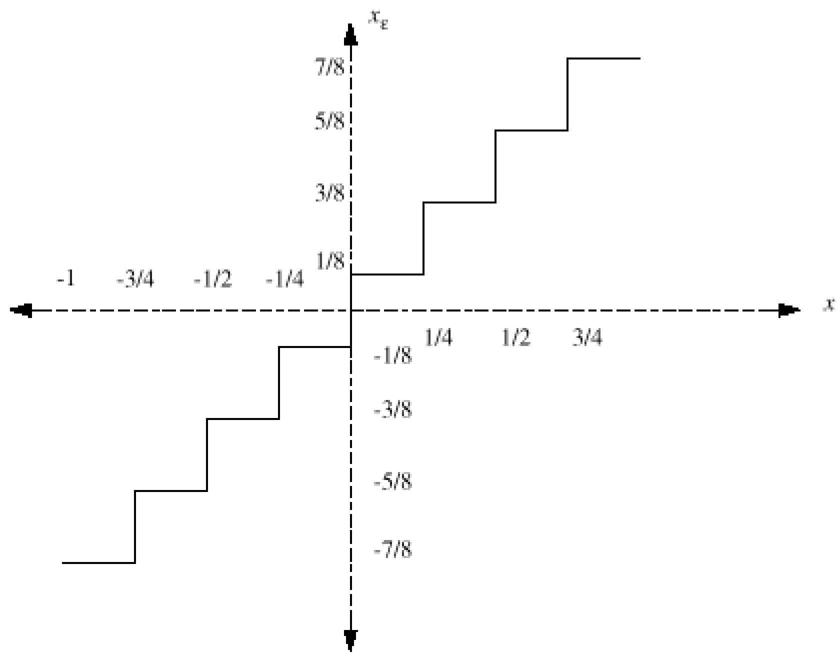


Figure 2. Uniform quantizer

The effect of rounding or truncation is to introduce an error whose value depends on the number of bits in the original number to the bits after quantization. For example, if a 12-bit input is multiplied by a 16-bit coefficient the result is 28-bit long and will need to be quantized back to 12 bits (for example). This quantization leads to errors whose effects are similar to those of the ADC noise, but could be more severe. In rounding, a number is rounded off to the nearest quantization level by choosing the higher-order bits closest to the unrounded result. This is achieved by adding half an LSB to the result. In truncation any portion of signal that requires greater than B bits is simply truncated, i.e. the most significant higher-order bits are retained and the lower order bits are discarded. The range of error is $-\Delta/2 \le e \le \Delta/2$ for rounding and $-\Delta \le e \le 0$ for truncation in two's complement system.

What is the difference between roundoff error and the quantization error is that the input signal x is discrete in amplitude, not continuos (it has already been quantized).

Analyzing quantization error exactly is a difficult due to nonlinear operation of quantízer. Modeling quantization using linear noise sources provide a reasonable approximation, i.e. additive noise is introduced to the unquantized value. The linear model has been shown to provide accurate predictions of statistical averages when the input is a widely varying signal. We may write

$$q(x) = x + \varepsilon \tag{3.10}$$

This model is illustrated in Fig. 3.

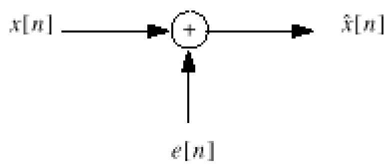


Fig. 3. Linear Quantization Noise Model

Modeling quantization with a linear noise source relies on several key assumptions:

1. The noise has a uniform distribution of amplitudes over one quantization period
2. The noise source is a wide-sense stationary white-noise process
3. There is no correlation between the noise source and the input to the quantizer, all other quantization noise sources, and the system input.

Since the noise amplitude is modeled as uniform over the allowable range, the mean and variance for rounding are

$$\begin{aligned} \mu_e &= 0 \\ \sigma_e^2 &= \frac{\Delta^2}{12} \end{aligned} \tag{3.11}$$

and for truncation

$$\begin{aligned} \mu_e &= -\frac{\Delta}{12} \\ \sigma_e^2 &= \frac{\Delta^2}{12} \end{aligned} \tag{3.12}$$

The nonlinearity of the quantization process can lead to zero-input limit cycles, which are oscillations at the output even when there is no input to the system.

It has been investigated widely quantizer performances , and some SNR versus bit comparisons can be presented in table 3.

Number of bits N	Number of bins r	Signal-to-Noise Ratio D (dB)
3	8	21
4	16	27
5	32	33
6	64	39
3	128	45
3	256	51

Table 3. SNR versus numb. of quantization bits

R is the ratio of the total quantizer range to the bin width (resolution). It can be seen that SNR increases by about 6 dB with the addition of each bit.

3.3 Coefficient quantization errors

The primary effect of quantizing the filter coefficients into the finite number of bits is to alter the positions of the poles of H(z) in the z-plane. The fewer the number of bits used to represent the coefficients the more will be the deviation in the pole and zero positions. As well as potential instability, deviations in the locations of the poles and zeros also lead to deviations in the frequency response. It is good to remember, that if filter under consideration (for example, in MMSE receiver) is FIR, the performance may not be degraded in a sense than in IIR case. Moreover, the effects of finite wordlength on performance are more difficult to analyze in IIR filters than in FIR filters because of their feedback arrangements. Also in the stopband of filter, for example, it limits the maximum attenuation possible, thus allowing additional signal transmission. The performance degradation in FIR case can be shown in figure 4.

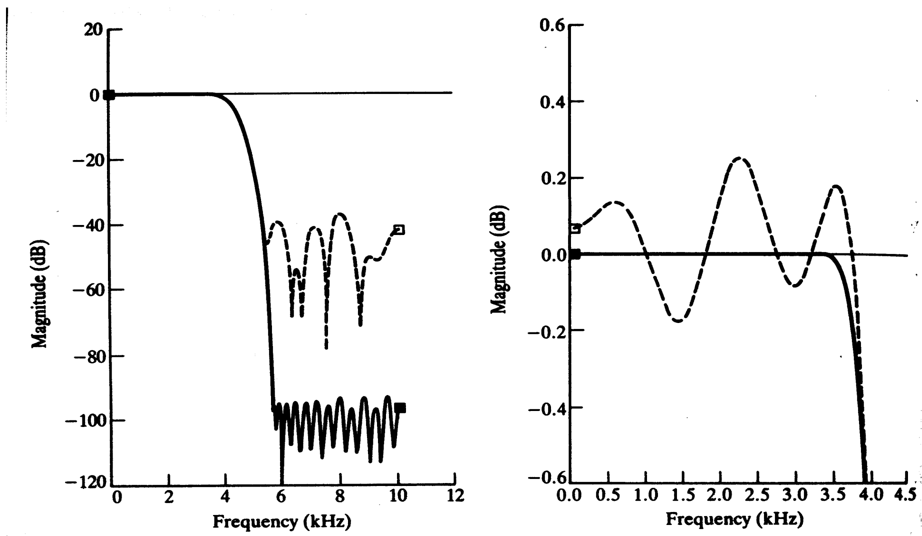


Fig. 4 (a) Effects of coefficients quantization (b) Passband; dashed line:quantized solid line: unquantized

The quantized and unquantized coefficients, $h_q(n)$ and h(n) are related as

$$h_q(n) = h(n) + e(n), \qquad n = 0,1,\cdots,N-1 \tag{3.13}$$

where e(n) is the error between quantized and unquantized coefficients. In frequency domain, Eq. (3.13) can be written as

$$H_q(\omega) = H(\omega) + E(\omega) \tag{3.14}$$

Physically e(n) can be viewed as the impulse response of the desired filter with another filter.

For frequency selective filters (lowpass, bandpass, bandstop filters), several researchers have developed bounds on the errors in the frequency response. These bounds could serve as useful guides in determining a suitable coefficient wordlength for a given filter. The bounds are useful in estimating coefficient wordlength requirements for adaptive FIR filters (e.g. MMSE) as the exact characteristics of these filters are not known a priori. For direct form FIR structure, assuming rounding ,the following are the most widely used bounds

$$|E(\omega)| = N2^{-B}$$

$$|E(\omega)| = 2^{-B}(N/3)^{1/2} \quad (3.15\text{-}3.17)$$

$$|E(\omega)| = 2^{-B}\left[(N\log_e N)/3\right]^{1/2}$$

where B is the number of bits used to represent each coefficient and N is the filter length. Bound (3.15) is an absolute bound, derived under worst-case assumptions and so it is overly pessimistic. Bounds (3.16) and (3.17) are statistical bounds and could give a more accurate estimate of the errors in the frequency response and coefficient wordlengths to use.

3.4 Scaling

To avoid the large errors overflow can create in a fixed-point system, scaling can be used to reduce or eliminate the possibility of overflow. Scaling implies that,before quantization, a signal is scaled by some factor s_L that reduces the signal energy and thus reduces the possibility of overflow. A block diagram can be shown in Fig. 6. The influence of scaling is that the output SNR is reduced by s_L^2. This occurs because the scaling occurs on the input before quantization, and thus before the quantization noise is added. Thus the signal is reduced while the noise is not.

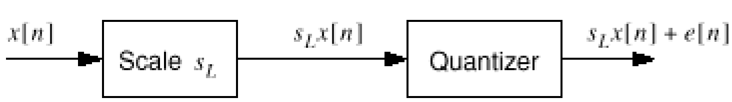


Fig. 6. Scaling model

There are three common approaches to this problem: sum (L1) scaling, L2 scaling, or L_∞ scaling, which all require the impulse response h(n) of system under consideration . The first technique is the most conservative and guarantees no overflow, the second is the least conservative, and the third is moderately conservative. For complex systems such as the digital CDMA receiver where a single-input single-output impulse response cannot be derived, an exact scaling technique cannot be developed. The alternative is to set the quantization range so that overflow is minimized based on a typical signal level.

In L_1 scaling the following restriction holds

$$x_{\max} < \frac{1}{\sum_{m=-\infty}^{\infty} \left| h_k(m) \right|} \qquad (3.18)$$

where $h_k(m)$ is a impulse response between input signal x(n) and a system node k. In this scaling it is ensured that overflow cannot occur at node k. Then we get the scale factor as follows

$$s_L = \frac{1}{\max_k \left[\sum_{m=-\infty}^{\infty} \left| h_k(m) \right| \right]} \qquad (3.19)$$

The drawback of this approach is that SNR may be degraded beyond the acceptable limits.
In L_2 scaling it is ensured that the energy in some node k is less than or equal to the energy of the input signal x(n). Scaling factor in this case is given by

$$s_L = \frac{1}{\sqrt{\sum_{n=-\infty}^{\infty} \left| h_k(n) \right|^2}} \qquad (3.20)$$

This form degrades SNR the least when we compare these three proposals.

L_∞ scaling only ensures that there will be no overflow if a sine wave is applied to the input. In this case input scaling factor is as follows

$$s_L = \frac{1}{\max_k \left| H_k\left(e^{j\omega}\right) \right|} \qquad (3.21)$$

4.0 Quantization errors in a multistage receiver

Now it should be investigated how analytical results diverge from the model where fixed-point quantization is used. It can be found different sources of error in multistage receiver. The received signal at the input of the receiver, r(t), must be quantized, so there will be an associated quantization error. Overflow is also possible, so care must be taken in choosing scale factor so that overflow does not occur at the input. From this stage

forward, every number must be quantized with B+1 bits since the rest of the receiver is digital.
The received signal is then passed to the correlation receiver, where several additions and multiplications take place, so each operation must be analyzed to determine where errors will mostly likely be introduced. If we ignore the effects of the coherent demodulation, the first multiplication takes place during the discrete correlation process, which is given by

$$Z_{k,i}^{(s)} = \sum_{i=1}^{NN_s} r_{k,i}^{(s)} a_{k,i} p_{T_s}\left(t - iT_s\right) \tag{4.1}$$

where $r_{k,i}^{(s)}$ is a sampled version of $r_k^{(s)}(t)$ at sample i and $a_{k,i}$ is the chip value of user k's PN sequence during sample i. Since $a_{k,i} \in \{1,-1\}$, the multiplication in the correlation process only causes a possible sign change of the received signal sample, and thus no overflow is possible and no truncation or rounding is required. In other words, no noise introduced at this point. When estimated signals are summed, then there is a possibility of overflow, although the result is not likely to exceed significantly the threshold value, so an appropriate scaling term can be chosen. And we also have to remember, that overflow is not a concern in intermediate additions if two's complement notation is used. The correlation value is used both in determining the bit value and the estimate of the received signal level. Then, if overflow does occur, clipping is the best approach to ensure that the result remains as close to the original value as possible.

Quantization error can occur also in estimation phase, as we can see from (4.1). Data bit estimate $\hat{b}_{k,i}$ is multiplied with a locally generated PN sequence. Because $\hat{b}_{k,i} \in \{-1,1\}$ the result will be a possible sign change of the chip values. There is no possible overflow and truncation or rounding is not necessary, no noise is introduced at this point. The next step is to multiply by the magnitude of decision statistic. Since the chip values are set {-1,1}, the result can be only a possible sign change and no noise is introduced. The estimate of power level during bit interval i in two-stage receiver is

$$\hat{P}_k^{(2)} = \frac{\left|Z_{k,i}^{(2)}\right|}{NN_s} \tag{4.2}$$

The denominator $\frac{1}{NN_s}$ is a scaling factor, which must be quantized and therefore there will also be quantization noise. The multiplication will not result in overflow since this is a constant that is less than one. Truncation or rounding will be necessary so noise is introduced here also. Finally I and Q estimates are formed by multiplying this result by the cosine and sin terms. Since each of these terms is less than one, overflow is not a concern, but truncation or rounding is necessary.

When these I and Q signals are formed, all signal estimates are summed together. Overflow is theoretically possible at this point, although summation should have a magnitude near of the original received signal, and if the initial original signal did not overflow, the reconstructed estimate should not as well. Summation will require truncation or rounding. The next step is to subtract the reconstructed estimates from the

received signal. Overflow (or underflow) is not likely to be a problem since two terms have similar magnitudes. Truncation or rounding is necessary since this residual signal should be close to zero. The next step is to add back the estimate of the desired user's signal, which have already been truncated or rounded when the estimates were summed. This signal is then passed to next correlation receiver.
It seems that overflow is not a problem as long as the initial received signal values can be quantized without overflow. There are a number points where noise due to truncation or rounding is introduced. Scaling may help to minimize the noise at certain points, since the approximate range of values of the resulting summation may be known.

4.1 Quantized receiver algorithms

Now we should investigate how receiver algorithms differ from the results presented in chapter 2.2.
The received signal is quantized at the input of the digital receiver with B bits, resulting in the quantized signal

$$r_\varepsilon(t) = n_c(t) + e_r(t) + \sum_{k=1}^{K} s_k(t-\tau_k) \tag{4.3}$$

where the first term represents the AWGN channel noise and the second term represents the quantization noise. The decision statistics in the first stage are made based on this quantized received signal. For a user k during bit interval i, this decision statistic is given by

$$Z_{k,i}^{(1)} = \int_{iT+\tau_k}^{(i+1)T+\tau_k} r_\varepsilon(t) a_k(t-\tau_k)\cos(\omega_c t + \varphi_k)dt \tag{4.4}$$

The next step is to form interference estimates as in chapter 2.2, but now the amplitude estimates $2Z_{k,i}^{(s)} / T$ will be first quantized before they can be used in digital receiver. The estimate of user k's received signal is given by

$$\hat{s}_k^{(s+1)}(t) = a_k(t-\tau_k)\cos(\omega_c t + \varphi_k)\sum_{i=-\infty}^{\infty}\left\{\frac{2}{T} Z_{k,i}^{(s)} + e_k^{(s)}(t)\right\} p_T(t-iT) \tag{4.5}$$

where $e_k^{(s)}(t)$ is the quantization error for user k's estimate at stage s. Next step is to form the new received signal at each stage s for each user j, $1 \le j \le K$, by subtracting out the interference estimates

$$r_{j,e}^{(s)}(t) = n_c(t) + e_r(t) + s_j(t) + \sum_{k=1,k\neq j}^{K}\left[s_k(t-\tau_k) - \hat{s}_k^{(s)}(t-\tau_k)\right] \tag{4.6}$$

The first term is the Gaussian channel noise, the second term is the quantization noise due to quantizating the received signal, the third term is the desired user's signal, and the final term is the residual MAI.

This received signal is then passed to the next bank of correlation receivers. At each stage , a new decision statistic $Z_{j,i}^{(s)}$ is obtained during bit interval i by correlating (4.6) with user j's spreading code

$$Z_{j,i}^{(s)} = \int_{iT+\tau_j}^{(i+1)T+\tau_j} r_{j,\varepsilon}^{(s)}(t) a_j\left(t-\tau_j\right)\cos\left(\omega_c t+\varphi_j\right)dt \qquad (4.7)$$

The decision statistic is then passed onto the next stage for use in an interference estimation until the final stage of the receiver has been reached. At that point the bit estimate i from user d is made as in (2.8).

Now, as in chapter 2.2, MAI is modeled with the help of improved Gaussian approximation (Cam). Now the difference is that the ability to cancel interference is degraded by the effects of quantization. Now the decision statistic can be written

$$Z_{j,i}^{(s)} = \eta_c + \varepsilon_r + \sqrt{\frac{P_j}{2}} T b_{j,i} + \sum_{k=1,k\neq j}^{K} I_{k,j}^{(s)} \qquad (4.8)$$

where the first term is a zero mean Gaussian random variable representing the correlated noise, the second term is the correlated noise related to quantizing the signal r(t), the third term represents the desired component, and the final term is the residual MAI after interference cancellation. This last term also includes the quantization noise due to the quantization of the interference estimates.

While the e_r, which is a noise due to quantization of received signal, is uniform over $[-\Delta/2, \Delta/2]$, the distribution of correlated version ε_r is unknown. With the help of simulations it can be shown that this can be approximated quite well using zero-mean Gaussian random variable, whose standard deviation can be obtained from a short simulation run.

As in chapter 2.2, the total MAI is analyzed with the help of (2.10). There is also no change how the μ_Ψ and σ_Ψ are defined, and these are given in (2.11) and (2.12). What is then different, is that the form of the improved Gaussian approximation for the probability of error has been changed due to addition of the error source ε_r in the decision statistic (4.8), since $Z_{k,i}^{(s)}$ is not necessarily a Gaussian random variable.

If we model ε_r as a Gaussian random variable than the probability of error simplifies to a form similar to (2.13), when quantization is not used. The only difference is that in denominator appears a term σ_{ε_r} that is the standard deviation of the correlated quantization noise source ε_r.

Since there is no interference cancellation in the first stage of the multiuser receiver, there is no interference estimation and thus no quantization of the interference estimate. For this reason, there is no difference compared to the characterization of the first stage MAI in 2.2. This is given in (2.16).

The residual interference at stage s from user k to to user j is given

$$I_{k,j}^{(s+1)} = I_{k,j}^{(1)} - \hat{I}_{k,j}^{(s+1)} \qquad (4.9)$$

but now , compared to (2.17) the interference estimate $\hat{I}_{k,j}^{(s+1)}$ contains the quantization noise caused by quantizing the estimate $2Z_{k,i}^{(s)}/T$. Residual interference is as follows

$$I_{k,j}^{(s+1)} = \cos(\varphi_k - \varphi_j)\left\{\left[\sqrt{\frac{P_k}{2}}b_{k,i-1} - \left(\frac{Z_{k,i-1}^{(s)}}{T} + \frac{\varepsilon_s}{2}\right)\right]\int_{iT+\tau_j}^{iT+\tau_k} a_k(t-\tau_k)a_j(t-\tau_j)dt + \left[\sqrt{\frac{P_k}{2}}b_{k,i} - \left(\frac{Z_{k,i}^{(s)}}{T} + \frac{\varepsilon_s}{2}\right)\right]\int_{iT+\tau_k}^{(i+1)T+\tau_j} a_k(t-\tau_k)a_j(t-\tau_j)dt\right\} \quad (4.10)$$

where ε_s is the quantization noise e_s when correlated with the PN sequence of user j.

In order to calculate BER with the help of Gaussian approximation, mean and variance of the effective power must be determined.

In reference (Cam) has been build a model where these different quantization effects have been taken under consideration. In Fig. 8-11 BER is plotted against E_b/N_0 for quantization bit levels 12, 8, 6 and 4 bits. Several observations can be made from these BER curves by comparing the accuracy of the analytical model with the simulation model.
First, in Fig. 7 where 12 bits was used. It can be seen that analytic and simulation models are in close agreement for stage 1 for the entire range E_b/N_0. The analytic results employing interference cancellation get optimistic (i.e. lower BER than in simulation) as E_b/N_0 increases. Next, if we consider the case of 8 bits, the result of stage 1 is very accurate across the range of E_b/N_0. For stages 2 and 3 , the analysis again become optimistic as E_b/N_0 increases. This same behaviour can been seen also in the case of 6 quantization bits, although now stage 1 performance is no more same, there is some difference. The reason for the increased difference in behavior in the case of stage 2 and stage 3 is that the quantization error can now be an appreciable portion of the estimate. Finally we examine the case of 4 quantization bits (Fig. 10). For the stages 2 and 3 the analysis is severely optimistic. The results show that attempting to cancel interference with such a low degree of numerical precision degrades the performance so much that stages 2 and 3 perform worse than stage 1.

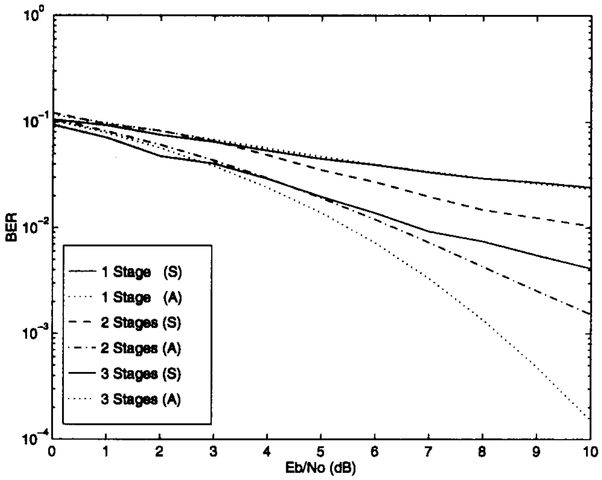


Fig. 7. BER vs. E_b/N_0 for 12 bits

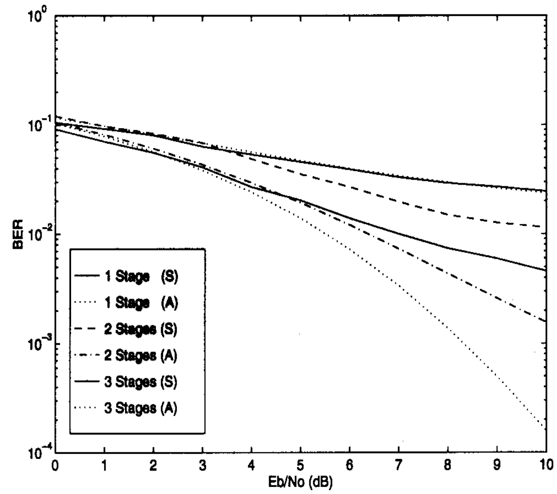


Fig. 8. BER vs. E_b/N_0 for 8 bit

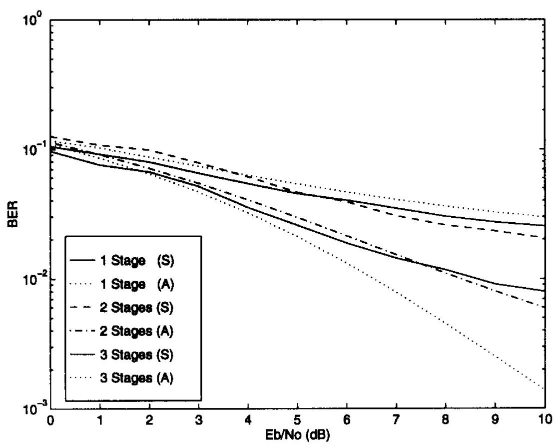


Fig. 9. BER vs. E_b/N_0 for 6 bits

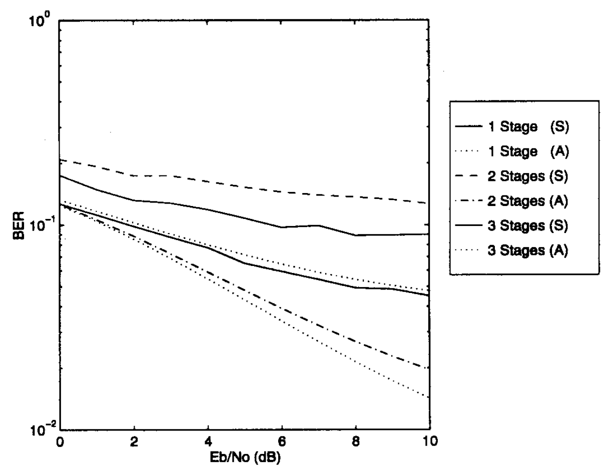


Fig.10. BER vs. E_b/N_0 for 4 bits

In figures 11 and 12 can be seen a comparison of performances of stage 1 and stage 3 with a different levels of quantization. It can be shown that the results of 12 bits are nearly identical to the case when quantization is not used. The reason is of course the fact that enough fine bin spacing is used to accurately quantize each interference estimate, as well as initial received signal. It can also be seen that the performance of the receiver is nearly the same regardless of whether 8 or 12 bits is used. When considering the 6 bit case there is a clear reduction in performance as far as BER is considered. It seems that 8 bits in this case is an appropriate compromise.

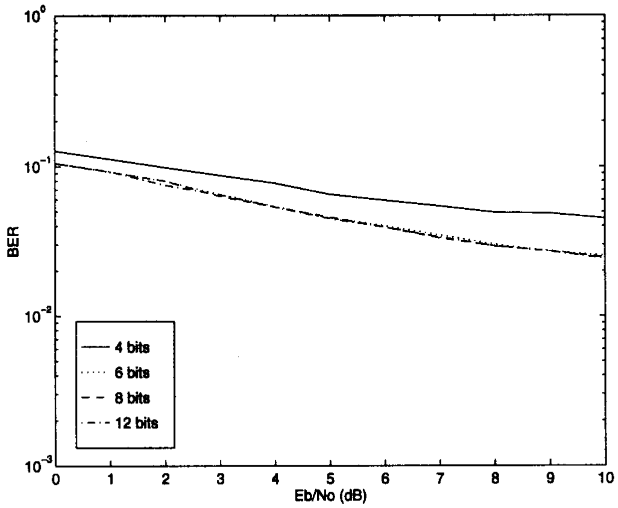


Fig. 11. BER vs. E_b/N_0 for one stage receiver

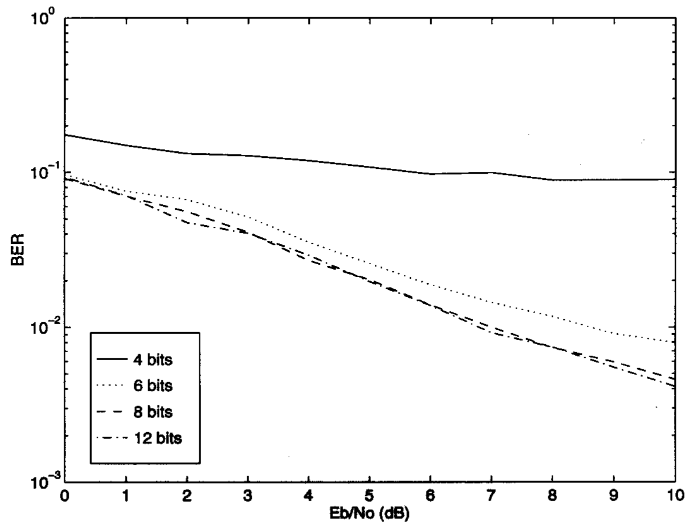


Fig. 12. BER vs. E_b/N_0 for three stage receiver

4.2 Different quantization strategies

There are different quantization strategies that can be employed in an effort to improve the system performance. First, a smaller quantization range can be chosen, an thus, for a same number of bits, the bin width will decrease. The drawback is that overflow may be more likely to occur, and that is the reason the system designer will have to tradeoff the increase in BER due to overflow against the decrease in BER because of the increased resolution to accurately quantize the interference estimates. In Fig. 13 can be shown the simulated BER vs. E_b/N_0 when 6 quantization bits is used, but the quantization range is (-15,15) compared to (-20,20) in preceding simulation results.

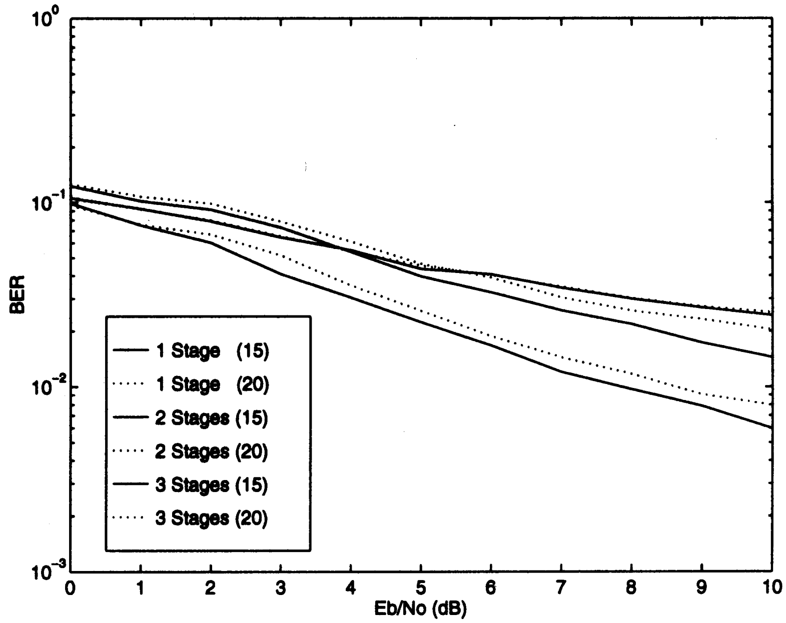


Fig. 13. BER vs. E_b/N_0 for different quantization ranges (15 and 20)

Overflow occurred for every value of E_b/N_0. The net effect is a slight decrease in the BER when interference cancellation (i.e. stages 2 or 3) is employed. In Fig. 14 the BER vs. E_b/N_0 for a 3 stage receiver is shown . Three cases are compared: six bits with a range of (-20, 20), six bits with a range of (-15,15), and eight bits with a range of (-20,20). Using six bits with a range of (-15,15) falls between the other two cases, so a conclusion is that by using this smaller range we can fix much of the degradation caused by going from 8 bits to six bits. This same effect can be shown to be in a 2 stage receiver.

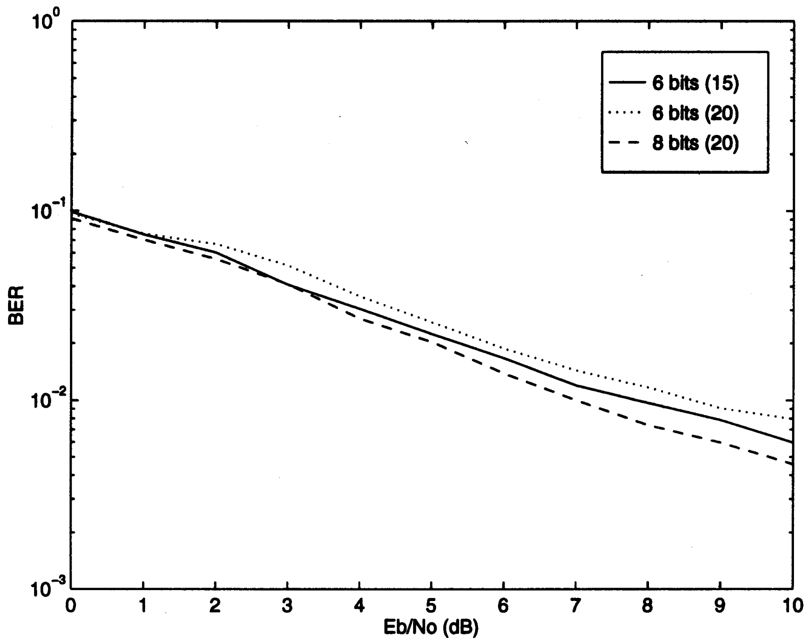


Fig. 14. BER vs. E_b/N_0 for three stage receiver;different quantization ranges

Another alternative would be to use non-uniform quantization when computing the interference estimates. Non-uniform quantization allows the bins to be non-uniformly distributed across the quantization range. Thus if we have a priori knowledge of the signal characteristics, we can design the quantizer to maximize system performance.

For a multistage receiver, there are two key types of signals we have to quantize: the received signal and the interference estimates. The received signal is fairly strong and random, so a uniform quantizer is the most appropriate. For interference estimates a situation is a more problematic. Using a non-uniform quantization, bins are concentrated at the smaller signal levels and thus more accurately model the individual interference estimates. The problem here is that once the estimates are summed together to perform interference cancellation , the resulting sum will have approximately the same dynamic range as the received signal $r(t)$ and therefore should be quantized in a same way (i.e. uniform quantization). The drawback of this modification is also that the non-uniform quantization will have to match the strength of each estimate, and overflow in any intermediate sums may cause overflow in the final result since we are changing quantization method with each sum in receiver. Since the appropriate pattern for the non-uniform bin distribution will depend upon the amplitude of the estimate, the pattern will also have to be dynamic and able to change over time. As a conclusion, the uniform quantization will provide a much more reasonable and practical solution from an implementation standpoint.

One of the method of approach is to use uniform quantizers in interference estimates but with a much smaller range or higher number of bits, thus decreasing the bin width and increasing the overall resolution of a quantizer. As before, however, the difficulty with this technique is that when the interference estimates are summed, the resulting signal has a similar dynamic range compared to the received signal, and will require a similar quantization range and a number of bits. It seems that if a number of users in a system is higher, then these more complex techniques presented may be more beneficial.

5.0 MMSE and DMF performance degradation due to finite wordlength effect

The class of minimum mean square error (MMSE) receivers has been shown to offer various implementation advantages over the optimum receiver for DS-CDMA systems. However, the performance of the MMSE receiver is affected by the quantization of the adaptive filter coefficients and the analog-to-digital converter (ADC) output. Moreover, in order to ensure that the input is within the operating range of the ADG, the received signal has to be properly scaled. This scaling also affects the performance of the receiver, since the power of desired user's signal varies as the power of MAI varies.

The received signal for an asynchronous DS-CDMA system with K users, whose chip and symbol durations are, respectively, T_c and T, can be written as

$$r(t) = \sum_{k=1}^{K} d_k\left(t-\tau_k\right)c_k\left(t-\tau_k\right)\cos\left(\omega_c+\theta_k\right)+n_w\left(t\right) \qquad (5.1)$$

The received signal is scaled so that the values at the input of the ADC lie between 1 and -1, in order to prevent overflow. Assuming that the power of each user is known, the scaling factor a is given by

$$a = K + \Delta \qquad (5.2)$$

where Δ is a noise margin.

On the other hand, even if a perfect power control is used, it is difficult to know the received power at the receiver since it depends on the relative phase of each user. In general, this scaling is done by the automatic gain control (AGC) amplifier. However, AGC normalizes the total received power, so that without information of the number of active users and the maximum power control error, there is no guarantee that overflow will not occur at the ADC.
We can also consider the effect of the scaling parameter on the convergence of the LMS algorithm. The LMS algorithm is convergent in the mean square sense if and only if the step size parameter μ satisfies the condition

$$0 < \mu < \frac{2}{\lambda_{\max}} \qquad (5.3)$$

where $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix of the received signal. For a non-fading channel, the autocorrelation matrix, after the scaling is given by $R = R^{'} / a^2$ where $R^{'}$ is the matrix before scaling. It can also shown that eigenvalue satisfies the following relation

$$\lambda_i = \frac{\lambda_i^{'}}{a^2} \qquad (5.4)$$

Using this 5.3 can be written in a following way

$$0 < \mu < \frac{2a^2}{\lambda_{\max}} \qquad (5.5)$$

From this we can see that the relative value of step size parameter varies with the number of the active users. As a consequence of this, with an optimum μ for same small number of active users, the training sequence has to be longer when the number of users is larger.

5.1 Quantization effects in performance

It is assumed that each sample of the input vector is represented by B_d bits plus a sign bit, and each filter coefficient is represented by B_c bits plus a sign bit. If we assume that the quantization process is to round off to the nearest level whose step is uniform, the quantization error associated with an input vector is given by $\sigma_d^2 = 2^{-2B_d} / 12$. and the error associated with filter coefficients is given by $\sigma_c^2 = 2^{-2B_c} / 12$. If the same wordlength is used for both the input vector and the filter coefficients, and the step size parameter μ is sufficiently small, the total output MSE of the LMS algorithm, including the quantization effects, is given by

$$J = J_{\min} + \frac{1}{2}\mu J_{\min} trR + \frac{N\sigma_c^2}{2\mu} + \left(\left| w_{opt} \right|^2 + c \right) \sigma_d^2 \qquad (5.6)$$

where $J_{\min}$ is the MSE of the optimal (Wiener) filter with $w_{opt} = R^{-1} p$ and c is a constant which depends on the way the inner product of the input vector and the filter coefficients is computed. On the right hand side of (5.6), the two first terms are the MSE of the infinite-precision LMS algorithm. The third term arises because of the quantization of the filter coefficients .The fourth term arises because of the quantization of the input vector and that of the filter output. On the other hand, for the DMF receiver ,the quantization occurs only at the ADC. The input vector of the DMF is given by

$$u_m^{'} = u_m + \eta \qquad (5.7)$$

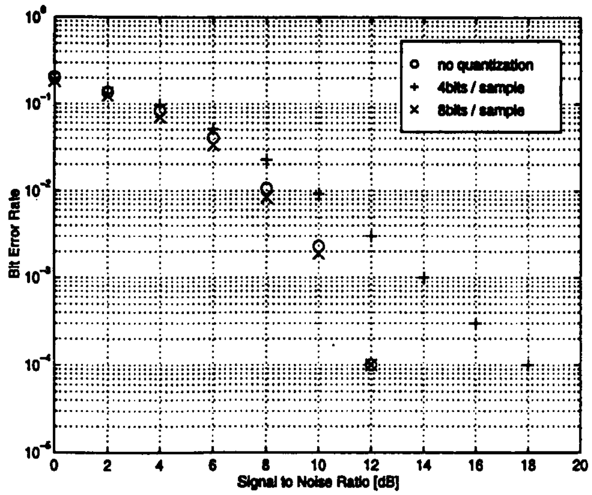
where the last term is a zero-mean Gaussian random sequence whose variance is σ_d^2. The coefficients of the DMF are, after a proper scaling, $w = ac_1/N$, so the variance of the quantization error at the output DMF is given by

$$J_{q,dmf} = \frac{a^2}{N^2} E\left[(c_1\eta)^2\right] = \frac{a^2\sigma_d^2}{N} \quad (5.8)$$

For a fading channel, the coefficients of the DMF are, after proper scaling, $w = a(m)c_1 / N\alpha_1(m)$, where a(m) is a scaling factor and α(m) is a Rayleigh random process. The variance of the quantization error at the output is given by

$$J_{q,dmf} = \frac{1}{N^2} E\left[\frac{a^2(m)}{\alpha_1^2(m)}(c_1\eta)^2\right] = \frac{1}{N^2} E\left[\frac{a^2(m)}{\alpha_1^2(m)}\right]\sigma_d^2 \quad (5.9)$$

In Fig. 15 and Fig. 17 are the simulation results of the MMSE receiver. It can be seen that the MMSE receiver is near-far resistant when the number of bits/sample is sufficiently large; however, there is significant degradation in performance when the number of bits/sample is small ,and this performance degradation becomes larger as the number of users increases. On the other hand, from the performance plots of DMF in Fig. 16 and Fig. 19, it can be seen that while it is not near-far resistant, it is less sensitive to the quantization effect. Moreover, when the number of bits per sample is equal to or less 4, the conventional DMF receiver outperforms the MMSE receiver. For a fading channel, the performance gain of the MMSE receiver over the DMF receiver decreases, which can be seen from the pictures 18 and 20-22.



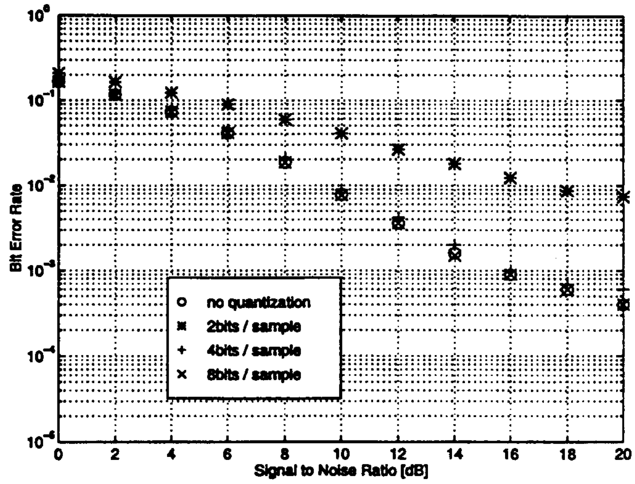
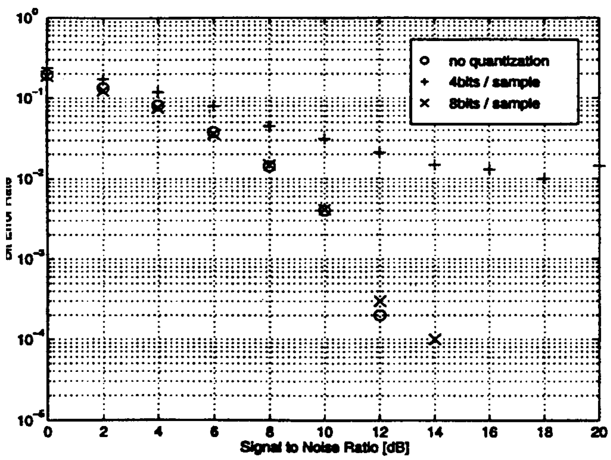


Fig 15. The performance of the MMSE receiver;K=4   Fig. 16. The performance of the DMF receiver; K=8



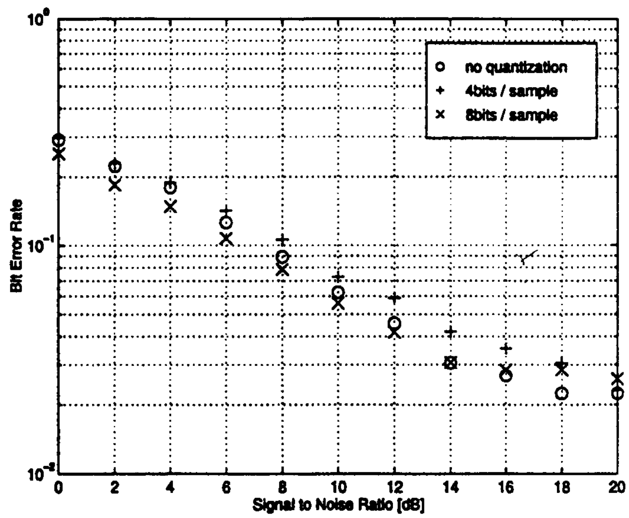


Fig. 17. The performance of the MMSE receiver;K=8   Fig. 18. The performance of the MMSE receiver for a fading channel;K=4

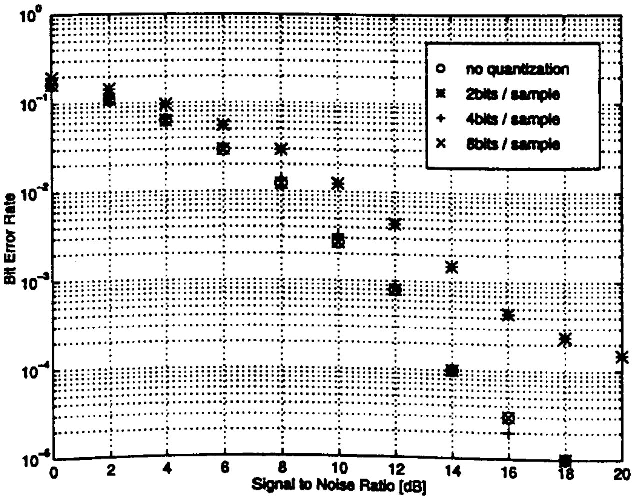


Fig 19. The performance of the DMF receiver;K=4

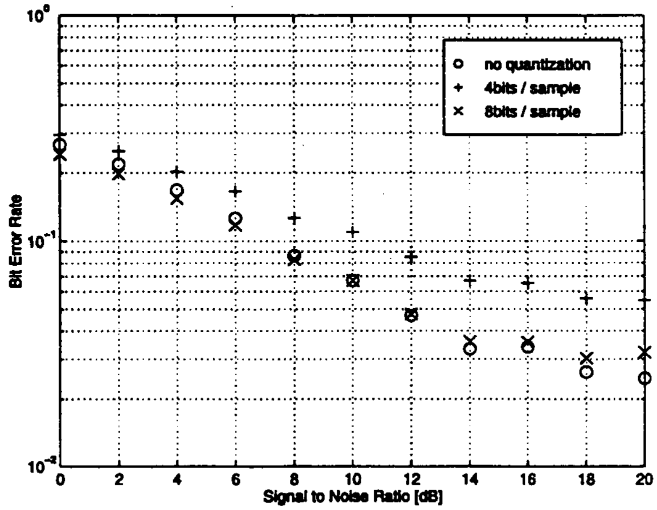


Fig. 20. The performance of the MMSE receiver for a fading channel;K=8

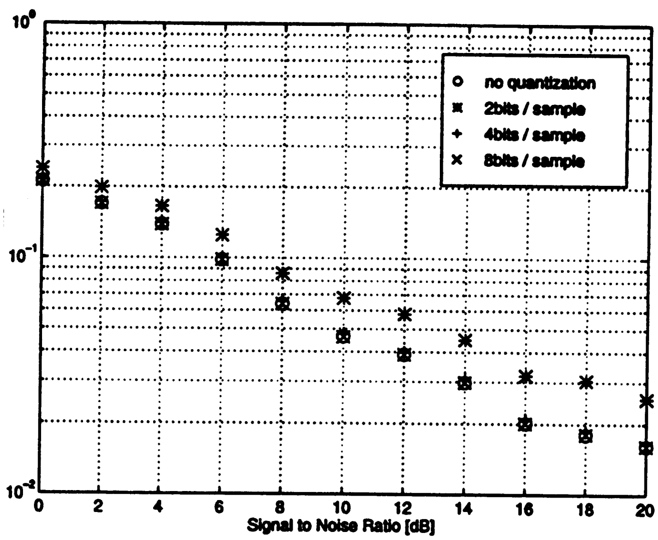


Fig.21. The performance of the DMF receiver for fading channel;K=4

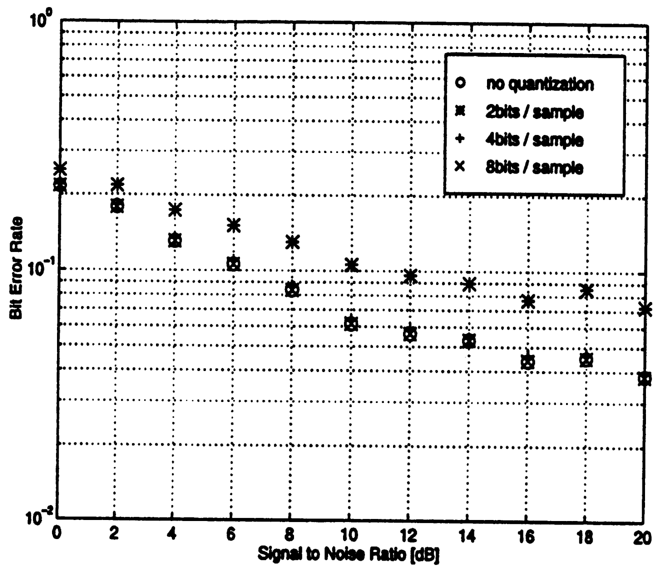


Fig. 22. The performance of the DMF receiver for a fading channel;K=8

6.0 Conclusions

The optimum receiver in the AWGN channel is too complex for practical implementation, and so sub-optimal techniques have been developed that retain much of the optimum receiver's performance but at a linear complexity. The two main classes are linear receivers based on decorrelation, and the non-linear receivers based on interfernce cancellation. Parallel cancellation has been shown through analysis and simulation (Cam) to provide an attractive combination of performance robustness and complexity. The effect of quantization in the multistage receiver has been investigated. The BER curves shows that, as long as the dynamic range of the input signal is constrained, the multistage receiver performs quite well when using fixed-point arithmetic. Almost no degradation is noticed when 8 bits are used and overflow is allowed only rarely at low E_b/N_0 , and only small degradation is noticed when 6 bits are used and overflow is allowed more frequently. Since these bit levels are easily achievable in practice, the multistage receiver can be implemented in hardware using fixed point arithmetic and still retain nearly all of the performance of the receiver when quantization is not used.

Also MMSE receiver and DMF receiver performance comparison has been evaluated. The MMSE receiver is attractive both because it has an implementation advantage over the optimum receiver for a DS-CDMA system, and because it outperforms the conventional DMF receiver. However, it was shown that the effect of the finite precision arithmetic can cause significant degradation to the MMSE receiver, and this degradation becomes larger as the MAI increases. Also, for a small number of bits per sample it was shown that the DMF outperforms the MMSE receiver. Moreover, the scaling causes additional performancee degradation to the MMSE receiver, since the variation of the scaling factor can cause variation in the relative value of the step size parameter of the LMS algorithm. This means that a long training sequence is necessary for the adaptive filter to converge for a large number of users. The same effect causes only a variation of noise and MAI in DMF receiver.

References

(Cam) Rick Cameron: "Fixed-Point Implementation of a Multistage Receiver" (Doctoral Thesis, Jan. 13, 1997)

(Ich) Emmanuel C. Icheafor, Barrie J. Jervis: "Digital Signal Processing: A Practical Approach" Addison-Wesley, 1993

(Pro) John G. Proakis, Dimitris G. Manolakis: "Introduction to digital signal processing" Macmillan Publishing Company, New York,1989

(Mer) Maarit Merikallio: "Optimum multiuser detection", Licentiate Course on Signal Processing in Communications, FALL-97

(Lok) Jouko Lokio: "Suboptimal Multiuser Detectors(AWGN)", Licentiate Course on Signal Processing in Communications, FALL-97

(Yas) Kazuyki Yasukawa and Laurence B.Milstein:"Finite Word Length Effects on the Performance of MMSE Receiver for DS-CDMA Systems", 1997 International Symposium on Personal, Indoor and Mobile Communications, Proceedings, volume 2/3, pages 724-728.

Riccardo De Gaudenzi:"The Influence of Signal Quantization on the performance of Digital Receivers for CDMA Radio Networks", Telecommunications Systems, Vol.8 No. 1 January-February 1997, pages 89-97

(Wei) P.Wei, J.R.Zeidler, and WW.H.Ku:"Adaptive Interference Suppression for CDMA Overlay Systems", IEEE Journal on Selected Areas in Communications, Dec. 1994, vol. 12 numb. 9, pages 1510-1522.

(Wer) Stefan Werner: "Adaptive multiuser Multiple-Antenna Receivers for CDMA Mobile Reception", Licentiate Course on Signal Processing in Communications, FALL-97