A Software Radio Architecture for Linear Multiuser Detection

Ivan Seskar and Narayan B. Mandayam
Wireless Information Network Laboratory (WINLAB)
Rutgers University
73 Brett Road, Piscataway, NJ 08854-8060
EMail: seskar@winlab.rutgers.edu, narayan@winlab.rutgers.edu
Fax: (732)-445-3693

Abstract

The integration of multimedia services over wireless channels calls for provision of variable quality of service (QoS) requirements. While radio resource management algorithms (such as power control and call admission control) can provide certain levels of variability in QoS, an alternate approach is to use reconfigurable radio architectures to provide diverse QoS guarantees. In this paper, we outline a novel reconfigurable architecture for linear multiuser detection, there by providing a wide range of bit-error-rate requirements amongst the constituent receivers of the reconfigurable architecture. Specifically, we focus on achieving this dynamic reconfiguration via a *software* radio implementation of linear multiuser receivers. Using a unified framework for achieving this reconfiguration, we partition functionality into two core technologies (field programmable gate arrays (FPGA) and digital signal processor (DSP) devices) based on processing speed requirements. We present experimental results on the performance and reconfigurability of the software radio architecture as well as the impact of fixed point arithmetic (due to hardware constraints).

This work was presented in part at the Conference on Information Sciences and Systems (CISS), March 1998, Princeton University. This work is supported in part by ONR under Grant No. 313/96/0211 and by the NSF under Grant Nos. EEC-9714727 and NCR-970636

1 Introduction

Future wideband CDMA systems will be required to support multimedia traffic with varied characteristics and quality of service (QoS) requirements. The integration of diverse services in the same system motivates the need for increasing capacity as well as dynamic QoS control. An important type of spread spectrum is direct sequence code division multiple access (DS-CDMA), used for communication in a multiple access environment. In a conventional CDMA system, all users interfere with each other. Potentially significant capacity increases and near/far resistance can theoretically be achieved if the negative effect that each user has on others can be canceled. A more fundamental view of this is multiuser detection [1], in which all users are considered as signals for each other. Then, instead of users interfering with each other, they are all being used for their mutual benefit by joint detection. The drawback of optimal multiuser detection is one of complexity so that suboptimal approaches are being sought. There is a wide range of possible performance/complexity combinations possible (see [1, 2, 3] and the references therein). Much of the present research is aimed at finding appropriate tradeoff between complexity and performance. The theory of multiuser detection has, thus, far outpaced the applicability. This paper aims at advancing the theory in directions which will facilitate its practical applications. Specifically, we focus on developing *software* radio architectures for implementing linear multiuser detectors for wideband CDMA communications.

In the past, radio systems have migrated from analog to digital in several ways including system control to source and channel coding to hardware technology. Recent progress in radio prototyping has extended the horizons of radio technology by liberating chronic dependence on hard-wired characteristics, including frequency band, channel bandwidth and channel coding. This effort has lead to the feasibility of dynamically programmable receivers or *software* radios [4]. The evolution of programmable hardware and increased flexibility via increased programmability has been accomplished by a combination of techniques. These include multiband antennas and RF conversion, wideband analog to digital (A/D) and digital to analog (D/A) conversion and the implementation of IF, baseband and bit-stream processing functions in general purpose programmable processors. In general, a canonical software radio architecture can be thought of as the comprehensive, consistent set of functions, components and design rules according to which systems of interest may be organized, designed and constructed. For a specific architecture, the key issues in design are in partitioning of functions and components such that functions are assigned to components and interfaces among components correspond to interfaces among functions. In an advanced application, a software radio does not just perform signal reception, but it also entails several other specialized functions. These include characterizing energy distributions in adjacent channels, recognizing the mode of incoming transmission, adaptive interference rejection, adaptive channel estimation and equalization as well as forward error correction. Further, software radios also support incremental service enhancements through a wide range of software tools. These tools may be used in analyzing the radio communications channels, improving design for optimum performance enhancements as well as testing and prototyping. An example of such a software radio implementation for CDMA systems is presented in [5] where a simple successive interference cancelation scheme [6] was implemented. The novelty of the design in [5] included the appropriate partitioning of functionality and design to allow for dynamic reconfiguration via programmable processors.

In this paper, we address the issue of dynamically reconfiguring linear multiuser architectures/detectors with a view to providing a dynamic QoS in an integrated multiple service wireless environment such as those in future wideband CDMA systems. The approach in this work is based on dynamically reconfiguring architectures (via software, i.e., a software radio) amongst different linear multiuser detectors to achieve a range of bit-error-rate (BER) requirements. Specifically, we consider four different linear receivers, namely, the matched filter receiver, the approximate decorrelator[7], the exact decorrelator[8] and the minimum mean-squared-error detector[9]. We first develop a unified architecture describes all the above linear detectors. Then, based on this unified approach we show how dynamic reconfigurability amongst these four receiver structures can be achieved via a software radio implementation. Our implementation is based on partitioning the functionality of the architecture between two core technologies (FPGA and DSP) according to processing speed requirements.

The paper is organized as follows. In section 2, we present a unified framework that describes all the above linear detectors as equivalent single-user receivers. We then present the software radio architecture for linear multiuser detection in section 3. In section 4 we present the experimental results on the performance of the software radio for linear multiuser detection, and conclude in section 5.

2 A Unified Architecture for Linear Multiuser Detection and Dynamic Reconfigurability

To overcome the complexity issue of optimum multiuser detection, several suboptimal schemes have been proposed (see [1, 2, 3]). These suboptimal schemes comprise both linear and nonlinear detectors. Our emphasis in this work is on linear multiuser detection schemes since they are relatively easier to implement. Specifically, we focus on the following linear detectors : the matched filter (MF) detector (which can be thought of as a degenerate multiuser receiver), the decorrelator (DC)[8], the approximate decorrelator (AD)[7] and the minimum mean-squared error (MMSE) detector[9], and develop a unified framework for describing these linear receivers. For simplicity in illustration, we will consider the case of synchronous reception of K users in a system. In this case, the output of the matched filter for the k^{th} user can be written as

$$y_k = A_k b_k + \sum_{j \neq k} A_j b_j \rho_{jk} + n_k, \;\; k = 1, \cdots . K \tag{1}$$

where A_k, and $b_k \in \{-1, 1\}$, are the amplitude and the bit of the k^{th} user respectively, and $n_k = \sigma \int_0^T n(t) s_k(t) dt$, with $s_k(t)$ being the signature waveform of the k^{th} user which is assumed to have unit energy. The crosscorrelation between the signature waveforms is defined as

$$\rho_{jk} = \int_0^T s_j(t) s_k(t) dt. \tag{2}$$

The matched filter outputs for the users in the system can be expressed in vector form as

$$\mathbf{y} = \mathbf{RAb} + \mathbf{n}, \tag{3}$$

where $\mathbf{y} = [y_1 \cdots y_K]^\top$, $\mathbf{b} = [b_1 \cdots b_K]^\top$, $\mathbf{A} = \text{diag}\{A_1 \cdots A_K\}$, and $\mathbf{n}$ is a zero-mean Gaussian random vector with covariance matrix equal to $\sigma^2 \mathbf{R}$. The matrix $\mathbf{R}$ of crosscorrelations among the user pairs is such that its ij^{th} entry is ρ_{ij}. Using the above representation, any of the linear multiuser schemes mentioned above can be realized as a linear transformation of the received vector $\mathbf{y}$ as

$$\hat{b}_k = \mathrm{sgn}(\mathcal{L}\mathbf{y})_k, \tag{4}$$

where $\mathcal{L}$ is the appropriate linear transformation. Specifically, for the matched filter, $\mathcal{L}$ takes the form

$$\mathcal{L}_{MF} = \mathbf{I}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix. For the decorrelator [8]

$$\mathcal{L}_{DC} = \mathbf{R}^{-\mathbf{1}}. \tag{6}$$

For the approximate decorrelator [7],

$$\mathcal{L}_{AD} = \mathbf{I} - \delta\mathbf{J}, \tag{7}$$

where $\mathbf{J}$ is derived (see [1]) on the basis of the assumption that $\mathbf{R}$ is strongly diagonal as $(\mathbf{I} + \delta\mathbf{J})^{-1} = \mathbf{I} - \delta\mathbf{J} + o(\delta)$. For the MMSE detector [9],

$$\mathcal{L}_{MMSE} = (\mathbf{R} + \sigma^2\mathbf{W}^{-\mathbf{1}})^{-1}, \tag{8}$$

where $\mathbf{W} = \mathbf{A}^\top\mathbf{A}$. The linear structure of the detectors given above further allows them to be realized as single-user detectors for each user. Specifically, each of the above systems can be realized for the k^{th} user by correlating the incoming signal with a "modified" filter $h_k(t)$. Specifically, the decorrelator is realized as

$$h_k(t) = \sum_{j=1}^{K} \mathbf{R}^{+}_{\mathbf{kj}} s_j(t), \tag{9}$$

where $\mathbf{R}^{+}_{\mathbf{kj}}$ denotes $(\mathbf{R}^{-\mathbf{1}})_{kj}$. The approximate decorrelator can be realized as

$$h_k(t) = s_k(t) - \sum_{j \neq k} \rho_{jk} s_j(t), \tag{10}$$

and the MMSE detector as

$$h_k(t) = \sum_{j=1}^{K} (\mathbf{R} + \sigma^{\mathbf{2}}\mathbf{W}^{-\mathbf{1}})^{+}_{\mathbf{kj}} s_j(t), \tag{11}$$

where $(\mathbf{R}+\sigma^{\mathbf{2}}\mathbf{W}^{-\mathbf{1}})^{+}_{\mathbf{kj}}$ again denotes $[(\mathbf{R}+\sigma^2\mathbf{W}^{-\mathbf{1}})^{-1}]_{kj}$. Note that for the matched filter $h_k(t) = s_k(t)$. Thus it is seen that any of the linear multiuser detectors described above can be realized by appropriately choosing the filter taps of the "modified" matched filter $h_k(t)$. This forms the basis of the software radio architecture we develop in this paper for linear multiuser detection.

Further, under some assumptions[1] on the signature sequences used in the system, it can be shown that these linear multiuser detectors yield a variable QoS that covers a wide range of values[2]. Specifically, when the measure of quality is the BER achieved by a user, the relative performance of these receivers may be classified as[3]

$$\mathrm{BER}_{MF} \geq \mathrm{BER}_{AD} \geq \mathrm{BER}_{DC} \geq \mathrm{BER}_{MMSE}. \tag{12}$$

Thus reconfiguring the detectors among the matched filter and the structures in (9), (10), and (11) allows the option of a variable QoS from moderate (for a matched filter) to very high (for the MMSE detector). As an example, a user may switch to a matched filter configuration when using voice traffic while the MMSE mode may be preferred for data traffic. Further, we can also account for different data rates by deriving appropriate single-user linear filters for the multirate schemes outlined in [12, 13].

3 Software Radio Architecture for Linear Multiuser Detection

Figure 1 shows the canonical software radio architecture for linear multiuser detection. The architecture includes channel processing (such as translation from IF to baseband), environment processing (e.g., estimation of signal and interference parameters and correlations), matched filtering and information bit-stream processing (e.g., FEC or convolutional decoding, soft decisions etc.). These functionalities are partitioned into two core technologies based on processing speed requirements. These two technologies are based on field

[1] If the crosscorrelations among all signature sequences are very low, then all the detectors achieve similar performance

[2] The MMSE detector achieves the same performance (or worse) as the decorrelator at very high signal-to-noise ratios, and is better at lower signal-to-noise ratios [10]

[3] In the absence of perfect power control and under the special case of two users, there are counterexamples to the general relationship regarding the relative performance of linear multiuser receivers (see [11] for details and also other similar conjectures)

programmable gate arrays (FPGA) and digital signal processor (DSP) devices. The key idea behind the software architecture is to dynamically reconfigure the matched filter according to the desired QoS (corresponding to the one of the appropriate linear detectors). In the following sections of the paper, we present a detailed description of the software radio architecture that we use to allow variable QoS for users in a CDMA system. Specifically, we focus on the partitioning of functionality and the design of building blocks that allow this architecture to dynamically reconfigure between the various linear detection schemes.

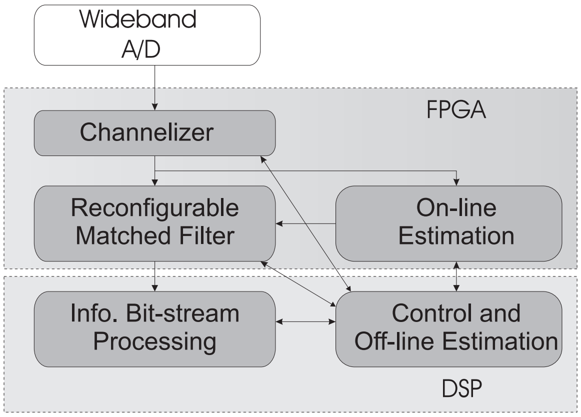


Figure 1: Functional Architecture of a Software Radio for Linear Multiuser Detection

3.1 Logical Partitioning of the Architecture

We first begin by describing the logical partitioning of the functionality required by a given user under the reconfigurable architecture for linear multiuser detection. In Figure 2, we show the necessary operations required by each type of linear detector among the MF, AD, DC and MMSE receivers. Even though the different types of linear receivers are shown as separate entities in Figure 2, essentially each user's radio could possibly have only one variable filter-tap receiver implemented using Xilinx FPGAs[4]. All classes of the above receivers require two common generic operations namely, estimation of path delays of the users and the generation of PN sequences $\{s_k(t)\}$ of the users in the system. The MF

[4]The software radio could also use two sets of filters to accommodate smooth transition during reconfiguration

is probably the simplest in that for any user k, it just uses the information from these two generic operations in determining the timing offset of the PN sequence $s_k(t)$ for the specific user. For the AD, the complexity is slightly higher than that of the MF in that the "modified" matched filter taps are adjusted according to the formulation in equation (10). As seen in Figure 2, the additional functionality required here is the crosscorrelation values $\{\rho_{kj}\}$ and also the signature of sequences of all the users $\{s_k(t)\}$. From (9), we see that the functional operations required for the DC are additionally the computation of the inverse matrix of crosscorrelations $\mathbf{R^{-1}}$ and also the column vector corresponding to the k^{th} user, i.e., $\mathbf{R^{+}_{kj}}$. The MMSE receiver (see equation (11)) incurs the additional complexity over the DC in that it also requires estimates of the received signal powers of the users in the system. In this paper, we do not explicitly address the issue of estimating the path delays or the received signal powers of the users in the system. We assume that the software radio architecture proposed here does have the functionality required to do both the estimation operations similar to that used in conventional radio designs. However, the software radio architecture presented here is versatile in that it can easily allow a variety of signal processing algorithms to be implemented for accomplishing the required estimation.

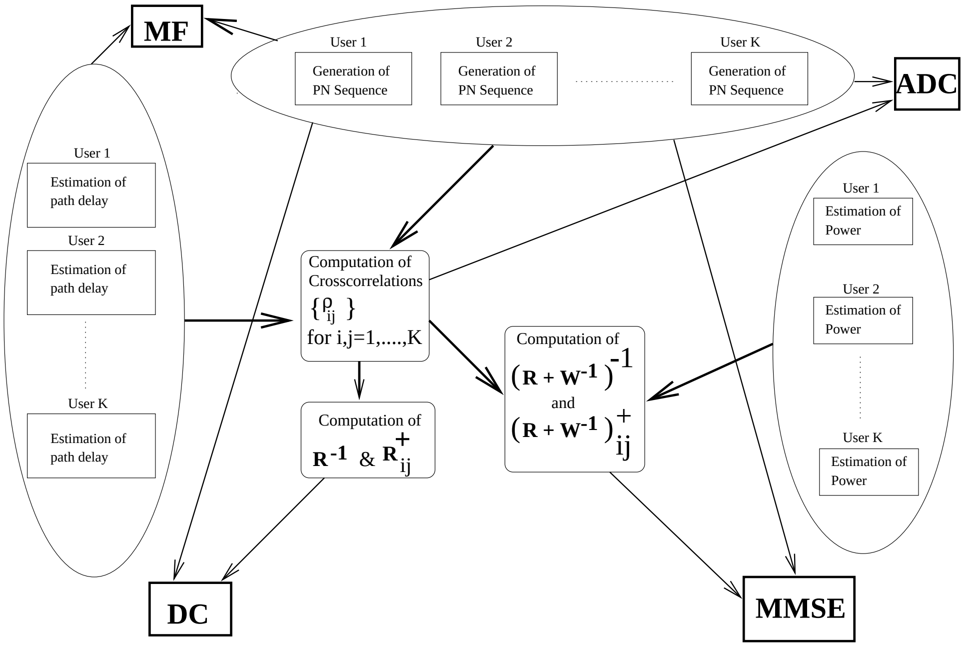


Figure 2: Logical Partitioning of Functionality in a Software Radio Receiver for Linear Multiuser Detection

3.2 Practical Implementation

In this section, we discuss the practical implementation aspects of the software radio architecture for linear multiuser detection. Specifically, we focus on the testbed for linear multiuser detection followed by the partitioning of resources between FPGA and DSP devices, and finally, the details on the implementation of the correlation elements constituting the reconfigurable radio receiver.

3.2.1 Testbed

The core of the hardware part of the testbed (see Figure 3) is based around APTIX MP3 board [14] with up to 12 FPGA components (up to 432,000 programmable gates) for sample-level processing tasks with rates of up to 30.0 MHz. Information-rate tasks (Viterbi decoding, deinterleaving etc.) are performed by the VMEbus based Pentek 4270 Quad 'C40 DSP Processor board [15]. As a baseband front-end, the testbed has two sets of dual (I and Q) analog-to-digital converters for input (Analog Devices AD9762XR 12-bit, 41 MHz converters [16] and PENTEK 6472 10-bit, 70 MHz converters [17]) and one set of digital-to-analog converters (Analog Devices AD9042ST 12-bit, 100 MHz [18]). Analog Devices converters are connected to the FPGA board through the custom adapters while PENTEK converters are connected to the DSP via a Multiband Digital Receiver (PENTEK 4272) [19] acting as a *channelizer* in Figure 1. The channelizer performs frequency down-conversion, low-pass filtering and decimation of the sampled baseband signal. It is used for selection of the service bandwidth (i.e., the *tuning band*) from among those available in the sampled signal. The Multiband Digital Receiver used as a channelizer has two narrowband receivers with dynamic range of $1KHz - 1MHz$ and one wideband receiver with dynamic range of $2MHz - 35MHz$. Thus it is capable of supporting a wide range of output sample rates.

A SUN Workstation accesses the VMEbus through a Bit3 SUN-Sbus to VMEbus adapter and is used as a primary data stream source as well as development host. Development tools are centered around Signal Processing Worksystem. SPW [20] is a computer-aided design (CAD) tool that allows for the simulation and design of the complex communication systems based on block diagrams. It has a rich library of common communication

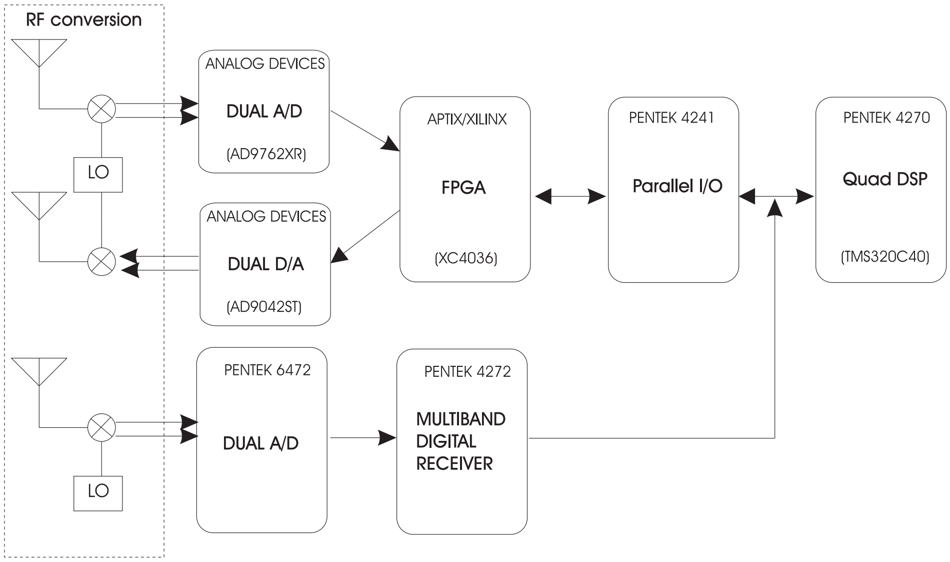


Figure 3: Testbed block diagram

block as well as facilities for creation of custom coded blocks. Once entered, the block diagram of the target system can be simulated by using SPW's Signal Flow Simulator [21]. If these floating-point simulations are producing satisfactory results, the block diagram can be partitioned into DSP and hardware parts.

The part of the design that is targeted for DSP implementation is prepared by SPW's Code Generation System (CGS) (or MultiProx in case of partitioning into multiple DSPs [22]) and is directly downloadable into the Quad TMS320C40 floating-point DSP board.

SPW's Hardware Design System (HDS) is used to model the behavior of a fixed-point part of the design. Again SPW's Signal Flow Simulator is used to verify the fixed-point model functionality. After the design is verified the corresponding Hardware Description Language (HDL) code is automatically generated. This code is then synthesized (and/or simulated by the event driven HDL simulator like Synopsys VSS [23]) by appropriate set of tools (e.g., Synopsys Design Compiler [24]). The resulting design is further processed by the XILINX XACT tool [25] in order to generate the FPGA chip layout and routing[5], there by producing the configuration bit-stream[25]. This configuration bit-stream[6] defines

[5]At this stage the hardware portion of the design can to be further re-partitioned into multiple FPGA components if it does not fit into a single FPGA

[6]This is not to be confused with the information bit-stream

the combinatorial circuitry, flip-flops, interconnect structure, and the I/O buffers inside a particular FPGA device. APTIX tools [14] are used to interconnect the FPGAs, connect FPGAs and DSPs through parallel I/O board and for routing of debugging signals to the control probes of the logic analyzer.

3.2.2 Resource Partitioning of the Architecture

As mentioned earlier, the software radio architecture for linear multiuser detection is partitioned into two core technologies, namely field programmable gate arrays (FPGA) and digital signal processor (DSP) devices. This partitioning is usually driven by the required functionality of the radio device and also the processing speed requirements. The algorithmic complexity of the linear multiuser receivers increases with increase in performance. Specifically, the complexity of the signal processing algorithms corresponding to the MF, AD, DC and MMSE receivers can be classified as

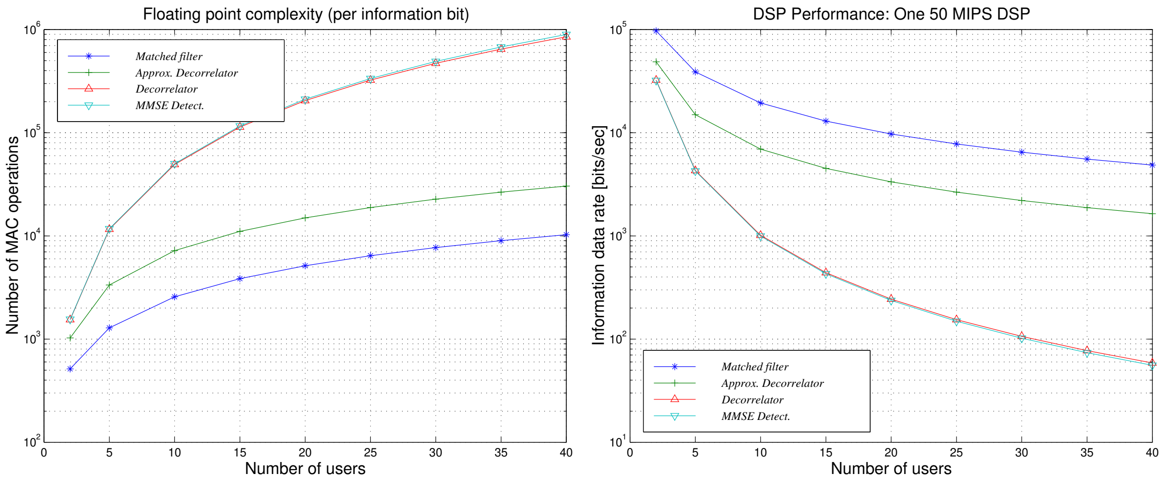
$$C_{MF} < C_{AD} < C_{DC} < C_{MMSE}, \tag{13}$$

where $C_{receiver}$ denotes the complexity of the particular *receiver*. In general, for a system with K users, processing gain N and oversampling factor O_s, the floating point complexity in terms of the number of multiply-accumulate (MAC) operations is given for each of the above receivers as (see also [10] for similar complexity trade-offs).

$$\begin{aligned}
C_{MF} &= K(NO_s + 1) \qquad (14)\\
C_{AD} &= K(NO_s + 1) + (K-1)NO_s + (K-1)\log_2 K\\
C_{DC} &= K(NO_s + 1) + 2K(K-1)NO_s + \frac{2}{3}(K)^3\\
C_{MMSE} &= K(NO_s + 1) + 2K(K-1)NO_s + 2(K)^2 + \frac{4}{3}(K)^3 + C(amp),
\end{aligned}$$

where $C(amp)$ denotes the complexity due to amplitude estimation that is incurred in the MMSE receiver (which is not explicitly considered in this paper). We show in Figure 4 (a), the number of MAC operations for each receiver as a function of the number of users. Alternately, we can evaluate the number of users that can be supported as a function of achievable information data rates (for the different receiver structures) where the active constraint is the limitations in the complexity or processing speeds of the DSP device.

In Figure 4 (b), a simple illustration is shown for a 50 MHz floating point TMS320C40 DSP. Thus we see that, an all DSP implementation imposes serious constraints on the achievable data rates when the number of users increases especially, when we operate with more complex receivers.



(a) MAC Operations vs. No. of Users

(b) Achievable Rates vs. No. of Users

Figure 4: Processing Speed Constraints

In this work, the software radio architecture for reconfigurable linear multiuser detection is implemented by partitioning the resources between FPGAs and DSPs as shown in Figure 5. The FPGA segment of the architecture includes the PN sequence generators, two reconfigurable blocks (that determine the filter taps of the appropriate linear receivers), an on-line estimator module and the correlator. The motivation for the particular selection of constituents that comprise the FPGA segment is that the functionality provided by each constituent here can be easily handled by the processing speeds of the FPGA hardware. The two reconfigurable blocks contain the core of the sample level processing for each of the multiuser detectors and are actually implemented in separate FPGA components to facilitate on-the-fly reconfigurability[7]. This particular partitioning enables us to reconfig-

[7]The above duplicity may be avoided in the XC62xx XILINX FPGA family which has in-system reprogrammability allowing sections of the device to be reprogrammed without disturbing circuits running in other sections

ure one of the blocks while the other is running and therefore, by oversizing the hardware, avoid loss of data during switchover to different receiver structures. The rest of the sample level processing logic (PN sequence generators, control multiplexer, and correlator) is implemented in a separate FPGA component since it does not require reprogrammability. The on-line estimation block is used to perform timing estimates of the incoming signals which are then fed along with the reference PN sequences to compute the appropriate filter taps. Further, there is also a provision for refining the on-line estimates by using more sophisticated off-line algorithms. Additionally, an FPGA component is used as an interface between the DSP segment and APTIX board [14].

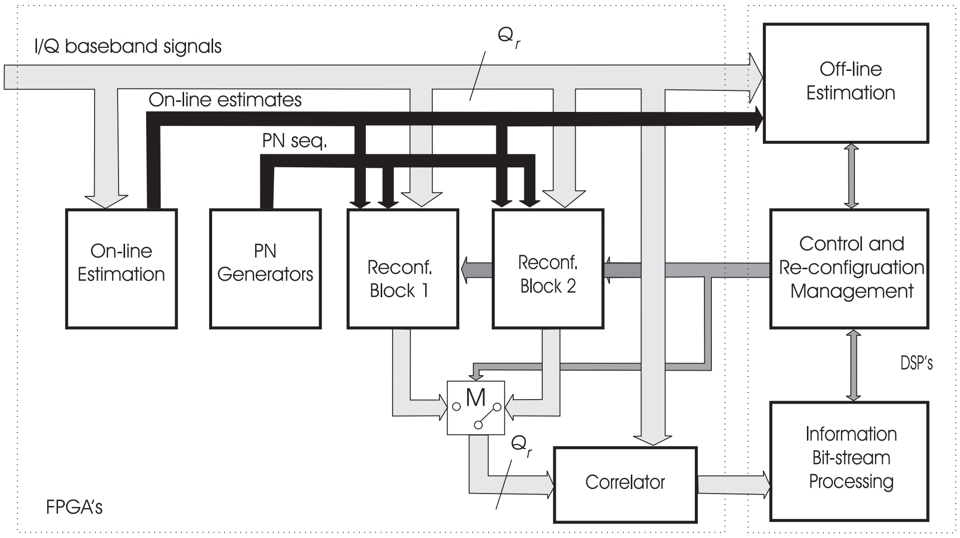


Figure 5: Block-diagram of Software Radio Implementation.

The more algorithmically complex operations required for the linear multiuser receivers is partitioned into the DSP segment of the software radio architecture shown in Figure 5. This segment is implemented using a a quad TMS320C40 DSP board. The operations performed in this segment include off-line estimation procedures, information bit-stream processing as well as control and reconfiguration management. The off-line estimation block includes estimation of powers of the received signals (for the MMSE detector), and computation of the inverse of the matrix of crosscorrelations (in the case of both the decorrelator and the MMSE detector). Further, it can also refine estimates of the on-line estimation block in the FPGA segment. The information bit-stream processing functions performed vary from error control techniques, such as FEC or convolutional decoding, to soft decision decoding. The control and reconfiguration management block in the DSP segment of the architecture basically determines the variable QoS that can be achieved by dynamic reconfiguration of the receiver structures in the FPGA segment. The reconfiguration of the blocks in the FPGA segment is directly controlled by the control and reconfiguration management block. The subsequent switching amongst receivers is achieved by a control multiplexer M (shown in Figure 5) that is also controlled by the control and reconfiguration management block. Thus the DSP segment of the device acts to achieve the variable QoS requirements of the specific type of service. Implicit in the control and reconfiguration management block is also the capability to interact with higher layer protocols/stacks to facilitate the QoS demand of a specific data stream. At the link layer, the actions of the DSP segment of the device also encompass environment processing such as sensing interference levels. The reader should note that the DSP devices have access to the configuration bit-streams for each of the receiver structures and can download the particular configuration bit-stream as and when required.

While we do not give the detailed schematics of each block in Figure 5, we present the schematic for the correlator block in Figure 6. This block is used in all four receiver implementations and is based on a binary multiplier as opposed to the traditional implementation with the sign bit inverter. This is due to the fact that the "modified" matched filters presented in equations (9), (10) and (11) result in filter taps (input Ref_PN in Figure 6) that are not necessarily binary (except in the case of the conventional matched filter receiver).

In Table 1, we show the relative hardware complexity of the different linear multiuser detection schemes in terms of the required number of configurable logic blocks (CLB). A CLB, in case of XILINX 4000 series FPGA, comprises of a pair of flip-flops and two independent (Boolean) logical four-input function generators [25]. As can be seen the complexity for each detector increases with both increasing number of users K and increasing precision in quantization (Q_r denotes the number of bits used in quantization). As expected, the matched filter complexity remains invariant to the number of users in the system and de-

Quantization	MF	AD	DC & MMSE
$Q_r = 4$ bits	25	$31 + 13K$	$37 + 3K(K-1)$
$Q_r = 5$ bits	27	$40 + 14K$	$47 + \lceil 13/4 \rceil K(K-1)$
$Q_r = 6$ bits	29	$57 + 15K$	$64 + \lceil 7/2 \rceil K(K-1)$

Table 1: Complexity comparison of different detectors (in number of CLB's)

pends only on the precision of quantization. Even though the decorrelator (and the MMSE) achieves better QoS, it comes at the expense of an exponential increase in complexity (with increasing number of users) over the AD detector. This directly maps to an increase in processing power requirements of the FPGA segment of the software radio architecture. This again motivates switching to lower order receivers when the QoS requirements are moderate. The complexity of operations required to be performed in the DSP segment of the architecture involve matrix inversion and off-line estimation of amplitudes, etc.

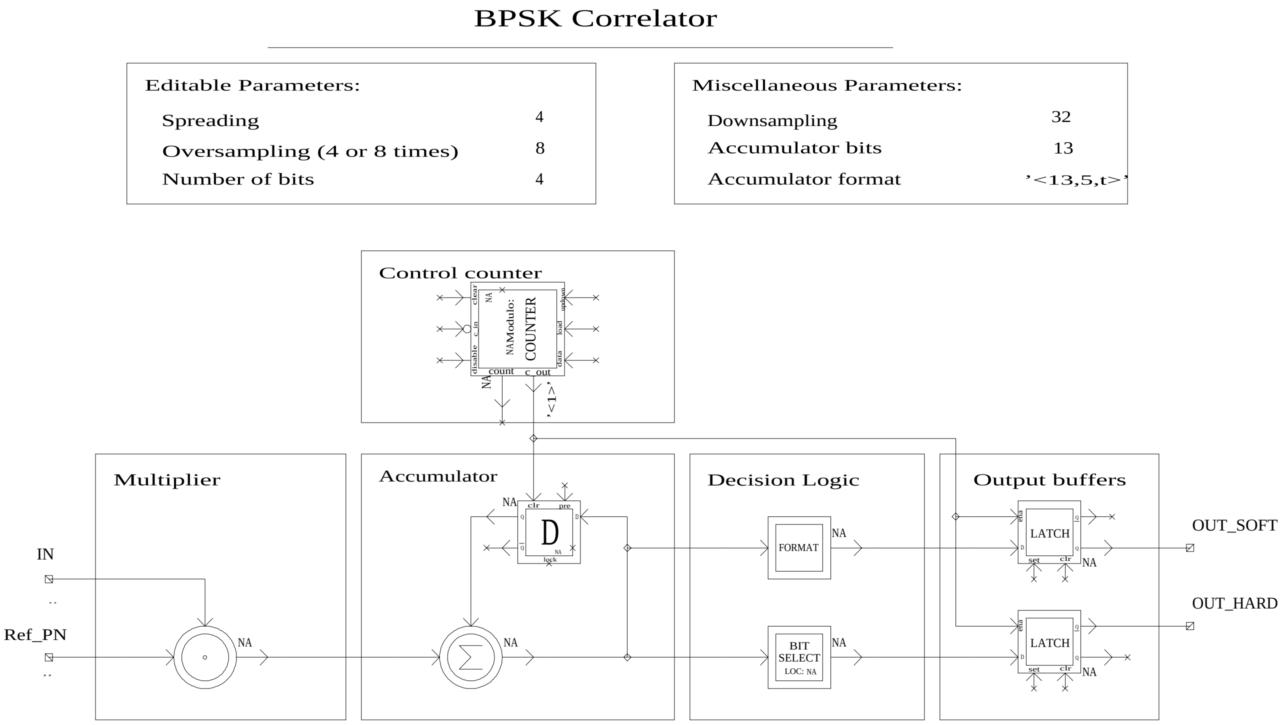


Figure 6: Modified Correlator Implementation

4 Experimental Results and Discussion

We begin the presentation of the experimental results by considering first, floating point implementations of the linear multiuser receivers. In all our experiments, the transmitter powers of the users is controlled perfectly so that they are all received at the same received power level. Further, the floating point results presented here agree very well with the analytical results on the performance of these receivers from [1, 7, 8, 9].

To show the flexibility of the software radio architecture in providing variable QoS measured as the bit-error performance (see equation 12), we consider two sets of experiments. In the first set of experiments, we fix the number of users in the system at $K = 15$. We then show the experimental bit-error-rate achievable for SNRs varying from $0\,dB$ to $10\,dB$ for each of the detectors, namely MF, AD, DC and MMSE for a system where the processing gain is $N = 64$ and $N = 128$. In Figure 7, it is seen that at low SNRs the range of QoS achievable is quite limited. However, at higher SNRs, the DC and MMSE provide a BER gain of upto three orders of magnitude compared to either the MF or the AD. As expected, the AD is again slightly better than the MF for the set of operating points considered here. For the case when $N = 128$, it is seen that the dynamic range of QoS achievable is greater among the detectors at high SNRs. Specifically, the AD provides upto an order of magnitude better performance than the MF.

A variation of the above experiments is presented in Figure 8, where we show the variations in QoS achieved for different number of users in the system for a fixed SNR of $10\,dB$. This shows that when the number of users in the system increases, the users can still operate at the same SNR, but can switch to a higher complexity detector to maintain the same QoS. It is seen that the dynamic range of switching from the MF to the AD is upto a load factor (defined as the ratio $\frac{K}{N}$) of $1/3$. This again conforms to the theoretical prediction on the load factor given in [7]. As an example, it is seen that when $N = 64$, if users in a system desire a BER of $\approx 10^{-3}$, they can operate on a MF receiver upto 5 users, and on an AD receiver upto 10 users and the decorrelator receiver upto 30 users without changing their received power levels. Additionally, if some of the users require more stringent BERs such as $\approx 10^{-5}$, they could operate entirely on a DC realization of the "modified" matched filter. Further, when the processing gain is $N = 128$, it is seen that the there is a greater range in achievable capacity (that is number of users that can be supported) by switching to the DC and MMSE receivers.

While the results presented uptil now have been based on using floating point arithmetic, we now focus on the effects of quantization in the performance of the above linear detectors. We consider fixed point arithmetic operations for the DSP processors in the software radio architecture and evaluate the average probability of error performance of all the detectors, namely the MF, AD and DC. Specifically, we consider fixed point operations using $2-bit$, $3-bit$, and $4-bit$ quantization. The experimental results for the matched filter are shown in Figure 9, for the approximate decorrelator in Figure 10, and the decorrelator in Figure 11. The performance results using fixed-point implementations are also compared with the floating point results. It is seen that all the detectors experience a degradation in performance, with the matched filter being the least sensitive and the decorrelator being the most sensitive to quantization effects. However, the encouraging note is that the use of even a $6-bit$ precision quantizer seems to pull performance close to that of the floating point reference results.

Another interesting impact of such fixed point operations in realizing detectors is the effect on the "near-far" resistance of these detectors. In Figure 12, we consider the average bit-error rate achieved by the MF, AD and DC receivers for the case of $4-bit$ quantization as a function of interference powers. In this experiment, the processing gain is taken to be $N=64$, and the number of users is $K=5$. The desired user's power is fixed, while the interfering powers are increased. As expected, the MF and AD are not near-far resistant, and show degradations in performance as the interfering powers increase. Further, the AD shows a more graceful degradation in performance relative to the matched filter. The DC, which is theoretically near-far resistant [8], however fails to maintain this property in the presence of quantization effects. However, the BER performance is still superior to the other two detectors.

5 Conclusion

In this paper, we have presented a reconfigurable software radio architecture for linear multiuser detection in CDMA systems. The reconfigurability of linear multiuser receivers allows for the integration of multimedia services over wireless channels with variable quality of service (QoS) requirements. While radio resource management algorithms can provide certain levels of variability in QoS guarantees, we have shown that reconfigurable radio architectures can also provide diverse QoS guarantees ranging in several orders of magnitudes in terms of BER requirements. We have showed the feasibility of achieving this dynamic reconfiguration via a *software* radio implementation of linear multiuser receivers. We first presented a unified framework for achieving this reconfiguration, and then partitioned functionality into two core technologies (field programmable gate arrays (FPGA) and digital signal processor (DSP) devices) based on processing speed requirements. Our experimental results on the performance and reconfigurability of the software radio architecture show that diverse services such as voice and data traffic can be simultaneously supported using such a software radio architecture for linear multiuser detection. The impact of fixed point arithmetic that often arises due to hardware constraints, shows that the degradation in receiver performance can be compensated by using as low as $4-bit$ quantization even for complex multiuser detection algorithms. Future work includes studying the effect on-line estimation algorithms and also the impact of processing delays on QoS guarantees.

References

[1] S. Verdú, *Multiuser Detection*, Cambridge University Press, New York, 1998.

[2] A. Duel-Hallen, J. Holtzman, and Z. Zvonar, "Multiuser Detection for CDMA Systems ", in *IEEE Personal Communications*, April 1995, pp. 46–58.

[3] S. Moshavi, "Multiuser Detection for DS-CDMA Communications", in *IEEE Communications Magazine*, October 1996, pp. 124–136.

[4] J. Mitola, "The Software Radio Architecture", in *IEEE Communications Magazine*, May 1995, pp. 26–38.

[5] K. Pedersen, T. Kolding, I. Seskar, and J. Holtzman, "Practical Implementation of Successive Interference Cancellation in DS/CDMA Systems", in *Proceedings of ICUPC'96*, September 1996, pp. 321–325, Cambridge, MA.

[6] P. Patel and J. M. Holtzman, "Analysis of a Simple Successive Interference Cancellation Scheme in DS/CDMA System", in *IEEE JSAC — Special Issue on CDMA*, June 1994, vol. 12, pp. 796 – 807.

[7] N. B. Mandayam and S. Verdú, "Analysis of an Approximate Decorrelating Detector", *Wireless Personal Communications*, vol. 6, pp. 97–111, January 1998.

[8] L. Lupas and S. Verdú, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels", *IEEE Transactions On Information Theory*, vol. 35, no. 1, pp. 123–136, January 1989.

[9] U. Madhow and M. Honig, "MMSE Interference Suppression of Direct-Sequence Spread-Spectrum CDMA", *IEEE Transactions On Communications*, vol. 42, no. 12, pp. 3178–3188, December 1994.

[10] R. M. Buehrer, S. N. Correal, and B. D. Woerner, "A Comparison of Multiuser Receivers for Cellular CDMA", in *Proc. of 1996 IEEE Global Telecommunications Conference*, November 1996, pp. 1571–1577, London, UK.

[11] G. V. Moustakides and H. V. Poor, "On the Relative Error Probabilities of Linear Multiuser Detectors", *Proceedings of the* 36^{th} *Annual Allerton Conference on Comm., Control & Computing*, September 1998, Allerton, IL.

[12] M. Saquib, R. Yates, and N. B. Mandayam, "Decorrelating Detectors for a Dual Rate Synchronous DS/CDMA System", *Wireless Personal Communications*, To Appear.

[13] J. Chen and U. Mitra, "Further Results for Multirate Decorrelators for Synchronous DS-CDMA Systems", in *Proc. of the 34th Annual Allerton Conference on Communication, Control and Computing*, October 1996, Allerton, IL.

[14] Aptix Corporation, *FPCB Development System documentation*, 1993.

[15] PENTEK, Inc, *Operating Manual: MODEL 4270 Quad TMS320C40 DSP Board*, 1995.

[16] Analog Devices, Inc., *Data Sheet: AD9762* 12-bit, 100 MSPS+ TxDAC D/A Converter*, 1996.

[17] PENTEK, Inc, *Operating Manual: MODEL 6472 Dual Channel, 70 MHz, 10-bit, A/D Converter*, 1996.

[18] Analog Devices, Inc., *Data Sheet: AD9042 12-bit, 41 MSPS Monolithic A/D Converter*, 1996.

[19] PENTEK, Inc, *Operating Manual: MODEL 4272 Multiband Digital Receiver*, 1996.

[20] Inc. Alta Group of Cadence Design Systems, "Signal Processing WorkSystem (SPW)", in *SPW Software Release 3.1 documentation*, June 1994.

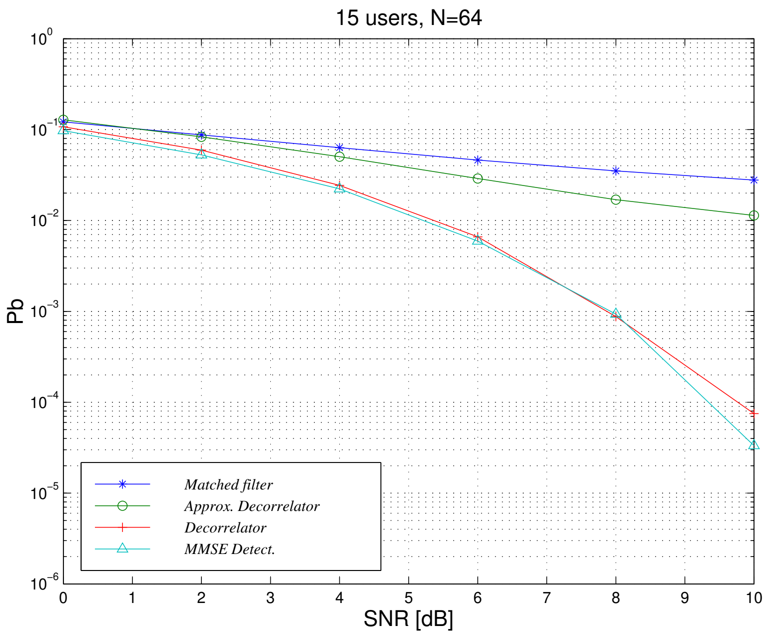
[21] Alta Group of Cadence Design Systems, Inc., *SPW Signal Flow Simulator*, 1996.

[22] Alta Group of Cadence Design Systems, Inc., *SPW MultiProx*, 1996.

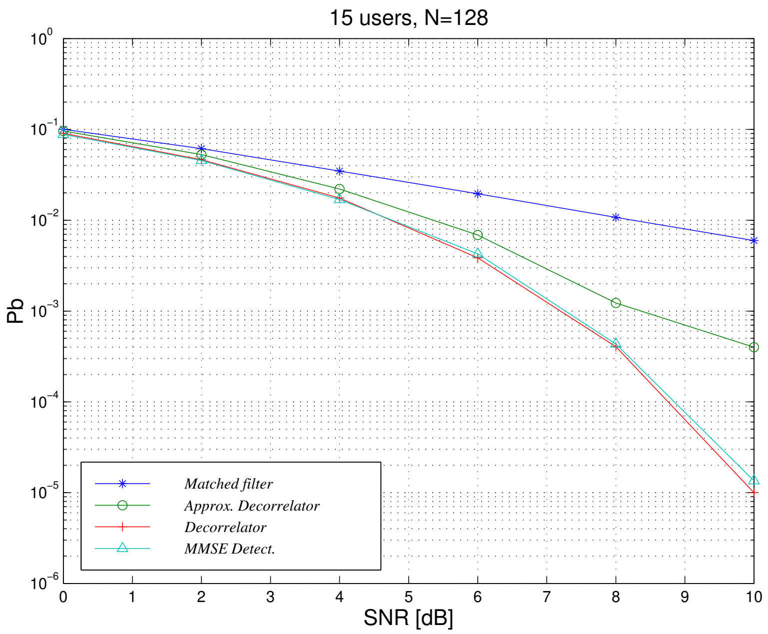
[23] SYNOPSYS, Inc, *VHDL System Simulator (VSS)*, 1997.

[24] SYNOPSYS, Inc, *Design Compiler*, 1997.

[25] XILINX, *XACT Reference Guide*, 1994.

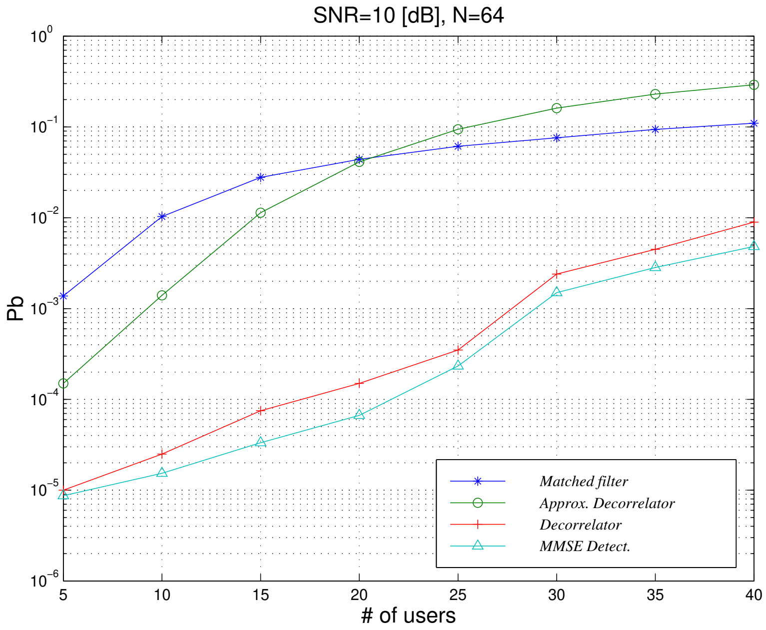


(a) BER range vs SNR for $N = 64$ and $K = 15$ users

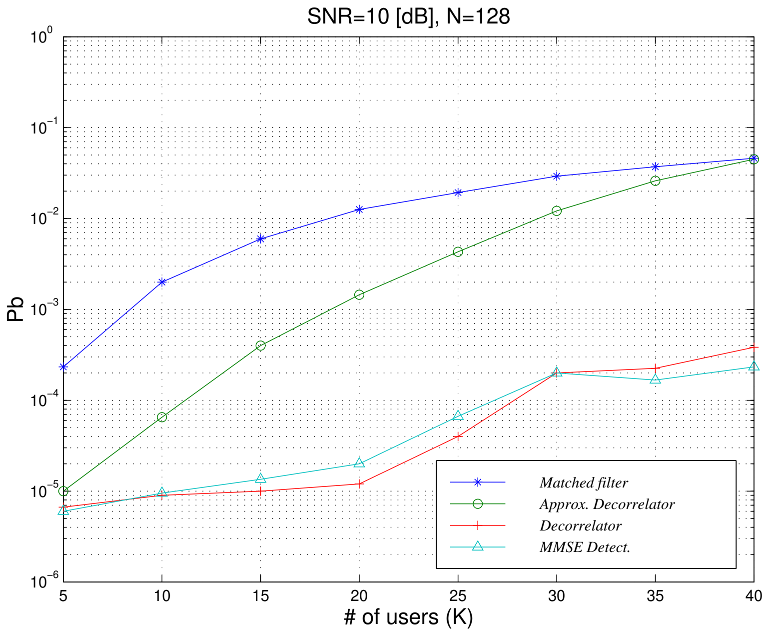


(b) BER range vs SNR for $N = 128$ and $K = 15$ users

Figure 7: BER range vs SNR



(a) BER range vs number of users for $N = 64$, $SNR = 10dB$



(b) BER range vs number of users for $N = 128$, $SNR = 10dB$

Figure 8: BER range vs Number of users

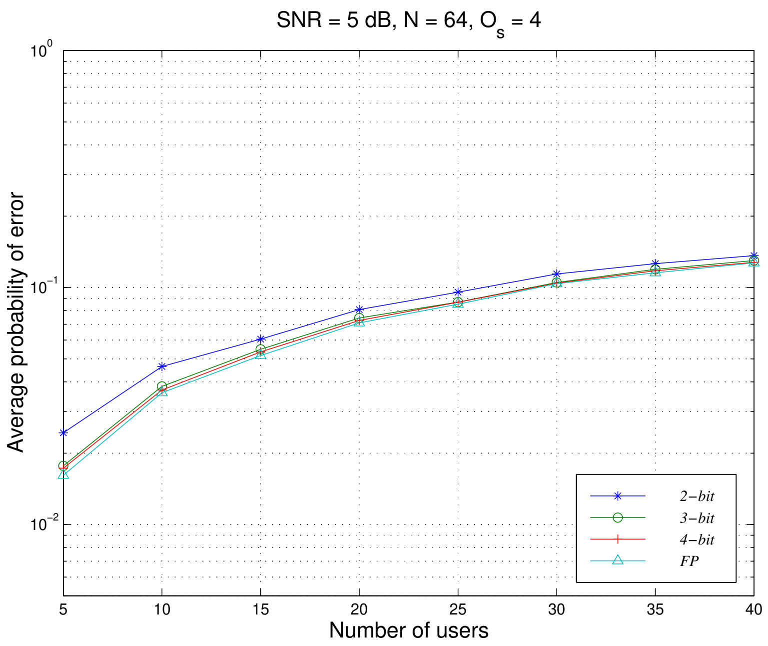


Figure 9: Matched filter quantization effects

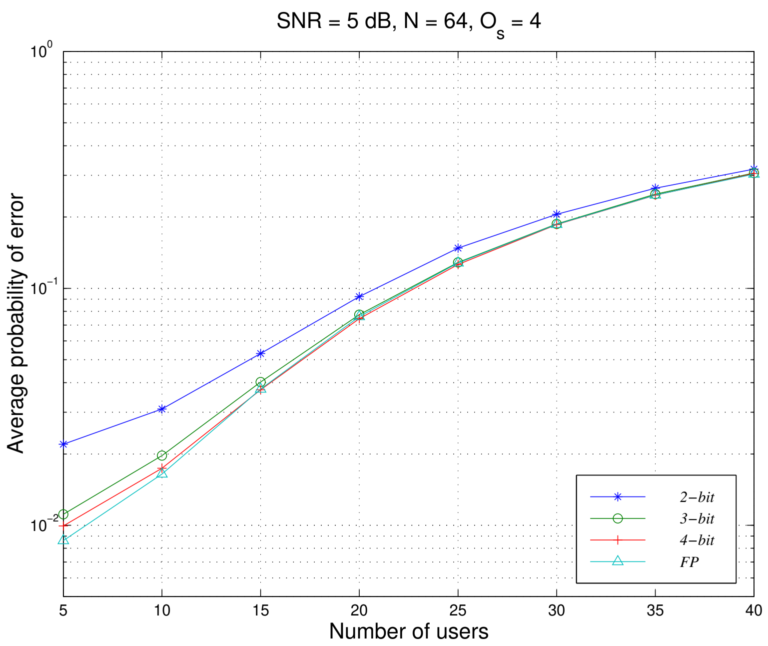


Figure 10: Approximate Decorrelator quantization effects

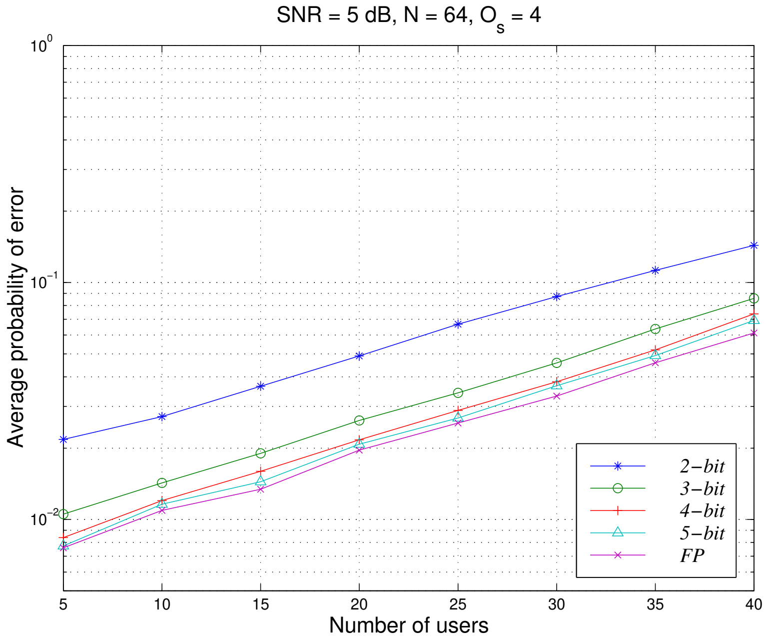


Figure 11: Decorrelator quantization effects

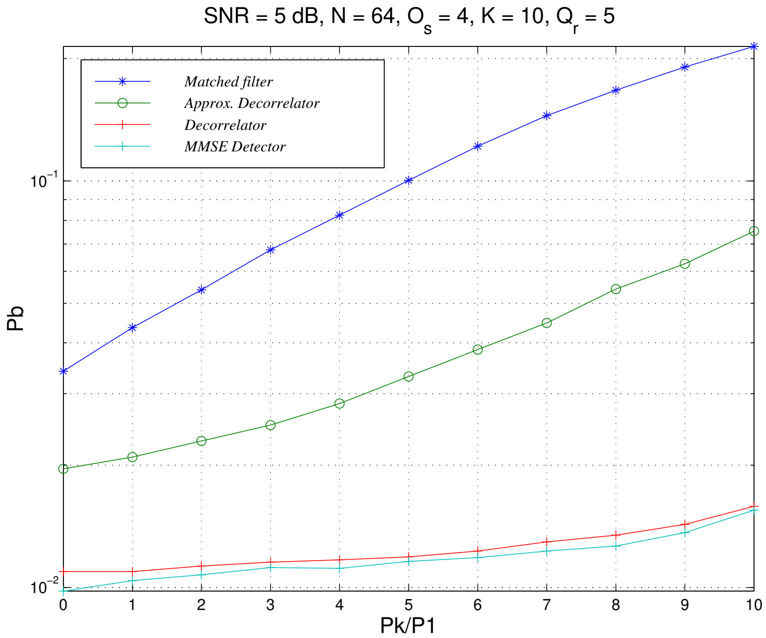


Figure 12: Near-far resistance with 4-bit quantization