

An All-Digital, High Data-Rate Parallel Receiver

M. Srinivasan and C.-C. Chen
Communications and Systems Research Section

G. Grebowsky and A. Gray
Goddard Space Flight Center, Greenbelt, Maryland

The all-digital, high data-rate parallel receiver that is currently being developed jointly by the Jet Propulsion Laboratory (JPL) and Goddard Space Flight Center (GSFC) is presented. The role of JPL has been to analyze and simulate the receiver architecture and subsystems. Implementation of the receiver using field-programmable gate arrays (FPGAs) and subsequent application-specific integrated circuit (ASIC) design take place at GSFC. The parallel receiver architecture that is currently being implemented differs from the original multirate filter-bank-based parallel architecture that was first developed by JPL. This alternate parallel receiver (APRX) is essentially a frequency-domain implementation of detection filtering and symbol-timing correction and is significantly easier to implement than the original version of the parallel receiver (PRX). It is shown that the APRX is equivalent to both the PRX and the conventional serial receiver in terms of performance. Results on the effect of analog antialiasing filter bandwidth and analog-to-digital sampling offset on the receiver performance are presented, along with discussion and results of the frequency-domain digital data-transition tracking-loop simulation.

I. Introduction

Current NASA Earth-orbiting missions and commercial satellite systems call for downlink data rates of several hundred megabits per second. The parallel receiver project undertaken by the Jet Propulsion Laboratory (JPL) in collaboration with Goddard Space Flight Center (GSFC) has as its goal the design and implementation of a low-cost, all-digital receiver that can process high data rates. In [1], a receiver architecture was proposed that utilizes only a small number of high-speed components, including analog-to-digital (A/D) converters along with a majority of lower-speed components operating in parallel. The parallel receiver (PRX) architecture that was developed in [1] is based upon multirate digital filter-bank theory [2].

The all-digital receiver performs the functions of demodulation to baseband, matched filtering for symbol detection, and carrier and symbol synchronization. The multirate filter-bank architecture that is used in the PRX for performing the filtering operations (rejection of double frequency terms and matched filtering) is derived in [2] and is shown in Fig. 1. In summary, the input signal sequence is parallelized into subsequences that occupy separate frequency subbands by using the theory of perfect reconstruction filter banks [2]. Demodulation, lowpass filtering, and matched filtering are then inserted

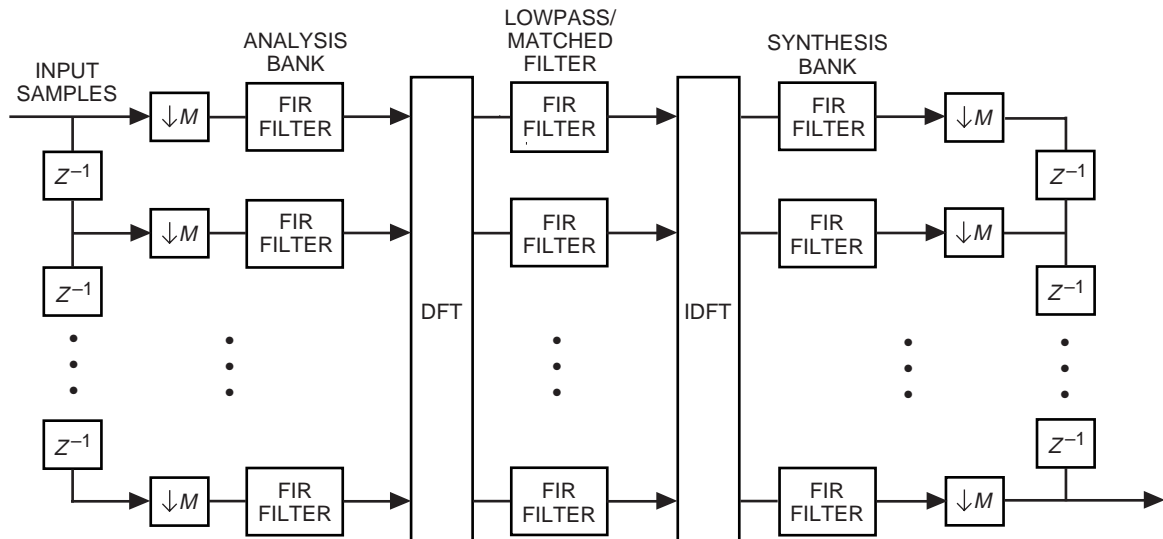


Fig. 1. The multirate filter-bank PRX.

into the system. In Fig. 1, the input signal is parallelized into $2M$ signal paths and decimated by M . The resulting subsequences are filtered by a combination of discrete Fourier transform (DFT)-based analysis and synthesis filter banks separated by subband matched filters. The signal is then interpolated by M and converted back to serial form. Conventional carrier and symbol-timing recovery loops [3] and hard symbol decisions can be formed using the parallel signal paths at the output of the synthesis filter bank. The finite impulse response (FIR) filters that are shown in the analysis and synthesis portions of Fig. 1 are actually filters that result from the polyphase decomposition of analysis and synthesis filters chosen by the designer. Ideally, these analysis and synthesis filters will be designed to have linear phase and to minimize the distortion function for full-band reconstruction [1,2]. The FIR filters that are shown in the lowpass/matched-filtering portion of Fig. 1 are designed by passing the desired matched-filter impulse response through each of the synthesis filters in order to obtain the impulse responses of the subband matched filters. Some of the subbands are disconnected in order to implement the lowpass filtering for rejection of double frequency terms. Thus, the process of designing a receiver using the architecture of [1] involves choosing appropriate analysis and synthesis filters and finding their polyphase decompositions, and then calculating the subband matched filters from the desired matched filter and the synthesis filters.

In [1], it was shown through synchronous simulation that no loss in terms of symbol-error probability is incurred through use of the parallel architecture as compared with the conventional serial receiver. However, after detailed implementation analysis conducted by the Microelectronics Systems Branch at GSFC, the architecture described above was found to be prohibitively high in gate count and, therefore, not a practical candidate for development in the near future. An alternate parallel receiver (APRX) architecture of much lower complexity was proposed. This lower-complexity receiver simply performs the filtering operations in the frequency domain by eliminating the analysis and synthesis filters and using the DFT to compute linear convolution (the “overlap and save” method [4]). The frequency-domain approach is shown to have performance identical to that of the serial receiver and the parallel receiver of [1]. Furthermore, by using this approach, we also can implement symbol-timing correction in the frequency domain, a process that has advantages over time-domain digital timing recovery. One feature of the multirate filter-bank approach that is not preserved in the new architecture is the ability to provide (at the outputs of the first DFT in Fig. 1) discrete-time sequences corresponding to each subband. This feature may be of use in multicarrier communication systems or residual carrier/subcarrier systems.

II. Overview of the APRX Architecture

Prior to entering the digital receiver, an intermediate stage downconverts the RF data signal to an intermediate frequency (IF) appropriate for A/D conversion. A bandpass filter (BPF) is used to reject noise and limit the data bandwidth to prevent aliasing following A/D conversion. The filtered analog signal then is sampled at rate $f_s = 4W$, where W is the transmitted data rate and $2W = B$ is the bandwidth of the antialiasing filter. Note that $f_s = 4W$ is the Nyquist rate for bandpass sampling and that the IF frequency must satisfy $f_c^{IF} = (2k+1)W$, for some integer k [5]. The parallel receiver currently is designed to operate at four samples per symbol; therefore, the maximum antialiasing filter bandwidth must be $B = 2/T_{sym}$, where T_{sym} is the symbol duration. Once we have the digital IF signal, it is digitally mixed with a copy of the IF carrier, the double frequency terms produced by the mixing are rejected by a lowpass filter, and the resulting baseband signal is match filtered so that bit decisions can be made.

The APRX architecture is based upon implementation of the lowpass and matched filters in the frequency domain via the discrete-time Fourier transform (DTFT). In the time domain, matched filtering consists of convolving a time-reversed version of the input received signal with the symbol pulse shape. Since convolution in the time domain corresponds to multiplication in the frequency domain, matched filtering can be performed in the frequency domain by multiplying the Fourier transform of the received signal by the Fourier transform of the pulse signal and then taking the inverse Fourier transform of the product. A lowpass filter can be added to this structure simply by zeroing out components of the product Fourier transform that correspond to the stop band of the filter.

In a digital system, the DFT, which is a sampled version of the DTFT, is used. However, multiplication of two DFT sequences is equivalent to circular convolution of the two time-domain sequences [4]. In linear convolution, one sequence is linearly shifted with respect to the other in order to calculate an output value, whereas in circular convolution, the sequence is circularly shifted. In other words, circular convolution of two finite sequences corresponds to linear convolution of the infinitely periodic extensions of the two sequences. If two sequences of lengths L and M are circularly convolved, the resulting sequence of length $\max(L, M)$ contains $\min(L, M) - 1$ time-aliased values, i.e., the first $\min(L, M) - 1$ values do not agree with those that result from the linear convolution of the two sequences. Therefore, when we take the inverse DFT (IDFT) of the product DFT sequences, only $\max(L, M) - \min(L, M) + 1$ of the values are true linear convolution values. In the APRX, only these unaliased values are output, and the overlap and save method [4] is used to provide all linear convolution values.

The APRX implementation used by GSFC is shown in Fig. 2. The noisy IF signal is filtered and sampled to yield a digital signal with 4 samples per symbol. The digital signal is split into 32 parallel paths, decimated by 16, and passed through a digital mixer bank equal in frequency to that of the sampled IF carrier. The DFT of the 32 data points is then taken and multiplied by the DFT of the matched filter. Lowpass filtering in order to reject double frequency terms from mixing is performed by zeroing out the middle 16 components in the frequency domain, which correspond to the high-frequency terms. Finally, the IDFT is performed, and the middle 16 parallel outputs (which are unaliased and correspond to 4 symbol periods) are used for detection, tracking, etc. This process is repeated once every 16 A/D clock cycles. The 16 points at the output of the IDFT are 16 samples of the convolution integral of the input sequence with the matched-filter impulse response function. Among these 16 samples are 4 peaks that correspond to the matched-filter outputs of 4 symbols. There are a few other points to note here. First of all, by parallelizing into 32 paths, but decimating only by 16, each DFT operates on 16 points from the previous cycle along with 16 new points. This provides the overlap required for calculating all of the linear convolution values. Secondly, by lowpass filtering in the frequency domain via zeroing of high-frequency components, we are limited by the resolution of the DFT. This does not appear to pose a problem, however, and simulation indicates little or no loss due to this implementation. Finally, note that the length of the symbol pulse sequence is only 4, so circular convolution with the 32-point data sequence should result in only 3 aliased points in the IDFT output (neglecting the effect of the frequency-domain lowpass filter). Therefore, we should actually be able to take more than 16 values

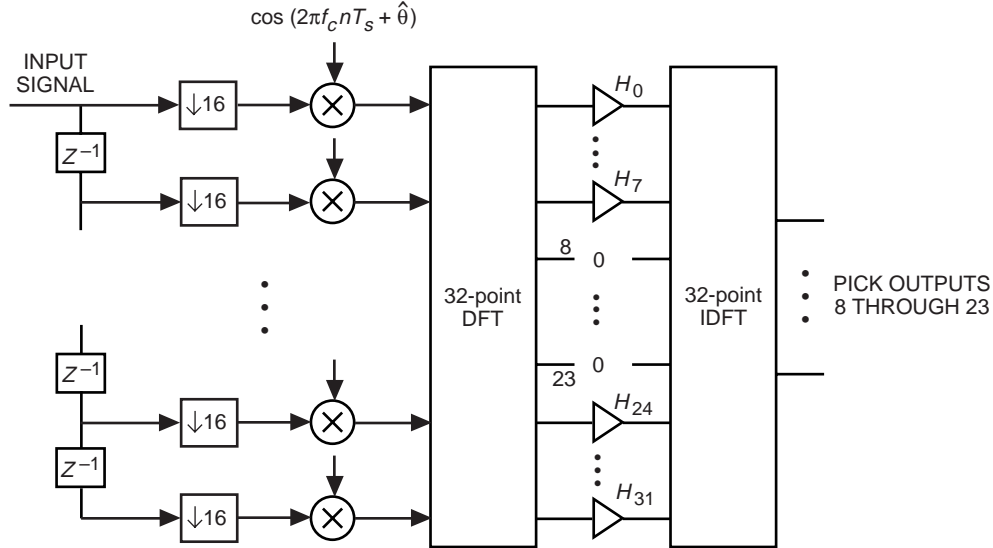


Fig. 2. The frequency-domain APRX.

at the output of the IDFT. However, for implementation convenience, it was decided that only 16 output values, or 4 symbols, would be output at a time.

For nonreturn-to-zero (NRZ) rectangular-shaped pulses, the frequency-domain matched-filter coefficients are found by taking the DFT of the sequence consisting of 4 ones followed by 28 zeros. The zero padding is included to extend the length of the sequence to 32 in order to match the length of the input sequence. Of course, because of the bandpass filtering, the rectangular pulse will be spread out and deformed, so the detection filter actually should be matched to the distorted pulse shape. We have chosen to postpone study of improving the detection filter until a later time, when it will be discussed in another article along with equalization for the mitigation of intersymbol interference (ISI), which also is caused by the bandpass filtering.

III. Error Probabilities From Synchronous Simulation

The performance of the APRX for uncoded BPSK signals was evaluated and compared to the performance of the original PRX and the conventional serial receiver via a software simulation. The simulation block diagram is shown in Fig. 3. Note that downconversion to baseband also is performed inside the APRX, PRX, and serial blocks. The initial simulations were performed with perfect knowledge of carrier phase and symbol timing, and bit-error probabilities were calculated. The continuous binary-phase shift keyed (BPSK) data-modulated carrier was represented digitally using 64 samples per symbol. Following the addition of white Gaussian noise, the signal was filtered by a type I Chebyshev 10th-order bandpass filter with time-bandwidth product $BT = 2$. An ideal A/D converter was simulated by downsampling by 16 in order to produce 4 samples per symbol. The downsampling was performed using the optimum A/D sampling offset, i.e., the location of the first symbol sample relative to the symbol boundary was fixed to yield the lowest error probability. The effect of sampling offset in the conventional serial receiver is studied in [6]. We will examine the sampling offset issue further in this article when we discuss the symbol-timing recovery loop. After downconverting to baseband, the signal was run through each of the three types of receivers, and binary decisions were made on the output, followed by calculation of error probabilities.

The synchronous simulation results are shown in Fig. 4, along with two theoretical curves. The first theoretical curve that we plot is the error probability for an ideal BPSK matched-filter receiver, which is given by

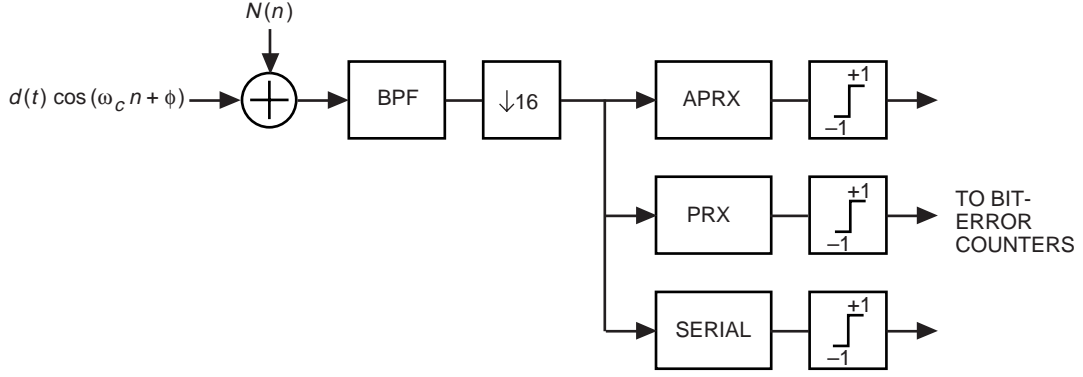


Fig. 3. The simulation setup for receiver comparison.

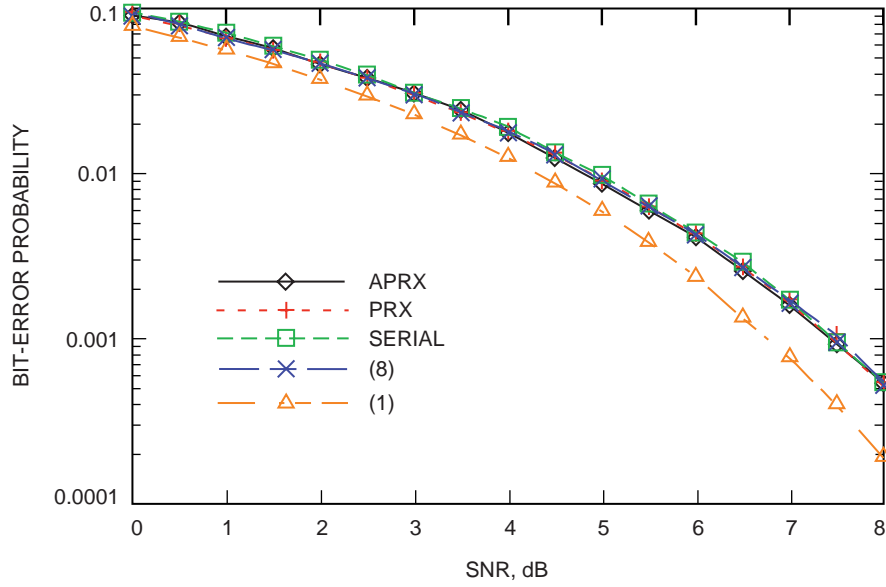


Fig. 4. The error probabilities of APRX, PRX, and serial receivers.

$$P_e[\text{ideal}] = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (1)$$

where E_b/N_0 is the bit signal-to-noise ratio (SNR). The second theoretical curve is the error probability resulting from use of the same receiver when the signal is filtered with an ideal (brick-wall) bandpass filter of time-bandwidth product $BT = 2$, resulting in ISI. We now derive the expression for this error probability.

Let us deal with the equivalent baseband system. The received signal is given by

$$\begin{aligned} r(t) &= s(t) + n(t) \\ &= \sqrt{P} \sum_m a_m p_T(t - mT) + n(t) \end{aligned} \quad (2)$$

where P is the received power, $\{a_m\}$ is the transmitted ± 1 data sequence, $p_T(t)$ is a rectangular pulse of length T , and $n(t)$ is additive white Gaussian noise with power spectral density $N_0/2$. This signal is passed through an ideal lowpass filter with cutoff frequency $W = B/2 = 1/T$. The output of this filter is given by

$$\begin{aligned}
r'(t) &= \int_{-\infty}^{\infty} s(t-\tau)h(\tau)d\tau + \int_{-\infty}^{\infty} n(t-\tau)h(\tau)d\tau \\
&= \int_{-\infty}^{\infty} \sqrt{P} \sum_m a_m p_T(t-\tau-mT) 2W \operatorname{sinc}(2\pi W\tau) d\tau + n'(t) \\
&= \frac{\sqrt{P}}{\pi} \sum_m a_m [\operatorname{Si}(2\pi W(t-mT)) - \operatorname{Si}(2\pi W(t-(m+1)T))] + n'(t)
\end{aligned} \tag{3}$$

where $\operatorname{Si}(t) = \int_0^t \sin(x)/x dx$ and $n'(t) = \int_{-\infty}^{\infty} n(t-\tau)h(\tau)d\tau$. This signal is passed through an integrate-and-dump filter (the matched filter for NRZ pulses), whose output, $X(n)$, is compared to zero in order to determine the value of a_n . Without loss of generality, we consider $X(0)$, which is given by

$$\begin{aligned}
X(0) &= \frac{\sqrt{P}}{\pi} \sum_m a_m \int_0^T [\operatorname{Si}(2\pi W(t-mT)) - \operatorname{Si}(2\pi W(t-(m+1)T))] dt + \int_0^T n'(t) dt \\
&= \frac{\sqrt{P}}{2\pi^2 W} \sum_m a_m \left[\int_{2\pi W|m|T}^{2\pi W(|m|+1)T} \operatorname{Si}(u) du - \int_{2\pi W(|m|-1)T}^{2\pi W|m|T} \operatorname{Si}(u) du \right] + \int_0^T n'(t) dt \\
&= \frac{\sqrt{P}}{\pi^2 W} \left(a_0 \int_0^{2\pi WT} \operatorname{Si}(u) du + \sum_{m=1}^{\infty} \frac{a_{-m} + a_m}{2} \left[\int_{2\pi WmT}^{2\pi W(m+1)T} \operatorname{Si}(u) du - \int_{2\pi W(m-1)T}^{2\pi WmT} \operatorname{Si}(u) du \right] \right) \\
&\quad + \int_0^T n'(t) dt
\end{aligned} \tag{4}$$

The variance of the noise portion of Eq. (4) is calculated as

$$\begin{aligned}
\operatorname{Var} \left[\int_0^T n'(t) dt \right] &= E \left[\left(\int_0^T n'(t) dt \right)^2 \right] = \int_0^T \int_0^T R_{n'}(s-t) ds dt \\
&= \int_0^T \int_0^T N_0 W \operatorname{sinc}(2\pi W(s-t)) ds dt \\
&= \frac{N_0}{2\pi^2 W} \int_0^{2\pi WT} \operatorname{Si}(u) du
\end{aligned} \tag{5}$$

where we have used the fact that the autocorrelation of $n'(t)$ is $R_{n'}(\tau) = N_0 W \text{sinc}(2\pi W\tau)$. The conditional probability of error given the data vector \mathbf{a} can now be calculated using Eqs. (4) and (5) as follows:

$$\begin{aligned} P_{e|\mathbf{a}}(ISI) &= \frac{1}{2}P[X(0) \leq 0|a_0 = 1] + \frac{1}{2}P[X(0) \geq 0|a_0 = -1] \\ &= \frac{1}{2}Q\left(\sqrt{\frac{2E_b}{\pi^2 N_0 WT}}(K + f(\mathbf{a}))\right) + \frac{1}{2}Q\left(\sqrt{\frac{2E_b}{\pi^2 N_0 WT}}(K - f(\mathbf{a}))\right) \end{aligned} \quad (6)$$

Here, $E_b = PT$ is the input energy per bit, $K = \sqrt{\int_0^{2\pi WT} \text{Si}(u)du}$, and

$$f(\mathbf{a}) = \frac{1}{K} \sum_{m=1}^N \frac{a_{-m} + a_m}{2} \left[\int_{2\pi WmT}^{2\pi W(m+1)T} \text{Si}(u)du - \int_{2\pi W(m-1)T}^{2\pi WmT} \text{Si}(u)du \right] \quad (7)$$

where we have restricted the effect of ISI to N bits on either side of a_0 . We then use Eq. (6) to calculate the average error of probability:

$$P_e(ISI) = 2^{-(2N+1)} \sum_{\mathbf{a}} P_{e|\mathbf{a}} \quad (8)$$

Equation (8) is plotted in Fig. 4 for $WT = 1$ and $N = 2$. From this graph, we see that the PRX performs practically identically to the conventional serial receiver (which confirms the results from [1]) and that the APRX also matches these receivers in error probability. We also see that the simulation results closely match the analytical curve from Eq. (8), which represents a loss of about 0.7 dB from the lossless ideal of Eq. (1).

IV. Effect of the Bandpass Antialiasing Filter

The effect of the bandpass antialiasing filter on the performance of the receiver was studied briefly. The antialiasing filter limits the bandwidth of the incoming data signal in order to prevent aliasing when sampling in the A/D conversion stage. Since we are bandpass sampling at the rate $f_s = 4/T$, aliasing will occur if the IF bandwidth B is greater than $2/T$ [5]. On the other hand, filtering the data spectrum causes intersymbol interference. Therefore, there is a trade-off between the effects of aliasing and ISI—a larger BT filter causes more aliasing but less ISI, whereas a smaller BT causes less aliasing and more ISI. Furthermore, the order of the bandpass filter also must be considered, as lower-order filters have a more gradual cutoff (resulting in more aliasing), while higher-order filters have a sharper cutoff (resulting in more ISI).

Simulations were performed in order to determine the optimum filter order and bandwidth. Several type I Chebyshev bandpass filters with 0.1-dB passband ripple were tested, and error probabilities were calculated when $E_b/N_0 = 4.4$ dB (the SNR required to achieve an error probability of approximately 0.01). The simulations were performed using a serial sum-and-dump detection filter (although results also apply to the APRX), with perfect carrier phase reference and timing, and with averaging over A/D sampling offsets. The results are shown in Fig. 5. We see from this plot that, for each different filter order, there is a minimum in the error probability that represents the trade-off point between the effects of aliasing and ISI. The BT location of the minimum increases as the filter order increases, which is expected, since a higher-order filter has a sharper cutoff and, hence, causes less aliasing than does a

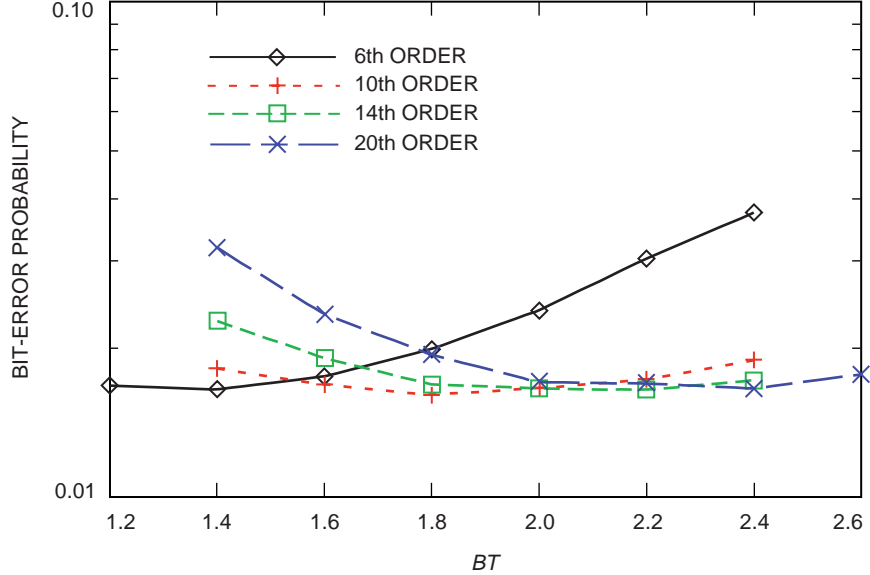


Fig. 5. The error probabilities using type I Chebyshev bandpass filters.

lower-order filter of the same bandwidth. The actual value of this minimum error probability does not vary significantly. Although the simulations were performed for a specific class of infinite impulse response (IIR) filters and a fixed SNR, we expect the trend of the results to hold in general. Based on these results, we performed the subsequent simulations and analysis of the parallel receiver using a 10th-order type I Chebyshev bandpass filter with $BT = 2$, and we recommend that the analog antialiasing filter chosen for implementation have similar specifications.

V. Carrier Phase Tracking Loop

Carrier phase estimation and tracking is performed in the APRX in a standard fashion, using a high SNR Costas loop for suppressed-carrier BPSK signals [3]. A block diagram is shown in Fig. 6. The double lines represent parallel signal paths. At the output of the IDFTs, only the 4 pins containing the peaks of the sum-and-dump operation on 4 symbols are used for phase detection. The hard-limited in-phase output and the quadrature-phase output of the parallel arm filters are multiplied to give the phase error, which may be accumulated and then filtered with an IIR filter to track phase perturbations. This is input to the numerically controlled oscillator, which generates the phase reference used to downconvert the IF signal to baseband (in parallel). The design and analysis of the Costas loop, including specification of loop filter and bandwidth, update rate, etc., follows the general methodology found in references such as [3,7].

Figure 7 is a graph of the carrier phase-error variance as a function of E_b/N_0 as calculated from simulations using the Costas loop. This simulation is of a second-order Costas loop with loop bandwidth 0.001 and update rate equal to one-fourth the symbol rate, with a carrier phase offset introduced for the receiver to track. The theoretical expression for BPSK phase-error variance is [3]

$$\sigma_\phi^2 = \frac{1}{\rho \mathcal{S}_L} = \frac{N_0 B_l}{\text{Perf}^2(\sqrt{E_b/N_0})} \quad (9)$$

where P is the signal power and B_l is the loop bandwidth. The term $\rho = P/(N_0 B_l)$ is the SNR in the phase-locked loop bandwidth, and $\mathcal{S}_L = \text{erf}^2(\sqrt{E_b/N_0})$ is the squaring loss [3,7]. We see from Fig. 7 that

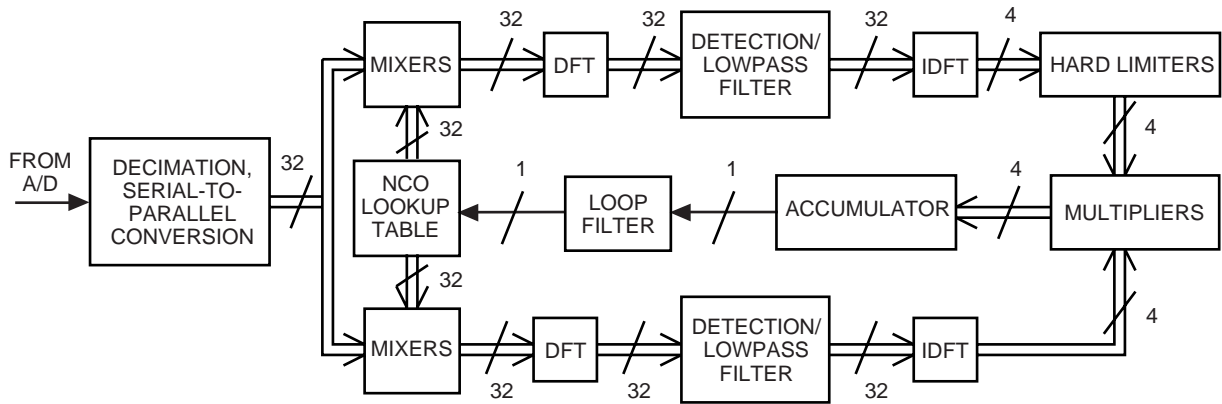


Fig. 6. The Costas loop for carrier phase tracking.

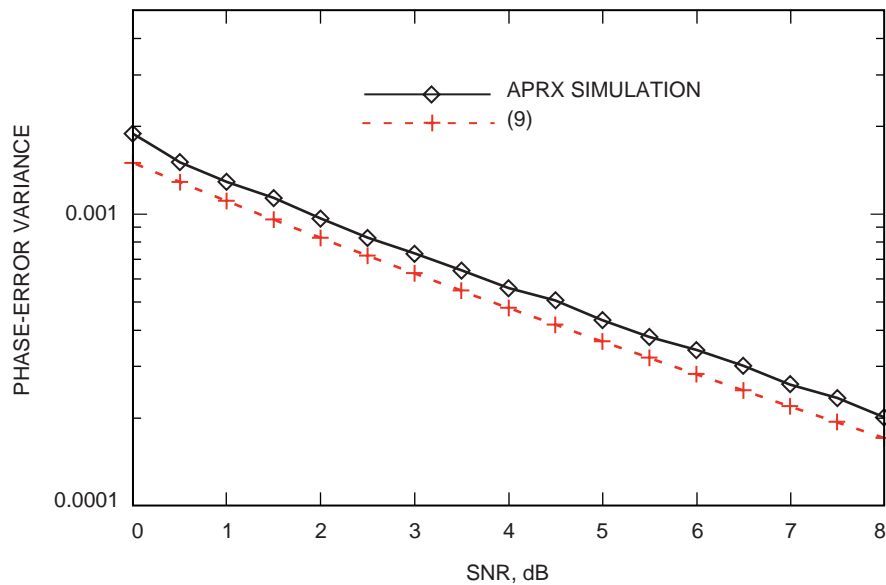


Fig. 7. The phase-error variance for the BPSK Costas loop.

the phase-error variance from simulation differs from theory by about 0.7 dB, an amount that is equal to the loss caused by bandpass filtering that is shown in Fig. 4.

VI. Symbol-Timing Recovery Loop

In order to implement the detection filtering of the baseband signal, the data symbol boundaries need to be known. In a serial digital receiver, an accurate estimate of the symbol phase is needed to adjust the symbol clock so that the sum-and-dump operation is performed on the samples that correspond to the current symbol. For NRZ data, one method of deriving the symbol phase for NRZ symbols is to use the data-transition tracking loop (DTTL) [3,8,9]. Figure 8 shows the serial digital DTTL. The upper branch of this loop integrates across one symbol duration in order to provide an estimate of the polarity of the present symbol and compares it to the polarity of the previous symbol to indicate the occurrence of a data transition. The lower branch estimates the timing error by integrating across a transition in order to measure the deviation from zero. The product of the two branches yields the symbol phase error with the appropriate sign. This phase error is filtered and used to control the numerically controlled oscillator

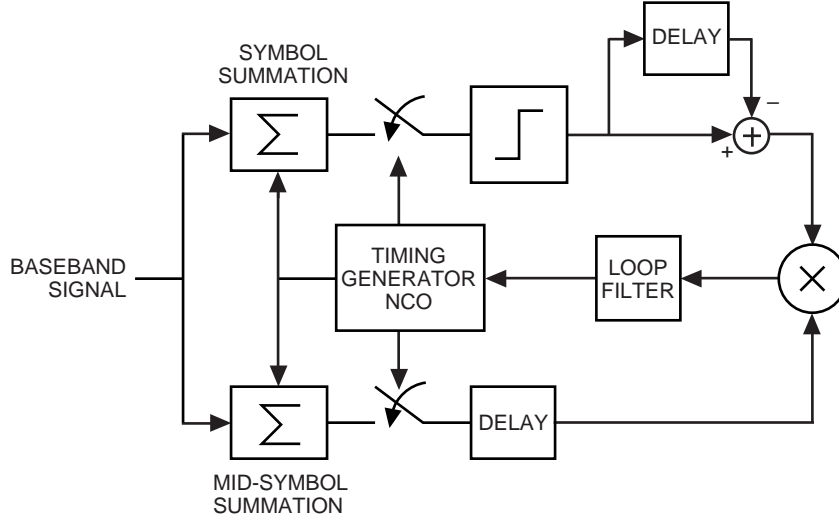


Fig. 8. The digital DTTL for NRZ symbols.

that clocks the sum-and-dump interval. Note that there is an inherently finite resolution to the digital DTTL due to the fact that symbol phase errors can be corrected only to the extent that samples may be included or excluded from the current symbol. In other words, there is a range of undetectable phase errors when using the digital DTTL as described above. Of course, the more samples per symbol, the higher the resolution of the digital DTTL and the closer the digital DTTL is to the analog version.

In the APRX, the outputs of the in-phase and mid-phase integrators are to be found as specific pins in the block output of the inverse DFT block of Fig. 2. One possible implementation of the DTTL in the APRX would involve multiplying the in-phase and mid-phase pin outputs, adding them together, and then using the filtered result to control a commutator that closes the loop by deciding which output pins from the inverse DFT correspond to the correct in-phase and mid-phase integrator values. This implementation is shown in Fig. 9. The performance of this loop should be identical to that of a similarly parameterized serial digital DTTL [8,9] and will have the same limited phase resolution.

A more natural implementation of the DTTL in the APRX follows from utilizing the frequency-domain structure. This implementation is shown in Fig. 10. Noting that a time delay corresponds to a phase shift in the frequency domain, we may correct the timing by inserting phase correctors after performing the matched filtering in the frequency domain. This phase correction will have the effect of shifting the desired in-phase and mid-phase integrator values to a fixed set of selected pins at the output of the inverse DFT. The frequency-domain DTTL is desirable from an implementation standpoint because the required output lines from the inverse DFT are fixed and a commutator routing switch is not needed. More importantly, frequency-domain phase correction allows us to effectively solve the problem caused by A/D sampling offset.

At this point, we find it relevant to discuss the issue of the A/D sampling offset upon receiver performance. The effect of few samples per symbol and varying sampling offsets is documented in [6]. In an analog receiver, once there is perfect symbol synchronization, an ideal matched filter detects the k th rectangular pulse data symbol by integrating the baseband signal from kT_{sym} to $(k+1)T_{sym}$, yielding the maximum possible symbol SNR. In a digital receiver, the integration operation is replaced by a summation over the samples of the desired symbol. The i th sample of the k th symbol occurs at time $kT_{sym} + iT_s + \tau$, where τ is the time offset of the first symbol sample with respect to the beginning of the pulse (see Fig. 11). Clearly, with finite bandwidth causing a distortion in pulse shape, the value of τ will affect the amplitudes of the symbol samples and, hence, the output symbol SNR. In [6], it was shown that,

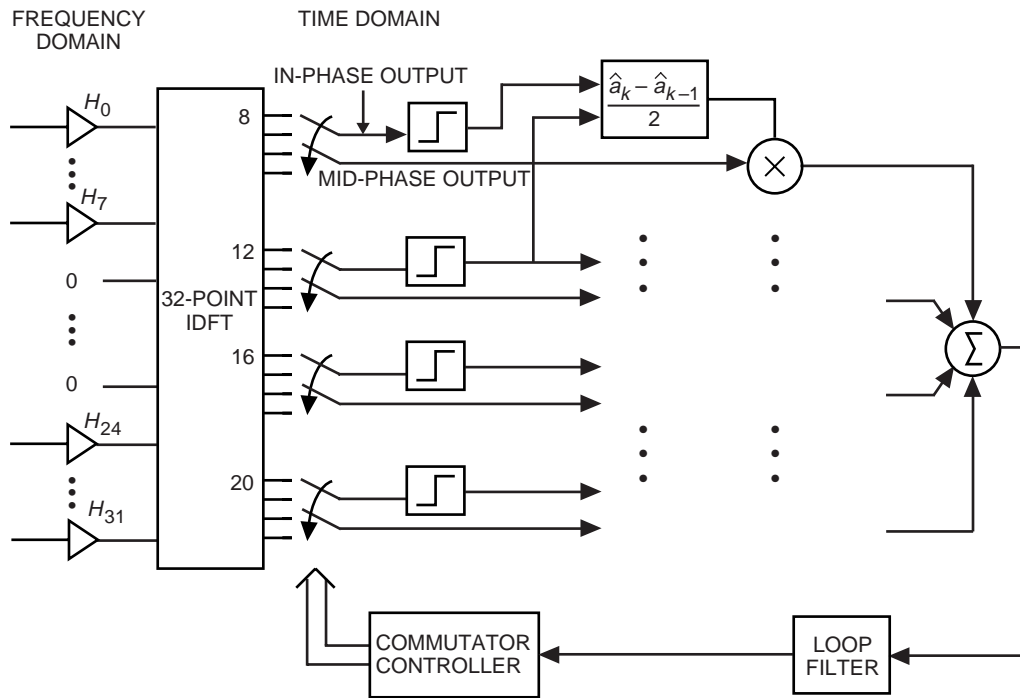


Fig. 9. Implementation of the parallel DTTL in the time domain.

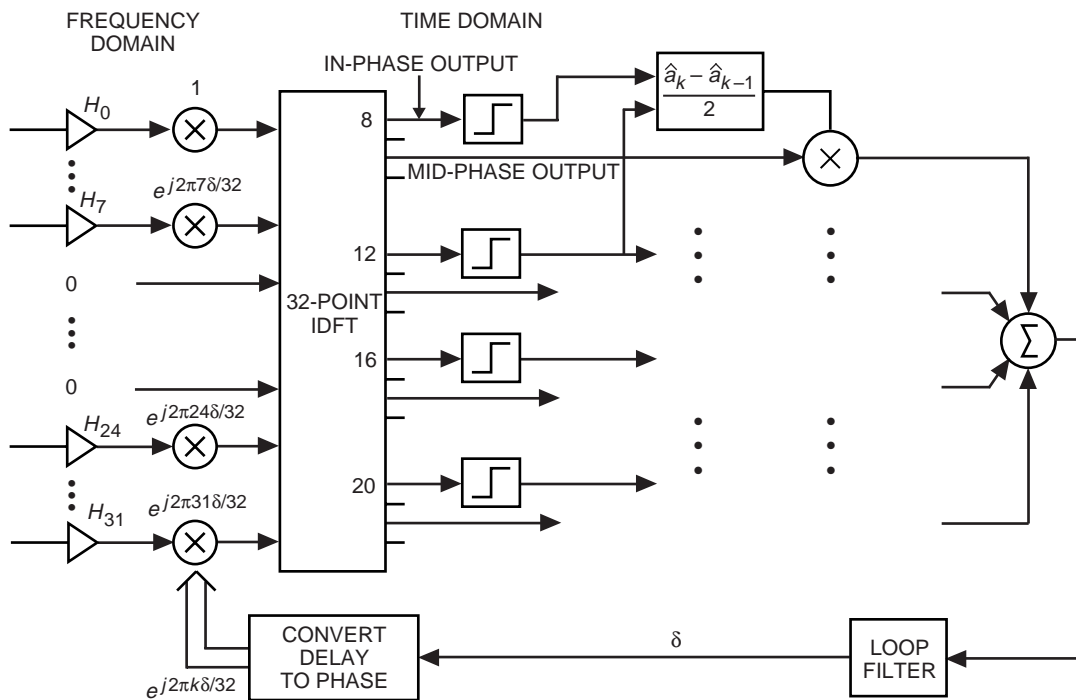


Fig. 10. Implementation of the parallel DTTL in the frequency domain.

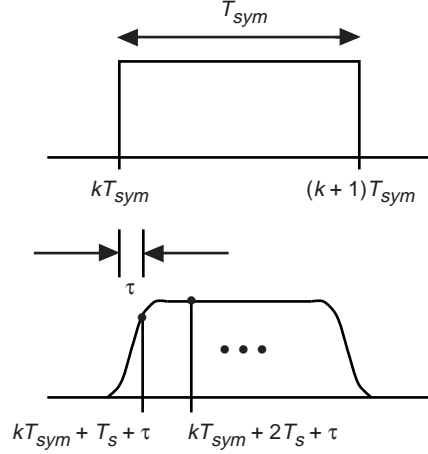


Fig. 11. Offset in sampling.

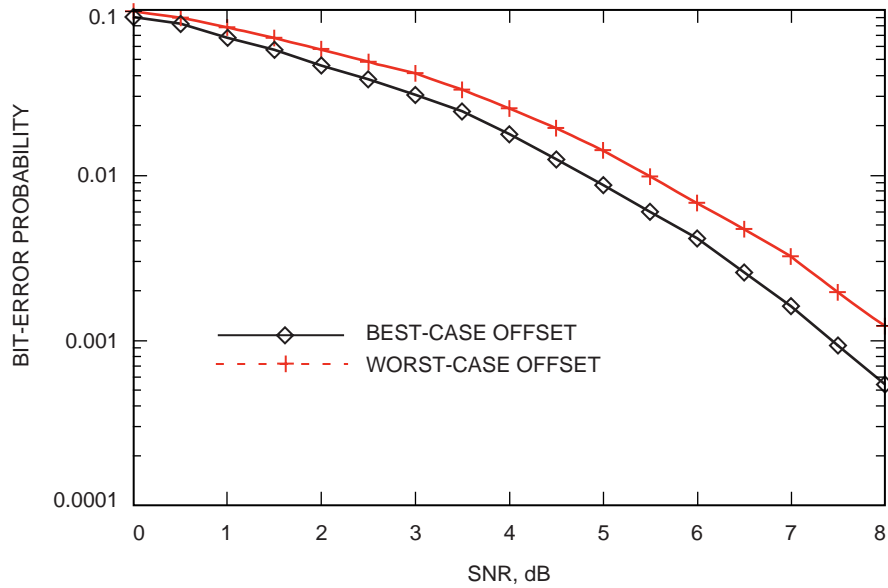


Fig. 12. The simulated bit-error probabilities for two different sampling offsets.

for a serial baseband receiver at a sampling rate of four times the symbol rate, the output SNR varies by 1 dB as the sampling offset varies with respect to the symbol boundaries. The best case is when the zero crossing between symbols occurs midway between adjacent samples. The worst case occurs when the zero crossing is one of the symbol samples. From synchronous passband simulations of the APRX, we find that changing the sampling offset from best case to worst case causes a loss of 0.8 to 1.0 dB. Figure 12 shows the simulated bit-error probabilities versus input SNR for the best- and worst-case sampling offsets. Two possible remedies for alleviating this loss have been suggested. One is to synchronize the sampling clock with the symbol clock so that the sampling offset is made optimal. This may not always be desirable, e.g., the ultrastable clock used to synchronize the sampling clock may be needed for ranging applications and, hence, may not be manipulated [1]. A second solution is to use a weighted integrate-and-dump detection filter in which the minimum mean-squared error criterion is used to derive coefficients for the detection filter. This equalization method is described in [10] and leads to a different set of filter weights for each sampling offset value. The appropriate detection filter weights would have to be calculated (or loaded in from a lookup table, resulting in finite resolution) after an estimate of the sampling offset is made via

the symbol synchronization loop. This process would lead to a time-varying detection filter that changes with the symbol phase output from the symbol synchronization loop. The process of incorporating this technique into the multirate filter-bank parallel receiver is described in [1].

The solution to dealing with the sampling-offset problem arises quite naturally when the frequency-domain architecture of the APRX is used, and is much easier to implement than the solution proposed in [1]. In the time-domain implementation of Fig. 9, the estimated symbol delay, δ , which may not be an integer, is effectively truncated to an integer number of samples, since all the timing loop is doing in that case is to change the pin numbers used for deriving the data. On the other hand, in Fig. 10, the phase correction, $e^{2\pi k\delta/32}$, that is applied to each frequency-domain component, k , adjusts not only for the integer number of samples that the symbols are delayed by but also for the fractional number of samples, which corresponds to the sampling offset. In other words, multiplying the N -point discrete Fourier transform of a sequence by $e^{2\pi k\delta/N}$ is equivalent to sampling a delayed version of the continuous time signal. The relationship between the two signals is illustrated in Fig. 13 and is proven as follows.

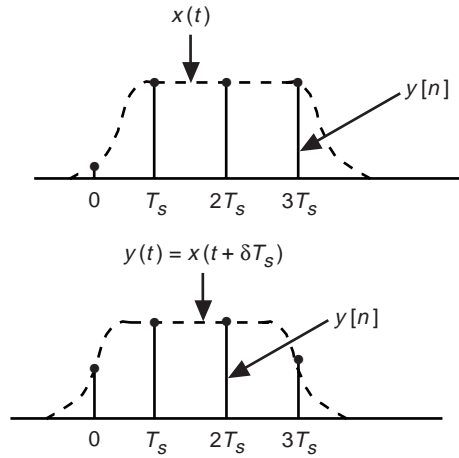


Fig. 13. Frequency-domain timing correction produces a sequence delayed by a fraction of a sampling interval.

Let $x(t)$ be a band-limited continuous time signal with Fourier transform $X_c(\Omega)$. If $x(t)$ is shifted by the amount $-\delta T_s$, then the Fourier transform of $y(t) = x(t + \delta T_s)$ is given by $Y_c(\Omega) = X_c(\Omega)e^{j\Omega\delta T_s}$. If we now sample $y(t)$ at frequency $1/T_s$ (such that the Nyquist criterion is satisfied), the resulting sequence $y[n] = y(nT_s)$ has the discrete-time Fourier transform $Y_d(e^{j\omega})$ given by

$$\begin{aligned} Y_d(e^{j\omega}) &= \frac{1}{T_s} Y_c\left(\frac{\omega}{T_s}\right) = \frac{1}{T_s} X_c\left(\frac{\omega}{T_s}\right) e^{j\omega\delta}, \quad |\omega| < \pi \\ &= X_d(e^{j\omega}) e^{j\omega\delta}, \quad |\omega| < \pi \end{aligned}$$

since $X_d(e^{j\omega}) = X_c(\omega/T_s)/T_s$ is the discrete-time Fourier transform of $x[n] = x(nT_s)$, the sequence obtained by sampling $x(t)$ at rate $1/T_s$. Now the N -point discrete Fourier transform is obtained by sampling the discrete-time Fourier transform at points $\omega = 2\pi k/N$, for $-N/2 \leq k \leq N/2 - 1$ (assuming that N is even). Therefore, if we denote the DFT of $y[n]$ as $Y(k)$, we have

$$\begin{aligned}
Y(k) &= Y_d(e^{j2\pi k/N}) = X_d(e^{j2\pi k/N})e^{j2\pi k\delta/N}, & -\frac{N}{2} \leq k \leq \frac{N}{2} - 1 \\
&= X(k)e^{j2\pi k\delta/N}, & -\frac{N}{2} \leq k \leq \frac{N}{2} - 1
\end{aligned}$$

where $X(k)$ is the DFT of $x[n] = x(nT_s)$. From this last step, we see that the DFT obtained by multiplying the input sequence DFT by $e^{j2\pi k\delta/N}$ is the same as the DFT of the input sequence delayed by δ .

Figures 14 and 15 show the symbol phase-error variance and bit-error probability obtained from simulations using the frequency-domain DTTL. The symbol tracking loop is second order with loop bandwidth 0.001. Simulations were run using the same best- and worst-case offsets that yielded the curves in Fig. 12. In Fig. 14, the simulated phase-error variances are compared to the theoretical phase-error variance for an analog DTTL. If λ denotes the normalized symbol phase error, then its analytical variance is [8,9]

$$\sigma_\lambda^2 = \frac{B_l T_{sym} \mathcal{S}_L}{2(E_b/N_0)} \tag{10}$$

where B_l is the symbol loop bandwidth and \mathcal{S}_L is the symbol loop squaring loss, equal to

$$\mathcal{S}_L = \frac{1 + 0.5E_b/N_0 - 0.5 \left((1/\sqrt{\pi}) e^{-E_b/N_0} + \sqrt{E_b/N_0} \operatorname{erf} \left(\sqrt{E_b/N_0} \right) \right)^2}{\left(\operatorname{erf} \left(\sqrt{E_b/N_0} \right) - 0.5\sqrt{E_b/(N_0\pi)} e^{-E_b/N_0} \right)^2} \tag{11}$$

In Fig. 14, we see some loss in phase-error variance of the simulated DTTL. This is believed to be caused by distortion in the pulse shape caused by ISI, which would affect estimation of the symbol phase.

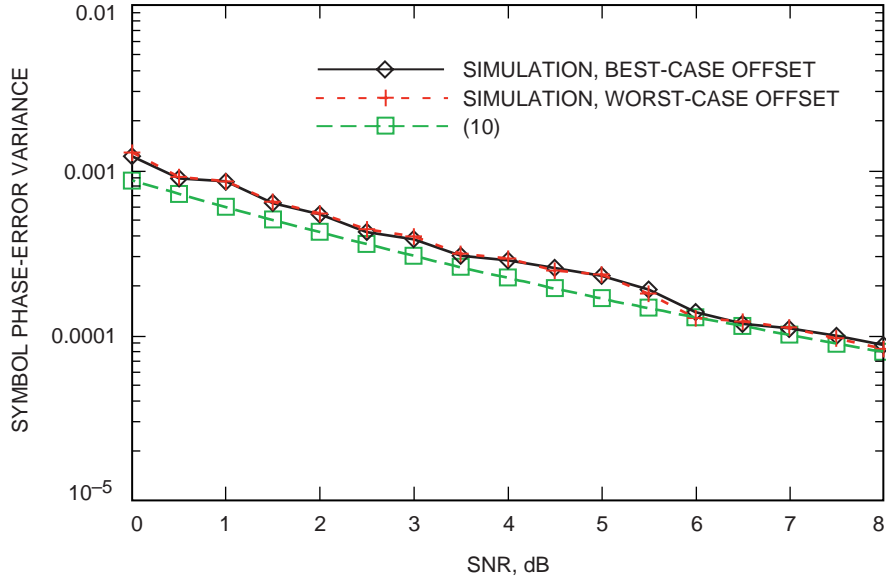


Fig. 14. The simulated symbol phase-error variances for best- and worst-case sampling offsets using frequency-domain DTTL versus theoretical curve.

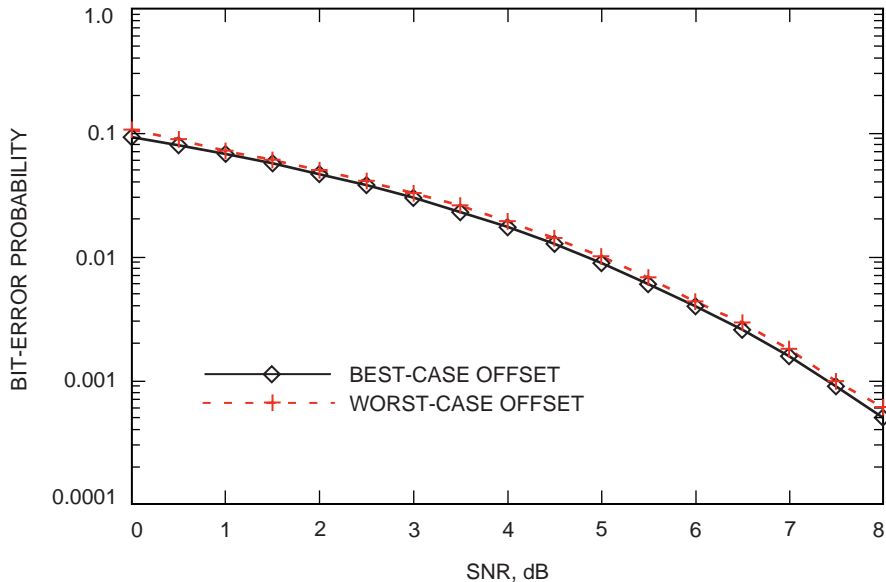


Fig. 15. The simulated error probabilities for best- and worst-case sampling offsets using frequency-domain DTTL.

Figure 15 should be contrasted to Fig. 12 in order to see the effectiveness of the frequency-domain timing recovery loop in correcting for sampling offset. The 0.8-dB difference has been reduced to about 0.2 dB. In theory, we expect the two different sampling offsets to result in the same bit-error rate curve. The 0.2-dB difference that still exists may be due to distortions in the pulse shape and/or aliasing caused by the signal not being truly bandlimited.

VII. Conclusions and Further Work

The alternate parallel receiver (APRX) architecture that is being developed by JPL and GSFC was presented. It was shown that the frequency-domain implementation of the APRX is equivalent to the conventional serial receiver and the original multirate filter-bank parallel receiver in terms of error probability. The trade-off between antialiasing filter bandwidth and order was analyzed. The implementation of carrier phase and symbol-timing recovery loops was addressed, and simulations showed performance in line with expected results from theory. The issue of A/D sampling offset was investigated, and it was shown that the loss caused by nonideal sampling offset can be compensated for in a very simple, low-complexity manner by using a frequency-domain symbol-timing correction scheme.

The work documented in this article indicates that the parallel frequency-domain receiver is an excellent candidate for high-rate communications. It is conceptually simple, is significantly less complex than the multirate filter-bank receiver, and suffers no loss relative to the conventional implementations. The next major focus of work shall be the incorporation of equalization filters into the parallel architecture. Other work in progress indicates that the APRX architecture may be modified easily for modulation formats other than BPSK. The simplest extension is for quadrature-phase shift keyed (QPSK) signaling. Simulations for QPSK signal tracking and detection have already been performed, again yielding results that show no loss from comparable serial receivers. Preliminary work shows that unbalanced QPSK, which is a signaling scheme that creates implementation challenges due to unequal data rates in the two channels, also may be received by using two parallel receivers. Other modulation schemes to be addressed in the future include 8-ary phase shift keying (8-PSK) and Gaussian minimum shift keying (GMSK).

References

- [1] R. Sadr, P. P. Vaidyanathan, D. Raphaeli, and S. Hinedi, *Parallel Digital Modem Using Multirate Digital Filter Banks*, JPL Publication 94-20, Jet Propulsion Laboratory, Pasadena, California, August 1994.
- [2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1993.
- [3] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, New York: Dover Publications, 1973.
- [4] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1989.
- [5] R. M. Gagliardi, *Introduction to Communications Engineering*, New York: John Wiley and Sons, 1988.
- [6] R. Sadr and W. J. Hurd, "Detection of Signals by the Digital Integrate-and-Dump Filter With Offset Sampling," *The Telecommunications and Data Acquisition Progress Report 42-91, July-September 1987*, Jet Propulsion Laboratory, Pasadena, California, pp. 158–173, November 15, 1987.
- [7] S. Aguire and W. J. Hurd, "Design and Performance of Sampled Data Loops for Carrier and Subcarrier Tracking," *The Telecommunications and Data Acquisition Progress Report 42-79, July-September 1984*, Jet Propulsion Laboratory, Pasadena, California, pp. 81–95, November 15, 1984.
- [8] W. C. Lindsey and T. O. Anderson, "Digital-Data Transition Tracking Loop," *International Telemetry Conference*, Los Angeles, California, pp. 259–271, October 1968.
- [9] M. K. Simon, "Optimization of the Performance of a Digital Data Transition Tracking Loop," *IEEE Transactions on Communications*, vol. COM-18, no. 3, pp. 686–687, October 1970.
- [10] R. Sadr, "Detection of Signals by Weighted Integrate-and-Dump Filter," *The Telecommunications and Data Acquisition Progress Report 42-91, July-September 1987*, Jet Propulsion Laboratory, Pasadena, California, pp. 174–185, November 15, 1987.