A Functional Description of the Advanced Receiver

S. Hinedi
Communications Systems Research Section

This article describes the breadboard Advanced Receiver II (ARX II) that is currently being built for future use in NASA's Deep Space Network (DSN). The hybrid analog/digital receiver performs multiple functions including carrier, subcarrier, and symbol synchronization. Tracking can be achieved for residual, suppressed, or hybrid carriers and for both sinusoidal and square-wave subcarriers. Other functions such as time-tagged Doppler extraction and monitor/control are also discussed, including acquisition algorithms and lock-detection schemes. System requirements are specified and a functional description of the ARX II is presented. The various digital signal-processing algorithms used are also discussed and illustrated with block diagrams.

I. Introduction

The Advanced Receiver (ARX) [1] is a system that has been under development for future use in the Deep Space Network (DSN) [2]. It is intended to replace various receivers currently in use to demodulate and process signals from deep-space spacecraft. The ARX has been under development for several years and has undergone several tests [3–6] and modifications to improve its performance. It has also been used as a testbed for high-dynamic frequency tracking for Global Positioning System (GPS) applications [7–9].

Two versions of the ARX have been built, ARX I and ARX II; they differ mainly in the processing strategy used. In ARX I, the sampling clock was driven by the symbol-synchronization loop and hence, an integer number of samples per symbol was obtained. Since the sampling period was not fixed, the time base varied. As a result, time-tagged digital Doppler extraction was unfeasible. In ARX II, the sampling clock was fixed, resulting in the possibility of a noninteger number of samples per symbol. However, there is a fixed time base and the system can be easily synchronized to a station clock. This article concerns itself only with ARX II, as its architecture was chosen as the final candidate for a possible future implementation of a Block V receiver. The system requirements and a functional description of the operating principles of the receiver are given. Performance trade-off issues as a function of complexity are discussed and design decisions are clarified.

Before discussing the design of the ARX II, it is important to understand the different requirements of the system and the constraints they might impose on its architecture. First, functional and performance requirements will be discussed. Before doing that however, it is worthwhile to make sure that the reader understands what is meant here by the term "receiver." The received signal, which may be at L (1628–1708 MHz), S (2200–2300 MHz) or X (8400–8500 MHz) band, gets downconverted in an open-loop fashion to an intermediate frequency (IF) in the

200–400-MHz range. Following that downconversion, all operations including further downconversion to more appropriate frequencies, carrier demodulation, and symbol detection are all included in what is referred to here as the "receiver." Therefore, the input to the receiver is a signal in the 200–400-MHz range and one of its outputs consists of the detected soft symbols. These symbols are later decoded in several stages (Reed-Solomon as well as convolutional or Viterbi decoding) to produce bits. That decoding is not part of what is referred to here as the "receiver." Other decoding schemes for source decompression might also be present but are not considered as part of the ARX II. Another important output of the ARX II is a time-tagged received phase cycle count, which is eventually used in other subsystems to perform ranging and navigation functions.

This article is divided into six sections. Section II discusses system requirements, both functional and technical. The architecture of the receiver is the topic of Section III, along with the interface issues that affect and limit the ARX II design. Section IV deals with the tracking algorithms and the other signal-processing schemes employed to perform the various tracking tasks. Implementation versus performance trade-off issues are discussed in Section V and the conclusions are presented in Section VI.

II. Receiver Requirements

In order to assess the performance of the system, a set of performance requirements is needed to serve as a reference against which a measured set of data can be compared. Secondly, a more basic set of functional requirements is needed to define the operations required from the receiver: What is it supposed to do? What are its inputs and outputs? Finally, interface issues need to be defined for the receiver to operate with other subsystems. The latter issue is probably the most important because it typically dictates certain features in the receiver. These issues, as well as other matters such as size, power consumption, etc., constitute the set of requirements and may result in constraints that the system engineer should be aware of prior to starting the design.

Functionally, the receiver must perform two fundamental tasks: Doppler extraction and telemetry processing. Doppler extraction involves tracking a pure tone in the presence of time-varying Doppler and recording its phase, time tagged for further off-line processing. Telemetry requires demodulating a signal whose spectrum is data-rate dependent and producing soft detected symbols to be used by other subsystems (for example, a Reed-Solomon decoder). Note that while Doppler extraction requires narrow bandwidth, the telemetry task needs wide bandwidth to accommodate the highest data rate foreseen. The latter task involves demodulating the symbols of a received signal that can be either binary phase shift keyed (BPSK), quadrature phase shift keyed (QPSK), offset QPSK (OQPSK), or minimum shift keyed (MSK). In OQPSK, the in-phase and quadrature baseband pulses are shifted with respect to each other by half a symbol period while in MSK, sinusoidal pulse shaping is employed instead of rectangular pulses. For BPSK, the signal might have a residual carrier component, in which case the data spectrum will be shifted away using a square or a sinusoidal subcarrier. In the case of a fully suppressed BPSK carrier, the subcarrier may or may not exist. The data itself can be either non-return-to-zero (NRZ) or Manchester (bi-phase) encoded. In communication terms, the receiver needs to perform carrier tracking, whether residual or suppressed, subcarrier tracking, and symbol synchronization for the different modulation formats listed above. Various lock detectors are required to monitor the loops and to help automate the receiver's operation. Furthermore, a friendly receiver/user interface is highly desirable to facilitate operations.

Now that the system's functionality has been defined, performance-related issues need to be addressed to specify the receiver's requirements. In order to support current and future (near-term) missions, it has been decided that a predetection bandwidth of 16 MHz (3-dB bandwidth) is more than adequate. The latter bounds the highest data rate that can be processed and is roughly half the processing (or effective sampling) rate that can be accommodated with current technology in a digital implementation.

The dynamic range of the receiver should be about 73 dB to handle carrier-to-noise ratios (CNRs) from 0 to 73 dB-Hz (CNR is defined as the ratio of carrier power P_C to the one-sided noise spectral density level N_0). Moreover, the data-rate capability should be from 8 symbols-per-second (SPS) to 6.6 MSPS with direct BPSK modulation and from 8 SPS to 700 kSPS in the presence of a subcarrier, whose frequency lies in the 100-Hz to 2-MHz range. The modulation index, the parameter that delegates power to the data, varies between 0 deg and 90 deg, the first corresponding to a pure tone, the latter to a suppressed carrier. Radio (symbol signal-to-noise ratio) loss due to the carrier loop should be less than 0.3 dB over data rates from 8 SPS to 6.6 MSPS. Similar bounds exist on the loss due to subcarrier and symbol-synchronization loops over the data rates from 8 SPS up to 1 MSPS.

These bounds, from a designer's point of view, determine the range of loop bandwidths that need to be han-

dled. The receiver should be able to acquire the carrier within 10 sec at 8 dB above loop threshold and acquire the subcarrier frequency and symbol epoch within 20 sec for symbol rates above 200 SPS and symbol signal-to-noise ratios (SNR) greater than −2.5 dB; otherwise, acquisition should occur within 6000 symbols. The predict uncertainty is about ± 10 kHz in carrier frequency, ± 1 kHz in subcarrier frequency, and ± 20 SPS for data rates above 20 kSPS or ± (symbol rate/2000) for data rates below 20 kSPS.

Furthermore, the receiver should monitor its status by measuring and reporting E_S/N_0 (symbol energy-to-noise ratio) to within ± 0.1 dB, the carrier and subcarrier frequencies to within ± 0.1 Hz, the symbol rate to within ± 0.1 SPS, and the carrier static phase error to within ± 1 deg. Moreover, the receiver should report in-lock/out-of-lock status for all loops; perform sideband aiding, Doppler-rate aiding, fast Fourier transforms (FFTs) for acquisition as well as for monitor/control, and real-time loop bandwidth optimization; and implement loop types I, II, and III for all the various loops with different update rates that range from 100 Hz to 1 kHz.

The last set of requirements to be discussed is related to the interface issues that need to be met for the new system to operate with existing equipment. For example, the receiver will be supplied only 1-MHz, 5-MHz, and 10-MHz references from the Frequency and Timing Subsystem (FTS). Any additional references that are needed by the receiver must be generated internally. Furthermore, to supply time-tagged carrier phase cycle count, a time code translator (TCT) is needed to read the time from the station's time code generator (TCG); this ensures that the receiver's time is synchronized with the station time. Also, the receiver needs to interface with the sequential ranging assembly (SRA) subsystem by providing it with an analog 10-MHz locked IF signal. The SRA itself will undergo upgrading in the near future and it will then only accept digital baseband signals. Therefore, the receiver should be able to provide both analog IF and digital baseband signals to interface with existing and future SRAs.

Only the key elements of the requirements that directly influence the design have been discussed above since a complete specification would be lengthy and inappropriate for the purpose of this article.

III. The Architecture of the Advanced Receiver

Both ARXs were implemented in a hybrid analog/digital fashion to offer flexibility and low cost. The front end, which performs mainly filtering and downconversion operations, was implemented using analog hardware, while the "signal-processing part" was built using the latest commercially available digital hardware.

Figure 1 depicts a top-level block diagram of the receiver with digital loop closure. The first stage of conversion from L-, S-, or X-band to the 200–400-MHz range is accomplished by an external subsystem (IF Distribution Subsystem) and is not part of the receiver. Its output, which constitutes the input to the ARX II, is further downconverted to a more appropriate frequency (70 MHz) for fine-tuned filtering and then to a different lower frequency (10 MHz) for sampling. Because of the high data-rate requirement of 6.6 MSPS, the 10-MHz IF signal is sampled at roughly 40 MHz, digitally in-phase and quadrature (I and Q) mixed to baseband, and filtered with the half-band filters (HBFs) to remove the double-frequency terms. The baseband I and Q signals are then processed to digitally perform carrier, subcarrier, and symbol synchronization, respectively. Note that the additional downconversion to 10 MHz could be avoided by undersampling the 70-MHz IF signal at 40 MHz. This would not violate Nyquist sampling because the signal is band-limited to 20 MHz. However, this would require higher quality analog filters (than their counterparts at 10 MHz) to reduce the potential aliasing.

Both analog and digital loop closures were implemented to satisfy the interface to the current and future SRAs. The closures are depicted in Fig. 2 along with the appropriate frequencies. In the analog loop closure, the downconverted 70-MHz signal is mixed with an 80-MHz phase-locked signal to produce the 10-MHz signal required by the current SRA. The analog-to-digital (A/D) converter operates on the 10-MHz IF locked signal to produce 8-bit samples at 40 MHz. The 10-MHz IF is then digitally removed using a look-up table. After filtering to remove the sum frequency term, the resulting signal has a 10-MHz effective bandwidth and a 40-MHz processing rate. A decimation by two is performed to reduce the processing rate to the required 20-MHz rate specified by Nyquist sampling. The Q samples are then accumulated to further reduce the rate to the residual-carrier loop update rate to enable a software implementation of the loop filter. The filter outputs a frequency-error estimate at the loop update rate and adjusts the nominal frequency (4 MHz) of the numerically controlled oscillator (NCO). Note that the NCO's phase is actually changing at the much higher rate of 20 MHz. This feeds the digital-to-analog (D/A) converter whose output is a 4-MHz sine wave. The latter is subsequently mixed with a fixed 84-MHz signal to produce the 80-MHz analog signal that closes the loop.

The operating principle of the digital loop closure is identical except that the 80-MHz signal is now a fixed refer-

ence and an NCO running at 40 MIIz with a 10-MHz nominal frequency is now used instead of a look-up table. In either case, A/D conversion is performed at the 10-MHz IF signal rather than at baseband to avoid the potential problem of DC bias generated by the A/D converter. Moreover, the sampling frequency can be changed to 39.8 MHz to avoid any subharmonics of the sampling clock that might be generated by the A/D in the signal band of interest. A software command can switch the system between analog and digital closure without losing carrier lock. However, the sampling frequency cannot be changed on the fly.

A. Signal Model

A functional block diagram of the ARX II is shown in Fig. 3 for the digital-loop closure case. Depicted are the residual carrier tracking phase-locked loop (PLL), the suppressed carrier tracking loops for both BPSK and QPSK (Costas loop and cross-over Costas loop respectively), the subcarrier loop (also a Costas loop), the symbol-synchronization loop (data-transition tracking loop [DTTL]), lock detectors for the residual carrier loop and for the various Costas loops, sideband aiding, a symbol SNR estimator (split-symbol moments estimator [SSME]), and weighted integrate-and-dump filters (WIDFs) for symbol detection. A total-power automatic gain control (AGC) circuit is present to ensure that the signal is within the dynamic range of the A/D converter. The basic operating principles of each loop will be discussed briefly and the reader is referred to the appropriate references for additional information.

The received signal at the output of the AGC is given by

$$r(t) = \sqrt{2P}\sin[\omega_i t + \Delta D(t) + \theta_c] + n(t) \tag{1}$$

where P is the average signal power, ω_i the IF frequency (70 MHz) in rad/sec, θ_c the carrier phase in rad, Δ the modulation index, $D(t) = d(t)\mathrm{Sin}(\omega_{sc}t + \theta_{sc})$ with $\mathrm{Sin}(x) = \mathrm{sgn}(\sin(x))$ for a square wave ($\mathrm{sgn}(x)$ denotes the "signum" function) and $\mathrm{Sin}(x) = \sin(x)$ for a sine-wave subcarrier, ω_{sc} the subcarrier frequency in rad/sec, θ_{sc} the subcarrier phase in rad, and $d(t)$ the data modulation, i.e.,

$$d(t) = \sum_{k=-\infty}^{+\infty} a_k p(t - kT) \tag{2}$$

with $a_k = \pm 1$ equally likely and $p(t)$ the baseband NRZ or Manchester pulse limited to T seconds. The narrow-band noise $n(t)$ can be written as

$$n(t) = \sqrt{2}n_c(t)\cos(\omega_i t + \theta_c) - \sqrt{2}n_s(t)\sin(\omega_i t + \theta_c) \tag{3}$$

where $n_c(t)$ and $n_s(t)$ are statistically independent stationary band-limited white Gaussian noise processes with one-sided spectral density N_0 watts/Hz and one-sided bandwidth W Hz (roughly 20 MHz in this case). The 70-MHz IF signal is downconverted to 10 MHz and sampled at 40 MHz.

B. Residual Carrier Tracking

In-phase and quadrature digital mixing are employed to convert the 10-MHz digital IF signal to the following baseband samples [10]

$$I_{cr}(n) = \sqrt{P}\,[D(n)\sin\Delta\sin\phi_c(n) + \cos\Delta\cos\phi_c(n)] + n_I(n) \tag{4}$$

$$Q_{cr}(n) = \sqrt{P}\,[D(n)\sin\Delta\cos\phi_c(n) + \cos\Delta\sin\phi_c(n)] + n_Q(n) \tag{5}$$

where $n_I(n)$ and $n_Q(n)$ are independent Gaussian random variables with variances $N_0/2T_s$ and $\phi_c(n)$ is the carrier phase error at time nT_s where T_s denotes sampling interval. Note that the power available for residual carrier tracking is $P_C = P\cos^2\Delta$ due to the nonzero modulation index. The subscripts "cr" denote the carrier (c) residual (r) component. Several other subscripts will be used: "sc" to denote subcarrier, "cs" to indicate the carrier suppressed component and "sy" to denote symbol-synchronization loop related terms.

In residual phase tracking, the Q_{cr}s are accumulated over N_1 samples to reduce the processing rate from 20 MHz to the 100-Hz–1-kHz range to enable a software implementation of the loop filter $F(z)$. Due to the averaging (accumulation) operation, the first component of Q_{cr} becomes zero due to summing over several cycles of the subcarrier. In the ARX II, the loop filters are all given by

$$F(z) = G_1 + \frac{G_2}{1 - z^{-1}} + \frac{G_3}{(1 - z^{-1})^2} \tag{6}$$

where

$$G_1 = rd/T_u \tag{7}$$

$$G_2 = rd^2/T_u \tag{8}$$

$$G_3 = krd^3/T_u \tag{9}$$

and

$$d = 4B_L T_u(r - k)/r(r - k + 1) \tag{10}$$

T_u denotes the loop update time, B_L the design loop bandwidth in Hz; r is typically 2 or 4 and is equal to 4ξ where ξ is the damping ratio, and k is a type-III loop gain parameter ($k = 0$ for type-II loop) with typical values ranging from 1/4 to 1/2. The filter of Eq. (6) was derived from an equivalent analog filter using the impulse-invariant transformation (IIT). The actual loop bandwidth B_L^a might be larger than the designed B_L depending on the product $B_L T_u$. Generally, for $B_L T_u < 0.05$, the actual loop bandwidth is very close to the design loop bandwidth B_L. The infinite impulse response (IIR) digital filter described above was found to provide the "best" performance for small values of $B_L T_u$ in terms of gain margins, transient response, and steady-state error performances [11]. Other filters derived from optimum estimation theory were also considered [12–16] but were not implemented due to their additional complexity and relative performance.

In the case of the PLL, the filter provides a phase-error estimate $\widehat{\phi_c}$ which adjusts the frequency of the NCO every update period. An averaged version of the in-phase component of the residual carrier, I_{cr}, is used to provide an indicator for lock detection. Assuming that the residual carrier loop is in lock, the averaged in-phase samples are roughly unity (assuming proper normalization) since they are proportional to $\cos\phi_c$. In fact, that same quantity can be used to verify in real time that the loop gain is actually what it is supposed to be and thus that the operating bandwidth is actually B_L as designed. The normalization involved depends on the actual implementation of the hardware and will not be discussed in this article. A cycle count of the received phase is performed and time tagged using station time to enable digital Doppler extraction.

C. Subcarrier Tracking

In an ideal carrier-tracking situation, $\phi_c(n) = 0\ \forall n$ and $Q_{cr}(n)$ of Eq. (5) becomes proportional to the subcarrier waveform $D(n)$. Hence, the input to the subcarrier tracking loop is the quadrature component of the residual carrier. Mixing $Q_{cr}(n)$ with the subcarrier in-phase and quadrature references and ignoring the double-frequency terms, one obtains

$$I_{sc}(n) = \sqrt{P_D}\, d(n) F_I(\phi_{sc}(n)) \cos(\phi_c(n)) + N_I(n) \quad (11)$$

and

$$Q_{sc}(n) = \sqrt{P_D}\, d(n) F_Q(\phi_{sc}(n)) \cos(\phi_c(n)) + N_Q(n) \quad (12)$$

where $P_D = P\sin^2\Delta$ is the data power, ϕ_{sc} is the subcarrier phase error in rad, and $N_I(n)$ and $N_Q(n)$ are independent zero-mean Gaussian random variables with variances $N_0/2T_s$ and $N_0 W_{sc}/2T_s$, respectively. W_{sc} denotes the width of the window in fraction of cycles that might be used in the quadrature arm to improve the performance for square-wave tracking ($W_{sc} \le 1$ and $W_{sc} = 1$ corresponds to no window).

For a square-wave subcarrier with window W_{sc}, the I and Q phase functions $F_I(\phi)$ and $F_Q(\phi)$ are given by [17]

$$F_I(\phi) = 1 - \frac{2}{\pi}|\phi| \qquad |\phi| \le \pi \quad (13)$$

and

$$F_Q(\phi) = \begin{cases} (2/\pi)\phi & |\phi| \le (\pi/2)W_{sc} \\ \mathrm{sgn}(\phi)W_{sc} & (\pi/2)W_{sc} \le |\phi| \\ & \le \pi(1 - W_{sc}/2) \\ 2\mathrm{sgn}(\phi) - (2/\pi)\phi & \pi(1 - W_{sc}/2) \le |\phi| \le \pi \end{cases} \quad (14)$$

whereas in the case of a sine wave, they become

$$F_I(\phi) = \cos\phi \qquad F_Q(\phi) = \sin\phi \quad (15)$$

The signals $I_{sc}(n)$ and $Q_{sc}(n)$ are then accumulated over a symbol duration to produce

$$I_{sc,mf}(k) = \sqrt{P_D}\, a_k F_I(\phi_{sc}(k)) \cos(\phi_c(k)) + N_{I,mf}(k) \quad (16)$$

and

$$Q_{sc,mf}(k) = \sqrt{P_D}\, a_k F_Q(\phi_{sc}(k)) \cos(\phi_c(k)) + N_{Q,mf}(k) \quad (17)$$

where the subscript "mf" indicates the output of the matched filter. The control signal for the matched filters is derived from the symbol synchronization loop and will be discussed later. The noises $N_{I,mf}(k)$ and $N_{Q,mf}(k)$ are still independent with respective variances $N_0/2T$ and $N_0 W_{sc}/2T$, where T denotes symbol duration and k the discrete time kT.

The error signal of the subcarrier loop is obtained by forming the product of $N_{I,mf}(k)$ and $N_{Q,mf}(k)$ to wipe out the data, accumulating over several symbols to reduce the processing from the symbol rate to a more appropriate subcarrier loop update rate, and feeding the average to the loop filter. The in-phase and quadrature subcarrier signals

are also individually squared and subsequently subtracted from one another to provide a lock indicator signal to the operator [18].

D. Suppressed Carrier Tracking and Sideband Aiding

For suppressed carrier tracking, the subcarrier in-phase component, $I_{sc,mf}(k)$ given by Eq. (16), is also the suppressed carrier in-phase component since it is already proportional to $\cos\phi_c(k)$ and to $F_I(\phi_{sc})$. In order to generate the quadrature component of the suppressed carrier, the in-phase component of the residual carrier ($I_{cr}(n)$ defined by Eq. 4) is used since it is proportional to $\sin\phi_c$. It is demodulated with the same subcarrier reference used to get $I_{sc}(n)$ and accumulated over a symbol period. Hence, the suppressed carrier signals become

$$I_{cs}(k) = \sqrt{P_D}a_k F_I(\phi_{sc}(k))\cos\phi_c(k) + N_{I,mf}(k) \quad (18)$$

and

$$Q_{cs}(k) = \sqrt{P_D}a_k F_I(\phi_{sc}(k))\sin\phi_c(k) + N^c_{Q,mf}(k) \quad (19)$$

where $\phi_c(k)$ is the average carrier phase error over the kth symbol and $N^c_{Q,mf}(k)$ is the noise quadrature component with variance $N_0/2T$. The phase-error estimate from the BPSK suppressed carrier tracking loop can be combined with its counterpart from the residual carrier tracking loop, with the appropriate weights, to form a combined carrier phase-error estimate, which ultimately adjusts the frequency of the carrier NCO. This is referred to as *sideband aiding* since the power in the data or sidebands is also used to track the carrier along with the power of the residual carrier. Depending on the modulation format (BPSK or QPSK), the samples $I_{cs}(k)$ and $Q_{cs}(k)$ undergo different processing to produce a phase-error estimate; for BPSK, the error signal is proportional to their product whereas for QPSK, the feedback signal is derived using $I_{cs}\text{sgn}(Q_{cs}) - Q_{cs}\text{sgn}(I_{cs})$ (where $\text{sgn}(x)$ denotes the "signum" function) as described in [19].

E. Symbol Synchronization

Obviously, symbol synchronization needs to be maintained to enable subcarrier and suppressed-carrier tracking. This is accomplished by the digital data-transition tracking loop (DTTL) [20], which uses the in-phase signal of the subcarrier, $I_{sc}(n)$, as its input. Two integrate-and-dump filters are used, one to accumulate over a symbol and the second over a symbol transition with possible windowing. The first, $I_{cs}(k)$, is already performed by both the subcarrier and suppressed carrier loops and can thus be used in the symbol transition detector. From Fig. 3, the second signal is obtained at the output of the symbol transition WIDF, then multiplied by the first and accumulated to form the symbol phase-error estimate. A lock indicator on the symbol synchronization loop can be obtained by the SSME output which provides a symbol SNR estimate to the operator. The quality of those estimates is a function of both the performance of the symbol synchronization loop and the number of samples per symbol, as will be discussed later.

IV. Algorithm Descriptions

It is very important from an operational point of view to understand the limitations of the various algorithms involved to decide whether the system itself is not performing satisfactorily, or whether the system's status-reporting method is generating misleading data. This requires not only an intuitive understanding of each algorithm, but also an understanding of the possible correlation between the various parameters and coupling among the different loops. In this section, the loops are discussed and their respective performances indicated, but not derived. Moreover, the limitations of the various analyses are clarified and the underlying assumptions are identified.

A. Performance of the BPSK Carrier Tracking Loops

First consider the performance of the simplest loop, the digital PLL. It is well known that the phase-error variance is given by [21]

$$\sigma^2_{\phi_{cr}} = \frac{N_0 B^a_{L,cr}}{P_C} \quad (20)$$

where $B^a_{L,cr}$ is the actual operating bandwidth of the residual carrier loop. The latter can be different from the design bandwidth $B_{L,cr}$ and is dictated by the difference between the predicted P_C/N_0, denoted $(P_C/N_0)_p$, and the actual P_C/N_0. Typically, the loop bandwidth is dependent on the loop gain, which depends on the amplitude of the incoming signal. But because of the total-power AGC present in the IF stage, the amplitude of the signal at the output of the A/D converter in units of the least significant A/D levels becomes dependent on the incoming P_C/N_0. During system initialization, the loop gain is normalized so as to obtain the desired loop bandwidth $B_{L,cr}$ based on the available predict $(P_C/N_0)_p$. The equations describing the various normalizations are beyond the scope of this article.

Note that when the actual P_C/N_0 is different from its predicted counterpart, the amplitude of the signal will be different from that predicted at the A/D output and

the normalization performed will not provide the designed loop gain, and hence, the desired loop bandwidth. However, the loop bandwidth can be monitored in real time by averaging the in-phase residual carrier component over an appropriate length of time and comparing it with the predicted average, which for convenience can be normalized to unity. If the incoming amplitude is different from its predict, a correction can be made to renormalize it properly. For the averaging process to be reliable, it needs to be performed over a period during which the PLL has approximately zero steady-state phase error due to dynamics. Otherwise, the averaged quantity will reflect more than the difference in loop gain.

A steady-state phase-error estimate can be obtained in real time by computing

$$\widehat{\phi_{ss}}(j) = \tan^{-1}\left(\frac{\sum_{n=j}^{n=j+L+1} Q_{cr}(n)}{\sum_{n=j}^{n=j+L+1} I_{cr}(n)}\right) \tag{21}$$

where L controls the estimation period. In the ARX II, $\widehat{\phi_{ss}}(j)$ can be sent to the monitor's screen at different rates ranging from once every second to once every minute. Another very useful parameter is the incoming P_C/N_0, which is also estimated by the receiver according to

$$\widehat{\frac{P_C}{N_0}}(j) = \frac{\left(\sum_{n=j}^{n=j+L+1} I_{cr}(n)\right)^2}{2T_{u,cr}\sum_{n=j}^{n=j+L+1} Q_{cr}^2(n)} \tag{22}$$

It can also be sent to the screen at the estimation rate ($T_{u,cr}$ is the residual carrier loop update period). The main disadvantages of that estimator are that first, it requires residual carrier lock, and second, the phase jitter needs to be "small," otherwise, a degraded estimate results, the amount of which highly depends on the residual-carrier loop SNR.

A residual carrier lock indicator can be mechanized by averaging the $I_{cr}(n)$ samples of Eq. (4) over a sufficient period and comparing the result to a threshold to obtain a decision on the lock status. If in-lock is reported, the receiver continues its regular tasks. However, if an out-of-lock status is indicated, an additional verification of the status is performed to lower the probability of a false alarm or indication. This process is well documented elsewhere [2] and is referred to as *sequential detection*. In the case of a final out-of-lock decision, the loop is disabled and an FFT is performed on the complex in-phase and quadrature samples using a separate FFT channel to measure the frequency difference between the signal and the NCO. Since a separate channel is used, the FFT can be performed at a different rate than the residual carrier update rate and can involve several FFT sweeps which are eventually noncoherently averaged to reduce the effect of the noise; this is particularly useful for very weak signals. A peak detection is then performed on the averaged power spectrum, the NCO is shifted accordingly, and the loop is enabled. FFTs can still be performed once the loop is enabled to verify that frequency acquisition and tracking are actually in progress. Other strategies such as Adaptive Least Squares [22] or Frequency Sweeping [20, 21] have been considered for automating this process so that the receiver can locate the signal, acquire it, track it, verify that the right signal is being tracked by comparing predicts and measurements, monitor its status, and proceed accordingly depending on the lock indicators.

Depending on the value of the modulation index Δ, the carrier might become fully suppressed and thus, needs to be tracked with a Costas loop. The latter forms a phase discriminator by forming the product of $I_{cs}(k)$ and $Q_{cs}(k)$ at the symbol rate and accumulating the result over M_4 symbols to average the phase error. Obviously, the time constant of the phase-error process is much larger than the symbol duration and the averaging does indeed reduce the noise. The performance of the loop in terms of phase-error variance is easily derived and is given by [10]

$$\sigma^2_{\phi_{cs}} = \frac{N_0 B^a_{L,cs}}{P_D S_{L,BPSK}} \tag{23}$$

where $B^a_{L,cs}$ is the actual loop bandwidth of the suppressed carrier loop and E_S/N_0 is the symbol energy-to-noise ratio equal to $P_D T/N_0$; T is the symbol period defined earlier and $S_{L,BPSK}$ is the Costas loop "squaring loss" given by

$$S_{L,BPSK} = \frac{1}{1+\frac{1}{2E_S/N_0}} \tag{24}$$

Here again, depending on the difference between the incoming P_D/N_0 and the predicted P_D/N_0, $(P_D/N_0)_p$, the actual loop bandwidth $B^a_{L,cs}$ might be different from its design counterpart. However, unlike the residual carrier loop, monitoring the loop bandwidth in real time cannot be performed by the loop itself due to the presence of the data modulation but can be accomplished by a separate P_D/N_0 estimator.

A "square law"-type lock detector is used to monitor the BPSK suppressed carrier loop status. This detector is not shown in Fig. 3 on the Costas loop but is shown for the subcarrier loop. The detector generates a signal proportional to $\cos 2\phi_{cs}(k)$ by squaring the samples $I_{cs}(k)$ and $Q_{cs}(k)$, subtracting them, averaging over many sym-

bols and thresholding the result. A sequential detection scheme can also be used to improve the detector's performance. This detector was chosen over the "absolute"-type detector because it requires 1 dB less in symbol energy-to-noise ratio to achieve the same lock-detection probability for a given false-indication rate [18].

B. Sideband-Aiding Performance

In most instances, the incoming carrier is neither a residual nor a fully suppressed carrier, but rather a mixture of the two. In those cases, both the carrier and the data power can be used to perform carrier synchronization and should in principle jointly outperform any individual tracking scenario. This is referred to as *sideband aiding* since the power in the sidebands (or data) is used with the carrier power to perform carrier tracking. In this case, both loop update rates are identical and the joint carrier phase-error estimate, $\widehat{\phi_c}$, that ultimately adjusts the frequency of the carrier NCO is given by the weighted sum of the individual estimates as

$$\widehat{\phi_c} = \alpha\widehat{\phi_{cr}} + (1-\alpha)\widehat{\phi_{cs}} \tag{25}$$

where $\widehat{\phi_{cr}}$, $\widehat{\phi_{cs}}$ are the individual phase-error estimates by the residual and suppressed carrier loops respectively and α is the optimum weighting factor [10] given by

$$\alpha = \frac{\cos\Delta\sqrt{P_D + N_0/2T}}{\sin\Delta + \cos\Delta\sqrt{P_D + N_0/2T}} \tag{26}$$

Note that the optimum weight α requires knowledge of both P_D and N_0 individually and not just the ratio. When the carrier is fully suppressed (i.e., $\Delta = 90$ deg), $\alpha = 0$ and $\widehat{\phi_c} = \widehat{\phi_{cs}}$. On the other hand, when the carrier is a pure tone (i.e., $\Delta = 0$ deg), $\alpha = 1$ and $\widehat{\phi_c} = \widehat{\phi_{cr}}$ as it should. When using the optimum weight α, the performance of the sideband-aided loop becomes

$$\sigma^2_{\phi_c} = \frac{1}{1/\sigma^2_{\phi_{cr}} + 1/\sigma^2_{\phi_{cs}}} \tag{27}$$

where $\sigma^2_{\phi_{cr}}$, $\sigma^2_{\phi_{cs}}$ are the variances of the individual tracking loops given by Eqs. (20) and (23), respectively.

C. Performance of the Costas Cross-over Loop

The remaining carrier tracking loop to be discussed is the QPSK loop, which also processes the $I_{cs}(k)$ and $Q_{cs}(k)$ samples to provide a phase-error estimate. The received signal in this case is given by

$$r(t) = \sqrt{P}\,[d_1(t)\sin(\omega_i t + \theta) + d_2(t)\cos(\omega_i t + \theta)] + n(t) \tag{28}$$

where $d_1(t)$, $d_2(t)$ are the in-phase and quadrature modulations similar to Eq. (2) and T the QPSK symbol duration. The implemented loop is the so-called Costas cross-over loop, which is an approximation to the maximum-likelihood (ML) estimator at high signal-to-noise ratios.

The error signal that forms the input to the loop filter is obtained by averaging M_3 samples of the form

$$e(k) = I_{cs}(k)\mathrm{sgn}(Q_{cs}(k)) - Q_{cs}(k)\mathrm{sgn}(I_{cs}(k)) \tag{29}$$

The average can be shown to be proportional to the phase error ($\phi_{QPSK}(k)$), assuming a linearized model. The performance of the loop has been derived in various places and is given by [19]

$$\sigma^2_{\phi_{QPSK}} = \frac{1}{\rho S_{L,QPSK}} \tag{30}$$

where $\rho = P/N_0B_L$ is the loop signal-to-noise ratio of a "classical" PLL and $S_{L,QPSK}$ is the QPSK loop "squaring" loss. It can be shown that [19]

$$S_{L,QPSK} = \frac{\left[\Phi(\sqrt{R_d/2}) - \sqrt{2R_d/\pi}e^{-R_d/2}\right]^2}{1 + R_d - 2\left[\frac{1}{\sqrt{\pi}}e^{-R_d/2} + \sqrt{R_d/2}\Phi(\sqrt{R_d/2})\right]^2} \tag{31}$$

where $R_d = PT/N_0 = E_s/N_0$ is the QPSK symbol energy-to-noise ratio and $\Phi(x)$ is the error function given by

$$\Phi(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \tag{32}$$

This loop was chosen for simplicity of implementation.

The other two candidates, the ML loop and its low SNR approximation, require more hardware to implement and do not offer significant advantages. The ML loop requires implementation of the hyperbolic tangent function, which can be achieved with read-only memory (ROM) chips, but the input to the ROM chips has to be scaled by the signal amplitude and the noise spectral level. These are not known to the receiver and therefore must be esti-

mated. The sensitivity of the performance of the ML loop with respect to those parameters is not known and needs to be determined before the loop can be chosen as an implementable candidate. The generalized Costas loop (which is a low SNR approximation to the ML loop) requires implementing squaring operations, and thus scaling is also required to handle the wide dynamic range of signal-to-noise ratios and symbol rates involved. The Costas cross-over loop is the easiest to implement since the signum functions use only the sign bit of the input signals and are thus very straightforward to implement digitally. The remaining operations involve multiplications, additions, and accumulations, which are easily done. Furthermore, the Costas cross-over loop does not suffer from a significant degradation in performance [19] in the region of interest.

D. Subcarrier Loop Performance

Once carrier acquisition and tracking are achieved, subcarrier (if present) and symbol synchronization are still required to eventually detect the received symbols. Note that a subcarrier will not be used with QPSK, but only with nonsuppressed BPSK (i.e., $\Delta \neq 90$ deg), in which case a residual carrier component is present. The subcarrier can be either a square wave or a sine wave and is used to shift the data spectrum away from the residual component. In fact, bi-phase modulation can be thought of as a subcarrier which shifts the data away from the carrier by $1/T$ Hz. Often, one would like to shift the data by more than $1/T$ Hz so that enough guard band is included to reduce the interference from the tails of the data spectrum.

The Costas loop described earlier can be used to track both sine-wave and square-wave subcarriers. In this case, an averaged, filtered version of the product of $I_{sc,mf}(k)$ and $Q_{sc,mf}(k)$ is used to drive the subcarrier NCO. For square-wave subcarriers, a windowing operation can be performed on the Q-channel. This results in improved tracking performance, but smaller acquisition pull-in range [17]. In that case, the phase jitter is given by

$$\sigma^2_{\phi_{sc}} = \left(\frac{\pi}{2}\right)^2 W_{sc} \left(\frac{N_0 B^a_{sc}}{P_D}\right)(1 + N_0/2E_s) \qquad (33)$$

while for a sine-wave subcarrier, one has

$$\sigma^2_{\phi_{sc}} = \left(\frac{N_0 B^a_{sc}}{P_D}\right)(1 + N_0/2E_s) \qquad (34)$$

where B^a_{sc} is the actual subcarrier loop bandwidth and W_{sc} the subcarrier window. A square-law detector is used to indicate subcarrier lock to the operator and to facilitate system automation.

E. Symbol Synchronization Performance

For both the suppressed carrier and subcarrier loops to function properly, symbol epoch tracking has to be maintained so that the loop arm-filter outputs are actually those given by Eqs. (16) and (17). The task of symbol synchronization is accomplished by the data-transition tracking loop (DTTL), which uses the $I_{sc}(n)$ samples of Eq. (11) as its input. The DTTL was chosen because of its simplicity and improved performance when using a window. The other two candidates, the "Absolute Value Type of Early-Late Gate Symbol Synchronizer" (AVTS) and the "Difference-of-Squares Loop" (DSL) [20], are harder to implement and do not offer significant advantages.

In the DTTL, two accumulators are used, one to detect the received symbols and the other to accumulate over symbol transitions. The first is typically referred to as the "in-phase" filter and the other as the "mid-phase" filter (WIDF with symbol transition window in Fig. 3). Following the in-phase operation, a transition detector outputs a 0 or ± 1 to indicate a no-transition, a $+1$ to -1, or a -1 to $+1$ transition respectively. The output is then multiplied by the mid-phase accumulator to wipe out the effect of the data and accumulated over several symbols to reduce the rate from the symbol to the loop update rate. Note that a windowing operation can be performed during the mid-phase accumulation and that results in a decrease in the loop phase jitter.

The performance of the loop when the arm filters are analog devices has been evaluated elsewhere [20] and the phase-error variance is given approximately by

$$\sigma^2_{\phi_{sy}} \simeq 2\pi^2 W_{sy} \left(\frac{N_0}{E_s}\right)\left(\frac{B^a_{sy}}{R_s}\right) \frac{1}{\Phi^2(\sqrt{E_s/N_0})} \qquad (35)$$

where B^a_{sy} is the actual symbol synchronization loop bandwidth, W_{sy} the symbol transition window, and R_s the symbol rate. In the ARX II, the "typical" analog arm filters are digital accumulators and the overall loop performance is still given by Eq. (35) as long as the number of samples per symbol (denoted by β) remains "large." However, the number of samples per symbol is not always "large" since at the high data-rate goal of 6.6 Msymbols/sec, only 3 to 4 samples per symbol are available (because of the 20-MHz processing rate). Furthermore, since the sampling clock is fixed and not driven by the symbol loop, β is not in general an integer even though it is a real number.

This problem was examined in a separate study [25]. It was concluded that the loop phase jitter will remain

small as long as "some" Doppler rate is present in the communication link. In the deep-space applications of interest, Doppler rate due to Earth rotation is enough to help the DTTL achieve sufficiently small phase jitter. The main problem for the DTTL occurs when the number of samples per symbol is exactly a small integer number, for example, $\beta_I = 4$. (When β is an integer, it will be denoted by β_I). In this case, the phase error increases dramatically due to the resolution offered by exactly 4 samples/symbol and Eq. (35) is no longer valid due to the additional error of the "self-noise," which was not accounted for. That error is equal to the variance of a uniformly distributed random variable over $(-1/2\beta_I, 1/2\beta_I)$, which for $\beta_I = 4$ gives a 73 mcycle root-mean-squared (rms) error! (In general, the rms error due to the self-noise of an integer number of samples per symbol is given by $1/(2\sqrt{3}\beta_I)$.)

Note that the true symbol phase jitter is not the sum of the white-noise and the self-noise jitters. Rather, it is related to both through a nonlinear equation involving the integral of the density function of the phase-error process. Only in special circumstances is the sum assumption a "good" approximation, but it can be used throughout as a rough rule of thumb. In real communication links, nonzero Doppler rate is typically present and is enough to guarantee either that β will not be an integer or that it will be an integer for a "short" period of time only. In the latter case, the symbol synchronization loop will not be affected due to the long averaging performed with the "typical" small loop bandwidth (1–100 mHz) it operates with.

F. Symbol SNR Estimator Performance

One way to monitor the symbol-synchronization loop performance in real time is through the use of a symbol SNR estimator. This is different from other loops where a binary decision is provided on the status of the loop. In this case, the estimator will continuously provide real-time estimates of the symbol SNR, which are compared to the expected symbol SNR. If the numbers are "close," symbol phase lock is assumed and system operation is resumed. If the numbers are different, then three cases are possible: either the loop is out of lock, the received symbol SNR is different from that expected, or, the estimator itself is providing misleading data due to its own limitation. From a user's point of view, these possible scenarios are too confusing and an ultimate binary decision is highly desirable. The most disturbing of the three outcomes is the case involving misleading data from the SNR estimator. This is obviously highly dependent on the estimator structure used and thus an understanding of the algorithm is required.

The algorithm used in the ARX II is the so-called *"Split Symbol Moment Symbol SNR Estimator"* (SSME) [26], which was originally designed to operate in an additive white Gaussian noise (AWGN) channel. A table look-up technique [27] could also be used, but the SSME was chosen due to its simplicity.

The input to the estimator is the data stream after it has been demodulated, i.e., the sequence of $I_{sc}(n)$ samples. Assuming that perfect carrier and subcarrier lock has been achieved, the samples become

$$I_{sc}(n) \simeq \sqrt{P_D}d(n) + N_I(n) \tag{36}$$

These samples are accumulated over a symbol duration by the matched filters in the various loops, where the accumulation start and stop times are provided by the DTTL. In fact, the carry-out bit of the symbol NCO can be used to clock all the accumulators that operate over a symbol duration. The SSME uses those samples and accumulates them over the first and second halves of a symbol period using a control signal derived from the symbol NCO phase. These are shown in Fig. 3 as *"First-Half WIDF"* and *"Second-Half WIDF."* By further processing these outputs, an averaged SNR estimate can be obtained over many symbols.

The principle of operation of the algorithm is as follows. A total (signal plus noise)-power estimate is obtained by summing, squaring, and averaging the outputs of the half WIDFs. Simultaneously, a signal-power estimate is obtained by forming the product of the half-WIDF outputs and then averaging over many symbols. A noise-power estimate is then easily obtained by subtracting the previous estimates with the appropriate weighting. Finally, a symbol-SNR estimate is derived from the separate signal- and noise-power estimates by taking the ratio. It was shown [28] that the algorithm works well when operating at low symbol rates and in the presence of additive white Gaussian noise (AWGN). However, at high symbol rates, filtering of the data becomes significant and does result in erroneous estimates. These estimates depend on β (number of samples per symbol), the data rate, and the effective filtering in the data path. This is because the SSME is operating in a region that it was not designed for. The user should be aware of this limitation of the SSME and act accordingly.

G. Weighted Integrate-and-Dump Filters

Another algorithm requiring description is the weighted integrate-and-dump filter (WIDF). It is well known in detection theory that the optimum detector in an AWGN channel is the matched filter followed by a sampler.

The filter's impulse response is matched to the incoming signal such that at the sampling instant maximum symbol SNR is achieved.

In digital circuits, one way to detect the symbols is to accumulate the samples over a symbol duration. By accumulation is meant a straight summation. This will maximize the symbol SNR as long as all the samples have the same SNR. However, this is rarely the case because the first and last samples of a symbol are affected by the rise and fall times of the pulse. Consequently, they will have different pulse amplitudes and thus, different sample SNRs.

When "many" samples per symbol are available, the first and last samples form a minority of all the samples and a straight accumulation will probably provide the maximum symbol SNR that can be achieved. However, when only three or four samples per symbol are taken, the first and last samples form a substantial percentage of the samples and thus need to be accounted for. This is accomplished by weighting the samples while accumulating them to improve the symbol SNR. This is easily implemented using a multiply-and-accumulate chip (MAC) with the weights being read from a memory chip.

The optimum weights depend on the filtering involved, on the number of samples per symbol, and on where the first sample occurs within the symbol (typically referred to as sampling offset); the latter is estimated by the symbol-synchronization loop. Assuming perfect knowledge of the sampling offset, it has been shown that weighting provides an average 0.2-dB improvement in symbol SNR over the straight accumulation (in DSN applications, this is a significant improvement). Moreover, when the sampling offset is exactly zero, the improvement can be as large as 1.5 dB in symbol SNR [28, 29].

In a practical system, less gain would be achieved due to the jitter in estimating the sampling offset and due to not knowing the exact filtering taking place. Nevertheless, some gain is expected in telemetry SNR since the weights are not assumed to be very sensitive to those parameters. The effect of filter distortion on telemetry SNR has also been examined [31] and it was determined that an amplitude ripple of less than 0.42 dB would guarantee less than 0.01-dB loss in symbol SNR.

V. Implementation Trade-off Issues

The previous section provided a description of the various algorithms employed to perform the necessary tasks to demodulate the received signal and monitor the receiver's performance. Implementation issues were purposely avoided to give the reader a clear understanding of each algorithm's performance based on theoretical results. In this section, some of the characteristic losses inherent to our implementation are discussed and clarified.

The starting point of the digital hardware is the A/D operation, which operates at 40 MHz using eight bits. As far as SNR losses due to signal quantization are concerned, the use of four or more bits has negligible effect on sample SNR. In fact, for four bits, the loss is about 0.04 dB and is practically zero for more bits. Eight bits were used in the ARX II to provide some protection against the presence of relatively strong interference. For example, for an interference-to-noise ratio (at the input to the A/D converter) of 32.6 dB, 8 and 4 bits result in about 0.01-dB and 12.36-dB losses respectively. The loss assumes that the interference contributed only to quantization noise. A more realistic scenario would be to consider the effect of some spectral components due to interference in the signal frequency band, as this would create additional in-band noise.

Following the digital conversion, carrier demodulation is performed followed by low-pass filtering. At the output of the low-pass filter, the signal spectrum extends up to 10 MHz only and thus a sampling rate of 20 MHz is sufficient to perform further signal processing. The combination of low-pass filtering at 40 MHz and sample decimation by two can be implemented in an equivalent processing at 20 MHz [32] using half-band FIR filters. The filter response is shown in Figs. 4 (a) and (b) along with its finite-bit implementation. It is clear that a 9-bit implementation would not result in significant changes in filter response whereas an 8-bit version would contribute more pass-band ripple and higher sidebands.

Because of the finite-bit implementation, scaling is required throughout the receiver to accommodate the wide range of signal-to-noise ratios and symbol rates anticipated. As an example, most accumulators accept 16-bit inputs and output a 16-bit result. The accumulation is usually performed using 32 bits but only 16 of those can be accessed at the maximum chip speed of 20 MHz. Typically, the most significant 16 bits are hardwired to the next stage and software-controlled scaling is used to make sure that the result of the accumulation does indeed lie in that range. As mentioned previously, all loop filters are implemented in software and their outputs are the various frequency errors that need to be compensated for by the respective NCOs.

In the digital loop closure, the fractional phase is accumulated internally using 32 bits but only the most significant 12 bits are used in the look-up table, resulting in a

0.24-mcycle resolution. The phase is recorded using 48 bits for the integer part and 12 bits for the fractional part. The NCO is running at 10 MHz with a 40-MHz clock, resulting in 9.3-mHz ($4 \times 10^7/2^{32}$) frequency resolution.

In the analog loop closure, the fractional phase is recorded using 14 bits (resulting in an improved 0.06-mcycle resolution), two of which are accumulated externally using the carry-out bit of the NCO. The NCO is actually running at 16 MHz with a 20-MHz clock, but it effectively produces a 4-MHz signal because of the additional bits. The latter also improves the frequency resolution to 1.16 mHz ($2 \times 10^7/2^{34}$), but limits the maximum nominal frequency to one fourth of the clock rate (the practical limit of an NCO is roughly 40 percent of the clock). That in return limits the dynamic excursion of the frequency to ± 1 MHz in the analog loop closure, whereas it is about ± 5 MHz in the digital closure. Time tagging is performed using ± 1 μsec accuracy with a ± 1 nsec stability. The logging rate can be chosen by software from 1, 2, 5, 10, or 20 Hz.

Note that in both analog and digital loop closures, 16-bit sine and cosine numbers are generated to reduce the in-phase and quadrature DC offsets [32] to an acceptable level. Since the subcarrier frequency and symbol rate are always less than the IF carrier frequency, the frequency resolution of their respective NCOs will be much higher because they also use 32-bit commands. From the symbol and the subcarrier NCOs, only the most significant 8 bits of fractional phase (3.9-mcycle resolution) are used to generate the subcarrier waveform and to clock all the WIDFs. This will limit the number of windows that can be used to eight, with W_{sc} or W_{sy} equal to 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, or $\frac{1}{128}$ (7.81 mcycle) only. This limit does not affect the system performance in any way, and the number of available windows can be increased at the cost of additional hardware.

VI. Conclusion

This article provided a functional description of the Advanced Receiver currently under development for future use in NASA's DSN. The requirements were specified and the receiver's architecture described. Moreover, the various signal-processing schemes were briefly discussed and their limitations clarified. The receiver incorporates functions currently available in a variety of receivers and does so digitally and with improved performance and options for spacecraft designers.

Acknowledgment

I thank the ARX II design team, which includes Mr. J. Statman, Mr. Hector Del Castillo, Mr. Roland Bevan, and Mr. Remi Labelle.

References

[1] D. H. Brown and W. J. Hurd, "DSN Advanced Receiver: Breadboard Description and Test Results," *TDA Progress Report 42-89*, vol. January–March 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 48–66, May 15, 1987.

[2] J. H. Yuen, *Deep Space Telecommunication Systems Engineering*, New York: Plenum Press, 1983.

[3] D. H. Brown, W. J. Hurd, V. A. Vilnrotter, and J. Wiggins, "Advanced Receiver Tracking of Voyager II Near Solar Conjunction," *TDA Progress Report 42-93*, vol. January–March 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 75–82, May 15, 1988.

[4] W. J. Hurd, D. H. Brown, V. A. Vilnrotter, and J. D. Wiggins, "Telemetry SNR Improvement Using the DSN Advanced Receiver with Results From Pioneer 10," *TDA Progress Report 42-93*, vol. January–March 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 64–82, May 15, 1988.

[5] V. A. Vilnrotter, W. J. Hurd, and D. H. Brown, "Optimized Tracking of RF Carriers with Phase Noise, Including Pioneer 10 Results," *TDA Progress Report 42-91*, vol. July–September 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 141–157, November 15, 1987.

[6] V. A. Vilnrotter, D. H. Brown, and W. J. Hurd, "Spectral Estimation of Received Phase in the Presence of Amplitude Scintillation," *TDA Progress Report 42-93*, vol. January–March 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 8–17, May 15, 1988.

[7] V. Vilnrotter, S. Hinedi, and R. Kumar, "Frequency Estimation Techniques for High Dynamic Trajectories," *IEEE Trans. on AES*, vol. AES-25, no. 4, pp. 559–577, July 1989.

[8] S. Aguirre and S. Hinedi, "Two Novel Automatic Frequency Tracking Loops," *IEEE Trans. on AES*, vol. AES-25, no. 5, pp. 749–760, September 1989.

[9] S. Hinedi and J. Statman, "Digital Accumulators in Phase and Frequency Tracking Loops," *IEEE Trans. on AES*, vol. AES-26, no. 1, pp. 1–13, January 1990.

[10] R. Sfeir, S. Aguirre, and W. J. Hurd, "Coherent Digital Demodulation of a Residual Carrier Signal Using IF Sampling," *TDA Progress Report 42-78*, vol. April–June 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 135–142, August 15, 1984.

[11] S. Aguirre, W. J. Hurd, R. Kumar, and J. Statman, "A Comparison of Methods for DPLL Loop Filter Design," *TDA Progress Report 42-87*, vol. July–September 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 114–124, November 15, 1986.

[12] C. A. Pomalaza-Raez and W. J. Hurd, "Improved Carrier Tracking by Smoothing Estimators," *TDA Progress Report 42-79*, vol. July–September 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 96–106, November 15, 1984.

[13] C. A. Pomalaza-Raez and W. J. Hurd, "Carrier Tracking by Smoothing Filter Can Improve Symbol SNR," *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 57–62, November 15, 1985.

[14] R. Kumar and W. J. Hurd, "A Class of Optimum Digital Phase Locked Loops for the DSN Advanced Receiver," *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 63–80, November 15, 1985.

[15] J. I. Statman, "A Recursive Solution for a Fading Memory Filter Derived From Kalman Filter Theory," *TDA Progress Report 42-86*, vol. April–June 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 70–76, August 15, 1986.

[16] J. I. Statman and W. J. Hurd, "An Estimator-Predictor Approach to PLL Loop Filter Design," *TDA Progress Report 42-86*, vol. April–June 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 77–89, August 15, 1986.

[17] W. J. Hurd and S. Aguirre, "A Method to Dramatically Improve Subcarrier Tracking," *TDA Progress Report 42-86*, vol. April–June 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 103–110, August 15, 1986.

[18] A. Mileant and S. Hinedi, "Costas Loop Lock Detection in the Advanced Receiver," *TDA Progress Report 42-99*, vol. July–September 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 72–89, November 15, 1989.

[19] M. K. Simon, "On the Optimality of the MAP Estimation Loop for Carrier Phase Tracking BPSK and QPSK Signals," *IEEE Trans. on Comm.*, vol. COM-27, no. 1, pp. 158–165, January 1979.

[20] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, New Jersey: Prentice-Hall Inc., 1973.

[21] S. Aguirre and W. J. Hurd, "Design and Performance of Sampled Data Loops for Subcarrier and Carrier Tracking," *TDA Progress Report 42-79*, vol. July–September 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 81–95, November 15, 1984.

[22] R. Kumar, "Fast Frequency Acquisition via Adaptive Least Squares Algorithm," *TDA Progress Report 42-85*, vol. January–March 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 53–60, May 15, 1986.

[23] S. Aguirre, D. H. Brown, and W. J. Hurd, "Phase Lock Acquisition for Sampled Data PLL's Using the Sweep Technique," *TDA Progress Report 42-86*, vol. April–June 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 95–102, August 15, 1986.

[24] S. Aguirre, "Acquisition Times of Carrier Tracking Sampled Data Loops," *TDA Progress Report 42-84*, vol. October–December 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 88–93, February 15, 1985.

[25] U. Cheng and S. Hinedi, "Performance of the All-Digital Data Transition Tracking Loop in the Advanced Receiver," *TDA Progress Report 42-99*, vol. July–September 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 60–71, November 15, 1989.

[26] M. K. Simon and A. Mileant, "SNR Estimation for the Baseband Assembly," *TDA Progress Report 42-85*, vol. January–March 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 118–126, May 15, 1986.

[27] V. A. Vilnrotter and E. R. Rodemich, "Table Look-Up Estimation of Signal and Noise Parameters From Quantized Observables," *TDA Progress Report 42-87*, vol. July–September 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 104–113, November 15, 1986.

[28] B. Shah and S. Hinedi, "Performance of the Split Symbol Moments SNR Estimator in the Presence of Inter-Symbol Interference," *TDA Progress Report 42-98*, vol. April–June 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 157–173, July 15, 1989.

[29] R. Sadr and W. J. Hurd, "Detection of Signals by the Digital Integrate-and-Dump Filter With Offset Sampling," *TDA Progress Report 42-91*, vol. July–September 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 158–173, November 15, 1987.

[30] R. Sadr, "Detection of Signals by Weighted Integrate-and-Dump Filter," *TDA Progress Report 42-91*, vol. July–September 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 174–185, November 15, 1987.

[31] R. Sadr and W. J. Hurd, "Filter Distortion Effect on Telemetry SNR," *TDA Progress Report 42-88*, vol. October–December 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 59–66, February 15, 1986.

[32] R. Sadr and W. J. Hurd, "Digital Carrier Demodulation for the DSN Advanced Receiver," *TDA Progress Report 42-93*, vol. January–March 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 45–63, May 15, 1988.

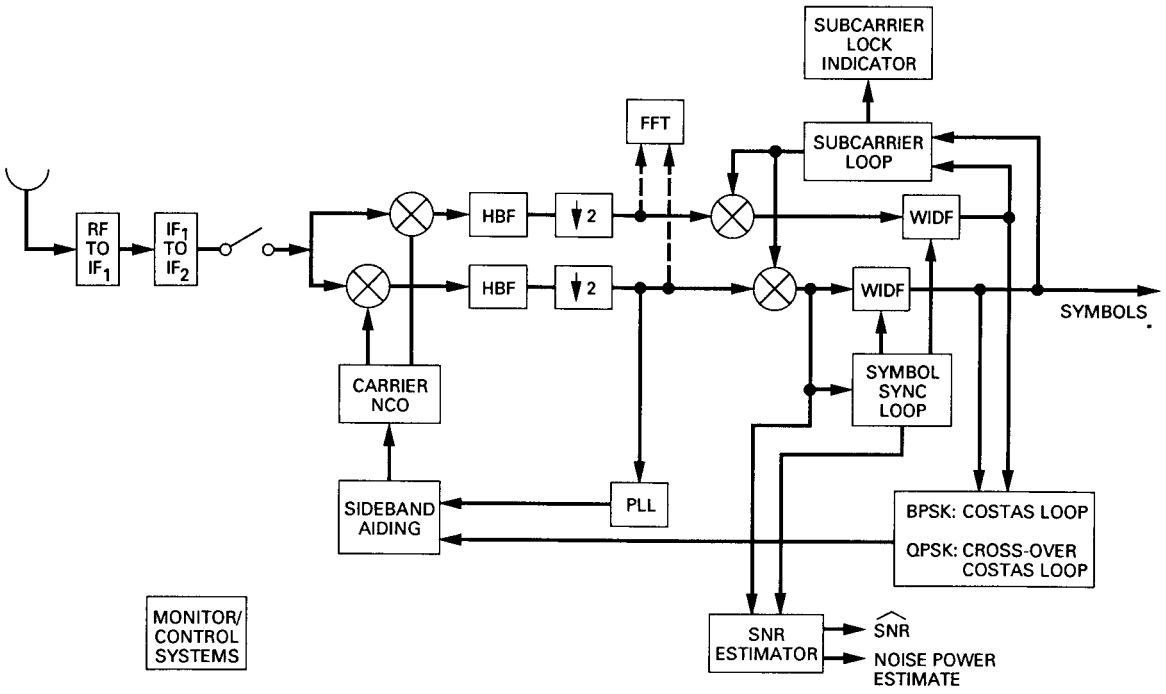

Fig. 1. Top-level block diagram of the ARX II.

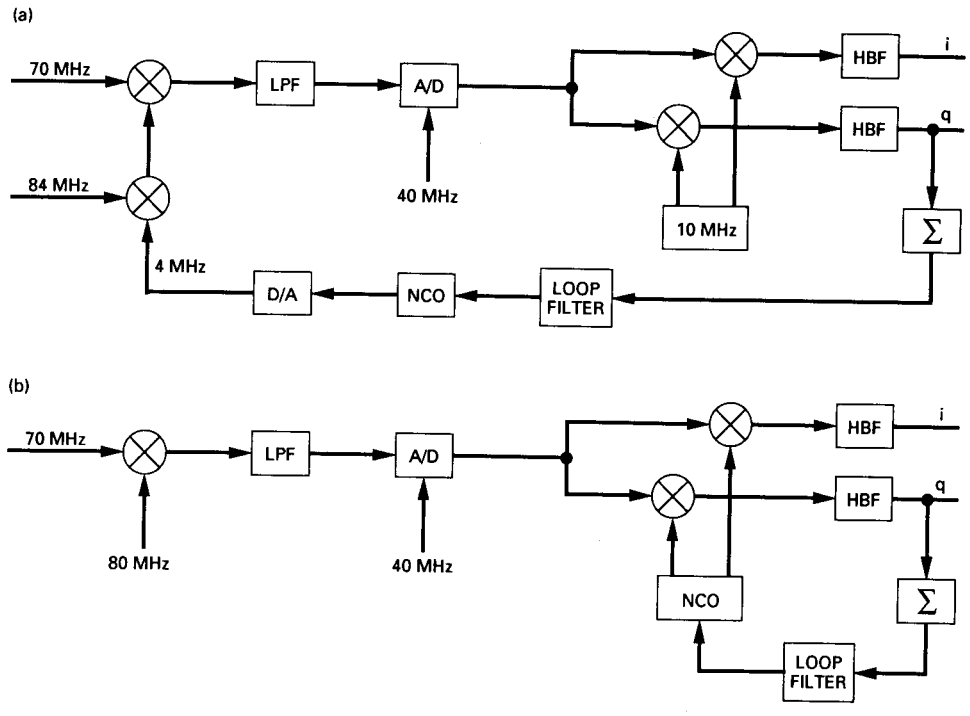

Fig. 2. Loop closures: (a) analog and (b) digital.

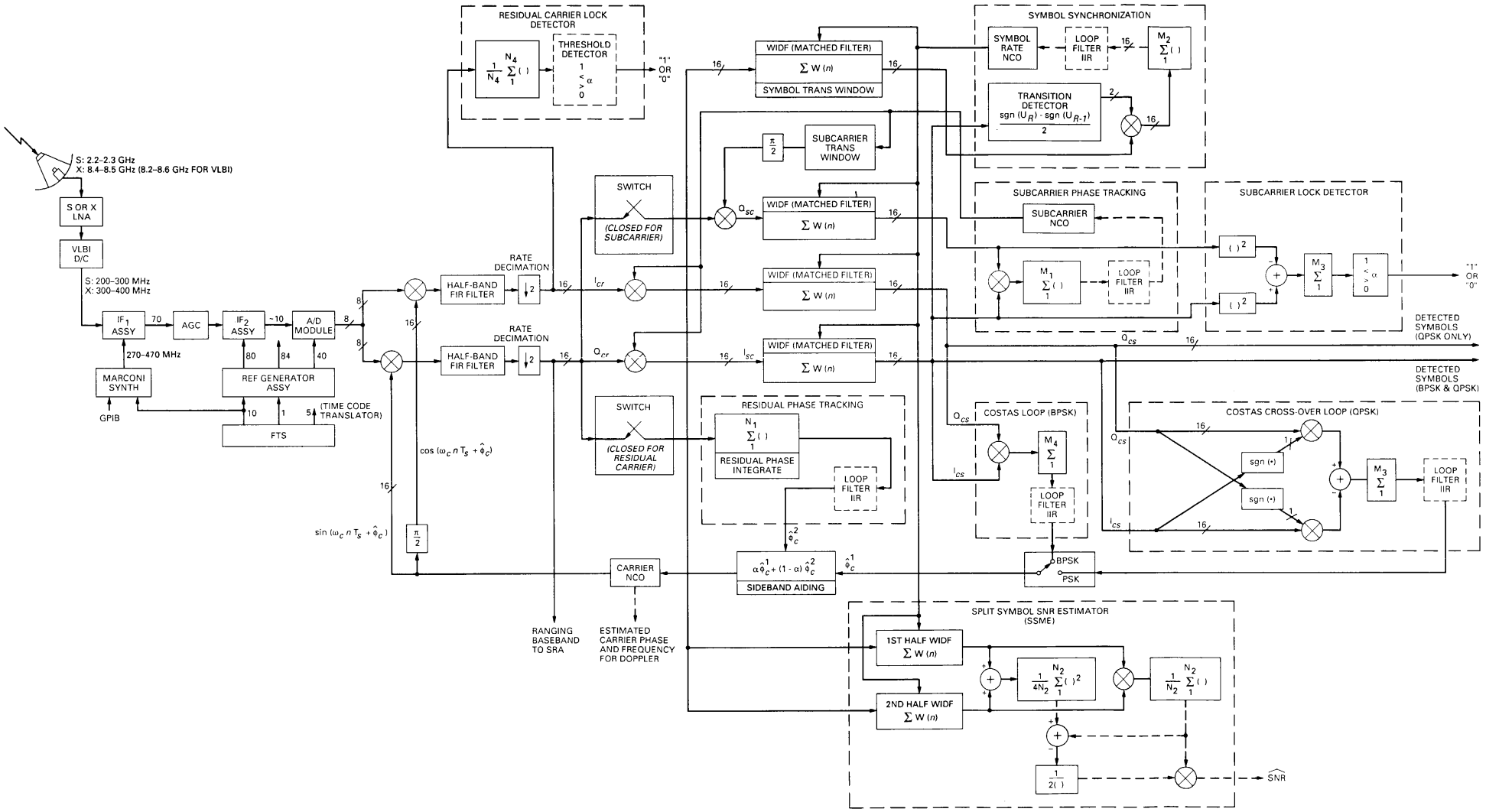

Fig. 3. ARX II functional block diagram.

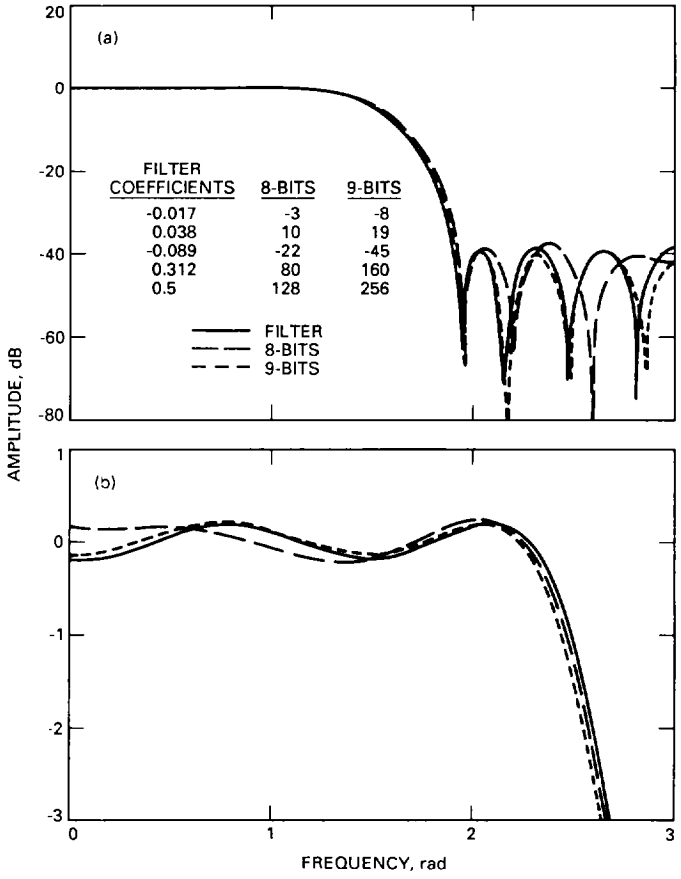

Fig. 4. Half-band filter: (a) amplitude response and (b) amplitude ripple.