

# Micro Communications and Avionics Systems First Prototype (MCAS1): A Low Power, Low Mass In Situ Transceiver

M. Agan,<sup>1</sup> A. Gray,<sup>1</sup> E. Grigorian,<sup>1</sup> D. Hansen,<sup>1</sup> E. Satorius,<sup>1</sup> and C. Wang<sup>1</sup>

*This article provides an overview of the communications system that is being developed as part of the Micro Communications and Avionics Systems (MCAS). The first phase (MCAS1) effort is being focused on a digital binary phase-shift-key (BPSK) system with both suppressed- and residual-carrier capabilities. The system is being designed to operate over a wide range of data rates from 1 kb/s to 4 Mb/s and must accommodate frequency uncertainties up to 10 kHz with navigational Doppler tracking capabilities. As such, the design is highly programmable and incorporates efficient front-end digital decimation architectures to minimize power consumption requirements. The MCAS1 design uses field programmable gate array (FPGA) technology to prototype the real-time MCAS1 communications system. Ultimately, this design will migrate to a radiation-hardened, application-specific integrated circuit (ASIC). Specific emphasis in this article is focused on the digital front end and BPSK demodulation portions of the MCAS1 receiver.*

## I. Introduction

The objective of the Micro Communications and Avionics Systems (MCAS) effort is to develop chip-level telecommunications systems to meet the unique needs of NASA's short-range, low-power, space and planet-surface communications. NASA is moving into an era of much smaller space exploration platforms that require low mass and power. This new era also is planning to incorporate in increasing numbers miniature rovers, probes, landers, aerobots, gliders, and multiplatform instruments, all of which have short-range communications needs (in this context, short range is defined as non-DSN links). Presently these short range (or in situ) communications needs are being met by a combination of modified commercial solutions (e.g., Sojourner) and mission-specific designs. The problem with commercial-based solutions is that they are high power, high mass, and single-application-oriented solutions that achieve low levels of integration and are designed for a benign operating environment. The problem with the mission-specific designs is that the resultant short-range communication systems do not provide the performance and capabilities to make their use for other missions desirable.

MCAS is primarily targeted at potential JPL users in the space exploration arena, such as the Mars Exploration Directorate (which can use this for various microspacecraft short-range communication links, such as an orbiter-lander, orbiter-rover, orbiter-microprobe, orbiter-balloon, and orbiter-sample return

---

<sup>1</sup> Communications Systems and Research Section.

canister), and multiple proposed Discovery missions (e.g., balloons, gliders, and probes). MCAS also has applicability to any space mission that has a short-range communications requirement, such as the International Space Station intravehicular and extravehicular wireless communications links [Codes U & M] and wireless sensor and short-range ground links. MCAS is a multiphase effort that will evolve over time and take advantage of advances in communications integrated circuit (IC) technology that will lead to increasingly more integrated solutions, with the eventual goal being the inclusion of microelectromechanical systems (MEMS) oscillators and filters onto single-chip transceivers.

The primary goal of MCAS1, which is the first prototype being developed under the MCAS effort, is to achieve a higher level of system integration at the chip level, thus allowing significant mass, power, and size reductions, at lower cost, for a broad class of very small platforms requiring short-range communications. Towards this end, a design approach has been devised that takes advantage of commercial IC advances when they are applicable to the space environment and utilizes custom design when performance and feature requirements dictate. The realization of this approach has resulted in the maximization of the transceiver functions performed in the digital domain. These digital functions initially will be implemented with field programmable gate array (FPGA) technology for purposes of real-time demonstration and testing. The final MCAS1 design then will incorporate the digital functions into an application-specific integrated circuit (ASIC). The resulting single digital ASIC then can be fabricated in a radiation-hardened process. The transceiver functions that must be in the analog domain consist primarily of the RF upconversion and downconversion. The approach with the RF subsystem design is to use space-qualified parts when available and leverage the large investment that industry has made in developing highly integrated devices for the commercial wireless markets. A space-qualified RF design will be developed through proper selection of these parts (i.e., selection of GaAs components for their inherent radiation hardness).

The functionality of MCAS1 is exhibited in the detailed block diagram in Fig. 1. At this point, the design is focused on the physical layer of the communications link, and it is assumed that any protocol is executed external to the MCAS1 transceiver board. Additionally, the antenna and diplexer, while allowed for in the design, are not included as part of MCAS1. Emphasis in this article will be focused primarily on the digital portion of the transceiver, including the data modulation process (Section II), the receiver front-end processing (Section III), and the demodulation process (Section IV). A complete description of the MCAS1 transceiver design is given in the *MCAS1 Design Document*.<sup>2</sup> In addition, the requirements driving the design can be found in the *Functional Requirements* document.<sup>3</sup>

## II. MCAS1 Data Encoding and Waveform Modulation

This section provides a description of the MCAS1 encoding process from the baseband input bits to the binary phase shift keying (BPSK) modulator. First, however, we note that to be compliant with the proposed Consultative Committee for Space Data Systems (CCSDS) proximity link recommendation, a V.35 scrambler/descrambler is incorporated into the MCAS1 transceiver for optional use with uncoded bit transmissions. The use of scrambling helps to ensure that a sufficient density of data transitions occurs in the transmitted data to aid in the bit-timing recovery at the receiver.

The next step after scrambling the transmit bit stream is the differential encoding of the bits. Due to the inherent phase ambiguity of the BPSK constellation, differential encoding can be utilized to transmit the difference in phases between consecutive bits rather than the actual bits themselves, thus obviating the need to determine the absolute phase at the receiver. The processing performed in the transmitter to implement the differential encoding is shown in Fig. 2 and is given by the following relationship:

---

<sup>2</sup> *Micro Communications and Avionics Systems: MCAS1 Design Document*, Draft (internal document), Jet Propulsion Laboratory, Pasadena, California, March 1999.

<sup>3</sup> D. Hansen, *Functional Requirements: MCAS1 UHF Transceiver*, Draft (internal document), Jet Propulsion Laboratory, Pasadena, California, December 1, 1998.

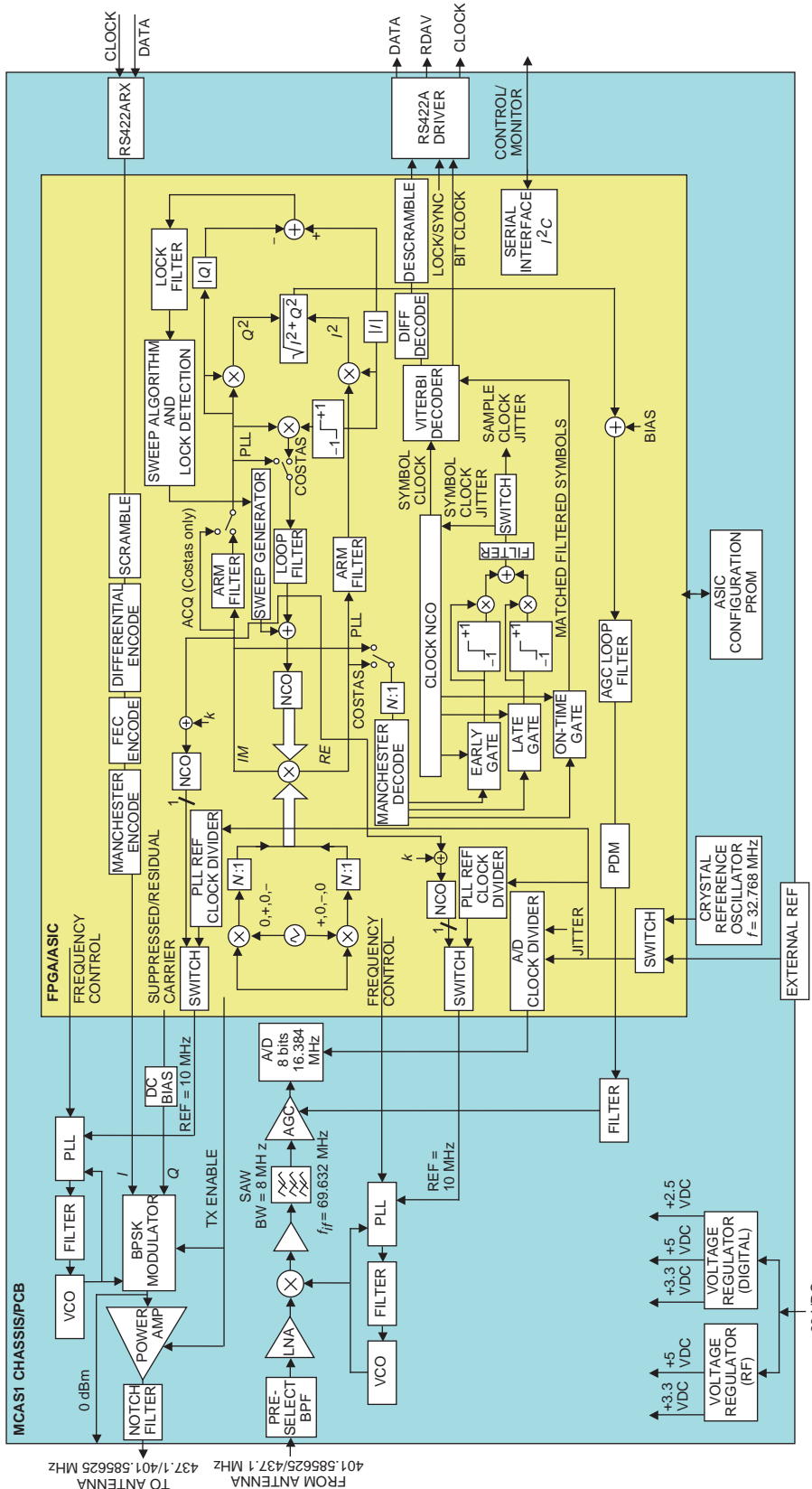


Fig. 1. The MCAS1 block diagram.

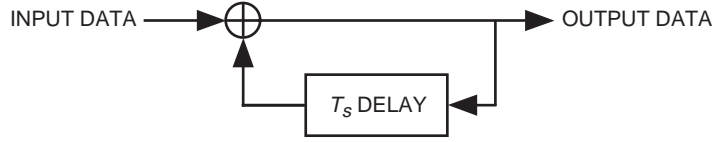


Fig. 2. The differential encoder block diagram.

$$y(k) = x(k) + y(k - 1)$$

where  $x(k)$  = the input,  $y(k)$  = the output, and both are logical 0 or 1.

The differential encoder/decoder implementation of MCAS1 was procured as part of the Viterbi decoder soft core acquired from Mentor Graphics [1]. Differential encoding may be enabled or disabled.

Following differential encoding is convolutional encoding to provide error detection and correction capability. The convolutional encoder is the optimal (in terms of free distance), constraint-length-7, rate-1/2 code as depicted in Fig. 3. The inverter is included to make the encoder compatible with the standard NASA  $K = 7, r = 1/2$  convolutional code and, by association, the proposed CCSDS proximity link recommendation. This inverter ensures there are transitions in the symbols when an all-zero bit pattern is input to the encoder. The inverter may be switched out of the circuit if desired. Two symbols are generated for each input bit into the encoder; consequently, the channel symbol rate is twice the input bit rate. The actual implementation of the convolutional encoder is in the form of soft core acquired from Mentor Graphics [1]. The convolutional encoding may be disabled when uncoded operation is desired.

Normally, a square-wave pulse shape is transmitted with an associated nonreturn-to-zero-level (NRZ-L) waveform, i.e., the binary 1/0 output from the convolutional encoder is routed directly to the phase modulator. When required, the transmitter can be set to Manchester encode the transmitted symbols. Manchester encoding (also known as biphase-level) represents a binary one as a one for the first half of the bit period and a zero for the second half of the bit period. Manchester encoding a zero translates to a zero during the first half of the bit period and a one for the second half of the bit period. Because of its spectral shape, Manchester encoding generally will be enabled when residual-carrier modulation is utilized to prevent the modulated data from interfering with the performance of the receiver carrier-tracking and data-detection circuits (see Section IV).

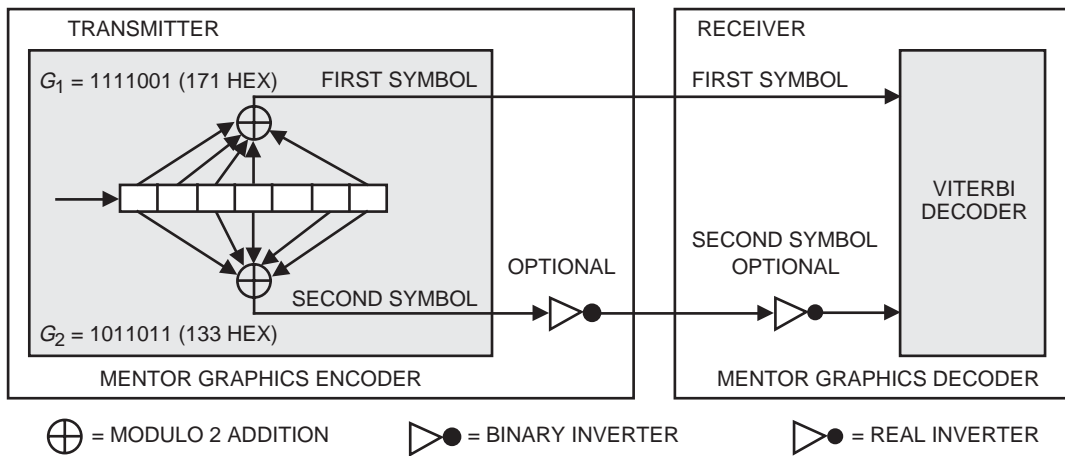
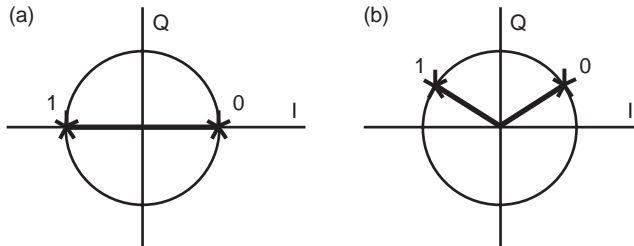


Fig. 3. The convolutional encoder (rate 1/2, 171, and 133 generators).

After Manchester encoding, the encoded baseband data are used to phase modulate the carrier. The required output from the MCAS1 transceiver is a phase-modulated waveform centered at a frequency of 437.1 MHz or 401.585625 MHz that is the input to the diplexer or antenna. This BPSK modulation is achieved through the use of a phase-modulator device that will have as inputs a 437.1-MHz or 401.585625-MHz analog carrier and the encoded baseband data. The output is either a 437.1-MHz or 401.585625-MHz phase-modulated waveform.<sup>4</sup> The output waveform is geometrically described by the signal constellation, as illustrated in Fig. 4. This constellation depicts the phase of the output signal when it is translated to baseband. For BPSK modulation, a logical zero is mapped into a phase of zero radians, and a logical one is mapped into a phase of  $\pi$  radians. As indicated in Fig. 4, the modulator will have the capability to transmit either a suppressed carrier or a residual carrier with a 57-deg modulation index.

The BPSK-modulated transmit signal is amplified by the power amplifier, which nominally transmits 500 mW. Following the power amplifier, a notch filter centered at the receive frequency is utilized to attenuate transmit spurious signals in the receive band to ensure that the spur power level is well below the receiver sensitivity. The 500-mW power amplifier also can be used to drive a higher power amplifier if required by the operational scenario. The design is specified to accommodate a transmit power amplifier of up to 10 W (i.e, a 10-W power amplifier is baselined that can be driven by 500 mW, and the allowed receive spur-level specification must be met for a 10-W transmit level).



**Fig. 4. The phase modulator transmitted signal constellation: (a) suppressed carrier and (b) residual carrier.**

### III. MCAS1 Receiver Front-End Processing

In this section, we describe the MCAS1 receiver front-end. With reference to Fig. 1, this comprises the automatic gain control (AGC), the analog-to-digital converter (ADC), and the digital downconverter/decimator. These are described separately in this section.

#### A. AGC

As indicated in Fig. 1, the AGC controls the voltage level input to the ADC based on a control-voltage signal generated digitally in the FPGA/ASIC (described below). The AGC amplifier provides a 60-dB dynamic range with a typical transfer curve, as depicted in Fig. 5. As is seen, the gain is approximately linear over the control-voltage range from 1.5 to 3.5 volts. For AGC control-voltage levels less than 1.5 volts, the AGC gain saturates at 30 dB (weak-input-signal limit) whereas, for control-voltage levels above 3.5 volts, the AGC gain limits at approximately  $-30$  dB (strong-signal limit). In the latter case, input-signal levels from the IF filter that exceed the AGC dynamic range will cause the ADC to saturate, thereby creating clipping distortion and, thus, forcing the ADC to approach the 1-bit performance limit.

Insofar as the ADC input dynamic range is concerned, the 60-dB AGC dynamic range is more than sufficient. Specifically, the input received-signal level can vary over a 70-dB range from  $-140$  dBm to

<sup>4</sup> Though the transmit synthesizer is required to operate only at the 437.1-MHz and 401.585625-MHz frequencies as implemented, it will be tuneable within the range from 400 MHz to 445 MHz to allow the use of the MCAS1 transceiver in frequency-division multiple-access (FDMA) scenarios.

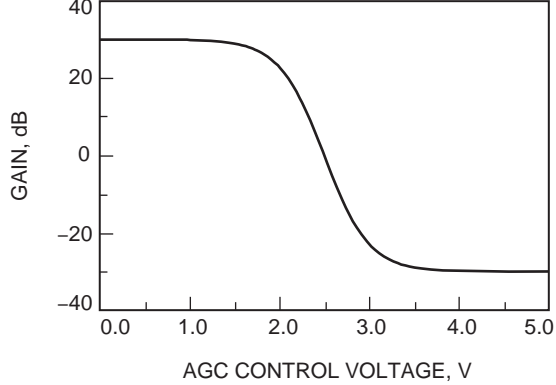


Fig. 5. The AGC transfer curve.

−70 dBm, depending on the data rate, transmitter–receiver range, etc.<sup>5</sup>. However, because of the wide IF filter bandwidth (6.5 MHz), the corresponding variation in total input power to the ADC is only approximately 30 dB: −103 dBm to −70 dBm, which could readily be maintained by the AGC *if only* the AGC were being controlled to maintain the ADC input power at a fixed level. In actuality, the AGC is being controlled to maintain a constant power level at the output of the Costas arm filters, where the bandwidth is generally much narrower than the IF filter bandwidth.

In particular, the AGC is based on a single feedback control-loop design with the AGC control voltage extending back from the Costas arm-filter outputs, as indicated in Fig. 1. The digital AGC error signal,  $E_{AGC}$ , is generated from the Costas arm-filter outputs,  $I$  and  $Q$ , via

$$E_{AGC} = K_{\text{gain}} \times \left(1 - \sqrt{I^2 + Q^2}\right) \quad (1)$$

where  $K_{\text{gain}}$  controls the time constant of the AGC as well as the variance of the resulting amplitude gain estimate. Typically,  $K_{\text{gain}} = 10^{-4}$  provides a reasonable compromise between a fast AGC response time and a low-noise gain estimate.<sup>6</sup> The particular error signal of Eq. (1) is chosen such that the AGC forces the complex magnitude of the Costas arm-filter outputs,  $\sqrt{I^2 + Q^2}$ , to be unity on average.<sup>7</sup> This in turn helps to regulate the Costas-loop bandwidth over a reasonably wide range of input signal levels (to be discussed in further detail). For the purpose of simplifying the implementation, the complex magnitude of the Costas arm-filter outputs may be approximated by [2]:

$$\sqrt{I^2 + Q^2} \approx \begin{cases} |I| + 0.375 \times |Q|, & \text{if } |I| > |Q| \\ |Q| + 0.375 \times |I|, & \text{if } |Q| > |I| \end{cases} \quad (2)$$

Calculations show that the relative error associated with this approximation is less than 7 percent.<sup>8</sup>

The error signal in Eq. (1) is integrated in the AGC loop filter, i.e.,

<sup>5</sup> D. Hansen, Section 4.1.3.4, op. cit.

<sup>6</sup> E. Satorius and C. Wang, “MCAS Receiver AGC Design Considerations,” JPL Interoffice Memorandum (internal document), Jet Propulsion Laboratory, Pasadena, California, June 4, 1999.

<sup>7</sup> This is approximately the same (within 1 dB) as forcing the rms power,  $\sqrt{\langle I^2 + Q^2 \rangle}$ , to unity, i.e., in the case of independent complex Gaussian samples,  $\sqrt{\langle I^2 + Q^2 \rangle} / \sqrt{\langle I^2 + Q^2 \rangle} \approx 1.13$  (1.05 dB). This correction factor can be incorporated into the error reference voltage, i.e., replace 1 by 1/1.13 in Eq. (1).

<sup>8</sup> E. Satorius and C. Wang, op. cit.

$$V_{\text{out}} = V_{\text{out}} + E_{\text{AGC}} \quad (3)$$

and the magnitude of the result,  $|V_{\text{out}}|$ , is used to generate the AGC gain,  $K_{\text{AGC}}$ , via the nonlinear transfer curve,  $f(\cdot)$ , illustrated in Fig. 5, i.e.,

$$K_{\text{AGC}} \text{ (dB)} = f(|V_{\text{out}}|) \quad (4)$$

This gain is then used to scale the AGC input.

A critical issue with this approach is the impact of the AGC on the operation of the ADC as well as the internal digital arithmetic implemented in the FPGA/ASIC. As will be discussed in Section III.B, ideally the input ADC voltage is scaled to achieve an optimal trade-off between ADC quantization noise and clipping distortion. In contrast, the AGC loop attempts to maintain the complex magnitude of the Costas arm-filter outputs to be unity on average. Thus, there is no guarantee that this criterion of unity rms Costas arm-filter outputs will enable the ADC to operate at its optimal input scaling (loading) point or even prevent the ADC from saturating.

To alleviate this situation, fixed gains are distributed throughout the digital data paths.<sup>9</sup> These gains are programmable, dependent upon the data rate and the digital decimation factor (see Section III.C), and are used for purposes of minimizing the effects of digital quantization noise and saturation. Denoting the product of these fixed gains by  $K_F$ , we find that in steady state,<sup>10</sup>

$$K_{\text{AGC}} \approx \frac{1}{K_F \sqrt{\alpha_s P + \alpha_n N_0 B_{\text{IF}}}} \quad (5)$$

where  $P$  denotes the input signal power to the ADC;  $\alpha_s$  represents the fraction of this power reaching the output of the Costas arm filters;  $N_0$  is the noise spectral level at the IF filter; and  $\alpha_n$  represents the fraction of the input noise power,  $N_0 B_{\text{IF}}$ , that reaches the Costas arm-filter outputs.

By appropriate choice of  $K_F$ , it has been shown that the AGC gain can be expressed as<sup>11</sup>

$$K_{\text{AGC}} = K_{\text{ADC}}^* \times \hat{K}_{\text{AGC}} \quad (6)$$

where  $K_{\text{ADC}}^*$  denotes the optimal ADC loading point in the small signal limit and  $\hat{K}_{\text{AGC}}$  is the normalized AGC gain, which is always less than unity. In this way, the AGC gain always scales the ADC input to its optimal loading point in the small signal limit and, as the signal power increases, the AGC gain decreases from  $K_{\text{ADC}}^*$  by the factor  $\hat{K}_{\text{AGC}}$ , thereby avoiding saturation at least until the AGC dynamic range is exceeded (see Fig. 1).

Plots of  $\hat{K}_{\text{AGC}}$ , dB, versus the symbol energy-to-noise spectral level,  $E_s/N_0$ , are presented in Fig. 6 for different data rates,  $R_s$ , assuming (1) a 0-dB threshold  $E_s/N_0$ ; (2) an IF filter bandwidth  $B_{\text{IF}} = 6.5$  MHz; and (3) a 6-bit ADC. As  $E_s/N_0$  increases above the threshold, the AGC will automatically scale the ADC input such that its rms level is always less than or equal to the small-signal optimal loading point. Given that the optimal loading point increases with the input SNR, this implies the ADC will always be operating below its optimal point where the quantization noise power increases more gradually as a function of the ADC loading point (see Section III.B).

<sup>9</sup> Ibid.

<sup>10</sup> Ibid.

<sup>11</sup> Ibid.

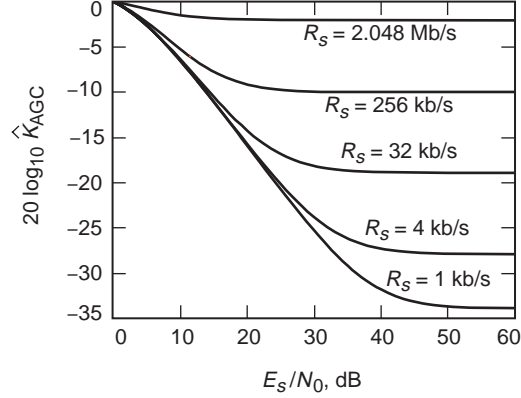


Fig. 6.  $\hat{K}_{\text{AGC}}$ , dB, versus  $E_s/N_0$  for different data rates.

As the input-signal power level increases,  $\hat{K}_{\text{AGC}}$  continues to decrease until the signal power becomes commensurate with the total input noise power,  $N_0 B_{\text{IF}}$ . Beyond this,  $\hat{K}_{\text{AGC}}$  asymptotes to a level dependent upon the data rate—the higher the data rate, the larger the asymptotic level of  $\hat{K}_{\text{AGC}}$ . The lower data rates impose the most severe constraint on the *usable* AGC dynamic range. For example, when  $R_s = 1$  kb/s,  $\hat{K}_{\text{AGC}}$  asymptotes to approximately  $-33$  dB or, equivalently, the AGC gain  $K_{\text{AGC}}$  decreases to 33-dB below the optimal small-signal loading point of the ADC. Referring to Section III.B (Fig. 10), this approximately matches the dynamic range of a 6-bit ADC, i.e., at 30-dB below the optimal small-signal loading point of a 6-bit ADC, the quantization noise power equals the total input power.

To be conservative, we define the worst-case AGC dynamic range corresponding to the lowest symbol rates, 1–4 kb/s, to be the point at which  $K_{\text{AGC}}$  falls to 25-dB below the optimal small-signal loading point for a 6-bit ADC. With reference ahead to Fig. 10, this corresponds to an input-to-quantization noise power ratio of 5 dB for a 6-bit ADC and, with reference to Fig. 6, this also corresponds to a signal-power dynamic range,  $E_s/N_0$ , of approximately 30 dB at either  $R_s = 1$  or 4 kb/s. This (30 dB) is sufficient to cover the signal dynamic range due to transmitter–receiver range or antenna-pattern variations. Note from Fig. 6 that, at the larger data rates, e.g., 32 kb/s and above, the AGC gain remains well within the ADC dynamic range for all values of  $E_s/N_0$ , and, thus, the maximum allowable signal dynamic range in these cases matches the entire AGC dynamic range of 60 dB. Large signals exceeding this dynamic range ultimately will cause clipping distortion at the ADC.

In addition to maintaining linearity in the receiver front-end, the AGC serves to regulate the increase in Costas-loop bandwidth,  $B_L$ , with the input signal level. Based on the analysis presented in [3, Chapter 3],<sup>12</sup> plots of the Costas-loop bandwidth expansion,  $B_L/B_{L0}$  ( $B_{L0}$  denotes the Costas-loop bandwidth at threshold), are plotted versus  $E_s/N_0$  in Fig. 7, with and without the AGC.

These data are generated based on the following assumptions: (1) a 0-dB threshold  $E_s/N_0$  and (2) an IF filter bandwidth  $B_{\text{IF}} = 6.5$  MHz. As is seen, the AGC limits the Costas-loop bandwidth expansion to a factor of approximately 1.4 over the range  $0 \text{ dB} < E_s/N_0 < 20 \text{ dB}$ . This represents a dramatic improvement over the case of operation without the AGC. Furthermore, the rate of expansion with the AGC is much slower than the rate of increase of the carrier-tracking loop SNR, which is proportional to  $(E_s/N_0)/B_L$  [3, Chapter 3]. With reference to Fig. 7, it can be seen that, with the AGC,  $(B_L/B_{L0})/(E_s/N_0) < 1$  and approaches 0 as  $E_s/N_0$  increases indefinitely. Thus, the carrier-tracking loop SNR will continue to increase above its threshold level due to the limiting action of the AGC. Without the AGC,  $(B_L/B_{L0})/(E_s/N_0)$  also approaches zero with increasing  $E_s/N_0$ , but at a slower rate, thereby resulting in a degraded carrier-tracking loop SNR relative to the AGC implementation.

<sup>12</sup> Ibid.



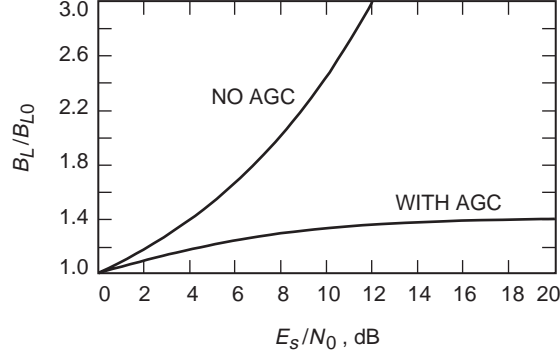


Fig. 7.  $B_L/B_{L0}$  versus  $E_s/N_0$ .

## B. ADC

The MCAS1 receiver employs first-order, bandpass sampling wherein the IF frequency band is mapped directly down to digital baseband. Denoting the IF and ADC sampling frequencies by  $f_{\text{IF}}$  and  $F_s$ , respectively, then as long as the frequency band,  $f_{\text{IF}} - F_s/4 \leq f \leq f_{\text{IF}} + F_s/4$ , coincides with one of the image bands,  $kF_s/2 \leq f \leq (k+1)F_s/2$ , for some integer  $k$ , the input at  $f_{\text{IF}}$  will be mapped into the baseband interval,  $0 \leq f \leq F_s/2$ , as a result of bandpass sampling. This leads to the following condition on  $f_{\text{IF}}$  and  $F_s$ :

$$f_{\text{IF}} = (2n + 1) \times \frac{F_s}{4} \quad (7)$$

where  $n$  is a positive integer. Choosing  $f_{\text{IF}}$  and  $F_s$  to satisfy Eq. (7) guarantees that the IF frequency will be mapped down to the center of the Nyquist band, i.e., down to  $F_s/4$ . Furthermore, to maintain the lowest possible ADC sample rate and avoid aliasing, it is desirable that  $F_s$  just exceed twice the IF filter bandwidth,  $2B_{\text{IF}}$ .

When the bandpass sampling system was designed, the ADC sampling rate was chosen to accommodate an integral, power-of-two number of samples per symbol at all symbol rates: 4.096 Msym/s, 2.048 Msym/s, ..., 1 ksym/s. To achieve a minimum of 4 samples per symbol at the highest symbol rate of 4.096 Msym/s, an ADC sampling rate of  $F_s = 16.384$  MHz was chosen. Given  $F_s$ , admissible IF frequencies are obtained from Eq. (7). Based on the availability of IF filters as well as other considerations,<sup>13</sup> an IF near 70 MHz is desired. The closest admissible IF frequency satisfying Eq. (7) occurs when  $2n + 1 = 17$ , corresponding to  $f_{\text{IF}} = 17 \times 4.096$  MHz = 69.632 MHz. To accurately bandpass sample at this IF, the ADC full-power bandwidth must be at least 150 MHz with a corresponding sampling aperture uncertainty (jitter) of approximately 5 ps.<sup>14</sup> Based on a prior analysis,<sup>15</sup> the impact of this jitter on the carrier-loop tracking performance is negligible.

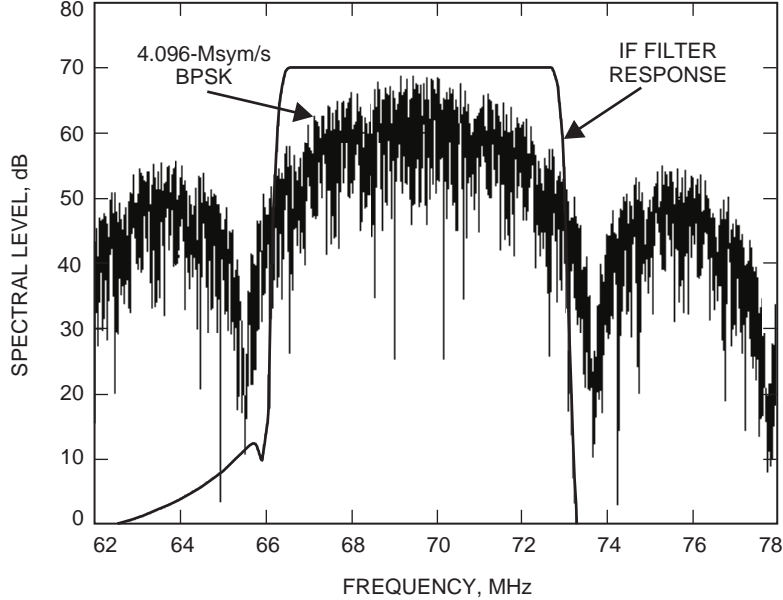
As noted above, the IF filter bandwidth must be less than the Nyquist frequency,  $F_s/2 = 8.192$  MHz, or else significant aliasing will occur. An IF filter bandwidth of  $B_{\text{IF}} = 6.5$  MHz is chosen,<sup>16</sup> which eliminates any possibility of aliasing at the cost of some performance degradation at only the highest symbol rates. To illustrate this, a unit-amplitude, 4.096-Msym/s BPSK signal is synthesized at  $f_{\text{IF}} = 69.632$  MHz. A sample spectral plot is presented in Fig. 8. Superimposed on the BPSK signal spectrum is the IF bandpass

<sup>13</sup> *Micro Communications and Avionics Systems: MCAS1 Design Document*, op. cit.

<sup>14</sup> These specifications are based on data provided in [4].

<sup>15</sup> "MCAS1 Conceptual Design Review RFA's," JPL Interoffice Memorandum (internal document), Jet Propulsion Laboratory, Pasadena, California, December 17, 1998.

<sup>16</sup> *Micro Communications and Avionics Systems: MCAS1 Design Document*, op. cit.



**Fig. 8. Typical filter response relative to a 4.096-Msym/s BPSK signal at 69.632-MHz IF.**

filter response corresponding to a 6.5-MHz bandwidth. As is seen, a small portion of the main lobe is attenuated, resulting in a signal-power loss of approximately 0.5 dB. However, this filter will prevent aliasing, as indicated in Fig. 9, where spectral plots of the IF-filtered BPSK signal are presented before and after bandpass sampling. As is seen, the bandpass-filtered portion of the BPSK signal remains intact after the bandpass sampling operation.

In addition to the ADC sample rate, the number of bits must be considered. In Fig. 10, plots are presented of the ADC quantization noise-to-input-power ratio, in dB, versus the input scaling or “loading” factor for a 4-bit, 6-bit, and 8-bit ADC at different input signal-to-noise-power ratios (SNRs). As is seen, there is always an optimal loading point for each size of ADC. Above this point (closer to the 0-dB loading factor), clipping distortion limits ADC performance, whereas below this point, ADC quantization noise is the limiting factor. So, for best results with an 8-bit ADC at either  $-40$  dB or  $-15$  dB SNR, the input should be scaled such that its rms level is approximately  $-12$  dB relative to the ADC full-scale voltage. For a 6-bit ADC at either  $-40$  dB or  $-15$  dB SNR, the optimal loading point is approximately  $-10$  dB relative to full scale, and, for a 4-bit ADC at either  $-40$  dB or  $-15$  dB input SNR, it is about  $-8$  dB relative to full scale. As the input SNR increases to 10 dB, the optimal loading point increases, depending upon the number of ADC bits.

As the number of ADC bits increases, the quantization noise (at the optimal loading point) decreases correspondingly. The implications of this on system performance can be conveniently expressed in terms of the SNR degradation resulting from the ADC (assuming operation at or near the optimal loading point):

$$\frac{\overline{\text{SNR}}}{\text{SNR}} = 1 + P_Q \frac{(1 + \text{SNR})}{\ell^2} \quad (8)$$

where  $\overline{\text{SNR}}$  denotes the ADC output SNR;  $P_Q = 2^{-2(b-1)}/12$  is the variance of the ADC quantization noise ( $b$  denotes the number of ADC bits); and  $\ell$  is the loading factor. As is seen, SNR increases linearly with SNR (no degradation) until the quantization noise becomes commensurate with the input noise, after which point the degradation starts to become proportional to the input SNR.

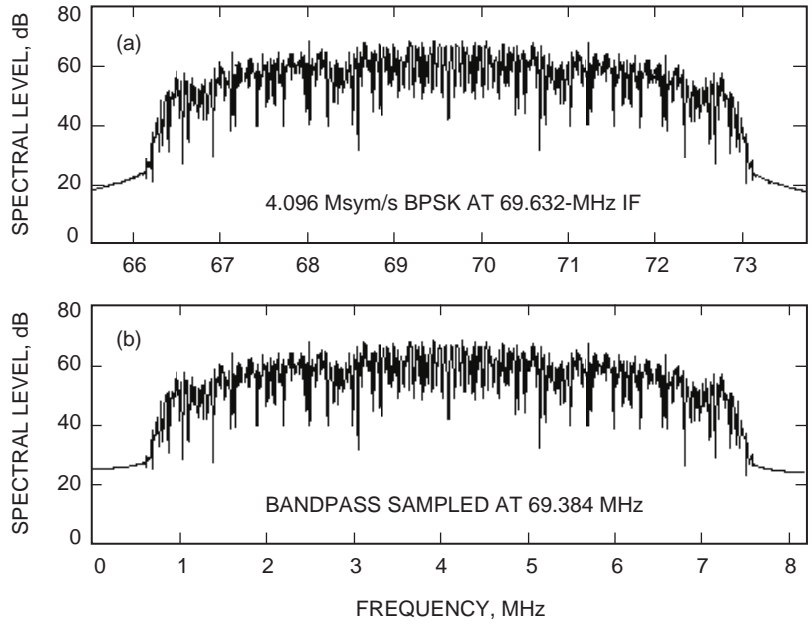


Fig. 9. BPSK spectra (a) before bandpass sampling at 16.384 MHz and (b) after bandpass sampling at 16.384 MHz.

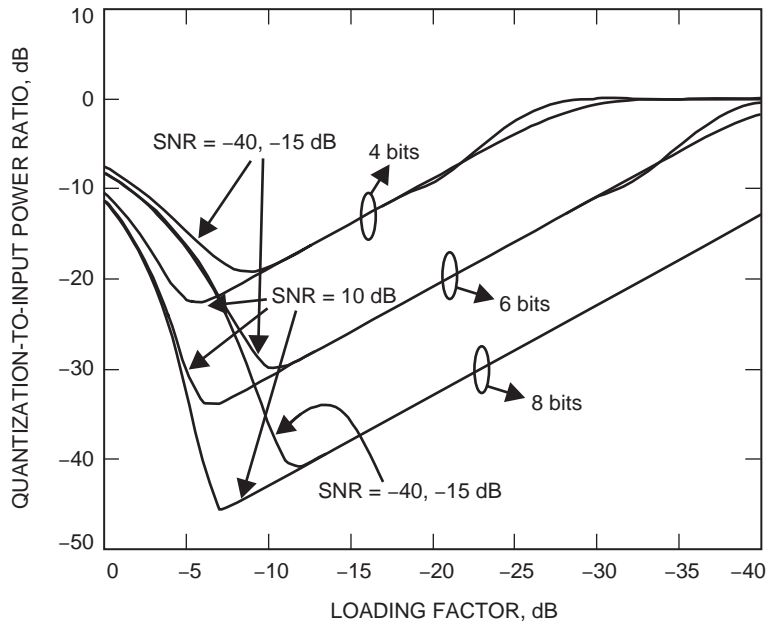


Fig. 10. Quantization noise-to-input power ratio, dB.

SNR degradation in dB,  $10 \log_{10}(\text{SNR}/\overline{\text{SNR}})$ , is tabulated in Table 1 over the input SNR range from  $-40$  dB to  $10$  dB. As is seen, either a 6-bit or 8-bit ADC provides negligible degradation ( $<0.1$  dB) over the SNR range of interest; however, as the number of ADC bits is decreased to 4, there is noticeable degradation ( $0.241$  dB) at the upper end ( $10$  dB) of the input SNR range. By way of comparison, a 1-bit ADC incurs a loss of at least  $2$  dB over the entire SNR range.

**Table 1. SNR degradation resulting from the ADC.**

ADC input SNR, dB	SNR degradation, dB			
	8 bits	6 bits	4 bits	2 bits
-40	0.0004	0.004	0.045	0.431
-15	0.0004	0.005	0.046	0.444
10	0.0012	0.019	0.241	2.367

Since both the ADC quantization noise and the input noise are spectrally white out to the Nyquist frequency, the above SNR degradation translates directly into an identical degradation in the symbol-energy-to-noise spectral density ratio,  $E_s/N_0$ , regardless of the data rate. The only impact of data rate, for a given  $E_s/N_0$ , is on the input ADC SNR via

$$\text{SNR} = 2 \frac{E_s/N_0}{F_s T_s} \quad (9)$$

where  $F_s = 16.384$  MHz, as determined above, and  $R_s \equiv 1/T_s$  is the symbol rate. For typical MCAS1 operational scenarios, the product,  $F_s T_s$ , will vary between 4 (corresponding to the highest symbol rate of 4.096 Msym/s) and  $2^{14}$  (corresponding to the lowest symbol rate of 1 ksym/s). Over the  $E_s/N_0$  range of interest, 0 dB to 10 dB, SNR ranges between approximately -40 dB and 10 dB, which corresponds to the range presented in Table 1.

The SNR degradations tabulated in Table 1 should be compared with other system degradations—most notably, the system noise figure, which is specified at 3-dB nominal.<sup>17</sup> Consequently, it is highly desirable to maintain the ADC SNR degradation to significantly less than 3 dB. This rules out a 1-bit ADC (2-dB degradation) and, with reference to Table 1, this also rules out a 2-bit ADC converter. Even a 4-bit ADC is problematical for two reasons: (1) there is nearly a 0.25-dB degradation at the high end of the SNR range which, when combined with additional digital implementation losses, approaches the 3-dB system noise figure specification and (2) any additional dynamic range, e.g., to accommodate radio frequency interference (RFI), will significantly increase the 4-bit ADC output SNR degradation—especially at the high end of the input SNR range. So, for example, if an in-band RFI is present (assume sinusoidal) with an interference-to-noise-power ratio (INR) of 20 dB and a 4-bit ADC is used, then the optimal loading point will be near 0 dB (see Fig. 10) and, using Eq. (8) and assuming the input SNR is 10 dB, then the ADC SNR degradation increases from 0.24 dB to almost 0.6 dB. Based on all of these considerations, the most reasonable choice for MCAS1 implementation is either the 6- or 8-bit ADC.

### C. Digital Downconversion and Decimation

Digital downconversion and decimation directly follow the ADC. Based on a study,<sup>18</sup> which examined both single- and dual-channel analog/digital downconversion schemes, a digital complex baseband downconversion scheme was chosen as the preferred design approach from the standpoint of computational efficiency and flexibility. This approach is depicted in Fig. 11 and comprises (1) digital complex mixing from  $F_s/4 = 4.096$  MHz down to baseband, followed by (2) digital decimation via a first-order, cascaded integrator-comb (CIC) filter [5,6]. Note that the digital mixing functions do not require multiplication and, furthermore, the CIC filters are multiplierless and, thus, the entire structure can be implemented

<sup>17</sup>D. Hansen, op. cit.

<sup>18</sup>E. Satorius, "Candidate MCAS Receiver Front-End Architectures and Issues," JPL Interoffice Memorandum (internal document), Jet Propulsion Laboratory, Pasadena, April 27, 1998.

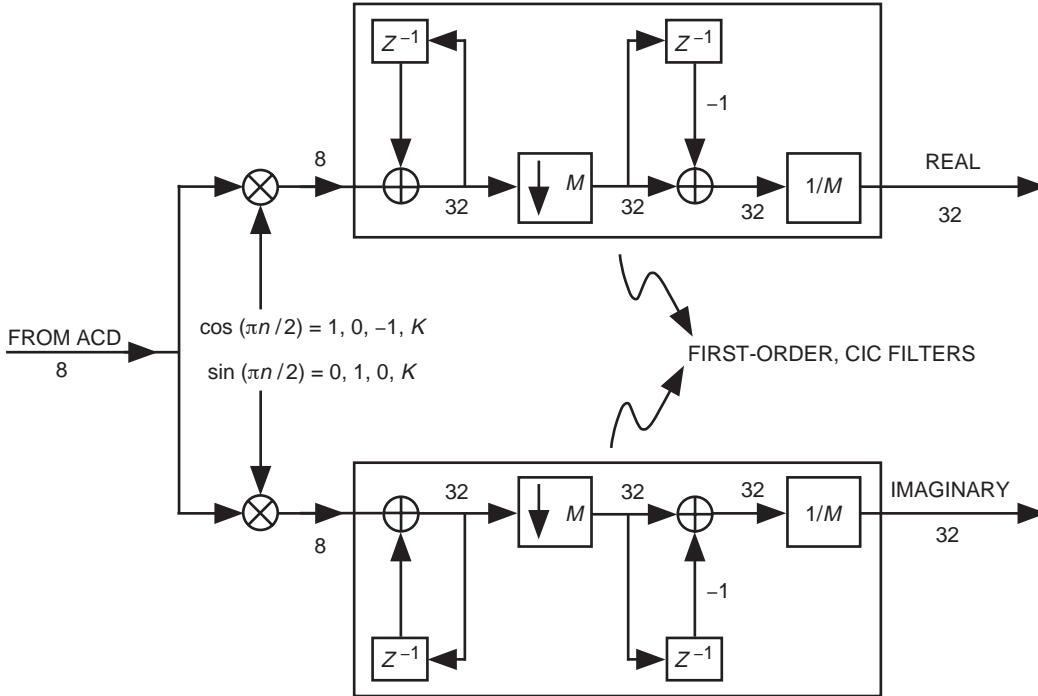


Fig. 11. Digital complex basebanding and decimation.

efficiently in the FPGA/ASIC. Also indicated in Fig. 11 are the typical data bit widths used to implement the digital downconverter and decimator (all data are represented in two's complement notation and all indicated data bit widths include the sign bit).

The decimation factor,  $M$ , is programmable and is dependent upon the input data rate. To accommodate symbol-timing recovery (Section IV),  $M$  typically is chosen so that there are at least 16 samples per symbol after decimation, except at the highest data rates. So, at 1.024 Msym/s, 2.048 Msym/s, or 4.096 Msym/s,  $M$  will nominally be set to 1 (no decimation), in which case the remainder of the digital receiver (Costas loop, symbol-timing recovery, etc.) will run at the input sampling rate, 16.384 MHz. As the data rate is lowered below  $R_s = 1.024$  Msym/s down to 8 ksym/s,  $M$  is increased proportionately such that

$$\frac{F_s}{R_s \times M} = 16 \quad (10)$$

Below  $R_s = 8$  ksym/s,  $M$  remains fixed at 128 to accommodate Doppler offsets. Note that, as  $M$  increases up to 128, more of the input noise to the ADC is filtered out by the CIC filters, thereby reducing the total CIC output power. This necessitates a rescaling operation after the CIC filters, as described in Section IV.

Since there are typically at least 16 samples per symbol after decimation, the effects of the CIC decimation filter on system performance are negligible. This is indicated in Fig. 12, where a unit-amplitude, 256-ksym/s BPSK signal spectrum is presented along with the CIC-filter response corresponding to  $M = 4$  [from Eq. (10)]. Although there is appreciable CIC-filter droop out to the decimated Nyquist frequency, 2.048 MHz, the actual degradation in  $E_s/N_0$  is less than 0.05 dB—a small price to pay for such a computationally efficient, digital decimator.

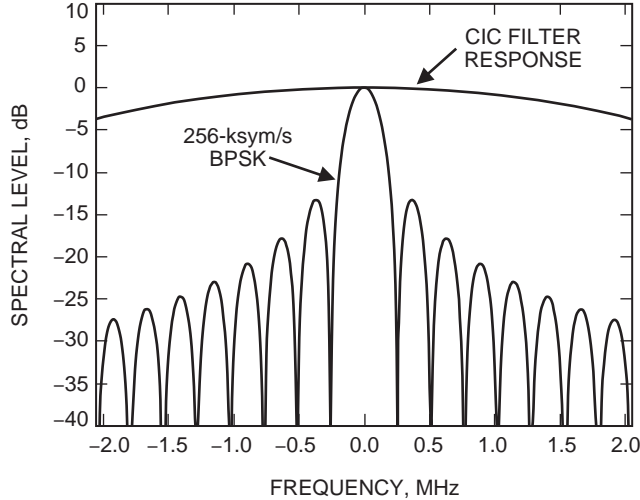


Fig. 12. CIC filter response relative to a 256-ksym/s BPSK signal decimated by  $M = 4$ .

## IV. MCAS1 Demodulation

In this section, the various elements of the demodulation process are presented, including the carrier-recovery loop (Section IV.A); the Doppler-frequency extraction for navigation (Section IV.B); the symbol-timing recovery (Section IV.C); the convolutional decoder (Section IV.D); the differential decoder (Section IV.E); and the descrambler (Section IV.F).

### A. Carrier-Tracking Loop

The carrier-tracking loop portion of the MCAS1 transceiver is designed to acquire and track the phase of the received signal. The signal can be suppressed-carrier BPSK, residual-carrier BPSK with a modulation index of  $\pi/3$ , or unmodulated. The carrier-tracking loop should operate for all of the required symbol rates from 1 ksym/s to 4 Msym/s, signal-to-noise ratios, and CIC-filter decimated sampling rates. It needs to track the carrier phase reliably when the received signal strength varies over many orders of magnitude. The tracking-loop bandwidth is programmable from 10 Hz to 10 kHz to meet the tracking and acquisition requirements for various communications scenarios. In addition, the carrier-tracking loop needs to acquire and track received signals with maximum frequency offsets of  $\pm 10$  kHz when the received signal is at 400 MHz (UHF) and  $\pm 50$  kHz when the received signal is at 2 GHz (S-band). The tracking loop also can support navigation by supplying the instantaneous phase of the received signal.

Figure 13 shows the block diagram of the MCAS1 carrier-tracking loop. The loop follows the ADC, digital downconverter, and the CIC decimation filter. The complex baseband loop input is multiplied by the complex output of the numerically controlled oscillator (NCO). The product of the complex multiplication is split into the real and the imaginary data paths. The signal path following the real output is termed the real arm of the carrier-tracking loop, and the path following the imaginary output is the imaginary arm of the loop.

Both the real and imaginary signals are filtered by a pair of identical low-pass arm filters,  $G(f)$ , with a programmable cut-off frequency. After the arm filters, one or both of the arm-filter outputs are used to form the input to the loop filter,  $F(f)$ , depending on whether the tracking loop is operated in the Costas loop or the residual carrier-tracking mode (labeled PLL mode in Fig. 13). There are three switches (SWs) in the MCAS carrier-tracking loop. SW1 and SW2 are selected depending on whether the loop is



$$\text{LPF}(r(t) \times e^{-j\omega t}) = \frac{\sqrt{2}}{2} \left[ \sqrt{P} a_k \cos \theta + N_c \cos \theta - N_s \sin \theta \right] + j \frac{\sqrt{2}}{2} \left[ \sqrt{P} a_k \sin \theta + N_c \sin \theta + N_s \cos \theta \right] \quad (13)$$

The signal then is multiplied by  $e^{-j\hat{\theta}}$ , where  $\hat{\theta}$  is the carrier-tracking loop estimate of  $\theta$ . The product of the complex multiplication is

$$\begin{aligned} \text{LPF}(r(t) \times e^{-j\omega t}) \times e^{-j\hat{\theta}} &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} a_k \cos \varphi + N_c \cos \varphi - N_s \sin \varphi \right] \\ &+ j \frac{\sqrt{2}}{2} \left[ \sqrt{P} a_k \sin \varphi + N_c \sin \varphi + N_s \cos \varphi \right] \end{aligned} \quad (14)$$

where  $\varphi = \theta - \hat{\theta}$ . We note that, since the tracking loop operates in baseband, there are no double frequency terms. When the loop is locked, i.e.  $\varphi \approx 0$ , the received data,  $a_k$ , can be recovered from the real part of the product. The real and the imaginary parts of the product are identical to the outputs of the in-phase and quadrature-phase detectors of a conventional passband Costas loop except for the constant coefficient  $\sqrt{2}/2$ , which arises from the approximate removal of the double frequency term by the CIC filter and/or the Costas low-pass arm filters. With proper scaling, the complex implementation of the MCAS1 carrier-tracking loop has performance identical to a conventional passband Costas loop with its NCO operating at the carrier frequency,  $\omega$  [8].

When the received signal is a suppressed-carrier BPSK signal, the real and the imaginary outputs of the complex multiplication are filtered by a pair of programmable arm filters,  $G(f)$ . The arm filters are discrete implementations of a first-order low-pass Butterworth filter with a programmable cut-off frequency. The arm filters are used to reduce noise in the carrier-tracking loop, but the cut-off frequency should not be so low that the signal power is reduced by the arm filters significantly. It is found that the cut-off frequency that minimizes the tracking-loop error for the arm filters is approximately equal to the received symbol rate,  $R_s$ , for nonreturn-to-zero (NRZ)-coded data [3,9]. For the MCAS receiver, there can be from 4 to 128 samples per symbol (after CIC decimation). Therefore, the cut-off frequency needs to be programmable between  $f_s/128$  and  $f_s/4$ .

When the received signal is a suppressed-carrier BPSK signal, the output of the real arm filter is passed through a hard limiter with the output equal to +1 or -1 depending on the polarity of the real arm-filter output. It has been shown that, with the operating  $Es/No$  at 0 dB or above, the limiter can reduce the squaring loss,  $S_L$ , of the Costas loops [3]. Squaring loss is caused by the multiplication of the real and imaginary arm signals. This operation is required to remove the data polarity. The penalty of this squaring operation is that noise in the imaginary arm is multiplied by both the signal and noise of the real arm, resulting in poorer noise performance. With the hard limiter, the cross-multiplier before the loop filter,  $F(f)$ , can be replaced by a combination of a switch and an inverter. The input to the loop filter is inverted when the output of the real arm-filter output is negative. Instead of a real multiplier, the switch and inverter reduce the complexity of the digital circuits, thus reducing the power consumption. A Costas loop with a hard limiter is termed a polarity-type Costas loop.

The MCAS1 carrier-tracking loop is a second-order loop. A second-order loop has the advantage of being able to track a constant frequency offset without incurring any tracking error and possesses good stability properties. The transfer function of a second-order, continuous-time loop is



$$H(f) = H(\omega)|_{\omega=2\pi f} = \frac{1 + \frac{2\zeta}{\omega_n}j\omega}{1 + \frac{2\zeta}{\omega_n}j\omega - \left(\frac{\omega}{\omega_n}\right)^2} \Bigg|_{\omega=2\pi f} \quad (15a)$$

where

$$\zeta = \sqrt{\frac{A\tau_2^2}{4\tau_1}} \quad (15b)$$

is the loop-damping factor and

$$\omega_n = \sqrt{\frac{A\tau_2^2}{\tau_1}} \quad (15c)$$

is the natural frequency of the loop;  $A$  is the product of the loop gain,  $\sqrt{P}$ , and the AGC multipliers (see below); and  $\tau_1$  and  $\tau_2$  are determined by the transfer function of the loop filter,  $F(f)$ :

$$F(f) = \frac{1 + \tau_2(j2\pi f)}{\tau_1(j2\pi f)} \quad (16)$$

For the MCAS1 carrier-tracking loop,  $\zeta$  is chosen to be 0.707 to give the loop desirable transient responses. The one-sided noise equivalent bandwidth of the loop is defined as

$$B_L \equiv \int_0^{\infty} |H(f)|^2 df \quad (17)$$

Typically,  $B_L$  is much smaller than  $R_s$  for the loop to track properly. In the discrete-time MCAS implementation, the transfer function of the loop filter,  $F(z)$ , is [10]

$$F(z) = F_1 + \frac{F_2 z}{z - 1} \quad (18)$$

where  $F_1 = 8/3 \times B_L$ ,  $F_2 = 32/9 \times B_L^2 \times T$ , and  $T = 1/f_s$  is the sample period of the loop. The output from the loop filter is used by the NCO to form the current phase estimate.

The carrier-tracking-loop performance usually is expressed in terms of tracking-error variance,  $\sigma_\varphi^2$ . The tracking-loop bandwidth is related to the tracking-error variance by

$$\rho = \frac{1}{\sigma_\varphi^2} = \frac{P \times S_L}{N_0 B_L} \quad (19)$$

where  $\rho$  is the tracking-loop signal-to-noise ratio, i.e., the loop SNR, and  $S_L$  is the squaring loss and is between  $-3$  dB and  $-0.3$  dB, depending on  $E_s/N_0$  and the arm-filter bandwidth [3]. It is found that, in order for the Costas loop to track reliably with low probability of loss of lock and cycle slips, the loop SNR should be above the following threshold:  $\rho \geq 17$  dB.

In other words, the tracking-loop bandwidth,  $B_L$ , should be chosen to be small enough to meet the above inequality and yet large enough to reduce acquisition time. Equivalently,

$$B_L \leq 10^{0.1 \times (10^* \log 10(P/N_0) + S_{L[\text{dB}]} - 17)} = 10^{0.1 \times (E_s/N_{0[\text{dB}]} + 10^* \log 10(R_s) + S_{L[\text{dB}]} - 17)} \quad (20)$$

where the last equality follows from  $P = E_s \times R_s$  for suppressed-carrier BPSK signals. The above loop SNR threshold of 17 dB, however, is not a hard limit. If the loop SNR is only slightly less than 17 dB, the carrier-tracking loop should still track the received phase reliably. However, as  $\rho$  continues to decrease, the loop will start to lose lock and experience more frequent cycle slips.

From Eqs. (15) and (17), the tracking-loop bandwidth,  $B_L$ , and the loop-damping factor are functions of the AGC gain. From Eq. (5),

$$K_{\text{AGCT}} \times \sqrt{\frac{\alpha_P P}{2} + \alpha_N R_s N_0} = 1 \quad (21)$$

where  $K_{\text{AGCT}}$  denotes the product of the AGC multiplier,  $K_{\text{AGC}}$ , and the fixed gains,  $K_F$ , i.e.,  $K_{\text{AGCT}} \equiv K_{\text{AGC}} \times K_F$  (see Section III.A);  $\alpha_P$  is the fraction of the signal power at the input to the Costas loop that reaches the output of the arm filters; and  $\alpha_N$  represents the fraction of the noise power in the signal bandwidth,  $R_s N_0$ , that reaches the output of the arm filters. Both  $\alpha_P$  and  $\alpha_N$  are functions of the arm-filter cut-off frequency, which in turn depends on the throughput rate of the loop in terms of the number of samples per symbol. They are independent of the received signal power.

We now denote the output of the real arm filter as  $[G_{RE}(f)]_{\text{out}}$ . The contribution of the signal at the input of the loop filter is equal to

$$K_{\text{AGCT}} \times \sqrt{\frac{\alpha_P P}{2}} \times \text{sgn}\{[G_{RE}(f)]_{\text{out}}\} \times \sin \varphi \quad (22)$$

where  $\text{sgn}\{x\}$  is the signum function and is equal to 1 when  $x > 0$  and  $-1$  when  $x < 0$ . The MCAS carrier-tracking loop is designed so that the magnitude of the input to the loop filter is equal to 1 at the threshold value of  $E_s/N_0$  (typically 0 dB). Thus, we multiply the input to the loop filter by the factor  $1/(K_{\text{AGCT}} \times \sqrt{\alpha_P P/2})$ , evaluated at threshold. This is termed the bandwidth correction factor. From Eq. (21),

$$\frac{1}{K_{\text{AGCT}} \times \sqrt{\alpha_P P/2}} \Big|_{E_s/N_0=\text{threshold}} = \sqrt{1 + 2 \frac{\alpha_N}{\alpha_P \times E_s/N_0}} \Big|_{E_s/N_0=\text{threshold}} \quad (23)$$

The bandwidth-correction factor is a function of the threshold  $E_s/N_0$  as well as of the number of samples per symbol and is independent of the data rate. As discussed in Section III.A (see Fig. 7), as  $E_s/N_0$  increases above threshold, the loop bandwidth will expand correspondingly—however, at such a slow rate that the tracking-loop SNR also will continue to increase as desired.

When the received signal is a residual-carrier BPSK signal or a pure sinusoidal tone, the residual carrier-tracking mode of the MCAS carrier-tracking loop should be used. The received signal with noise in these cases can be expressed as

$$r(t) = s(t) + n(t) = \sqrt{2P} \cos(\omega t + a_k \delta + \theta) + \sqrt{2}N_c \cos(\omega t + \theta) - \sqrt{2}N_s \sin(\omega t + \theta) \quad (24)$$

where  $0 \leq \delta \leq \pi/2$  is termed the modulation index and  $N_c$  and  $N_s$  denote independent Gaussian noise terms, as before. The desired signal can be expanded as

$$s(t) = \sqrt{2P} \cos(a_k \delta) \cos(\omega t + \theta) - \sqrt{2P} \sin(a_k \delta) \sin(\omega t + \theta) \quad (25)$$

Since  $a_k = \pm 1$ , we have that

$$\left. \begin{aligned} \cos(a_k \delta) &= \cos(\delta) \\ \sin(a_k \delta) &= a_k \sin(\delta) \end{aligned} \right\} \quad (26)$$

We now define  $P_c \equiv P \cos^2(\delta)$  as the discrete carrier power and  $P_d \equiv P \sin^2(\delta)$  as the data power;  $s(t)$  can be expressed as the sum of a sinusoidal tone and a BPSK-modulated signal with a 90-deg phase shift:

$$s(t) = \sqrt{2P_c} \cos(\omega t + \theta) - \sqrt{2P_d} a_k \sin(\omega t + \theta) \quad (27)$$

The total received signal power is

$$P_c + P_d = P \cos^2(\delta) + P \sin^2(\delta) = P \quad (28)$$

When  $\delta = 0$  ( $P_d = 0$  and  $P_c = P$ ),  $s(t)$  represents the pure unmodulated tone. The low-pass filtered downconverted signal at the input to the carrier-tracking loop is

$$\begin{aligned} \text{LPF} (r(t) \times e^{-j\omega t}) &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} \cos(a_k \delta + \theta) + N_c \cos \theta - N_s \sin \theta \right] \\ &+ j \frac{\sqrt{2}}{2} \left[ \sqrt{P} \sin(a_k \delta + \theta) + N_c \sin \theta + N_s \cos \theta \right] \end{aligned} \quad (29)$$

Since the carrier-tracking loop bandwidth is much narrower than the data rate, the loop will track only the phase term,  $\theta$ . The input is multiplied by  $e^{-j\hat{\theta}}$ , which is formed using the estimate,  $\hat{\theta}$ , of the phase as in the case for a suppressed-carrier BPSK signal. The product of the complex multiplication is

$$\begin{aligned} \text{LPF} (r(t) \times e^{-j\omega t}) \times e^{-j\hat{\theta}} &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} \cos(a_k \delta + \varphi) + N_c \cos \varphi - N_s \sin \varphi \right] \\ &+ j \frac{\sqrt{2}}{2} \left[ \sqrt{P} \sin(a_k \delta + \varphi) + N_c \sin \varphi + N_s \cos \varphi \right] \end{aligned} \quad (30)$$

A conventional phase-locked loop (PLL) often is used to track a residual BPSK signal, as in the case of the Mars Global Surveyor (MGS) relay with Deep Space 2 (DS2). In a conventional PLL, the output of the phase detector, i.e., the product of the received signal and the voltage-controlled oscillator (VCO) output (ignoring second harmonic terms), is given by

$$r(t) \times \left\{ -\sqrt{2} \sin(\omega t + \hat{\theta}) \right\} = \sqrt{P} \sin(a_k \delta + \varphi) + N_c \sin \varphi + N_s \cos \varphi \quad (31)$$

The imaginary part of the MCAS complex multiplication output is the same as the phase detector output of the PLL except for the  $\sqrt{2}/2$  factor, which arises as previously discussed [Eq. (14)]. The imaginary part of the complex product is filtered by the arm filter and then fed to the loop filter to form the phase estimate,  $\hat{\theta}$ , as in a conventional PLL. With proper scaling, the MCAS carrier-tracking loop is identical to a conventional passband PLL with the NCO centered at the carrier frequency.

When the loop is locked, i.e.,  $\theta - \hat{\theta} \approx 0$ , or equivalently,  $\sin \varphi \approx 0$  and  $\cos \varphi \approx 1$ , the real part of the product of the complex multiplication can be expressed as

$$\begin{aligned} RE \left\{ \text{LPF} \left( r(t) \times e^{-j\omega t} \right) \times e^{-j\hat{\theta}} \right\} &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} \cos(a_k \delta + \varphi) + N_c \cos \varphi - N_s \sin \varphi \right] \\ &\approx \frac{\sqrt{2}}{2} \left[ \sqrt{P_c} + N_c \right] \end{aligned} \quad (32)$$

The real part has an average DC component that is directly proportional to the square-root of the carrier power,  $P_c$ , since both  $N_c$  and  $N_s$  are zero mean. The imaginary part is

$$\begin{aligned} IM \left\{ \text{LPF} \left( r(t) \times e^{-j\omega t} \right) \times e^{-j\hat{\theta}} \right\} &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} \sin(a_k \delta + \varphi) + N_c \sin \varphi + N_s \cos \varphi \right] \\ &\approx \frac{\sqrt{2}}{2} \left[ \sqrt{P_d} a_k + N_s \right] \end{aligned} \quad (33)$$

Note that the data are recoverable from the imaginary part, with power proportional to  $P_d$ . Thus, for a suppressed-carrier BPSK received signal, data are recovered from the real part while, for a residual-carrier BPSK signal, the data are recovered from the imaginary part.

In a conventional PLL, there is no arm filter following the phase detector. In the MCAS residual carrier-tracking mode, the arm filters are included in the loop to reduce the noise power to the loop filter. Since both of the arm filters are also required for the AGC loop, inclusion of the arm filters in the loop does not increase the implementation complexity or power consumption. The output of the imaginary arm filter is filtered by a one-pole loop filter,  $F(z)$ , as in the Costas-loop mode, making the PLL a second-order loop. Squaring is not required in a PLL for tracking a residual-carrier BPSK or unmodulated signal.

When the loop is operated in the residual carrier-tracking mode, the imaginary part of the complex multiplication product is used to drive the loop. From Eq. (30), the imaginary output can be expressed as

$$\begin{aligned}
IM \left[ \text{LPF}(r(t) \times e^{-j\omega t}) \times e^{-j\hat{\theta}} \right] &= \frac{\sqrt{2}}{2} \left[ \sqrt{P} \sin(a_k \delta + \varphi) + N_c \sin \varphi + N_s \cos \varphi \right] \\
&= \frac{\sqrt{2}}{2} \left[ \sqrt{P_d} a_k \cos(\varphi) + \sqrt{P_c} \sin(\varphi) + N_c \sin \varphi + N_s \cos \varphi \right] \quad (34)
\end{aligned}$$

The loop tracks  $\sqrt{P_c/2} \sin \varphi$  as  $\varphi$  changes slowly with time. The fraction of data power in the tracking-loop bandwidth,  $P_d/2$ , hinders the loop's ability to track the residual carrier and should be considered interference. The interference power is a function of the ratio  $B_L/R_s$  and the data sequence  $a_k$  and can be computed from

$$P_I = \int_{-\infty}^{\infty} S_d(f) |H(f)|^2 df \quad (35a)$$

where  $S_d(f)$  denotes the power spectral density of the data sequence  $a_k$  and is equal to ( $T_s$  denotes the symbol period)

$$S_d(f) = T_s \frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2} \quad (35b)$$

when  $a_k$  is NRZ coded or

$$S_d(f) = T_s \frac{\sin^4(\pi f T_s/2)}{(\pi f T_s/2)^2} \quad (35c)$$

when  $a_k$  is Manchester coded. As the ratio  $B_L/R_s$  increases, the fraction of data power in the loop bandwidth increases and, thus, the interference increases. If  $a_k$  is NRZ-coded, the data power centers around DC and can severely interfere with the carrier-tracking loop. While the MCAS carrier-tracking loop still may be able to track the received carrier phase, depending on the ratio of  $B_L/R_s$ , residual-carrier BPSK with NRZ-coded data should be avoided. If  $a_k$  is Manchester-coded, there is a null at DC and most of the energy is centered around  $f = R_s$ . If  $B_L/R_s$  is much smaller than 1, the data do not affect the carrier-tracking loop.

The tracking loop performance of the residual carrier-tracking-loop mode can be measured in terms of

$$\frac{1}{\sigma_\varphi^2} = \rho = \frac{P_c}{N_0 B_L + P_I} \quad (36)$$

Compared with the Costas-loop SNR, there is no squaring loss, but there is an additional interference term,  $P_I$ , in the denominator. If the loop is used to track a pure tone, there is no data interference and, thus,  $P_I = 0$ .

As an example, if a residual-carrier BPSK NRZ signal at 8 ksym/s and with a modulation index of  $\pi/3$  is tracked by a residual carrier-tracking loop with a loop bandwidth of 100 Hz, the ratio of the interference to the received signal power is 0.074. Without any noise, the loop SNR is equal to 11.3 dB. However, if the

signal is Manchester coded instead, the ratio of the interference to the received signal power is 0.000275, and, without any noise,  $\rho = 35.6$  dB, which is much greater than the threshold needed for a residual carrier-tracking loop to track reliably. Manchester-coded residual-carrier BPSK has been implemented for the DS2-to-MGS relay and currently is being reviewed for the CCSDS in situ relay standard.

For a residual carrier-tracking loop, the loop SNR,  $\rho$ , should be chosen based on bit-error rate (BER) and the loop's cycle slip characteristic. The mean time between cycle slips,  $\bar{\tau}_{\text{slip}}$ , for a second-order residual carrier-tracking loop with  $\varsigma = 0.707$  and  $\rho \geq 2$  can be approximated by [14; p. 252]

$$\bar{\tau}_{\text{slip}} \approx \frac{10^{0.6 \times \rho}}{B_L} \quad (37)$$

As  $\rho$  increases, the mean time between cycle slips increases. The probability of having a cycle slip in  $\tau$  seconds can be approximated by

$$P_{\text{slip}}(\tau) \approx 1 - e^{-(\tau/\bar{\tau}_{\text{slip}})} \quad (38)$$

Since cycle slip causes the tracking loop to be out of lock and the data to be erroneous momentarily until the loop recovers, it is desirable to keep the probability of cycle slips very low. Typically, the probability of cycle slips is required to be less than  $10^{-6}$  for every 1-second interval so that the error caused by cycle slips will be much less than the  $10^{-5}$  BER requirement. Under this condition, the loop SNR must be greater than 12 dB at  $B_L = 10$  kHz and greater than 11 dB at  $B_L = 10$  Hz.

The tracking error,  $\sigma_\varphi^2 = 1/\rho$ , also affects the BER. For a BPSK link, the BER can be written as [3; p. 205]

$$\text{BER} = \frac{1}{2} \int_{-\pi}^{\pi} \text{erfc} \left( \sqrt{\frac{E_s}{N_0}} \cos \varphi \right) p(\varphi) d\varphi \quad (39)$$

where  $\text{erfc}(x)$  is the complementary error function and  $p(\varphi)$  is the probability density function of the phase error, which is a function of  $\rho$ . For  $\rho \leq 10$  dB, the BER curve as a function of  $E_s/N_0$  reaches an asymptotic floor above  $10^{-5}$ , regardless of  $E_s/N_0$ . For  $\rho = 10$  dB, an additional 1.2 dB is needed to achieve a  $10^{-3}$  BER. For  $\rho = 12$  dB, an additional 0.9 dB of  $E_s/N_0$  is required to achieve a  $10^{-3}$  BER, as compared with the ideal BPSK curve. For  $\rho = 14$  dB, the additional  $E_s/N_0$  required for a  $10^{-5}$  BER is less than a 0.4-dB BER and, for  $\rho = 16$  dB, it is less than 0.2 dB.

Based on the above considerations, for the MCAS1 residual carrier-tracking loop, we impose the following minimum loop SNR guideline:  $\rho \geq 12$  dB for good cycle slip and BER characteristics. However, for an uncoded link that requires a BER of  $10^{-5}$ , we suggest the loop SNR be programmable up to approximately 14 dB. It should be noted that the above 12-dB threshold is not a hard limit. An MCAS carrier-tracking loop operating in the residual carrier-tracking-loop mode can track the received phase when the loop SNR is lower than 12 dB; however, the probability of cycle slips increases very quickly as the loop SNR decreases below the 12-dB threshold.

Since the received signal strength for an MCAS1 carrier-tracking loop operating in the residual carrier-tracking mode is affected by the AGC loop in the same way as in the Costas-loop mode, the signal strength needs to be adjusted properly to normalize the tracking-loop gain. In the PLL mode, the MCAS1 carrier-tracking loop tracks only the residual-carrier, while the AGC loop adjusts the gain based on the power of both the carrier and the data in addition to the noise, i.e.,

$$K_{\text{AGCT}} \times \sqrt{\alpha_{P_d} \frac{P_d}{2} + \alpha_{P_c} \frac{P_c}{2} + \alpha_N R_s N_0} = 1 \quad (40)$$

where  $\alpha_{P_d}$  is the fraction of the data power at the input to the carrier-tracking loop that reaches the output of the arm filters;  $\alpha_{P_c}$  is the fraction of the carrier power at the input to the carrier-tracking loop that reaches the output of the arm filters; and  $\alpha_N$  again represents the fraction of the noise power in the signal bandwidth,  $R_s N_0$ , that reaches the output of the arm filters. Note that  $\alpha_{P_c} = 1$  when the carrier is contained within the tracking-loop bandwidth. We now define the ratio of carrier power to data power as

$$\chi \equiv \frac{P_c}{P_d} = \frac{\cos^2 \delta}{\sin^2 \delta} \quad (41)$$

The bandwidth-correction factor for the residual carrier-tracking mode is then

$$\frac{1}{K_{\text{AGCT}} \times \sqrt{\alpha_{P_c} \frac{P_c}{2}}} \Bigg|_{E_s/N_0=\text{threshold}} = \sqrt{1 + \frac{\alpha_{P_d}}{\alpha_{P_c} \times \chi} + 2 \frac{\alpha_N}{\alpha_{P_c} \times \frac{E_s}{N_0} \times \chi}} \Bigg|_{E_s/N_0=\text{threshold}} \quad (42)$$

where  $P_d = E_s \times R_s$ . As in the Costas-loop mode, the bandwidth-correction factor is a function of the threshold  $E_s/N_0$  as well as the number of samples per symbol that determines the bandwidth of the arm filters (and thus  $\alpha_{P_d}$ ,  $\alpha_{P_c}$ , and  $\alpha_N$ ), but is independent of the data rate.

We now briefly discuss the acquisition algorithm for the MCAS carrier-tracking loop in the Costas-loop mode. The algorithm for the residual carrier-tracking mode has not yet been developed, although it is expected to be very similar to that for the Costas-loop mode, which is described in the following. This algorithm is used to aid the digital Costas loop in acquiring phase/frequency lock. This is accomplished by sweeping through a user-specified range (sweep range) of NCO frequencies at a user-specified sweep rate and comparing the difference between the real and imaginary arm-channel power estimates. The frequency sweeping is accomplished in discrete increments that are maintained for a user-specified period of time. The real and imaginary arm-power estimates are obtained by averaging  $M$  samples of data over each frequency increment interval. A lock detector output signal is generated from the difference between the real and imaginary arm-filter output power estimates. This signal then is used to determine if the Costas loop is locked by comparing it with a user-programmable threshold. The duration of each frequency increment and the lock-detector threshold are functions of  $E_s/N_0$  and the Costas-loop filter bandwidth.

The algorithm is always in one of two states: (1) in frequency/phase lock (verification state) or (2) out of frequency/phase lock (acquisition state). The user may specify that  $N1 = 1, 2, 3$ , or 4 consecutive threshold “hits” by the lock detector output indicates that the Costas loop is in phase and frequency lock before the algorithm transitions from acquisition to verification state. Likewise, the user may specify that  $N2 = 1, 2, 3$ , or 4 consecutive threshold “misses” indicates that the Costas loop is not in phase and frequency lock before the algorithm transitions from a verification state back to an acquisition state. When  $N1 > 1$ , the probability of false lock,  $P_{FL-\text{Total}}$ , is given by [3]

$$P_{FL-\text{Total}} = (P_{FL})^{N1} \quad (43)$$

where  $P_{FL}$  denotes the false lock probability when  $N1 = 1$ . Similarly, when  $N2 > 1$ , the probability of false alarm,<sup>20</sup>  $P_{FA-Total}$ , is given by [3]

$$P_{FA-Total} = (P_{FA})^{N2} \quad (44)$$

where  $P_{FA}$  denotes the false alarm when  $N2 = 1$ .

The algorithm allows the user to clear the Costas-loop filter registers after every frequency sweep. This is required when sweeping over large frequency ranges at low SNRs. When the Costas loop is not in lock, the values in the filter accumulators average to zero over long periods of time. However, it is possible for a bias to become present in the digital accumulators in the Costas-loop filter even when it is not in lock. This bias may be temporary, an anomaly in the noise sequence, or it may due to a DC bias in the receiver system. In any case, when such a bias exists, it can delay or even preclude Costas-loop acquisition. This problem can be eliminated by periodically clearing the filter accumulators when the algorithm is in the acquisition state. The maximum rate at which the Costas filter registers can be cleared is equal to the rate that the NCO frequency is incremented, i.e., every  $M$  samples.

The Costas loop can false lock onto harmonics of the data that are half multiples of the data rate [11]. To counteract these false locks, the imaginary Costas arm filter can be switched out of the Costas loop (see Fig. 13). This greatly reduces the probability of false lock [12]. For  $E_s/N_0 \leq 20$  dB, this virtually eliminates false locks. However, for  $E_s/N_0 > 20$  dB, false locks are still possible but unlikely [11]. For data rates that are less than or equal to twice the sweep range, the algorithm switches the Q-channel arm filter out of the Costas loop during the acquisition state and back into the Costas loop once the acquisition algorithm transitions to the verification state.

The algorithm provides the real and imaginary arm-filter output power estimates as well as the number of attempts to acquire and the no-lock/lock flag to the user through status registers.

We conclude this section with a description of the digital implementation of the MCAS1 carrier-tracking loop building blocks: (1) the  $\{\text{NCO output}\} \times \{\text{complex baseband input}\}$  complex multiplier (Section IV.A.1); (2) the loop arm filters (Section IV.A.2); (3) the acquisition versus tracking switch, SW3 in Fig. 13 (Section IV.A.3); (4) the hard limiter and switch SW1 in Fig. 13 (Section IV.A.4); (5) the loop filter (Section IV.A.5); and (6) the NCO (Section IV.A.6). Throughout the digital portion of the MCAS1 transceiver design, all adders and multipliers are pipelined to maximize the throughput rate. Thus, there is a delay following all adders and multipliers. The additional delay in the loop can affect the transfer function of the MCAS1 carrier-tracking loop, but if the total delay is much less than  $1/B_L$ , which is the case for MCAS1 applications, the loop transfer function remains unchanged. Rounding rather than truncation arithmetic is used to reduce quantization noise effects. All signals are represented in two's complement notation, and all indicated data bit widths include the sign bit.

**1. Complex Multiplier.** Both the real and the imaginary outputs from the CIC filters are 6-bits wide, and the complex NCO outputs are 8-bits wide, as indicated in Fig. 14. The output from the complex multiplier is rounded to 6 bits.

**2. Arm Filters:  $G(\mathbf{z})$ .** The real and imaginary outputs from the  $\text{CIC} \times \text{NCO}$  complex multiplier (Section IV.A.1) are filtered by a pair of programmable arm filters. The arm filters are discrete implementations of a first-order, low-pass Butterworth filter with programmable normalized cut-off frequencies between  $f_s/128$  and  $f_s/4$ . The transfer function is given by

---

<sup>20</sup> Equivalently, the probability of false indication of loss of lock.



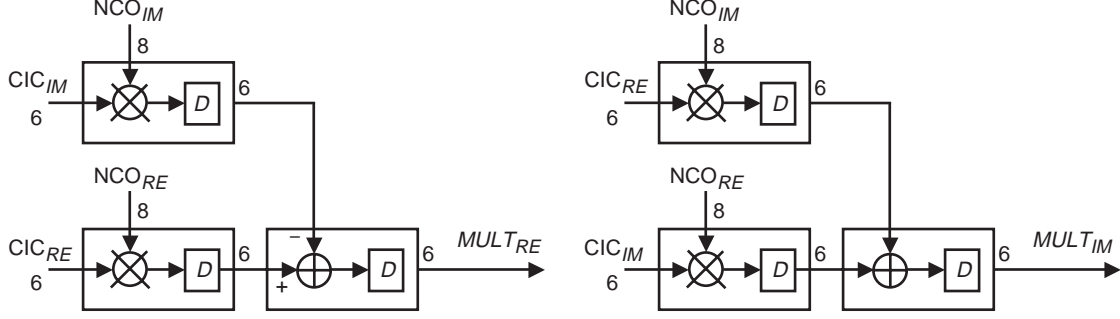


Fig. 14. Block diagram of the complex multiplier.

$$G(z) = G_{\text{gain}} \frac{1 + z^{-1}}{1 - G_1 z^{-1}} \quad (45)$$

where  $G_{\text{gain}}$  and  $G_1$  are programmable parameters that determine the cut-off frequency of the filter.

Figure 15 shows the structure of the arm filters. Because pipelining is required to increase the throughput rate, the feedback portion of the filter has two delays instead of the one delay typically used in a first-order Butterworth filter. This introduces an additional pole in the pipelined Butterworth filter transfer function. Thus, an additional stage is added to remove one of the poles in the feedback section to produce the proper filter response. The transfer function of the resulting augmented, pipelined Butterworth arm filter is given by

$$G(z) = G_{\text{gain}} \frac{1 + z^{-1}}{1 - G_1^2 z^{-2}} (1 + G_1 z^{-1}) \quad (46)$$

Instead of  $G_{\text{gain}}$  and  $G_1$ , three parameters— $G_{\text{gain}}$ ,  $G_1$ , and  $G_1^2$ —need to be programmed to achieve the desired cut-off frequency. The ranges of the three parameters for cut-off frequencies between  $f_s/128$  and  $f_s/4$  are shown in Fig. 16.

Because the arm-filter cut-off frequency can be as small as  $f_s/128$ , the output power from the arm filters can be much smaller than the input power. Thus, an additional stage of scaling by a programmable constant, AGC2, is necessary so that quantization loss does not degrade the tracking-loop performance. This additional stage of scaling is shown in Fig. 17. The output of the real arm filter is denoted by  $G_{RE}$ ; the output of the imaginary arm filter is  $G_{IM}$ ; and the CIC decimation filter function is denoted by the decimation filter block (see Fig. 11).

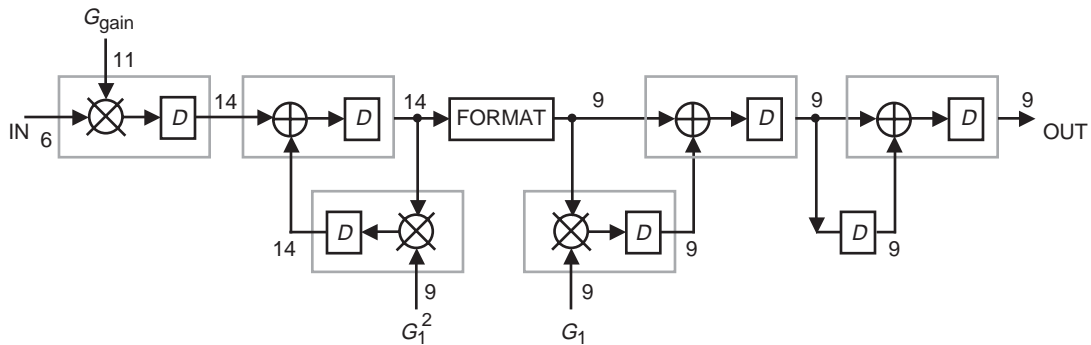


Fig. 15. Block diagram of the arm filter.

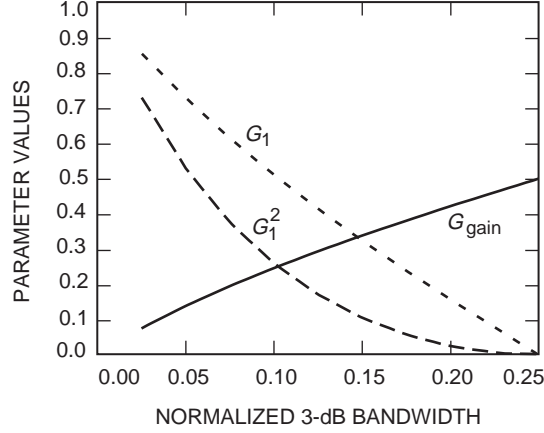


Fig. 16. The range of programmable values of arm-filter parameters.

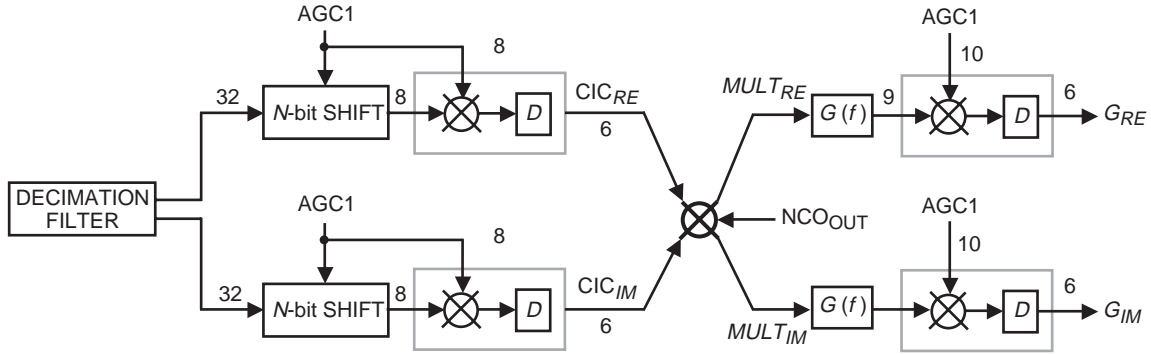


Fig. 17. The scaling of arm-filter outputs.

**3. SW3.** The block diagram of the circuitry surrounding the switch, SW3, is shown in Fig. 18. The position of SW3 indicates whether the tracking loop is acquiring or tracking the received phase in the Costas-loop mode. As discussed previously in this section, the imaginary arm filter of the Costas loop is bypassed to avoid false locks during acquisition. SW3 then chooses between the output signal from the imaginary arm filter or the imaginary output signal from the complex multiplication, thus bypassing the arm filter. The latter signal needs to be scaled by a scaling factor, AGC2, so that the loop will have the same loop gain during tracking and acquisition. It also needs to be delayed to compensate for pipelining and arm-filtering delays so that the real and imaginary output arm signals are properly aligned at the cross-multiplication. The group delay of the arm filter changes as the cut-off frequency changes. The delay of the imaginary output is designed to be programmable depending on the cut-off frequency of the real arm filter. The output from the imaginary arm filter is formatted so that the two inputs to SW3 have the same bit width as indicated in Fig. 18.

**4. Hard Limiter and SW1.** Figure 19 shows the block diagram of the hard limiter that follows the real arm filter and SW1, which determines whether the product of the cross-multiplication or just the imaginary arm-filter output should be filtered by the loop filter. The position of SW1 is chosen depending on whether the loop is operated in the Costas-loop mode or the residual carrier-tracking-loop mode (denoted by PLL in Fig. 19).

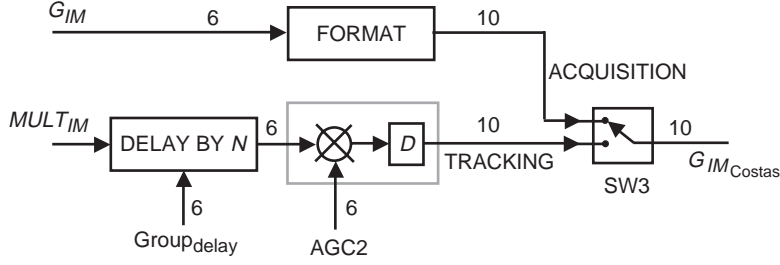


Fig. 18. SW3 and the surrounding circuit.

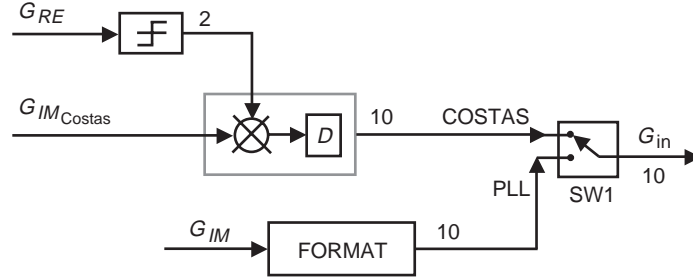


Fig. 19. The hard limiter and SW1.

**5. Loop Filter:  $F(z)$ .** As previously described in Sections III.5.1 and III.5.2, the loop-filter portion of the MCAS carrier-tracking loop is a one-pole filter with the following transfer function:

$$F(z) = F_1 + \frac{F_2 z}{z - 1} \quad (47a)$$

where

$$F_1 = \frac{8}{3} \times B_L \quad (47b)$$

and

$$F_2 = \frac{32}{9} \times B_L^2 \times T \quad (47c)$$

Again,  $T = 1/f_s$  is the sample period of the loop and  $f_s$  is the sample rate.  $F_1$  and  $F_2$  are programmable parameters that determine the tracking loop bandwidth. From MCAS requirements,<sup>21</sup>  $10 \text{ Hz} \leq B_L \leq 10 \text{ kHz}$  and  $1/16.384 \times 10^6 \leq T \leq 1/4/1000$ . Under these two conditions,  $26.7 \leq F_1 \leq 2.67 \times 10^4$  and  $0.0022 \leq F_2 \leq 8.89 \times 10^4$ . Thus, a minimum of  $\lceil \log_2(2.67 \times 10^4/26.7) \rceil = 10$  bits are required to represent  $F_1$  and  $\lceil \log_2(8.89 \times 10^4/0.0022) \rceil = 26$  bits are required for  $F_2$ , where  $\lceil x \rceil$  denotes the ceiling function and is equal to the smallest integer that is greater than  $x$ .

The bit width required for  $F_2$  can be reduced from 26 bits by noting that  $B_L$  usually is chosen to be the largest bandwidth such that the tracking requirements,  $\rho \geq 17 \text{ dB}$  (Costas-loop mode) and  $\rho \geq 12 \text{ dB}$  (PLL mode), or the navigation requirement,<sup>22</sup>  $\rho \geq 24.5 \text{ dB}$ , can be met. Using

<sup>21</sup> D. Hansen, op. cit.

<sup>22</sup> "MCAS1 Conceptual Design Review," Viewgraph Presentation (internal document), Jet Propulsion Laboratory, Pasadena, California, July 30, 1998.

$$\rho = \frac{P \times S_L}{B_L \times N_0} = \frac{E_s}{N_0} \times \frac{R_S}{B_L} \times S_L \quad (48)$$

where  $T = 1/f_s = 1/(k \times R_S)$ ,  $E_s/N_0 \geq 0$  dB, and  $k = 4, 8, \text{ or } 16$ ,<sup>23</sup>  $F_2$  can be computed for different values of  $R_s$  and  $E_s/N_0$ . As a result,  $F_2$  can be bounded by  $0.004 \leq F_2 \leq 3000$ . Thus, a minimum of only  $\lceil \log_2(3000/.004) \rceil = 20$  bits is actually required to represent  $F_2$ . With the bit width of the loop filter input equal to 10 bits, a minimum 10-bit by 20-bit  $F_2$  multiplier is required.

To further simplify the carrier-tracking-loop hardware and reduce the number of multipliers, the  $F_1$  and  $F_2$  multipliers in the loop filter are programmed using the products of the bandwidth correction factor; a constant,  $1/2\pi/1000$ ; and  $F_1$  and  $F_2$ . As discussed previously in this section, the bandwidth-correction factor is needed to normalize the tracking-loop gain to achieve the desired loop bandwidth and transient response. The bandwidth-correction coefficient ranges from 1.0 to 6.25. Multiplication by the constant,  $1/2\pi/1000$ , is actually part of the NCO design (Section IV.A.6). Without this normalization, the output from the loop filter is in units of radians/second. In order to simplify the NCO design, the loop filter output is normalized by  $1/2\pi$  to convert it to the unit of cycles/second or Hertz. The NCO phase estimate will then be in the unit of cycles rather than radians. Normalization by  $1/1000$  corresponds to division by the base sampling rate of 1 kHz. The output from the loop filter is thus in the unit of kHz.

By combining the various multiplication factors together, two multiplication operations are eliminated. The products of  $F_1$  and  $F_2$  with the bandwidth correction factor and the factor  $1/2\pi/1000$  are denoted by  $F_{1_{mod}}$  and  $F_{2_{mod}}$ , respectively. The corresponding loop-filter structure is illustrated in Fig. 20. The parameter ranges for  $F_{1_{mod}}$  and  $F_{2_{mod}}$  are provided in Table 2. For  $F_{1_{mod}}$ , a minimum of 14 bits (1 sign bit, 5 integer bits, and 8 fractional bits) is required. For  $F_{2_{mod}}$ , a minimum of 23 bits (1 sign bit, 2 integer bits, and 20 fractional bits) is required. For the representation of  $F_{1_{mod}}$ , an extra bit is included after the decimal point so that the loop bandwidth can be programmed with higher resolution. Therefore, a 10-bit by 15-bit multiplier is required for  $F_{1_{mod}}$ . To simplify the design effort and to reduce the size of the 10-bit by 23-bit  $F_{2_{mod}}$  multiplier, the same 10-bit by 15-bit multiplier design also is used with  $F_{2_{mod}}$  by categorizing the value of  $F_{2_{mod}}$  as either large or small. When  $F_{2_{mod}}$  is above a threshold,  $F_{2_{TH}}$ , the input to the loop is multiplied by  $F_{2_{mod}}$ . When  $F_{2_{mod}}$  is below the threshold,  $F_{2_{mod}} \times 2^{10}$  is used to multiply the loop input instead of  $F_{2_{mod}}$ . After the multiplication, the product is right shifted by 10 bits to form the proper result (see Fig. 20). In this way, the complexity of the multiplier is significantly reduced.

**6. NCO.** The MCAS carrier-tracking-loop NCO, as shown in Fig. 21, first multiplies the input by the sampling period,  $T = 1/f_s$ . Since  $f_s$  can only be a power of two multiplied by 1 kHz and the loop filter has already normalized the NCO input to the unit of kHz, this multiplication can be performed by

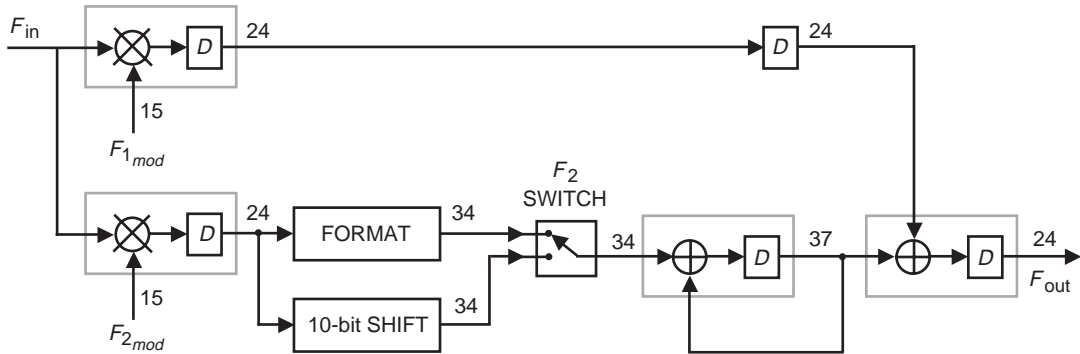


Fig. 20. Block diagram of loop filter  $F(z)$ .

<sup>23</sup> Except for when  $R_s \leq 4$  ksym/s, in which case,  $f_s = 128$  kHz.

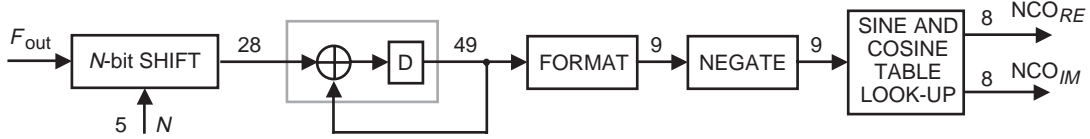
a simple bit shift by  $\log_2(f_s/1000)$  bits. The output from the bit shifter is summed with the previous phase estimate to form the current phase estimate,  $\hat{\theta}$ .

Because the phase estimate is in the unit of cycles, the 8 most significant bits after the decimal point and the sign bit of the phase estimate are used to form the complex output of the NCO,  $e^{-j\hat{\theta}}$ , through sine and cosine table look up. The table look up using the 8 most significant fractional bits provides a reasonable trade-off between low spur noise (no more than  $-48$  dBc) and memory required to store the table.<sup>24</sup>

For purposes of navigation, the MCAS carrier-tracking loop is required to provide cycle and phase information. The maximum frequency offset that the MCAS loop is designed to track is 50 kHz, corresponding to the case of an RF input at S-band. With the nominal integration window of 10 Hz for navigation, the cycle counter, or equivalently, the integer part of the phase register, needs to be greater than  $\lceil \log_2(5 \times 10^4 \times 10) \rceil = 19$  bits. Two additional bits are added for margin.

**Table 2. Parameter ranges for  $F_{1mod}$  and  $F_{2mod}$ .**

Parameter	$F_{1mod}$		$F_{2mod}$	
	Minimum	Maximum	Minimum	Maximum
$F_1$ and $F_2$	$8/3 \times 10$	$8/3 \times 10,000$	0.004	3000
Bandwidth correction	1.0	6.25	1.0	6.25
$1/2\pi/1000$	$1/2\pi/1000$	$1/2\pi/1000$	$1/2\pi/1000$	$1/2\pi/1000$
Product	0.00424	26.53	$6.37 \times 10^{-7}$	2.98



**Fig. 21. The NCO block diagram.**

## B. Navigation: Doppler Phase Measurement

Missions like the Mars Relay may be required to provide Doppler estimates derived from the received signal. This section describes how MCAS Doppler frequency estimates are obtained. The method described herein is applicable to either the Costas-loop mode or the PLL mode of operation. The technique basically derives the Doppler frequency estimate from the difference between two instantaneous phase outputs from the phase register of the NCO. The resulting frequency estimate is equivalent to counting the elapsed phase cycles over a fixed time interval,  $T$ . Simulations show that this technique can provide the accuracy needed to locate a lander or rover with sufficient transmit power.

The MCAS Doppler frequency estimate,  $f_{est}$ , is given by

$$f_{est} \equiv \frac{\theta(t+T) - \theta(t)}{2\pi T} \quad (49)$$

<sup>24</sup> "MCAS1 Conceptual Design Review RFA's," op. cit.

where  $T$  is the time between two instantaneous phase measurements,  $\theta(t)$  and  $\theta(t + T)$ . Assuming that  $T$  is sufficiently large such that  $\theta(t + T)$  and  $\theta(t)$  are independent (i.e.,  $T \gg 1/B_L$ , where  $B_L$  is the noise-equivalent bandwidth of the tracking loop), then the variance of  $f_{est}$ ,  $\sigma_f^2$ , is approximated by

$$\sigma_f^2 = \frac{2\sigma_\theta^2}{(2\pi T)^2} \quad (50)$$

where  $\sigma_\theta^2$  denotes the variance of the phase measurements. Simulation results have verified the accuracy of Eq. (50) for both PLL- and Costas-loop modes. As seen from Eq. (50), the standard deviation of the Doppler frequency-estimation error is inversely proportional to  $T$ . Thus,  $T$  should be as large as possible while still yielding a meaningful frequency estimate. Typically,  $T$  is on the order of 10 to 60 s for a 10-min pass.<sup>25</sup>

Based on data obtained from W. Folkner,<sup>26</sup> in order to achieve 1-km accuracy for one-way Doppler positioning, the frequency measurements should have an accuracy of 1 mm/s for 1-min averaging using an oscillator with an Allan deviation of  $2 \times 10^{-10}$  or less. At 400 MHz, 1 mm/s corresponds to 0.0013 Hz. To be conservative, we assume that  $T = 10$  s, in which case we have, from Eq. (50),

$$\sigma_f = \sqrt{\frac{2\sigma_\theta^2}{(2\pi T)^2}} = \sqrt{\frac{2/\rho}{(2\pi T)^2}} \leq 0.0013 \text{ Hz} \quad (51)$$

where  $\rho$  denotes the loop SNR. Substituting  $T = 10$  s into Eq. (51), we find that  $\sigma_\theta^2 \leq 0.00351$  and  $\rho \geq 24.6$  dB, satisfying the 1-km navigation requirement. This requirement for loop SNR is more stringent than that typically required for communications with a Costas loop, i.e.,  $\rho \geq 17$  dB.

The MCAS carrier-tracking loop is designed so that the phase variance due to digital quantization errors is a small fraction of 0.00351. The above phase variance constraint is a conservative bound on phase jitter and can be relaxed somewhat while still meeting the sub-1-km navigation requirement.<sup>27</sup> Detailed simulations and analysis are required to determine a tighter bound.

### C. Symbol-Timing Recovery

The symbol-timing-recovery algorithm is based on the absolute value type of the early-late gate symbol synchronizer discussed in [7]. A combination of the early-late gate circuit and a random walk filter are employed as a symbol-timing estimator, as shown in Fig. 22. This subsystem generates a timing signal that can be used to jitter either the symbol clock or the sample clock of the analog-to-digital (A/D) converter. The nominal mode of operation calls for jittering the A/D clock because this minimizes the sampling-offset-induced degradation in bit-error-rate performance. In certain situations when it is desirable to optimize navigation performance, the symbol clock may be jittered, thus keeping a near constant time base for the phase samples entering the tracking loop.

The noisy baseband received symbols for which the timing is to be recovered first are routed through a CIC decimation filter, as indicated in Figs. 1 and 22. The CIC is identical to that defined in Section III.C and is used to decimate to 16 samples/sym for the lower symbol rates (i.e., 1, 2, 4, ..., 1024 ksym/s) and thus to narrow the bandwidth and improve the SNR prior to timing recovery. At 2048 ksym/s, 8 samples/sym are used for timing recovery and, at 4096 ksym/s, 4 samples/sym are used.

<sup>25</sup> W. Folkner, personal communication, Tracking Systems and Applications Section, Jet Propulsion Laboratory, Pasadena, California, July 30, 1998.

<sup>26</sup> Ibid.

<sup>27</sup> Ibid.

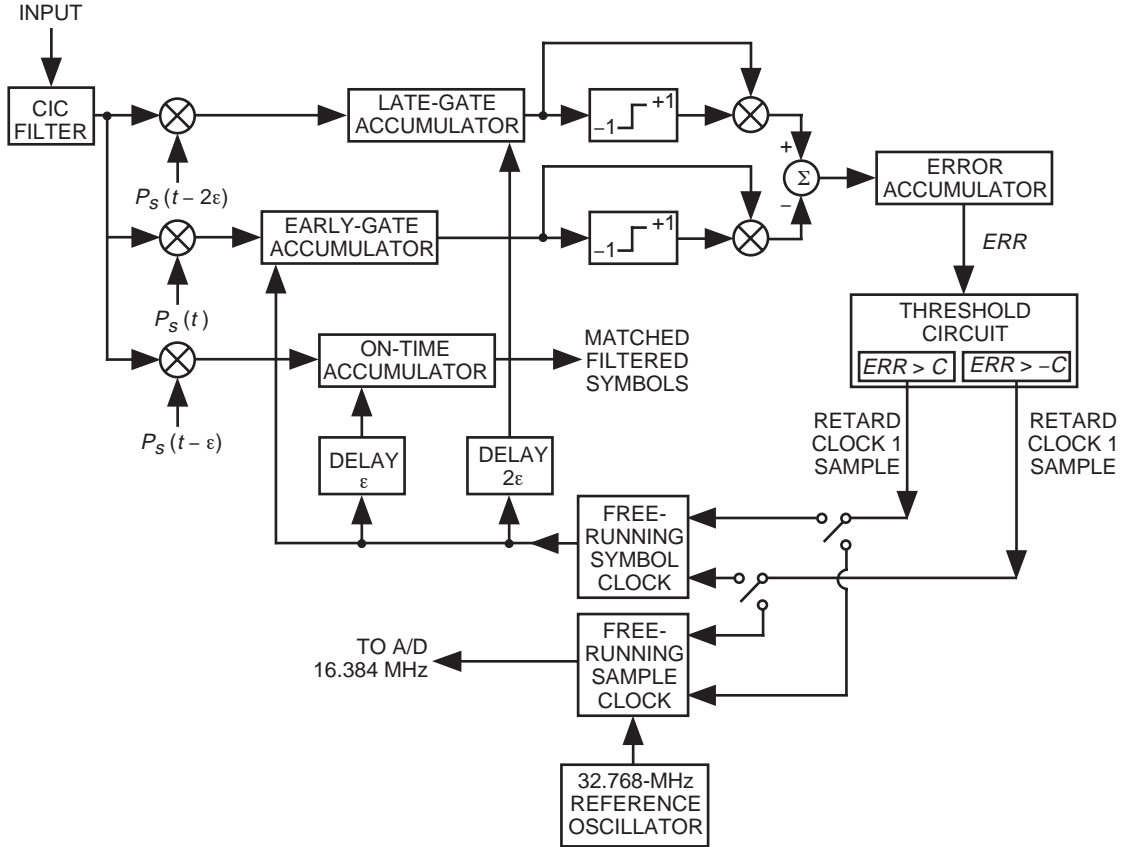


Fig. 22. Block diagram of the symbol-timing estimator.

The early-late gate symbol-synchronization algorithm is motivated by the maximum a posteriori (MAP) estimation of an unknown parameter in Gaussian noise. As with the MAP estimator, the received BPSK signal that has passed through the CIC first is routed to cross-correlators. The correlators operate on three different subintervals of the received signal (i.e., early, late, and on time), multiplying the received signal by an appropriately delayed stored replica of the transmitted pulse and then integrating (see Fig. 22).

The nominal operating mode of the symbol-timing-recovery circuit corresponds to NRZ pulse shaping, in which case the pulse shape depicted in Fig. 22 is  $P_s(t) = 1$ . In the case of the Manchester-encoded pulse shape, a stored replica of the Manchester  $P_s(t)$  (a binary 1 is represented as a 1 for the first half of the symbol period and a 0 for the second half of the symbol period) is multiplied by the received signal for each of the three arms of the timing recovery circuit.

The multiplications by the pulse shape are followed by an early integrator, an on-time integrator, and a late integrator. The three integrators are controlled by synchronous clocks that are slightly offset in time. The integration period is  $T_s$  for all three integrate-and-dump channels, with the early integration starting a quarter of a symbol early and the late integration starting a quarter of a symbol late (i.e.,  $\epsilon = T_s/4$  in Fig. 22, which is shown to be optimal in [7]). The on-time integration is equivalent to a matched filter for the transmitted symbols when the symbol timing is perfectly aligned.

The outputs from the early and late integrators are utilized to generate a timing-error signal by differencing their respective absolute values. If the value of the early channel is greater than that of the late channel, an error signal is generated that slows down the clock and, if the late channel value is

greater than that of the early channel, a signal is generated to speed up the clock. The absolute value process eliminates the dependence of the error signal on bit polarity.

The timing-error signal is used to advance/retard the phase of either the free-running symbol clock or the free-running sample clock. The timing jitter induced by noise is suppressed by a random walk filter [13]. The random-walk filter consists of an error accumulator and threshold circuit. The accumulator is essentially an up/down counter accumulating the advance/retard errors ( $ERR$  in Fig. 22). If  $ERR$  exceeds the threshold,  $C$ , then a correction pulse will be issued to correct for the jitter accordingly. The estimate of the timing,  $cl(t)$ , is a stream of impulses nominally at the input symbol rate,  $1/T$ , as depicted in Fig. 23. We denote the error between the time of occurrence of the  $k$ th impulse,  $\tau_k$ , and the true  $k$ th symbol transition time,  $t_k = t_0 + kT$ , as  $e_k = \tau_k - t_k$ , which is termed the timing-jitter error. The symbol-timing-recovery subsystem is required to maintain an instantaneous timing jitter,  $e_k$ , satisfying  $e_k/T \leq 0.01$ .

The random-walk filter has two modes of operation: acquisition and tracking. During acquisition, the modulator transmits a pattern with guaranteed transitions every symbol. The random-walk-filter constant ( $C$  in Fig. 22) will be set at a low value,  $1 \leq C \leq 100$ , during acquisition to allow rapid symbol synchronization. After synchronization is detected, the random-walk filter is switched to a much higher value during the tracking mode to prevent symbol slippage,  $100 \leq C \leq 10,000$ . The value chosen for  $C$  in the tracking mode is dependent upon the instantaneous jitter requirement given above,  $e_k/T \leq 0.01$ , as well as external factors such as oscillator stability. The actual operational value will be selected experimentally during development.

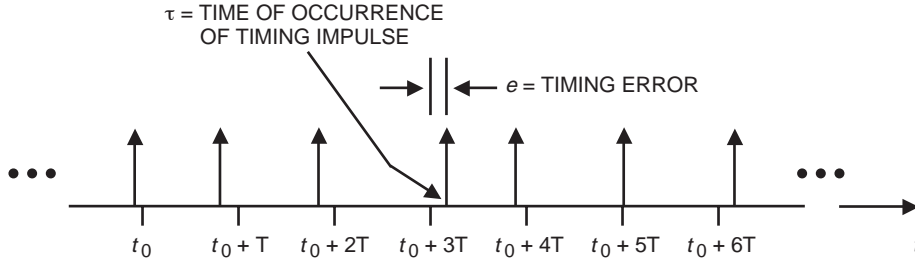


Fig. 23. The symbol-timing signal,  $cl(t)$ .

#### D. Convolutional Decoder

The Viterbi decoder is utilized to provide error correction of the received symbols that were convolutionally encoded at the transmitter. The decoder utilizes soft-decision symbol inputs to calculate the branch and state metrics of the trellis state transition diagram, which are utilized to determine the maximally likely transmitted symbols. The Mentor Iventra Viterbi encoder/decoder soft core was specified and procured as a synthesizable Verilog register transfer level (RTL) model [1]. The parameter description for this core as it was downloaded is delineated in Table 3.

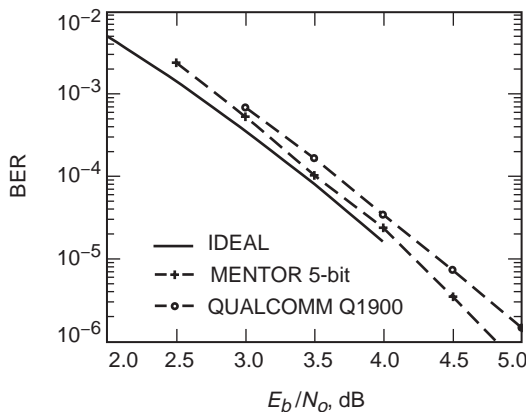
Simulated performance of the decoder using the Mentor bit-true C behavioral model is plotted in Fig. 24. As can be seen from this curve, an  $E_b/N_0$  of 4.25 dB is required to achieve the required  $10^{-5}$  bit-error rate. This curve represents the performance with the input symbols quantized to 5 bits. As a point of reference, the ideal (effectively infinite quantization) decoder performance curve is plotted together with a Qualcomm decoder performance curve (3-bit quantization).

The Mentor Viterbi decoder detects an out-of-synchronization condition but requires implementation of external logic to take the action to swap the input symbols upon indication of an out-of-synchronization condition. Additionally, to be compatible with the CCSDS standard, an inverter must be incorporated in the design external to the decoder, as shown in Fig. 3.



**Table 3. Viterbi decoder parameter description.**

Parameter	Value	Description
$L$	7	Constraint length; length of encoder shift register. The number of trellis states (ACS operations per symbol) is given by $2^{L-1}$ .
$n$	2	Codeword length; number of code bits. The code rate is given by $1/n$ .
$q$	5	Soft-decision word length. Each of the $n$ code "bits" is quantized to $q$ binary bits.
traceback_depth	48	Trace-back/chain-back depth; number of branches in traced-back paths.
g0 - gn-1	$g0 = 171, g1 = 133, g2 = 0, g3 = 0, g4 = 0, g5 = 0, g6 = 0, g7 = 0, g8 = 0, g9 = 0$	$g0$ is the code generating function (in octal) associated with received code "bit" 0; $g1$ is the generating function (in octal) for code bit 1, etc.
ber_insync_en	yes	Determines if BER and synchronization monitor options are incorporated.
symbol_period	800	Number of symbols that define the period to gather the normalization statistics to determine the synchronization status.
swidth	8	State metric word length.
number_of_PEs	64	$1 < \text{number\_of\_PEs} < 2^{L-1}$ . The number of hardware ACS processing elements used to implement the $2^{L-1}$ ACS operations. Setting the $\text{number\_of\_PEs} = 2^{L-1}$ results in a fully parallel hardware implementation offering the highest speed.



**Fig. 24. Mentor Viterbi decoder rate-1/2,  $K = 7, 171/133$  code performance.**

### E. Differential Decoder

The Viterbi decoder output can be passed to the differential decoder. The differential decoder can be enabled or disabled. The differential decoder performs the same differencing operation depicted in Fig. 2 to determine the original transmitted bit. It should be noted that the differential encoder/decoder can cause error multiplication (i.e., one differential decoder input bit error can corrupt two output bits), resulting in a 0.2-dB loss when enabled.

## F. Descrambler

The descrambler follows the differential decoder in the receive chain and can be enabled or disabled. The descrambler is discussed in Section II. It should be noted that descrambling can cause error multiplication and that the Mentor Graphics implementation of the descrambler incurs a 0.3-dB loss when the scrambling/descrambling is enabled.

## V. Summary

This article has provided a detailed overview of the MCAS1 communications system, including the digital design considerations leading to a low-mass and -power transceiver. The system is being designed to operate over a wide range of data rates from 1 kb/s to 4 Mb/s and must accommodate frequency uncertainties up to 10 kHz with navigational Doppler tracking capabilities. As such, the design is highly programmable and incorporates efficient front-end digital decimation architectures to minimize power-consumption requirements. MCAS1 implements most of its transceiver functions with digital FPGA/ASIC technology. This leaves just the RF upconversion and downconversion to be implemented in the analog domain. The approach with the RF subsystem design is to use space-qualified parts when available and leverage the large investment that industry has made in developing highly integrated devices for the commercial wireless markets. The ultimate goal of the MCAS communications effort is to enable reliable communications at a significant mass, power, size, and cost reduction and for a broad class of very small platforms requiring short-range communications.

## References

- [1] "Viterbi Encoder/Decoder Technical Product Brief," Mentor Graphics Corp., 1997.
- [2] "Harris Digital Costas Loop HSP50210 Product Description," File No. 3652.2, Harris Corp., 1996.
- [3] *Deep Space Telecommunications Systems Engineering*, J. Yuen, Ed., New York: Plenum Press, 1983.
- [4] "Analog Devices AD9200 Specification Sheet," Analog Devices, Inc., 1998.
- [5] E. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 155–162, April 1981.
- [6] A. Kwentus, Z. Jiang, and A. Willson, "Application of Filter Sharpening to Cascaded Integrator-Comb Decimation Filters," *IEEE Transactions on Signal Processing*, vol. 45, pp. 457–467, February 1997.
- [7] W. Lindsey and M. Simon, *Telecommunication Systems Engineering*, Englewood Cliffs, New Jersey: Prentice Hall, 1973.
- [8] M. Simon and W. Lindsey, "Optimum Performance of Suppressed-Carrier Receivers With Costas Loop Tracking," *IEEE Trans. on Comm.*, vol. COM-25, no. 2, pp. 215–227, February 1977.
- [9] M. Simon, "Tracking Performance of Costas Loops With Hard-Limited In-Phase Channel," *IEEE Trans. on Comm.*, vol. COM-26, no. 4, pp. 420–432, April 1978.

- [10] S. Aguirre and W. J. Hurd, "Design and Performance of Sampled Data Loops for Subcarrier and Carrier Tracking," *The Telecommunications and Data Acquisition Progress Report 42-79, July-September 1984*, Jet Propulsion Laboratory, Pasadena, California, pp. 81-95, November 15, 1984.  
[http://tmo.jpl.nasa.gov/tmo/progress\\_report/42-79/79H.PDF](http://tmo.jpl.nasa.gov/tmo/progress_report/42-79/79H.PDF)
- [11] M. Simon, "The False Lock Performance of Costas Loops With Hard-Limited In-Phase Channel," *IEEE Transactions On Communications*, vol. VOM-26, no. 1, January 1978.
- [12] C. Cahn, "Improving Frequency Acquisition of a Costas Loop," *IEEE Transactions On Communications*, vol. COM-25, no. 12, December 1977.
- [13] D. Paranchych and N. Beaulieu, "Use of Second Order Markov Chains to Model Digital Symbol Synchroniser Performance," *IEE Proc.-Commun.*, vol. 143, no. 5, October 1996.
- [14] H. Meyr and G. Ascheid, *Synchronization in Digital Communications*, vol. 1, New York: Wiley & Sons, 1990.