

DIGITAL IMPLEMENTATION OF A BPSK DEMODULATOR

Robert C. Meier
Member Technical Staff
Cincinnati Electronics Corporation
Space Electronics Division
Mason, Ohio 45040

KEYWORDS

Demodulation, BPSK, Costas Loop, Early Late Gate

ABSTRACT

Some Telemetry systems today receive a binary phase shift keying modulation format. Typically, to demodulate BPSK requires using a carrier synchronizer followed by a bit synchronizer. Demodulation of BPSK can be accomplished using digital signal processing techniques to implement both synchronizers. This paper describes a digital system that demodulates a 16 KHZ, 2KBPS BPSK signal.

In order to evaluate these techniques, the theory of operation was evaluated. Additionally, a computer simulation of the demodulator was developed. The computer simulation was implemented using Pascal.

The techniques were optimized to give maximum performance while requiring minimum hardware and power in an actual implementation.

INTRODUCTION

This project is a Digital Implementation of a BPSK Demodulator. The project includes investigating a means of performing the carrier sync and the bit sync. A digital implementation of a Costas Loop is used for the carrier sync. An Early-Late Gate technique is used for the bit sync in this project. The project includes a software model of the digital demodulator. This model is used to determine bit error rates for various noise levels.

SYSTEM DESCRIPTION

The BPSK Demodulator is part of a complete transponder system. The demodulator receives a 16 kHz BPSK subcarrier. The data rate is 2 kbits per second. The demodulator generates data and clock as well as a command lock signal. The command lock signal indicates whether or not the demodulator is providing data.

The command threshold for this project was for a receiver capable of producing a command bit error rate of 5.4×10^{-7} when the total input power to the receiver is greater than -115 dBm.

DESIGN APPROACH

The functions of the demodulator are implemented using Digital Signal Processing techniques. The two main sections of the demodulator are the carrier sync and the bit sync. Digital implementation of a Costas Loop is used for the carrier sync. An Early-Late Gate technique is used for the bit sync in this project.

CARRIER SYNC

The first thing that comes to mind when thinking of synchronization is a phase lock loop (PLL). The lock-in ability of the PLL makes it a top choice for a system that has a subcarrier. Therefore, the logical place to begin is with some type of PLL carrier synchronizer.

The basic Costas loop is shown in Figure 1. With no modulation, the Q-arm acts as a PLL with phase detector error voltage at the output of the Q-arm multiplier. With modulation, the Q-arm polarity flips everytime the modulation changes sign. The I-arm multiplier gives a signal in quadrature with the Q-arm. If the phase is nearly correct for tracking, then the I-arm output is the data message.

The goal is to design a system that will sample the carrier at just the right points. The sampling frequency for this project is four times the carrier frequency. The desired sample points occur at the zero crossings and at the plus and minus peaks of the sinusoid. The filter should be designed to maximize the I channel and minimize the Q channel energy. This will occur in the system described above if the I samples occur at the peaks and the Q samples occur at the zero crossings. To accomplish this, the demodulator will alternate between I and Q samples. Additionally, the absolute value of every other pair of I-Q samples will be taken. The sampling scheme is illustrated in Figure 2 for perfect carrier sync. Since the sampling will not, in all probability, begin exactly at the zero crossings and peaks, and since the carrier will not always be exactly 16 kHz, the demodulator must compensate for phase differences and frequency deviations.

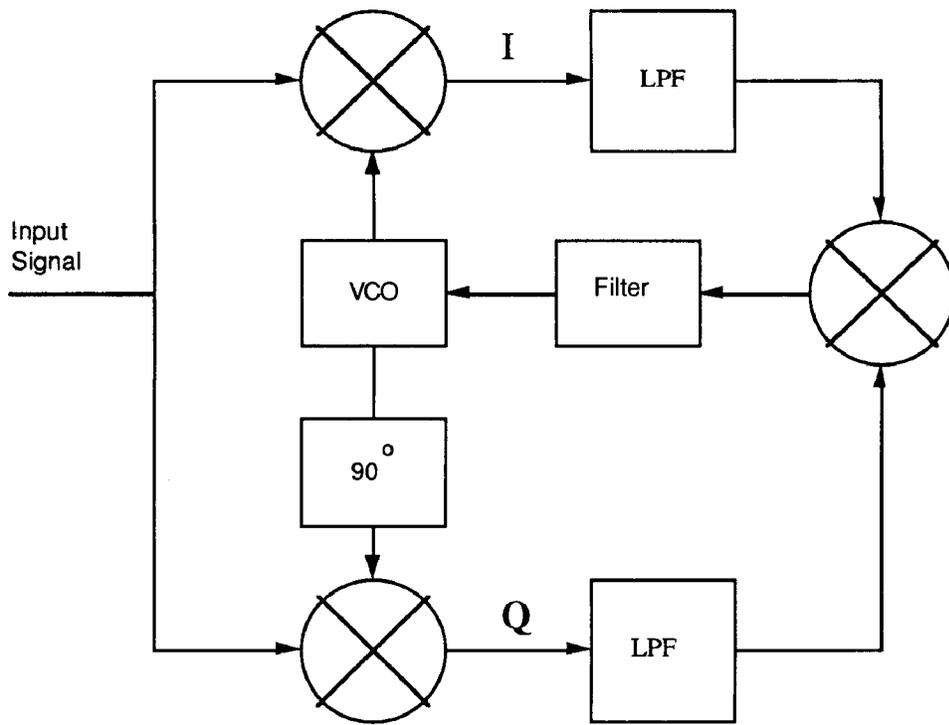


Figure 1 Costas Loop

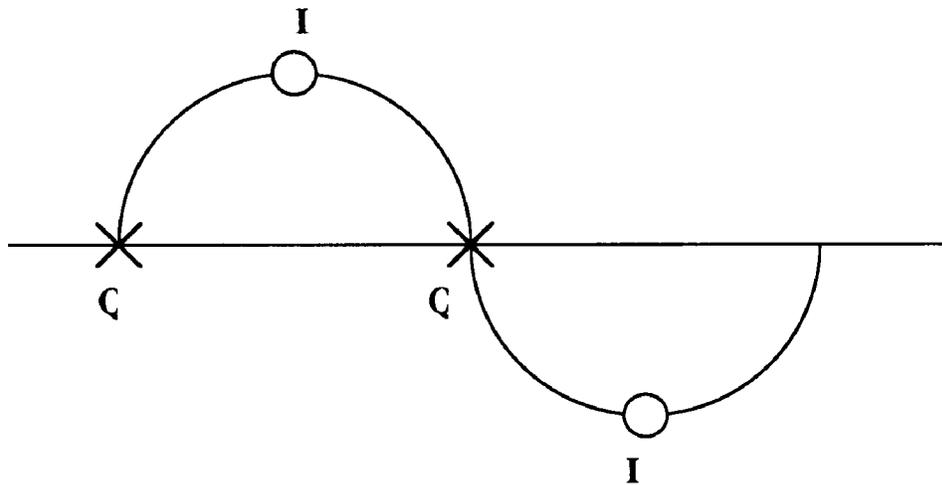


Figure 2 Sampling Scheme

Since digital techniques are being used to accomplish this task, the basic Costas loop will be implemented as shown in Figure 3. The Lowpass filters consist of a pair of integrators. These integrators will sum a total of four samples each. The loop filter will be an Infinite Impulse Response filter. The filter design involves obtaining a loop transfer function and performing a bilinear transform to move into the digital domain. For this project, the passive filter closed loop transfer function was used. Once the transfer function is obtained, the bilinear transform is used to transform the analog filter into a digital filter. The carrier synchronizer block diagram is shown in Figure 4.

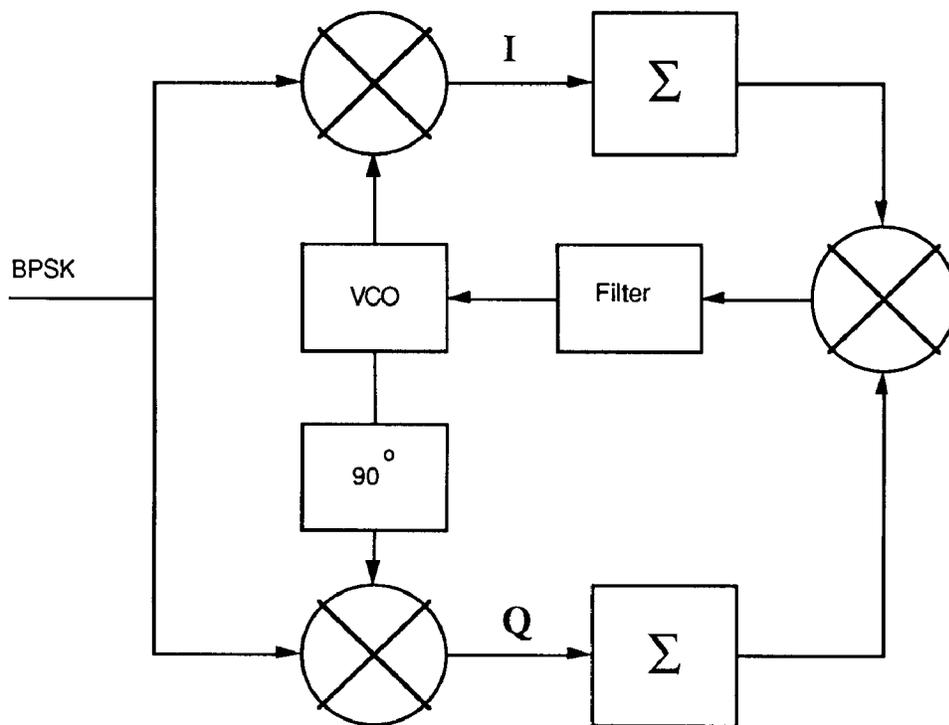


Figure 3 Digital Costas Loop

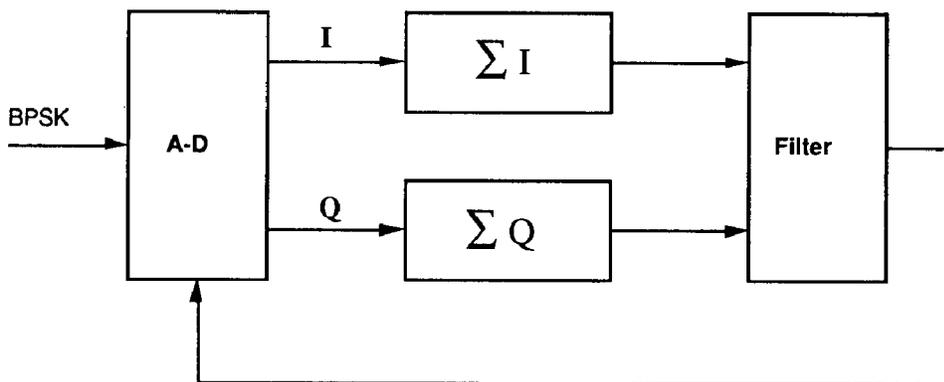


Figure 4 Loop Block Diagram

The coefficients are derived based upon the following qualitative description. The sampling frequency of the system will be set to 64 kHz nominal. It is only a nominal value because it is the sampling frequency that will be adjusted to actually lock to the carrier. So, for a 16 kHz carrier, this results in four samples per period. For 2 kbps data, this results in 32 samples per bit. The system clock frequency of the demodulator system, assuming a 16 MHz clock is used on a TMS320C25 digital signal processor, is $16/4 = 4$ MHz. A divide by four is required because the TMS320C25 uses an internal 4 phase clock. The 4 MHz (250 nanoseconds) is the smallest increment available in the system and will be used to vary the sampling frequency of the A-D converter.

Adding 250 nanoseconds to the sampling frequency every sample changes the sampling frequency from 64 kHz to 62.992 kHz. This sampling frequency would be just right for a carrier frequency of 15748 kHz. Instead, consider adding 250 nanoseconds every eighth sample. By adding 250 nanoseconds every eighth sample, the sampling frequency becomes 63.872 kHz which corresponds to a carrier frequency of 15.968 kHz. This represents a difference of 32 Hz when compared to 16 kHz.

The VCO gain factor is determined by taking the difference between the center frequency of 16 kHz and the carrier frequency corresponding to an adjustment of 250 nanoseconds every eighth sample. This yields the 32 Hz (approximately) value discussed above. The 32 Hz figure is now in terms of Hz per volt. Multiplying the Hz per volt term by 2π gives a result in terms of radians per second-volt.

To obtain an expression for the phase detector gain factor, consider the following. A mixer multiplies two sinusoids together. In a phase lock loop, the phase detector output is proportional to the difference in phase between the two inputs. In the case where the two sinusoids are at the same frequency but different phase, the output of the mixer is four times the DC term ($2 \cdot \cos(x)$). Using the point on the sinusoid where the slope is the largest as a reference and shifting 0.1 radian as the incremental value yields $\pi/2 - 0.1$ radians as the difference. Using this as the 'x' value in the above expression gives 0.1997 volts per 0.1 radian. Assuming an 8 bit A-D converter is used, the peak digital value will be 128. Multiplying 0.1997 by 128 gives 25.5574 volts per 0.1 radian. Finally, multiplying by 10 gives an expression for the phase detector gain factor in volts per radian.

Once the VCO and phase detector gain factors are obtained, the time constants for the loop can be calculated using the equations for the natural frequency of the loop and the damping factor. The natural frequency of the loop must be chosen such that the loop can lock quickly yet be insensitive to jitter. If the natural frequency is too low, the loop will be insensitive to jitter but may be unable to track the signal of interest. For this application, 70 Hz was chosen as the natural frequency to provide a tradeoff between jitter and tracking. The damping factor was chosen to be 0.5. This value was obtained from a plot of noise bandwidth versus damping factor to give minimum noise bandwidth.

Once the time constants are known, the transfer function can be obtained. For this project, the passive filter was chosen. The transfer function for the passive filter is given in terms of the Laplace transform.

The analog transfer function must now be converted to the digital domain. An analog filter can be converted into an Infinite Impulse Filter. Converting the analog transfer function of the loop into the digital domain is accomplished using a bilinear transform. Then the difference equation is obtained and the coefficients are in terms of the time constants.

Several implementation methods are available for IIR filters. The digital filter is shown in Figure 5.

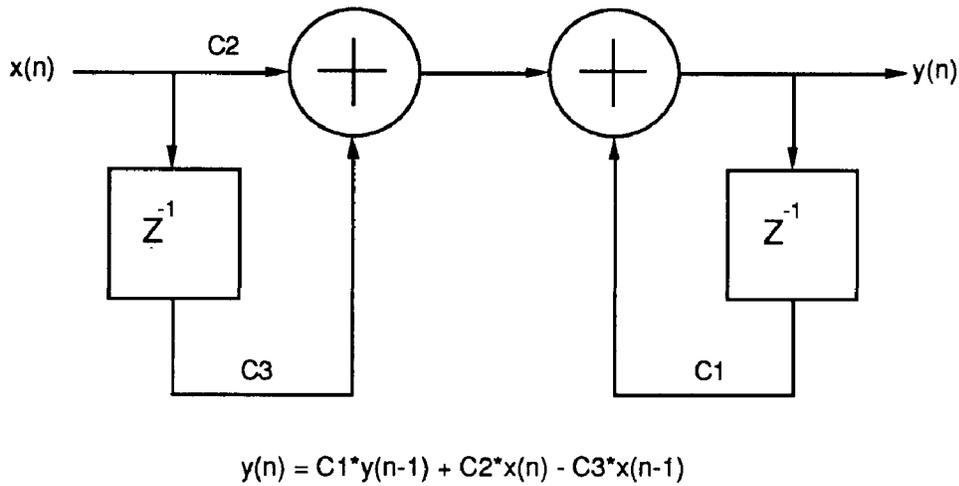


Figure 5 Digital Filter

Clock Sync

Once the system has locked to the carrier, the I channel provides the data. However, even though the carrier lock has been achieved and the system is providing optimum I-arm information, a clock must be generated to accompany the data.

To begin the discussion of clock synchronization, consider the following. Even though the system has locked to a carrier, the VCO in the carrier synchronizer can operate in one of two phases. Therefore, it is not possible to determine whether the current bit is a one or a zero without further information. This is a fundamental ambiguity of all phase-shift modulation formats. If information is transmitted in N different phases, there can be N-fold ambiguity in the data recovery. Ambiguity resolution is accomplished in this system by using a 128 bit preamble. The preamble alternates between one and zero for a total of 128 bits. If the output of the demodulator is inverted during the preamble, then the phase can be flipped by the user to produce true data.

For this project, an Early-Late Gate synchronizer is used. The Early-Late Gate synchronizer is popular for rectangular pulses. This type of synchronizer is shown in Figure 6. Each integrator performs an integration over a time interval of T/2 seconds. The Early gate integrates in the T/2 preceding the location of the data transition. The Late gate integrates in the T/2 following the location of the data transition. For zero timing error, the data transitions occur exactly on the boundary of the early and late gates. The timing error is given by the difference between the early integration and the late integration. The difference produces an error voltage that is applied to the loop.

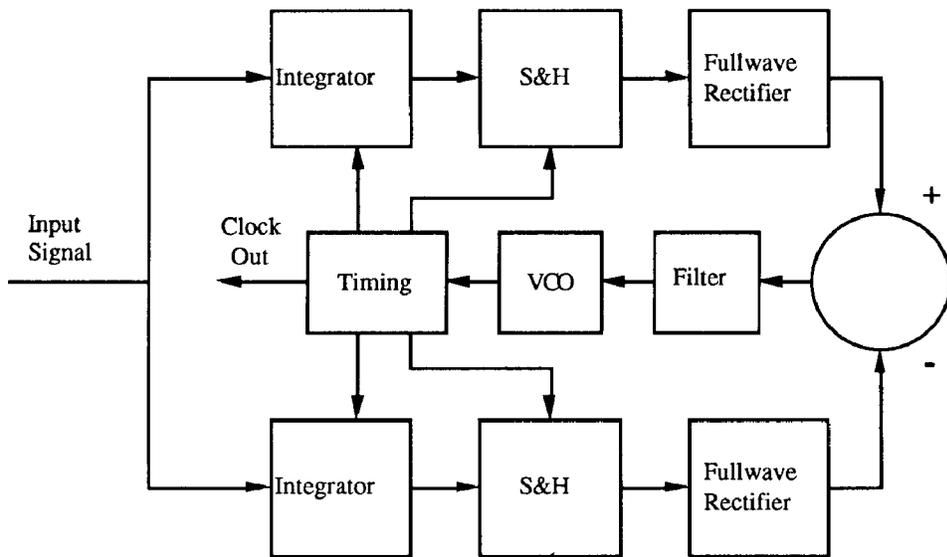


Figure 6 Early-Late Gate Synchronizer

For the clock synchronizer, samples are taken directly from the I-channel of the carrier synchronizer. Therefore, only half as many samples exist on which the clock synchronizer can operate. Figure 7 illustrates five possible cases for the sampling of the I-arm data. The best possible scenario exists for Case 3. In Case 3 the early-late boundary occurs on the data transition. In Cases 2 and 4, the greatest possible error voltage exists. In Case 2, the transition occurs midway through the late gate. In Case 4, the transition occurs midway through the early gate. For Cases 1 and 5, the error voltage is again zero. However the early and late gates have switched positions.

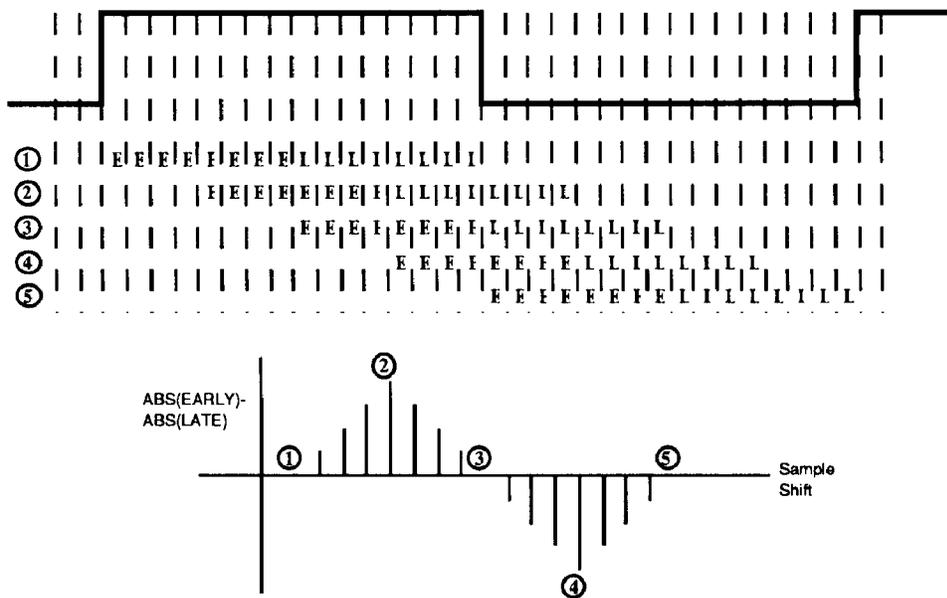


Figure 7 Early-Late Gate Sampling

Since, once again, digital techniques are being used to implement the clock synchronizer, the early-late gate will be implemented as shown in Figure 8. The integrators are implemented as a summation of early and late gates. These integrators will sum a total of eight I samples each. Eight I samples represents half a bit time. The loop filter will be an Infinite Impulse Response filter. The filter design involves obtaining a loop transfer function and performing a bilinear transform to move into the digital domain. For this project, the passive filter closed loop transfer function was used. Once the transfer function is obtained, the bilinear transform is used to transform the analog filter into a digital filter. The approach used on the clock loop filter is the same as that used on the carrier loop filter.

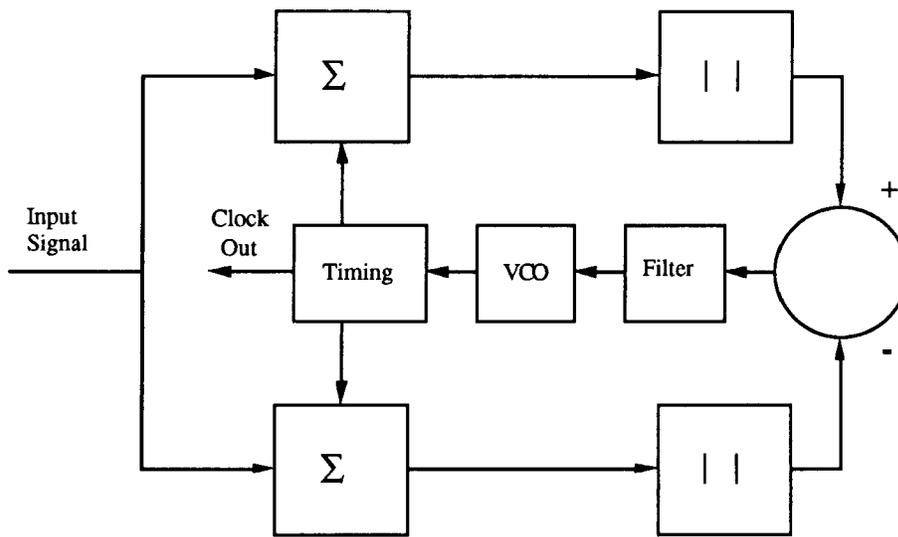


Figure 8 Digital Early-Late

The data rate for the clock synchronizer is 2 kHz nominal. For 2 kbps data, this results in 32 samples per bit. However, only the I samples are used in the clock synchronizer. So, a total of 16 samples per bit are used. Eight of the samples are in the early gate and the other eight samples fall into the late gate. Adjusting the loop by one sample corresponds to moving 125 Hz since $2000/16 = 125$ Hz. Adding 8 milliseconds (125 Hz) to the data rate yields a VCO gain factor of 117 Hz per volt. Multiplying the Hz per volt term by 2π gives a result in terms of radians per second-volt. This is the desired result.

To obtain an expression for the phase detector gain factor, consider the following. For this rectangular NRZ pulse, the amplitude will change from V_{max} to V_{min} for 180 degree change in phase. Assuming an 8 bit A-D converter is used, the peak-to-peak digital swing value will be 256. If each sample represents $180/16=11.25$ degrees, then this becomes the smallest resolution possible in terms of the phase. Dividing 256 by 11.25 gives an expression in terms of 256 Volts/degree. Converting to radians and removing the factor of 256 gives a phase detector gain factor of 5.093 volts per radian.

Once the VCO and phase detector gain factors are obtained, the time constants for the loop can be calculated using the same equations for the natural frequency of the loop and the damping factor as were used earlier. The natural frequency of the loop must be chosen such that the loop can lock quickly yet be insensitive to jitter. If the natural frequency is too low, the loop will be insensitive to jitter but may be unable to track the signal of interest. For this application, 10 Hz was chosen as the natural frequency to provide a tradeoff between jitter and tracking. The damping factor was chosen to be 0.5. This value was obtained from a plot of noise bandwidth versus damping factor to give minimum noise bandwidth.

Once the time constants are known, the transfer function can be obtained. For the clock synchronizer, the same passive filter used earlier was chosen. The analog transfer function must now be converted to the digital domain. An analog filter can be converted into an Infinite Impulse Filter. Converting the analog transfer function of the loop into the digital domain is accomplished using a bilinear transform. Then the difference equation is obtained and the coefficients are in terms of the time constants.

NOISE CONSIDERATIONS

For a binary transmission system, the bit rate is measured in bits per second. For a binary transmission system with a bit rate r_b , average received power S_r and noise density N_o , the system can be characterized by two parameters: E_b and E_b/N_o . The E_b parameter refers to the average energy per bit and is given as S_r/r_b . The E_b/N_o parameter is known as the ratio of the bit energy to noise density. To achieve a bit error rate of less than 5.4×10^{-7} in a PSK system, an E_b/N_o ratio of 11 dB or greater is required.

The noise density expressed in terms of the noise temperature referred to the receiver input is given as 4×10^{-21} Watts per Hz. Converting this value to decibels yields a value of -174 dBm per Hz. A bit rate of 2000 Hz corresponds to $10 \cdot \text{LOG}(2000) = 33$ dB. Adding this to -174 gives -141 dBm per 2000 Hz. Assuming a 5 dB noise figure in the receiver raises the noise level to -136 dBm per 2000 Hz. Since the specification calls for a bit error rate at a signal level greater than -115 dBm, this leaves a margin of 21 dBm E_b/N_o for this system.

In order to use this ratio in the simulator, noise calculations were performed for a variety of ratios to yield a noise multiplier. This noise multiplier takes the sampling rate and the data rate into account. The calculations are given in detail in Appendix C. Table 1 gives the results of the noise multiplier calculations for various values of E_b/N_o .

Eb/No	Noise Power in Signal	Noise Vrms in Signal	Noise Power in Bit Rate	Noise Vrms in Bit Rate	Noise Mult
3	0.2506	0.5006	0,0625	0.2500	2.0024
4	0.1991	0.4462	0.0625	0.2500	1.7846
5	0.1581	0.3976	0.0625	0.2500	1.5905
6	0.1256	0.3544	0.0625	0.2500	1.4176
7	0.0998	0.3159	0.0625	0.2500	1.2634
8	0.0792	0.2815	0.0625	0.2500	1.1260
9	0.0629	0.2509	0.0625	0.2500	1.0036
10	0.0500	0.2236	0.0625	0.2500	0.8944
11	0.0397	0.1993	0.0625	0.2500	0.7972
12	0.0315	0.1776	0.0625	0.2500	0.7105
13	0.0251	0.1583	0.0625	0.2500	0.6332
14	0.0199	0.1411	0.0625	0.2500	0.5643
15	0.0158	0.1257	0.0625	0.2500	0.5030
16	0.0126	0.1121	0.0625	0.2500	0.4483
17	0.0100	0.0999	0.0625	0.2500	0.3995
18	0.0079	0.0890	0.0625	0.2500	0.3561
19	0.0063	0.0793	0.0625	0.2500	0.3174
20	0.0050	0.0707	0.0625	0.2500	0.2828
21	0.0040	0.0630	0.0625	0.2500	0.2521

TABLE 1 Noise Multipliers

Results

The Pascal simulation was actually accomplished in two major parts. The first part was the carrier synchronizer. The output of the carrier synchronizer gave the number of 250 nanosecond shifts required to compensate for frequency and phase differences. With a carrier frequency of 16032 Hz, the output of the filter was a constant one sample shift. This is the shift required every eight samples to adjust 16032 Hz. With an input frequency of 16016 Hz, the output of the filter alternates between zero and one sample shift every eight samples. The filter also adjusted for phase differences. Unlike frequency differences which stay present, the initial phase difference was eliminated immediately. It is interesting to note that, for an initial phase difference of 90 degrees, the I and Q channels come out reversed for the case where noise is not used. Ironically, noise actually helps the loop get the I and Q channels straight when the initial carrier phase is 90 degrees.

The bit synchronizer also worked as planned. However, it was determined that a digital phase lock loop filter was not needed in the implementation of the early-late gate in this

application. By using the difference between the absolute value of the early gate and the absolute value of the late gate, the bit synchronizer worked well. The sample start point was placed at each possible location in the first bit. The early-late gate synchronizer was able to pull the difference, if any, and synchronize the output bit with the I channel data.

The real test of the project was whether the system could meet the E_b/N_0 called for in the specification. The system was tested under a variety of initial conditions and a bit error curve was generated. There are actually 3 bit error curves, one for each of the three carrier frequencies that were tested. Figures 9 thru 11 give the bit error curves for each of the three carrier frequencies. These curves indicate that an E_b/N_0 ratio of approximately 16 dB is required to meet the bit error rate of 5.4×10^{-7} . Since an E_b/N_0 ratio of 21 dB was determined in the noise calculations for this system, this leaves 5 dB of margin for this system.

It is interesting to note the following point about sample clipping. With the sample clipping enabled, a 2 to 3 dB E_b/N_0 benefit was realized. This is due to the fact that any noise added to the sample is eliminated in the I channel for perfect carrier synchronization. So, for a smaller bit error rate, clipping may actually buy some E_b/N_0 margin for a given application.

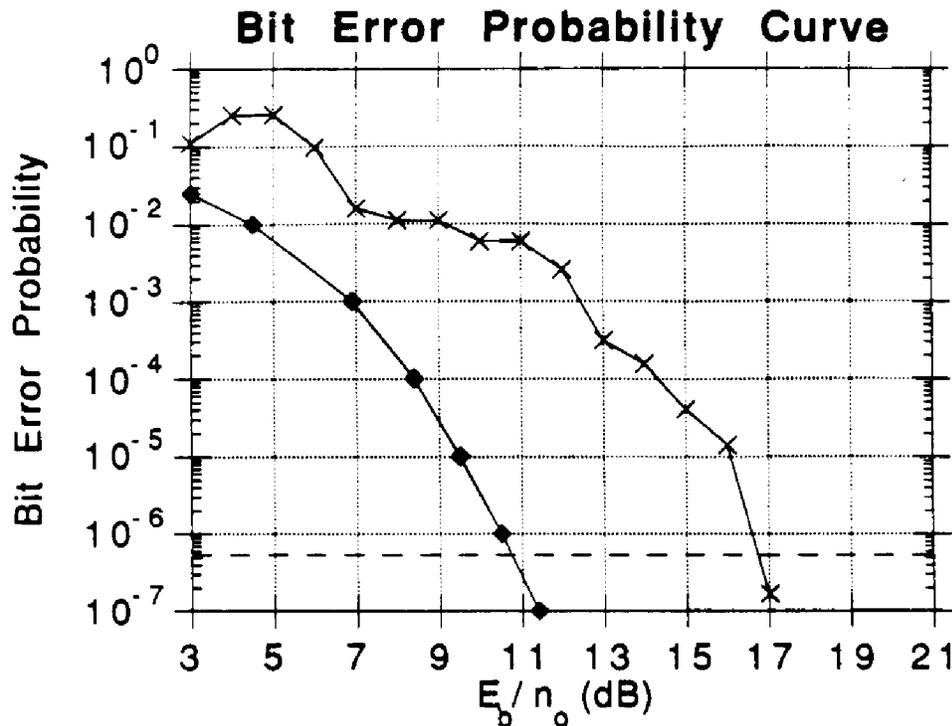


Figure 9 BER Curve F=15930 Hz



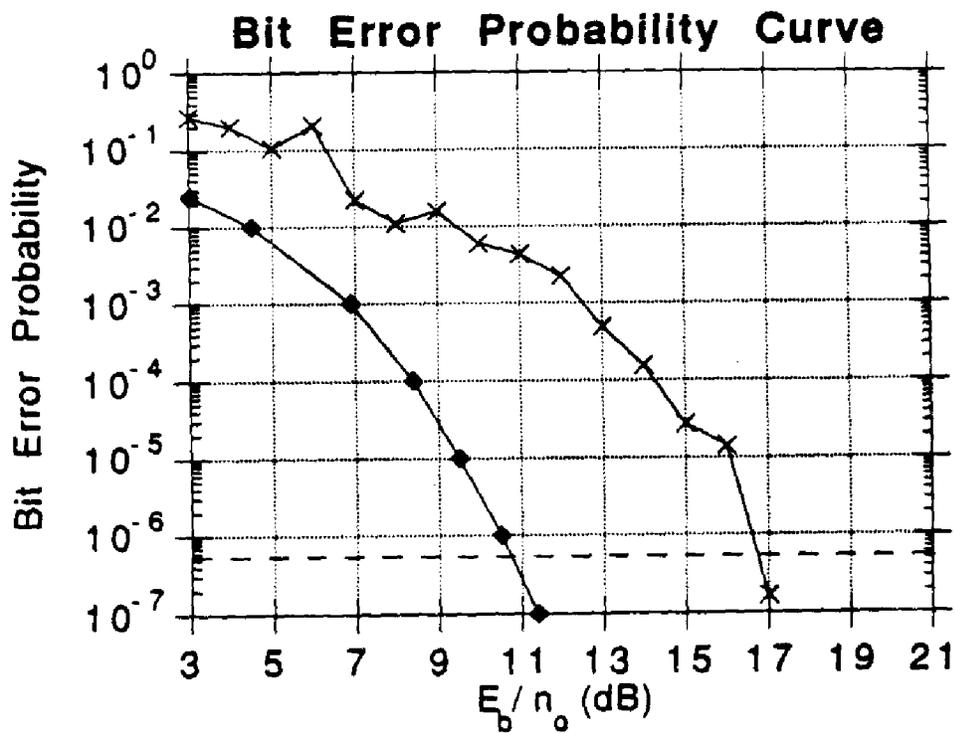


Figure 10 BER Curve F=16000 Hz

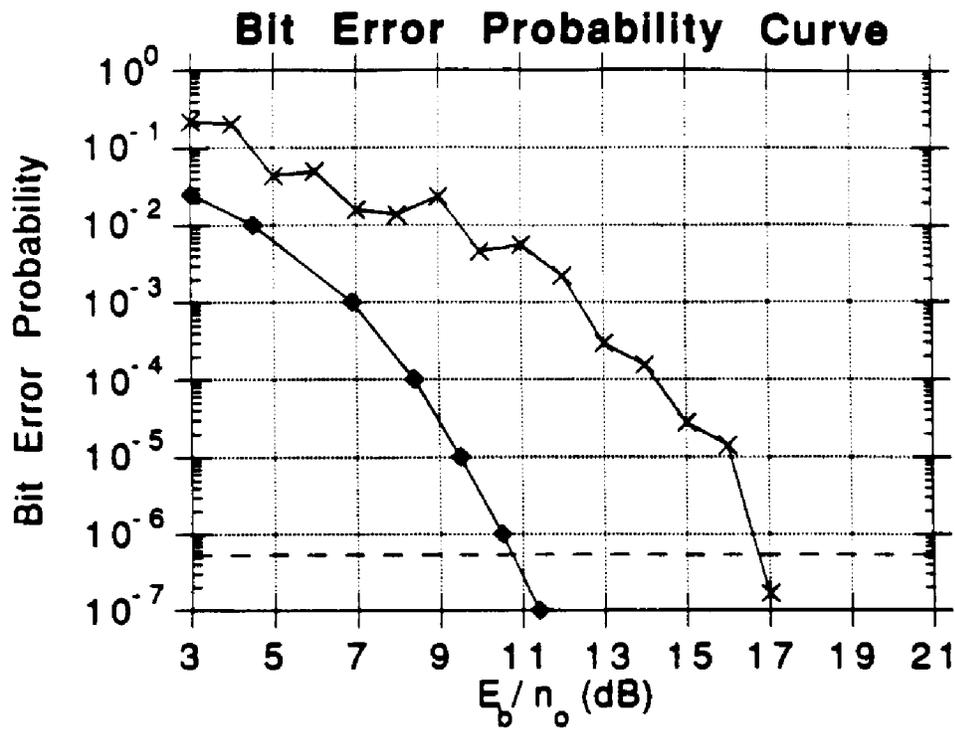
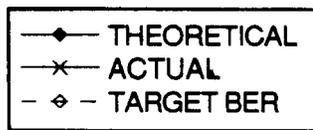


Figure 11 BER Curve F=16070 Hz



REFERENCES

- Carlson, A. Bruce. Communication Systems Third Edition. New York: McGraw-Hill, 1986.
- Chen, Chi-Tsong. One-Dimensional Digital Signal Processing. New York: Marcel Dekker, Inc., 1979.
- Gardner, Floyd M. Phaselock Techniques. New York: John Wiley & Sons, 1979.
- Rabiner, L.R. and Schafer, R.W. Digital Processing of Speech Signals. New Jersey: Prentice-Hall. Inc., 1978.
- Texas Instruments. TMS320C2x User's Guide. Texas Instruments Inc., 1990.