1996

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

BPSK DEMODULATION USING DIGITAL SIGNAL PROCESSING

Prepared By: Thomas R. Garcia

Academic Rank: Assistant Professor

Institution and Department: Purdue University
Department of Electrical Engineering Technology

NASA/MSFC:

Laboratory: Astrionics
Division: Optics and Radio Frequency
Branch: Radio Frequency

MSFC Colleague: Paul Kennedy

Introduction

A *digital communications* signal is a sinusoidal waveform that is modified by a binary (digital) information signal. The sinusoidal waveform is called the *carrier*. The carrier may be modified in amplitude, frequency, phase, or a combination of these. In this project a binary phase shift keyed (BPSK) signal is the communication signal. In a BPSK signal the phase of the carrier is set to one of two states, 180 degrees apart, by a binary (i.e., 1 or 0) information signal.

A *digital* signal is a sampled version of a "real world" time continuous signal. The digital signal is generated by sampling the continuous signal at discrete points in time. The rate at which the signal is sampled is called the *sampling rate* (f_s). The device that performs this operation is called an analog-to-digital (A/D) converter or a digitizer. The digital signal is composed of the sequence of individual values of the sampled BPSK signal.

Digital signal processing (DSP) is the modification of the digital signal by mathematical operations. A device that performs this processing is called a digital signal processor. After processing, the digital signal may then be converted back to an analog signal using a digital-to-analog (D/A) converter.

The goal of this project is to develop a system that will recover the digital information from a BPSK signal using DSP techniques. The project is broken down into the following steps:

(1) Development of the algorithms required to demodulate the BPSK signal.
(2) Simulation of the system.
(3) Implementation a BPSK receiver using digital signal processing hardware.

Algorithm Development

A digital communications receiver extracts the information signal from a digital communications signal. For a BPSK signal, the receiver must be able to recreate a local oscillator (LO) signal that matches the exact frequency and phase of the transmitted signal carrier. A tracking loop for the LO will track any variation in frequency and phase that the transmitter carrier may undergo. For this project a *Costas Loop* is used [1].

The Costas loop generates a phase error term, filters it, then uses this error to adjust the LO phase. Figure 1 is a block diagram showing the algorithms used to implement the Costas loop.

The transmitted BPSK signal is sampled at the receiver by a high speed A/D. The output of the A/D is a sequence of samples called $s(n)$.

$$s(n) = m(n)\cos(\omega_c n + \theta_1) \qquad (1)$$

where: $m(n)$ is the information signal (+1 or -1)
$\omega_c n$ is the carrier frequency
θ_1 is the carrier phase offset

$s(n)$ is downconverted by a complex LO. The downconverted signal may be expressed as:

$$b(n) = s(n)\cos(\omega_c n + \theta_2 n) - js(n)\sin(\omega_c n + \theta_2 n) \qquad (2)$$

where: $\theta_2 n$ is the LO phase offset

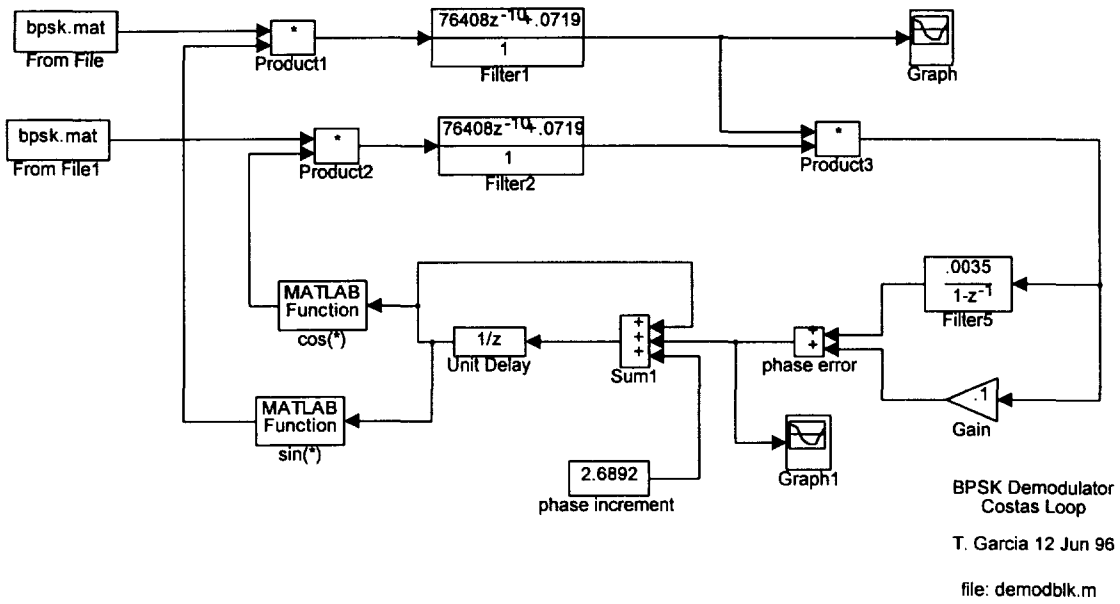

Figure 1 , SIMULINK Block Diagram of a Costas Loop

Substituting the expression for $s(n)$ from equation (1) and filtering out the ω_c products from this operation results in the following:

$$b(n)=\frac{m(n)}{2}\left[\cos(\theta_1-\theta_2 n)-j\sin(\theta_1-\theta_2 n)\right] \quad (3)$$

When the receiver is tracking the carrier, both phase offsets are equal $(\theta_1 - \theta_2 n = 0)$. Then:

$$b(n)=\frac{m(n)}{2} \quad (4)$$

The information signal is recovered.

In order to insure that $\theta_2 n = \theta_1$ the Costas loop is used. First a phase error term is generated. This is accomplished by multiplying the real and imaginary parts of $b(n)$ in equation (2) and filtering out the high frequency terms.

$$q(n)= b(n)_{real} * b(n)_{imag}= m^2(n)\cos(\theta_1-\theta_2 n)\bullet\sin(\theta_1-\theta_2 n) \quad (5)$$

Using a trigonometry identity this may be rewritten as:

$$q(n)=\frac{m^2(n)}{2}[\sin 2(\theta_1-\theta_2 n)] \quad (6)$$

When the phase error, $(\theta_1-\theta_2 n)$, is very small this may be approximated by:

$$q(n) \approx m^2(n)(\theta_1 - \theta_2 n) \qquad (7)$$

This phase error term now passes through an infinite impulse response loop filter that determines the tracking loop dynamics. The output of the loop filter is the phase error term to be used to correct the LO phase:

$$\phi(n)_{error} = \alpha q(n) + \sigma(n) \qquad (8)$$

where $\sigma(n) = \beta q(n) + \sigma(n-1)$

α is a gain term

β is the gain of the recursive part of the filter

Now that the phase error correction term has been determined, the *next* phase increment for the LO may be evaluated.

$$\phi(n+1) = \phi(n) + \omega_c T + \phi(n)_{error} \qquad (9)$$

where $\phi(n)$ is the current phase

$\omega_c T$ is the phase advance due to the carrier frequency

This process is repeated as each sample arrives from the A/D.

Simulation

This system is simulated on a PC computer before hardware is assembled. Simulation on the computer allows the mathematical algorithms for digital signal processing to be developed and analyzed. The characteristics of the receiver under various conditions are analyzed.

Three different simulation methods, MATLAB [2], SIMULINK [3], and C program were used. MATLAB is a high-level numerical computation and graphics package. The Costas loop simulation is written following the above derivation. With the Signal Processing Toolbox digital filters may easily be designed and analyzed. The lowpass FIR filter was designed in this manner. SIMULINK is a graphical modeling and simulation tool for control and DSP systems analysis. It is an add on to MATLAB. The block diagram (Figure 1) for the Costas loop was drawn using SIMULINK. When the simulation is started, SIMULINK performs the mathematical processing of the connecting blocks. The C program was used since the actual DSP hardware code is developed using C. A C compiler then compiles the program into the DSP code and loads it into the hardware.

All three simulation methods yield the same results since the algorithms were identical. The simulation consists of the following components.

(1) information signal generation
(2) BPSK signal generation
(3) receiver

Information Signal Generation

A maximal-length sequence PN code was generated using MATLAB. This is realized using a linear feedback shift register [5]. A sequence of length 127 bits is used. This sequence is used as the information digital signal. In the simulation the bit rate of the PN sequence is 500 kHz.

BPSK Signal Generation

The PN sequence has values of "1" for one state and "-1" for the other. When this sequence is multiplied by a sinusoidal waveform, BPSK modulation results. The multiplication by "1" has no effect of the sinusoid while multiplying by "-1" results in a 180° phase shift. For simulation purposes, the sinusoid signal has a frequency of 10.7 MHz

A signal of maximum frequency content F_{max} will be exactly recovered from its sample values if the signal is sampled at a rate of $F_s > 2F_{max}$. The rate $F_s = 2F_{max}$ is called the Nyquist rate and represents the minimum sampling rate possible [4].

The simulation uses a 10.7 MHz carrier being modulated by a 500 Kbit/sec data sequence. Therefore a sampling rate of at least 11.2 MHz * 2 = 22.4 MHz is needed. A sampling rate of 25 MHz was selected. The sampling rate of 25 MHz results in 50 samples of each information bit and 2.34 samples every period of the 10.7 MHz carrier.

The carrier is capable of being swept in frequency and to have a frequency step. This allows analysis of the demodulation performance when the transmitter carrier frequency is not stationary.

Receiver

The receiver is simulated using the Costas loop algorithms. The MATLAB and C simulations use the mathematical expressions from the algorithm development in basically the same form as described above. MATLAB uses large arrays to contain the input and output data. The C program reads in the BPSK sampled data from a file and writes out the recovered data to a file to simulate actual DSP hardware operation. Plotting of results was done on MATLAB for both methods.

The SIMULINK simulation is done with pre-defined blocks of mathematical operations. These blocks are connected to form the Costas loop and the simulation is started. The BPSK sampled data is read in from a file. The output is plotted in a graph window.

The lowpass FIR filter was designed using MATLAB. MATLAB uses the Parks-McCellan algorithm using the Remez exchange algorithm and Chebyshev approximation theory to design a filter with a minimum error between the desired frequency response and the actual frequency response. The filter length is 20.

Results

The results of a simulation of a swept frequency on the transmitter from 9.7 MHz to 11.7 MHz are shown in Figures 2 - 3. Figure 2 shows the phase error before and after filtering. As the carrier approaches the bandwidth of the system, the filtered phase error ramps up to the right showing that the LO is tracking the ramping frequency shift in the transmitter. When the carrier frequency exceeds the bandwidth on the left side, the loop loses coherency. Notice the bandwidth in which the loop may successfully track the transmitter is approximately 500 kHz. This bandwidth is set by the lowpass filter. Figure 3 shows the recovered data on the I channel.

Hardware

Hardware implementation was not accomplished due time constrains. The DSP hardware for this project consists of the following

Sun	SARCstation 10	Microcomputer
Creative Engineering Concepts	AD700	High Speed Digitizer
Bit3	466	Bus Adapter
Pentek	4272	Multiband Digital Receiver
Pentek	4284	Digital Signal Processor

Conclusion

Digital signal processing techniques may be developed to demodulate digital communication signals. The algorithms developed on this project successfully demodulated a BPSK signal. The carrier frequency was swept and the receiver was able to maintain lock and recover the information signal. The three simulation methods responded in the same manner. The results of the simulation show that the Costas loop designed will track the carrier signal with a $\pm$ 245 kHz frequency variation. The next step in this project is to complete the hardware realization and to develop algorithms for other digital communication signal formats.

References

1. S.A. Tretter, *Communication System Design Using DSP Algorithms*, Plenum Press (1995).
2. The Math Works Inc., *Student Version of MATLAB, Version 4.0* (1994)
3. The Math Works Inc., *Student Version of SIMULINK* (1994)
4. J. G. Proakis, D. G. Manolakis, *Digital Signal Processing, Principles, Algorithms, and Applications*, Macmillan (1992)
5. R. E. Ziemer, R. L. Peterson, *Digital Communications and Spread Spectrum Systems*, Macmillan(1985)

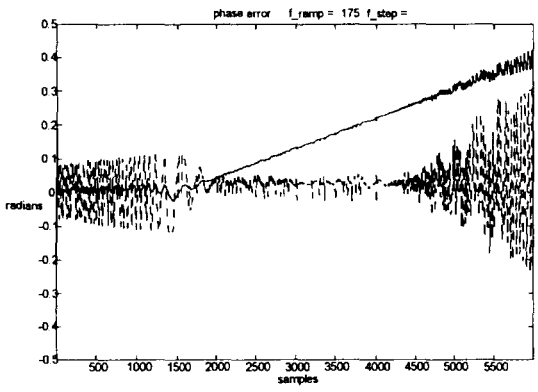

Figure 2, Phase Error filtered and Unfiltered

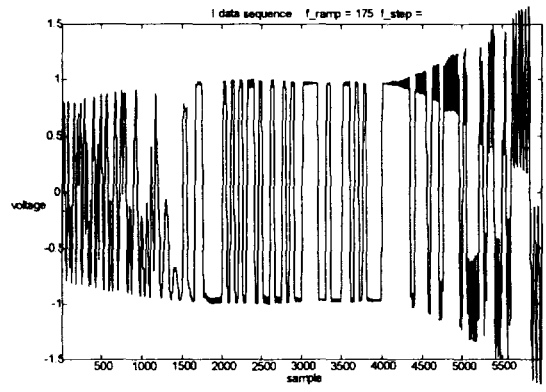

Figure 3, I Channel Recovered Data