

GMSK Demodulator Using Costas Loop for Software-Defined Radio

Santosh Shah and V Sinha
The LNM Institute of Information Technology
 Jaipur - 303012, India
 santosh.jnt@gmail.com, vsinha@iitk.ac.in

Abstract

DSP based implementation of Gaussian Minimum Shift Keying (GMSK) demodulator using Costas loop has been described. The demodulator consists of a Costas loop for carrier recovery, data recovery, and phase detection. For low data rates, we have tested the scheme on a TI's DSP TMS-320C6713. These are implemented using MATLAB and SIMULINK tools and tested on TI's DSP using Code Composer Studio (CCS). Carrier has been recovered using a loop of center-frequency locking scheme as in Binary Phase Shift Keying (BPSK) Costas-loop. Demodulator performance for AWGN channel for different data rates is shown. Performance is measured for Bandwidth Time product ($BT = 0.3$ and 0.5) with data rate between 8 kbps to 256 kbps. Bit synchronization and image coding and decoding are designed according to the need of demodulator. The scheme may find its use in Software-Defined Radio (SDR) applications.

1. Introduction

As in the standard GSM system, GMSK modem is widely used because of its constant envelope and bandwidth efficiency. In this paper we focus on software development of GMSK demodulator. We use Costas loop for demodulator instead of commonly described techniques like four-state adaptive maximum-likelihood sequence estimation (MLSE), coherent and noncoherent detections, or a matched filter detection. Possible application of the concept can be used in Software-Defined Radio (SDR) by replacing all analog components of a transceiver with programmable DSPs. We give particular attention to the system optimization so far as memory requirements for software development are concerned. Mathematical Aspects of the Proposed Method, are discussed in section (2) and in section (3), Demodulator Using BPSK Costas Loop is presented. Section (4) & (5) discuss Bit Error Rate (BER) Performance and the Limitation and Solution of our approach. Section (6) describes TI DSP base Implementation of the model using MATLAB /

Simulink. Finally conclusions are presented in section (7).

2. Mathematical Aspects of the Proposed Method

2.1 GMSK Modulator

The data $d_i(t)$ to Gaussian Minimum Shift Keying (GMSK) modulator is first differentially encoded [6] by performing modulo-2 addition of the current and previous bits, giving $b_i(t)$ (as in the standard GSM system),

$$b_i(t) = d_i(t) \text{ XOR } d_{i-1}(t) \quad (1)$$

$b_i(t)$ has a value either 1 or 0. As input to the GMSK modulator must either be +1 or -1, we convert 1 to -1 and 0 to +1 using,

$$\hat{b}_i(t) = 1 - 2 b_i(t) \quad (2)$$

The modulating data, $\hat{b}_i(t)$ are then passed through a Gaussian filter which has the response $h(t)$ [6]

$$h(t) = 1 \frac{1}{\sqrt{2\pi\sigma T}} e^{-\frac{t^2}{2\sigma^2 T^2}} \quad (3)$$

where

$$\sigma = \sqrt{\ln 2} / (2\pi BT) \quad (4)$$

T is the bit period and B is the 3-dB Gaussian filter bandwidth. The BT product is the relative bandwidth of Gaussian filter which is used in the GSM system and is set to 0.3.

The $\hat{b}_i(t)$ after passing through the filter is then interlaced into odd $\hat{b}_{oi}(t)$ and $\hat{b}_{ei}(t)$ even bits. Modulated signal is then generated by using the following equation [7].

$$v_{GMSK}(t) = A [\hat{b}_{ei}(t) \sin 2\pi (t/4Tb)] \cos \omega_0 t + A [\hat{b}_{oi}(t) \cos 2\pi (t/4Tb)] \sin \omega_0 t \quad (5)$$

here first and second term represents the inphase and quadrature phase components respectively as in Figure 1.

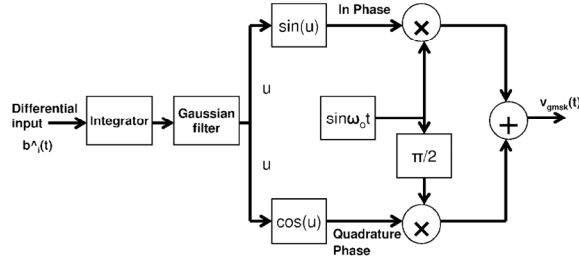


Figure 1. GMSK modulator.

2.2 GMSK Demodulator

As mentioned above carrier is recovered using a loop of center-frequency locking scheme from a BPSK Costas-loop [5]. Subsequently carriers $\cos\omega_0 t$ and $\sin\omega_0 t$ are extracted from Voltage Controlled Oscillator (VCO) and $\pi/2$ phase shifter of the Costas loop. Then $\cos\omega_0 t$ is multiplied with the whole GMSK signal which is received at the receiver and let say an intermediate signal $x(t)$, is produced. Similarly $y(t)$ is produced by using carrier $\sin\omega_0 t$. $x(t)$ and $y(t)$ are given by the following Equations respectively.

$$x(t) = v_{GMSK}(t) \times \cos\omega_0 t \quad (6)$$

$$y(t) = v_{GMSK}(t) \times \sin\omega_0 t \quad (7)$$

When $x(t)$ is passed through a low pass filter inphase component $[b_{ei}(t) \sin 2\pi(t/4T_b)]$ is recovered. Similarly $y(t)$ gives quadrature phase component $[b_{oi}(t) \cos 2\pi(t/4T_b)]$.

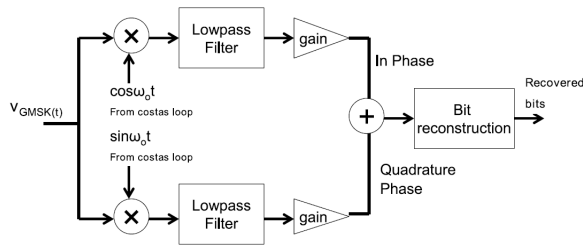


Figure 2. GMSK Demodulator.

Then we generate the components $\sin 2\pi(t/4T_b)$ and $\cos 2\pi(t/4T_b)$ using known values of T_b , and recovered the original odd and even bit sequences $b_{oi}(t)$ and $b_{ei}(t)$ as shown in Figure 2.

3. Demodulator Using BPSK Costas Loop

The mechanism of the Costas loop [1, 2] carrier recovery is to iterate its internally generated carrier from the VCO into the correct phase and frequency based on the principle of coherency and orthogonality. The outputs of the both Low Pass Filter (LPF) give the information about the signals which is modulated using BPSK [8]. But the information we have taken only the carriers coming out from VCO and $\pi/2$ phase shifter (i.e. $\cos\omega_0 t$ and $\sin\omega_0 t$ respectively).

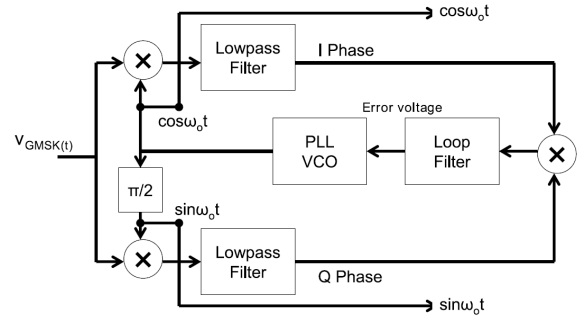


Figure 3. BPSK Costas loop.

Figure 3 shows BPSK Costas loop. Two LPF represents two parallel tracking loops (I and Q) (i.e. Inphase and Quadrature phase), simultaneously a loop filter is used to drive the product of the I and Q components of the signal that drives the VCO. Once the frequency of the VCO is equal to the suppressed-carrier frequency, an error voltage is produced proportionally by the I and Q multiplication, which is passed through the loop filter and then VCO to control the frequency of VCO (i.e. carrier frequency).

4. BER Performance

We present our system response in terms of bit error rate vs. E_b/N_0 plots and compared with [1, 3, 4]. Parameters for simulation are sampling frequency, symbol rate, carrier frequency, data rate (between 8 kbps to 256 kbps), and bandwidth time product ($BT = 0.3$ and 0.5).

We divide our BER performance in two parts according to the equation given below.

$$(E_b/N_0)dB = (S/N)dB - (10 \log(K))dB + (10 \log(f_s/f_b))dB \quad (8)$$

where $K = \ln(M)$, M is constellation point, $1/f_b$ is symbol rate, and f_s is sampling frequency.

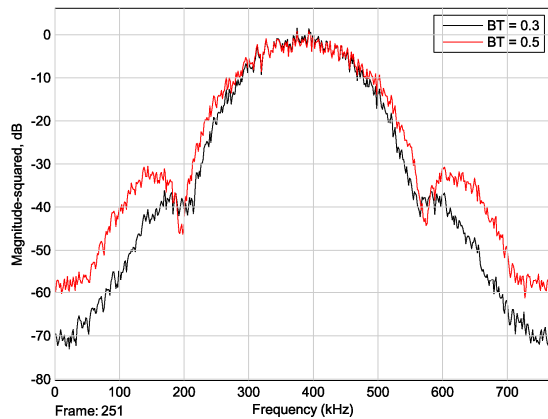


Figure 4. GMSK spectrum for BT = 0.3 and 0.5 with the data rate 256 kbps.

4.1 When sampling frequency is fixed

This is the case in which we take a common sampling frequency for all simulations (as required for DSP). With reference to the Equation 8, when sampling frequency is fixed the E_b/N_0 is directly proportional to symbol rate. A plot for this condition is shown in Figure 5 with data rates (64 kbps and 128 kbps) and BT (0.3, and 0.5). A GMSK spectrum is also presented in Figure 4 with data rate 256 kbps and BT = 0.3, 0.5. Plot illustrates the performance of the system degrade as data rate is increases.

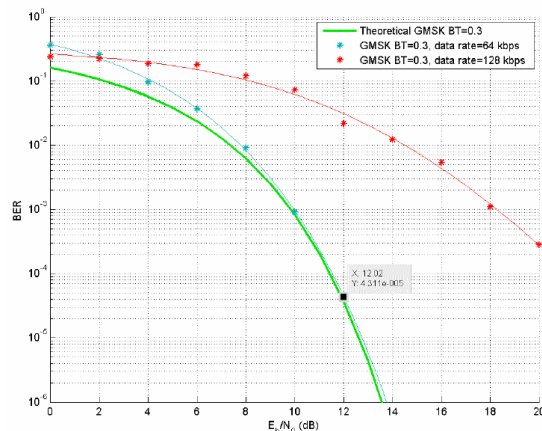


Figure 5. BER vs. E_b/N_0 polt for fixed sampling frequency.

4.2 When carrier frequency is varying

Here we show the performance of system with different data rates, when carrier frequency is varying. We can change the sampling frequency but Figure 6 is for fixed sampling and varying carrier frequencies. Plot

illustrate that performance degrades when one goes for higher data rates with increase in carrier frequency.

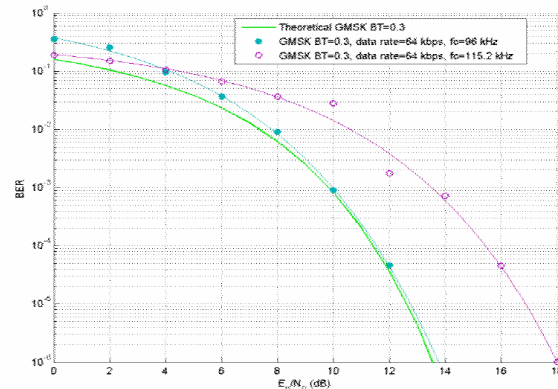


Figure 6. BER vs. E_b/N_0 polt for fixed carrier frequency.

4.3 Some other results from DSK board

Some DSP generated GMSK signals are shown by Figures 7 and 8. Figure 7 shows the inphase and quadrature phase signals with random data sequence. Figure 8 shows the GMSK spectrum (with BT product = 0.1, 0.3 and 0.8) and GMSK modulated signal. Here we have taken bit rate = 8 kbps and carrier frequency = 12 kHz, as our DSP has ADC and DAC with maximum of 96 kHz sampling frequency, but the program can also be used for higher frequencies with appropriate ADC and DAC.

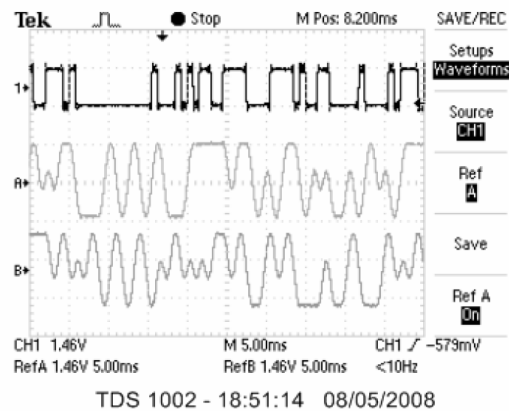


Figure 7. GMSK input bits, Inphase and Quadrature phase signal.

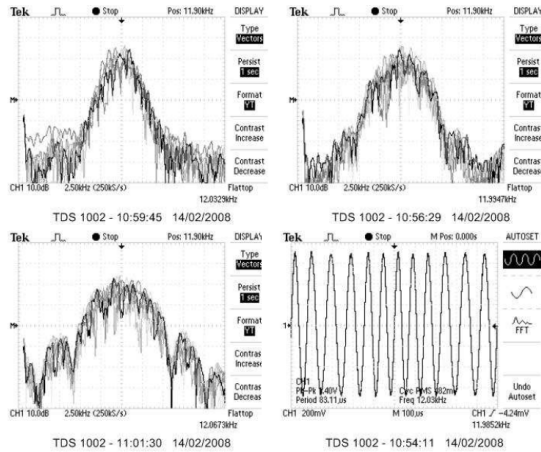


Figure 8. GMSK spectrum for BT = 0.1, 0.3, and 0.8.

5. Limitation and Solutions

The loop filter in Costas loop, for carrier recovery, is to find out the phase changes giving corresponding d.c. values. If the phase change does not occur for a long time, the output of the filter is a zero d.c. value, and hence carrier tracking gets stopped. This condition occurs when the information bit sequence has a 0 or 1 for a long time. To circumvent this to arise we consider three situations for preventing long 0 or 1 bit sequence;

5.1 Voice

When the silence occurs between the communication bursts or get saturated due to sudden noise like thundering. The solution could be Automatic Gain Control (AGC).

5.2 Text data

As we have encoded text data using ASCII (American Standard Code for Information Interchange) codes, the equivalent code for "space" is 20H (i.e. "0 1 0 0 0 0" five continuous 0 sequence) and for "~" is 7E (i.e. "1 1 1 1 1 0" six continuous 1 sequence). With many combinations we reached with the result, that no more than twelve continuous 0's or 1's come in the encoded data sequence. So solution is to keep the loop filter cutoff frequency is $< \text{data rate} / 12$ and overcome the problem.

5.3 Image

Encoding technique for the image is different from the other as image has a range of integer values from 0 to 255 corresponds to dark to bright pixel respectively.

So the integers range 0-255 needs an 8-bit integer to bit converter. If the area of dark or bright pixel in an image is large enough, then binary equivalent sequence (i.e. repeated 0's or 1's) is too large. At this situation the detection is not possible through our method as loop filter of Costas loop stops tracking the carrier signal. One solution for this situation is as follows:

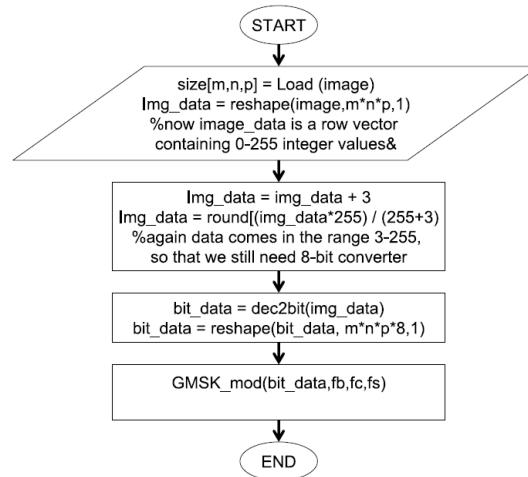


Figure 9. Flow chart for Image encoding.

Encoding process: We have taken the image data and save in a row vector as shown in the flow chart (Figure 9) and then, to avoid the occurrence of data '0', we added integer value 3 to each elements of row vector. Now the integer data range we obtained is 3 to 258, so we needed a 9-bit integer to bit converter. To avoid the need of additional bit, we multiplied with a factor 255*258 to each elements of row vector, and taken the rounding value, which produced integer range from 3 to 255. Now we converted integer to binary data and finally arranged in the bit stream to use in GMSK modulator.

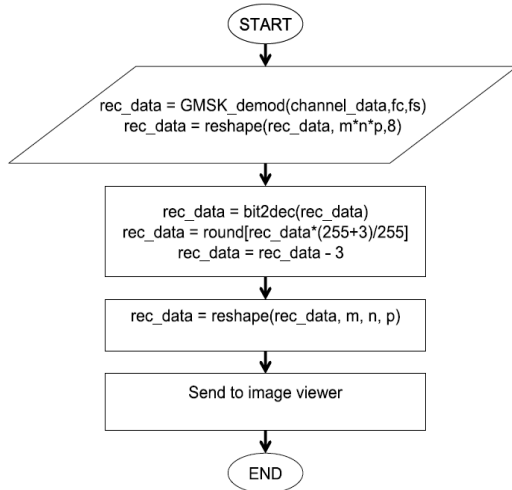


Figure 10. Flow chart for Image decoding.

Decoding process: At the demodulator reverse process has to be done to obtain the original integer data range from 0 to 255 for the image as shown in flow chart (Figure 10).

6. TI DSP Implementation in MATLAB / Simulink

This section gives the reference about our DSK based implementation in MATABL / Simulink. Here two Figures (11 and 12) demonstrate the program which we have implemented in MATLAB and transferred using Code Composure Studio (CCS) in to DSP board.

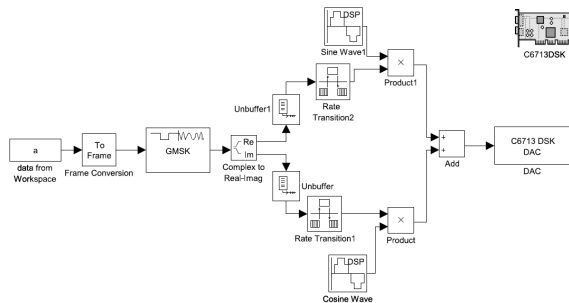


Figure 11. GMSK Modulator implemented in MATLAB for TI DSP TMS-320C6713.

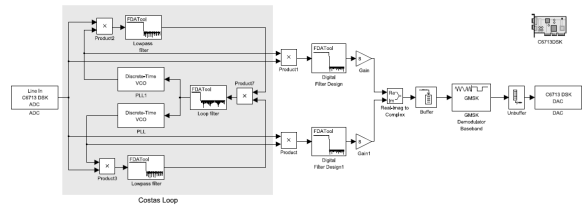


Figure 12. GMSK Demodulator implemented in MATLAB for TI DSP TMS-320C6713.

7. Conclusion

Unlike fixed device the use of DSPs provides us flexibility and reconfigurability and allows us to change the system parameters without changing the hardware. We have demonstrated the performance of the model by changing various parameters. This philosophy can aptly be use in SDR applications. We have shown improvement in BER performance compared to existing methods [1, 3, 4], particularly as Figure 5 and 6. The real time GMSK signal and spectrum is also demonstrated.

Acknowledgment: The financial support from the LNM Institute of Information Technology is acknowledged.

8. References

- [1]. R. M. Gunther M. A. Sessler and E. Vassallo. Gmsk demodulator implementation for esa deep-space missions. In *Proceedings of the IEEE, Contributed paper*, volume 95, pages 2132–2141, November 2007.
- [2]. Y. y. Hiroshi Suzuki and H. Kikuchi. A single-chip msk coherent demodulator for mobile radio transmission. In *IEEE Transaction on Vehicular Technology*, volume VT-34, pages 157–168, November 1885.
- [3]. K. H. Hwang and M. A. Wickert. A soft output gmsk demodulator using a 4-filter mlse for small bt product. *IEEE*, pages 2957–2961, 2002.
- [4]. J. S. Ismail Lakkis and S. Kato. A simple cohearent gmsk demodulator. *IEEE*, pages A-112 – A-114, 2001.
- [5]. G. Mitchell and T. Guichon. Digital costas loop design for cohearent microsatellite transponders. *IEEE AC Paper*, pages 1197–1209, 2002.
- [6]. C.-C. L. Raymond Steel and P. Gould. *GSM, cdmaOne and 3G Systems*. John Willy and Sons, Ltd., 2001.
- [7]. H. Taub and D. L. Schilling. *Principles of Communication Systems*. TATA McGRAW-HILL Edition, New York, 2005