Digital Communication Module for ISTnanosat

André Matias, Gonçalo Tavares
Instituto Superior Técnico – Universidade de Lisboa
INESC-ID
andre.m.mattias@tecnico.ulisboa.pt

Abstract— Nowadays, satellite communications are widely used and there is an increasing need to use digital communication systems implemented using software (Software-Defined Radio - SDR).

This project consists on developing a digital communications module through the implementation of three different full-duplex modems in a Digital Signal Processor (DSP) with ultra-low energy consumption. The purpose of these modems is to allow the conversion between binary information and an appropriate analog signal to be transmitted (and vice-versa) ensuring energetic and spectral efficiency.

The three implemented modems use three different modulations: Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK) and Gaussian Minimum-Shift Keying (GMSK). The BPSK and QPSK modems use phase modulation whereas the GMSK modem uses frequency modulation. The aim of this project was to develop and test an energy efficient and reliable digital communications module, in order to be implemented in a nanosatellite, particularly in the ISTnanosat-1.

The Digital Signal Processor (DSP) used in this project is a Texas Instruments (TI) TMS320C5505 and audio CODEC is a TLV320AIC3204, also from TI. These components are programmed in a TI development board C5505 eZdsp using the Code Composer Studio 5 (CCS).

All implemented modems were developed using a 24 kHz sampling frequency, and the BPSK and GMSK modems work at 6 kbps binary rate whereas the QPSK modem operates at 12 kbps.

Index Terms— BPSK, DSP, GMSK, ISTnanosat, Modem, QPSK, Satellite, Software Defined Radio, SDR.

I. INTRODUCTION

A. Satellite Communications

EVER since technology allows it, satellite communications have been a priority. However, the demanding requirements of this activity (requires resistant materials, communications over very long distances, energetic autonomy, etc.) have always been a great barrier.

The Software-Defined Radio (SDR) technology has seen an increase on its usage and consists on a digital implementation using DSPs (Digital Signal Processors) and/or FPGAs (Field-Programmable Gate Array) components of the radio system. The purpose of SDR is to design more versatile, reliable and cheaper radio transmission systems through digital implementation on DSPs or FPGAs (both reprogrammable).

B. Hardware Used in the SDR implementation

As mentioned before, the SDR techniques use, with few exceptions, DSPs and/or FPGAs.

The advantages of using DSPs stems from several factors, including the very low power consumption, low prices, the possibility of using C programming language and the low latency in real-time processing. However, the intrinsic serial architecture allows little or no parallelism at all, making its processing capability rather limited.

Regarding the use of FPGAs, advantages derive mainly from the processing speed and its inherent parallel architecture. However, these devices are typically more expensive and, for the same task, consume two to three times more power than the DSPs [1]. Additionally, they are programmed using HDL (VHDL or Verilog), a more difficult to work with programming language (comparing to C language).

There are several recent solutions that take advantage of the benefits of the DSPs and FPGAs, such as the one shown in [2]. In order to achieve this goal it is common to use the FPGA only for the demodulating operating at an intermediate frequency, and the DSP is used for baseband signal processing only. Thus, it is possible to take advantage of the parallelism and higher processing power FPGA in demodulation, and take advantage of the low latency and low power consumption of the DSP in baseband signal processing.

C. The ISTnanosat Project

The ISTnanosat-1 project was created in 2010 and is expected to be the first Portuguese satellite entirely developed by teachers, students and radio amateurs.

This satellite mission will try to measure a phenomenon designated “Flyby Anomaly”, which consists on an unexpected change of spacecrafts speed after their gravitational assist when passing by the Earth.

D. Paper Structure

The paper is divided into six key sections: the first describes the problem and the type of a possible solution; the second is used to describe the generic solution and the necessary means for its implementation; the third describes the algorithms to be implemented in each modem; the fourth describes the work performed; the fifth shows some results of the implemented modems; in the sixth a critical analysis is made and some improvements are proposed.
II. SYSTEM ARCHITECTURE

A. Generic Modem

The purpose of this system is to transform digital information (bits) into signals in intermediate frequency (IF) and vice-versa. The generic scheme is represented in Figure 1:

Regarding transmission, the first block represented in Figure 1 is optional and comprises a convolutional encoder, which adds redundancy of information. The second is a bit scrambler that can increase the randomness of bits to send. It is followed by a differential encoder to eliminate the effect of the uncertainty of the absolute phase. The pulse shaping filter reduces the bandwidth (theoretically infinite) of digital broadcasting signals and greatly improves spectral efficiency. Finally, the signal modulates a sinusoidal carrier using BPSK, QPSK, or GMSK modulation [4, 5].

On the receiver side of the modem, the process has the reverse order. The first block is the demodulator, followed by a bit synchronization block that will determine the most suitable instant to sample the signal in order to recover the data. Following the bit synchronization module is the differential decoder, a descrambler and a Viterbi decoder (optional), which are complementary to the differential encoder, scrambler and convolutional encoder blocks (respectively). The output of the Viterbi decoder are the received data bits.

The previously summarized system was implemented using the values shown in Table I:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Frequency</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Carrier Frequency</td>
<td>6 kHz</td>
</tr>
<tr>
<td>Bitrate (Db)</td>
<td>6 kbps</td>
</tr>
</tbody>
</table>

B. Used Hardware and Software

The board used in this project is a Texas Instruments C5505 eZdsp containing the C5505 DSP, the AIC3204 audio CODEC, an USB interface, etc. The DSP has fixed-point arithmetic (with 16-bit words) and it is extremely efficient in terms of power consumption. This has been one of the main reasons for choosing this DSP to be part of ISTnanosat-1, because the power consumption is critical on a satellite.

The software used to program the eZdsp board was the Code Composer Studio 5, which is a development environment based on Eclipse 3.7 and it is optimized for programming Texas Instruments DSPs using C language. It also allows automatic generation of the Assembly code from the C language program.

III. DIGITAL SIGNAL PROCESSING ALGORITHMS

In this solution, two types of signal modulation were used: phase modulation and frequency modulation. The implemented phase modulations were BPSK (Binary Phase-Shift Keying) and QPSK (Quadrature Phase-Shift Keying), whereas the used frequency modulation was GMSK (Gaussian Minimum-Shift Keying).

The BPSK modulation is a two-level phase modulation that induces $180^\circ$ phase jumps to the carrier when there is a bit transition in the modulating signal. The QPSK modulation assigns $0^\circ$, $90^\circ$, $180^\circ$ or $270^\circ$ carrier phase jumps, depending on the value of two bits of the modulating signal {"00", "01", "11" or "10"}.

Regarding the used GMSK modulation frequency, it consists on a numerically controlled oscillator (NCO), whose center frequency is the center frequency of the carrier and whose frequency variation is controlled by the output pulses of the modulating signal (bits), after passing through a Gaussian filter. It is noted that this frequency variation is controlled so that the carrier is a continuous signal using a modulation index $m = 0.5$, and thus, a highly spectrally efficient solution.
A. BPSK Modem

The layout of the implemented BPSK emitting and receiving system is the same as the one represented in Fig. 1, aside from the convolutional encoder and its matching Viterbi decoder.

In this implementation all of the blocks operate at a sampling frequency of 6 kHz, except for the modulator and demodulator (24 kHz) and the bit synchronizer (12 kHz).

1) Scrambler and Differential Coder

The transmitter uses the input data (bit\(_n\)) that presents itself as a sequence of bits that do not possess the necessary emission characteristics. On the one hand, it can contain long sequences of equal logic value, and on the other hand is not totally random.

In order to make the bits more random, the initial emitting block is a bit scrambler, and it is implemented using the following polynomial:

\[
P(x) = 1 + x^{-6} + x^{-7}
\]  

The scramblers output is then connected into a differential encoder which avoids the effect of the channel phase ambiguity, and it performs the operation of equation 2:

\[
d_n = d_{n-1} \oplus e_n
\]

2) Pulse Shaping Filter e Modulator

Before filtering it is necessary to convert the bit values ('0' or '1') into NRZ values ('1' or '1' respectively), and it is the NRZ signal that is filtered by a raised cosine pulse shaping filter with \(\alpha = 0.5\).

The raised cosine filter is given by:

\[
H(f) = \frac{T}{2} \left(1 + \cos \left(\frac{\pi f T}{\alpha} \left(1 - \frac{1}{2T} \right)\right)\right)
\]

being \(T\) the duration of one symbol, and \(0 < \alpha \leq 1\) the parameter that controls the bandwidth excess between zero (\(\alpha = 0\)) and \(1/2T\) (\(\alpha = 1\)).

The BPSK modulator is conceptually very simple, performing the following operation:

\[
y_n = x_n \cdot \cos(2\pi f_c t)
\]  

being \(f_c\) the center frequency of the carrier signal to be modulated by the output of the pulse shaping filter \(x_n\).

3) Costas Loop

The first block of the BPSK receiver is a demodulator. In this implementation, the used algorithm was the Costas Loop which consists of a phase locked loop (PLL) that, in addition to carrier synchronization, allows the demodulation of BPSK signals simultaneously. Fig. 2 shows the block diagram of the Costas Loop algorithm.
4) Time Error Detector Loop

After the Costas Loop it is necessary to recover the bits logical value from the demodulated pulses. This process is done by the bit synchronization loop, which uses the algorithm of Time Error Detecting (TED) [5], [6], [7]. It should be noted that the feedback control loop for bit timing recovery is valid for any input signal whose shape is a succession of pulses. The layout of this feedback loop is shown in Figure 3.

The bit synchronism is done using an interpolation technique. This technique consists in computing an estimate of the signal at an arbitrary time between two received samples $\mu_k$.

The TED algorithm uses three successive interpolated samples (the previous decision interpolated sample, the current intermediate interpolated sample and the current decision interpolated sample).

This algorithm generates an error signal proportional to the delay/advance between the current interpolation point and the optimum interpolation point. The loop filter that receives the TEDs output is done using a first order IIR filter and performs a weighted average value between the current error value and the previous error value. This filter smoothes the variations of the NCO control word and allows the NCO to stabilize, on average, at the frequency that leads to an optimum instant interpolation. The samples to be interpolated are given by the NCO underflows, meaning that a flag is activated indicating the necessity of interpolation on the next sample, in order to perform interpolation in time $\mu_k$ obtained by the absolute value of the NCO word.

It should be noted that the TED algorithm only uses two interpolated samples.

5) Differential Decoder and Descrambler

The differential decoder and the descrambler perform the complementary operation of those done by the differential encoder and the scrambler (respectively). Thus, at the output of the descrambler, in the absence of noise, the original sequence of bits can be recovered.
B. QPSK Modem

The QPSK modem is identical to the BPSK modem, at a high hierarchical level. However, since the QPSK maps two bits per symbol (two times the BPSK spectral efficiency) there are some changes to be made concerning the BPSK modem.

The differential coding has to be done at symbol hierarchy level, and whereas in the BPSK modulation there was only one bit per symbol, in the QPSK modulation there are two. Thus, whereas the BPSK modem performed a XOR operation between the current and the previously sent symbols (2), the QPSK modem has to perform a two dimensions differential coding using a table. As expected, the differential decoder performs the coding complementary operation (using a matched table [8]).

Previous to the pulse shaping filter, and after the computation of the bitstream to be sent, the latter is again divided into two bitstreams (in-phase signal and quadrature signal), each bitstream being passed through the filter like a BPSK signal. The modulator block operates as two BPSK modulators, in-phase signal modulates the cosine carrier wave signal and quadrature signal modulates the sine carrier wave. The outputs of the modulators are subtracted and sent to the channel.

In addition to the aforementioned change in the differential decoder, the greatest difference on the receiver side of the modem is the structure of the demodulator, that is shown in Fig. 4:

![QPSK Demodulator](image)

Fig. 4. QPSK Demodulator.

Despite the scheme suggestion that, after the hard decisions, there is a multiplication, one of the operands has NRZ values (‘1’ or ‘-1’), and can be implemented through a simple signal change when the hard decision has the value ‘-1’. Thus, the computational effort of the alternative scheme is very similar to the one used in the BPSK modem.

C. GMSK Modem

The layout of the emitting and receiving GMSK system is implemented as shown in Fig. 1.

In this implementation all of the blocks operate at a sampling frequency of 6 kHz, except for the modulator and demodulator (24 kHz) and the bit synchronizer (12 kHz).

The implemented GMSK transmitter is practically the same as BPSK modem up to the pulse shaping filter, due to the fact that the same scrambler algorithm is applied to the bits and the same differential encoder is applied to the output of the scrambler. It is however preceded by a (optional) block that applies convolutional coding to the bitstream.

1) Gaussian Filter

The first major difference compared to BPSK transmitter is the fact that the pulse shaping filter has a Gaussian step response, given by:

\[
g(t) = \frac{1}{2} \cdot \left[ \text{erf}(\beta t) - \text{erf}(\beta (t - T)) \right]
\]

where

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt
\]

The Gaussian filter was implemented using a FIR filter with a variable number of coefficients, depending on the value of BT (BT = 0.5 and BT = 0.3). The filter includes an interpolator with an interpolation factor L = 4.

The method used for obtaining the coefficients of the Gaussian filter was sampling the response g(t), being the normalized sampling instants multiples of 1 / L and contained in the interval \{N / 2; N / 2\}, where N is the number of symbols that interfere with each other, causing intersymbol interference (ISI).

2) Integrator and FM Modulator

The integration of the Gaussian filters output is performed using an integrator ramp, from a numerically controlled oscillator (NCO).

The frequency modulation is performed using the relaxation oscillator whose expression is given by

\[
f_0(t) = \frac{f_0 \Delta(t)}{2\pi} = f_c + f_m \cdot g(t)
\]

being \(f_c\) carrier center frequency, \(f_m=\pm \frac{f_0}{4}\) the frequency to be multiplied by the output of the gaussian filter \(g(t)\), in order to make possible this modulation to have a modulation index \(m=0.5\) (Minimum-Shift Keying).

3) FM Demodulator

In order to perform GMSK demodulation, a quasi-coherent architecture has been adopted, this being a very computationally efficient architecture that does not cause a significant loss of performance.

The implemented solution is represented in Fig. 5:
GMSK modem, mainly due to the quasi-coherent ("free-running") receiver, that has the best time performance of all the receivers. The BPSK modem has a very similar time performance comparing with the GMSK modem, due to the inherent simplicity of this modulation scheme. As it should be expected, the QPSK modem is the computationally more demanding modem, but it should be noted that the bitrate of this modem is two times faster than the others.

In this project the detailed study of the modems response in the presence of noise was not possible to perform because it was not possible to get access to the required equipment for accurate measurements of the signal power and noise power. Therefore, it is only presented the theoretical behavior in the presence of white noise. The bit error rate (BER) theoretical curves [10] are represented in Fig. 7.

As it is perceptible, the BPSK modem has the best theoretical response in the presence of Gaussian noise, whereas the QPSK modem presents the worst.

### IV. RESULTS AND PERFORMANCE ANALYSIS

The processing timings of the transmitters, receivers and full modems are shown in Table II:

<table>
<thead>
<tr>
<th></th>
<th>Processing Timing (us)</th>
<th>Percentage of the Sampling Period (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK Transmitter</td>
<td>7.7</td>
<td>18.480</td>
</tr>
<tr>
<td>BPSK Receiver</td>
<td>17.92</td>
<td>43.008</td>
</tr>
<tr>
<td>QPSK Transmitter</td>
<td>11.32</td>
<td>27.168</td>
</tr>
<tr>
<td>QPSK Receiver</td>
<td>22.52</td>
<td>54.048</td>
</tr>
<tr>
<td>GMSK Transmitter</td>
<td>8.8</td>
<td>21.120</td>
</tr>
<tr>
<td>GMSK Receiver</td>
<td>16.24</td>
<td>38.976</td>
</tr>
<tr>
<td>BPSK Modem</td>
<td>25.62</td>
<td>61.488</td>
</tr>
<tr>
<td>QPSK Modem</td>
<td>33.84</td>
<td>81.216</td>
</tr>
<tr>
<td>GMSK Modem</td>
<td>25.04</td>
<td>60.096</td>
</tr>
</tbody>
</table>

As can be seen in the table, the more efficient modem is the

---

**Fig. 5.** GMSK Demodulator.

**Fig. 6.** BPSK Modem Test Results.

It can be observed that the pulse shaping filter has a 50% excess bandwidth, and thus the 6 kbps BPSK signal has a bandwidth of (6x1.5) = 9 kHz. It is also perceptible that the transmitted BPSK signal is centered at the carrier center frequency, 6 kHz.
Fig. 7. Theoretical BER response of BPSK modem (Blue), GMSK modem with BT=0.5 (Green), GMSK modem with BT=0.3 (Pink) and QPSK modem (Red).

B. QPSK

Fig. 8 shows the transmitter in-phase bitstream (yellow) and the two received bitstreams, after the bit synchronization loop, being the blue bitstream the in-phase received signal and the pink bitstream the received quadrature signal. The figure also shows the spectrum content of the transmitted QPSK signal (below).

As can be seen, the blue signal (received in-phase signal) has no error comparing to the yellow signal (transmitted in-phase signal), just a delay, thus demonstrating the proper functioning of the modem. Beside that fact, it is noted that the spectrum content of the QPSK signal is centered at 6 kHz (carrier center frequency) and that the signal does not exceed the limit imposed by the Nyquist theorem.

C. GMSK

Fig. 9 represents the modulating signal (yellow), the demodulated unfiltered signal (pink), the demodulated and filtered signal (in blue) and also the spectrum content of the transmitted signal, after modulation (bottom).

After demodulation, as already mentioned, was applied a low pass FIR filter with 20 coefficients, in order to eliminate the high frequency component (noise). The filter output is represented in blue, and it is clear the almost total attenuation of the high frequency component, thereby improving data recovery and decoding conditions and performance.
V. CONCLUSIONS AND FUTURE WORK

As described in the previous section, any of the three modems shows promising results, guaranteeing a good performance if they were to be implemented in ISTnanosat-1.

The BPSK modem has been implemented using a simple and intuitive structure that allows very similar processing timings to the GMSK modem (the fastest implemented modem). In addition, the BPSK modem has the best theoretical response to Gaussian noise.

The advantages of the GMSK modem are mostly the processing timing (fastest implemented modem) and the robustness to variations in the carrier center frequency (due to Lyon algorithm).

Finally, the QPSK modem has as main advantage the fact that the speed of the bitstream is twice the speed of any of the other implemented modems.

Having in mind the global development of this project, there are some aspects that should be considered for the implementation of the digital communications module on the ISTnanosat-1.

The first aspect is related to the need to build a transponder with reduced noise, since the used development board (reduced price and with a large number of features) has too much noise to ensure a high-performance implementation.

The second aspect should be the exact measure of the modems performance in the presence of noise, with suitable gear and after implementing the transponder mentioned above. Then, the three modems may be incorporated in the satellite transponder, due to its SDR nature.

In the developed implementation, due to the high computational complexity of the Viterbi decoding, when this algorithm was used the sampling frequency had to be reduced four times, in order to allow the proper processing timing for the Viterbi decoding. This means that the bitstream speed, when using Viterbi decoding, would fall four times as well. This way, if the GMSK modem is implemented in the DSP used in this project, this algorithm and the complementary convolutional encoding should not be considered.

Apart from this fact, Viterbi decoders typically have two components: decoding the convolutional code using the Viterbi algorithm, and channel estimation and subsequent matched filtering. This last feature can greatly improve the performance of the GMSK modem, but it has an extremely high computational complexity. Thus, in order to significantly improve the performance of the GMSK modem, it would require much more processing power, and it is recommended the use of a hybrid system (FPGA and DSP), such as the one used in the ESA paper [2].

Since ISTnanosat-1 will necessarily have at least one FPGA board with great processing power [11], it is recommended the use of part of its processing capacity, in order to perform the Intermediate Frequency (IF) processing, including the channel matched filtering and Viterbi decoding. The ultra low power DSP should only be used for the baseband processing, because it is computationally much less demanding.

Then, the intermediate frequency processing would consist on carrier modulation and the eventually the use of the full Viterbi decoding (convolutional decoding using Viterbi algorithm and matched filtering after channel estimation). The baseband processing would mostly consist on performing the scrambler and descrambler operations, differential coding and decoding, and the bit timing recovery using the time error detector loop.

VI. REFERENCES