Software $\pi/4$-DQPSK Modem:
A Student Project Using the TMS320C6201 EVM Board

Stephan Weiss, Steve J. Braithwaite, and R. David Stewart

Communications Research Group
Dept. of Electronics & Computer Science
University of Southampton
Southampton SO17 1BJ, UK
{sw1,sjb,rds}@ecs.soton.ac.uk

Abstract. This paper reports on a student project performed at the University of Southampton jointly by 4th year MEng students within the course "Advanced Radio Communications". The aim was to design a software modem capable of transmitting 16kb/s of data, whereby random number generation, advanced modulation, pulse shaping, synchronisation, and error counting techniques had to be applied. The ultimate aim was the implementation on a Texas Instruments TMS320-C6201 EVM board, which dictated some of the specifications of the design.

1. INTRODUCTION

Communications theory is an essential part of the 3 and 4 year courses in electronic engineering related studies at the University of Southampton. Besides a strong emphasis of the fundamentals, the students are encouraged to also gain experience on a systems level. For this purpose, a course "Advanced Radio Communications" for 4th year MEng student in the academic year 1999/00 was concerned with the design and implementation of a software $\pi/4$-DQPSK modem. In this paper, we aim to give an overview of the aspects considered in the modem, and how the implementation was organised in order to permit modularity and testing.

The overall setup of the software modem is shown in Fig. 1, with independent units for transmitter and receiver. The transmitter maps an incoming bit stream onto inphase and quadrature signals modulated by $\pi/4$ differential QPSK [1]. After upsampling both inphase and quadrature component, Kingsbury filters $h_1[n]$ are applied for pulse shaping in order to restrict the bandwidth and to prohibit intersymbol-interference (ISI) on the receiver end [3, 5]. After digital-to-analogue conversion, standard laboratory hardware can be used to further modulate the signals, add noise interference, or apply amplitude and phase distortion to the signals.

The receiver was designed to run at 16$\times$ the symbolrate to provide high resolution for the clock recovery. As the ADC imposed restrictions, the analogue inphase and quadrature signals were sampled at 32kHz and upsampled to 128kHz. The filters $h_2[n]$ are suitably selected Kingsbury filters [3]

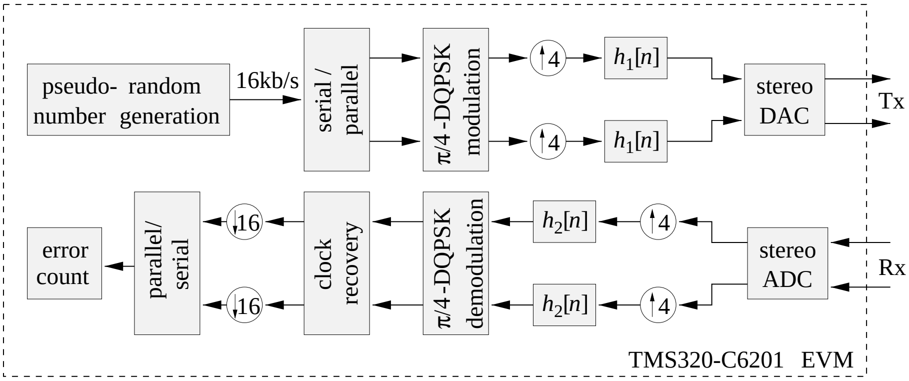

Fig. 1: Baseband unit of a 16kb/s modem on a TMS320-C6201 EVM board producing inphase and quadrature signals; the transmitter and receiver units are independent and can be placed on separate DSPs.

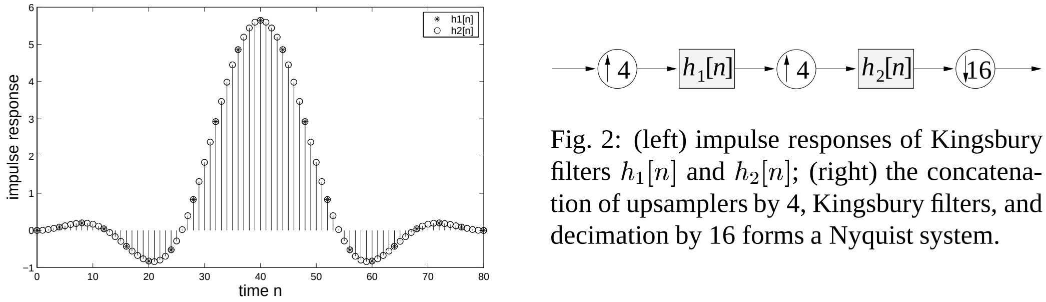

Fig. 2: (left) impulse responses of Kingsbury filters $h_1[n]$ and $h_2[n]$; (right) the concatenation of upsamplers by 4, Kingsbury filters, and decimation by 16 forms a Nyquist system.

working as both interpolation filters and pulse shaping / matched filters. After $\pi/4$-DQPSK demodulation, and a robust clock recovery, the originally transmitted bit stream should have been recovered.

For testing purposes, the transmitted bit stream was created on-board by a pseudo-random number generator, as shown in the block diagram in Fig. 1. When synchronised, the same generator could be employed in the receiver part to count potential errors occuring due to channel distortions introduced between transmission and reception of the signal. This distortion could either be imposed on the external IF or HF lab equipment, or in a testing stage by adding Gaussian noise to an internal link from the transmitter output to the receiver input.

In the following, we discuss the system that was implemented by our students as course work. Sec. 2 will be dedicated to the transmitter circuit, including both the data generation, modulation, and pulse shaping. The transmission through an internal or external channel is subject of Sec. 3. Details of the receiver are disclosed in Sec. 4, and the error counting method for testing and simulation purposes are described in Sec. 5. Finally, the organisation of students' contributions and a brief explanation of the implementation on the TMS320-C6201 EVM are reported in Sec. 6.

2. TRANSMITTER

On the transmitter side, a pseudo-random number (PRN) generator is employed to produce a bit stream at 16kbits/s. The PRN are produced by a linear feedback shift register of 32 bits length, and feedback is based on a primitive polynomial yielding a maximum length sequence that only repets after 2^{32} iterations. As indicated in Fig. 1, this bitstream is multiplexed onto an inphase and quadrature channel both running at 8kbits/s.

A $\pi/4$ differential quadrature phase shift keying (DQPSK) transfers the two channels onto inphase and quadrature components via a constellation map. The rationale behind employing a $pi/4$ phase shift between successive constellations is that the envelope will be non-zero at all times and hence offers advantages with respect to the potential non-linearity of amplifiers utilised in subsequent analogue modulation stages [1]. The differential encoding is implemented using a lookup table, whereby depending on the current symbol value, the phase is rotated by a given amount of $\pi/4 + n \cdot \pi/2, n \in \{0, 1, 2, 3\}$. The inphase and quadrature component are output at the symbol rate of 8kHz.

For a bandlimited transmission, Kingsbury filters [3] were employed. In the original specification, the analogue system part was to be simulated by oversampling the signals by a factor of 16 to 128kHz. At this rate, the timing recovery was to take place in the receiver. As however the DACs on the EVM board are laid out for audio applications, the maximum DAC rate of 48kHz set a limit. To cope with this, the signals were only oversampled by a factor of 4 in the transmitter to 32kHz and output through the DACs. For data acquisition, the sampling rate was again set to 32kHz, but upsampling by a factor of 4 gave the desired oversampling factor of 16 with respect to the symbol rate in order to perform the timing recovery with sufficient resolution at 128kHz.

The split into different sampling rates was performed with two Kingsbury filters $h_1[n]$ and $h_2[n]$ with 20 and 80 coefficients respectively as shown in Fig. 2(a). The upsampling to 128kHz in the receiver in two stages and subsequent decimation is given in Fig. 2(b). The overall circuit in Fig. 2(b) forms a Nyquist system allowing perfect recovery of the transmitted symbols in the absence of quantisation and channel distortions.

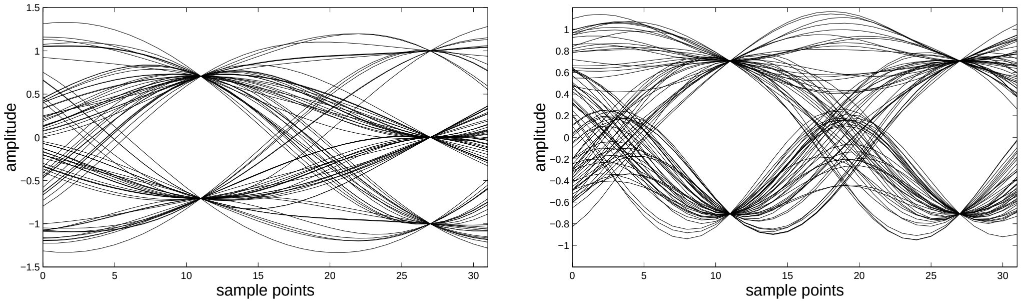

Fig. 3: (left) eye diagram of $\pi/4$-DQPSK modulated inphase without distortion; (right) inphase component after derotation in the receiver.

3. CHANNEL SIMULATION

The output of the transmitter, sampled at 32kHz, are passed to the stereo DAC on the EVM board. The resulting analogue signals for inphase and quadrature component can be modulated up to IF and RF by analogue lab hardware, and modulated back down again to baseband. The baseband signal can finally be sampled at 32kHz in the stereo ADC and fed to the receiver circuitry implemented on the EVM board.

Alternatively, the inputs to the DAC can be connected to the outputs of the ADC by software, hence implementing a complete simulation environment on the board. In the implementation pursued in this project, white Gaussian noise can be added to this channel. The Gaussian noise is derived from two uniformly distributed random variables (created by LFSRs) through the Box-Müller algorithm [4].

4. RECEIVER

The receiver circuit as shown in Fig. 1 consists of the matched Kingsbury filter $h_2[n]$ operated at the upsampled rate of 128kHz, followed by the $\pi/4$-DQPSK demodulation, and the timing recovery. The optimal filtering was already discussed in Sec. 2, and results in the eye diagram given in Fig. 3. It clearly shows the $\pi/4$ rotation between the symbol spaced eye points with 2 or 3 distinct levels.

The $\pi/4$-demodulation performs a rotation of the data by $\pi/4$ over a symbol period, such that the resulting derotated inphase and quadrature signals exhibit the characteristics of the eye diagram shown in Fig. 3(right). This derotated signal can now be employed for timing recovery by searching for the symbol spaced crossing points. The timing recovery employed here is based on subtracting the level of the eyepoint from the modulus of the inphase or quadrature signal. From the current symbol sampling point, the variance in the two adjacent oversampled bins is measured over 512 symbol periods. Then the bin with the lowest variance is selected as the new symbol sampling point. This procedure works for both intial synchronisation and potential clock mismatch between transmitter and receiver. Thus, the symbols are extracted from the oversampled I and Q components by decimation by a factor of 16.

Finally, the data is decoded via a QPSK constellation map and multiplexed onto a single bit stream. Detection and counting of errors in the bit stream with respect to the transmitted sequence will be discussed in the following.

5. ERROR COUNT

The error counting method is closely connected to the PRN generation in the transmitter, and is performed in two stages depicted in the flow graph in Fig. 4. In a synchronisation stage, synchronisation with the pseudo-random number generator in the transmitter is established. Synchronisation is achieved by feeding the received sequence into a LFSR with a generator polynomial identical to the transmitter. In the second stage, the synchronised PRN generator is switched to feedback mode, and the the produced PRN sequence is compared to the received sequence. Any mismatch in the two bit streams is counted as a bit error.

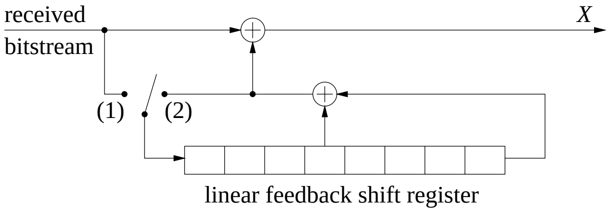

Fig. 4: Error counter in the receiver: in switch position (1), $X = 0$ indicates synchronisation; in switch position (2), $\sum X$ gives number of errors occured (after synchronisation).

6. PROJECT ORGANISATION AND IMPLEMENTATION

The project was performed by three students, amongst whom the responsibilities for different parts of the transmitter and receiver implementation were shared. Additionally, one student was assigned responsibility for the overall software integration. The main software loop was dictated by the frame synchronisation provided by the communication between the DSP processor and the DAC/ADC codec chip [2]. In this loop, the other functions were then integrated.

The attention of the students was particularly drawn towards the computational budgeting, and the fixed point arithmetic of the processor used. For the budgeting, the computationally most intensive task was identified to be the pulse shaping. To exploit redundancies, the multiplication with expanding zeros introduced by upsampling by a factor 4 were avoided by polyphase implementations of the filters $h_1[n]$ and $h_2[n]$ [6]. This polyphase implementation introduced computational savings by a factor of 4 comapred to the direct flow graph implementation of Fig. 1. With regards to fixed point arithmetic, the filter coefficients and signal were scaled such the likelihood for an overflow would be minimum as long as the channel noise added according to Sec. 3 was within the minimum specified SNR limit of 0 dB.

7. CONCLUSIONS

We have reported on the implementation of a complete software modem with separate transmitter and receiver units involving modulation and pulse shaping. The data to be transmitted was simulated by a PRN generator. Distortion was introduced in the form of channel noise. An error counter in the receiver was described that allowed to account for any bit errors resulting from this channel distortion. Additionally, the channel and any further IF or RF modulations could be added into an external loop via the stereo DAC and ADC on the EVM board.

7. REFERENCES

[1] A. H. Aghvami. "Digital Modulation Techniques for Mobile and Personal Communication Systems". *Electronics & Communication Engineering Journal*, Vol.5(No.3):pp.125–132, June 1993.

[2] S. Dirksen. An audio example using dsp/bios. Texas Instruments Application Report SPRA598, November 1999.

[3] N. G. Kingsbury. "Transmit and Receive Filter for QPSK Signals to Optimise the Performance on Linear and Hard-Limited Channels". *IEE Proceedings F*, Vol.133(No.4):pp.345–355, July 1986.

[4] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 3rd edition, 1991.

[5] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice Hall, Englewood Cliffs, 1988.

[6] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, 1993.