



Experimental Study of DQPSK Modulation on SDR Platform

^{1,2}Eko Marpanaji, ²Bambang Riyanto Trilaksono, ²Armein Z.R. Langi,
²Adit Kurniawan, ²Andri Mahendra & ²Thay Liung

¹Departement of Electronics Engineering, Yogyakarta State University
Karangmalang, Yogyakarta 55281, Telp. +62-274-586168, Indonesia

²School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Jl. Ganesa 10 Bandung 40132, Telp. +62-22-2501661, Fax. +62-22-2534133, Indonesia

Abstract. This Paper addresses Differential Quadriphase Shift Keying (DQPSK) modulation implemented on SDR platform for the development of digital data communications based on Software-Defined Radio (SDR). DQPSK modulation performance perceived at Packet Error Rate (PER) is evaluated in terms of Eb/No or S/N ratio, carrier frequency, bit rate, gain, roll-off factor of root Nyquist filter or root raised cosine filter, and payload size from delivered data. Based on these results, the smallest PER could be obtained by setting Eb/No value greater than 20 dB, carrier frequency of at least 0,3 MHz, optimum bit rate of 200 kbps, optimum range payload size of 2000 up to 4000 bytes, and roll-off factor of Nyquist or root-raised cosine filter of 0.4.

Keywords: *differential quadriphase shift keying; packet error rate; root raised cosine; software-defined radio; usrp.*

1 Introduction

This paper describes the implementation of *Differential Quadriphase Shift Keying* (DQPSK) modulation scheme for digital data communication on *Software-Defined Radio* (SDR) platform. Chen (2007) reported the performance of *Quadriphase Shift Keying* (QPSK) modulation scheme for sending digital video data [1]. In this paper, we investigated the performance of DQPSK modulation scheme implemented on SDR platform using *Universal Software Radio Peripheral* (USRP) from GNU Radio. DQPSK modulation scheme is chosen because it is widely used in wireless and cellular communication systems such as Satellite, CDMA, and cable modem.

The rest of this paper is organized as follows. Section 2 discusses SDR architecture along with SDR's characteristics. Section 3 presents the DQPSK modulation scheme. Section 4 presents system configuration used in this research. Section 5 presents DQPSK implementation and test result, followed by Conclusion in Section 6.

2 Software-Defined Radio (SDR)

Software-defined radio (SDR), or software radio (SR) for short, was first introduced in 1991 by Joseph Mitola [2]. The word of SDR was used to show a radio class that could be re-configured or re-programmed [3] so that the mode and the frequency of a wireless communication application can be set by the software functionality. Ideally, SDR offers flexibility, re-configurability, scalability and as multi mode as possible.

SDR architecture is developed based on conventional radio functions. Unlike in the conventional radio in that all functions of signal processing are carried out fully by the hardware, the signal processing on SDR is carried out as much as possible by the software. The major key in building SDR is the placement of ADC and DAC components as a divider between analog and digital domains, allowing the signal processing functions can be carried out using software.

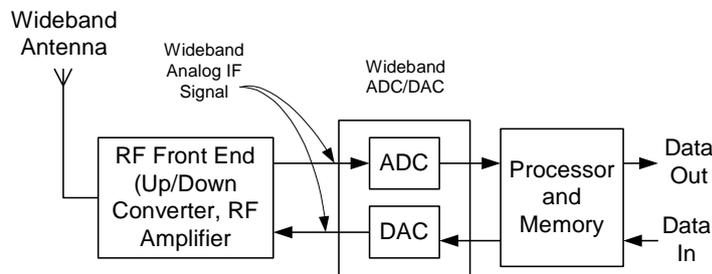


Figure 1 Realistic SDR Architecture

The ideal SDR architecture will place ADC/DAC as close as possible with the antenna, and also close to processor for performing digital signal processing and radio functions. Therefore, if the communication standard changes, it can be anticipated by updating the software [2,3]. However, ADC/DAC wideband is expensive and due to the technological limitation, the more realistic SDR architecture places ADC/DAC wideband after Down Converter/Up Converter (see Figure 1). With this architecture, the conversion from analog to digital and vice versa is carried out on Intermediate Frequency (IF) signal that has lower frequency than RF signal. Today, this kind of SDR architecture is being developed widely. Figure 2 shows SDR architecture for both transmitter and receiver.

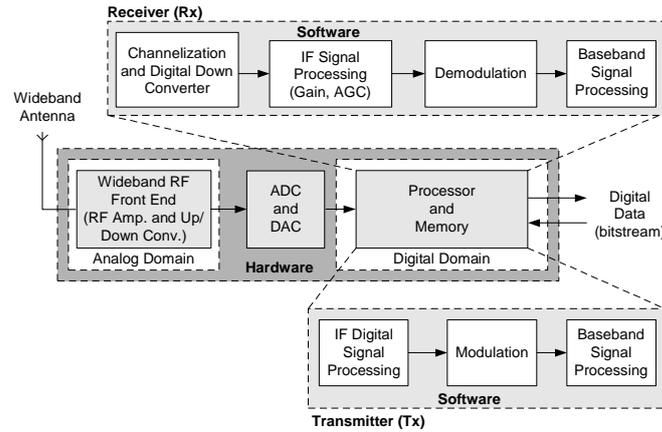


Figure 2 SDR Architecture for Transmitter and Receiver

The SDR platform performs transmitting and receiving functions. The transmitter (Tx) performs base band signal processing, modulation, digital IF signal processing, and sending RF signal to the air. The receiver (Rx) performs RF signal processing, channelization, digital IF signal processing, demodulation, and base band signal processing. As shown in Figure 2, the computation in the receiver is more complex than in the transmitter.

ADC and DAC components will determine a point in which radio functions can be performed through a software. This point is usually called digital access point. The realistic SDR architecture will use a software to process digital IF signal. The major challenge in choosing ADC's location is the problems of limited sampling frequency and economics factor. If the sampling frequency of Nyquist is hard to be done because the digitalization of IF signal is heavy, then the alternative is to apply under sampling toward IF signal so that it reduces the computation complexity in the processor. The formula to determine under sampling frequency is shown below:

$$\frac{2f_c + B}{m+1} \leq f_s \leq \frac{2f_c - B}{m}; m \in N \quad (1)$$

Where f_c is the center of frequency, B is bandwidth, f_s is sampling frequency chosen until $f_s > 2B$ [4]. For example, the SDR system uses 20 MHz IF signal with 5 MHz bandwidth can be sampled with frequency of 22,5 MHz for $m=1$, or 17,5 MHz for $m=2$, or 11,66 MHz for $m=3$.

The next attention is on the computation complexity, which will determine the type and the size of processor needed. The processor performs digital signal

processing for both the receiver and transmitter. Besides identifying the computational complexity of the software, the scalability factor for future development is also observed. All those steps will determine the processor's size usually stated with *Million Operations per Second* (MOPS) or *Million Instructions per Second* (MIPS).

The largest computational requirement is on the receiver side especially on filter part in channelization, which ranges from 100 to 200 operations/sample [3,5,6]. Applying the sampling speed of 22.5 MHz will result in 22.5 MSPS, so that it needs 2,250 - 4,500 MIPS. Other computational-intensive processes are modulation and demodulation. As an illustration, modulation and demodulation FM require more or less 100 to 150 MIPS [7]. If a single processor cannot handle the needed computational requirement, then the SDR architecture can use parallel computation architecture by involving several processors or implementing a part of signal processing function in form of FPGA.

The following sections describes the SDR implementation for digital data communication using Differential Quadriphase Shift Keying (DQPSK) modulation scheme along with the test result to observe its PER.

3 Differential Quadriphase Shift Keying (DQPSK) Modulation

DQPSK modulation is a variant of $\pi/4$ -QPSK modulation scheme whose differential encoding is added to the bits encoding prior to the modulation. Differential means that the information is not carried by the absolute state; it is carried by the transition between states. The advantage of using differential encoding is free from phase ambiguity if the constellation is rotated by an effect in the communications channel in which the signal passes through. This problem can be overcome by using the data to change rather than set the phase. The QPSK modulation scheme using differential encoding is usually called as $\pi/4$ -DQPSK or DQPSK.

The equation differential encoder to produce symbols is:

$$s_k = s_{k-1} \oplus b_k \quad (2)$$

where s_k is current output symbol, s_{k-1} is previous output symbol, and b_k is current bit input. The \oplus sign indicates modulo-2 operation. So s_k only changes state (from binary '0' to binary '1' or from binary '1' to binary '0') if b_k is a binary '1'. Otherwise it remains in its previous state. At the receiver, the received signal is demodulated to yield $s_k = \pm 1$ and then differential decoder reverses the encoding procedure (decoding process) and produces:

$$b' = s_k \oplus s_{k-1} \quad (3)$$

since binary subtraction is the same as binary addition. Therefore, $b'_k = 1$ if s_k and s_{k-1} differ and $b'_k = 0$ if they are the same. Hence, if both s_k and s_{k-1} are *inverted*, b_k will still be decoded correctly as b'_k . Thus, the 180° phase ambiguity does not matter.

The DQPSK modulation format uses two QPSK constellations offset by 45 degrees ($\pi/4$ radians) as depicted in Figure 3. Transitions must occur from one constellation to the other, or each DQPSK symbol will reside in one of eight points in the constellation diagram. This guarantees that there is always a change in phase at each symbol, making clock recovery easier. The data is encoded in the magnitude and direction of the phase shift, not in the absolute position on the constellation.

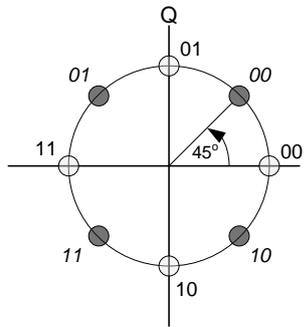


Figure 3 DQPSK constellation diagram with identical Gray coding rotated by 45°

Table 1 summarizes a possible set of relationships between the phase transitions in the DQPSK modulation scheme and the incoming Gray code dibits [8].

Table 1 Correspondence between input dibit and phase change for DQPSK modulation.

<i>Gray-Encoded Input Dibit</i>	<i>Phase change, $\Delta\theta$ (radians)</i>
00	$\pi/4$
01	$3\pi/4$
11	$-3\pi/4$
10	$-\pi/4$

DQPSK employs root Nyquist filter or root raised cosine filter as a pulse-shaping filter. This filter is used to reduce a number of spurious signals or to control the shape of digital data. The impulse response of root Nyquist filter is determined by equation [3]:

$$g(t) = \frac{1}{\pi t} \frac{1}{1 - \left(\frac{4\alpha t}{T_b}\right)^2} \sin\left\{2\pi t \left[1 - \alpha \frac{-1}{T_b}\right]\right\} + \frac{1}{\pi} \frac{4\alpha / T_b}{1 - \left(\frac{4\alpha t}{T_b}\right)^2} \cos\left\{2\pi t \left[1 + \alpha \frac{-1}{T_b}\right]\right\} \quad (4)$$

where α is roll-off factor of root raised cosine filter. This parameter controls the shape and bandwidth of the incoming signal.

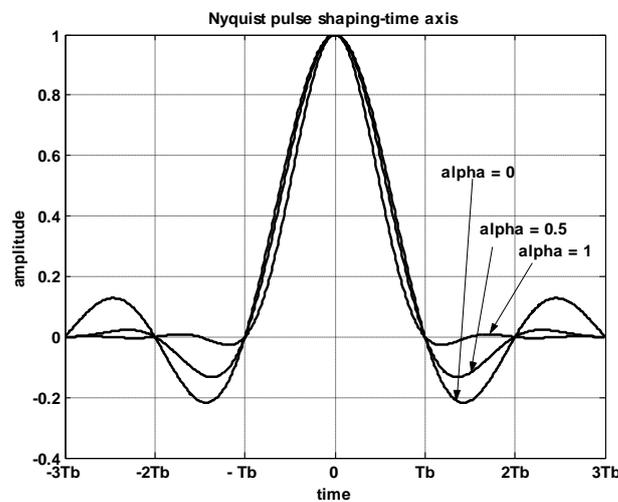


Figure 4 Impulse response of a Nyquist filter in time domain with several chosen roll-off factors.

Figure 4 shows the impulse response of a root Nyquist filter over time on different roll-off factor (α) values. One of the characteristics of the Nyquist filter is that we always obtain zero impulse response at nT_b (n is integer: 1, 2, 3, ...). Therefore, when we set the synchronization point at nT_b , one symbol never interferes with other symbols at this point.

The advantages of DQPSK are:

- the phase transitions from one symbol to the next are restricted to $\pm\pi/4$ and $\pm3\pi/4$ so it will reduce the symbol ambiguity.

- using root raised cosine filtering, it has better spectral efficiency than *Gaussian Minimum Shift Keying* (GMSK), the other common cellular modulation type.

3.1 DQPSK Modulator

Figure 5 shows the process behind a DQPSK modulator. It starts by changing binary bit stream 0 and 1 into bipolar bit stream -1 and 1. Next, a mapping circuit processes the bipolar bit stream into dibits for *I-channel* (Ich) and *Q-channel* (Qch). Differential encoders separately converts each dibit into its corresponding phase change for DQPSK modulation using the rules depicted in Table 1. A pulse-shaping filter is then applied to reduce spurious signals produced by the output of differential encoder. Finally, the outputs of pulse-shaping filter are converted into analog signal and are modulated using RF signals $\cos(2\pi f_c t)$ for Ich signal and $\sin(2\pi f_c t)$ for Qch signal, where f_c is carrier frequency. Both Ich and Qch signals are combined together to produce DQPSK signal.

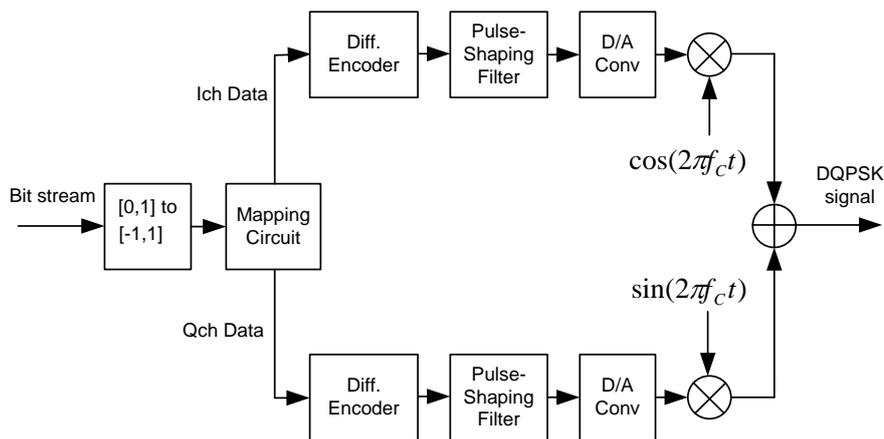


Figure 5 DQPSK Modulator

3.2 DQPSK Demodulator

At the receiver or demodulator of DQPSK, process is reversed. Figure 6 shows the block diagram of DQPSK demodulator.

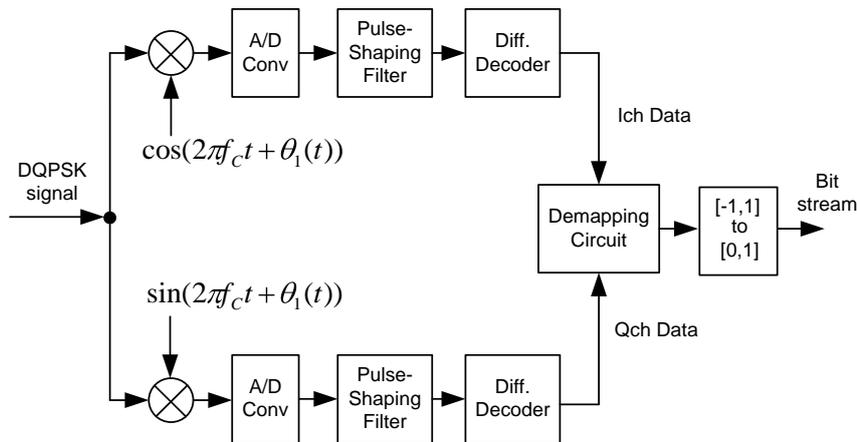


Figure 6 DQPSK Demodulator

The demodulator of DQPSK first eliminates the carrier frequency for recovering the Ich and Qch signals. A/D Converters transform these signals into digital formats before passing them through a pulse-shaping filter and then differential decoder. The latter generates the dibit information. The Demapping Circuit will then dissociate the dibit information, changing bit stream -1 and 1 into bit stream 0 and 1, recovering the data transmitted by DQPSK modulator.

4 System Configuration

The system's configuration employed in this research is shown in Figure 7. The front end of SDR platform uses USRP GNU Radio peripheral as an up/down converter. The platform employs a personal computer (PC) as the processor. SDR system configuration that uses PC as the processor is often referred to as Software Radio (SWR). In this configuration, hence, the modulator and demodulator of DQPSK can be implemented using software. The A/D and D/A converters remain implemented using hardware, which are located in USRP. The PC connects to USRP via USB 2.0 port.

The mainboard specifications of USRP used in this research are as follows: (a) USB 2.0 port for connection to computer; (b) 12-bit ADC with sampling speed of 64 MSPS, allowing it to conduct digitalization process with range frequency of aliasing - 32 MHz until 32 MHz; (c) 14-bit DAC with clock frequency 128 MSPS so that Nyquist frequency equals to 64 MHz; and (d) analog signal whose yield is limited to 10 mWatt. The daughterboard uses Basic Tx and Rx so that there is no process of up/down converter and the transmitter frequency is limited to 50 MHz maximum.

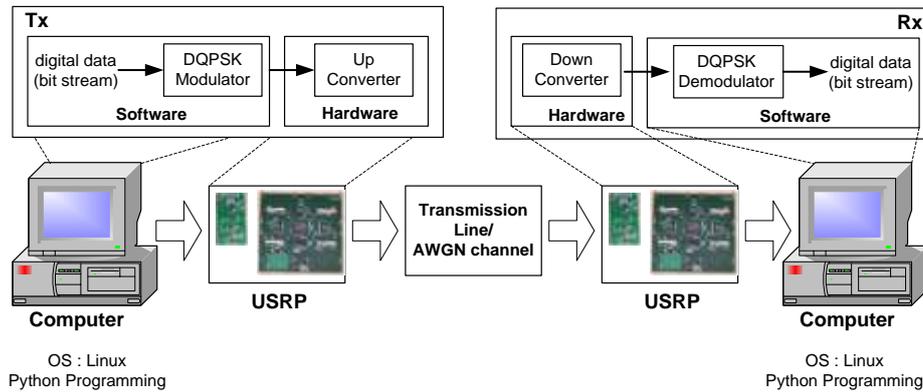


Figure 7 System configuration of SDR platform

The PC specification is summarized in Table 2. The computer must have USB version 2.0 port to support the connection to USRP board. The programming language uses Python and a few functions are written in C++.

Table 2 Computer specifications that used in this research

No	Component	1 st Computer	2 nd Computer
1.	Processor	AMD Athlon XP1800+, 1.53 GHz, fsb 533 MHz	Intel Pentium 4, 2.93 Ghz, fsb 533 MHz
2.	RAM	DDR 333 MHz, 256 Mbyte	DDR 400 MHz, 2 x 256 MByte
3.	Operating System	Linux Fedora Core-4, 2.6.11-1.i369	Linux Fedora Core-4, 2.6.11-1.i369

The performance of system can be measured using one of the parameters often employed in QoS, that is:

$$\text{BER} < 10^{-3}, \text{PLR or PER} < 10^{-2}, \text{spread delay} < 100 \text{ ms, and Gos} > 95\% [4].$$

In this research, the performance of DQPSK modulation scheme is measured in Packet Error Rate (PER) because data are transmitted in the form of packet. Transmitting data as a packet will facilitate the SDR software integration with TCP/IP protocol for future development.

5 Implementation and Test Result

DQPSK modulator and demodulator are implemented in SDR platform using PC and USRP board that are commonly used in SDR experiment from GNU Radio. The Board is a front side of SDR system for performing digitalization function (ADC-DAC) and for processing filter channelization. The digital signal processing was performed by personal computer with USB port as a link between USRP and PC. The Operating system is Linux with python programming language for developing the application. All softwares used for digital signal processing was open source softwares from GNU Radio.

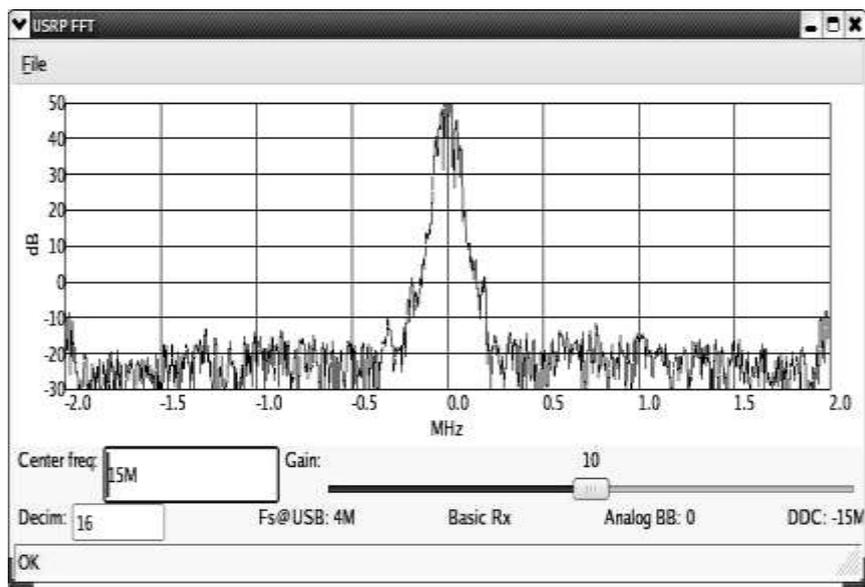


Figure 8 Spectrum DQPSK signal

The experiments observe the performance of digital communication based on SDR system using USRP board with DQPSK modulation scheme. The transmission channel of coaxial cable RG58 was used to connect transmitter and receiver because of the limitation of Basic TX output power. The experiments were carried out to observe Packet Error Rate (PER) stated by the percentage of exactly accepted packet toward variation of E_b/N_0 , bit rate, carrier frequency, roll-off factor, gain, and also payload size.

Figure 8 shows the spectrum of DQPSK signal, which has 15 MHz carrier frequency, bit rate 256 kbps, and roll-off factor 0.35. The results of observations are shown in Figures 9 - 14.

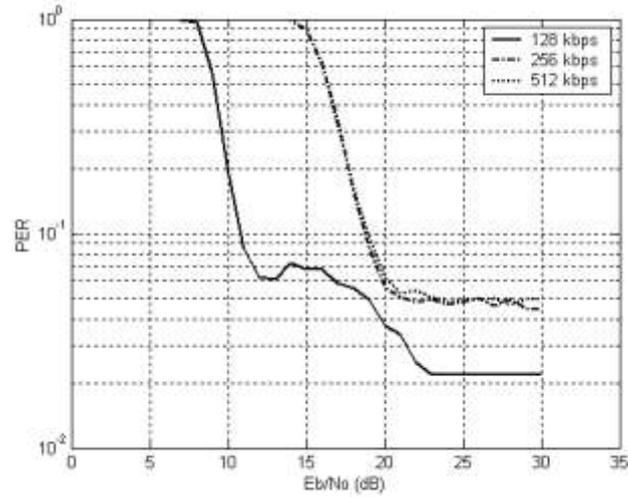


Figure 9 The effect of Eb/No on PER

The figure shows the PER values with 128 kbps, 256 kbps, and 512 kbps bitrate over Eb/No. The greater the value of Eb/No is the smaller the PER obtained. PER values remain low for Eb/No higher than 20 dB.

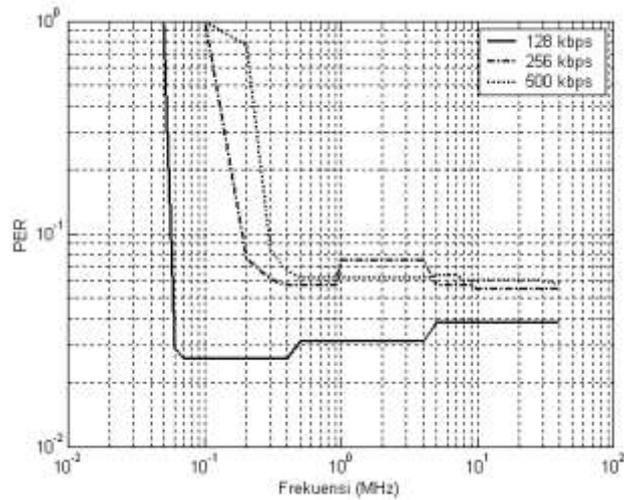


Figure 10 The effect of carrier frequency on PER

Figure 10 depicts the effects of variation of frequency carrier modulations on the PER values. The figure shows these effects for data bit-rate transmissions of 128 kbps, 256 kbps, and 500 kbps. In these experiments, the maximum frequency that has been tried is 44 MHz, and the USRP using existing Basic TX

cannot handle beyond that frequency. As shown in Figure 10, the lowest PER value can be obtained if $f_c > 0.06$ MHz for 128 kbps, $f_c > 0.2$ MHz for 256 kbps, and $f_c > 0.3$ MHz for 500 kbps. In another word, the minimum frequency than can still support the data transmission with bit-rate greater than 256 kbps is 0.3 MHz.

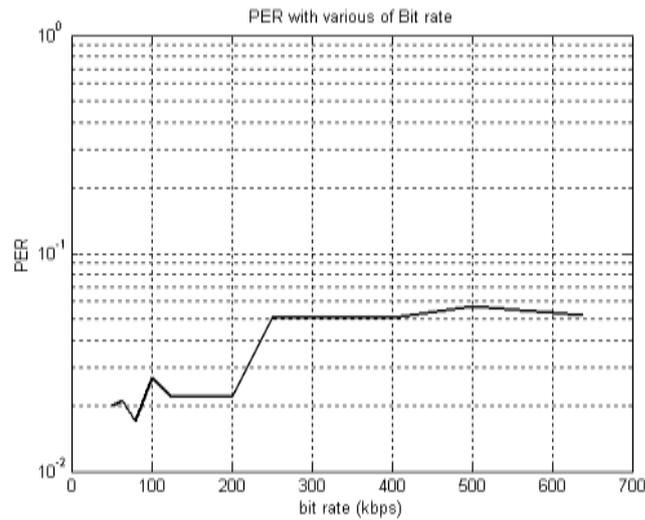


Figure 11 The effect of bit rate on PER

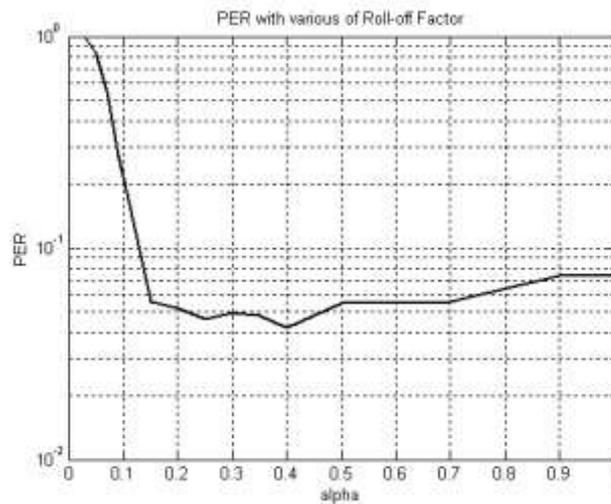


Figure 12 The effect of roll-off factor on PER

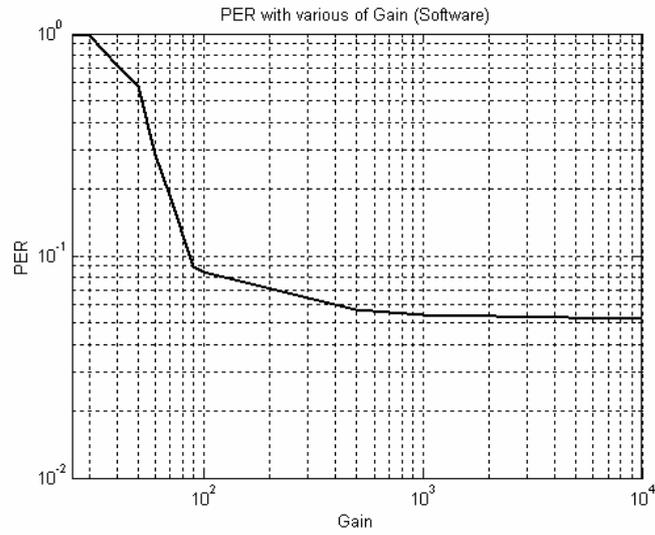


Figure 13 The effect of gain PER

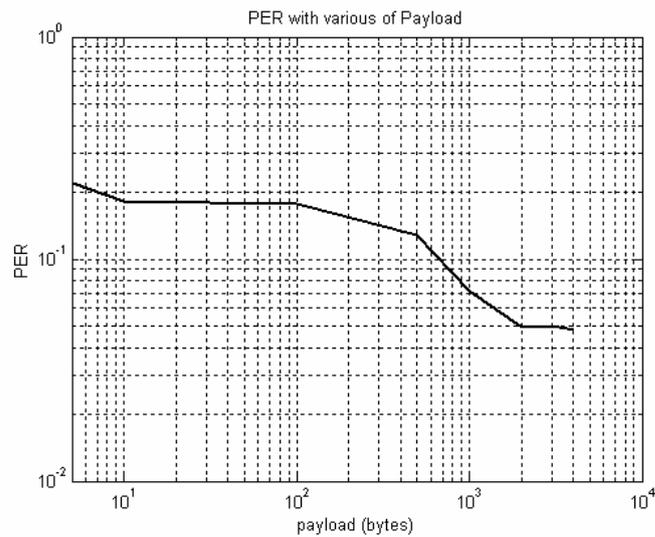


Figure 14 The effect of payload size on PER

Figure 11 shows PER values over bit-rate. The optimum bit-rate for the modulation of this DQPSK is 200 kbps. Bit rates greater than 200 kbps will increase the PER value and bit rates from 250 up to 600 kbps produce higher PER values. Figure 12 shows PER values over roll-off factor at Nyquist filter.

The bigger roll-off factor values will result in smaller PER values, and the optimum value for roll-off factor is 0.4.

Figure 13 depicts PER values over the gain values. As shown in the figure, the greater gain will result in the smaller PER value. However, for the gain values bigger than 500, the PER values do not significantly change. Hence, the optimum value for gain is 500.

Figure 14 shows PER values over payload size. The figure indicates that the bigger transmitted payload value will produce the smaller PER. The maximum data size for this system is 4092 bytes, hence the optimum value for the payload is 4092 bytes.

Based on these experiment results, the PER values obtained by varying Eb/No, carrier frequency, bit rate, roll-off factor, gain, and payload are close to, but are still above 10^{-2} , which have not satisfied the QoS standard. Therefore, there are still rooms for further improvement in future work.

6 Conclusion

In conclusion, the modulation scheme of DQPSK implemented at SDR platform using USRP from GNU Radio has better performance based on PER value. The lowest PER value can be obtained at Eb/No value > 20 dB, optimum bit-rate of 200 kbps, carrier frequency bigger than 0,3 MHz, roll-off factor (alpha) of 0.4, optimum gain of 500, and optimum payload size of 4092 byte.

Future research will work on the SDR software development for improving the system's performance and work on the application development that integrates the communications protocol such as TCP/IP.

Acknowledgment

We would like to thank to the Dean of STEI ITB and the chief of PPTIK ITB who have given opportunity also donation in order to accomplish this research. This research is funded by ITB Research fund year 2007 Contract#: 0047/K01.03.2/PL2.1.5/I/2007. We also thank to all people who contribute to this research.

References

- [1] Chen, Z., *Performance Analysis of Channel Estimation and Adaptive Equalization in Slow Fading Channel*, http://www.wu.ece.ufl.edu/projects/channelEstimation/full_files.zip, (2007).

- [2] Reed, J.H., *Software Radio: A Modern Approach to Radio Engineering*, New Jersey: Prentice Hall, pp. 2-3, 2002.
- [3] J. Mitola III, *Software Radio Architecture, Object-Oriented Approaches to Wireless Systems Engineering*, Canada: John Wiley & Sons, Inc, pp. 1-34, 2000.
- [4] Angoletta, M.E., *From Analog to Digital Domain*, http://humanresources.web.cern.ch/humanre-sources/external/training/special/ DISP2003/DISP-2003_L01A_20Feb03.pdf, (20 Feb 2003).
- [5] Salkintzis, A.K., Nie, H., Mathiooulos, P.T., *ADC and DSP Challenges in the Development of Software Radio Base Stations*, IEEE Personal Communications, (1999, August).
- [6] Dick, C.H., *Design and Implementation of High-Performance FPGA Signal Processing Datapath for Software Define Radios*, [http:// www.eetasia.com/ARTICLES/2001AUG/2001AUG09_ICD_AMD_RFD_TAC01.PDF](http://www.eetasia.com/ARTICLES/2001AUG/2001AUG09_ICD_AMD_RFD_TAC01.PDF), (2001).
- [7] Huie, J., D'Antonio, P., Pelt, R., *Synthesizing FPGA Cores for Software-Defined Radio*, <http://www.altera.com/cp/fpga-cores-for-sdr.pdf>, (2005).
- [8] Haykin, S., *Communication System*, 4th ed. New York: John Wiley & Sons, Inc, pp. 362-365, 2004.