Utilization of Software-Defined Radio in Power Line Communication between Motor and Frequency Converter

A. Pinomaa, H. Baumgartner, J. Ahola, and A. Kosonen
Department of Electrical Engineering, Institute of Energy Technology
Lappeenranta University of Technology (LUT)
Lappeenranta, Finland
antti.pinomaa@lut.fi

Abstract—In this paper, a power line data transmission link based on Software-Defined Radio (GNU Radio) for motor cable communication between an electrical motor and a frequency converter is developed and tested. The test environment includes a frequency converter, an electrical motor (2.2 kW), and a 90-meter-long motor power cable. Two differential phase shift keying (DPSK) modulations, DBPSK and DQPSK, are used and their performance is compared with and without forward error correction (FEC). Experiments prove that GNU Radio with USRP (Universal Software Radio Peripheral) is a competent platform to be used as a testing and developing power line communication applications.

Keywords-bit error ratio; convolutional coding; GNU Radio; inverter; power line communication (PLC); software-defined radio

I. INTRODUCTION

It has been shown that the feeder cable between a motor and an inverter can be used as a communication medium in variable-speed electric drives. It is a feasible alternative to sensor cabling in condition monitoring and/or the control of electrical machines in an industrial environment [1], [2]. The utilization of a motor cable in communication eliminates the substantial costs involved in cable installation and maintenance, and simplifies the configuration of the electrical drive. In addition, traditional separate communication cables can be susceptible to physical damage and external interferences.

Software-Defined Radio (SDR) can be considered to be a radio communication system, where some of its functional components, such as filters, mixers, modulations, etc., are implemented with software components. This makes it possible to configure the radio according to the requirements of the application and the characteristics of the communication channel. During the past few years, software-defined radios and their successors cognitive radios (CR) have been intensively investigated [3]–[5]. Another project carried out with a GNU Radio is presented in [6]. In this paper, an implementation of SDR (GNU Radio) is used to implement a power line data transmission link between a motor and an inverter. The designed link operates at the center frequency of 13.56 MHz and uses DPSK modulations in data transmission. The performance of the link is studied. During the tests, the operation parameters of the link are adjusted, such as modulation, bit rate, transmit power, and forward error correction.

The structure of this paper is organized as follows. In Section II, the communication channel and its characteristics are studied. Section III concentrates on GNU Radio and implementation of the data transmission link. The laboratory tests and analyses are carried out in Section IV. Finally, Section V concludes the paper.

II. COMMUNICATION CHANNEL

A. Description of the Test Environment

The test environment was built in the Laboratory of Power Electronics at Lappeenranta University of Technology. The test setup comprised a Strömberg 2.2 kW induction motor (3-phase, 4-pole), a 90-meter-long Pirelli MCCMK 3x35+16 motor cable, an ABB ACS400 frequency converter, two PCs, two USRPs (Universal Software Radio Peripherals), and two coupling interfaces. The test setup is illustrated in Fig. 1. The inductive coupling interfaces, based on ferrite rings, connect the USRPs differentially to the motor power cable as illustrated in Fig. 2.

An additional high pass filter in the coupling interface

Fig. 1. Data transmission concept between an electric motor and a frequency converter. The signal used for communication is coupled differentially between two phase conductors of the motor cable with inductive signal couplers [7].
consists of six small signal diodes and a passive LC filter. The diodes $D_1$–$D_6$ (Fig. 2) act as a transient suppressor to protect the USRPs against voltage spikes induced by the switching of the inverter output stage that includes for example insulated gate bipolar transistors (IGBTs).

**B. Channel Characteristics**

The communication channel consists of an electric motor, an inverter, a motor cable, and two inductive coupling interfaces. The simulated channel gain of the communication channel, from the motor to the inverter, as a function of frequency is illustrated in Fig. 3. As it can be seen, the attenuation of the communication channel is as lowest in the frequency band of 7.5–10 MHz, and it increases as a function of frequency after 10 MHz. The ferrite rings and the LC high-pass filter presented in Fig. 2 strongly attenuate frequencies below 7 MHz.

Measured noise power spectral density (PSD) at the terminals of the coupling interface at the inverter end is illustrated in Fig. 4. The switching frequency of the inverter was adjusted to $f_{sw} = 4$ kHz. According to [8], the inverter is mainly a source of asynchronous impulsive noise.

The majority noise PSD (Fig. 4) is concentrated on the frequencies lower than 10 MHz. At higher frequencies PSD decreases steeply, and becomes almost flat when $f > 15$ MHz.

From Figs. 3 and 4, it can be seen that the optimal carrier frequency for communication would be between 10–15 MHz, where the channel gain of the communication channel is higher than -30 dB and the noise PSD is almost flat. Thus, the license-free ISM (Industrial, Scientific, and Medical) radio frequency band of 13.56 MHz center frequency, which is also a well-known RFID frequency sub-band, is chosen to be used as a carrier frequency. With a software-defined radio, with a D/A-converter based front-end, the carrier frequency can be changed relatively easily, but in the tests presented in this paper it is kept constant. According to Fig. 3, the channel gain at 13.56 MHz is about -24 dB.

**III. IMPLEMENTATION OF SDR FOR MOTOR CABLE COMMUNICATION**

A software-defined radio is a system, where components such as modulation, amplification, and filtering are carried out by software instead of electrical components. The main benefit of using a software-defined radio is the versatility of the system compared with pure hardware-based solutions. The same radio device can be used for different applications only by modifying the software. For example, the same platform can be used for the reception of analogue FM radio broadcasts and digitally modulated and coded information by changing software.

**A. GNU Radio**

GNU Radio is an open source signal processing package for examining and developing software-defined radios. It is a multiplatform package that can be used both in Unix- and Windows-based systems. GNU Radio applications are written in Python scripting language, and the application consists of “flow graphs” and “blocks”. Blocks are written in C++ to handle the processor intensive signal processing tasks. By
using SWIG, C++ blocks can be used in Python applications, which make it straightforward to deploy the test software.

A flow graph is a Python application that connects the blocks written in C++ together. It is also possible to make additional signal processing in Python. However, processor intensive tasks in Python are not recommended because of the poor performance of the interpreted code. Thus, Python only connects the C++ blocks; they do not need to be interpreted and can be run at native speed.

In the case of a simple application, it is not necessary to write C++ blocks, because GNU Radio includes a collection of ready-to-use blocks. A simple transmitting flow graph may consist, for example, of a file source, a packet modulator, amplification, and USRP blocks. The desired Python code to connect these GNU Radio blocks together is roughly 50 lines of code.

B. USRP

Because GNU Radio is only a software package, it is necessary to have an interface to the real world. The USRP is a hardware device developed by Ettus Research LLC and designed especially to be used with a GNU Radio.

The USRP consists of fast A/D and D/A converters, a USB (Universal Serial Bus) 2.0 controller, FPGA (Field-Programmable Gate Array), and RF frontends; LFTX30M and LFRX30M (0-30 MHz). The sampling frequency for a 12-bit A/D converter is 64 MHz and for a 14-bit D/A converter 128 MHz. Using a high-performance USB host controller, the USB can sustain 32 MBps payload data transfer speed. The USB is half-duplex, and thus the 32 MBps transfer speed requires to be partitioned between the transmission and the receiving directions [9]. FPGA takes care of decimating and interpolating of high bit rate signals in order to be transferred over the relatively slow USB link.

The samples from/to the USRP are IQ-modulated and consist of a 16-bit I-part and a 16-bit Q-part. Assuming the bit rate of USB interface to be 32 MBps, the maximum sampling frequency is 8 MS/s.

C. Test bench

The test bench is illustrated in Fig. 1. Data is sent from the motor end along a 90-meter-long power cable to the inverter end. In all tests, the motor has no load, it is running idle. The USRPs are connected to the medium through inductive coupling interfaces.

The objective of the research is to implement a SDR-based PLC system, and to study bit error ratios (BER) as a function of data rates, modulations, and transmission power in the test channel. In the test bench, data packets consisting of random characters with packet sizes of 1500 and 3000 characters were sent from the transmitter to the receiver. Next, the contents of the transmitted and received packets were compared bit by bit. Two modulations; DBPSK (Differential Binary Phase Shift Keying) and DQPSK (Differential Quadrature Phase Shift Keying) were used. Also the effect of forward error correction (FEC) to the BER, based on convolutional coding and interleaving, were tested. In all tests, the center frequency of the carrier wave was adjusted to 13.56 MHz.

IV. TESTS AND ANALYSIS

A. Bit Rates

The modulation techniques applied were DBPSK and DQPSK. By default, the GNU Radio defines the minimum values for bit rates. The bit rate is calculated by three parameters; DA/AD converter rate, interpolation and decimation factors, and the number of samples per symbol. The minimum value of the bit rate of the USRP for DBPSK modulation is 35.714 kbps and 71.428 kbps for DQPSK modulation.

Different bit rates were tested from the minimum to the rate of 500 kbps. According to the tests, the number of error bits increased exponentially when the data transmission speed was increased. With DBPSK modulation at the maximum gain, the BER at 35.714 kbps transmission rate was 1.56·10⁻⁵. Correspondingly, at the transmission rate of 500 kbps no data frames were received.

B. Transmission Power

The value of the amplitude gain parameter of the signal to be sent from the transmitting USRP to the communication channel varies between 0 and 32767. It is possible to use amplitude values up to 20000 for DBPSK and 30000 for DQPSK without data corruption. The effect of the amplitude gain parameter to the transmission power on a 50 Ω load is presented in Fig. 5.

The transmission power fed to the communication channel was measured from the sending USRP with an oscilloscope Tektronix TDS 3012 over 50 Ω termination impedance. Welch’s power spectrum estimate of the measured DBPSK modulated signal is presented in Fig. 6. It shows that the
transmission power is around -50 dBm/Hz with the amplitude setting of 20000.

C. Received Power and SNR

According to Fig. 4, the noise PSD at the carrier frequency of 13.56 MHz is -90 dBm/Hz. The total noise power at the receiver at the 100 kHz transmission band around the 13.56 MHz center frequency is obtained to be -40 dBm (Fig. 4). The transmission power can be estimated from the transmission spectrum illustrated in Fig. 6. The transmission power at the 100 kHz transmission band of the carrier frequency of 13.56 MHz for the DBPSK-modulated signal with a maximum working gain value (20000) is approximately -2.1 dBm. According to Fig. 3, the gain of the communication channel is -24 dB at the carrier frequency of 13.56 MHz, and thus the received signal power is -26.1 dBm. Hence, the signal to noise ratio at the receiver is 13.9 dB. The transmission power depends on the modulation and the setting of the amplitude parameter which can vary between 0 and 32767. The transmission power as a function of the amplitude parameter of the USRP for both modulations is presented in Fig. 5.

There is also a graphical extension for GNU Radio that can be used to develop test environments rapidly. All the GNU Radio blocks are implemented into the graphical environment GNU Radio Companion (GRC). With the GRC, it is easy to see the signal that USRP sends to the computer. By connecting the receiving USRP to a FFT sink block in the GRC, the baseband signal can be graphically analyzed. The FFT plot of the received signal is presented in Fig. 7. It visualizes the influence of switching the inverter on to the SNR.

D. Coding

Convolutional coding is a FEC method, which can correct error bits that are sequential. Coding increases the amount of transmitted bits. The convolutional coding rate of ½ used in this study doubles the amount of the transmitted data. In the test environment, corrupted bits appear as bursts of multiple errors. These are caused by the impulsive noise generated by the switching of the inverter output stage. The locations of error bits in a single packet in this application are presented in Fig. 8. The figure is from a single measurement where DBPSK-modulated uncoded data and 35.714 kbps bit rate were used. With this bit rate, the duration of one packet in the channel is approximately 340 ms.

Fig. 8 shows that corrupted bits appear as bursts of multiple bits (2-6 sequential error bits). Due to bursts, the convolution coding rate of ½ is not very efficient to correct these kinds of errors. The data should be interleaved so that the error bursts are not sequential. The convolutional coded data is thus divided into blocks of 32 bits, which are interleaved such as that sequential bits are as far from each other as possible (minimum distance 7). By sending this interleaved data and de-interleaving it at the receiving end, it is possible to correct also the error bursts of multiple bits. Despite of interleaving, there still remain some error bits.

Interleaving was used with convolutional coding in this research. Coding was made with MatLab. The convolutional coding rate was ½, the constraint length was 3 and the used generator vectors were [1 1 0] and [1 1 1]. The block size was 32 bits and the bits are arranged in interleaving according to vector [1 8 15 22 29 4 11 18 25 32 7 14 21 28 3 10 17 24 31 6 13 20 27 2 9 16 23 30 5 12 19 26].

E. BER

Bit error ratio (BER) is the ratio of error bits and the total number of bits sent. Usually, BER is defined as a function of
Eb/No. BER calculation can be made with MatLab by comparing transmitted and received data. Eb/No is normalized Signal to Noise Ratio (SNR). A theoretical value for BER can be calculated. The value depends on the modulation and SNR. According to [10], the theoretical BER for DPSK modulation is:

$$P_b = \frac{1}{\log M} \left[ 1 - \text{erf} \left( \sin \left( \frac{\pi}{M} \sqrt{\log_2 M \left( \frac{E_b}{N_0} \right)^{1/2}} \right) \right) \right],$$

(1)

where $M$ is the number of the symbols in an alphabet and $E_b/N_0$ the normalized SNR. $M = 2$ in DBPSK and $M = 4$ in DQPSK. The BER in (1) is for additive white Gaussian channels and here this is not the case. Hence, (1) is used only as a theoretical reference. The BER plots for uncoded and convolutional coded DBPSK- and DQPSK-modulated data are presented in Figs. 9 and 10. The BER curves in Figs. 9 and 10 show that the theoretical BER curve and the BER measurements of the transmission channel are not identical. In (1), the noise is white Gaussian noise, which is not true in the test environment. The devices and components used in the test are not ideal either, and thus the waveforms cannot be equal. However, the shapes of the curves are similar, and the measurements follow the shape of the theoretical BER curve. From Figs. 9 and 10 it can be seen in both modulations that BER decreases more steeply with convolutional coded data than uncoded one as a function of Eb/No.

The BER curves of convolutional coded data are not directly comparable with the uncoded ones because the effective bit rate is half of the original. It is reasonable to compare the BERs of the coded DQPSK and the uncoded DBPSK, where effective bit rates are equal. The BER vs. Eb/No of the uncoded DBPSK and the convolutional coded and interleaved DQPSK are illustrated in Fig. 11.

By comparing the BER results presented in Fig. 11, it can be seen that when Eb/No is between 8-14 dB, the uncoded DBPSK is just a slightly more efficient than the convolutional coded DQPSK. But between 16-18 dB the BER of coded DQPSK- is decreasing more steeply than the uncoded DBPSK-modulated data. Hence, in this case it is possible to reach the same BER with less power with the uncoded DBPSK- than with the coded DQPSK-modulated data.

Increasing the transmission speed leads to exponential growth of the BER. When increasing the bit rate, the same amount of power available is only divided into a wider bandwidth that leads to a decrease in the SNR. With this implementation, it is not possible to reach higher bit rates. By using an RF amplifier, SNR can be increased at higher transmission rates; the issue, however is outside the scope of this paper.

Fig. 8. Locations of error bits inside one packet (packet size 12000 bits). Modulation DBPSK and the effective bit rate was 35.714 kbps. The x-axis indicates the number of bits, and on the y-axis 0 indicates uncorrupted bit and 1 error bit. Spikes in the figure include 2–6 sequential error bits.

Fig. 9. BER – Eb/No of the DBPSK-modulated signal. The bit rate was 35.714 kbps. The effective bit rate of the convolutional coded and interleaved data is half of the uncoded data.

Fig. 10. BER – Eb/No of the DQPSK-modulated signal. The bit rate was 71.428 kbps. The effective bit rate of the convolutional coded and interleaved data is half of the uncoded data.
F. Applications

As shown above, the GNU Radio and USRPs can be used for developing and testing power line data transmission links. In this case a narrowband data transmission link operating at center frequency of 13.56 MHz was developed. However, GNU Radio is not limited to narrowband data transmission. Latencies of GNU Radio and the USRP system have been measured in [11]. According to [11], latencies of hundreds of microseconds or few milliseconds are possible. Latencies in [11] include the latency of the GNU Radio (the computer), the USB latency, and the USRP hardware latency, so the total transmit-receive latency of the PLC system is two times the latency of one end.

Because of the block-based USB, real-time operations are not possible with the present GNU Radio release (3.2.2). If real-time response is not essential, latencies are slow enough for using the GNU Radio in the research of communication channels or control applications.

V. CONCLUSION

The goal of this paper was to study the utilization of software-defined radio (GNU Radio) to implement a power line communication link between an electric motor and a frequency converter. First, the optimal frequency band for communication was selected. Next, a narrowband data transmission link with a 13.56 MHz center frequency that uses DPSK modulations for communication was designed and described. Third, the effect of transmit power and bit rates on the performance of data transmission were tested and analyzed. Finally, a forward error correction method and its influence on BER in the investigated channel were tested. According to experiments, the GNU Radio and USRP can be used in developing and testing power line communication links. For example the effects of different parameters, such as different modulations, frequency band, error correction methods, transmit power, and bit rates can be examined. According to experiments carried out, the GNU Radio is a feasible a low-cost platform for designing, testing and analyzing data transmission links.

REFERENCES