XXI IMEKO World Congress "Measurement in Research and Industry"
August 30 – September 4, 2015, Prague, Czech Republic

APPLICATION OF RECURSIVE LEAST SQUARE ALGORITHM TO ADAPTIVE CHANNEL EQUALIZATION.

Radek Martinek[1], *Jan Vanus*[1], *Michal Kelnar*[1], *Petr Bilik*[1], *Jan Zidek*[1]

[1]VSB TU Ostrava, Department of Cybernetics and Biomedical Engineering, 17. listopadu 15, 708 33, Ostrava Poruba, Czech Republic (radek.martinek@vsb.cz, jan.vanus@vsb.cz, michal.kelnar@vsb.cz, petr.bilik@vsb.cz, jan.zidek@vsb.cz).

Abstract – This paper deals with modern wireless transmission systems. It focusses on real implementation of recursive least squares (RLS) channel equalizer into the system of software defined radio (SDR), in order to minimize the distance between the signal and SNR transmitting channel noise and then reduce the error rate of bit error rate (BER) transmission. The authors aimed to conduct a real measurement using universal software radio peripheral (USRP). Experimental results suggest that researched RLS equalizer embodies better BER values in comparison to commercially most common equalizers of the least mean square (LMS) algorithm group. Moreover, the conducted experiments show that the usage of the SDR conception is very suitable for testing new principles in channel equalization field.

Keywords: Bit Error Rate, Signal to Noise Ratio, Recursive Least Squares algorithm, Software Defined Radio, Universal Software Radio Peripheral.

1. INTRODUCTION

Continuous grow of data transmission using wireless networks can be observed during the year 2014. The providers of wireless transmission of data demand more effective use of the frequency band. This demand corresponds to the appearance of new wireless technologies and standards. Therefore, new systems must be developed, which would lead to minimization of influence of disturbing phenomena in the radio transmission channel [6], [7] of modern wireless transmission systems.

This would lead to an increase of transmission speed with stress on efficiency of the frequency band use. The chosen approach, software defined SDR radio [5], [9] is very suitable for development and testing of such new systems. The key role for these transmission systems plays the software, which can be flexibly changed according to user's needs. The flexibility of used measuring apparatus is stressed in this study. Advantages of this approach in view of designing real applications are described.

The authors deal with the implementation of equalization techniques [6], [7] in the SDR system in order to minimize the signal-to-noise ratio SNR [9] of the transmission channel and thus reduce the transmission error rate BER [6].

The authors use the results of their own study published in [6], [7] concur on it and extend it. In [7] the authors dealt with the implementation of LMS equalizer into SDR system. The results of conducted experiments suggested that equalization with the use of LMS algorithm is [7] insufficient for modern communication systems. Researched adaptive LMS algorithm is simple and undiscerning from a mathematical point of view [7]. On the other hand, it reached smaller convergence speed in real applications and there was a larger filtration process error; this matter of fact is motivated by plenty of other publications, e.g. [3], [4]. Therefore, authors address in this paper real implementation of normalized LMS (NLMS - Normalized Least Mean Squares) equalizer [6]. This algorithm should have better results, as several recent studies suggest e.g. [6], [7].

2. RLS ALGORITHM

RLS is a basic representative of the second class of adaptive algorithms – algorithms based on the Kalman filtration theory [1], [2]. The basic difference from the LMS algorithm family [10] is its own stochastic concept. The RLS method works with the mean values of variables calculated at time instants instead of the sample average means calculated from several realisations of a stochastic process. The filter structure remains the same as in the case of LMS algorithm, only the adaptive process is different with regard to application of average means. As a result, RLS algorithms are characterised by much higher computational complexity than the LMS algorithms. This difference is so huge (RLS tasks are higher by one order than LMS tasks) that it often leads to conclusions that RLS algorithms have no practical usage. On the other hand, if we look at the convergence speed, we find out that the convergence speed of RLS is several times higher than that of LMS. This ensues from the application of time averaging, which predicts highly accurate values.

$$\xi(n) = \sum_{k=1}^{n} \rho_n(k) e_n^2(k) \qquad (1)$$
$$\rho_n(k) = \lambda^{n-k}$$

where k =1, 2, 3… n, parameter λ denotes the "forgetting factor" and is defined within the range of 0 to 1.

2.1. Derivation of the RLS algorithm

The RLS cost function given by eq. 1 indicates that within time n all the preceding estimated error values are required from the very moment of initiation of the RLS algorithm. To put it simply, as the time passes, the amount of data required for processing of the algorithm increases. Therefore, due to its limited memory and abilities, the RLS algorithm in its purest form is practically inapplicable. However, the derivation presumes processing of all the data. In practice, though, the processing only involves a finite number of preceding values corresponding to N sequence of the RLS FIR filter. To derive the algorithm, we first define as the FIR filter output, n as the currently used weight vector, and k as the input vector of the preceding time. The estimated error value is the difference between the required output within time k, and the corresponding value. These and other suitable definitions are expressed by means of eq. 2, for k=1,2,3… n [1].

$$
\begin{aligned}
&y_n(k) = \mathbf{w}^T(n)\mathbf{x}(k) \\
&e_n(k) = d(k) - y_n(k) \\
&\mathbf{d}(n) = [d(1), d(2) \ldots d(n)]^T \\
&\mathbf{y}(n) = [y_n(1), y_n(2) \ldots y_n(n)]^T \\
&\mathbf{e}(n) = [e_n(1), e_n(2) \ldots e_n(n)]^T \\
&\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)
\end{aligned} \tag{2}
$$

If we define X(n) as the matrix consisting of n previous input column vector up to the present time, then $y(n)$ may alternatively be expressed as [1]:

$$
\begin{aligned}
&\mathbf{X}(n) = [\mathbf{x}(1), \mathbf{x}(2), \ldots \mathbf{x}(n)] \\
&y(n) = \mathbf{X}^T(n)\mathbf{w}(n)
\end{aligned} \tag{3}
$$

The cost function may be expressed in the form of a matrix vector using of the diagonal matrix including the weighted coefficients.

$$
\begin{aligned}
&\xi(n) = \sum_{k=1}^{n} \lambda^{n-k} e_n^2(k) = \mathbf{e}^T(n)\tilde{\Lambda}(n)\mathbf{e}(n) \\
&\text{where } \tilde{\Lambda}(n) = \begin{bmatrix} \lambda^{n-1} & 0 & 0 & \ldots & 0 \\ 0 & \lambda^{n-2} & 0 & \ldots & 0 \\ 0 & 0 & \lambda^{n-3} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}
\end{aligned} \tag{4}
$$

By putting in the values from equations 2 and 3, the cost functions may be extended and later reduced, as in equation 4.

$$
\begin{aligned}
&\tilde{\theta}_\lambda(n) = \lambda\tilde{\theta}_\lambda(n-1) + \mathbf{x}(n)\mathbf{d}(n) \\
&\overline{\mathbf{w}}(n) = \tilde{\Psi}_\lambda^{-1}(n)\tilde{\theta}_\lambda(n) \\
&= \tilde{\Psi}_\lambda^{-1}(n-1)\tilde{\theta}_\lambda(n-1) - \mathbf{k}(n)\mathbf{x}^T\tilde{\Psi}_\lambda^{-1}(n-1)\tilde{\theta}_\lambda(n-1) + \mathbf{k}(n)d(n) \\
&= \overline{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\overline{\mathbf{w}}(n-1) + \mathbf{k}(n)d(n) \\
&= \overline{\mathbf{w}}(n-1) + \mathbf{k}(n)(d(n) - \overline{\mathbf{w}}^T(n-1)\mathbf{x}(n)) \\
&\overline{\mathbf{w}}(n) = \overline{\mathbf{w}}(n-1) + \mathbf{k}(n)\overline{e}_{n-1}(n) \\
&\text{where } \overline{e}_{n-1}(n) = d(n) - \overline{\mathbf{w}}^T(n-1)\mathbf{x}(n)
\end{aligned} \tag{4}
$$

Then we derive the gradient of the above stated cost function formula with regard to the filter weights. By means of making it converge to zero, we obtain the coefficients of the filter which minimises the cost function.

$$
\begin{aligned}
&\tilde{\Psi}_\lambda(n)\overline{\mathbf{w}}(n) = \tilde{\theta}_\lambda(n) \\
&\overline{\mathbf{w}}(n) = \tilde{\Psi}_\lambda^{-1}(n)\tilde{\theta}_\lambda(n)
\end{aligned} \tag{5}
$$

In this way the $\mathbf{\Psi}(n)$ matrix may be extended and later rearranged in the recursive layout, which may then be used for a special form of lemma inverse matrix in order to find the inverse matrix required for the computation of the weight vector update. The $\mathbf{k}(n)$ vector, known as the amplifying vector, is included in order to simplify the computation [1].

$$
\begin{aligned}
&\tilde{\Psi}_\lambda^{-1}(n) = \lambda\tilde{\Psi}_\lambda^{-1}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \\
&= \lambda^{-1}\tilde{\Psi}_\lambda^{-1}(n-1) - \frac{\lambda^{-2}\tilde{\Psi}_\lambda^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\tilde{\Psi}_\lambda^{-1}(n-1)}{1+\lambda^{-1}\mathbf{x}^T(n)\tilde{\Psi}_\lambda^{-1}(n-1)\mathbf{x}(n)} \\
&= \lambda^{-1}(\tilde{\Psi}_\lambda^{-1}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\tilde{\Psi}_\lambda^{-1}(n-1)) \\
&\text{where } \mathbf{k}(n) = \frac{\lambda^{-1}\tilde{\Psi}_\lambda^{-1}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^T(n)\tilde{\Psi}_\lambda^{-1}(n-1)\mathbf{x}(n)} \\
&= \tilde{\Psi}_\lambda^{-1}(n)\mathbf{x}(n)
\end{aligned} \tag{6}
$$

The vector $\theta_\lambda(n)$ given by equation 4 may also be expressed in the recursive form. By substituting this expression for $\mathbf{\Psi}^{-1}$ from equation 6 into equation 5, we may finally get the weight filter vector for the RLS algorithm, as in equation 7.

$$
\begin{aligned}
&\tilde{\theta}_\lambda(n) = \lambda\tilde{\theta}_\lambda(n-1) + \mathbf{x}(n)\mathbf{d}(n) \\
&\overline{\mathbf{w}}(n) = \tilde{\Psi}_\lambda^{-1}(n)\tilde{\theta}_\lambda(n) \\
&= \tilde{\Psi}_\lambda^{-1}(n-1)\tilde{\theta}_\lambda(n-1) - \mathbf{k}(n)\mathbf{x}^T\tilde{\Psi}_\lambda^{-1}(n-1)\tilde{\theta}_\lambda(n-1) + \mathbf{k}(n)d(n) \\
&= \overline{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\overline{\mathbf{w}}(n-1) + \mathbf{k}(n)d(n) \\
&= \overline{\mathbf{w}}(n-1) + \mathbf{k}(n)(d(n) - \overline{\mathbf{w}}^T(n-1)\mathbf{x}(n)) \\
&\overline{\mathbf{w}}(n) = \overline{\mathbf{w}}(n-1) + \mathbf{k}(n)\overline{e}_{n-1}(n) \\
&\text{where } \overline{e}_{n-1}(n) = d(n) - \overline{\mathbf{w}}^T(n-1)\mathbf{x}(n)
\end{aligned} \tag{7}
$$

2.3. Implementation of the RLS algorithm

As mentioned earlier in this paper, the memory of the RLS algorithm is limited by the finite number of values corresponding to the sequence of the filter weight vector. In this context, two aspects of the RLS implementation have to be mentioned: First, while the inverse matrix is essential for derivation of the RLS algorithm, the computation of the inverse matrix is not needed for the implementation itself, which considerably reduces the computational complexity of the algorithm. Second, contrary to the algorithms based on LMS, the current variables are updated in terms of iterations and used together with the values of the preceding iteration.

To implement the RLS algorithm, the following steps must be performed in the order stated below:

1. The filter output is calculated using the filter weights from the preceding iteration and the current input vector:

$$
\overline{y}_{n-1}(n) = \overline{\mathbf{w}}^T(n-1)\mathbf{x}(\mathbf{n}) \tag{8}
$$

2. The medium amplification vector is calculated by means of equation:

$$
\begin{aligned}
&\mathbf{u}(n) = \tilde{\Psi}_\lambda^{-1}(n-1)\mathbf{x}(n) \\
&\mathbf{k}(n) = \frac{1}{\lambda + \mathbf{x}^T(n)\mathbf{u}(n)}\mathbf{u}(n)
\end{aligned} \tag{9}
$$

3. The estimated error value is calculated by means of equation:

$$\bar{e}_{n-1}(n) = d(n) - \bar{y}_{n-1}(n) \tag{10}$$

4. The filter weight vector is updated using equation 10 and the amplification vector is given by equation 9:

$$\mathbf{w}(n) = \overline{\mathbf{w}}^{T}(n-1) + \mathbf{k}(n)\bar{e}_{n-1}(n) \tag{11}$$

5. The inverse matrix is computated by means of equation:

$$\tilde{\Psi}_{\lambda}^{-1}(n) = \lambda^{-1}(\tilde{\Psi}_{\lambda}^{-1}(n-1) - \mathbf{k}(n)\left[\mathbf{x}^{T}(n)\tilde{\Psi}_{\lambda}^{-1}(n-1)\right] \tag{12}$$

3. IMPLEMENTATION AND RESULTS

For verification of the real features of the designed system USRP, a passive omnidirectional WiFi Antenna TP-LINK TL ANT2405C with 5 dBi gain – was connected to output of broadcast part of the system and the receiver was affixed with the passive panel sector WiFi Antenna Centurion Wireless Technologies with 9 dBi of gain. In the transmitter the signal was generated by the studied modulations (8-QAM, 64-QAM, 128-QAM]) on the carrier frequency 1,96 GHz, power level of 10 dBm signal with a bandwidth 3,84 MHz, Roll factor α of the root raised cosine filter 0,33 and symbol rate 2,625 $\mathrm{MS}\cdot\mathrm{s}^{-1}$. On the receiver side, the quality of the received signal was evaluated by the BER measurement. The measurement was carried out in the laboratory of the size of 6×14 m, while transmitting and receiving antennas were situated in opposite corners of the laboratory. The room was equipped by the conventional office equipment. A constellation diagram [9] of received signal without applying adaptive equalizer for 128-QAM is shown in Fig. 1. Fig. 2 shows constellation diagram of received signal with RLS adaptive equalizer applied for 128-QAM.

The paper authors used to implement their own system used on virtual instrumentation. The concept of the measuring apparatus is shown in Fig. 3.

Effectiveness of channel equalizer is characterized by the difference between signal and noise in output, see Fig. 4.

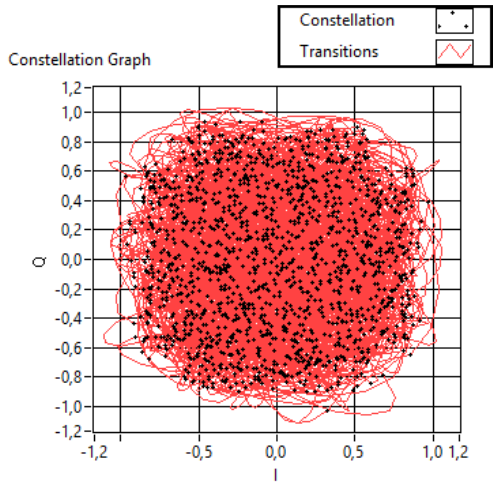

Fig.1 Constellation diagram of received signal without equalization for 128-QAM.

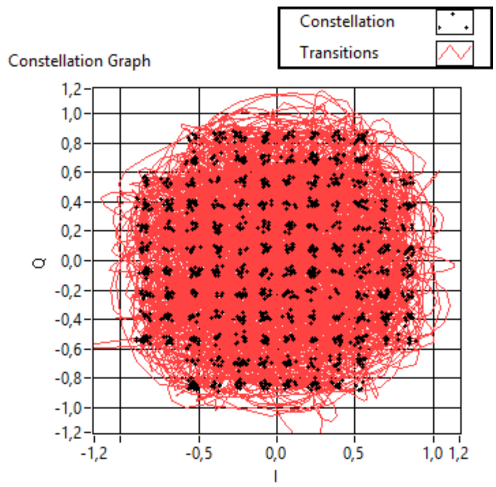

Fig. 2 Constellation diagram of received signal if RLS adaptive equalizer is used, for 128-QAM.

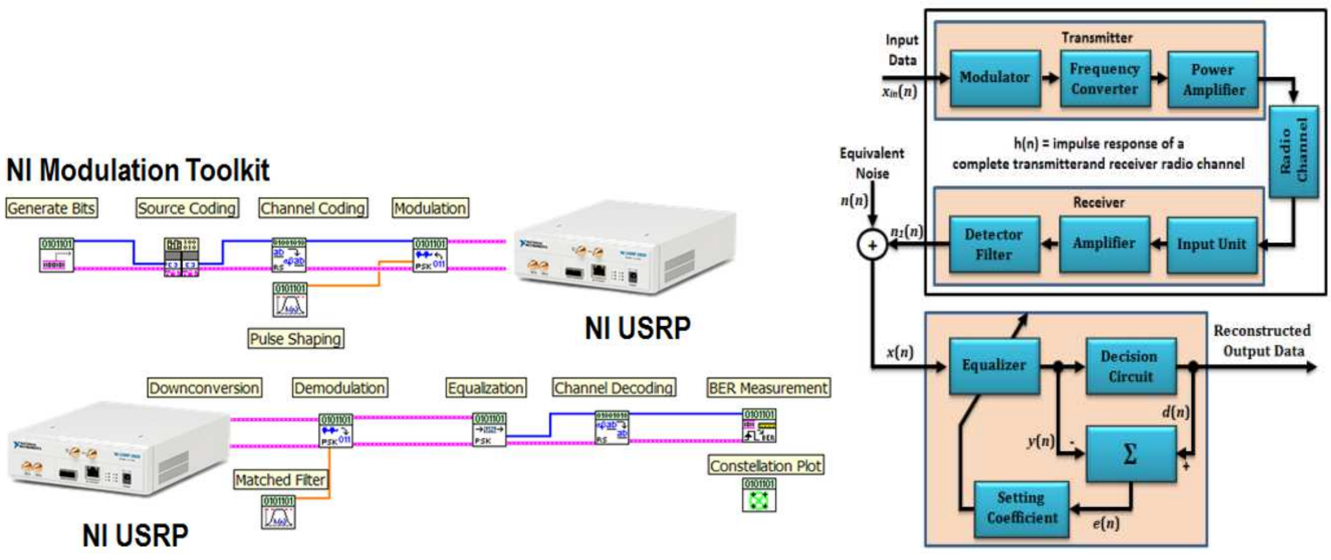

Fig. 3. Universal software radio peripheral: transmitter – receiver and general scheme of equalizer.

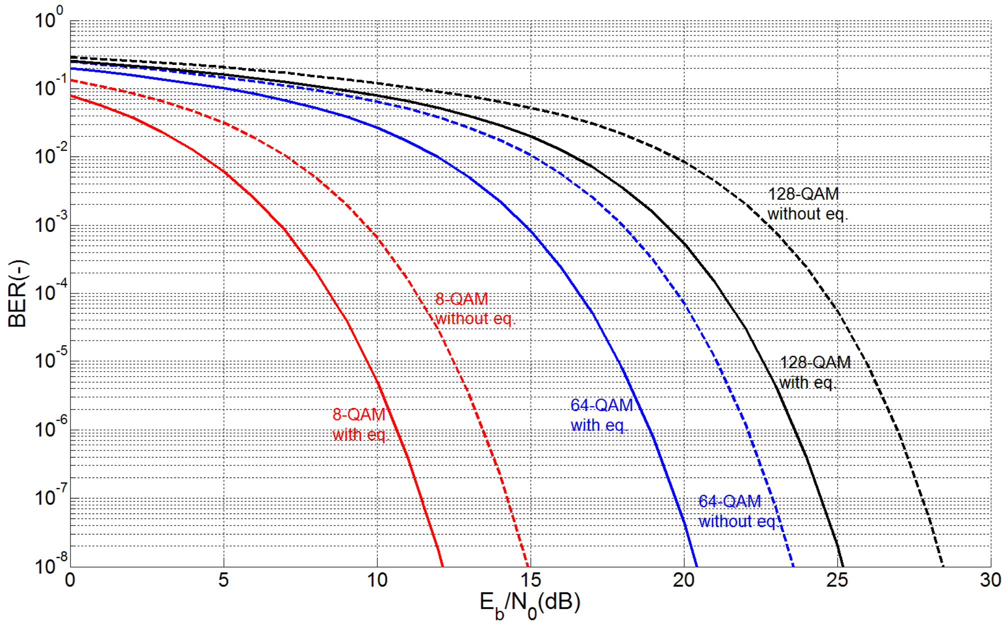

Fig. 4. BER vs E_b/N_0 for recursive least squares channel equalizer (8-QAM, 64QAM, 128-QAM).

4. CONCLUSIONS

The basic goal of this paper is to investigate the application of an algorithm based on adaptive filtering in channel equalization. The main objective is to achieve a high convergence rate in order to meet the requirements for short training time and good equalization properties. In this paper, we have developed a new model based on recursive least squares algorithm.

Linear equalization using RLS algorithm was implemented as a part of experiments using above described methodology. Equalization efficiency varies considerably depending on the signal character or the used modulation. However, generally, the linear equalization can be considered as a suitable method for suppressing the interference in the case of high bit error rate. This type of equalization can have also destructive influence on the signal at low bit error rate caused just by the noise generated in equalization process.

The main contribution of the work lies in the implementation of equalization techniques for wireless transmission systems designed for generation and analysis of digitally modulated signals, see Fig. 3. The hardware of these systems is based on general USRP. The functionality of the system is determined by SW designed in graphical development environment LabVIEW.

ACKNOWLEDGMENTS

This work is partially supported by the Science and Research Fund 2014 of the Moravia-Silesian Region, Czech Republic and projects SP2015/154, SP2015/181 of Student Grant System, VSB-TU Ostrava.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 2001. 4th ed. ISBN 978-0130901262.

[2] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer, New York, 2008. 3rd ed. pp. 632. ISBN 978-0-387-31274-3.

[3] J. Bard and V. J. Kovarik, *Software defined radio: the software communications architecture*. John Wiley and Sons, Wiltshire, 2007. ISBN 978-0470865187.

[4] M. Rupp, "Robust Design of Adaptive Equalizers", *IEEE Transactions on Signal Processing*, vol. 60, pp. 1612-1626. April 2012

[5] LabVIEW System Desing Software. National Instrument [online]. 2012. Available at: http://www.ni.com/labview.

[6] R. Martinek and J. Zidek, "The Real Implementation of ANFIS Channel Equalizer on the System of Software-Defined Radio", *IETE Journal of Research*. Taylor & Francis, London, UK, 2014, Volume 60, Issue 2, pp. 183-193, June 2014.

[7] R. Martinek and J. Zidek, "The real implementation of NLMS channel equalizer into the system of software defined radio", *Advances in Electrical and Electronic Engineering*, vol 10, Issue 5, pp. 330-336, December 2015.

[8] R. Martinek and J. Zidek, "Use of Adaptive Filtering for Noise Reduction in Communication systems", *The International Conference Applied Electronics (AE)*. pp. 215-220, Pilsen, Czech Republic, September 2010.

[9] R. Martinek, A. M. Wohaishi and J. Zidek, "Software Based Flexible Measuring Systems for Analysis of Digitally Modulated Systems", *The 9th Roedunet International Conference, RoEduNet*. pp. 397-402, Sibiu, Romania, June 2010.

[10] J. Vanus and V. Styskala, "Application of variations of the LMS adaptive filter for voice communications with control system", *Tehnicki Vjesnik-Technical Gazette*, vol. 18, pp. 553-560, December 2011.

[11] P. Koudelka, J. Latal, P. Siska, J. Vitasek, A., Liner, R. Martinek and V. Vasinek, "Indoor visible light communication: modeling and analysis of multi-state modulation", SPIE 9224, *Laser Communication and Propagation through the Atmosphere and Oceans III*, October 2014.

[12] P. Koudelka, P. Soltys, R. Martinek, J. Latal, P. Siska, S. Kepak and V. Vasinek, "Utilization of M-QAM modulation during optical wireless car to car communication", *Conference Proceedings: Optical Fibre Technology*, Melbourne, Australia, 6-10 July 2014, pp. 452-454, 2014.

[13] R. Martinek and J. Zidek, "A System for Improving the Diagnostic Quality of Fetal Electrocardiogram", *Przeglad Elektrotechniczny*, vol. 88, Warszawa, Poland, pp. 164-173, 2012.