Hardware Module of Fisher Linear Discriminant for Biomedical Signals Classification

Denis A. Prilutsky[1], Arsenty P. Gusev[2]
Bauman Moscow State Technical University
Moscow, Russia
[1]denis.prilutsky@outlook.com, [2]xitren@ya.ru

Dmitry A. Balovnev
Brook INEUM
Moscow, Russia
balovnev_d@ineum.ru

***Abstract*—This paper is to show approach to implementation of biomedical signals classifier hardware module based on the Fisher linear discriminant. Complexity estimation of classifier training algorithm is given. The structure of the classifier based on FPGA, CPLD allowing to parallelize data flows on the set number of channels is developed.**

***Keywords*— analysis of biomedical data; Fisher linear discriminant; pipeline; FPGA; CPLD**

I. Introduction

Classification of multidimensional data is widely applied in biological and medical researches [1, 2, 3]. One of the effective algorithms for signals division into two classes is the Fisher linear discriminant (FLD) which reduces space dimension from initial to one by projecting of multidimensional data on a straight line. The quality of learning of the classifier depends on amount of signs. Increase in amount of signs, as will be seen below, leads to increasing of computational complexity of an algorithm. Therefore, for reduction of dimension, preprocessing of entrance data, for example, decimation of an entrance, signal can be applied. However, such approach can be unacceptable in systems where the useful signal can have amplitude, comparable with an amplitude of noise. In the Fig. 1 the frequency spectrum of EEG (electroencephalography) signal received as a result of an experiment is shown.

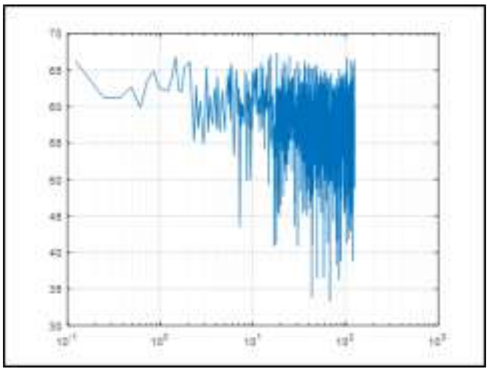

Fig. 1. Spectrum of the EEG signal obtained as a result of an experiment

In medical measurements using microcontroller systems based wearable devices, implementation of the classifier is practically not possible. Moreover, in devices where the number of leads can be more than 128 it is completely impossible due to the limited performance of such systems.

To solve this problem, it is possible to use FPGA (Field Programmable Gate Array), CPLD (Complex Programmable Logic Device), which allow designing of parallel computing architecture for multidimensional data.

The purpose of this project is to develop hardware module that implements a FLD-based classification for one input signal measuring element.

In order to check that such a classifier is worth designing, the learning algorithm for the classifier based on FLD in the experiment using the P300 method [4] will be considered and computational complexity of the algorithm will be estimated.

II. Brief Description Of The Experiment

The P300 method has two modes of operation: the training mode and the control mode.

In the training mode, on the computer monitor or on the special device of the bearer, stimuli are given – images that should cause the reaction P300 in the occipital areas of the brain – the evoked potential (EP), appearing on average after about 300 ms and having a positive amplitude on the EEG. P300 is a reaction to the expected visual stimulus, which is presented at a certain time in a certain position on the screen.

Fig. 2 demonstrated the images shown to the person in the training and management mode.

The stimulus in our experiment is a randomly changing background in the arrows' images.

The arrow in the brown color is highlighted, which is currently the target – for which the person needs to be watched during the entire training period. The stimulus in this position is the reaction P300. On each target image, the number of stimuli presented (the number of background fires) is ~5–7. The selection of the target image and presentation of the stimulus is carried out independently and randomly.

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

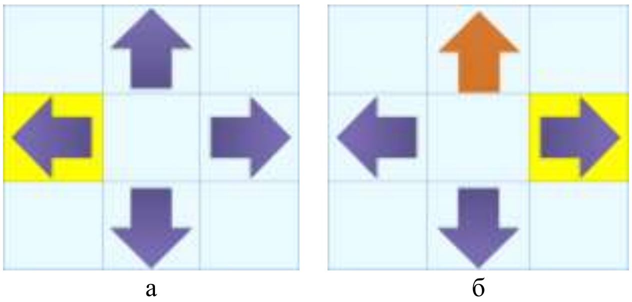

Fig. 2. Images presented to a person in the training and management mode: a) background lighting of the picture – stimulus; b) the target image is highlighted in brown

Recording EEG is continuous. Synchronously with it, the stimulus data is recorded – the time and position of the presentation.

In the training mode it is necessary to register and divide by the classifier two types of reactions: P300 is present; P300 is missing. The training period lasts for 5–7 minutes. The next stage is the control mode.

In the control mode, a distinction is made for the presence in the P300 reaction signal or its absence using the classifier trained at the previous stage. An important point here is that we all continue to present the same images and incentives and record the timestamps of their presentation. The person focuses his attention on the selected picture – the reaction of P300 is detected and, in case the number of reactions is more than 3-5, the selected arrow is highlighted.

Next, we describe in more detail the preparation of measurement data for the training of the FLD-based classifier.

III. Preparation of Data of Measurements

We divide the EEG into fragments relative to the time of appearance of the bearer incentives.

Define the input vector of the signs for the following analysis:

$$S = \left[\bar{u}_0 \quad \bar{u}_1 \quad \dots \quad \bar{u}_{m-1}\right]^T,$$

where $\bar{u}_i = \left(u_0 \quad u_1 \quad \dots \quad u_{n-1}\right)$ – measured stresses from the surface of the human scalp; m – the dimension of the signs vector; n – number of measurements. This number in the experiment $n = 125$.

To train the classifier, you need to divide the vectors with target EPs and non-target ones. To separate the data, a vector is used that contains information about when the visual signal source was triggered, which makes it possible to extract vectors containing EP. We get two classes – two matrices containing sequences of signs with targeted stimuli, S_1, and non-targeted, S_2:

$$S_1 = \left[\bar{u}_{p_0} \quad \bar{u}_{p_1} \quad \dots \quad \bar{u}_{p_{m_1-1}}\right]^T;$$

$$S_2 = \left[\bar{u}_{d_0} \quad \bar{u}_{d_1} \quad \dots \quad \bar{u}_{d_{m_2-1}}\right]^T,$$

where $p_0, p_1, \dots, p_{m_1-1}$ – indices of the signs of the original matrix containing target incentives; m_1 – the number of features that contain targeted incentives; the dimension of the resulting vector; $d_0, d_1, \dots, d_{m_2-1}$ – indices of characteristics of the original matrix containing non-targeted incentives; m_2 – number of sequences that do not contain targeted stimuli; the dimension of the resulting vector.

Let's describe the learning process of the classifier.

IV. Training Of Classifier

Calculate the mean values, $\bar{S}_1$ and $\bar{S}_2$, for each class:

$$\bar{S}_1 = \left[\bar{U}_{p_0} \quad \bar{U}_{p_1} \quad \dots \quad \bar{U}_{p_{m_1-1}}\right]^T =$$

$$= \frac{1}{m_1}\left[\sum_{i=0}^{n-1} u_{p_0,i} \quad \sum_{i=0}^{n-1} u_{p_1,i} \quad \dots \quad \sum_{i=0}^{n-1} u_{p_{m_1-1},i}\right]^T;$$

$$\bar{S}_2 = \left[\bar{U}_{d_0} \quad \bar{U}_{d_1} \quad \dots \quad \bar{U}_{d_{m_1-1}}\right]^T =$$

$$= \frac{1}{m_2}\left[\sum_{i=0}^{n-1} u_{d_0,i} \quad \sum_{i=0}^{n-1} u_{d_1,i} \quad \dots \quad \sum_{i=0}^{n-1} u_{d_{m_2-1},i}\right]^T.$$

Next, we calculate the covariance matrices for each class:

$$\Sigma_1 = \operatorname{cov}(S_1);$$

$$\Sigma_2 = \operatorname{cov}(S_2).$$

The covariance matrices obtained have a dimension m_1 and m_2.

We calculate the intraclass scatter matrix:

$$S_W = \Sigma_1 + \Sigma_2.$$

The resulting matrix has dimension $m_1 = m_2$.

We calculate the vector W:

$$W = S_W^{\,*}\left(\bar{S}_1 - \bar{S}_2\right),$$

where $S_W^{\,*} = S_W^{\,T} \times \left(S_W \times S_W^{\,T}\right)^{-1}$ – pseudoinverse matrix whose dimension is equal to the dimension of the matrix of the intraclass spread.

We find the projections of two classes on a vector W:

$$Y_1 = W^T \times S_1;$$

$$Y_2 = W^T \times S_2.$$

The dimension of the obtained vectors is m_1 and m_2. Define the threshold for two classes. Find the average values for the first and second class:

$$M_1 = \frac{1}{m_1} \sum_{i=0}^{m_1 - 1} y_{p,i};$$

$$M_2 = \frac{1}{m_2} \sum_{i=0}^{m_2 - 1} y_{d,i},$$

we obtain the value of the threshold for finding the vector belonging to the class:

$$w_N = \frac{| M_2 - M_1 |}{2}.$$

Next, to classify an arbitrary vector X it is multiplied by a vector W and addition with the received threshold:

$$d(X) = \sum_{i=0}^{N-1} w_i \times x_i + w_N.$$

The decision to belong to the class is made:

$$\operatorname{sgn}\left[d(X) \right] = \begin{cases} \Omega_1; \\ \Omega_0; \\ \Omega_2, \end{cases}$$

where Ω_1, Ω_2 – first and second class; Ω_0 – the vector does not belong to either the first or second class.

Next, we estimate the complexity of the applied computations.

V. Computational complexity of algorithm

Let's give the number of arithmetic operations in the notation "O".

Multiplication of matrices:

$$O_{mult}(m, n, p) = m \times n \times p,$$

where m – number of lines of the first matrix; n – the number of columns of the first matrix and the number of rows of the second; p – number of columns of the second matrix.

Addition or subtraction of matrices:

$$O_{sum}(m, n) = m \times n.$$

Computation of the inverse or pseudoinverse matrix:

$$O_{pinv}(m) = m^3.$$

Vector of matrix mean values:

$$O_{Expected}(m, n) = m \times n.$$

Calculation of the covariance matrix:

$$O_{cov}(m) = m^3 + m.$$

Combining the dependency data to calculate the vector W we obtain the total number of operations. The dependence of the number of arithmetic operations on the dimension of the input data is shown in the Fig. 3.

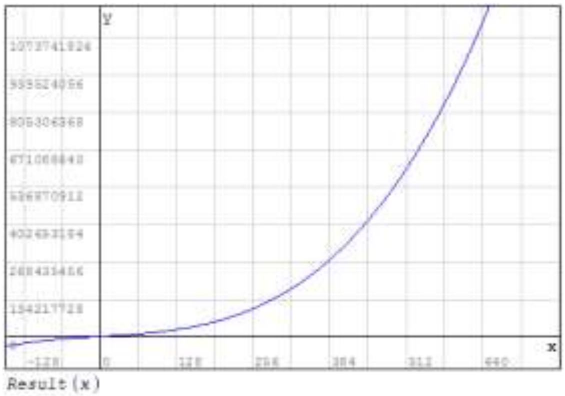

Fig. 3. Dependence of the number of operations on the dimension of the input characteristic vector

As can be seen from the figure, the number of operations depends nonlinearly on the dimension of the feature vector. To increase the performance of the classifier, it is required to develop a module structure, based on existing architectures.

VI. Choice Of System Architecture

For use of advantages of CPLD the architecture of Pipeline which is simple and reliable conveyor architecture has been chosen. It can consist of any quantity of components (Filter) which will transform or filter data before transmitting them through channels (Pipe) on other components. Components of such system begin the work on arrival of all data with the channel. The architecture is often used as the simple sequence, but she can also be used for more complex structures. In the figure 4 the example creation of system on the basis of this architecture is shown.

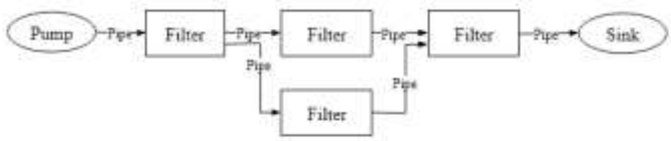

Fig. 4. An example of system on the basis of architecture of Pipeline

Pump is data source. It can be the static text file or the input equipment which is constantly generating new data. Sink or the consumer is the receiver of data. It can be other file, the database or the screen of the computer. In our case data source is the SPI interface. The receiver – the subsequent cascade of the filter, or a conclusion of data to the microcontroller.

Further, we will describe the proposed solution.

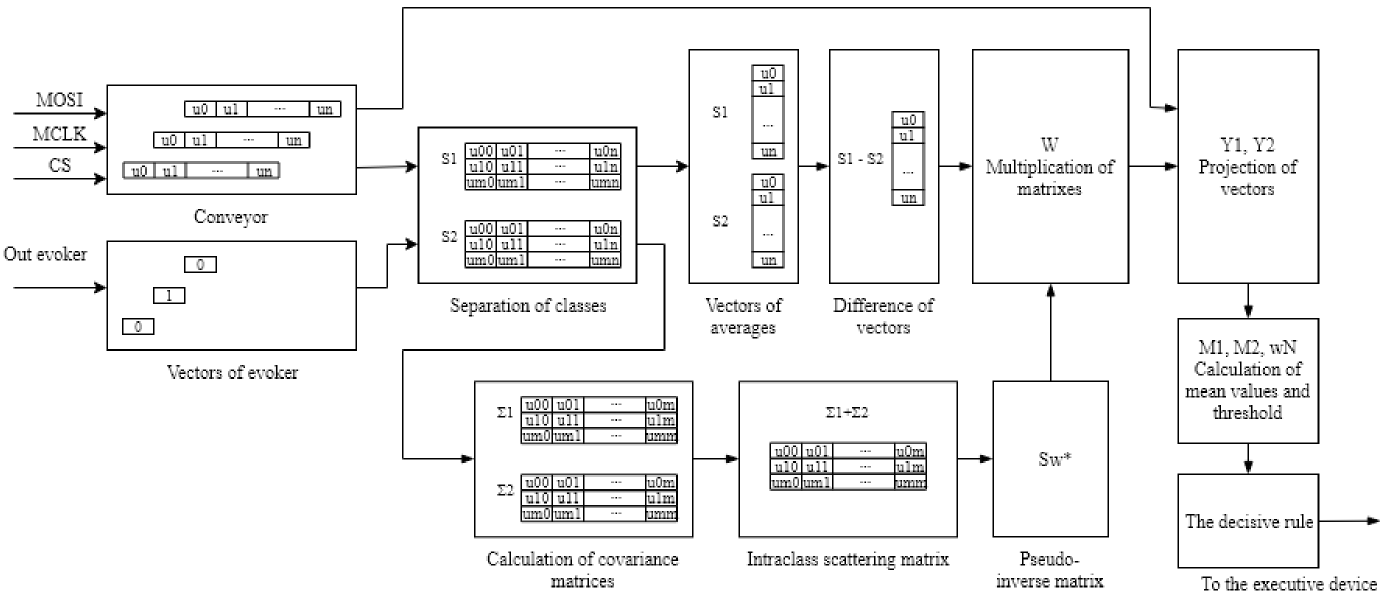

Fig. 5. Structure of the module of the classifier

VII. The Proposed Solution

In the Fig. 5 the offered structure of the module of the classifier realized in CPLD is shown. This structure can be set with use of language of the description of the equipment of integrated circuits, such as VHDL (Very High Speed Integrated Circuits Hardware Description Language), or Verilog HDL.

The conveyor – accumulates the data coming from SPI port (Serial Peripheral Interface). Further, there is a division of vectors according to a vector of signs which is accepted synchronously with arrival of data of the conveyor. Storage of vectors is made in the synthesized memory of a chip. After division of vectors into classes, recalculation of a pseudo-inverse matrix proceeding from property of a covariance matrix is made:

$$\operatorname{cov}(X,Y)=E(X\times Y)-\overline{X}\times\overline{Y}.$$

and a vector, keeping average values. As a result of multiplication of matrixes we will receive a matrix of coefficients of W. With arrival of a special signal of the status of the end of training, projection of entrance vectors on the received straight line of W begins and by the decisive rule classification is made: information is transferred on a serial port to the microcontroller.

VIII. Conclusion

When synthesizing a configuration for chips of Lattice Semiconductor of the LCMX02 series the number of LUTs (look-up table) for formation of logic of the module was <1000 for one block of the classifier. Storage of intermediate calculations in the synthesized memory of a chip has made 33 EBR (Embedded Block RAM).

Thus the hardware module of the classifier on the basis of FLD has been received.

References

[1] Aboul Ella Hassanien, Ahmad Taher Azar, Brain-Computer Interfaces. Intelligent Systems Reference Library Vol.74, 416 p.

[2] Metody statisticheskoj obrabotki medicinskih dannyh: Metodicheskie rekomendacii dlya ordinatorov i aspirantov medicinskih uchebnyh zavedenij, nauchnyh rabotnikov / sost.: A.G. Kochetov, O.V. Lyang., V.P. Masenko, I.V.ZHirov, S.N.Nakonechnikov, S.N.Tereshchenko M.: RKNPK. 42 p. (*In Russian*)

[3] Guang-Zhong Yang, Body Sensor Networks, 2014

[4] Juri Kropotov, Quantitative EEG, Event-Related Potentials and Neurotherapy, 2008.

[5] Rhythms of the Brain / G. Buzsaki, Oxford: Oxford University Press, 2006.

[6] Pattern classification and scene analysis / Richard O. Duda, Peter E. Hart, 2001.

[7] M. Karkooti, J.R. Cavallaro and C. Dick, FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm, 2005.

[8] A. Irturk, S. Mirzaei, and R. Kastner, An efficient FPGA implementation of scalable matrix inversion core using QR decomposition, Univ. California, San Diego, San Diego, CA, USA, UCSD Tech. Rep. CS2009-0938, 2009.