INNOVATIVE INFOTECHNOLOGIES FOR SCIENCE, BUSINESS AND EDUCATION

Vol. 1(18) 2015

CONTENTS

IITSBE, **Vol. 1(18) 2015**

E-government: from E-registration and E-information board to E-governance

Aleksei Iurasov[1,2a]
[1] Department of Business Management,
Vilnius Gediminas Technical University, Saulėtekio avenue 11, LT-10223 Vilnius, Lithuania
[2] Vilnius Business College, Kalvariju str. 125, Vilnius, Lithuania

Received 11 November 2015, accepted 15 December 2015

Abstract. This article is both theoretical and an applied one. It is actuality determined by the necessity to form the theoretical background for setting up the e-government. Sometimes practical applications are carried out in directions, which are not covered by applied scientific developments. This is the reason for the low efficiency of e-government projects implementation. Unlike similar projects of corporate automation, investment in e-government does not lead to a significant reduction in public spending or qualitative changes in governance. At the time when the business is developing management innovation create new business processes and organizational forms, such as virtual enterprises, the state automates the business processes of the 20th century.

Citations: Aleksei Iurasov. E-government: from E-registration and E-information board to E-governance – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 3-6.

Keywords: e-government; e-governance; Decision Support Systems (DSS); balanced scorecard (BSC); Key Performance Indicators (KPI); geographical information systems (GIS).
JEL: C82 - Methodology for Collecting, Estimating, and Organizing Macroeconomic Data; E61 - Policy Objectives; Policy Designs and Consistency; Policy Coordination; H11 - Economic Impacts of Globalization: Policy; Structure, Scope, and Performance of Government; O38 - Technological Change: Government Policy; R50 - Regional Government Analysis: General.

Short title: E-government.

Introduction

For the moment being there is no common concept of the electronic government. There is only a set of general requirements defined by the citizens and businesses to be accepted by government of information society. Various categories of consumers are united in aspiration to receive more effective access to the information in order to reduce cost of transactions, to make interaction with state structures simpler, faster and more comfortable.

Practical work in this direction is not always based on the up-to-date scientific developments, and sometimes the work is carried out in directions, which are not covered by applied scientific developments. This is the reason for the low efficiency of e-government projects implementation [1].

Automation of work of the state (in the form of creation of the e-government) is at the initial stage of theoretical judgment, development of technologies, instruments and mentality of the government and the society [2]. Thus, in the current scientific workouts and also in published articles and reports of the conferences a great attention is put on the private questions: differences in the implementation of e-government services between developed and developing countries [3]; to characteristics of the implementation of e-government services in particular countries [4]; to awareness of citizens and businesses of the expected effect of the use of electronic government services [5]; to assessing the development of e-government [6-8].

The majority of authors usually approach the description and problem solving empirically [4]. This approach is characterized by general discussions about the principles, lack of mathematical modeling links, economic calculations, software engineering as the single system, government's support in decision-making [2]. It is caused by absence of the complex scientifically-proved approach to construction of electronic system of a government administration, the general standards of its functioning focused on improvement of quality of life of the population, growth of competitiveness in the given territory and other strategic targets. Very often under the name of "the electronic government" is given to separate sites which enables citizens and the enterprises making transactions with the state (or getting state services).

This brings a certain confusion of concepts: "Electronic government" is understood as the "Electronic information desk and registry" which is deprived from the functions of management in a territory on the basis of the advanced information technology [9].

[a]Email: aleksei.iurasov@gmail.com

1. Research analysis

In order to understand the difference, let us focus on definition: the electronic government represents the system of the government based on automation of all administrative processes at the national level and serving as objective a significant increase in the efficiency of the government [9]. The above-mentioned approach can not give neither the "significant increase in the efficiency of the government" nor "automation of all administrative processes at the national level".

The functional capabilities of the "automation of all administrative processes at the national level" will be illustrated by several examples. Suppose that the information accumulated in the automated systems of public institutions is processed on the basis of Data mining and aggregated into layers of GIS e-governance. Furthermore, becomes available in different levels of decision-making in a Decision Support System (DSS) executive authorities, which increases the speed and quality of decision making. Standard applications and algorithmic approach may be applied in several cases:

1. Handling investor who wants to build a plant for the production of building materials, the system will determine the optimal location of the enterprise. To do so, all necessary information is extracted from corresponding GIS layers: human resources, roads and engineering infrastructure, natural resources (sand, gravel, etc.). Thus, there are considered the priorities of territorial development such as support for depressed areas (high unemployment, low tax flow in municipal budget, etc.) and the options for the plant with their detailed argumentation are provided.

2. A serious seasonal task for the authorities becomes spring flooding. Flood forecast by meteorologists leads to the automatic calculation of the probability of loss for all the objects within the flood zone and flooding. The loss is defined on the grounds of the objects located in the terrain landscape according to GIS. For such purposes the information about the objects of engineering infrastructure, buildings and facilities, storage of hazardous chemicals and waste management, enterprises and farms, sites of cultural and historical value, etc. is used.

Furthermore, in order to find a solution, the system requires to specify the costs of flood prevention for specified areas. It might be applied to the large-scale projects, such as dam construction etc. Otherwise apart from running costs (but including costs of new working places, income tax, etc.), the costs of evacuation of people and property during the time of flooding (based on the size of compensation for the lost property) and the costs of people transportation/relocation are taken into the consideration. The system will output the summary which allows us to make well-grounded decisions on elimination of dangerous flood consequences. In some cases shore reinforcement and the construction of a dam will be accepted, in other cases, relocation of road or/and electricity transmission lines into safe place. Exceptional cases might require fund allocation for the evacuation of several elderly residents of a deserted village and compensation for the farm property losses.

3. The monitoring of economic, social-demographic, urban planning, natural- environment, etc. situation in the country in real time is reflected on the government website by means of Key Performance Indicators (KPI). For example, you can see a negative dynamics of indicator "standard of living". By clicking the button it enters the section of indicators that shows the standard of living. Firstly, there you can find the ratio of infected by influenza and healthy people is close to the level of epidemiological proportion and anti-epidemic measures should be carried out. Secondly, the increased mortality rate in road accidents requires necessary investments in road lighting and/or improvement of the road police activity. Thirdly, due to the increase in cardiovascular diseases, necessary prevention activities should be taken into the consideration.

4. Multi-type information systems of different public services will either be unified or replaced by a single integrated system (ensuring information transparency at all levels). For example, in the unified system all business processes of public servants will be reflected in the form of KPI indicator changes.

5. The State government could be treated as a mathematical function due to the following approaches:

a) target values of the main socio-economic indicators have been approved by elections (e.g., "standard of living");

b) there are well-known formulas for calculating the indicators;

c) there is information about the decision taking (there is a possibility to check how the *claimed* goals pass toward the *declared* goals);

d) the final result is formulated (for example, changes in the value of the index "standard of living").

6. It will be possible to choose not politicians that give certain promises, but the system of government objectives. For example, one politician proposes to increase the value of "standard of living" by improving the indicators included in the formula for calculating the "standard of living" taken from the sections "Health" and "Education". Whereas, his opponent in the elections demonstrates the feasibility of increasing the value of "Competitiveness of the territory / country", while improving "the ease of doing business level", "Investment attractiveness" and "Security". The opponent also argues that in the long term it will lead to the growth of "standard of living". The opponents operate on exact figures.

7. DSS of e-government collects the information on all decisions taken and the reasoning for retrospective analysis. The described solutions play an important role in increasing transparency, accountability of the government officials and focus on the implementation of strategic goals of e-government.

Apart from quality indicators mentioned above, technology of Decision Support System (DSS) and Business Intelligence (BI) are used for the operation of public administration. Taken together, these technologies are used to enhance the efficiency of public administration, and represent an integral part of the concept of government intelligence.

The database collected for manipulating purposes should combine all the information of public organizations and in the future will become the largest repository of information ever created by man. Accumulation of information could be proceeded by means of Data mining technology (autocorrelation, clustering transactions, Self-organizing Kohonen map, Expectation-maximization (EM) algorithm, logistic regression, neural networks, decision trees, etc.). The extraction and evaluation of the selected/sorted information will allow data user to develop the effective algorithms for decision making system.

Another important problem is the establishment of e-government. It means that the processes of re-engineering of the government, based on the principle of full use of new opportunities, must be provided by modern information technologies. The traditional offline business processes and organizational structures of the government are not effective in the information society.

The experience how to tackle the problem can be borrowed from the corporative automation area which is based on ERP technology. A key task is formulated as follows: a) forming a single logic of business processes of public administration; b) optimization in order to increase public efficiency; c) giving maximum transparency in these processes. In order to implement this task, the tools of modeling and optimization of business processes should be used (BPM) [10].

Fig. 1 represents structure of e-governance GIS.

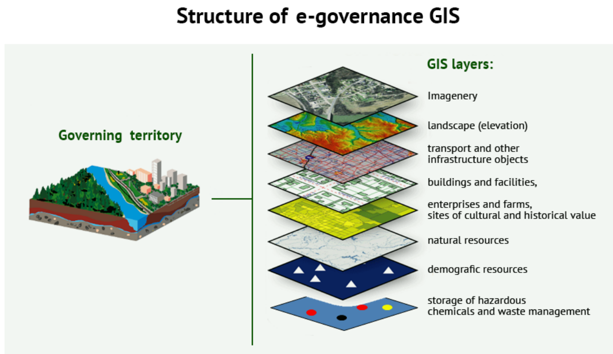

Fig. 1. Structure of e-governance GIS.

2. Unification and standardisation

In addition, some problems, which are typical for the initial phase of corporate automation, must be solved in order to establish e-government. For example, "flap automation", when part of the functions (business processes) are automated using a wide range of often incompatible systems that operate separately [1].

Firstly, the implementation of this approach within the EU will require unification of the power structures, clearly constructed and formal logic of the actions of public officials. For example, currently the executive authorities in the different Member States are different in both: their structure and order of related functions. The structure, as well as the procedure for allocation of responsibilities and the algorithms for performing state functions will eventually be unified.

In some cases, it is advisable to create some unified interstate standards for e-government. For example, among the EU countries there is quite high mobility of the population which makes it feasible to develop a unified structure of electronic passports for all EU member states (smart cards with certificates of electronic digital signature or biometric identification information, passport data, hospital card, medical insurance, diplomas and certificates of education, driver's license and documents of the state registration of rights to real estate, banking and billing information, etc.).

Secondly, the executive level of e-government uses several technologies presented below.

1. Citizen relationship management (based on CRM- technologies - in this case, instead of customers are the citizens, instead of commercial goods and services - government services) built on the "single window" principle. To provide a personalized service, taking into account individual circumstances and demands of people, all services and information should be organized not in terms of the structure of the state (by departments and agencies) but from the point of view of the citizens (in accordance with any events in the life of people). These life events or "episodes" can include birth, marriage, death of loved ones, change of residence and admission to the educational institutions, the own business organization, etc. Experience gained with the help of Data Mining and Business Intelligence (BI) is used (In commercial CRM-system) to improve service and increase customer satisfaction. This experience will also benefit in e-government.

2. E-taxes involve declaration and payment of taxes, fines and fees. Statements may be available not only under quarterly basis but also automatically in real time. The possibility of e-tax declaration may extend the order of tax payment which entails a revolutionary change in taxation.

3. Knowledge management involves the expert systems, based on artificial intelligence and Internet technologies, in order to improve the quality of administrative work, to reduce the execution time and to automate the processing of stored information.

Conclusions

The results of the implementation of science-based approach to the construction of e-government influence the formation of an information society to a much greater extent than the results of the currently accepted empirical approach.

Major changes will occur in the sectors of e-government as G2B (government to business - as a provider of services - the state, as well as a consumer - the business), G2G (Government to Government - in which entities on both sides are interacting - the state institutions), G2C (government to consumer - as a provider of services - the state, as well as a consumer - a real person).

Those who criticize the position of algorithmization activities of government officials claim that in politics often it is necessary to make decisions that cannot be justified in the achievement of the strategic objectives of Balanced Scorecard (according to decision at governmental level). For example, a politician can make decisions that may result not in growth but in declining of value in "living standards of the area". However, this argument cannot be accepted because this parameter (described by official in DSS system) represents the value or expression which not corresponds to the strategic objectives. Moreover, there are no formalized personal goals related to politics or its affiliated structures. Thus, this point of view does not indicate that the goals are of higher priorities than the target growth of living standards.

It should be noted that for certain states and developing companies the creation of an integrated DSS governance encompass difficult and complex project. The problem of project implementation at this level is closely associated with shortage of resources. The solution of the implementation problem at governmental level depends on the development of an open architecture DSS (starting with the structure of the GIS, KPI and BSC libraries). This will involve the creation of a library devoted to the decision-making algorithms alongside with the participants and stakeholders from different countries. As an analogy, you can offer an open architecture IBM PC (involve the creation of the personal computer industry by designers and manufacturers from around the world) and a consortium WWW (provide open standards for growth and development of the Internet). The transparency of decision-making mechanisms in e-government should be grounded from the first step of its development.

To stimulate this process, national laboratory "Standards decision support algorithms in the framework of e-government" may be established.

Historically, the Internet occurrence is related to the construction of the network services to meet the needs of public organizations. To solve the technological task (how to improve the reliability of net) the founders of the Network have implemented such mechanism of an information exchange which allows us to organize the connections among customers on a modern level of computer development and telecommunication technologies, which are more suitable in realizing many base functions of the state management.

References

1. Stanimirovic D. Vintar M. . Conceptualization of an Integrated Indicator Model for the Evaluation of e-Government Policies. – *Electronic Journal of e-Government* 11(1) (2013) 293-307.
2. Iurasov A. Electronic government: development of the scientifically-proved concept. The use of multidisciplinary research for the introduction of new training programs/modules and/or new teaching methods in the field of E-Commerce. – Proceedings of the international conference of the "ECOMMIS" TEMPUS project. TU Berlin, 2012.
3. Chen Y.N., Chen H. M., Huang W. E-Government Strategies in Developed and Developing Countries: An Implementation Framework and Case Study. – *Journal of Global Information Management* 14(1) (2006) 23-46.
4. Franke R., Kroenung J., Born F. Eckhardt A. Influential Factors for E-Government Success in the Middle East: Case Study Evidence from Saudi Arabia. – *Electronic Journal of e-Government* 11(1) (2015) 39-62.
5. Joseph B. K. du Plessis T. Consumers' Awareness of the Value of e-Government in Zambia: Empirical Evidence. – *International Journal of Electronic Government Research* 11(3) (2015) 1-23. – doi:10.4018/IJEGR.2015070101.
6. Gueorguiev I., Dimitrova S., Komitskal M. Balanced Scorecard Based Management Information System. A Potential for Public Monitoring and Good Governance Advancement. – *Electronic Journal of e-Government* 3(1) (2005) 29-38.
7. Codagnone C. e-Government Economics Project (eGEP). – Measurement Framework Final Version/RSO SPA, 2006.
8. Siskos E., Askounis D. & Psarras J. Multicriteria decision support for global e-governing evaluation. – <https://www.researchgate.net/publication/260251528_Multicriteria_decision_support_for_global_e-governing_evaluation>, accessed 2014 12 01.
9. Iurasov A.V. Basics of electronic commerce: the Textbook. – Moscow:Hot line-Telecom, 2013.
10. Niehaves B., Plattfaut R. & Becker J. Business process management capabilities in local governments: A multi-method study. – *Government Information Quarterly* 30(3) (2013) 217-225.

Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 1. Problem overview.

Vadim Gerasimov [1 a] , Gintaras Jonaitis [2], Vytautas Jonkus [1]
[1] Machine-to-Machine laboratory, Department of Radiophysics,
Faculty of Physics, University of Vilnius, Saulėtekio al. 9-III, LT2054 Vilnius, Lithuania
[2] Laboratory for Medical Rehabilitative and Assistive Technologies, Department of Biomechanics,
Faculty of Mechanics, Vilnius Gediminas Technical University, J. Basanavičiaus str. 28, Vilnius, Lithuania

Received 12 June 2015, accepted 5 August 2015

Abstract. General principles for creating the electronic system devoted for processing and recognition of myoelectric signals were observed. Due to functional importance, usage of Artificial Neural Networks (ANN) for recognicion purposes were estimated as principle.

Citations: Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 1. Problem overview – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 7-12.

Keywords: Artificial Neural Networks; ANN; recognition of myoelectric signals; electromyogram; linear regression method; supervised learning; logistic regression; artificial neuron.
Short title: Artificial Neural Networks. 1. Literature overview.

Introduction

Since the old times people were figuring out how to make their life easier by using machines and mechanisms. Only recently the technological development has become more rapid than anytime before. The control of mechanisms and their interaction with human stimulated the most considerable interest. Such control would not be just mere buttons, but rather sensors, suited for the registration of human body signal.

Such a sensor is a neuron activity scanner or myoelectric probes and electrodes. A signal sensed by these nodes can be barely interpreted seeking required accuracy. In that case signal processing algorithms, which automatically adapt to the required mode, are used. This function is performed by the elements of an *Artificial Inteligence* (AI) algorithm. In this case *Artificial Neural Networks* (ANN) were chosen.

The goal of this work could be formulated as follows: to observe the general principles for creating the electronic system devoted for processing and recognition of myoelectric signals.

1. Task overview

The muscular activity is registered by electrodes which form the electric signal as the difference in potential between the beginning node and the end node of certain muscle group. Such myoelectric electrodes (see Fig. 1) are connected to the respective muscle nodes using a conductive gel.

It is very important to realise the true connection circuit - ground of the probes must be connected to a human node lacking any muscles. It should be in very short distance to the point where registration of signal occurs - see Fig. 1.

Each time the electrodes are mounted, the difference in potential changes due to different mounting points being chosen, expendable electrodes may have different conductivity and differently accumulated electric charge can influence the registration of the signals.

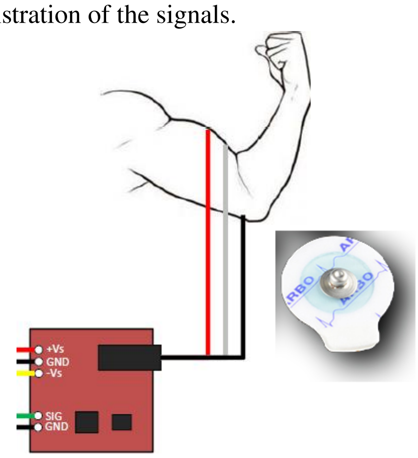

Fig. 1. Myoelectric expendable electrode (right) and basic myoelectric probing circuit. Two probe wires are connected to the beginning and the end of the muscle. The ground wire is connected to the elbow.

[a]Corresponding author, email: vadim.gerasimov@ateities.lt

Each signal is noisy. The noise is particularly powerful when the muscles are in motion while being probed. To reduce the noise, filters and an amplifier could be used. However, it could be expected that the ANN can average the noise as well.

It is insufficient to amplify the signals, they need to be processed, formatted and sent into the further processing node. Such node could be mobile like all the other nodes of the system. Then it is necessary to know how to implement the algorithm suited for such a mobile node - an embedded system. An open ANN library, viable with an embedded system, such as Raspberry Pi could be used.

The ANN architecture that would adapt itself to the existing calibrating data, and at the same time would not be a burden for the embedded system in terms of calculation power should be chosen.

Myoelectric signal recognition works were carried out long time ago. In 1990, Kelly et al [1] attempted to recognise myoelectric signals for prostheses control. At that time it was performed on a serial memory computer, and the processing power did not grant performance to solve the fundamental task. The method that was used in that work was the Hopfield network, suited for time dependent signal registration. That network uses a closed loop method for calculating the synaptic parameters. As stated in Ref. [1], classification was successful using *Sequential Least Square* algorithm.

For a similar purpose, A. Soares et al [2] tried to use *Multilayer Perceptron Backpropagation Feedforward Network* to recognise and classify the myoelectric signal in groups; what the signal was originated from and to control virtual prostheses. Network was comprised of at least 80 neurons. The tests were successful.

Electromyogram. The muscle signal in time distribution (potential time-resolved dependency) was titled as *Electromyogram* (EMG). The central nervous system, comprised of the spine cord and peripheral nerves, controls the actions of muscle fibres, which basically determines movements. The muscles consist of special cells, which are able to contract and relax. These actions are stimulated by the motor elements - neurons.

The EMG can be probed in two ways: a) surface mounted EMG (SEMG), when the electrodes are attached to the skin surface; and b) less popular intramuscular EMG - a needle shaped electrode is injected into the muscle. The former variant of electrodes was chosen due to low invasiveness. The SEMG measures muscle fibre electric potential difference for one motor unit. Such potential differences are called motor unit action potentials (MUAP). The potential difference is around 100 mV, but because of the fibre connecting layers and skin, the SEMG is a complex signal with very small amplitude (from μV up to 5 mV). Fig. 2 represents the typical SEMG as a EMG from skin surface.

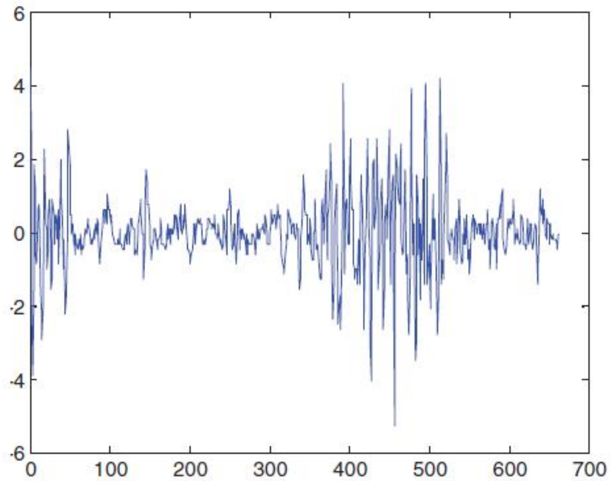

Fig. 2. Typical SEMG as a EMG from skin surface. Abscisse x represents the index of set, ordinate y - signal potential. Adapted according to Ref. [3].

2. Algorythms and learning processes

Learning a program or algorithm is a process that gives better performance, while storing information about the environment. Initially the multi-way algorithm could work not ideal due to absence of true conditions. To forecast all situations in life is impossible.

The task of robot in labyrinth is quite suitable for explaining. Let's say, a robot is given motion in a labyrinth. Each new labyrinth does not give any chance to the programmers to foresee each entangled turn; only a learning algorithm would be able to do something like that. The programmers could not expect any changes in a system, for example, changes in market prices that would be good to predict for future. There are such tasks that the programmers themselves do not know how to solve them; instead they rely on the learning algorithm.

Machine learning must have a feedback; otherwise the mechanism will not know what is *true* or *false*. There are several machine learning feedback types.

1. Unsupervised learning. At this point a machine or a robot must learn the structure of the given problem. These tasks are usually solved by data clustering or genetic algorithms. It means that the algorithm must choose the correct input data by itself. Suppose, if a robot-driver was let in town, it would have to know days, jammed by traffic, and free of it.

2. Reinforced learning. In this case the program must get "encouragement" or "penalty" for the results. This means, should a robot-driver hit a pillar, give him the *bad* signal, or give a *good* signal if the pillar was successfully bypassed. Which actions induced the encouragement or penalty is to be determined by the algorithm.

3. Supervised learning. Certain actions are provided to the programme taking into consideration sensory input data. Simply imagine, that the instructor dictates, where to turn the wheel, when the robot-driver drives up to the curb.

Table 1. Learning data.
Signal potential value from mioelectric sensor at different time (t_1,t_2,t_3,t_4,) and corresponding shift d

t_1,	t_2,	t_3,	t_4,	d, cm
0,07	0,09	0,10	0,09	28
0,08	0,11	0,14	0,10	46
0,08	0,10	0,11	0,11	25
0,06	0,09	0,11	0,07	36
0,07	0,08	0,1	0,07	49

Supervised Learning plays the most important role in the artificial intelligence systems. Also supervised learning is an objective of this work.

Suppose, sensors give some input data, which will be denoted as matrix **X**. Its columns are of a separate myoelectric sensor input data, and rows represent a separate sample. Logically though, one needs a **Y** matrix, the columns of which are the decision data set (that or this action), and the rows are the decision sample set. Further, an example that was actually implemented in this work will be provided.

Myoelectric sensors of system produce the time-dependent signals as values x_{tn} for recording. Index t represents the registration time and n - sensor order. Considering the **X**, one needs to know the y_{tn} data, which at the given time t show what decision n should be made. At this point some conditions are being created, which are listed in Table 1.

Definitely, the number of samples of data is too little, it should be several hundreds. The task could be formulated as follows: how certain data values could be related with a continuous function? This is where the approximation algorithm is of great help.

The task of a supervised algorithm is to find such a so called *hypothesis* function H, which can be adapted to table's patterns and other unforeseen results, which is to approximate the actual function $f(x)$. Fig. 3 represents the results of a hypothetic experiment, where scattered two-dimensional data could be estimated as weak-correlated dependency.

To make conclusions about the experiment, it is necessary to summarize the data point distribution, i.e. describe the data with some kind of a function, which would approximate the pattern. Fig. 3 represents also and two possible approximations: linear as well as parabolic.

The error of the applied function could be measured while testing on test data, which were not used while fitting the function, although still connected with the learning data. A well chosen hypothesis function has not only the least error with the learning data, but also with the test data, which means, that given new unknown input data, the function predicts the answer the best.

Sometimes a function $f(x, \cdots)$ could be estimated as stochastic. At this point one needs to know a relative distribution of a function $P(Y|x)$. If the output data are discreet and of finite number, then the task is called *classification*. When the

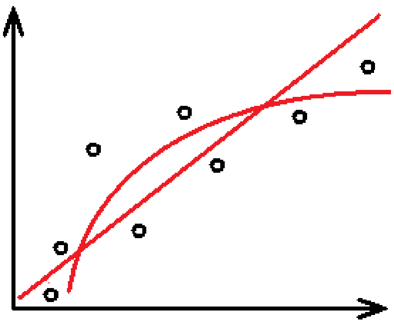

Fig. 3. Typical data of simulated experiment (open dots). Linear and parabolic approximations in red.

data are of *yes/no* type, then the task is called *Boolean classification*. If the output data are continuous, then the task is called *regression*. Solving a regression problem is to find the average expected value of y. Statement that **H** exactly matches the f function is false, the probability is equal to zero.

It is very important then to define the "best approximation". The error rate of a hypothesis is a quantity of $H(x){\neq}y$ occurrences in a (**X**,**Y**) space. A low error rate of an H function does not mean that chosen hypothesis function is good. There is a chance, that the function is precise to the given training data, though not to the test data. This effect is called *overfitting*, which is withdrawn when testing the function with the test data or rather called the validation data, decreasing the function's precision until reaching the least average error with the test data. It is called the cross validation and model selection.

This technique is implemented in the MatLab [4] modules. However, since the chosen instrument happens to be an open library which was installed in the processing unit, it has no validation options and function precision is chosen by default - first order.

4. Linear regression method

Let us formulate the following task [5]: we need to adjust a linear function y to the set of data points:

$$y = \theta_1 \cdot x + \theta_0 \tag{1}$$

Three vectors: **X**, **Y** and **Θ** represent the arguments, values and weight coefficients (also known as the synaptic parameters in the neural networks). Choosing the weight parameters is intended to minimize the overall empirical error. Traditionally that is achieved using the function of difference square L_2:

$$Error(\mathbf{H}_e) = \sum_{j=1}^{N} L_2(y_j, \mathbf{H}_e(x_j)) \tag{2}$$

$$Error(\mathbf{H}_e) = \sum_{j=1}^{N} (y_j - \mathbf{H}_e(x_j))^2 \tag{3}$$

$$Error(\mathbf{H}_e) = \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 \tag{4}$$

This task requires to minimize the function $Error$. First derivative is equal to zero:

$$\frac{\partial}{\partial\Theta_0}\sum_{j=1}^{N}(y_j-(w_{\dot{1}}x_j+w_0))^2=0 \tag{5}$$

$$\frac{\partial}{\partial\Theta_1}\sum_{j=1}^{N}(y_j-(w_{\dot{1}}x_j+w_0))^2=0 \tag{6}$$

The solution of the system is then unique:

$$\Theta_1=\frac{N\cdot(\sum x_{\dot{j}}y_j)-(\sum x_j)\cdot(\sum y_j)}{N(\sum s_j^2)-(\sum x_j)^2} \tag{7}$$

$$\Theta_0=\frac{(\sum y_j-\Theta_1(\sum x_j))}{N} \tag{8}$$

The learning algorithms are based upon the change of the weight coefficients so that the error function is as small as possible. In case of a straight line, the space of weight coefficients is two dimensional. Such function, arguments of which are only two weight coefficients and the value of it is the error, has a concave shape and a global minimum. The successful choosing of such a minimum terminates the learning in terms of a linear regression. Choosing other functions, far more complex than a mere straight line, may render issues with representing the weight coefficients. In this situation an overall optimization problem must be solved altogether.

To do that, hill leap or a gradient descent algorithm is used, while tracking the gradient value in the given point. Starting from any parameter set, and:

$$\Theta_1 \leftarrow \Theta_1 - \alpha\frac{\partial}{\partial\Theta_i}Error(\mathbf{\Theta}) \tag{9}$$

Here α represents so called learning rate or size of the step. It can be constant or change in time. Let's express the hypothesis function $\mathbf{H}$ as the multiparametric function:

$$\mathbf{H}_{sw}(\mathbf{x}_j)=\Theta_0+w_1\cdot x_{j,i}+\cdots+w_n\cdot x_{j,n} \tag{10}$$

Here $x_{j,0}$ is equal to one. Such function can be represented as a product of two matrices:

$$\mathbf{H}_{sw}=\mathbf{\Theta}\cdot\mathbf{X}_j=\sum_i\Theta_i\cdot x_{j,i} \tag{11}$$

Hence the solution can be found not only in the gradient proximity, but with an equation:

$$\mathbf{\Theta}^*=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{12}$$

5. Classification and logistic regression

Linear functions can also be used for classification. Seismic data task is very suitable for such purposes. Let x and y are seismic data as bulk and surface wave amplitudes. The probability of an earthquake or an underground explosion depends on these two data sets, which, in turn, are of great interest to seismologists or military experts.

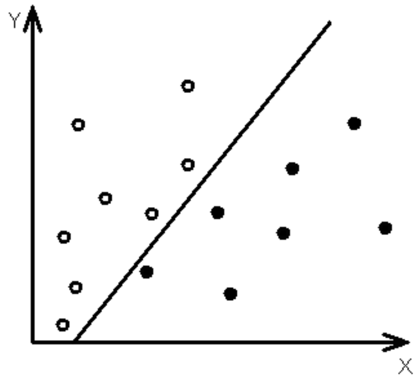

Fig 4. Typical seismic data (open dots). Straight line separates data groups into classes.

For example, lets determine the quake value as 0 and explosions as 1. Fig. 4 represents data distribution, where separation of data groups takes place.

The decision boundary is a line (or a surface in a multidimensional space) which separates two data classes. This boundary is called the linear separator. This time the equation can be inexplicit :

$$\theta_0+\theta_1\cdot x-y=0 \tag{13}$$

Hence the classification condition can be formulated as follow:

$$\mathbf{H}_\Theta=1, \quad if \quad (\mathbf{\Theta})\cdot\mathbf{X}\geqslant 0 \tag{14}$$

$$\mathbf{H}_\Theta=0, \quad if \quad (\mathbf{\Theta})\cdot\mathbf{X}<0 \tag{15}$$

Similar to the threshold function algorithm shown below is called the perceptron learning rule:

$$\Theta_i \leftarrow \Theta_i+\alpha(y-\mathbf{H}_\Theta(\mathbf{X}))\times x_i \tag{16}$$

Sometimes it is inconvenient to classify using threshold functions, when a continuous traverse over the separator is needed. Also the threshold function cannot be differentiated at the separator (because a derivative of a sudden leap function is delta function). For that purpose a continuous differentiable sigmoidal function was created:

$$\sigma(z)=\frac{1}{1+\exp(-z)} \tag{17}$$

Second order sigmoid containing linear regression function could be used for validation and regularization purposes:

$$\mathbf{H}_\Theta(\mathbf{X})=\sigma(\mathbf{\Theta}\cdot\mathbf{X})=\frac{1}{1+\exp(-\Theta\cdot X)} \tag{18}$$

Sigmoid and the threshold function are presented in Fig. 5. The sigmoid weight coefficient selection for error minimization is called the logistic regression. This problem cannot be solved analytically; however it can be solved using the "gradient descent". Eqns. (19-20) represent the mentioned algorithm.

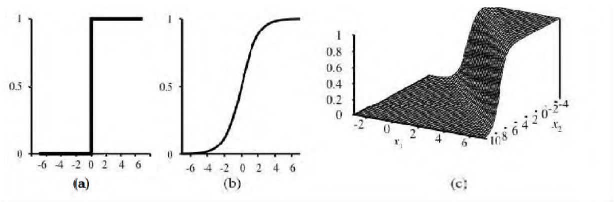

Fig 5. Different threshold functions: a) 1-D threshold function; b) 1-D sigmoid function; c) 2-D sigmoid function, suitable for data approximation presented in Fig. 4. Adapted according to Ref. [3].

$$\frac{\partial}{\partial \Theta_i} Error(\mathbf{\Theta}) = \frac{\partial}{\partial \mathbf{\Theta}_i} (\mathbf{y} - \mathbf{H}_{\Theta}(\mathbf{X}))^2 \quad (19)$$

$$\frac{\partial}{\partial \Theta_i} Error(\mathbf{\Theta}) = 2 \cdot (\mathbf{y} - \mathbf{H}_{\Theta}(\mathbf{X})) \times \mathbf{H}' \times x_i \quad (20)$$

The derivative of sigmoid function H satisfies the given condition:

$$\mathbf{H}'(z) = \mathbf{H}(z)(1 - \mathbf{H}(z)) \quad (21)$$

hence the error minimization procesure was done by following expression:

$$\Theta_i \leftarrow \Theta_i + \alpha(y - \mathbf{H}_{\Theta}(\mathbf{X})) \times \mathbf{H}_{\Theta}(\mathbf{X})(1 - \mathbf{H}_{\Theta}(\mathbf{X})) \times x_i \quad (22)$$

6. Neural networks

The term *neural networks* is derived from the brain activity research fields; however, the only thing they have in common is the idea and the components' name. The artificial neuron is comprised of several parts, including input node, working node (input function, bias weight, activation function) and output node (dendrite) - see Fig. 6.

The artificial neuron is basically a way the regression or the classification functions and their parts are represented - it is a concept. To be able to talk about *Artificial Neuron Network* (ANN), they should be connected into an entity - a certain topology. Neuron i is connected to the neuron k, so that the activation function a_i would be passed from i node to j node. Each node has its own weight parameter $\theta_{i,j}$ which determines the connection status. Additionally the neuron has also the bias weight node a_0=1 with a related weight parameter $\theta_{0,j}$. First, each j node calculates its input sum:

$$in_j = \sum_{i=0}^{n} \theta_{i,j} \cdot a_i \quad (23)$$

And then it interposes it into the activation function - usually sigmoid:

$$a_j = g(in_j) = g \cdot (\sum_{i=0}^{n} \theta_{i,j} \cdot a_i) \quad (24)$$

The ANN can be divided into two big topological groups: one way sequential and recursive (with a feedback) transfer. One way sequential network is grouped by layers.

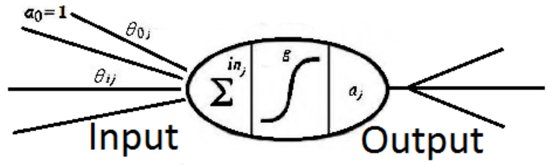

Fig. 6. Diagram of a neuron. Adapted according to Ref. [4].

During the execution mode each a layer receives the data only from the previous layer, and accordingly transfers only to the next layer. Fig. 7 represents the typical ANN diagram.

The input layer consists of the input information neurons, which receive the data directly or reduced from the sensors.

The intermediate or "hidden" layers are the main processing neurons, which contribute to the overall hypothetic function shape. The output neurons retrieve the data to the execution nodes directly or in a reduced form, for example, to an electromechanical system. Each argument of the neuron activation function stems from the previous neuron activation function value.

A single hidden layer can represent any continuous function to a certain precision and adding the second hidden layer enables the network to represent discontinuous functions. Adding more layers may represent any other kind of functions, though the working mechanism becomes rather difficult to understand. Due to it, choosing the ANN architecture demands model experimental tests - selecting such a network that would satisfy the objective conditions.

The task of the selected network is to find such a hypothetic function, which would give the needed feedback after taking certain inputs. In other words, the network training is a convergence of the hypothetic function to the real function, which in turn describes the pattern. To converge the function one needs to calculate the error in the output neuron, dependent on the synaptic parameter Θ:

$$\frac{\partial}{\partial \Theta} Error(\mathbf{\Theta}) = \frac{\partial}{\partial \Theta} [\mathbf{Y} - \mathbf{H}_{\mathbf{\Theta}}(\mathbf{X})] \quad (25)$$

$$\frac{\partial}{\partial \Theta} Error(\mathbf{\Theta}) = \frac{\partial}{\partial \Theta} \sum_{k} (y_k - a_k)^2 \quad (26)$$

$$\frac{\partial}{\partial \Theta} Error(\mathbf{\Theta}) = \sum_{k} \frac{\partial}{\partial \Theta} (y_k - a_k)^2 \quad (27)$$

Here Y represents the output data vector, which shows a single exemplar output datum, comprised of several parts; for example, if probed muscles are stretched - turn the motors with a certain speed, which is demanded by the controller. **H** represents the hypothetic function vector, parts of which evaluate the output neuron components separately (e.g. for each sensor). Θ represents the synaptic parameter matrix, the elements of which evaluate the components of the previous neurons and the output neurons.

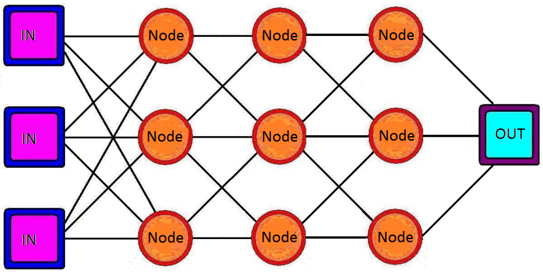

Fig. 7. ANN system based on three hidden layers.

The k index determines the number of the output neurons, which indicates how many controllable parameters are there, e.g. motor power or actuation. The calculated and minimized error in the equation is written only in terms of the output layer.

For the network to converge, it is necessary to transfer the calculated error from the output layer back to the hidden layers, down up to the input layer. This algorithm is called the back propagation algorithm.

Let Err_k is a component of the $(\mathbf{Y} - \mathbf{H})$ vector. Then we describe the modified error:

$$\Delta_k = Err_k \times g^{'}(in_k) \tag{28}$$

where

$$g^{'} = \frac{d}{dx}[\frac{1}{1 + \exp(in_k)}] \tag{29}$$

Hence the weight coefficient selection (or simply learning) rule is as follows:

$$\theta_{j,k} = \theta_{j,k} + \alpha \times a_j \times \Delta_k \tag{30}$$

The hidden layer j node function component error is calculated with the following equation:

$$\Delta_j = g^{'}(in_j) \sum_k \Theta_{j,k} \Delta_k \tag{31}$$

Briefly, the back propagation learning algorithm is in fact the minimization of the error between the output data and the calculated neural network output data in each layer where each node minimizes the error depending on the error of the j node, which in turn is the next after the i node. Mathematically the error calculation in each of the nodes and its transfer look like this:

$$\frac{\partial Error_k}{\partial \Theta_{i,j}} = -2(y_k - a_k)\frac{\partial a_k}{\partial \Theta_{i,j}} = -2(y_k - a_k)\frac{\partial g(in_k)}{\partial \Theta_{i,j}} \tag{32}$$

$$\frac{\partial Error_k}{\partial \Theta_{i,j}} = -2\Delta_k \Theta_{j,k} g^{'}(in_j) a_i = -a_i \Delta_j \tag{33}$$

Here Δ_j was described above, and the equation ultimately means, that the point of minimizing the error is to transfer the minimized error back into the next layer - actually the previously standing layer's nodes.

The neural network execution mode is comprised of data transfers into the input layer, and the calculation of the reactive data with the converged hypothetic function.

Conclusions

General principles for creating the electronic system devoted for processing and recognition of myoelectric signals were observed.

References

1. Micheale F. Kelly, Philip A. Parker, Robert N. Scott. The Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study. – *IEEE Tran. Biomed. Eng.* 37(3) (1990) 221-230.
2. Alcimar Soares, Adriano Andrade, Edgard Lamounier, Renato Carrijo. The Development of a Virtual Myoelectric Prosthesis Controlled by an EMG pattern Recognition System Based on Neural Networks. – *Jour. Intel. Info. Sys* 21(2) (2003) 127-141.
3. Ramaswamy Palaniappan. Biological Signal Analysis. – Ramaswamy Palaniappan Ventus Publishing, 2010.
4. <https://se.mathworks.com/help/nnet/ug/multilayer-neural-networks-and-backpropagation-training.html?requestedDomain=se.mathworks.com>, accessed 2014.06.25.
5. Stuart J. Russel, Peter Norvig. Artificial Intelligence. A Modern Approach. 3rd ed.. – Publ. Prentice Hall, 2010 – P. 1134.

Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 2. Hardware, software, interfaces

Vadim Gerasimov [1 a] , Gintaras Jonaitis [2], Vytautas Jonkus [1]
[1] Machine-to-Machine laboratory, Department of Radiophysics,
Faculty of Physics, University of Vilnius, Saulėtekio al. 9-III, LT2054 Vilnius, Lithuania
[2] Laboratory for Medical Rehabilitative and Assistive Technologies, Department of Biomechanics,
Faculty of Mechanics, Vilnius Gediminas Technical University, J. Basanavičiaus str. 28, Vilnius, Lithuania

Received 12 June 2015, accepted 5 August 2015

Abstract. Design of electronic system devoted for processing and recognition of myoelectric signals were observed. Description of used embedded systems were presented and analyzed.

Citations: Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 2. Hardware, software, interfaces – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 13-17.

Keywords: Artificial Neural Networks; ANN; recognition of myoelectric signals; electromyogram; linear regression method; supervised learning; logistic regression; artificial neuron.
Short title: Artificial Neural Networks. 2. Hardware, software

Introduction

Previous publication [1] represents literature overview - how to create the electronic system devoted for processing and recognition of myoelectric signals. Artificial Neural Networks (ANN) algorithm was described as an electronic model based on the neural structure of the human brain.

The hardware needed for scanning and amplification of myolectic signal was provided by Laboratory for Medical Rehabilitative and Assistive Technologies at Vilnius Gediminas Technical University. The hardware construction, assembly and testing was carried out at Machine to Machine Lab at Vilnius University in 2014÷2015. Such signal decoding and interpretation could be used in prostheses technology and other human-machine interface.

Work objectives could be formulated as follows:
a) to design an interface, suited for myoelectric signal recognition - from human body mounted sensors;
b) to construct a robot-manipulator controllable by afore mentioned interface.

1. Embedded systems

The *embedded system* represents a computer system, the purpose of which is precisely determined in a mechanical or an electronic system. Often it is equipped with an operating system (OS), which can be embedded or real-time.

The difference between them in execution manner is that the former, embedded system buffers its tasks [2], usually it has its own command interpreter and additional applications such as the Internet browser can be installed. The latter is real-time system; the performance of its tasks is carried out as soon as possible and the tasks are started not from command line or user interface, but programmed before booting the system. An embedded system may even be somewhat similar to that of a normal PC system [2]. Some embedded systems are able to compile the executable code inside without the use of external equipment.

Physically an embedded system can be used for different purposes - from digital clocks and music players, to industrial objects - traffic lights and production lines. Its user interface can be graphical (GUI) with various controls, console prompt, or without any control. A system with GUI may be connected to an external monitor or an embedded miniature screen. Most frequently used processors are RISC processors (Reduced Instruction Set Computing). These processors can be with an integrated memory or with external circuits, suited for the operational memory access. The number of cores can be from one to several, e.g. Raspberry Pi B and Pi 2 differ in core number. The second model may utilize its four cores for parallel computing software.

An embedded system may also be designed with a periph-

[a]Corresponding author, email: vadim.gerasimov@ateities.lt

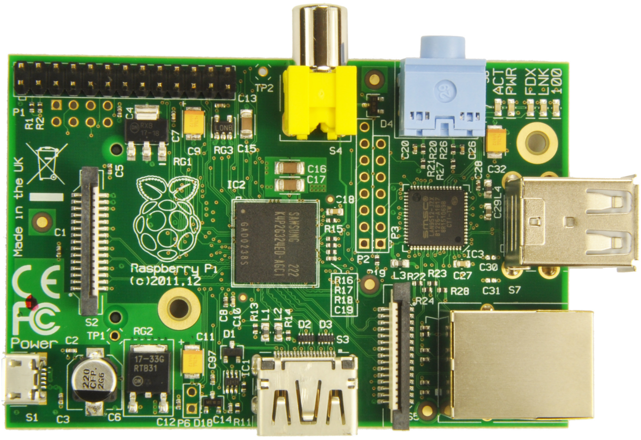

Fig. 1. Raspberry Pi B board. According to Ref. [4].

ery: asynchronous transmission line (RS-232, RS-422), synchronous transmission line (I2C, SPI), USB, card reader, network interface, timers, discreet/digital signal pins, analogue/digital converters. [3]

Embedded system software. The embedded system control can be implemented in several ways [3].

1. *Program loop.* It calls subprograms and executes them sequentially.
2. *Interrupt-controlled tasks.* A timer or a com-line after receiving control bytes switches tasks from one to another.
3. *Cooperative parallel task distribution.* The method is similar to a loop mechanism, but here the tasks are given their own environment for execution. Thus, when a task execution is no longer needed, it calls a routine process "PAUSE", which halts the execution.
4. *A multithread operating system.* Such OS contains a kernel, which assists tasks in switching, using priorities or semaphores. This is called a real-time OS.
5. *Micro-kernels.* This is a logical step forwards in comparison to a real-time OS. The difference is that the kernel allocates the memory and switches the processor from one task to another.
6. *Monolithic kernels.* This is a higher level, when the kernels have complicated control systems. Such a system can be of a Linux or a Windows basis, therefore they need high performance hardware (higher than just any ordinary microcontroller). These embedded systems may have separate drivers or programme packages.

There are different choices of embedded Linux OS distributions, intended for different processor architectures, for example: ARM, AVR32, MIPS, Xtensa. The possibilities of an OS depend on the distribution. For example, "OpenWRT" is intended for network switches or routers, "Debian" and "Ubuntu" based Linux systems more versatile, "Arch" is suited for a certain, fully programmable task to be executed.

Embedded system board - Raspberry Pi B The reason why this variant of an embedded system was chosen is its growing popularity. There were different attempts to use the basis of this board in constructing radio transmitters, control cores, or other probe nexuses. The Raspberry Pi B illustration is shown in Fig. 1.

The board is equipped with integrated circuits, suited for HDMI, SD card, USB, Ethernet RJ45 or audio connections control. The board also has a small battery of 26 digital pins for discreet digital functions or transition lines functionality of which covers, but is not limited to digital signalling, for example UART, SPI, IIC. Different equipment can be connected to the board: video cameras, GSM modules, sensors. The board is designed to utilize the Broadcom manufactured ARM11 (ARMv6) processor - the BCM2835.

BCM2835 processor. The Broadcom manufactured processor - BCM2835 utilizes the ARM11 architecture for the main processing node (with the ARMv6 instruction set), and Broadcom VideoCore IV for the graphical processing node [5]. Its technical specification are presented below. Second level 128 KB Cache memory module; 700 MHz CPU clock speed; High Definition and multichannel video processing supporting the most popular codec's; Embedded HDMI, NTSC, PAL, VGA interfaces; 256 MB SDRAM memory; DDR NAND Flash memory USB 2.0; SD Card 3.0; Ethernet 10/100 Mbps controller; Audio interface; SPI, UART, I2C; Is able to support Linux OS'es: Raspbian, Pidora, RISC OS, Arch, FreeBSD, OpenWrt, SlackWare;

Embedded system board - Raspberry Pi 2. The newest embedded system board was used for comparison and the mechanism control implementation (see Fig. 2). The second Pi model board (like the Pi B model) also has similar integrated circuits. Among those things it is also equipped with a bigger - 40 pin battery and four core processor ARM-Cortex A7 (ARMv7) BCM2836.

BCM2836 processor. The Broadcom's BCM2836 processor uses an ARM A7 architecture for the CPU (with the ARMv7 instruction set) [7]. Its additional specifications are (save for the basic BCM2835 ones): 900 MHz clock frequency; Only HDMI interface; 1 GB LPDDR2 SDRAM memory; Able to additionally support OS'es like: Android, Ubuntu, Windows 10.

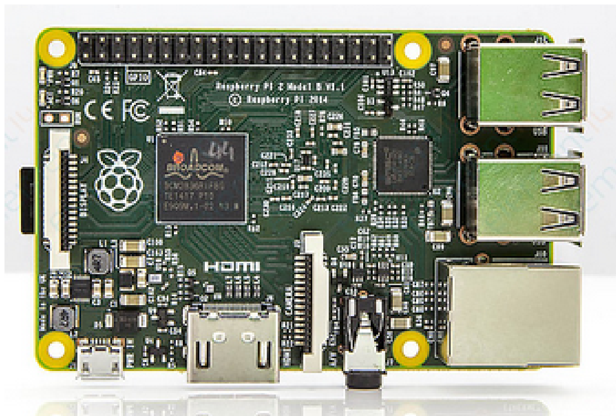

Fig. 2. Raspberry Pi 2 board. According to Ref. [6].

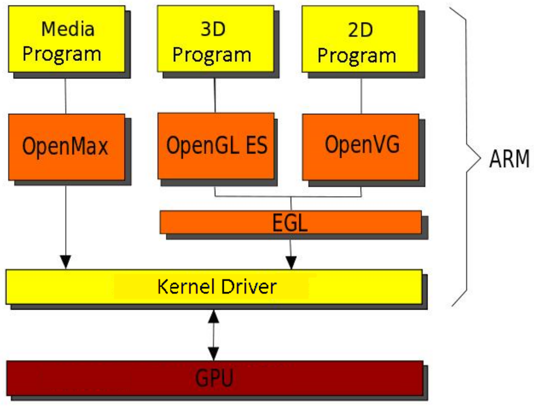

Fig. 3. Raspberry Pi software hierarchy diagram. Adapted according to Ref. [8].

2. Linux embedded operating system

Many operating systems were released for the Raspberry PI to fulfil different tasks; the most popular were listed above. Linux Raspbian OS was used in this work. This system is created in compliance with the GNU GPL licence. Its distribution stems from its progenitor - Linux Debian [9]. Work in the operating system may be carried out via the internet or/and the external monitor.

Writing and compiling of a code is possible on the board itself, which is very convenient - there is no need for external computer resources that should assist in constructing the code before porting it to the board. The compilation is performed like on a simple personal computer, entering a command in the prompt terminal: gcc (for C language), g++ (for C++ language) with the source code file names, object file names and an object connection key "-o". The Raspbian system even supports other *Integrated Development Environments* (IDE), offering a better comfort for programming, one of the IDE examples could be "CodeBlocks".

However, additional libraries, like the <math.h> and other periphery connecting libraries, like the <wiringPi.h> are reached using special terminal keys -lm and -lwiringPi. The peripheral library must be installed separately, and may also differ for different operating systems. The principle diagram of the system, its drivers and application software are illustrated in different layers in Fig. 3.

3. Parallel programming

To control several nodes at the same time, it is necessary to write a parallel programme. Parallel task execution on a system usually commences independently and asynchronously (unless it is a distributed calculation) - unlike in a sequential programme. The operating system, which is installed in an embedded board, takes care of the parallel programme execution to not hinder each other. Each separate programme is a different independent process, which is executed in allocated time on one of the processor's cores. However, one process can be decomposed into separate parallel nodes. Such coordinated operation (avoiding conflicts in data exchange) can be implemented in several ways [10].

1. Interaction via shared memory. Each processor core commences thread execution, which in turn belongs to a single process. Threads exchange their data via shared memory, allocated and common for the given process. This is implemented by using either the features of the programming language itself (like Java or C#), or using the help of libraries (such as used in this work - Posix PThreads for C language), or declaratively (using OpenMP library), either using the embedded compiler tools (like the High Performance Fortran). Such thread implementation demands additional control in thread interaction - Mutex'es, semaphores or monitors.

2. Interaction between thread using message passing. Each processor launches one thread, which exchanges messages with other threads that are executing on other processors. Implementation of such interaction is possible using MPI libraries or language features (like High performance Fortran, Erlang, Occam). Messages can be passed assynchronically or using a *rendezvous* method, when the sender is blocked until its messages are received by another thread.

In this work a library called Posix PThread was utilized for several reasons. Firstly, it is native in the Linux system. Secondly, its thread control is moderately explicit, comparing with an OpenMP, although not as much as in MPI, where one has to take care of each thread's message passing [14]. It is worth mentioning, that launching a few threads, which share variables, it is necessary to utilize mutexes, semaphores and monitors.

1. Mutex. An object/structure, which can be locked in one of the threads, before executing operations with shared variables, and unlocking them afterwards. In another thread, where the same mutex is attempted to be locked, the process halts (optionally) until the wanted mutex is unlocked by the different occupied thread.

2. Semaphore. A mechanism, working just like a mutex, which also has a quota for the variables - one can lock it or unlock it several times. In such case a variable can be reached by multiple threads, as many as is permitted by the semaphore. The set quota for one thread is, in fact, a semaphore's substitution for a mutex.

3. Monitor. An observer which lets the threads not only to wait for their turn to execute, but supplies additional conditions, as well as informs other threads of the changed statuses.

4. Equipment construction

4.1. Tools for probing myoelectric signal

The probing experiment was carried out using electrodes as described in Ref. [11]. These cables are protected from mechanical stresses and breaking, they are made of an nonsol-

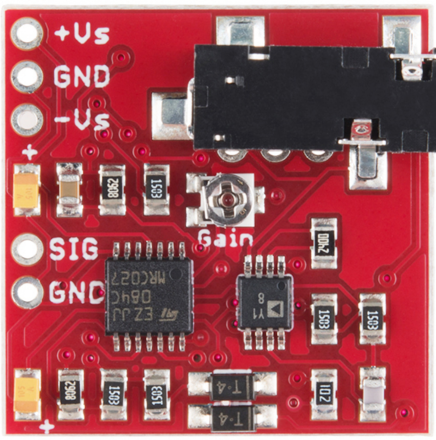

Fig. 4. Muscle Sensor v3 model muscle signal amplifier (manufacturer Advancer Technologies). Adapted according to Ref. [11].

derable metal and are relatively expensive. Disposable mounting stickers are prepared with a conductive gel. They are moulded on the sides of the measured muscle nodes, and the black ground wire on a bony area near the muscles. Muscle signal amplification takes place in a special precision amplifier circuit - see Fig. 3.

The principle of its operation is based on a differential amplifier AD8226 to strengthen microvolt level signals up to a couple of hundred millivolts or even 2÷3 V. The amplified signal is then inverted to the positive voltage, using the instrumental operational amplifier - TL084 model.

After the signal is being smoothed using another circuit node of TL084 model the signal received in such way is not just any set of modulated sinusoids, but rather a curve, which depends on the muscle tension intensity. If need be, the signal may be amplified or weakened at the second-stage amplifier before sending it to another node. A major drawback of this scheme is the implementation of the power supply - it must be powered by a three-pole power supply or a battery (with the plus, minus and ground terminals).

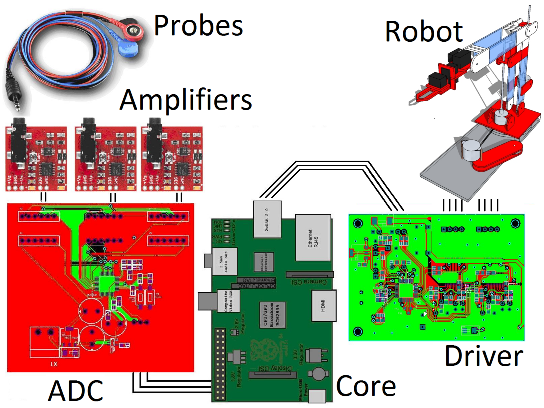

Fig. 5. The basic diagram of the system.

4.2. Main board construction

In order to scan a multi-channel signal, it was decided to construct a main board with a 32-bit ARM Cortex M3 processor (LPC1316). It was chosen because it has a number of pins connected to a 12 bit analogue-to-digital converter, as well as UART (Universal Asynchronous Receiver Transmitter) interface. The final layout is able to support up to six amplifier modules - see Fig. 5, left bottom part. It works as follows: every fifty milliseconds the signal is registered from the amplifier, converted into digital information and forwarded through the UART interface to other nodes. Transmission is constant, regardless of receiving node. Receiving node is described in the theory section - embedded systems Raspberry Pi B and Raspberry Pi 2.

4.3 Assistive mechanical robot

For the research of mechanism control a robot-manipulator E.A.R.L (Ergonomic Assistive Robotic Limb) was created - see Fig. 6. Its parts were manufactured using a laboratory milling machine. The vertical axis rotates due to belt transmission. Shoulder and elbow joints are driven using a worm-gear. These gear systems are powered using stepper motors, while the wrist actuation is handled by servo motors.

To control and drive the stepper and servo motors printed circuit boards were constructed. The operating mode of the above mentioned boards is selected using the UART interface. The operation commands are sent from the embedded system board to the stepper driver boards.

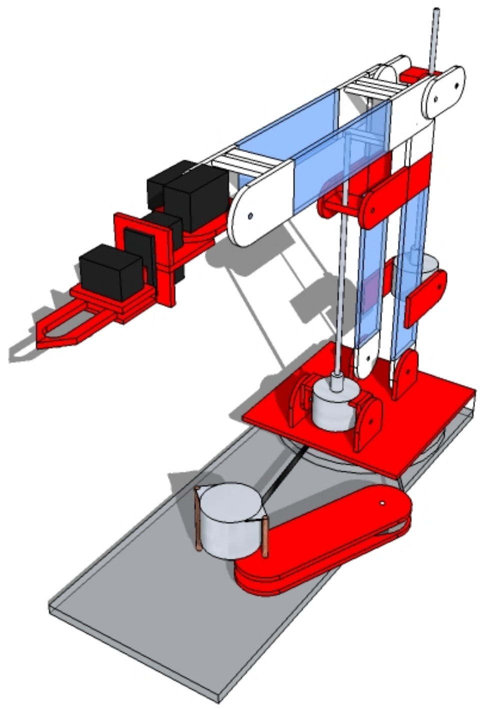

Fig. 6. Prototype of robot manipulator E.A.R.L. which was used to test the neural network control performance.

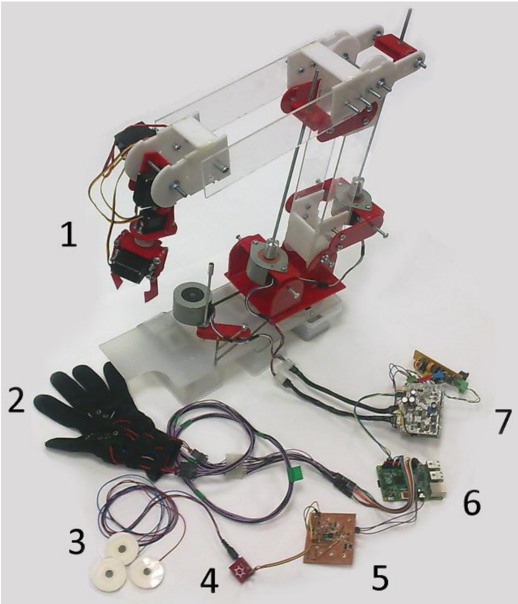

Fig. 7. Complete robotised system including: 1) robot manipulator; 2) remote control glove; 3) myoelectric probes; 4) amplifier; 5) signal converter; 6) Raspberry Pi 2 embedded system; 7) motor drive board.

Additional control is carried out with an ergonomic glove fitted with buttons and if pushed they signal the system.

Combining all the components into a single system, the functioning machine was developed as a hardware-software sollution (Fig. 5) E.A.R.L.

This system consists of muscle signal probes - stickers and the cables. They are connected to the muscle signal amplifier. The amplifier enhances about 50 thousand times, inverts and smothens the signal and feeds it into the distribution unit (main board). The distribution unit sends the data every 50 milliseconds using a command "ADC#XXXX", where # is the channel number and $XXXX$ represents the value from zero to 4095.

The embedded system, using the UART interface, receives and decodes incoming messages using one of the software threads, and depending on the selected programme mode, makes decisions. The remote control unit communicates directly with the embedded system. Its desicions are displayed either on the screen (in case of muscle signal decryption mode) or are sent to the motor control board (in case of robot control mode). The motor control board using a special processor and the UART interface accepts commands from the embedded system and using special driving circuits controls the voltage applied on the stepper motor coils, or the servo motor feedback system.

Fig. 7 represents complete robotised system including robot manipulator, remote control glove and the described electronic equipment.

Conclusions

1. The designed robotic mechanism E.A.R.L. (Ergonomic Assistive Robotic Limb) was succesfully created using several interfaces, suited for myoelectric signal recognition.
2. Package of monitoring program was created, adjusted and adapted for presented interfaces.

References

1. Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 1. Problem overview – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 7-12.
2. Michael Barr, Anthony Massa. Programming Embedded Systems, 2nd Edition with C and GNU Development Tools. – O'Reilly Media, 2006 – P 336.
3. <http://users.ece.cmu.edu/k̃oopman/iccd96/iccd96.html>, accessed 2015.01.17.
4. <http://elinux.org/RPi_Hardware>, accessed 2015.01.17.
5. <https://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>, accessed 2015.05.16.
6. <http://www.element14.com/community/community/raspberry-pi/raspberrypi2 ?ICID=rpimain-feature-products>, accessed 2015.05.16.
7. <http://www.adafruit.com/pdfs/raspberrypi2modelb.pdf>, accessed 2015.05.16.
8. <https://www.raspberrypi.org/wp-content/uploads/2012/01/RaspberryArch.jpg>, accessed 2015.05.16.
9. <https://www.raspbian.org>
10. Peter S. Pacheco. An Introduction to Parallel Programming. – Publ. Morgan Kaufmann, 2011 – p. 370.
11. <http://www.advancertechnologies.com/p/muscle-sensor-v3.html>, accessed 2015.01.17.

Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 3. Implementation of movement

Vadim Gerasimov [1] [a] , Gintaras Jonaitis [2], Vytautas Jonkus [1]
[1] Machine-to-Machine laboratory, Department of Radiophysics,
Faculty of Physics, University of Vilnius, Saulėtekio al. 9-III, LT2054 Vilnius, Lithuania
[2] Laboratory for Medical Rehabilitative and Assistive Technologies, Department of Biomechanics,
Faculty of Mechanics, Vilnius Gediminas Technical University, J. Basanavičiaus str. 28, Vilnius, Lithuania

Received 12 June 2015, accepted 25 August 2015

Abstract. Control program for monitoring the electronic system devoted for processing and recognition of myoelectric signals was described. Usage of Artificial Neural Networks (ANN) was implemented as an main part of decision support system.

Citations: Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 3. Implementation of movement – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 18-22.

Keywords: Artificial Neural Networks; ANN; recognition of myoelectric signals; electromyogram; linear regression method; supervised learning; logistic regression; artificial neuron.
Short title: Artificial Neural Networks. 3. Movement implementation.

Introduction

Previous publications [1] and [2] are devoted for describing of general principles for creating the electronic system devoted for processing and recognition of myoelectric signals and to design an interface, suited for myoelectric signal recognition - from human body mounted sensors.

This work is devoted for describing the software algorithm and the proceeding of the experiment of muscle task.

Muscle signal interpretation (arm motion and load) has been tested to train the device to understand how far a hand moved and what load was exerted on it. The algorithm is implemented on the Raspberry Pi B. Since moving the hand or any other muscle, results in the difference of electrical potential it may be expected that the smoothed signal will be proportional to the force generated by the muscles. It may also be expected that integrating the value of force twice regarding time it is possible to obtain a coordinate. The fact is that a muscle may be strained without moving the hand or accidentally, unintentionally stretching it (or due to seizures).

Work objectives could be formulated as follows:
a) to establish the optimal neural network architecture;
b) to create the code of a control program containing task decision module operating by means of an Artificial Neural Networks (ANN);
c) to describe the conditions of training;
d) to estimate the neural network convergence.

1. Algorithm of supervized learning

Since it is difficult to imagine what an integated function or its coefficients would look like, it is better to trust the neural networks to take on the calculation. Then it is necessary to take care which of the neural network architectures are best to use. Since it was decided to use a supervised learning, the *Feedforward Backpropagation* (BP-FF) architecture was selected as most suitable neural network model. This is a traditional architecture, in which the data is calculated from the starting node of the network to the ending in the execution mode and vice versa in training mode (detailed description in previous Ref. [1]).

The task was formulated as follow. As, for example, the hand moves bending at the elbow, starting with zero coordinate - see Fig. 1. About 20 measurements are registered every hundredth milliseconds, number of input neurons is equal to 20. The estimated coordinate should be output in the last neuron network layer the number of comprising neurons of which is obviously - one. The number of neurons in the hidden layer and the number of the hidden layers themselves were adjusted from one to ten neurons and from one to three layers respectively.

[a]Corresponding author, email: vadim.gerasimov@ateities.lt

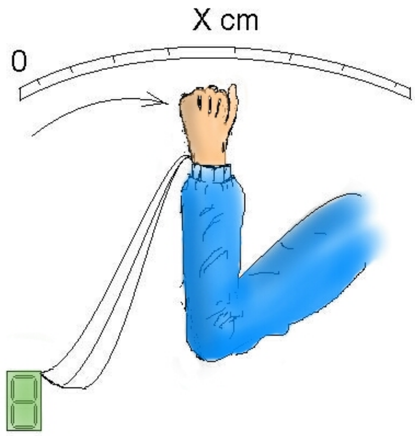

Fig. 1. Schema of biomechanical experiment. The data are entered and displayed on the terminal screen - digital indicator .

The other parameters like transfer functions have been chosen traditionally - symmetric sigmoid function in the middle layer and linear for fringe layers. Since all the data are reduced to [-1÷1] the accuracy was selected relative - 0.5%. Due to that the network convergence occurs within 30÷100 epochs - depending on the training sample number 10 or 30. Such small number of samples is due to the fact that all the data were submitted by moving the hand along the ruller and correcting it with a keyboard, rather than from prepared tables.

The hand weight load experiment was a similar task, though the arm muscle group needed to be strained using weight, or scales expecting the result to be output on the screen.

Writing a neural network algorithm from a scratch and making it functional is a very long and painstaking work, which may take a several years. Therefore, it was chosen to use the neural network library FANN (Fast Artificial Neural Network) [3], developed in Danmark in 2003. FANN is cross-platform library realized in twenty languages.

2. The muscle signal interpretation program

Muscle signal interpretation program was made according to Ref. [4]. Since the embedded system is connected either to the terminal or to the display, at least a terminal-user interface had to be made. The software code written in small blocks which perform a particular function by pressing different keys. This is implemented with the traditional cycle. "While" and the condition "if". Data reception from the UART interface is written using the wiring Pi library. The received message is of "ADC XXX" type, where XXX is a three-digit hexadecimal number. Only the numbers of the message are extracted and converted to decimal for compatibility of types and user control-diagnostic convenience.

Table 1. Predetermined parameters.

Parameter	Value
message size	180 symbol
single data sample size	20 units
the number of input neurons	20
the number of output neurons	1
the hidden layer number of neurons	1 ÷ 10
and the number of layers	3 ÷ 5
the accuracy of convergence	0,001
the maximum number of iterations	5000
the graphical message of epoch status	100 epochs

The predetermined parameters are presented in Table 1. When a character is entered into the terminal, depending on the key entered certain action occurs several operations must be provided - see Table 2.

The whole code was compiled on a Raspbian OS with the GCC compiler directly on the Raspberry Pi B, using the FANN libraries and compilation keys -lm and -lwiringPi.

3. Robotic manipulator algorithm

Complete robotised system including robot manipulator, remote control glove is described in previous Ref. [2].

The glove (1) has embedded buttons which have several functions. They switch modes or directly manage the robot joints. Their signals are directly scanned by an independent thread created by the main thread in the embedded system (6) - see Fig. 2.

Robot operation programme is implemented in the new Raspberry Pi 2 board. Regardless of the mode the motor control thread sends the appropriate commands to the motor control board (7). Switching the mode, the semaphores created by the main thread grant the permission to other threads. Learning thread which starts listening to the myoelectric

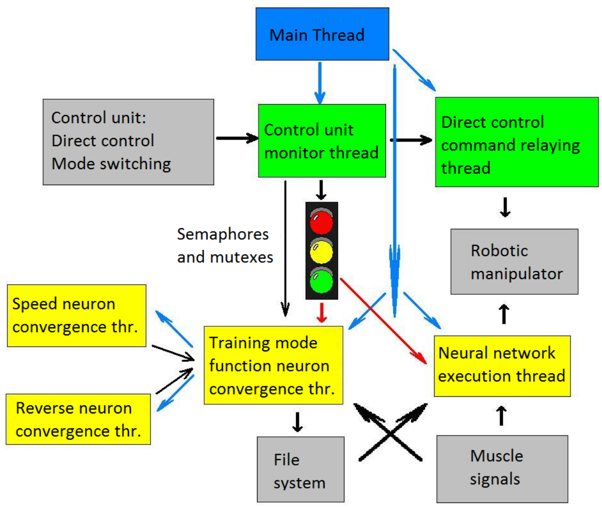

Fig. 2. Block diagram of manipulator control program. Rectangles: gray - external input/output devices; blue - the main independent thread; green - secondary independent thread; yellow - semaphore controlled secondary thread. Arrows: black - data transfer direction, blue - the creation of a secondary thread, red - semaphore control messages.

Table 2. List of operations.

Command	Abbr	Description
Introduction (new training data)	N	need to move one's hand immediately to set its final position and enter the moved distance into the program;
Write data to file	W	required by the FANN library mechanism for further data storage and training;
New data set recording	R	resets the counter;
Neural network training	L	with previously set parameters creates a structure in which the coefficients (synaptic parameters) change to adapt to the training data, they reach certain precision. Having completed the training, network parameters are recorded and saved to a file, and the memory is freed;
Execution mode	X	is similar to the recording mode, only here the data from the UART interface are fed into the trained network, which was loaded from a file. The converged network quickly calculates how far the hand has moved and displays the result in the output.
Exit	Q	changes the cycle "While" condition after which program is stopped the terminal line is returned to the system.

signals from the electrodes (3) via the amplifier (4) and the distribution plate (5) as soon as the signal is sent from the glove (the data which are learned are robot current instructions, the value of the motor speed and reverse setting).

The data are then saved in the file system. When the needed data are registered and the modes are switched, three other threads in parallel start to converge the neural network synaptic parameters. During the execution mode a constant myoelectric signal scanning takes place, but this time the data are sent to the neural network for it to make decisions. At the same time the robot direct control via the glove is also available. The text is illustrated by the diagram shown in Fig. 3.

4. Results

4.1. Myoelectric signal decryption experiment

The arm muscle signal depends on the fatigue and the initial load, as well as an accumulated charge of the system, which may result in additional signal offsets.

This may mean that after each reconnection of the electrodes and after a certain period of time and under certain circumstances it is necessary to recalibrate - additionally train the neural networks. Arm movements can be too early or too late, hence, the waveform may be different from the one that was expected. - see Fig. 3.

The arm movement should be carried out under the same conditions - if the training is carried out with vertical displacements, the testing must also be performed for vertical displacements. Sudden seizures and poor electrode fixation may give signal spikes, which can be unfiltered by the neural networks - see Fig. 4.

The training data show that (without the failed signal probing) the further the hand is moved, the bigger and brighter the signal's round spike. If the jump is actually generated measuring in other coordinates, while testing, the system will show different results. It means that the authentic signal (test case matching the training) will give the correct prediction.

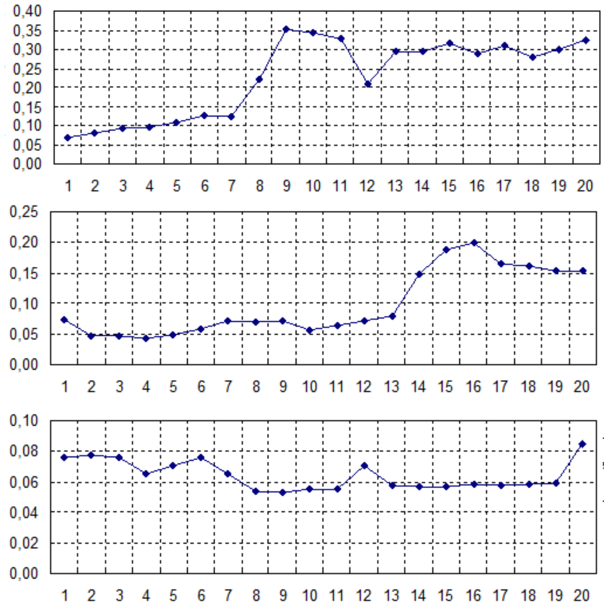

Fig. 3. Test electromyogram. Match of computed and measured signals (22 cm → 40 cm, top; 24 cm → 32 cm, center; 47 cm → 18 cm, bottom). Abscisse x represents an index of set; ordinate y represents the potential of amplified signal, V.

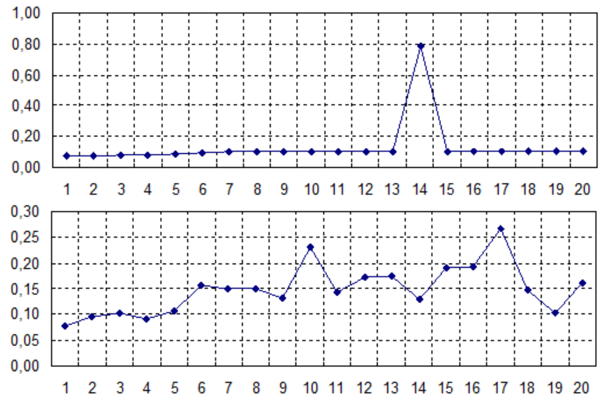

Fig. 4. Two electromiograms: calibration (top) and test(bottom). The measured signal is proportional to the displacement of 18 cm (top). Match of computed and measured signals (24 cm → 34 cm) (bottom).

If one expects the neural networks to converge all the vertical and horizontal displacement data, several hundred samples must be provided to the network for training for such a large architecture to converge into an integration-shift-compensation function. From this point overtraining problem may emerge.

In other words, the accuracy of displacement depends on the proper motion replication. If the movement was successfully measured - this error does not exceed a centimeter from all of the 50 cm scale. If the measurement is wrong, too soon before the probe had recorded the signal, or too late when the waveform is not recognized, the error can be ±20 cm.

Since the weight load measurement signal characteristic are invariant regarding time, the measurements and calibrations are more accurate as the same architecture that was used for distance measurements just averages the time measured signals. The neural network architecture varied, as mentioned before, from 1 to 10 neurons and from 1 to 3 layers.

The neural network architecture was modelled in MatLab R2012a software [9]. The regression coefficient accuracy graph (Fig. 5) showed that a fairly accurate measurement takes place starting from an architecture using 5 neurons. The number of layers has little effect.

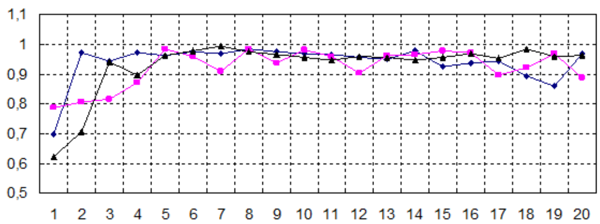

Fig. 5. Neural network displacement measurement regression accuracy dependent on the size of the neural network.

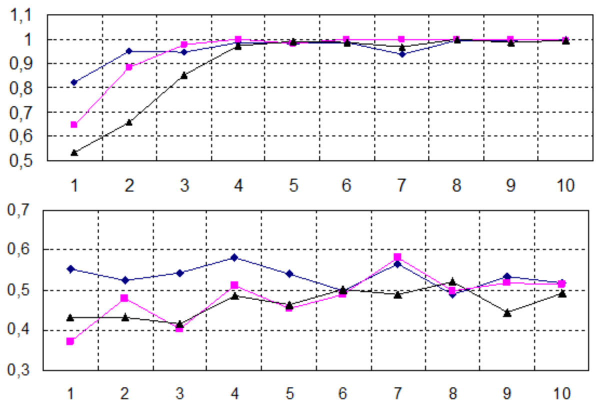

Fig. 6. Neural networks accuracy dependencies: top) on their size for the robot to interpret operations; bottom) on its size while being a continuous architecture.

4.2. Robotic manipulator control

It was tried to program different motion functions using different muscle stress intensity at the same time and to select the motor rotation speed. The whole neural network architecture and the isolated to different variables as the robot function such as motor speed and the reverse selection were tested. Their calculation was programmed simultaneously on a multicore processor.

The neural network architecture chosen for robot operation interpretation also varied in neuron and layer numbers. Fig. 5 shows that the networks converge best, starting with four neurons. However, the number of layers only shows how efficiently the network converges with smaller numbers of neurons.

The same neural network architecture has been tested for the motor speed interpretation (two individual muscle signals combined), as the one for displacement measuring. It turns out that single layer neural network accuracy is sufficient. Two and three layered networks are converging sufficiently as well as starting with 4 neurons. In the end the continuous neural network regression accuracy turned out to be poor. The poor performance of that architecture was not improved by increasing the number of the neurons or layers.

The data accuracy graph can be seen in Fig. 7. The X-axis represents the real value of the data, and Y-axis represents the neural network calculated data. The dashed line expresses the ideal correlation. The solid line expresses the average correlation between the data, and hence the accuracy of the neural network. The lines angle shows the accuracy of regression, which was presented in the above mentioned charts.

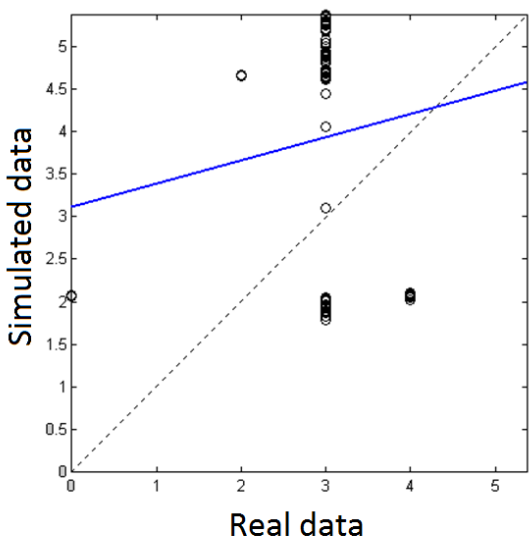

Fig. 7. Accuracy regression of robot operation data for the continuous architecture. The circles are real and calculated data correspondences; the solid line is the fitting of these correlations and the dashed line is the ideal correlation.

The random noise while controlling the robot with muscle tension is ignored and does not affect the performance. Moreover, if the noise spike makes a slight difference the inertia of the engines dampens the unindented movement. The neural network calculation on the Raspberry Pi 2 single core performs five times slower than on the Core 2, while on the Raspberry Pi B - eight times slower.

Conclusions

1. The designed robotic mechanism E.A.R.L. (Ergonomic Assistive Robotic Limb) can successfully be controlled by human muscle impulses decoded by the neural networks.
2. Recalibration must follow each reconnection to the system to the human body. It means that the system is not universal: when training it with one person, it is not suitable for testing with another person.
3. FF BP neuronetwork architecture used for displacement interpretation is suited only when all the variants are trained: with and without fatigue, for delayed and early signals and for offsets from electric charge and electrode conductivity.
4. A successful arm displacement measurement is possible only under the same conditions as in the training. Deviations from the original conditions lead to incorrect measurements.
5. The neural network architecture and data probing mechanism are suitable not only to determine the displacement coordinates, but other dynamic and kinematic parameters such as weight load.
6. The single-layer-architecture with five neurons could be titled as an optimal neural network architecture in all calculations.
7. Non-bonded neural networks converge more accurately than the continuous network.

References

1. Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 1. Problem overview – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 7-12.
2. Vadim Gerasimov, Gintaras Jonaitis, Vytautas Jonkus. Application of Artificial Neural Networks for Human Muscle Signal Analysis and Mechanical Equipment Control. 2. Hardware, software, interfaces – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(18)** 2015 – Pp. 13-17.
3. <http://leenissen.dk/fann/wp/>, accessed 2015.01.16.
4. <https://github.com/AdvancerTechnologies/Muscle-Sensor-v3/blob/master/Documents/Muscle%20Sensor%20v3%20Schematic.pdf>, accessed 2015.05.16.
9. <https://se.mathworks.com/help/releases/R2012a/index.html>