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.

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

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.

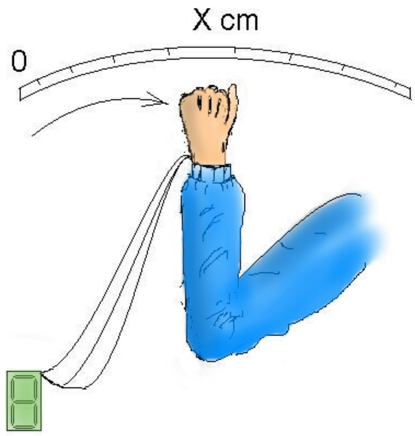

Fig. 1. Schema of biomechanical experiment. The data are entered and displayed on the terminal screen - digital indicator .

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.

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.

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	$1 \div 10$
and the number of layers	$3 \div 5$
the accuracy of convergence	0,001
the maximum number of iterations	5000
the graphical message of epoch status	100 epochs

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

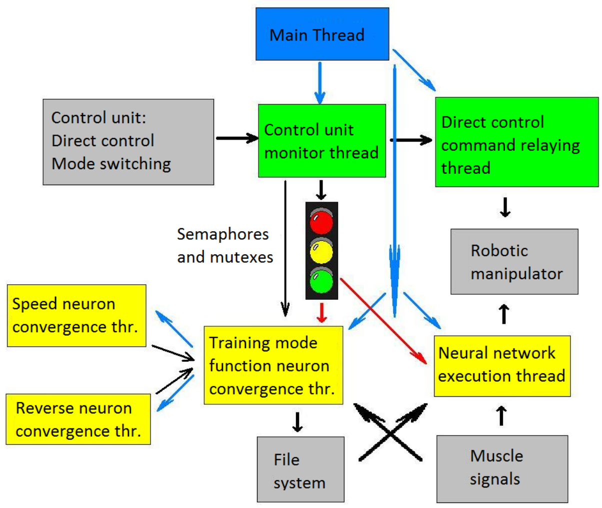

Fig. 2. Block diagram of manipulator control program. Rectangles: gray - external input/output devices; blue - the main independent thread; green - secondary independent thread; yellow - semaphore controlled secondary thread. Arrows: black - data transfer direction, blue - the creation of a secondary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

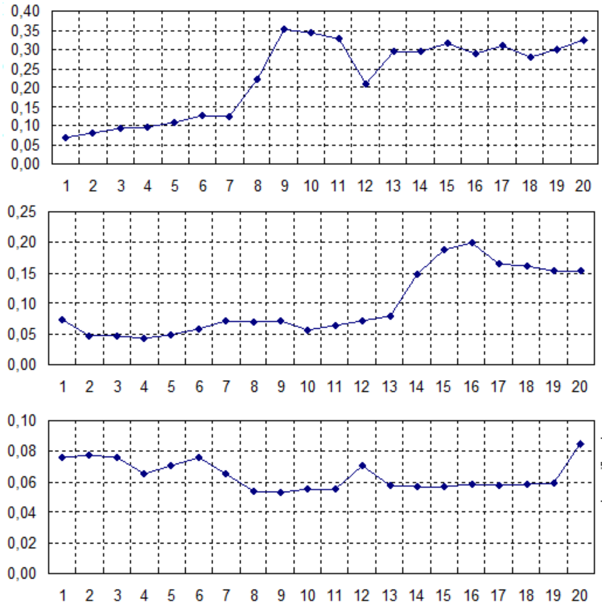

Fig. 3. Test electromyogram. Match of computed and measured signals (22 cm $\rightarrow$ 40 cm, top; 24 cm $\rightarrow$ 32 cm, center; 47 cm $\rightarrow$ 18 cm, bottom). Abscisse x represents an index of set; ordinate y represents the potential of amplified signal, V.

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.

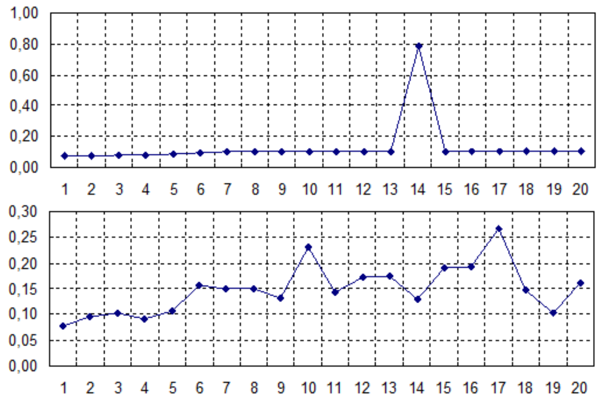

Fig. 4. Two electromiograms: calibration (top) and test(bottom). The measured signal is proportional to the displacement of 18 cm (top). Match of computed and measured signals (24 cm $\rightarrow$ 34 cm) (bottom).

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.

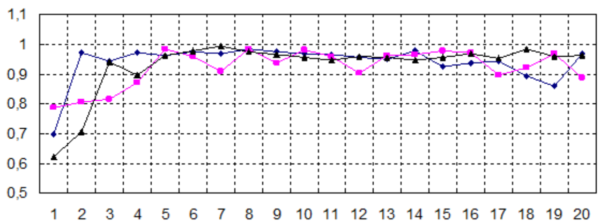

Fig. 5. Neural network displacement measurement regression accuracy dependent on the size of the neural network.

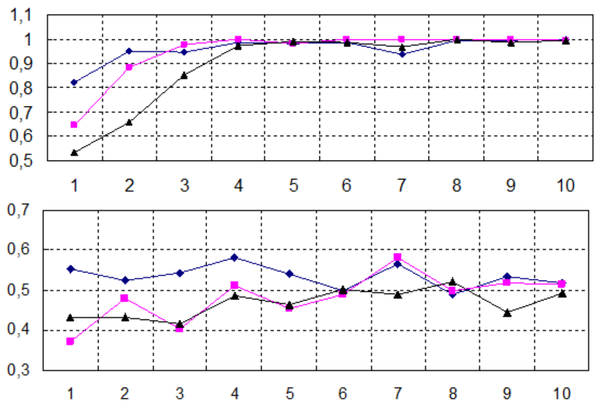

Fig. 6. Neural networks accuracy dependencies: top) on their size for the robot to interpret operations; bottom) on its size while being a continuous architecture.

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.

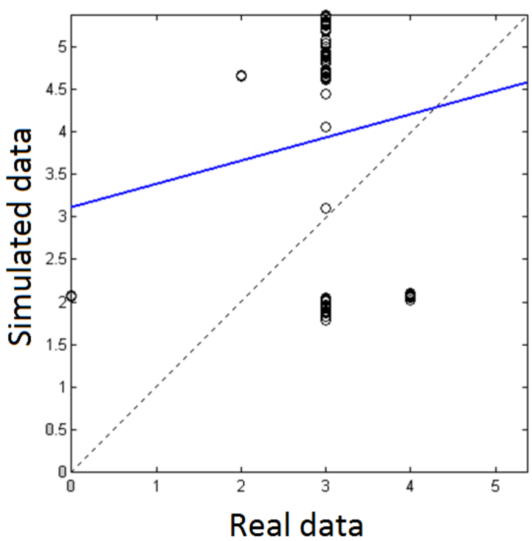

Fig. 7. Accuracy regression of robot operation data for the continuous architecture. The circles are real and calculated data correspondences; the solid line is the fitting of these correlations and the dashed line is the ideal correlation.

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>