Methods of using mobile Internet devices in the formation of the general professional component of bachelor in electromechanics competency in modeling of technical objects

Yevhenii O. Modlo[1[0000-0003-2037-1557]], Serhiy O. Semerikov[2,3,4[0000-0003-0789-0272]], Ruslan P. Shajda[1[0000-0002-7942-9592]], Stanislav T. Tolmachev[3[0000-0002-5513-9099]], Oksana M. Markova[3[0000-0002-5236-6640]], Pavlo P. Nechypurenko[2[0000-0001-5397-6523]] and Tetiana V. Selivanova[2[0000-0003-2635-1055]]

[1] State University of Economics and Technology,
5 Stephana Tilhy Str., Kryvyi Rih, 50006, Ukraine
eugenemodlo@gmail.com, rps21@mail.ru

[2] Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine
semerikov@gmail.com, acinonyxleo@gmail.com, vitro090@gmail.com

[3] Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine
stan.tolm@gmail.com, markova@mathinfo.ccjournals.eu

[4] Institute of Information Technologies and Learning Tools of NAES of Ukraine,
9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

Abstract. The article describes the components of methods of using mobile Internet devices in the formation of the general professional component of bachelor in electromechanics competency in modeling of technical objects: using various methods of representing models; solving professional problems using ICT; competence in electric machines and critical thinking. On the content of learning academic disciplines “Higher mathematics”, “Automatic control theory”, “Modeling of electromechanical systems”, “Electrical machines” features of use are disclosed for Scilab, SageCell, Google Sheets, Xcos on Cloud in the formation of the general professional component of bachelor in electromechanics competency in modeling of technical objects. It is concluded that it is advisable to use the following software for mobile Internet devices: a cloud-based spreadsheets as modeling tools (including neural networks), a visual modeling systems as a means of structural modeling of technical objects; a mobile computer mathematical system used at all stages of modeling; a mobile communication tools for organizing joint modeling activities.

Keywords: mobile Internet devices, bachelor of electromechanics competency in modeling of technical objects, general professional component of bachelor in electromechanics competency in modeling of technical objects, a technique of using mobile Internet devices in learning bachelors of electromechanics.

1 Introduction

Modernization of professional training of specialists in mechatronics at Ukrainian technological universities [21] based on a balance between the fundamental [19] and technological component of the training process necessitates the search for ICT training tools [20], which not only provide the opportunity for active experimentation anytime, anywhere, but also support the development of professionally important qualities of the future electrical engineer, among which the main place is taken by competence in the modeling of technical objects. Today, such universal teaching tools are mobile Internet devices (MID) [12; 26; 35].

This work is a further development of the research begun in articles [20] and [19], the purpose of which is to develop methods of using MID in the formation of bachelor in electromechanics competency in modeling of technical objects. The *purpose of the article* is the selection and justification of the MID-based software tools, the use of which contributes to the formation of the general professional component of this competency.

2 Results

2.1 Use of mobile Internet devices in the formation of competence in using various methods of representing models

The formation of such a general professional component of the competence of a bachelor of electromechanics in the modeling of technical objects, as a *competence in the application of various methods of representing models*, involves the acquisition of knowledge and skills in the construction of computer mathematical and simulation models, their algorithmic and structural description, and the selection of adequate ways of representing computer modeling tools. Based on the content of the competence, its formation and development takes place throughout the entire training of the bachelor of electromechanics, therefore, it is inexpedient to single out the leading disciplines for this process.

So, in the teaching the “Computing Engineering and Programming”, it is possible to solve the problem of numerical integration considered in module 9 “Definite and improper integrals” of the discipline “Higher Mathematics” in a different formulation [19]: instead of using the table representation of the function, use analytic, and instead of interpolating formulas, use stochastic Monte Carlo method. In this case, the mean value theorem for integrals is used, according to which for a curved trapezoid whose area under the graph of a continuous function on a closed, bounded interval is equal to the area of a rectangle whose base is the length of the interval and height is the mean value of the integrand in the interval.

Ilia M. Sobol in [34] proposes an algorithm for the approximate calculation of a definite integral by the Monte Carlo method, noting that “in practice [one-time] integrals ... are not calculated by the Monte Carlo method: for this there are more exact methods – quadrature formulas. However, the transition to multiple integrals changes the situation: quadrature formulas become very complex, and the Monte Carlo method remains almost unchanged" [34, p. 52].

So, to determine the volume under the surface of a function, the following stochastic algorithm can be applied:

1. Limit the surface of a rectangular box, the volume of which V_{par} calculated as the product of length (determined by the integration limits [*a*; *b*] by *Ox* axis) by width (determined by the integration limits [*c*; *d*] by *Oy* axes) by height (determined by the maximum value of the integrand *f*(*x*, *y*) on integration *D*).
2. Place in a certain parallelepiped a certain number of points *N*, the coordinates of which we will randomly choose.
3. Determine the number of points *K* that will be located below the surface of the function.
4. The volume *V*, limited by the function and coordinate axes, is given by the expression $V = V_{par}K/N$.

In order to implement interdisciplinary integration, it is advisable to jointly use this algorithm and the mean value theorem for integrals of a multiple integral:

1. We limit the integration plane to the corresponding limits [*a*; *b*] by *Ox* axes and [*c*; *d*] by *Oy* axes.
2. Place on a certain plane a certain number of points *N*, the coordinates of *N* we will choose at random.
3. At each point we measure the value of the integrand *f*(*x*, *y*) and find the arithmetic mean *M* for all *N* points.
4. The value of the double integral is given by the expression $I = M(b–a)(d–c)$.

The first implementation is performed in the same SageCell modeling environment [17; 25], with which the value of the integral was calculated by deterministic methods (fig. 1):

```
x,y=var('x,y') # symbol integration variables

@interact # interactive model with controls
def _(f=input_box(default=x*y-x^2+y^2, label="$f(x,y)=$", width=20),
   a=input_box(default=0, label="$Lower\ integration\ limit\ by\ Ox\ "
                               "axis\ a=$", width=3),
   b=input_box(default=2, label="$Upper\ integration\ limit\ by\ Ox\ "
                               "axis\ b=$", width=3),
   c=input_box(default=1.2,label="$Lower\ integration\ limit\ by\ Oy\ "
                               "axis\ c=$", width=3),
   d=input_box(default=7, label="$Upper\ integration\ limit\ by\ Oy\ "
                               "axis\ d=$", width=3),
   N=slider(vmin=1, vmax=100000, step_size=1, default=1000,
            label="$Grid\ points\ quantity$"),
   eps=input_box(default=0.1, label="$Accuracy $", width=6)):
```

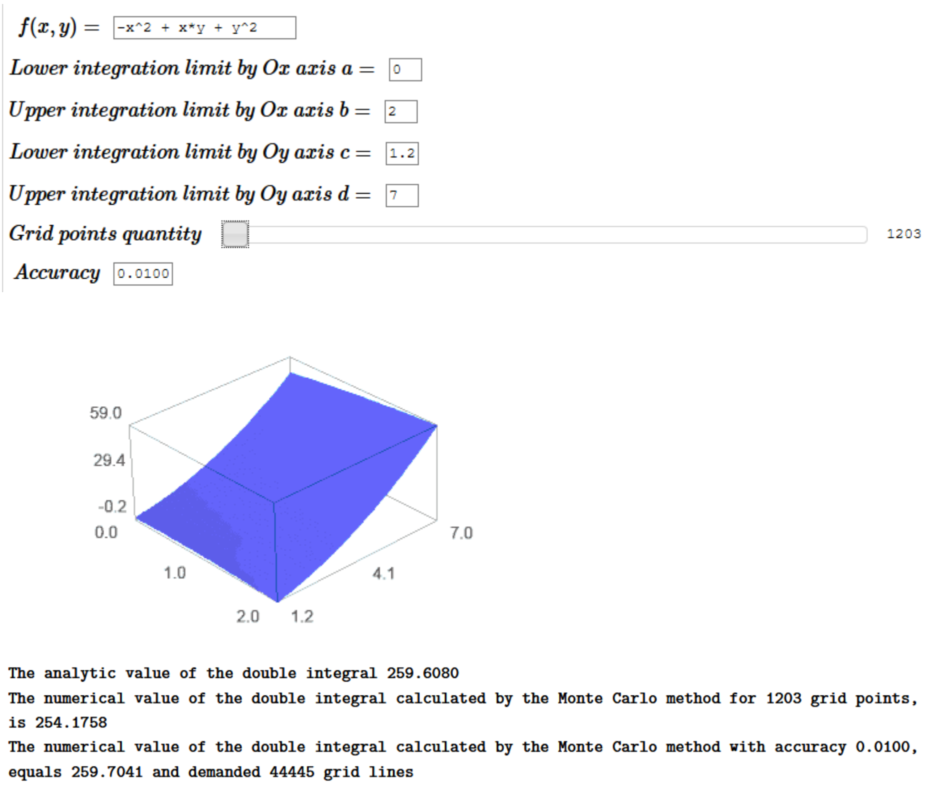
```
# analytic integration of the double integral
I0=integrate(integrate(f,x,a,b),y,c,d)
v=html("The analytic value of the double integral %1.4lf"%I0)

# Monte Carlo numerical integration
# with a known number of experiments
sum=0 # sum of function values f(x,y)
for i in range(N): # grid points quantity cycle
  xi=random()*(b-a)+a # random value generation for x and y
  yi=random()*(d-c)+c #in range [a; b] and [c; d]
  sum=sum+f(xi,yi) # sums accumulation
I=(sum/N)*(b-a)*(d-c) # integral calculations
v=v+html("\nThe numerical value of the double integral "
         "calculated by the Monte Carlo method for %s grid"
         " points,\nis %1.4lf"%(N,I))

# Monte Carlo numerical integration
# with predetermined accuracy
sum=0 # sum of function values f(x,y)
N=1 # minimum quantity of experiments
while true: # cycle without quantity of experiments limit
  xi=random()*(b-a)+a # random value generation for x and y
  yi=random()*(d-c)+c # in range [a; b] and [c; d]
  sumnext=sum+f(xi,yi) # calculating the next sum
  # break condition for the cycle - the module of the
  # difference between the next and previous
  # sum becomes less than the specified accuracy
  if(abs(sumnext-sum)<eps):
    break
  else:
    sum=sumnext
  N=N+1 # move on to the next experiment

I1=(sum/N)*(b-a)*(d-c) # integral calculations
v=v+html("\nThe numerical value of the double integral "
"calculated by the Monte Carlo method with accuracy %1.4lf,\n"
"equals %1.4lf and demanded %s grid lines"%(eps, I1, N))
# function surface visualization
show(plot3d(f, (x,a,b), (y,c,d)))
show(v)
```

Another implementation will be directed by students to conduct as many experiments as possible in order to clarify the fact that with the same number of experiments, the simulation results can differ significantly.



$f(x,y) =$ `-x^2 + x*y + y^2`

Lower integration limit by Ox axis a = 0

Upper integration limit by Ox axis b = 2

Lower integration limit by Oy axis c = 1.2

Upper integration limit by Oy axis d = 7

Grid points quantity 1203

Accuracy 0.0100

```
The analytic value of the double integral 259.6080
The numerical value of the double integral calculated by the Monte Carlo method for 1203 grid points,
is 254.1758
The numerical value of the double integral calculated by the Monte Carlo method with accuracy 0.0100,
equals 259.7041 and demanded 44445 grid lines
```

Fig. 1. The use of various models to calculate the value of a definite integral in a computer mathematics system

To do this, we use the mobile version of the spreadsheets, and enter the following values corresponding to the previous code. At first, the limits of integration:

A1	*a*
B1	0
A2	*b*
B2	2
C1	*c*
D1	1.2
C2	*d*
D2	7

The next step is to enter formulas for random values of *x* and *y* and calculate the function of them:

F1	*x*
F2	=RAND()*($B$2-$B$1)+$B$1
G1	*y*
G2	=RAND()*($D$2-$D$1)+$D$1
H1	*f*(*x*, *y*)

H2 =F2*G2-F2^2+G2^2

The last step is to calculate the average value of the function and the integral:

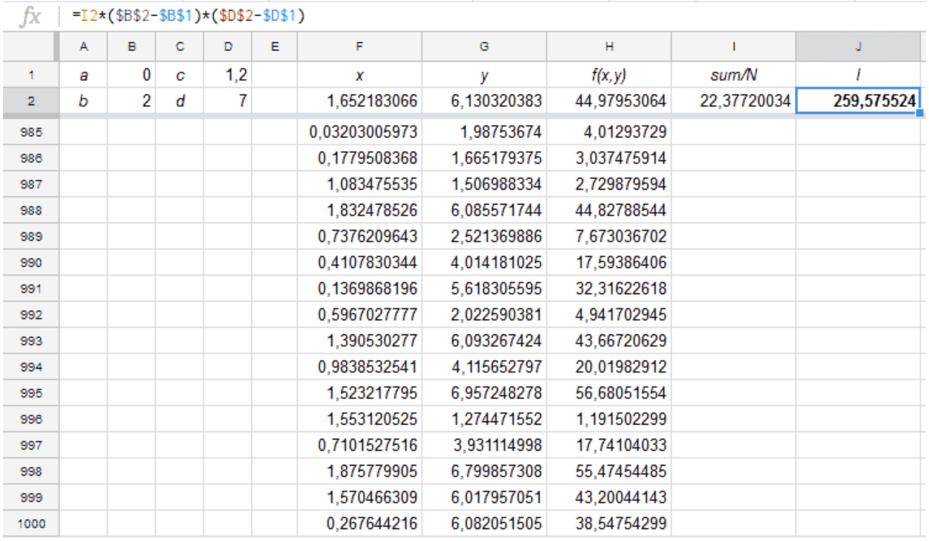
I1 *sum*/*N*

I2 =AVERAGE(H:H)

J1 *I*

J2=I2*($B$2-$B$1)*($D$2-$D$1)

In order to increase the number of points, you must copy the range F2:H2 in any number of other lines. The simulation results are presented in fig. 2.



fx =I2*($B$2-$B$1)*($D$2-$D$1)

	A	B	C	D	E	F	G	H	I	J
1	*a*	0	*c*	1,2		*x*	*y*	*f(x,y)*	*sum/N*	*I*
2	*b*	2	*d*	7		1,652183066	6,130320383	44,97953064	22,37720034	**259,575524**
985						0,03203005973	1,98753674	4,01293729		
986						0,1779508368	1,665179375	3,037475914		
987						1,083475535	1,506988334	2,729879594		
988						1,832478526	6,085571744	44,82788544		
989						0,7376209643	2,521369886	7,673036702		
990						0,4107830344	4,014181025	17,59386406		
991						0,1369868196	5,618305595	32,31622618		
992						0,5967027777	2,022590381	4,941702945		
993						1,390530277	6,093267424	43,66720629		
994						0,9838532541	4,115652797	20,01982912		
995						1,523217795	6,957248278	56,68051554		
996						1,553120525	1,274471552	1,191502299		
997						0,7101527516	3,931114998	17,74104033		
998						1,875779905	6,799857308	55,47454485		
999						1,570466309	6,017957051	43,20044143		
1000						0,267644216	6,082051505	38,54754299		

Fig. 2. Usage of the Monte Carlo method to calculate the value of a definite integral in mobile spreadsheets

A discussion of two implementations of the same model allows us to draw conclusions both on the correct implementation and on the advisability of using the selected modeling tool. So, when using spreadsheets, an increase in the number of experiments is possible only by adding new lines, however, any changes on the worksheet lead to the generation of new random numbers – instead of clarifying the results of the previous experiment, the student conducts a new experiment. At the same time, when using the model in a computer mathematics system, you can verify the erroneousness of the condition for completing the cycle – instead of `abs(sumnext-sum)<eps` must use condition `abs(sumnext/N-sum/(N-1))<eps` for all steps except the first: applying the precondition can lead to both early completion of the cycle (if the function value is close to zero in the first steps) and an overestimation of the number of iterations (if the function value becomes close to zero after reaching the required accuracy), and even to the "eternal" cycle (if the value of the function on the selected area will not be less accurate).

An important component of competence in the application of various methods of representing models is the formation of the ability to select an adequate way of representing models of computer modeling tools. One of the traditional ICT training tools for modeling is computer mathematics systems. In [23] it was pointed out that it should be used together with support systems for teaching bachelors in electromechanics for modeling technical objects, such as the Moodle LMS, supplemented by the developed SageCell filter [18]. The specified filter is able to implement a numerical solution of systems of differential equations describing mathematical models directly in the learning management system.

However, the use of only computer mathematics systems (even such powerful ones as SageMath [27]) for teaching modeling of technical objects of bachelors in electromechanics is not enough, since the synthesis and calculation of models of control systems, electric drive elements, etc. primarily use visual modeling tools that provide the opportunity to build dynamic models (discrete, continuous and models of systems with discontinuities elements). This determines the necessity and expediency of combining traditional computer mathematics systems with specialized libraries for modeling technical objects in environment for the visual construction of models. At the same time, the choice of environment for modeling should take into account the specifics of future professional activity, which for bachelors of electromechanics is the synthesis of the corresponding technical objects – electromechanical systems.

Mastering the modeling of technical objects provides theoretical and practical filling of the fundamental, general and specialized professional training for a bachelor’s of electromechanics. In this regard, it is desirable that the environment for their modeling gives the user access not only to traditional libraries for modeling continuous and discrete dynamic systems, but also to libraries for electric machines and power converters. In addition, to achieve the goal of learning mobility, the modeling environment must have a high level of cross-platform access (in particular, access via a Web interface) and be freely distributed.

In order to make a reasonable choice of the environment for technical objects modeling for bachelors of electromechanics, an expert assessment of the most common systems of visual modeling was carried out, the results of which are presented in the table 1.

Currently, the Scilab has the highest expert assessment, the advantages of which were recognized in 2011 by the French Ministry of National Education, Higher Education and Science, giving Scilab a recognition of its pedagogical significance for teaching mathematics “Reconnu d'Intérêt Pédagogique” [29].

According to [28], Scilab is a package of scientific programs for numerical calculations that provides a powerful open environment for calculations, similar to the Matlab language and a set of functions for mathematical, engineering and scientific calculations. The package is suitable for professional use and use in the universities, providing tools for various calculations from visualization, modeling and interpolation to differential equations and mathematical statistics. Execution of scripts written for Matlab is supported.

Scilab was created in 1990 by INRIA scientists (Institut national de recherche en informatique et en automatique – State Institute for Computer Science and Automation Research) [9] and ENPC (École nationale des ponts et chaussées – National School of Bridges and Roads). At first it was called Ψlab (Psilab). The Scilab Consortium was created in May 2003. To promote the use of Scilab as open source software in the academic and industrial fields. In July 2008, Scilab joined the Digital Foundation to improve technology transfer [10].

Table 1. Comparison of environments for technical objects modeling

Modeling environment	**A freely distributed version availability (“yes” – 3 points, “no” – 0 points, “yes (with restrictions)” – 1 point)**	**The number of supported operating systems (for each system – 0.5 points, for “∞” – 1 point)**	**The Web interface availability (“yes” – 3 points)**	**Libraries for modeling continuous systems (“yes” – 5 points)**	**Libraries for modeling discrete systems (“yes” – 5 points)**	**Libraries for modeling electric machines (“yes” – 2 points)**	**Libraries for modeling power converters (“yes” - 2 points)**	**Total**
Analytica	yes (with restrictions)	1 (W*)	yes	no	no	no	no	4,5
AnyLogic	yes	3 (WML)	no	no	no	no	no	4,5
GoldSim	no	1 (W)	no	no	no	no	no	0,5
Insight Maker	yes	∞ (JS)	yes	no	no	no	no	7
MapleSim	no	3 (WML)	no	no	no	yes	yes	5,5
Minsky	yes	3 (WML)	no	no	no	no	no	4,5
Rand Model Designer	no	1 (W)	no	yes	yes	no	no	10,5
Scilab Xcos	yes	3 (WML)	yes	yes	yes	no	no	17,5
Simantics System Dynamics	yes	1 (W)	no	no	no	no	no	3,5
Simile	yes (with restrictions)	3 (WML)	yes	no	no	no	no	5,5
Simulink	no	3 (WML)	no	yes	yes	yes	yes	15,5
Temporal Reasoning Universal Elaboration	no	1 (W)	no	no	no	no	no	0,5
Vensim	yes	2 (WM)	no	no	no	no	no	4
VisSim	no	1 (W)	no	no	no	no	no	0,5
Wolfram SystemModeler	no	3 (WML)	no	no	no	yes	yes	5,5

* W – Windows, M – macOS, L – Linux, JS – JavaScript

In June 2010, the Scilab Consortium announced the creation of Scilab Enterprises [4]. Scilab Enterprises develops and sales, directly or through an international network of affiliate service providers, a comprehensive suite of services for Scilab users. Scilab Enterprises also develops and maintains Scilab software. Scilab Enterprises' ultimate goal is to help make using Scilab more efficient and easier. Since February 2017, Scilab has been developed and published by ESI Group, an industrial virtual reality development company [5].

Scilab contains hundreds of mathematical functions with the ability to add new ones written in various languages (C, C++, Fortran, etc.). Various data structures are supported (lists, polynomials, rational functions, linear systems), an interpreter and a high-level language.

Scilab was designed as open system in which users can add their data types and operations on this data by overloading.

Many tools are available in the system:

- 2D and 3D graphics, animation;
- linear algebra, incl. work with sparse matrices;
- polynomial and rational functions;
- interpolation, approximation;
- differential equations;
- Scicos: a hybrid tool for modeling dynamic systems;
- optimization;
- signal processing;
- parallel computing;
- statistics;
- work with computer algebra;
- Fortran, Tcl/Tk, C, C++, Java, LabVIEW interfaces;

Scilab has a programming language similar to MATLAB; the system includes a utility that allows to convert Matlab documents to Scilab.

Scilab allows to work with elementary and a large number of special functions (Bessel, Neumann, integral functions), has powerful tools for working with matrices, polynomials (including symbolic), performing numerical calculations (for example, numerical integration) and solving linear algebra problems, optimization and simulation, powerful statistical functions, as well as a tool for building and working with charts. For numerical calculations, the libraries Lapack, LINPACK, ODEPACK, Atlas and others are used.

The package also includes Scicos, a tool for editing block diagrams and simulations (the Xcos add-on is an analogue of the Simulink package in MATLAB). There is the possibility of collaboration between Scilab and LabVIEW.

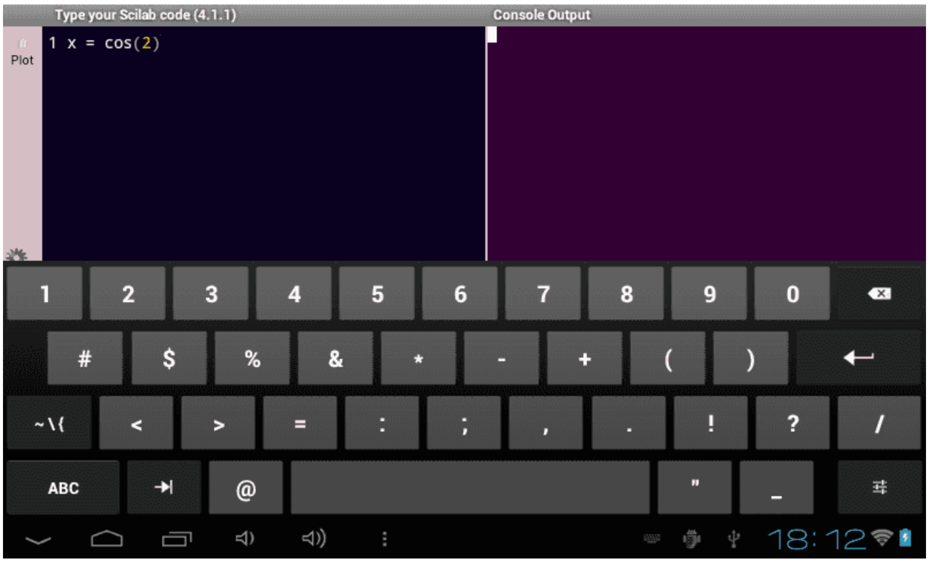
Distinctive features of Scilab:

- fee free;
- small size (distribution takes less than 150 Mb);
- the ability to run in the console without using a graphical interface. This allows automated calculations in batch mode.

Starting with version 6, the program is distributed under the GPL compatible license CeCILL license [1].

Despite the lack of usability of the text interface, it is easier to adapt to MID. Among the well-known Scilab implementations for the Android OS, we distinguish two – Scilab on Aakash and Scilab Console Free.

Scilab on Aakash [30] is an Indian development that is supported in both the Android and GNU/Linux Aakash versions. In Android, Scilab is part of the APL library (Aakash Programming Lab [7], developed by Indian Institute of Technology Bombay), which provides Scilab 5.4. All Scilab functions cannot be runnging on Android, so the developers provide the user interface with only 2 windows (fig. 3): enter commands and view the results of their execution.



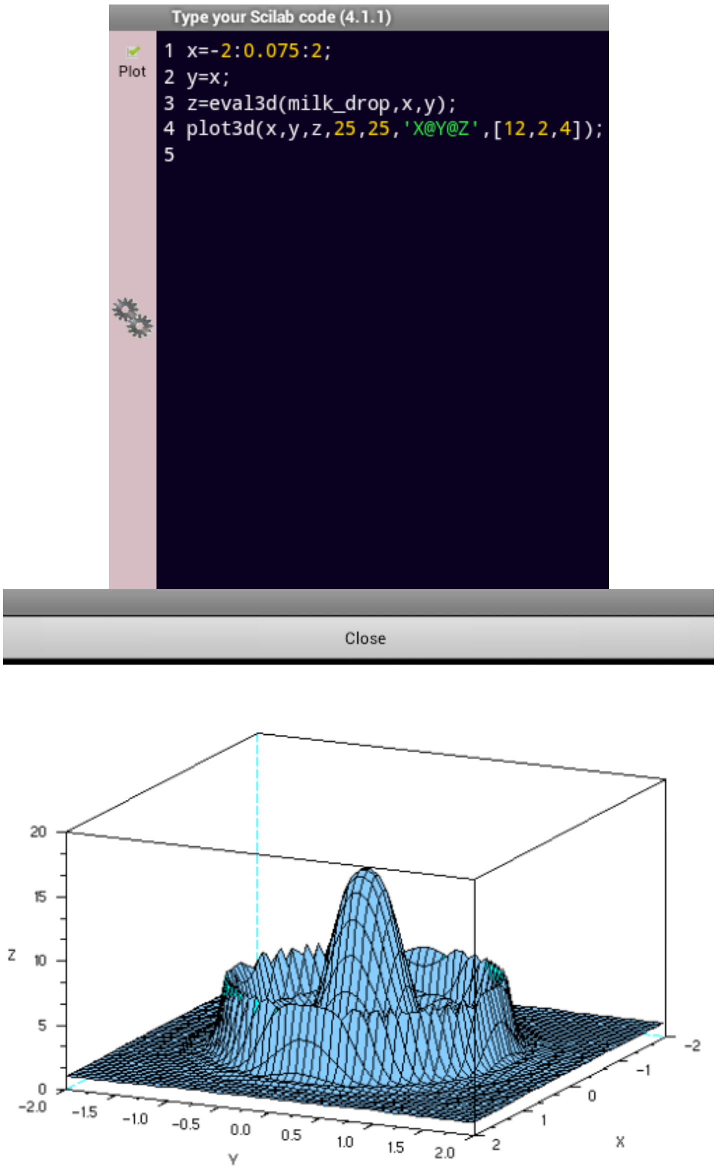
```
x = cos(2)
```

Fig. 3. Scilab on Aakash window interface

When executing graphic commands with the Plot option selected, the results of execution will be displayed in a separate window (fig. 4). This version of Scilab may be admitted as mobile, but not fully functional.

Scilab Console Free (fig. 5) – is a mobile version of Scilab for Android and iOS. Like Scilab on Aakash, this is not a fully functional version of Scilab 5.4.1: the graphic functions of Scilab and Xcos are not activated. The developers of this product went the easy way – to achieve mobility on the device you first have to install one of the Linux options (for example, using GNURoot Debian [2]).

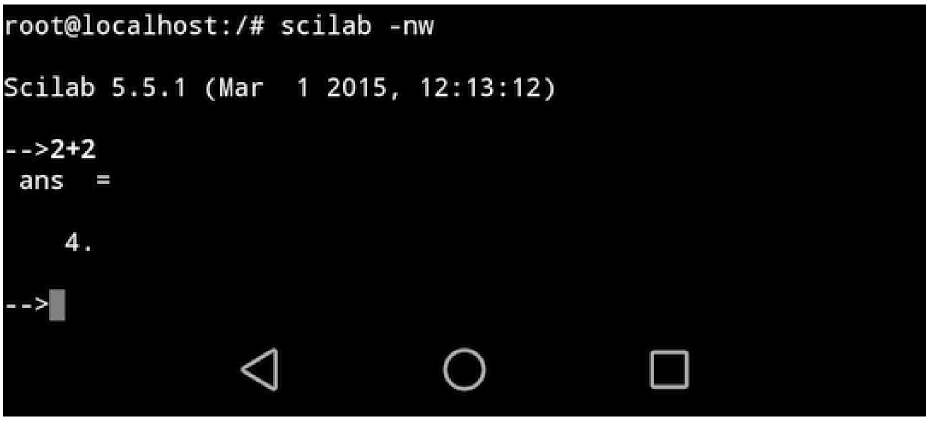
The mobility of this version of Scilab is achieved by transferring the Linux mobile operating system and its software environment to Android and iOS. Therefore, Scilab Console Free is quite demanding on the internal memory of the mobile device and the version of the operating system.



```
x=-2:0.075:2;
y=x;
z=eval3d(milk_drop,x,y);
plot3d(x,y,z,25,25,'X@Y@Z',[12,2,4]);
```

Fig. 4. Graphic plotting in Scilab on Aakash

The standard Web interface to Scilab, which is proposed by *Cloud Scilab* [36], does not provide an interface similar to the full-featured version – it is only possible to host user-created models with a Web interface (application deployment – “deploying” programs to the cloud server [15]) by creation and publication of interactive documents, the software part of which is performed on the server side, saves the resources of a MIDs (fig. 6).



```
root@localhost:/# scilab -nw

Scilab 5.5.1 (Mar  1 2015, 12:13:12)

-->2+2
 ans  =

    4.

-->
```

Fig. 5. Scilab Console Free launch window

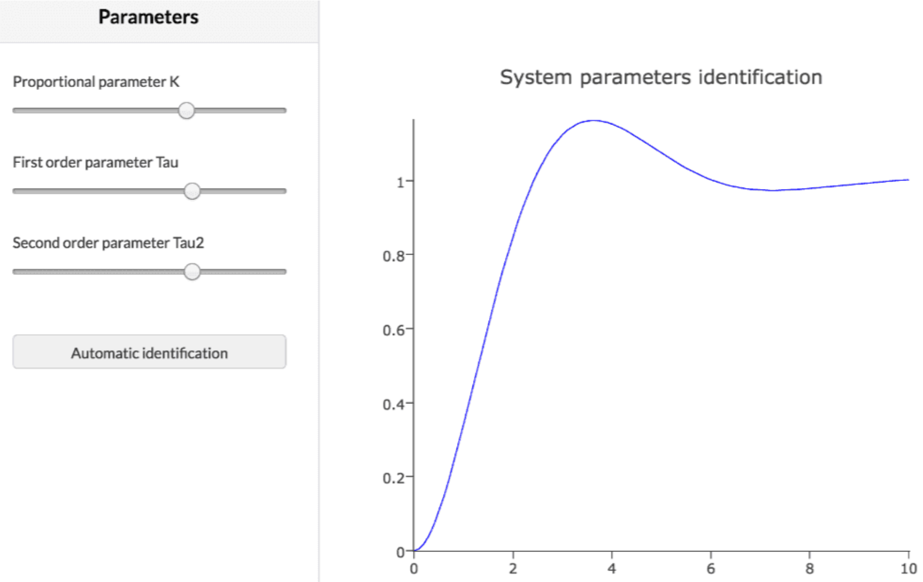

Parameters
Proportional parameter K
First order parameter Tau
Second order parameter Tau2
Automatic identification
System parameters identification
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10

Fig. 6. An example of an interactive document in Scilab Cloud

This provides several advantages:

- centralization of data used and created by the user program;
- no need to install software on the client side;
- concealment of code from end users (as a component of intellectual property protection);
- centralization of Scilab code to ensure the effective operation of programs.

Programs deployed on the Scilab Cloud are described by the Scilab language (both algorithms and the user interface). This allows you to create programs with a visual interface both for deployment in the cloud (work in Scilab Cloud with display via a web browser) and traditional execution (launch in Scilab on the user's computer).

Despite the high level of mobility, the use of Cloud Scilab is accompanied by a number of problems:

- deployment of programs is only available with administrative privileges (requires an additional fee to the cloud service provider);
- the interface is provided only for user models, and not to the entire Scilab interface.

In terms of functionality, this service is similar to the Wolfram Demonstrations Project, however, unlike the latter, all users can publish demonstrations, and not only service administrators comply with the document distribution model in CoCalc [14]. However, Scilab Cloud does not provide access to the Xcos module. In addition, in fact, the only source about Scilab Cloud is the materials of webinars conducted by Scilab Enterprises [3], which does not contribute to the widespread use of this Web-interface.

W3 Scilab is significantly more open [13]. W3 Scilab is the Indian Web-based interface to Scilab, which is the basis of Scilab on Aakash. The interface allows users to send short fragments of Scilab code to a remote server, receive Scilab code on this server, return and display execution results in a browser.

Xcos is an addition to Scilab, which allows the synthesis of mathematical models in the fields of mechanics, hydraulics, electronics and electromechanics. This visual modeling environment is designed to solve the problems of dynamic modeling of systems, processes, devices, as well as testing and analysis of these systems. In this case, the object is modeled (system, device, process), supplied graphically in the form of a block diagram that includes blocks of system elements and the connections between them (fig. 7).

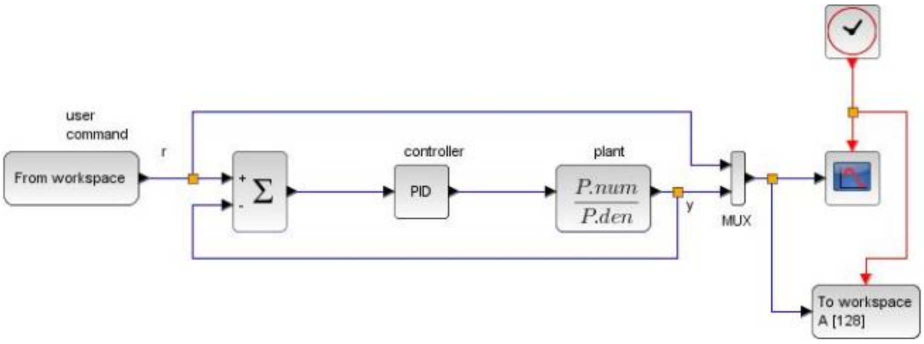


Fig. 7. Scilab Xcos model example

The most significant criteria that led to the choice of Scilab as a tool of teaching modeling of technical objects of bachelors in electromechanics are the availability of modeling continuous systems libraries (5 points), modeling discrete systems libraries

(5 points) and the presence of a Web interface (3 points). The latter makes it possible to use it on MID.

The Xcos on Cloud service provides the ability to build simulation models of technical objects (in particular, electromechanical systems) in a mobile Web browser. The current goal of the Xcos on Cloud project (formerly Xcos on Web [22]) is to recreate a fully functional version of Scilab Xcos with access via a mobile Web browser [6].

The main components of the Xcos on Cloud main window are presented in fig. 8.

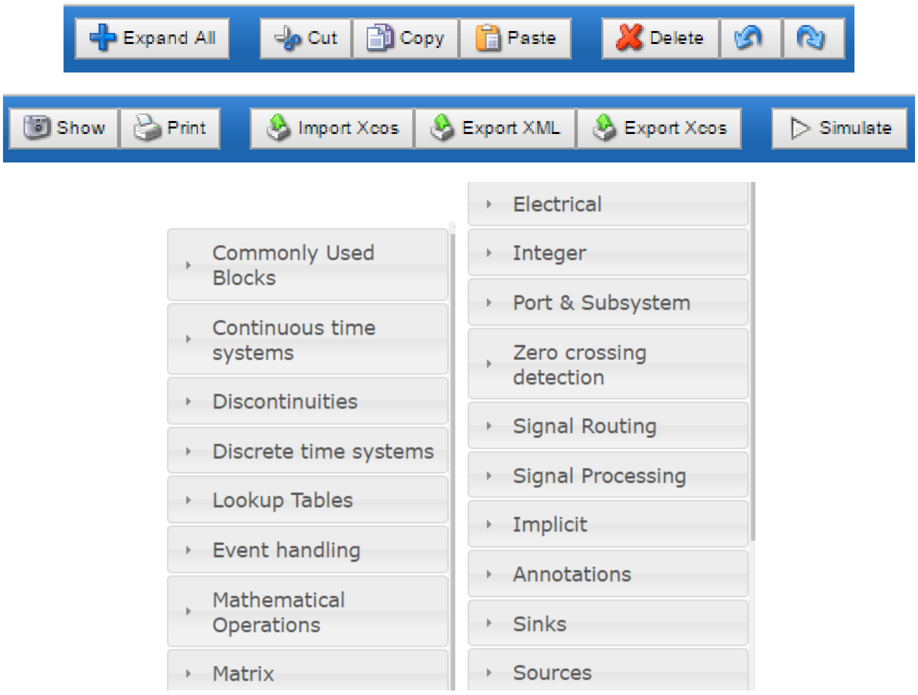


Fig. 8. Xcos on Cloud main window components

The main part of the Xcos on Cloud main window is occupied by the model building area. On the left side is the so-called palette of blocks – a library of elements from which the model is built. To use any block, just drag it from the palette to the model building area. Blocks are interconnected by communication lines.

In fig. 9 shows a DC motor model built in Xcos on Cloud. Blocks of the model have various parameters, user-configurable by double-clicking on the selected block. Unfortunately, the project is still at the preliminary development stage, so some settings are not available: for example, the first-order aperiodic unit (CLR) and amplifier (GAIN) blocks in the constructed model are unstable, as evidenced by the format specifier '%s'. This makes it impossible to conduct experiments on the model by the click of a button. *Simulate*. A temporary workaround for this problem is the ability of

Xcos on Cloud to exchange data with the traditional version of Scilab Xcos through data export tools.

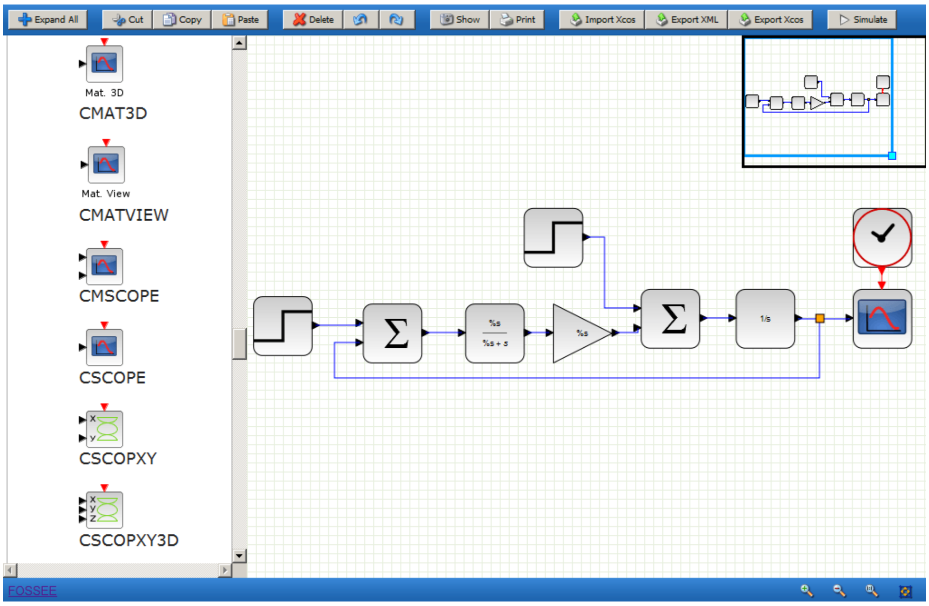


Fig. 9. DC Engine model in Xcos on Cloud

A fragment of an XML representation of a constructed model of a DC motor:

```
<?xml version="1.0" encoding="UTF-8"?>
<XcosDiagram background="-1" title="MavXcos">
    <mxGraphModel as="model">
        <root>
            <mxCell id="0"/>
            <mxCell id="1" parent="0"/>
            <BasicBlock blockType="c" id="2"
interfaceFunctionName="STEP_FUNCTION" parent="1"
simulationFunctionName="csuper"
simulationFunctionType="DEFAULT" style="STEP_FUNCTION">
...
        </root>
    </mxGraphModel>
  <mxCell id="1" parent="0" as="defaultParent"/>
</XcosDiagram>
```

The implementation of the full functionality of Scilab Xcos in Xcos on Cloud creates the conditions for a using MID in the training of bachelors in electromechanics, modeling of technical objects. Similar features are provided by the full virtualization of Scilab on rollApp [31].

2.2 Use of mobile Internet devices in the formation of competence in solving professional problems using ICT

The formation of a general professional component of the competence of a bachelor of electromechanics in the modeling of technical objects, such as *competence in solving professional problems using ICT*, provides for the acquisition of knowledge in the field of computer engineering and programming, the ability to create application software, and the skills to work with ICT tools to solve problems in field of electromechanics. The formation of this competence begins in the academic discipline “Computing Engineering and Programming” and occurs along with the formation of the general scientific component of the competence of the bachelor of electromechanics in the modeling of technical objects, in particular competencies in information and communication technologies and in applied mathematics. Its further development takes place in the process of developing models; they are considered in the academic disciplines “Theory of Automatic Control” and “Modeling of Electromechanical Systems”.

The purpose of studying the discipline “Theory of Automatic Control” is to master the methods of setting tasks, the principles of building automatic control systems, methods of analysis and synthesis of linear, nonlinear, impulse, digital, adaptive and optimal systems using modern software systems for modeling dynamic systems.

As a result of studying the discipline, students, in particular, should receive skills:

- compose differential equations for elements of automatic control systems and the system as a whole;
- draw up and transform structural diagrams of automatic control systems;
- determine the time functions and time characteristics of the automatic control system and its elements under the conditions of various types of signals;
- determine the frequency functions and characteristics of automatic control systems;
- analyze the stability of linear, impulse and nonlinear automatic control systems;
- calculate control devices (controllers) that provide the necessary quality indicators, including using the state space method;
- synthesize automatic control systems in the presence of random signals;
- apply adaptive methods to control non-stationary objects.

At the lectures of the second module “Properties and characteristics of closed control systems. Synthesis of linear continuous control systems”, the issues of influence on the system and requirements for the control process, stability of control systems, stationary (stable) modes of linear control systems, quality assessment of control systems with step and arbitrary actions, frequency methods for assessing the quality of control systems, approximate methods for choosing the control method and parameters are considered regulators, synthesis of control systems according to the logarithmic frequency characteristics. To consolidate this material, we propose laboratory works “Analysis of stability and quality of control systems” and “Synthesis and research of control systems for objects with a delay”. Consider the use of ICT tools to solve the second of them.

Objective of the work: to study the influence of the delay link on the stability and quality of the automatic control system.

The content of the work

1. The study of the influence of the delay link on the characteristics of the automatic control system.
2. Experimental obtaining the transient and frequency characteristics of the system with delay.

Theoretical information

The automatic control systems may include time delay, the equations of which are of the form:

$$y(t) = x(t - \tau), \tag{1}$$

where τ – delay time.

The transfer function of a time delay in accordance with the delay theorem (properties of the Laplace transform):

$$W_{\text{del}}(s) = e^{-s\tau}. \tag{2}$$

Automatic control systems, which include a time delay, are called time delay systems.

It can be connected in the direct circuit of the system or to the feedback circuit. Moreover, regardless of the location of the delay link, the characteristic equation of a closed system with delay has the form:

$$D_{\tau}(s) = Q(s) + R(s)e^{-s\tau} = 0, \tag{3}$$

where $Q(s)$ and $R(s)$ – polynomials in the denominator and numerator of the transfer function of the open system without delay.

This characteristic equation is not a polynomial and has an infinite number of roots. Therefore, to study the stability of delayed systems, it is necessary to use frequency stability criteria, such as the Nyquist stability criterion. The conclusion about the stability of the system can be made on the basis of the analysis of the amplitude-phase frequency response of an open system with delay.

It can be shown that the presence of a delay link does not change the module of $A(\omega)$ but introduces only an additional negative phase shift of $-\omega\tau$. By varying the delay time τ over a wide range, one can find its value at which the closed-loop system will be at the stability boundary. In this case, amplitude-phase frequency characteristic of the open-loop phase response of the delayed system will pass through the point (-1; 0) = -1+j0.

The latency τ_{cr} and the corresponding value of the frequency ω_{cr}, at which the AFC passes through the point (-1; 0), are called *critical*. For a critical case, the following conditions are true:

$$A(\omega_{\text{cr}}) = 1; \ \phi_{\tau}(\omega_{\text{cr}}) = -\pi. \tag{4}$$

The automatic control system will be stable if the delay time τ is less than critical: $\tau < \tau_{\text{cr}}$.

Work order

The subject of the study is an automatic control system, with a delay link.

1. The transfer function of the open-loop automatic control system without delay has the form:

$$W(s) = \frac{K}{s\ (T_1 s+1)(T_2 s+1)} \tag{5}$$

The value of the transmission coefficient K and the time constants T_1 and T_2 are shown in the table

Num of var.	**1**	**2**	**3**	**4**	**5**
***K*, s^{-1}**	5	3	1	10	8
T_1, s	0,5	0,1	0,05	0,05	0,04
T_2, s	0,01	0,05	0,01	0,2	0,1

Using the Scilab package, construct the transition characteristic $h(t)$ of the closed-loop system with unit negative feedback (fig. 10, 11). Make a conclusion about the stability of the system.

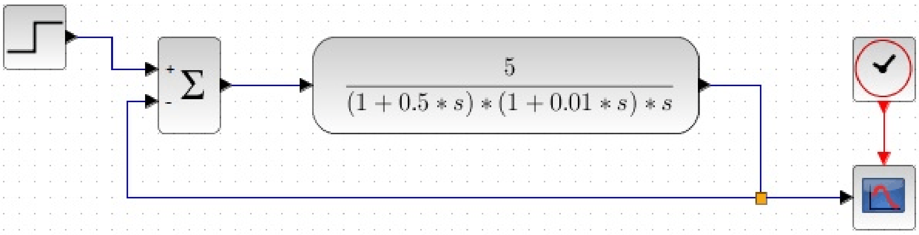


Fig. 10. Block diagram of a closed-loop system without delay

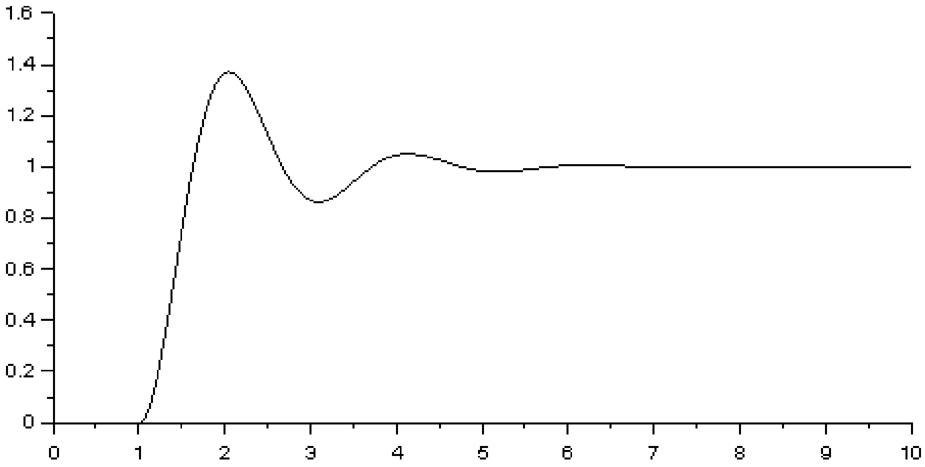


Fig. 11. Transient response of a closed-loop system with a single negative feedback without delay

2. Insert a time delay function with the delay time $\tau = 0{,}01$ s (fig. 12), into the direct chain of the system, construct the transition characteristic of the system (fig. 13) and make a conclusion how the time delay affects the quality of the transition process.

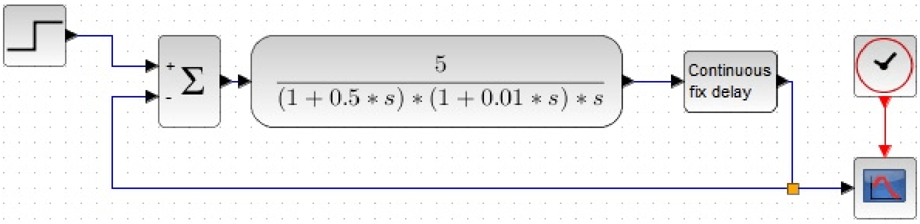


Fig. 12. Block diagram of closed-loop control system with delay time

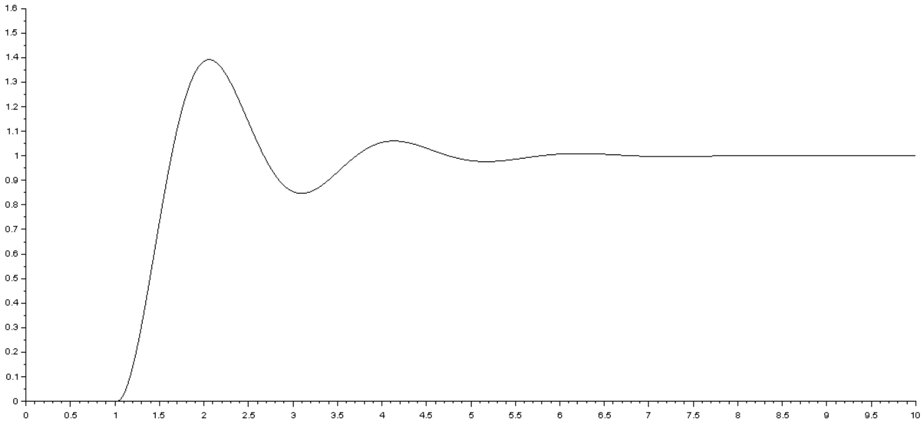


Fig. 13. Transient response of a closed-loop system with a negative feedback with delay $\tau = 0{,}01$ s

3. Increasing the time of delay, to trace how the transition characteristic of the system changes (fig. 14) to make the system lose stability (the transition process diverges).
4. To determine experimentally the value of the critical delay time τ_{cr}, when the system will be at the stability boundary (the transition process will be undamped).
5. Repeat steps 2-4 for the case when the time delay function included in the feedback circuit. Make some conclusions.

Further development of competence in solving professional problems by means of ICTs occurs in the process of completing term paper in the discipline “Theory of automatic control”. In order to develop skills for conducting professionally directed educational research, students can be invited to familiarize themselves with new results in the subject area of scientific publications in professional publications and start the coursework on their reproduction. So, in the work “Optimization of the servo-controlled automatic control system” [11] an example of servo-controlled control system of a working body of a paver with hydraulic drive is considered.

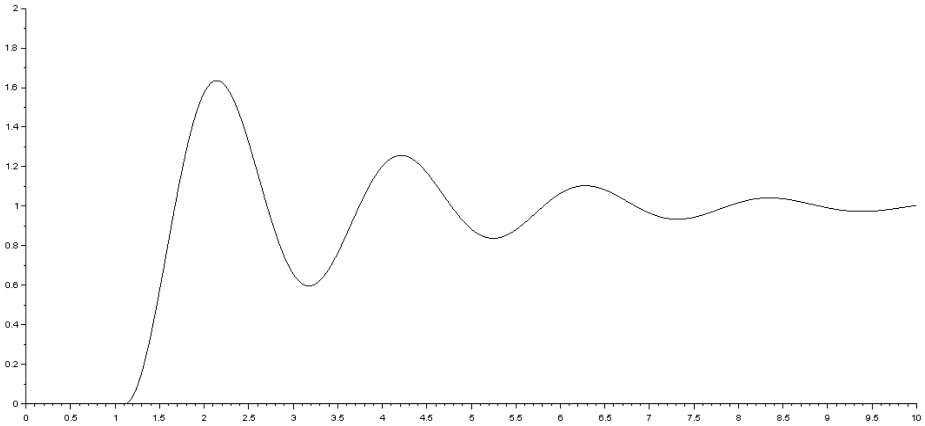


Fig. 14. Transient response of a closed-loop system with a negative feedback with delay $\tau = 0{,}1$ s

When moving the stacker along the base, which is ready for laying pavement on it, its running equipment (tracked or wheeled) makes uncontrolled random movements in the vertical direction under the influence of roughnesses in the microrelief of the base of the road. These movements are transmitted through the stacker frame and suspension of the screed to the working equipment, causing in turn uncontrolled movements of the screed, which entail a random change in the thickness and angle of the transverse slope of the stacked layer, thereby deteriorating the quality of the coating.

A simulation model of the hydraulic drive tracking system for the screed plate can be implemented in MATLAB Online, as suggested by the authors of [11]. The structure of the simulation model circuit (fig. 15) includes the following elements: a bi-directional hydraulic cylinder; three-position valve; hydraulic pump; controlled hydraulic lock; ideal hydraulic pressure sensor; the “smooths the paver plate” element; cylinder rod movement and speed sensor (feedback sensor) the ideal force sensor; the element “hydraulic fluid” (Oil-30W oil) is proportional to the servo valve of the hydraulic actuator (electro-hydraulic distributor that converts the electrical signal to movement) element “viscous friction”; ideal source of strength; disturbing influence of “microrelief”; PS-converter; capacity for working fluid; disturbing effects due to the influence of the work of other elements.

2.3 Use of mobile Internet devices in the formation of competencies in electric machines and critical thinking

In the discipline “Electrical Machines”, the formation of both the general scientific component of the competence of the bachelor of electromechanics in the modeling of technical objects (the leading means of competencies in basic sciences formation were mobile augmented reality tools), and such a general professional component as *competencies in electrical machines*, including knowledge of the structure and functioning principles of electrical machines, in particular: energy conversion processes (electromagnetic and electromechanical), characteristics of certain types of electrical machines, structure of asynchronous machines, synchronous machines, DC machines, transformers; the ability to calculate the parameters and characteristics of electrical machines. To form the latter, the leading tools are mobile computer mathematics systems.

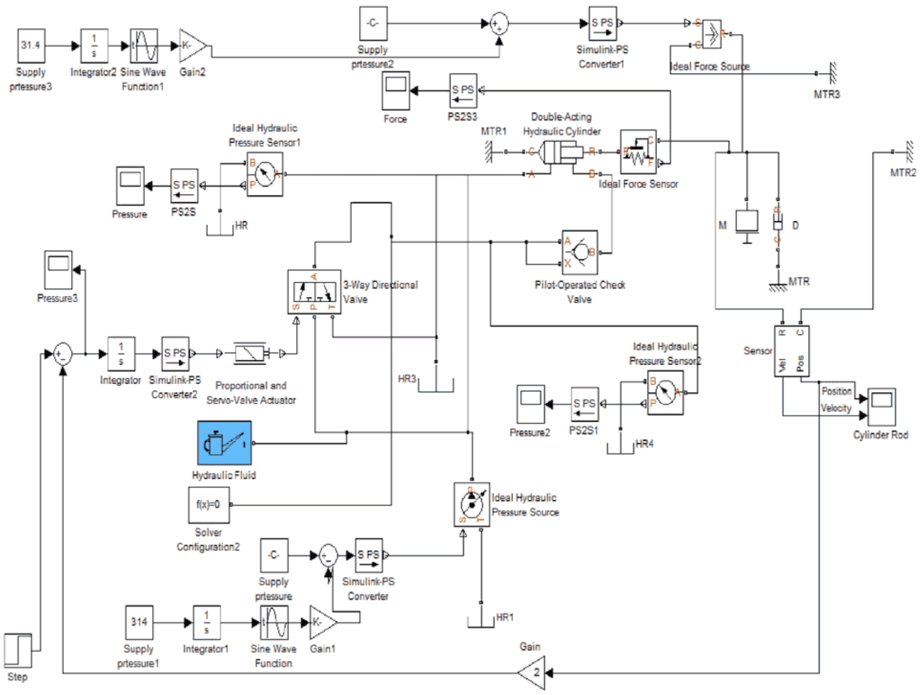


Fig. 15. Structural model of the hydraulic drive tracking system for the screed slab

The last general professional component of a bachelor of electromechanics competence in modeling technical objects is *competence in critical thinking* – the knowledge and skills of setting a problem with an insufficient amount of input data, analyzing the availability of methods and means of solving a problem, assessing your own readiness to solve a problem, independently searching for missing data and ways to solve the problem; the ability to control their own activities – both mental and practical; the ability to control the logic of deploying your own thoughts; the ability to determine the sequence and hierarchy of the stages of activity and the like. As well as for competence in the application of various ways of representing models, leading academic disciplines cannot be distinguished to this competence – its formation takes place throughout the entire training of a bachelor of electromechanics.

Neural network modeling is an effective means of modeling technical objects with a hidden or fuzzy structure [32; 33]. As the editors of the book "Neuro-Control and its Applications" note, the relevance of its application is due to the need to develop control methods for complex nonlinear systems: "The first examples of the development of control methods for nonlinear systems ... are mainly associated with methods for solving nonlinear differential equations that are adequate to a single-processor background of Neumann computational to cars. ... The development of computers with mass parallelism ... has led to the creation of fundamentally new algorithms and methods for controlling non-linear dynamic systems. They are associated with neural network algorithms for solving ordinary nonlinear differential equations and, as a result, with the inclusion of a neurocomputer in the control loop of a nonlinear dynamic system. ... The sufficiently wide development and spread of such algorithms has led ... to the creation of a whole branch of science called "neurocontrol" [24, pp. 9–10].

In an engineering context, intelligent management should have the following properties: learning ability and adaptability; survivability; simple control algorithm and user-friendly man-machine interface; the ability to incorporate new components that provide the best solutions in the face of limitations imposed by technical tools [24, p. 15].

The deep machine learning is a class of intelligent control algorithms that use multilayer neural networks with non-linear nodes. Let us consider the construction of a neural network model for approximating data described in Chapter 23 of "Neural Network Design" [8], and obtained using an intelligent sensor – one or more standard sensors connected to a neural network to obtain a calibrated measurement of one of the parameters.

The intelligent position sensor uses the voltage value from two photocells to estimate the location of the object. In fig. 16, an object located between a light source and two photocells is shown around. The object, moving along the *y* axis, casts a shadow on the photocells, which leads to a change in the voltages v_1 and v_2. When the position of the object *y* increases, the voltage v_1 first decreases, then the voltage v_2 decreases, then v_1 increases and finally v_2 increases (fig. 17).

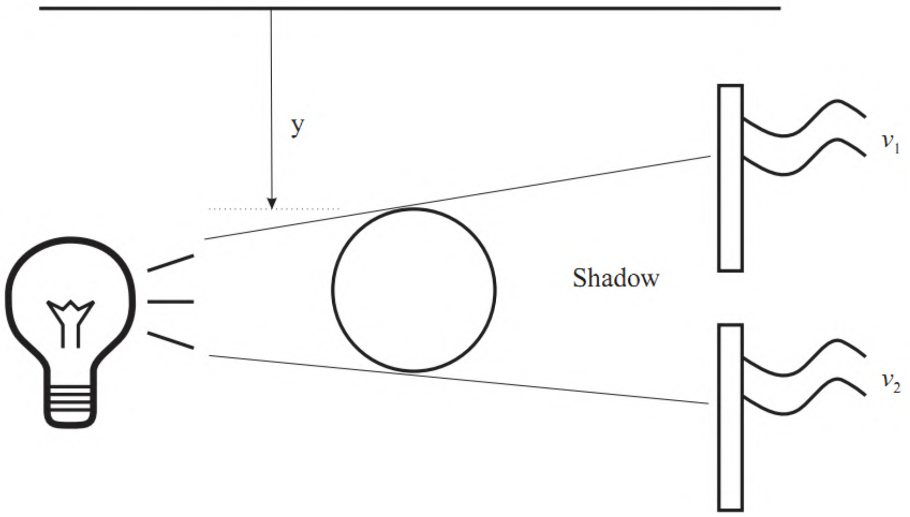


Fig. 16. Intelligent position sensor

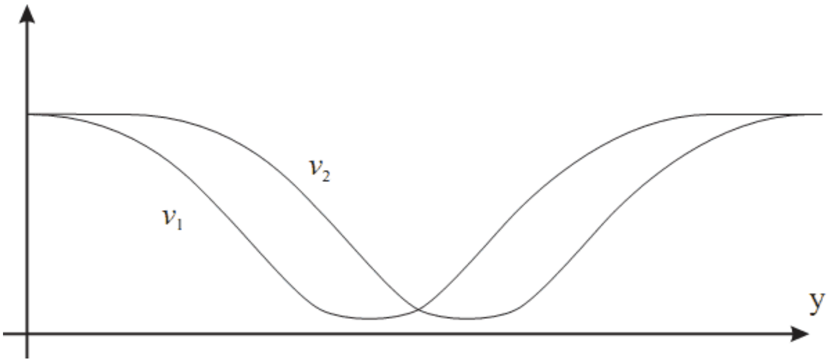


Fig. 17. An ideal model of the dependence of the voltages v_1, v_2 on the *y* coordinate of the object

The simulation purpose is to determine the position of the object by measuring two voltages. In order to collect data for approximation, two voltages of photocells are measured in a number of reference positions of the object. The authors of [8] used a table tennis ball for these experiments. In total there are 67 sets of measurements presented in the files ball_p.txt and ball_t.txt in the archive at the link http://hagan.okstate.edu/CaseStudyData.zip. Each point in graph 18 represents a voltage measurement in a calibration position. Coordinates are measured in inches and voltages are in volts. Flat areas of 0 volts for each curve occur where the shadow of the ball completely covers the sensor. If the shadow was large enough to cover both sensors at the same time, we will not be able to restore the coordinate from the voltage.

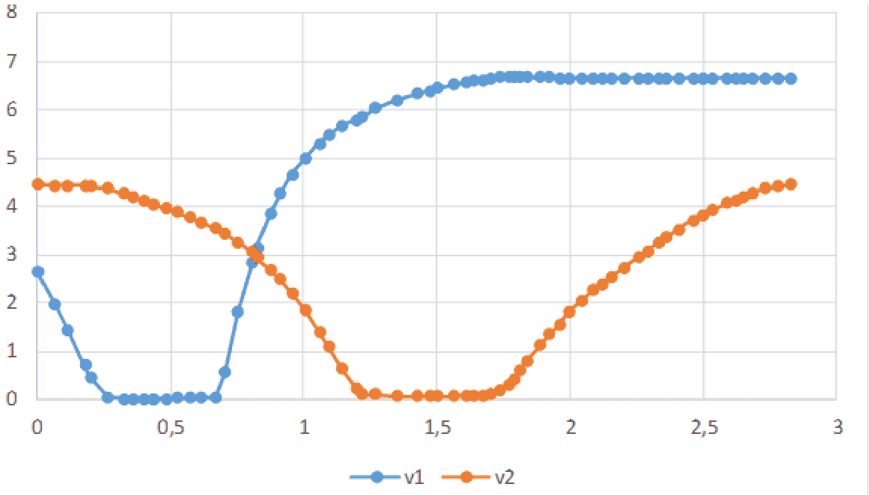


Fig. 18. The dependence of the voltages v_1, v_2 on the *y* coordinate of the object

To implement the deep learning model, we apply cloud-oriented Google Sheets with the Solver addition according to the technique developed in [16].

In order to determine the coordinate of the object, we will build a four-layer neural network with the architecture shown in fig. 19:

- input layer is a two-dimensional arithmetic vector (x_1, x_2), with the components are the corresponding measured voltages v_1, v_2, normalized according to the network activation function;
- the first hidden layer will have dimension 5 and is described by the vector ($h_1^{(1)}$, $h_2^{(1)}$, $h_3^{(1)}$, $h_4^{(1)}$, $h_5^{(1)}$);
- the second hidden layer will have dimension 3 and is described by the vector ($h_1^{(2)}$, $h_2^{(2)}$, $h_3^{(2)}$);
- output layer is y_{norm} value normalized according to the network activation function.

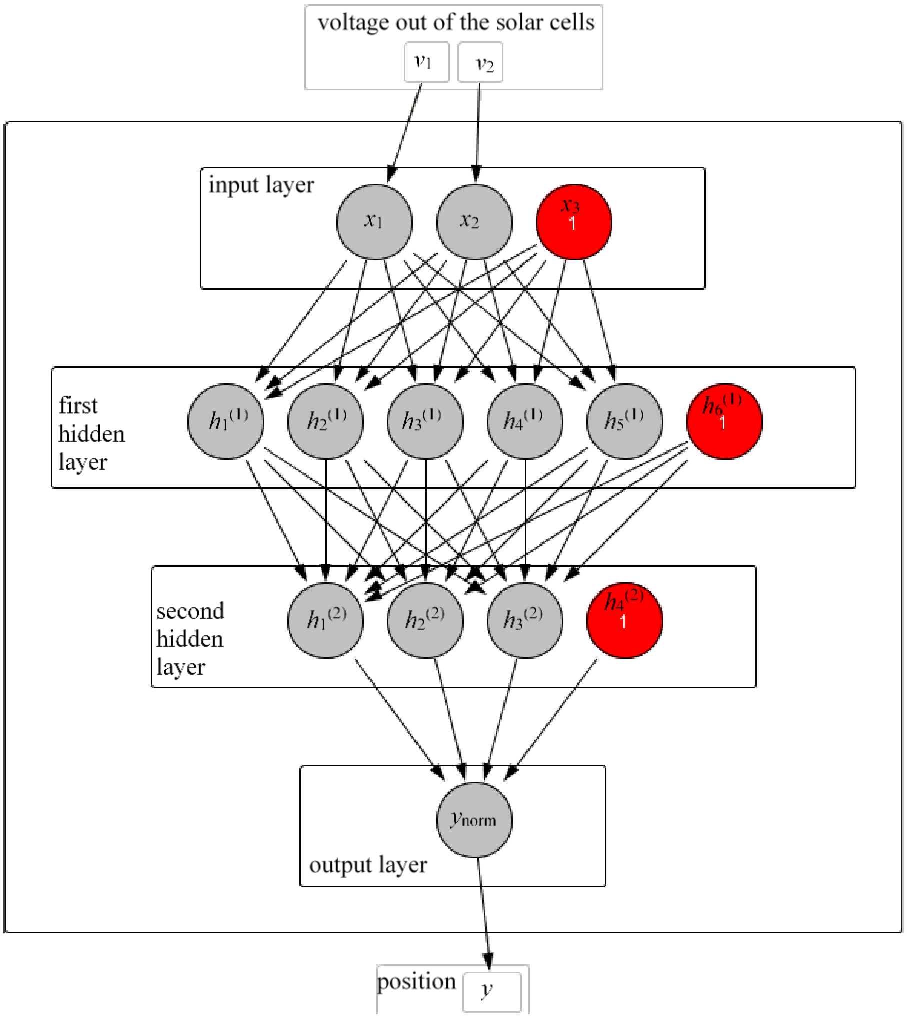
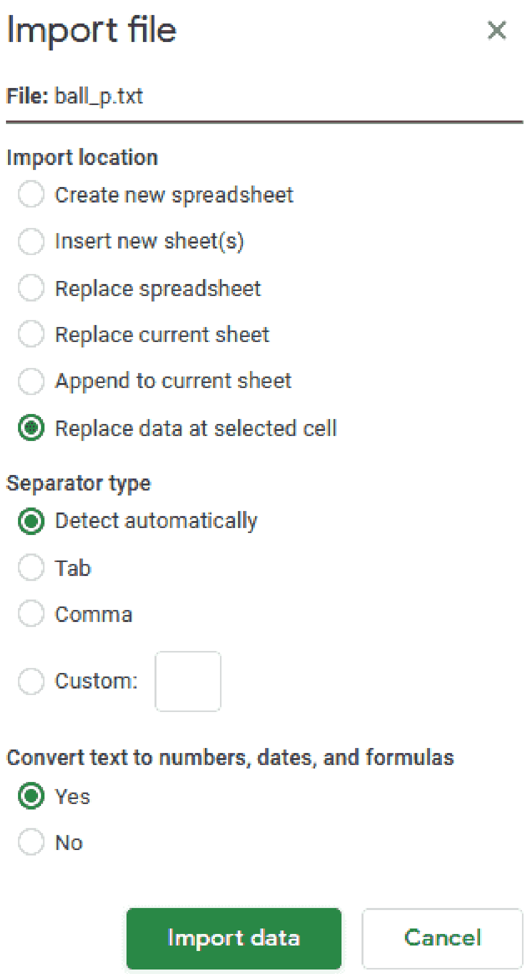


Fig. 19. Neural network architecture for solving the problem of determining the position of an object

Displacement (bias) neurons are added to the neurons of the input and hidden layers. Bias neurons value is always equal to 1 (in fig. 19 they are marked in red). A feature of bias neurons is that they do not have input synapses, and therefore cannot be located on the output layer.

First, enter the data of the measured voltages in the spreadsheets. Since the data is presented in a text file, we use the function of importing data into spreadsheets (fig. 20).



Import file

File: ball_p.txt

Import location

○ Create new spreadsheet
○ Insert new sheet(s)
○ Replace spreadsheet
○ Replace current sheet
○ Append to current sheet
● Replace data at selected cell

Separator type

● Detect automatically
○ Tab
○ Comma
○ Custom: □

Convert text to numbers, dates, and formulas

● Yes
○ No

Fig. 20. Import data to spreadsheet

As a result of import, the following values are entered in the table cells:

B1:BP1	v_1 voltage output data
B2:BP2	v_2 voltage output data
B3:BP3	y coordinate

For convenience of processing, we transpose the obtained data and put them in cells A10:A76 for v_1, B10:B76 for v_2, C10:C76 for y. To do this, we add the following values to the table cells:

A8	*data*
A9	v_1
B9	v_2
C9	y
A10	=TRANSPOSE(B1:BP3)

Based on the fact that the constructed neural network will have a polar activation function, all values at the network input should be normalized (reduced to the range [0; 1]). At the output of the network, you must perform the reverse operation – denormalization.

Normalizing is performed for each column separately. To do this, we find the minimum and maximum values for them by entering the following formulas in the cells:

E5	v_1
F5	v_2
G5	y
D6	*max*
D7	*min*
E6	=max(A10:A76)
E7	=min(A10:A76)

Further, the range E6:E7 is copied to F6:G7.

The essence of normalizing is easy to understand by the expression (5):

$$\text{normalized value} = \frac{\text{input value-minimum value}}{\text{maximum value-minimum value}}. \quad (5)$$

With this approach, the minimum value is normalized to 0, and the maximum up to 1. The normalized voltage values must be feed to the input layer of the neural network:

E8	*input layer*
E9	x_1
F9	x_2
E10	=(A10-E$7)/(E$6-E$7)

Cell E10 is distributed in the range E10:G76.

In accordance with the selected architecture of the neural network, add to the 2 neurons of the input layer a bias neuron. To do this, insert its name (x_3), in cell G9, and its value (1) in the range G10: G76. At this stage, the input layer is formed in the form of a signal vector (x_1, x_2, x_3).

The next step is to transmit the signal from the input layer of the neural network to the first hidden one. To determine the signal power, it is necessary to have weights of the neural network. Denote by:

- $w_{ij}^{xh(1)}$ the weight coefficient of the synapse, which connects the neuron x_i (i = 1, 2, 3) of the input layer with the neuron $h_j^{(1)}$ (j = 1, 2, ..., 5) of the first hidden layer;

- $w_{kp}{}^{h(1)h(2)}$ the weight coefficient of the synapse, which connects the neuron $h_k{}^{(1)}$ (k = 1, 2, ..., 6) of the first hidden layer to the neuron $h_p{}^{(2)}$ (p = 1, 2, 3) of the second hidden layer;
- $w_{dq}{}^{h(2)y}$ the weight coefficient of the synapse, which connects the neuron $h_d{}^{(2)}$ (d = 1, 2, ..., 4) of the second hidden layer with the neuron $y_{\text{norm}q}$ (q = 1) of the output layer.

Then the power of the signal arriving at the neuron $h_j{}^{(1)}$ of the first hidden layer is defined as the scalar product of the signal values at the input layer and the corresponding weighting coefficients. To determine the signal, we will go further to the second hidden layer, we apply the logistic activation function $f(S) = 1/(1+e^{-S})$, where S is the corresponding scalar product. The formulas for determining the signals on the first (6) and second (7) hidden and output (8) layers are:

$$h_j^{(1)} = f\left(\sum_{i=1}^{2+1} x_i w_{ij}^{xh(1)}\right), \tag{6}$$

$$h_p^{(2)} = f\left(\sum_{k=1}^{5+1} h_k^{(1)} w_{kp}^{h(1)h(2)}\right), \tag{7}$$

$$y_{\text{norm}q} = f\left(\sum_{d=1}^{3+1} x_d w_{dq}^{h(2)y}\right). \tag{8}$$

Accordingly, it is necessary to create three matrices:

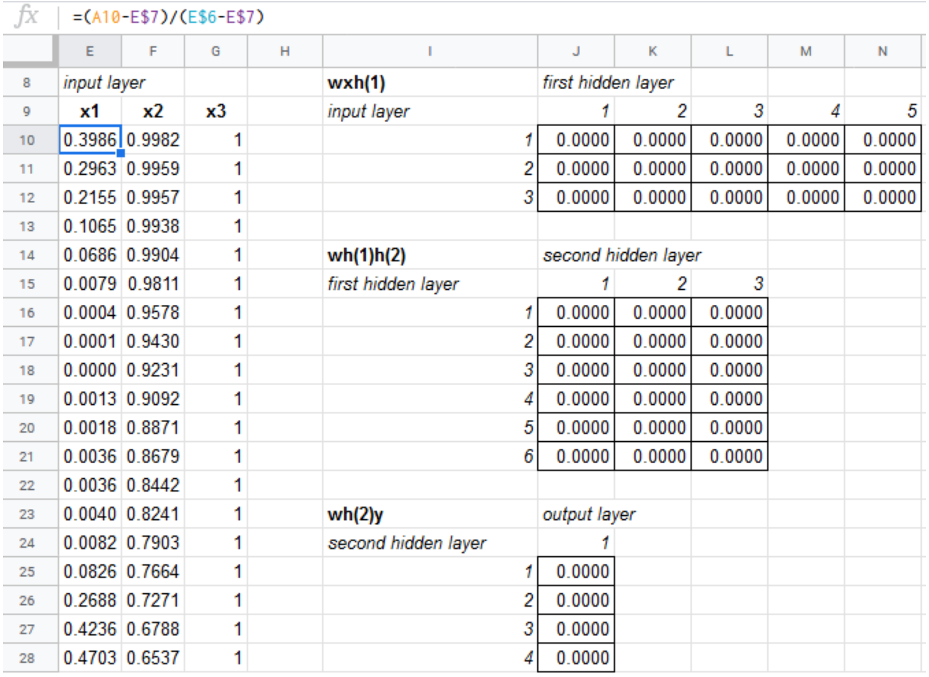
- The $w^{xh(1)}$ matrix of size 3×5 contains the weights connections of 3 neurons of the input layer (the first two contain normalized stress values, and the third is the bias neuron) with the neurons of the first hidden layer;
- The $w^{h(1)h(2)}$ matrix of size 6×3 contains the weights connections of 6 neurons of the first hidden layer (of which five are calculated, and the sixth is the bias neuron) with the neurons of the second hidden layer;
- The $w^{h(2)y}$ matrix of size 4×1 contains the weights connections of 4 neurons of the second hidden layer (of which three are calculated, and the fourth is the displacement neuron) with the neurons of the output layer.

For an "unlearned" neural network, the initial values of the weights can be set either randomly, or left undefined, or equal to zero. To implement the last method, fill the cells with these values:

I8	$w^{xh(1)}$
I9	*input layer*
J8	*first hidden layer*
J9	1
K9	=J9+1
I10	1
I11	=I10+1
J10	0
I14	$w^{h(1)h(2)}$
I15	*first hidden layer*
J14	*second hidden layer*
J15	1

K15	=J15+1
I16	1
I17	=I16+1
J16	0
I23	$w^{h(2)y}$
I24	*second hidden layer*
J23	*output layer*
J10	1
I25	1
I26	=I25+1
J25	0

To create matrices, it is necessary to copy K9 cell to the range L9:N9, I11 – to I12, J10 – to J10:N12, K15 – to L15, I17 – to I18:I21, J16 – to J16:L21, I26 – to I27:I28, J25 – to J26:J28 (fig. 21).



=(A10-E$7)/(E$6-E$7)

	E	F	G	H	I	J	K	L	M	N
8	*input layer*				**wxh(1)**	*first hidden layer*				
9	**x1**	**x2**	**x3**		*input layer*	*1*	*2*	*3*	*4*	*5*
10	0.3986	0.9982	1		*1*	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.2963	0.9959	1		*2*	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.2155	0.9957	1		*3*	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.1065	0.9938	1							
14	0.0686	0.9904	1		**wh(1)h(2)**	*second hidden layer*				
15	0.0079	0.9811	1		*first hidden layer*	*1*	*2*	*3*		
16	0.0004	0.9578	1		*1*	0.0000	0.0000	0.0000		
17	0.0001	0.9430	1		*2*	0.0000	0.0000	0.0000		
18	0.0000	0.9231	1		*3*	0.0000	0.0000	0.0000		
19	0.0013	0.9092	1		*4*	0.0000	0.0000	0.0000		
20	0.0018	0.8871	1		*5*	0.0000	0.0000	0.0000		
21	0.0036	0.8679	1		*6*	0.0000	0.0000	0.0000		
22	0.0036	0.8442	1							
23	0.0040	0.8241	1		**wh(2)y**	*output layer*				
24	0.0082	0.7903	1		*second hidden layer*	*1*				
25	0.0826	0.7664	1		*1*	0.0000				
26	0.2688	0.7271	1		*2*	0.0000				
27	0.4236	0.6788	1		*3*	0.0000				
28	0.4703	0.6537	1		*4*	0.0000				

Fig. 21. Fragment of a spreadsheet after normalizing input data and creating matrices of weighting coefficients

To calculate the scalar product of a row vector of input layer values by a column vector of the matrix of weight coefficients $w^{xh(1)}$, it is advisable to use the matrix multiplication function:

P8	*first hidden layer*
P9	$h_1^{(1)}$
Q9	$h_2^{(1)}$

R9	$h_3^{(1)}$
S9	$h_4^{(1)}$
T9	$h_5^{(1)}$
U9	$h_6^{(1)}$
P10	=1/(1+exp(-mmult($E10:$G10,J$10:J$12)))
U10	1

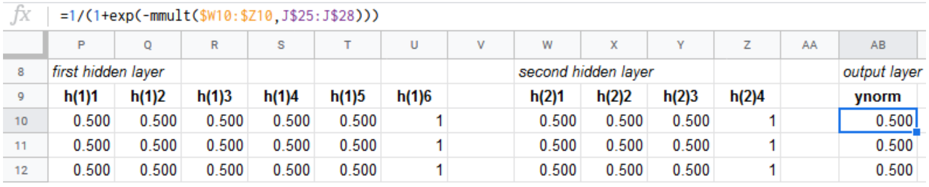
Next, copy cell P10 to the range P10: T76, and U10 to U11: U76.

Given that all elements of the matrix of weight coefficients $w^{xh(1)}$ are initially equal to zero, after copying the formulas, all measured elements of the hidden layer are equal to 0.5.

Similarly, we perform calculations of the elements of the second hidden and output layers:

W8	*second hidden layer*
W9	$h_1^{(2)}$
X9	$h_2^{(2)}$
Y9	$h_3^{(2)}$
Z9	$h_4^{(2)}$
W10	=1/(1+exp(-mmult($P10:$U10,J$16:J$21)))
Z10	1
AB8	*output layer*
AB9	y_{norm}
AB10	=1/(1+exp(-mmult($W10:$Z10,J$25:J$28)))

Next, copy cell W10 to the range W10:Y76, Z10 – to Z11:Z76, AB10 – to AB11:AB76 (fig. 22).



fx =1/(1+exp(-mmult($W10:$Z10,J$25:J$28)))

	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
8	first hidden layer							second hidden layer					output layer
9	h(1)1	h(1)2	h(1)3	h(1)4	h(1)5	h(1)6		h(2)1	h(2)2	h(2)3	h(2)4		ynorm
10	0.500	0.500	0.500	0.500	0.500	1		0.500	0.500	0.500	1		0.500
11	0.500	0.500	0.500	0.500	0.500	1		0.500	0.500	0.500	1		0.500
12	0.500	0.500	0.500	0.500	0.500	1		0.500	0.500	0.500	1		0.500

Fig. 22. Fragment of the spreadsheet after calculating the initial values of the weighting coefficients of the hidden and output layer

To obtain the result y_{calc} from the normalized value of the output layer, it is necessary to calculate it by the formula inverse to the original:

output value = minimum value + normalized value * (maximum value – minimum value)

To do this, enter the following values in the table cells:

AD8	*result*
AD9	y_{calc}
AD10	=$G$7+AB10*($G$6-$G$7)

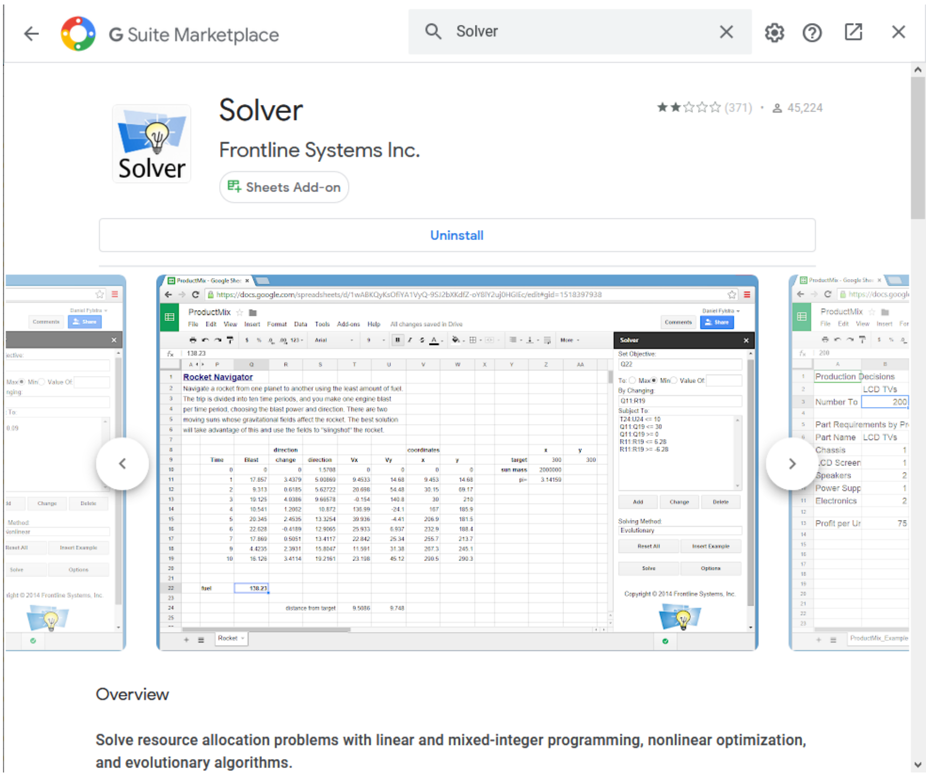
Next, copy cell AD10 to the range AD11: AD76.

Neural network training takes place by varying weights so that with each training step, the difference between the calculated values of y_{calc} and the desired (reference) values of y is reduced. To determine the difference between the calculated and the reference output vectors, we calculate the squares of the deviations and their sum:

AF8	*squared deviation*
AH8	*sum*
AF9	$(y-y_{\text{calc}})^2$
AH9	S
AF10	=(C10-AD10)^2
AH10	=sum(AF10:AF76)

Next, copy cell AF10 to the range AF11: AF76. Cell AH10 contains the sum of the squared deviations.

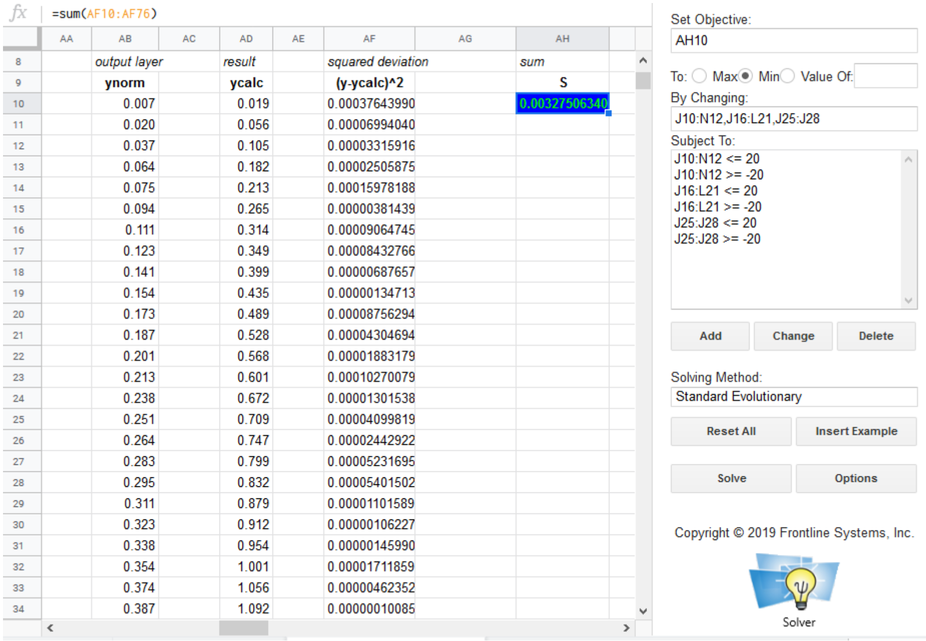
Within this formulation of training, the neural network can be considered as an optimization problem in which the objective function (the sum of the squared deviations in the cell AH10) should be minimized by varying the weights of the matrices $w^{xh(1)}$ (range J10:N12), $w^{h(1)h(2)}$ (range J16:L21) and $w^{h(2)y}$ (range J25:J28). To solve this problem, standard Google Sheets tools are not enough, so you need to install the Solver add-on by choosing Add-ons → Get add-ons ... (fig. 23).



Solve resource allocation problems with linear and mixed-integer programming, nonlinear optimization, and evolutionary algorithms.

Fig. 23. Install the Solver Google Sheets add-on

In fig. 24 shows the settings of the Solver add-on for solving the task: the objective function (Set Objective) is minimized (To: Min) by changing the values (By Changing) of the weighting matrices in the range (Subject To) from -20 to +20 using one of the optimization methods (Solving Method).



=sum(AF10:AF76)

	AA	AB	AC	AD	AE	AF	AG	AH
8		output layer		result		squared deviation		sum
9		ynorm		ycalc		(y-ycalc)^2		S
10		0.007		0.019		0.00037643990		0.00327506340
11		0.020		0.056		0.00006994040		
12		0.037		0.105		0.00003315916		
13		0.064		0.182		0.00002505875		
14		0.075		0.213		0.00015978188		
15		0.094		0.265		0.00000381439		
16		0.111		0.314		0.00009064745		
17		0.123		0.349		0.00008432766		
18		0.141		0.399		0.00000687657		
19		0.154		0.435		0.00000134713		
20		0.173		0.489		0.00008756294		
21		0.187		0.528		0.00004304694		
22		0.201		0.568		0.00001883179		
23		0.213		0.601		0.00010270079		
24		0.238		0.672		0.00001301538		
25		0.251		0.709		0.00004099819		
26		0.264		0.747		0.00002442922		
27		0.283		0.799		0.00005231695		
28		0.295		0.832		0.00005401502		
29		0.311		0.879		0.00001101589		
30		0.323		0.912		0.00000106227		
31		0.338		0.954		0.00000145990		
32		0.354		1.001		0.00001711859		
33		0.374		1.056		0.00000462352		
34		0.387		1.092		0.00000010085		

Set Objective: AH10

To: ○ Max ● Min ○ Value Of:

By Changing: J10:N12,J16:L21,J25:J28

Subject To:
J10:N12 <= 20
J10:N12 >= -20
J16:L21 <= 20
J16:L21 >= -20
J25:J28 <= 20
J25:J28 >= -20

Solving Method: Standard Evolutionary

Copyright © 2019 Frontline Systems, Inc.

Fig. 24. Optimization result for selected Solver parameters

To reduce the sum of squared deviations, Solver can be called up many times: it is advisable to experiment with the combined use of various optimization methods, changing the limits of variation of the weight coefficients. In this case, it is not necessary to try to prove the value of the sum of the distances to zero – this may be more than a (rather small) value.

For clarity of the simulation results, it is advisable to construct a graph of the ratio of experimentally obtained (measured) y values and calculated (approximated) y_{calc} (fig. 25). The quality of approximation is determined by the degree of deviation of the graph points from the beam, divides the first quadrant in half. From the figure and the sum of the squares of deviations obtained after optimization, we can conclude that the constructed neural network model is adequate.

To test the model limitations, we propose the next task: using given in the table 2 values of voltages v_1, v_2, calculate the y coordinate of the object, and explain the choice of voltages and obtained result.

Table 2. Model testing

No	v_1	v_2	y_{calc}	Explanation
1	0.000	4.000	0.467	Measurement only with the second sensor. The result is adequate because the pair v_1, v_2 has a match in the test data set
2	0.000	1.000	1.366	Measurement only with the second sensor. The result is inadequate

No	*v*1	*v*2	y_{calc}	Explanation
3	5.000	0.000	1.187	Measure only with the first sensor. The result is inadequate
4	1.000	0.000	1.368	Measure only with the first sensor. The result is inadequate
5	6.660	3.800	2.493	The data set is on the trend line. The result is adequate
6	3.000	3.000	0.809	Intersection of lines 1, 2. The result is adequate
7	0.000	0.000	1.374	Minimal values. The result is inadequate due to incorrect regularity
8	10.000	10.000	2.825	Numbers beyond the range of measured values. The result is the maximum value of the position

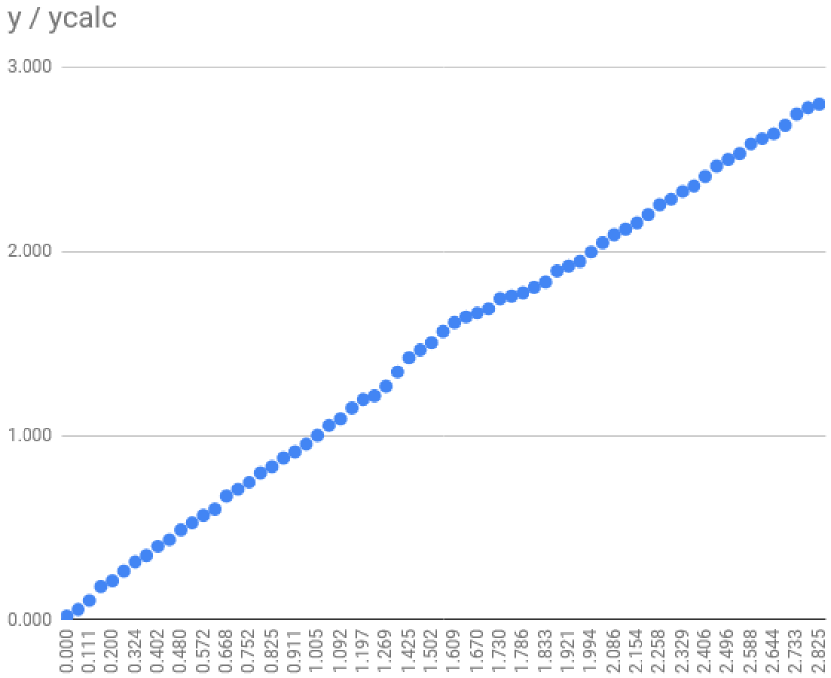


Fig. 25. Graph of measured *y* values and approximated y_{calc}

3 Conclusions

Thus, in the process of formation of the general professional component of the competence of a bachelor of electromechanics in the modeling of technical objects, it is advisable to use the following software for mobile Internet devices:

- cloud-based spreadsheets as modeling tools (including neural network);
- visual modeling systems as a tools of technical objects structural modeling;
- mobile computer mathematical systems used at all stages of modeling;
- mobile communication tools for organizing joint modeling activities.

References

1. CeCILL FREE SOFTWARE LICENSE AGREEMENT. Version 2.1. http://www.cecill.info/licences/Licence_CeCILL_V2.1-en.html (2013). Accessed 25 Oct 2016
2. Champion, C.: GNURoot Debian. Google Play. https://play.google.com/store/apps/details?id=com.gnuroot.debian (2018). Accessed 25 Oct 2018
3. Debray, Y., Bignier, P.: Webinar: Application Development with Scilab = Scientific & Engineering Application: Scilab & Scilab Cloud. https://www.youtube.com/watch?v=MaPKnUIEwoY (2016). Accessed 21 Mar 2017
4. ESI Group: History | www.scilab.org. https://www.scilab.org/about/company/history (2018). Accessed 25 Oct 2018
5. ESI Group: Home Page | www.scilab.org. http://www.scilab.org (2018). Accessed 25 Oct 2018
6. FOSSEE: Xcos. http://xcos.fossee.in (2019). Accessed 25 Oct 2019
7. GitHub - androportal/APL-apk: Aakash Programming Lab(APL). Provides programming environment for C, C++, Python and Scilab. https://github.com/androportal/APL-apk (2013). Accessed 17 Aug 2015
8. Hagan, M.T., Demuth, H.B., Beale, M.H., De Jesús, O.: Neural Network Design, 2nd edn. http://hagan.okstate.edu/NNDesign.pdf (2014). Accessed 17 Aug 2015
9. Inria, the French research institute for digital sciences. https://www.inria.fr/en (2019). Accessed 25 Oct 2019
10. Inria: Scilab takes off on its own. https://www.inria.fr/en/news/news-from-inria/scilab (2009). Accessed 25 Oct 2019
11. Ivanchura, V.I., Prokopev, A.P.: Optimizatciia slediashchei sistemy avtomaticheskogo upravleniia (Optimization of the tracking system of automatic control). Vestnik Sibirskogo gosudarstvennogo aerokosmicheskogo universiteta imeni akademika M. F. Reshetneva 5(38), 44–49 (2011)
12. Kanivets, O.V., Kanivets, I.M., Kononets, N.V., Gorda, T.M., Shmeltser, E.O.: Augmented reality mobile application developments for help to performance tasks from projection drawing. In: Kiv, A.E., Shyshkina, M.P. (eds.) Proceedings of the 2nd International Workshop on Augmented Reality in Education (AREdu 2019), Kryvyi Rih, Ukraine, March 22, 2019. CEUR Workshop Proceedings **2547**, 262–273. http://ceur-ws.org/Vol-2546/paper19.pdf (2020). Accessed 10 Feb 2020
13. Kumar, R.S.: W3 Scilab. https://sourceforge.net/projects/w3scilab (2013). Accessed 23 June 2014
14. Markova, O., Semerikov, S., Popel, M.: CoCalc as a Learning Tool for Neural Network Simulation in the Special Course "Foundations of Mathematic Informatics". In: Ermolayev, V., Suárez-Figueroa, M.C., Yakovyna, V., Kharchenko, V., Kobets, V., Kravtsov, H., Peschanenko, V., Prytula, Ya., Nikitchenko, M., Spivakovsky A. (eds.) Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2018), Kyiv, Ukraine, 14-17 May 2018, vol. II: Workshops. CEUR Workshop Proceedings **2104**, 338–403. http://ceur-ws.org/Vol-2104/paper_204.pdf (2018). Accessed 25 Oct 2018
15. Markova, O.M., Semerikov, S.O., Striuk, A.M., Shalatska, H.M., Nechypurenko, P.P., Tron, V.V.: Implementation of cloud service models in training of future information technology specialists. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 6th Workshop on Cloud Technologies in Education (CTE 2018), Kryvyi Rih, Ukraine, December 21,

2018. CEUR Workshop Proceedings **2433**, 499–515. http://ceur-ws.org/Vol-2433/paper34.pdf (2019). Accessed 25 Oct 2019

16. Markova, O.M.: Cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities. Dissertation, Kryvyi Rih State Pedagogical University (2018)
17. Merzlykin, P.V., Popel, M.V., Shokaliuk, S.V.: Services of SageMathCloud environment and their didactic potential in learning of informatics and mathematical disciplines. In: Semerikov, S.O., Shyshkina, M.P. (eds.) Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017), Kryvyi Rih, Ukraine, April 28, 2017. CEUR Workshop Proceedings **2168**, 13–19. http://ceur-ws.org/Vol-2168/paper3.pdf (2018). Accessed 21 Mar 2019
18. Modlo, E.O., Semerikov, S.O.: Development of SageMath filter for Moodle. New computer technology XII(Cloud technologies in education), 233–243 (2014)
19. Modlo, Ye.O., Semerikov, S.O., Bondarevskyi, S.L., Tolmachev, S.T., Markova, O.M., Nechypurenko, P.P.: Methods of using mobile Internet devices in the formation of the general scientific component of bachelor in electromechanics competency in modeling of technical objects. In: Kiv, A.E., Shyshkina, M.P. (eds.) Proceedings of the 2nd International Workshop on Augmented Reality in Education (AREdu 2019), Kryvyi Rih, Ukraine, March 22, 2019. CEUR Workshop Proceedings **2547**, 217–240. http://ceur-ws.org/Vol-2546/paper16.pdf (2020). Accessed 10 Feb 2020
20. Modlo, Ye.O., Semerikov, S.O., Nechypurenko, P.P., Bondarevskyi, S.L., Bondarevska, O.M., Tolmachev, S.T.: The use of mobile Internet devices in the formation of ICT component of bachelors in electromechanics competency in modeling of technical objects. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 6th Workshop on Cloud Technologies in Education (CTE 2018), Kryvyi Rih, Ukraine, December 21, 2018. CEUR Workshop Proceedings **2433**, 413–428. http://ceur-ws.org/Vol-2433/paper28.pdf (2019). Accessed 25 Oct 2019
21. Modlo, Ye.O., Semerikov, S.O., Shmeltzer, E.O.: Modernization of Professional Training of Electromechanics Bachelors: ICT-based Competence Approach. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings **2257**, 148–172. http://ceur-ws.org/Vol-2257/paper15.pdf (2018). Accessed 21 Mar 2019
22. Modlo, Ye.O., Semerikov, S.O.: Xcos on Web as a promising learning tool for Bachelor's of Electromechanics modeling of technical objects. In: Semerikov, S.O., Shyshkina, M.P. (eds.) Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017), Kryvyi Rih, Ukraine, April 28, 2017. CEUR Workshop Proceedings **2168**, 34–41. http://ceur-ws.org/Vol-2168/paper6.pdf (2018). Accessed 21 Mar 2019
23. Modlo, Ye.O.: Do vyznachennia poniattia mobilnoho Internet-prystroiu (To the Mobile Internet Device definition). In: Bykov, V.Yu, Spirin, O.M. (eds.) Zbirnyk materialiv III Vseukrainskoi naukovo-praktychnoi konferentsii molodykh uchenykh "Naukova molod-2015", 10 hrudnia 2015 roku, pp. 37–38. IITZN NAPN Ukrainy, Kyiv. http://lib.iitta.gov.ua/704728/1/%D0%97%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA%20%D0%BA%D0%BE%D0%BD%D1%84_%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%BC%D0%BE%D0%BB%D0%BE%D0%B4%D1%8C%202015_1.pdf (2015). Accessed 21 Mar 2020
24. Omatu, S., Khalid, M.B., Yusof, R.: Neiroupravlenie i ego prilozheniia (Neuro-Control and its Applications). Radiotekhnika, Moscow (2000)

25. Popel, M.V., Shokalyuk, S.V., Shyshkina, M.P.: The Learning Technique of the SageMathCloud Use for Students Collaboration Support. In: Ermolayev, V., Bassiliades, N., Fill, H.-G., Yakovyna, V., Mayr, H.C., Kharchenko, V., Peschanenko, V., Shyshkina, M., Nikitchenko, M., Spivakovsky, A. (eds.) 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2017), Kyiv, Ukraine, 15-18 May 2017. CEUR Workshop Proceedings **1844**, 327–339. http://ceur-ws.org/Vol-1844/10000327.pdf (2017). Accessed 21 Mar 2019
26. Rassovytska, M.V., Striuk, A.M.: Mechanical Engineers' Training in Using Cloud and Mobile Services in Professional Activity. In: Ermolayev, V., Bassiliades, N., Fill, H.-G., Yakovyna, V., Mayr, H.C., Kharchenko, V., Peschanenko, V., Shyshkina, M., Nikitchenko, M., Spivakovsky, A. (eds.) 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2017), Kyiv, Ukraine, 15-18 May 2017. CEUR Workshop Proceedings **1844**, 348–359. http://ceur-ws.org/Vol-1844/10000348.pdf (2017). Accessed 21 Mar 2019
27. Sage Cell Server [Electronic resource] / Access mode : http://sagecell.sagemath.org/
28. Scilab – Wikipedia. https://uk.wikipedia.org/w/index.php?title=Scilab&oldid=24491902 (2019). Accessed 17 Feb 2019
29. Scilab Enterprises S.A.S.: Scilab is recognized as having educational value by the French Department of Education: Press Release. Rocquencourt, July 26th 2011. https://www.scilab.org/content/download/514/4351/file/CP_Scilab_26072011_eng.pdf (2011). Accessed 17 Aug 2015
30. Scilab on Aakash | Scilab.in. http://www.scilab.in/scilab-on-aakash (2012). Accessed 23 Jun 2018
31. Scilab Online – rollApp. https://www.rollapp.com/app/scilab (2018). Accessed 25 Oct 2018
32. Semerikov, S.O., Teplytskyi, I.O., Yechkalo, Yu.V., Kiv, A.E.: Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings **2257**, 122–147. http://ceur-ws.org/Vol-2257/paper14.pdf (2018). Accessed 30 Nov 2018
33. Semerikov, S.O., Teplytskyi, I.O., Yechkalo, Yu.V., Markova, O.M., Soloviev, V.N., Kiv, A.E.: Computer Simulation of Neural Networks Using Spreadsheets: Dr. Anderson, Welcome Back. In: Ermolayev, V., Mallet, F., Yakovyna, V., Kharchenko, V., Kobets, V., Korniłowicz, A., Kravtsov, H., Nikitchenko, M., Semerikov, S., Spivakovsky, A. (eds.) Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2019), Kherson, Ukraine, June 12-15 2019, vol. II: Workshops. CEUR Workshop Proceedings **2393**, 833–848. http://ceur-ws.org/Vol-2393/paper_348.pdf (2019). Accessed 30 Jun 2019
34. Sobol, I.M.: Metod Monte-Karlo (Monte Carlo Method). Nauka, Moscow (1968)
35. Tkachuk, V.V., Shchokin, V.P., Tron, V.V.: The Model of Use of Mobile Information and Communication Technologies in Learning Computer Sciences to Future Professionals in Engineering Pedagogy. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings **2257**, 103–111. http://ceur-ws.org/Vol-2257/paper12.pdf (2018). Accessed 21 Mar 2019
36. Web application – Scilab.io. https://scilab.io/services/development/web-application (2016). Accessed 21 Mar 2017