Using Easy Java Simulations in Computer Supported Control Engineering Education

Milica B. Naumović, Nataša Popović, and Božidar Popović

Abstract—**This paper presents the software tool Easy Java Simulations (EJS) designed to build the interactive applications in Java. It has been created to be used by students, professors and scientists without special programming skills, and it is particularly suitable for use in engineering education. The problem known in optimal control theory as brachistochrone problem is considered confirming the ability of EJS tool to build networked solutions with high degree of interactivity and visualization.**

Index Terms—**Control engineering education, distance learning, optimal control, simulation.**

I. Introduction

THE rapid development of information and communication technologies allowed the educational process to become independent of time and space. This facilitated the introduction of distance learning in regular curriculum and simplified educational process both for teachers and students.

Interactive scientific simulations play important role in control systems education. They help students to understand the behavior of system or phenomenon under study in an appropriate way. Virtual and remote laboratories help them even more, since students can perform their laboratory work from any place at any time. Simulations, virtual and remote laboratories intended to be used in distant learning courses are located on the university servers. Distant users access them via Internet using their PCs or mobile devices and perform lab work through on-line experimentation services. This flexible education paradigm is shown in Fig. 1.

Sophisticated interactive simulations are usually created in Java programming language. Development of such simulations demands a lot of time and advanced programming skills. This means that in general only professional programmers can develop them. Scientists, professors and students, who were not professional programmers, were unable to develop them, although they had all necessary theoretical knowledge and understanding of system or phenomenon to be simulated. A group of scientists recognized this as a problem, and started to work on a software tool that would meet the requirements of non-professional programmers in developing useful and effective interactive simulations. This resulted in new software tool based on Java that was, because of its simplicity and ease in use, symbolically called Easy Java Simulations (EJS).

Useful interactive simulations, in the form of remote and virtual labs can be created with Easy Java Simulations [1]-[5] in almost every scientific field. The numerous examples of such simulations can be found in [1], [2]. Particularly interesting was to investigate the possibility to use EJS in optimal control theory to demonstrate the famous brachistochrone problem [6]-[8], and its solution. The results of this investigation are presented further.

The layout of the paper is as follows: in Section II we give a brief review of the brachistochrone problem solution that is one of the earliest applications of the calculus of variations; Section III addresses to Easy Java Simulations, as a modeling tool for rapid creation of simulations in Java. A concrete implementation related to the analysis of movement along the

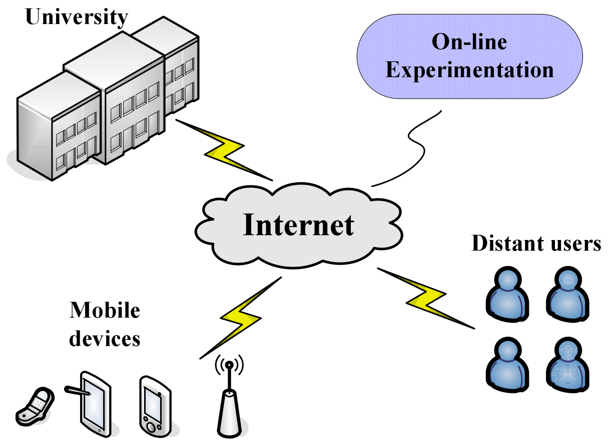


Fig. 1. Scheme of the flexible education paradigm.

This paper was supported in part by project Grant III44004 (2011-2014) financed by Ministry of Education and Science, Republic of Serbia. Some results of this paper were presented at 55th ETRAN Conference, Banja Vrućica, June 6-9, 2011.

Milica B. Naumović is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (phone: +381 18 529 441; fax: +381 18 588 399 ; e-mail: milica.naumovic@elfak.ni.ac.rs).

Nataša Popović is with the Faculty of Electrical Engineering, University of East Sarajevo, East Sarajevo, Bosnia and Herzegovina (e-mail: natasa.popovic@etf.unssa.rs.ba).

Božidar Popović is with the Faculty of Electrical Engineering, University of East Sarajevo, East Sarajevo, Bosnia and Herzegovina (e-mail: bozidar.popovic@etf.unssa.rs.ba).

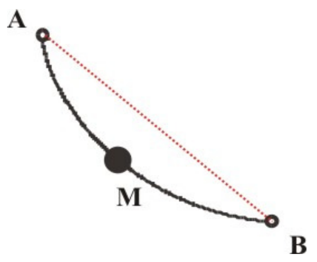


Fig. 2. The Brachistochrone Problem (Acta Eruditorum, June 1696, p. 269).

cycloid is illustrated in Section IV. Finally some conclusions of the work are presented.

II. The Brachistochrone Problem

A. A Problem Formulation and Solution Elements

The brachistochrone problem is one of the oldest problems that in fact initiated efforts towards calculus of variations [6].

Namely, in the June 1696 issue of Acta Eruditorum, Johann Bernoulli posed Invitation to all mathematicians to solve a new problem, which can be simply stated as follows [7]:

If in a vertical plane two points A and B are given, then it is required to specify the orbit AMB of the movable point M, along which it, starting from A with zero speed, and under the influence of its own weight, arrives at B in the shortest possible time.

Set the coordinate system as in Fig. 3. Let us note at some point the position of the particle M with mass m. Its kinetic and potential energy are respectively given by $mg(H-y)$ and $mv^2/2$, and, according to the conservation of energy, their sum is constant, i.e.

$$mg(H-y)+\frac{mv^2}{2}=K. \tag{1}$$

At the beginning, let the material point move without the initial speed $(y=0 \text{ and } v=0)$, and so $mgH=K$. In this way we get

$$v^2=2gy. \tag{2}$$

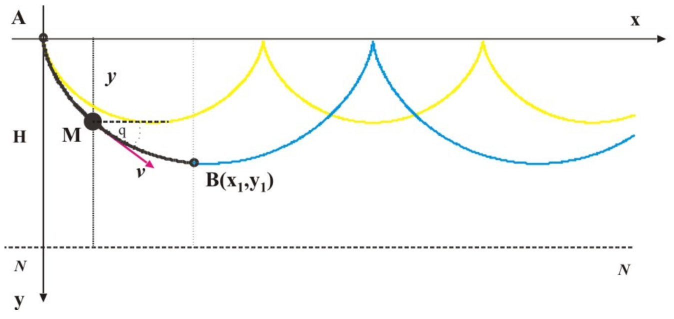


Fig. 3. Geometric interpretation of the brachistohrone problem.

Since $v=\dfrac{\mathrm{d}s}{\mathrm{d}t}$ and $\left(\dfrac{\mathrm{d}s}{\mathrm{d}x}\right)^2=\left(\dfrac{\mathrm{d}y}{\mathrm{d}x}\right)^2+1$, it follows

$$\mathrm{d}t=\frac{\mathrm{d}s}{\sqrt{2gy}}=\frac{\sqrt{1+y'^2}}{\sqrt{2gy}}\mathrm{d}x, \tag{3}$$

where

$$y' \overset{\text{def}}{=} \mathrm{d}y/\mathrm{d}x. \tag{4}$$

The total duration of the movement from A to B to be minimized is then given by

$$T=\frac{1}{\sqrt{2g}}\int_0^{x_1}\sqrt{\frac{1+y'^2}{y}}\,\mathrm{d}x. \tag{5}$$

B. The Variational Calculus – The Necessary Conditions for Minimum

We consider the problem of selecting a continuously differentiable function $x(t)\in\Re$ with respect to the set of all real-valued, continuously differentiable functions, in order to minimize the cost function

$$\mathrm{J}(x)=\int_{t_0}^{t_f}\Phi\left[x(t),\dot{x}(t),t\right]\mathrm{d}t \tag{6}$$

on the interval $\left[t_0,t_f\right]$. Therefore, it is necessary to determine the optimal trajectory $\hat{x}(t)$ from the set of all admissible trajectories $x(t)$. It is assumed that the function Φ is continuous in $x, \dot{x}$, and t, as well as has continuous partial derivatives with respect to x and $\dot{x}$ [6]. Recall that the cost function given in (6) is known as the Lagrange form. An admissible, but not necessarily optimal trajectory $x(t)$ for all $t\in\left[t_0,t_f\right]$ can be expressed as

$$x(t)=\hat{x}(t)+\varepsilon\eta(t), \tag{7}$$
(2)

where ε is a small number, and $\eta(t)$ is a variation in $x(t)$. Applying the theory of the variational calculus the necessary conditions can be developed [6]:

$$\frac{\partial\Phi}{\partial x}-\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\Phi}{\partial\dot{x}}=0, \tag{8}$$

and

$$\frac{\partial\Phi}{\partial\dot{x}}\eta(t)=0, \quad \text{for} \quad t=t_0,t_f. \tag{9}$$

The partial differential equation (8) is commonly known as the Euler-Lagrange (*E-L*) equation, while the associated transversality conditions are given by (9). By this equations a two-point boundary value differential equation is specified which, when solved, determines a candidate for an optimal trajectory in terms of a known Φ.

Some particular cases of *E-L* are:

① Suppose that the EULER function $\Phi\left[x(t),\dot{x}(t),t\right]$ is independent of the variable x (in dynamics this variable is called an ignorable coordinate). Then, *E-L* results in

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\Phi}{\partial\dot{x}}=0 \quad\Rightarrow\quad \frac{\partial\Phi}{\partial\dot{x}}=\text{const.}, \tag{10}$$

which is the principle of conservation of conjugate momentum in dynamics.

② Suppose we consider a time invariant system and the EULER function $\Phi\left[x(t),\dot{x}(t),t\right]$ is independent of the time t. Then we write:

$$\frac{\partial\Phi}{\partial x}-\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\Phi}{\partial\dot{x}}=\frac{\partial\Phi}{\partial x}-\frac{\partial^2\Phi}{\partial x\partial\dot{x}}\dot{x}-\frac{\partial^2\Phi}{\partial^2\dot{x}}\ddot{x}=0, \tag{11}$$

or

$$\dot{x}\frac{\partial\Phi}{\partial x}-\frac{\partial^2\Phi}{\partial x\partial\dot{x}}\dot{x}^2-\frac{\partial^2\Phi}{\partial^2\dot{x}}\dot{x}\ddot{x}=\frac{\mathrm{d}}{\mathrm{d}t}\left[\Phi-\dot{x}\frac{\partial\Phi}{\partial\dot{x}}\right]=0. \tag{12}$$

This means that

$$\Phi-\dot{x}\frac{\partial\Phi}{\partial\dot{x}}=\text{const.}, \tag{13}$$

which is referred to as the conservation of the Hamiltonian. It is in the literature known as the Beltrami identity.

③ If $\Phi\left[x(t),\dot{x}(t),t\right]$ is independent of $\dot{x}$, then EULER-LAGRANGE equation becomes simply

$$\frac{\partial\Phi}{\partial x}=0. \tag{14}$$

C. *The Solution of the Brachistochrone Problem*

In space-relative terms as in solving the brachistochrone problem (5), we have the EULER-LAGRANGE condition

$$\frac{\partial\Phi}{\partial y}-\frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial\Phi}{\partial y'}=0\ . \tag{15}$$

Since the Lagrangian $\Phi\left[y,y',x\right]=\sqrt{\left(1+y'^2\right)/y}$ in (5) is particularly nice since x does not appear explicitly. Therefore, $\partial\Phi/\partial x=0$, and we can immediately use the Beltrami identity

$$\Phi-y'\frac{\partial\Phi}{\partial y'}=C\ , \tag{16}$$

which yields

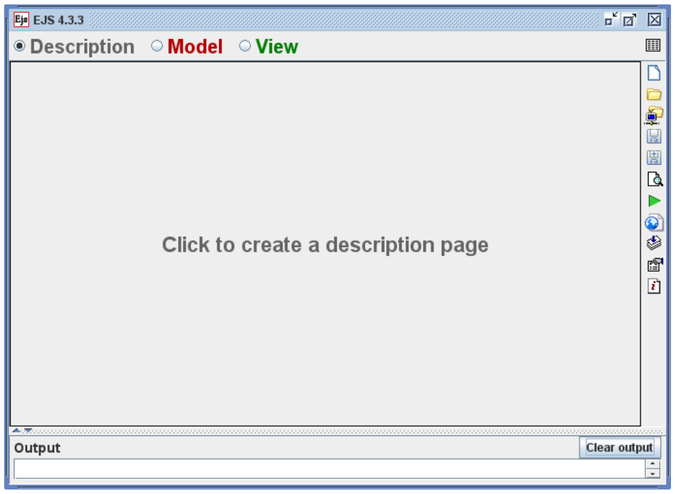


Fig. 4. EJS structure.

$$\frac{1}{\sqrt{2gy}\sqrt{1+y'^2}}=C\,. \tag{17}$$

After squaring both sides and rearranging we can compute

$$\left[1+\left(\frac{\mathrm{d}\,y}{\mathrm{d}\,x}\right)^2\right]y=k^2, \tag{18}$$

where the square of the old constant C has been expressed in terms of a new positive constant $k^2\left(=1/2gC^2\right)$. The equation (18) can easily be solved by the parametric equations

$$\begin{aligned} x&=\tfrac{1}{2}k^2\left(\theta-\sin\theta\right)\\ y&=\tfrac{1}{2}k^2\left(1-\cos\theta\right) \end{aligned}. \tag{19}$$

III. EASY JAVA SIMULATIONS FUNDAMENTALS

Easy Java Simulations is a freeware, open-source tool developed in Java, specifically designed for the creation of interactive dynamic simulations [1]. EJS was developed for the Open Source Physics Project [2], which was established to create and distribute curricular material for physics computation. As a free software tool, EJS is not only used in creating interactive simulations, but also to create virtual and remote laboratories [1], [2]. A very important feature of EJS for the control systems education is its possibility to use Matlab and LabView as external applications. This actually, enabled EJS to be used in remote laboratories development.

EJS architecture is based on the model-view-control paradigm [4]. It means that interactive simulations must consist of the following parts:

- system or phenomenon description using variables and relations among them,
- graphical representation of the system or phenomenon states,

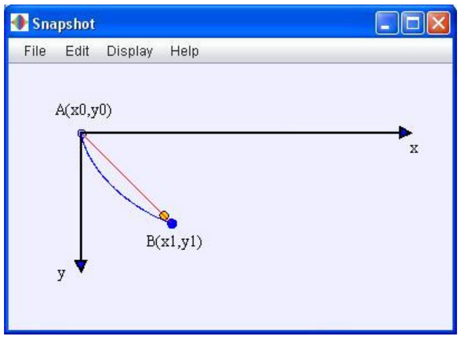


Fig. 5. Snapshot of the first cycloid simulation cycle.

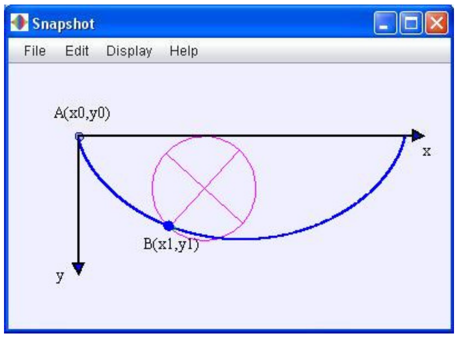
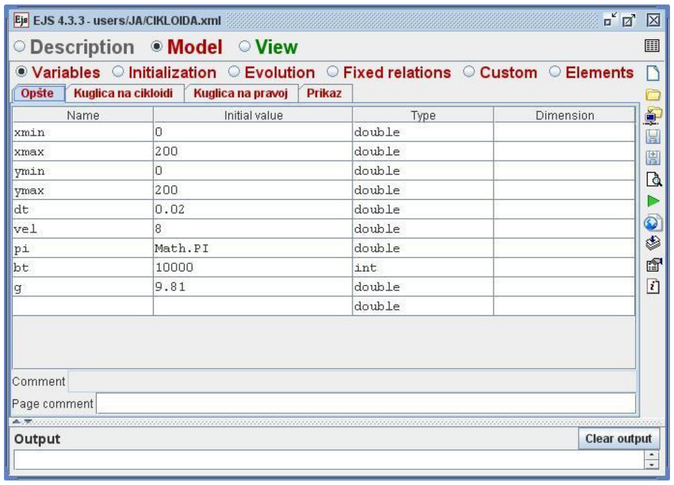


Fig. 6. Snapshot of the second cycloid simulation cycle.

- the specifications of the user's actions to perform on the simulation

Following the model-view-control paradigm, simulations in EJS are structured in two main parts: the Model and the View [3]-[5] (see Fig. 4). Additionally, there is a part named Description which contains the theoretical introduction of simulated system and gives students instructions how to use the simulation. The Model describes the behavior of the system or phenomenon under study using pages with variables, ordinary differential equations, and Java code. The View provides the visualization of the simulation and graphical elements for user interaction. Model and View are interconnected so that any change made in the Model automatically reflects on the View and, vice versa, any user interaction with the simulation automatically modifies the value of responding model variable.


EJS 4.3.3 - users/JA/CIKLOIDA.xml
Description [✓ Model] View
[✓ Variables] Initialization Evolution Fixed relations Custom Elements
Opšte
Kuglica na cikloidi
Kuglica na pravoj
Prikaz

Name	Initial value	Type	Dimension
xmin	0	double	
xmax	200	double	
ymin	0	double	
ymax	200	double	
dt	0.02	double	
vel	8	double	
pi	Math.PI	double	
bt	10000	int	
g	9.81	double	
		double	

Comment
Page comment
Output
Clear output

Fig. 7. EJS Model part.

The Model of the system is created by defining the set of variables that describe the system state and specifying the equations that determine how this state changes with time [5]. Ordinary differential equations are written in EJS ODE editor in a way similar when writing them on a paper. If necessary, additional Java code can be written in order to show, more faithfully, system behavior during simulation execution. Besides that, some user defined classes, methods and actions can be realized writing this additional Java code, especially when the simulated system or phenomenon is complex. The View is created using a built-in library of ready-to-use graphical elements. These elements have the properties which values can be modified to customize the elements' behavior and visual display. In this way it is possible to connect properties of the view's graphical elements with model variables. Graphical elements properties also provide the possibility of evoking user defined actions when user interacts with the simulation.

When both the Model of the system and the View are defined, EJS generates the simulation Java source code, compiles it, packages it into a *.jar* file, and runs it as an independent process. The simulation can also be generated as an applet, so that the simulation execution can be controlled from the web browser [5].

IV. Cycloid Virtual Lab

A. *Building the virtual lab*

The cycloid simulation is designed to encompass two cycles.

The first cycle shows moving of two particles from point A to point B of which the orange one is moving along the straight red line and the blue one along the unknown path. It can be seen that the blue particle reaches point B before the orange one, moving along a curve, Fig. 5.

The second cycle shows the rolling of a circle along the *x*-axis together with the blue particle fixed on it, Fig. 6. This cycle proves that the curve obtained by moving the blue particle in the first cycle is a cycloid. The building of the virtual lab in EJS implies the definition of both the Model and the View.

Defining the Model

The first step in building the simulation is defining the Model as it is given in Fig. 7. Although the Model part of EJS consists of six sections only four of them were used to develop cycloid simulation.

The most important task in defining the Model is to choose model variables in an appropriate way. This means that variables have to be chosen to be easily used to describe particles moving, to obtain clear visual display, and to provide simple user interaction. In the Variable section general variables, variables for the blue particle moving, variables for orange particle moving and variables for simulation visualization are defined. Variables that describe size and ratio of all graphical elements necessary for simulation, constants π and *g* (gravitational acceleration), and number of cycloid points are defined in general variables tab. Variables to define moving of blue particle and circle, and orange particle are defined in two separate tabs and some of them are: blue particle moving period, blue particle acceleration, circle radius, circle center coordinates, kinetic, potential and total energy. The last tab contains variables which determine what graphical elements are to be displayed in a particular simulation cycle, and whether the energy plots are to be visible or not.

The Initialization section contains simulation initial conditions, and the Evolution section contains ordinary differential equations which describe moving of both particles. A simple Java code to control simulation execution is written in the Fixed relations section as well as relations necessary for computation of kinetic, potential and total energy. This Java code determines the duration of simulation cycles and whether the simulation is to be paused or reset. The simulation is paused when the first cycle terminates, and is reset when the second cycle terminates.

Defining the View

Visual display of simulation is necessary for controlling the simulation execution. It provides simulation visualization and user interaction.

The View part of cycloid simulation is divided in drawing frame and plotting panel, Fig. 8. Drawing frame consists of drawing panel and buttons panel. Drawing panel contains all view elements necessary for simulation visualization (particles shape, arrows for *x*-axis and *y*-axis, text, trace for obtaining cycloid, analytic curve for displaying the cycloid,…), and

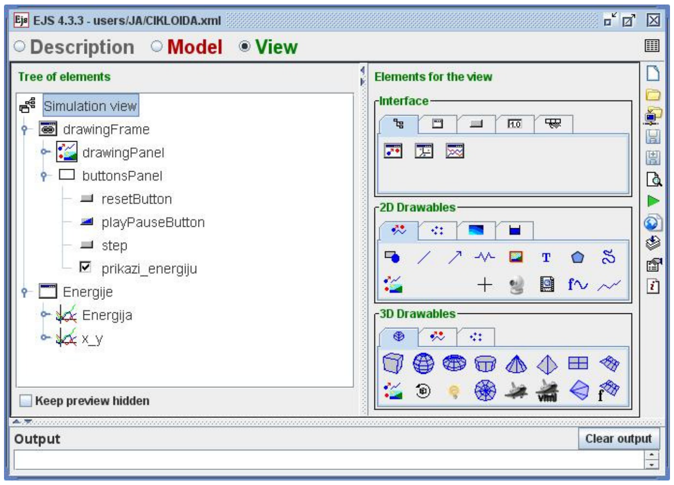
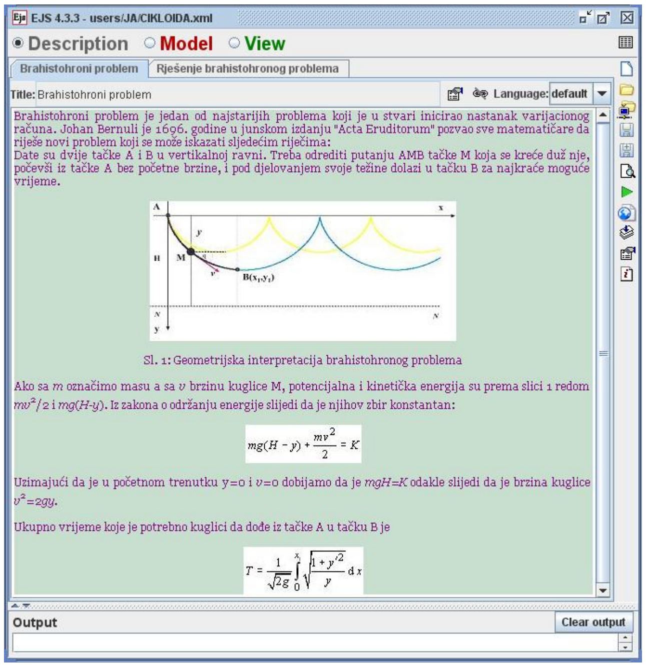


Fig. 8. EJS View panel with elements.



Title: Brahistohroni problem

Brahistohroni problem je jedan od najstarijih problema koji je u stvari inicirao nastanak varijacionog računa. Johan Bernuli je 1696. godine u junskom izdanju "Acta Eruditorum" pozvao sve matematičare da riješe novi problem koji se može iskazati sljedećim riječima:
Date su dvije tačke A i B u vertikalnoj ravni. Treba odrediti putanju AMB tačke M koja se kreće duž nje, počevši iz tačke A bez početne brzine, i pod djelovanjem svoje težine dolazi u tačku B za najkraće moguće vrijeme.

Sl. 1: Geometrijska interpretacija brahistohronog problema

Ako sa m označimo masu a sa v brzinu kuglice M, potencijalna i kinetička energija su prema slici 1 redom $mv^2/2$ i $mg(H\text{-}y)$. Iz zakona o održanju energije slijedi da je njihov zbir konstantan:

$$mg(H-y)+\frac{mv^2}{2}=K$$

Uzimajući da je u početnom trenutku y=0 i v=0 dobijamo da je $mgH=K$ odakle slijedi da je brzina kuglice $v^2=2gy$.

Ukupno vrijeme koje je potrebno kuglici da dođe iz tačke A u tačku B je

$$T=\frac{1}{\sqrt{2g}}\int_0^{x_1}\sqrt{\frac{1+y'^2}{y}}\,dx$$

Fig. 9. EJS Description part.

buttons panel contains buttons for simulation execution control and user-simulation interaction (play/pause, step and reset buttons, check box to turn on and of energy plots). Plotting panel displays energy plots and shows changes of *x* and *y* with time.

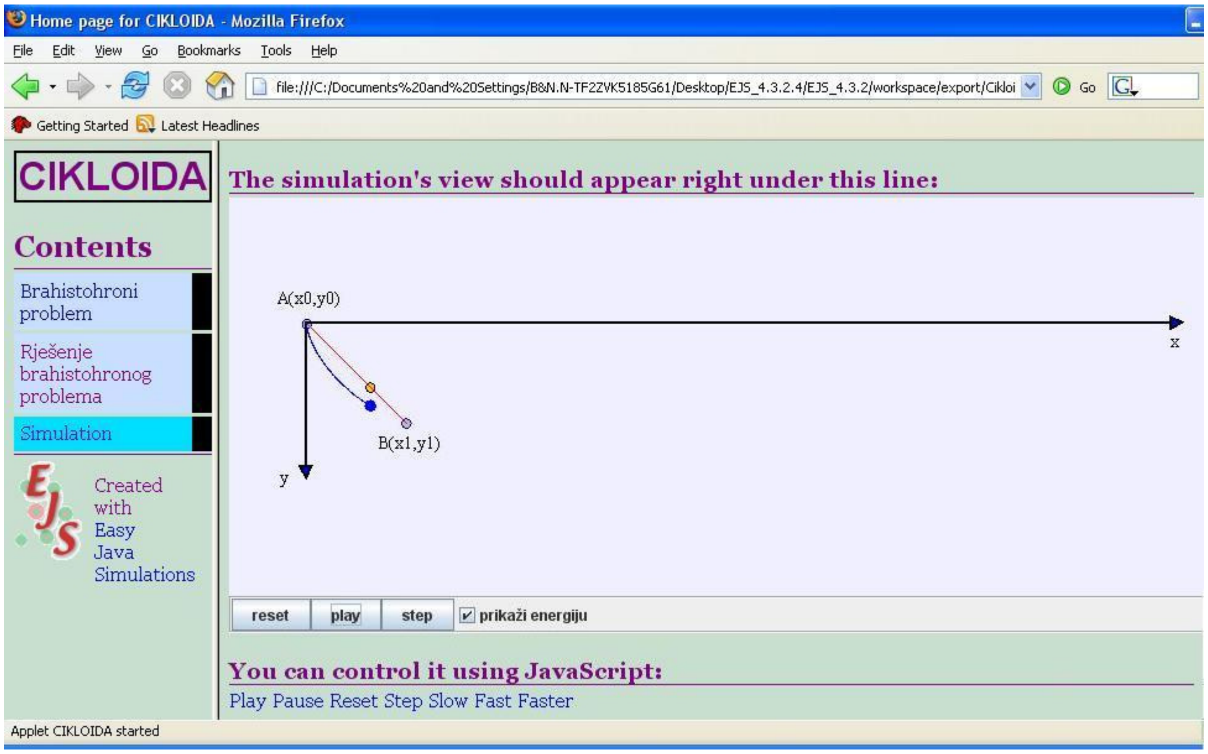
Defining the Description

Description part of cycloid simulation contains pages with basic theoretical information regarding the brachistochrone problem and its solution, Fig. 9.

B. *Running the virtual lab*

The cycloid simulation can be run as a stand-alone application and as an applet. For the distance learning implementation it is more convenient to run it as an applet given in Fig. 10.

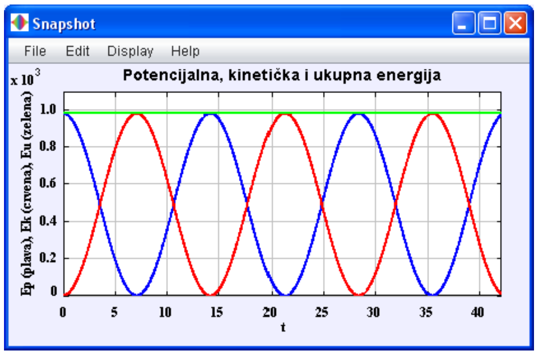
The contents of the generated HTML page can be seen on the left side of the page. The first two items represent theoretical introduction to the brachistochrone problem and its solution. Selecting them student gets familiar with the problem to be simulated. Clicking on simulation item, the applet is generated on the right side of the page. The control of the simulation execution and interaction are obtained by clicking on one of the control buttons (reset, play, and step). For showing the energy plots the check box has to be selected. In this way student can track how the energies of a blue particle change depending on its position. Fig. 11 shows potential (blue trace) and kinetic energy (red trace), and proves that total energy (green trace) is constant over the whole duration of blue particle movement as stated in Section II. This figure also shows the change of blue particle position along *x*-axis and *y*-axis.


Home page for CIKLOIDA - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/B&N.N-TF2ZVK5185G61/Desktop/EJS_4.3.2.4/EJS_4.3.2/workspace/export/Cikloi
Go
Getting Started
Latest Headlines
CIKLOIDA
Contents
Brahistohroni problem
Rješenje brahistohronog problema
Simulation
Created with Easy Java Simulations
A(x0,y0)
x
B(x1,y1)
y
reset
play
step
prikaži energiju
Play Pause Reset Step Slow Fast Faster
Applet CIKLOIDA started

The simulation's view should appear right under this line:

You can control it using JavaScript:

Fig. 10. Running cycloid simulation as an applet.



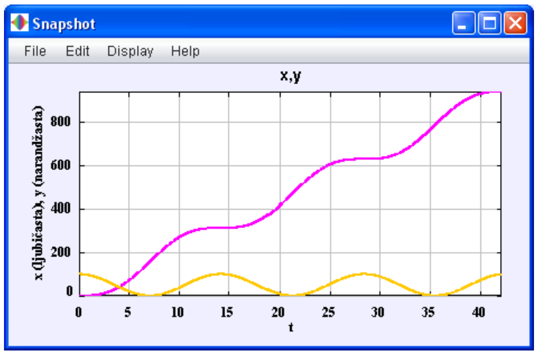


Fig. 11. Snapshots of energy plots and how *x* and *y* changes with time.

V. Conclusion

In this paper we presented a new approach to building the special interactive environment to be used in control systems education. It is based on the idea that the non-programming instructors create their own innovative pedagogical tools. To illustrate this approach, in order to improve the learning in optimal control theory, a concrete application in Easy Java Simulations related to the brachistochrone problem is developed. Future work will apply primarily to new uses of the considered concept.

References

[1] Easy Java Simulations' Home Page. Available: http://fem.um.es/Ejs

[2] The Open Source Physics Project, Available: http://www.opensourcephysics.org

[3] F. Esquembre, Easy Java Simulations: A Software Tool to Create Scientific Simulations in Java, Comput. Phys. Commun., vol. 156, no. 2, pp. 199–204, Jan. 2004.

[4] J. Sánchez, S. Dormido, F. Esquembre, R. Pastor, Interactive Learning of Control Concepts Using Easy Java Simulations, Available: http://www.imamu.edu.sa/dcontent/IT_Topics/java/ip_2_aad_34290345.pdf

[5] S. Dormido, H. Vargas, J. Sanches, N. Duro, R. Dormido, S. Dormido-Canto, F. Esquembre, Using Web-based laboratories for control engineering education, Intern. Conf. on Engin. Education – ICEE 2007, Coimbra, Portugal, September 2007, Available: http://www.dia.uned.es/ proyectos/dpi2004-01804/documents/congresos/Congreso41.pdf

[6] F.L. Lewis, V.L. Syrmos, *Optimal Control*, John Wiley&Sons Inc., New York, 1995, ch. 5.

[7] Sussmann, H.J., Willems, J. C., 300 Years of Optimal Control: From The Brachistochrone to the Maximum Principle, *IEEE Control Systems*, Vol. 17, No. 3, 1997, pp. 32-44.

[8] Sage, A. P., White Ch. C., *Optimum Systems Control*, Englewood Cliffs, NJ: Prentice-Hall, 1977, ch. 3.