Computer Simulation of Hybrid Systems by ISMA Instrumental Facilities

Yu.V. Shornikov, M.S. Myssak, D.N. Dostovalov

Abstract— A class of hybrid systems (HS) unresolved with respect to the derivative is considered. Architecture of instrumental environment is designed in accordance with CSSL standard. Library of original numerical solvers, embedded in simulation environment, is presented. Algorithm of numerical analysis of HS modes is given. Theorem about the choice of the integration step considering the HS event function dynamic has been formulated and proved. Algorithm of accurate HS event detection with implicit continuous behaviour models is designed. Testing of the proposed algorithms in the ISMA instrumental environment is performed. Example of specification and analysis of electric power systems models is given.

Keywords— computer aided analysis, software architecture, numerical simulation, differential equations, event detection, circuit simulation.

I. Introduction

Hybrid systems (HS) theory is a modern and versatile apparatus for mathematical description of the complex dynamic processes in systems with different physical nature. Such systems are characterized by the composition of the continuous and discrete behaviours. Earlier the ISMA instrumental environment [1, 2] examined models and methods of HS analysis, continuous modes of which are described by the Cauchy problem with constraints. In this paper the extension of class of systems by models unresolved with respect to the derivative is proposed. Numerical analysis of the new class of problems requires using a specific integration and HS event detection algorithms. The described algorithms are implemented in the ISMA and successfully tested.

II. Class of Systems

There are many systems (mechanical, electrical, chemical, biological, etc.), the behavior of which can be conveniently described as a sequential change of continuous modes. These systems are referred to as hybrid or event-continuous. Each mode is given by a set of differential-algebraic equations with the following constraints:

$$
\begin{aligned}
&y' = f\left(x, y, t\right), x = \varphi\left(x, y, t\right),\\
&pr : g\left(x, y, t\right) < 0,\\
&t \in \left[t_0, t_k\right], x\left(t_0\right) = x_0, y\left(t_0\right) = y_0,\\
&x \in R^{N_x}, y \in R^{N_y}, t \in R,\\
&f : R^{N_x} \times R^{N_y} \times R \to R^{N_y},\\
&\varphi : R^{N_x} \times R^{N_y} \times R \to R^{Nx},\\
&g : R^{N_x} \times R^{N_y} \times R \to R^{S}.
\end{aligned}
\qquad (1)
$$

The vector-function $g(x, y, t)$ is referred to as event function or guard. A predicate pr determines the conditions of existence in the corresponding mode or state. Inequality $g(x, y, t) < 0$ means that the phase trajectory in the current mode should not cross the border $g(x, y, t) = 0$. Events occurring in violation of this condition and leading to transition into another mode without crossing the border are referred to as one-sided. Many practical problems are characterized by stiff modes, and the surface of boundary $g(x, y, t) = 0$ has sharp angles or solution has several roots at the boundary [2]. Numerical analysis of such models by traditional methods is difficult or impossible, as it gives incorrect results. Therefore it is necessary to use special methods to detect events accurately.

Computer analysis of these systems is typically performed in simulation tools, best of which are Charon (USA), AnyLogic (Russia), Scicos (France), MVS (Russia), Hybrid Toolbox and HyVisual (USA), DYMOLA (Sweden) and etc.

In the simulation of electrical circuits, processes of chemical kinetics, electromechanical processes and many other applications a necessity arises to numerically analyze HS, modes of which are given by stiff implicit systems of high-dimensional differential equations with strict one-sided constraint:

$$F\left(x, x', t\right) = 0, pr : g\left(x, t\right) < 0, t \in \left[t_0, t_k\right], x\left(t_0\right) = x_0, \quad (2)$$

where $x \in R^N$ is the vector of state variables, $t \in R$ is the argument, $F : R^N \times R^N \times R \to R^N$ is a continuous vector-function for given mode of HS, $g : R^N \times R \to R$ is the event-scalar function or the guard, x_0 is the value at the initial point t_0.

The problem (2) is usually stiff that leads to the application of implicit numerical formulas required Jacobi matrix inversion. Due to the ease of implementation and good

This work was supported by grant 14-01-00047-a from the Russian Foundation for Basic Research, RAS Presidium project № 15.4 “Mathematical modeling, analysis and optimization of hybrid systems”.

Yu.V. Shornikov is with the Design Technological Institute of Digital Techniques Siberian Branch of Russian Academy of Science, Novosibirsk, Russia (e-mail: shornikov@inbox.ru).

M.S. Myssak, D.N. Dostovalov is with the Department of Automated Control Systems, Novosibirsk State Technical University, Novosibirsk, Russia (e-mails: maria.myssak@gmail.com, dostovalov.dmitr@gmail.com).

accuracy and stability properties Rosenbrock type methods [3, 4] are widely used in solving stiff problems.

III. ARCHITECTURE OF INSTRUMENTAL ENVIRONMENT

Development of simulation languages, simulators, simulation systems, etc. is essentially influenced by the CSSL (continuous system simulation language) Standard 1968 [4]. Although forty years old, the structures defined in CSSL Standard are used up to now. End of 90ties, CSSL extended to implicit systems, while a new modelling language, Modelica, was introduced. In principle, the modelling paradigm changed from signal flow − oriented modelling (explicit systems) to power − oriented modelling (implicit systems), from “causal” signal modelling to “acausal” physical modelling. The early CSSS standard determined basic necessary features for a simulator, the late developments to implicit systems fixed extended features for simulation systems − both referred as classical CSSL features. In 1968, the CSSL standard set first challenges for features of simulation systems, defining necessary basic features for simulators and a certain structure for simulators.

The CSSL standard also defines segments for discrete actions, first mainly used for modelling discrete control. So-called DISCRETE regions or sections manage the communication between discrete and continuous world and compute the discrete model parts. For incorporating discrete actions, the simulation engine must interrupt the ODE solver and handle the event. For generality, efficient implementations set up and handle event lists, representing the time instants of discrete actions and the calculations associated with the action, where in-between consecutive discrete actions the ODE solver is to be called. In order to incorporate DAEs and discrete elements, the simulator’s translator must now extract from the model description the dynamic differential equations (derivative), the dynamic algebraic equations (algebraic), and the events (event i) with static algebraic equations and event time, as given in Fig. 1 [5] (extended structure of a simulation language due to CSSL standard). In principle, initial equations, parameter equations and terminal equations (initial, terminal) are special cases of events at time t = 0 and terminal time. Some simulators make use of a modified structure, which puts all discrete actions into one event module, where CASE - constructs distinguish between the different events.

Simulation environment of complex dynamical and hybrid systems called ISMA (translated from Russian “Instrumental Facilities of Machine Analysis”) is developed at the department of Automated control systems of Novosibirsk state technical university (Russia) [6].

Specification of hybrid systems is carried out using graphical and symbolic languages that are the system content of instrumental environment. Analytical content is provided by numerical methods and algorithms for computer analysis corresponding to the chosen class of systems and methods for solving these models. ISMA environment is developed subject to simplicity of description of dynamical and hybrid

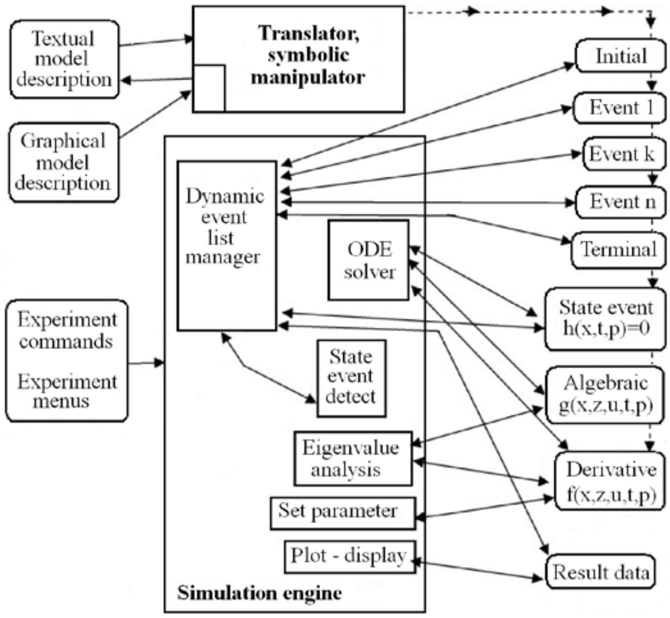

Fig. 1. Extended structure of a simulation system due to extensions of the CSSL standard with discrete elements and with DAE modelling

models in the language that is maximally close to the object language. Main features of ISMA are the following:

- Composition of hybrid models is carried out in visual structural-textual form;
- Structural form of model description corresponds to the classical description of systems by block diagrams and includes all necessary components such as integrators, accumulators, amplifiers, signal sources, nonlinear elements and others;
- Language of symbolic specification is approached maximal to the language of mathematical formulas;
- Special module for specification of problems of chemical kinetics in the language of chemical reactions which automatically translates them into a system of differential equations;
- A variety of traditional and original numerical methods included methods that are intended for the analysis of ODE systems of medium and high stiffness;
- Computer simulation in real time;
- Graphic interpreter called GRIN provides a wide range of tools for analysis and visualization of simulation results such as scaling, tracing, optimization, displaying in the logarithmic scale and phase plane;
- Extension of system functionality by adding new typical components and numerical methods.

Architecture of ISMA software package (Fig. 2) is designed [7] in accordance with CSSL to unify existing mathematical program software for analysis of problems in various object domains: chemical kinetics, automation, electricity, etc.

ISBN: 978-1-61804-251-4

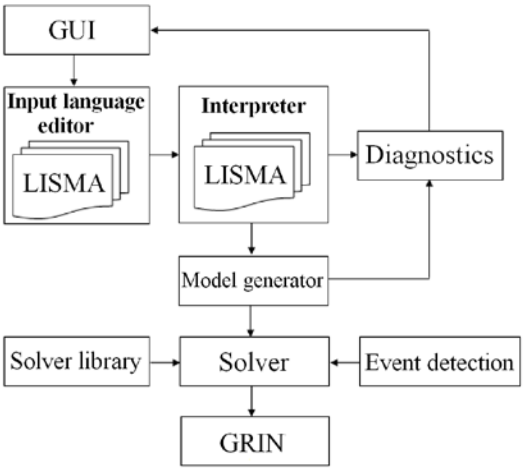

Fig. 2. ISMA architecture

IV. LIBRARY OF NUMERICAL METHODS

Peculiarities of numerical analysis are defined by the configuration and implementation of the solver in the scheme interpreter. Solver is configured to numerical analysis not only of smooth dynamical systems but also systems with ordinary discontinuity and stiff systems [2, 8]. For the analysis of the stiff modes new original m- phasic methods of p - order (Table I), developed by the authors, are included in the solver library.

TABLE I. LIBRARY OF NUMERICAL METHODS

Method (p, m)	**Description**
DISPF (5, 6)	Stability control, systems of medium and low stiffness
RADAU5 (3, 3)	Stiff systems
DISPF1_RADAU	Adaptive method DISPF in combination with RADAU5 with stiffness control, essentially stiff systems
DP78ST (8, 13)	Stability control, variable order and step, systems of medium stiffness and high precision
RKF78ST (7, 13)	Stability control, variable order and step, systems of medium stiffness and high precision
RK2ST (2, 2), RK3ST (2, 3)	Explicit methods with stability control for analysis of non-stiff systems
DISPS1	Algorithm of variable order with adaptive stability region
MK22 (2, 2), MK21 (2, 2)	Freezing of Jacobean matrix, stiff systems
MK11F	Algorithm of analysis of implicit problems

Libraries of standard blocks and numerical methods are implemented as independent application modules that are loaded at run time.

This approach allows to allocate in the application programming interface (API) a set of functions and classes required for the implementation of element libraries and numerical methods. Using the API any user with basic knowledge of object-oriented programming able to develop and built in the system new typical elements and numerical methods without recompiling the entire system.

V. EVENT DETECTION IN HYBRID SYSTEMS

The correct analysis of hybrid models is significantly depends on the accuracy of detection of the change of the local states of the HS. Therefore, the numerical analysis is necessary to control not only the accuracy and stability of the calculation, but also the dynamics of the event-function. The degree of approximation by the time the event occurred is defined by the behavior of event driven function.

Analyze the behavior of the event function $g(x,t)$. Let the method of the form $x_{n+1} = x_n + h_n\varphi_n$, where function φ_n is calculated in point t_n, is used for calculations.

Then the event-function $g(x,t)$ at point t_{n+1} has a form $g_{n+1} = g(x_n + h_n\varphi_n, t_n + h_n)$. Decomposing the g_{n+1} in a Taylor series and taking into account the linearity of g_{n+1}, we obtain the dependence of g_{n+1} of the projected step h_n:

$$g_{n+1} = g_n + h_n\left(\frac{\partial g_n}{\partial x}\cdot\varphi_n + \frac{\partial g_n}{\partial t}\right). \qquad (3)$$

Theorem. The choice of the step according to the formula

$$h_n = (\gamma - 1)\, g_n \Big/ \left(\frac{\partial g_n}{\partial x}\cdot\varphi_n + \frac{\partial g_n}{\partial t}\right), \gamma \in (0,1), \qquad (4)$$

provides the event-dynamics behavior as a stable linear system, the solution of which is asymptotically approaching to the surface $g(x,t) = 0$.

Proof. Substituting (4) in (3), we have $g_{n+1} = \gamma g_n$, $n = 0,1,2,...$ Converting recurrently this expression we get $g_{n+1} = \gamma^{n+1} g_0$. Given that $\gamma < 1$, then $g_n \to \infty$ takes place when $n \to \infty$. In addition, condition $\gamma > 0$ implies that function g_n does not change sign. Therefore, when $g_0 < 0$, $g_n < 0$ will be valid for all n. Then the guard condition will never cross the potentially dangerous area $g(x_n, t_n) = 0$, which completes the proof.

A. *Control of event function in the integration algorithm*

We complete the implicit problem's integration algorithm by the algorithm of the step control that takes into account the event function dynamics.

Let the solution x_n and $y_n = x'_n$ at the point t_n is calculated with the step h_n. In addition, the new accuracy step

h_{n+1}^{ac} is computed by the formula (4). Then the approximate solution at the point t_{n+1} is calculated as follows

Step 1. Calculate the functions

$$g_n = g\left(x_n, t_n\right), \frac{\partial g_n}{\partial x} = \frac{\partial g\left(x_n, t_n\right)}{\partial x}, \frac{\partial g_n}{\partial t} = \frac{\partial g\left(x_n, t_n\right)}{\partial t}.$$

Step 2. Calculate $g_n' = \frac{\partial g_n}{\partial x}\varphi_n + \frac{\partial g_n}{\partial t}$, where $\varphi_n = y_n$.

Step 3. If $g_n' < 0$, then $h_{n+1} = h_{n+1}^{ac}$ and go to the Step 6.

Step 4. Calculate the new "Event" step h_{n+1}^{ev} by the formula

$$h_{n+1}^{ev} = (\gamma - 1)\frac{g_n}{g_n'}.$$

Step 5. Calculate the new step h_{n+1} by the formula

Step 6. Go to the next integration step.

In the Step 3, unlike the previously presented algorithm [9], we determine the direction of event-function change. Near the boundary regime denominator (4) will be positive, and away from the boundary $g(x,t)=0$ it becomes negative. Then, defining the direction of event-function change, we do not impose any further restrictions on the integration step if the event-function is removed from the state boundary.

B. *Tests*

To illustrate the capacity for work of the proposed algorithms consider a simple hybrid system – a bouncing ball. Modal behavior can be given by an implicit system of differential equations

$$y' - v = 0, v' + a = 0, \tag{5}$$

where y is the height from the surface of the ball rebound, v is the ball velocity, a is the free fall acceleration. An event occurs at the moment when the ball touches the rebound surface $y = 0$, therefore the event function takes the form $g = -y$, and the predicate $pr: -y < 0$. At the moment of rebound the ball changes the moving direction. Let the rebound is inelastic, then when the event occurs the velocity changes according to the rule $v = -\alpha \cdot v$, where α is the retention rate of speed.

Moments of the ball rebound from the surface and values of the variable h when the event occurs are shown in Figure 3. A significant error $\varepsilon_1 \approx 0.75$ in detection of event changes is made when calculating event function without dynamics control (Fig. 3a). This leads to violation of condition of one-sidedness of the events and as a result to erroneous global solution. Using of algorithm of asymptotic approximation to the regime border (Fig. 3b) provides about an order more accurate detection of the moment when regime of HS has changed $\varepsilon_2 \approx 0.06$.

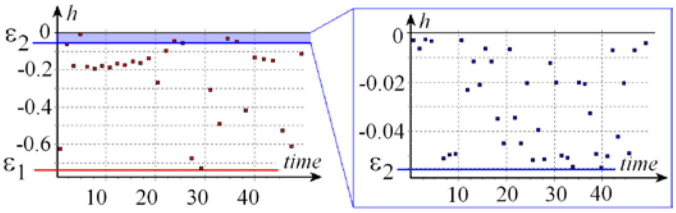

Fig. 3. Moments of event detection: a) without dynamics control; b) with asymptotic approximation to regime border

VI. SIMULATION OF FAULT IN AN ELECTRICAL NETWORK

As an illustration of a new class of systems and as a test case for the formulated algorithms analyze the model of three-phase fault in an electrical network. Schematic diagram of the electrical power system (EPS) built in graphics editor of the ISMA instrumental environment is shown in Fig. 4.

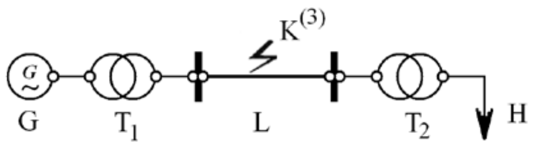

Fig. 4. Schematic diagram of the electrical network

Considered scheme consists of generator G, transformers T_1, T_2, line L and load H. In the equivalent circuit in Fig. 5 capacitive conductivity of the line and transformer non-load loses are not taken into account and the load is taken into account by approximately active and inductive reactance.

Transient is initiated by the contact closure K. In this case previously established mode of power system is changed to the new mode corresponding fault and another system configuration. Thus, the model is a two-mode hybrid system (HS) [2].

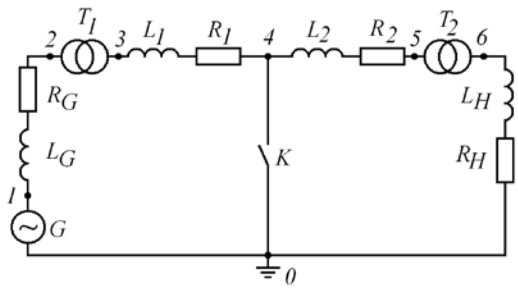

Fig. 5. Schematic diagram of the electrical network

The discrete behavior of the hybrid system is illustrated by the state chart shown in Fig. 6. State init corresponds to the functioning of EPS before the fault. Switching to state short corresponded to the fault condition occurs when a logical predicate pr is carried out.

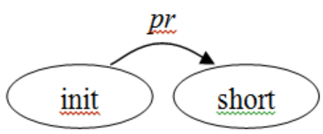

Fig. 6. Behavior map

In the graphics editor of schematic diagrams of EPS hybrid behavior is specified in the configuration editor window for the equivalent circuit of a transmission line L as shown in Fig. 7.

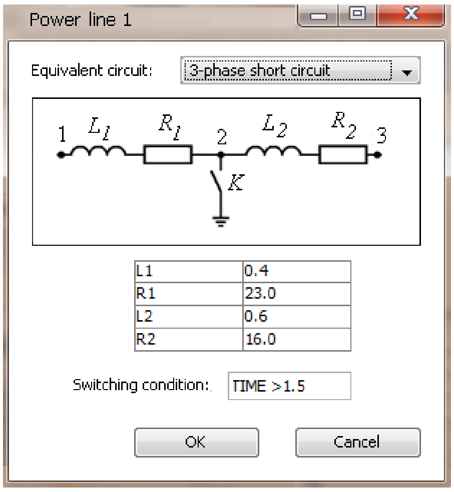

Fig. 7. Configuring the parameters of the equivalent circuit

The mathematical model is composed by Park-Gorev equations in rotating coordinate system (d,q) associated with the generator rotor G. Let the axis q is ahead of the axis d. Obtain a system of equations for the generator G

$$u_{1d}\cos(\theta-t)+u_{1q}\sin(\theta-t)+ri_{Gd}+$$

$$+L_d\frac{di_{Gd}}{dt}-L_{ad}\frac{di_f}{dt}-L_{ad}\frac{di_g}{dt}-\left(L_q i_{Gq}-L_{aq}i_h\right)\omega=0,$$

$$-u_{1d}\sin(\theta-t)+u_{1q}\cos(\theta-t)+ri_{Gq}+$$

$$+L_q\frac{di_{Gq}}{dt}-L_{aq}\frac{di_h}{dt}+\left[L_d i_{Gd}-L_{ad}\left(i_f+i_g\right)\right]\omega=0,$$

$$-u_f+r_f i_f+L_f\frac{di_f}{dt}+L_{ad}\frac{di_g}{dt}-L_{ad}\frac{di_{Gd}}{dt}=0,$$

$$r_g i_g+L_g\frac{di_g}{dt}+L_{ad}\frac{di_f}{dt}-L_{ad}\frac{di_{Gd}}{dt}=0,$$

$$r_h i_h+L_h\frac{di_h}{dt}-L_{aq}\frac{di_{Gq}}{dt}=0,$$

$$\frac{d\omega}{dt}=\frac{T_{\partial}+\left[\left(L_d-L_q\right)i_{Gq}+L_{aq}i_h\right]i_{Gd}-L_{ad}i_{Gq}\left(i_f+i_g\right)}{T_J},$$

$$\frac{d\theta}{dt}=\omega.$$

Here the index f refers to the excitation winding and indices g and h refers to the longitudinal and transverse damper contours respectively.

Equations for the area of the equivalent circuit 1-2:

$$u_{1d}-u_{2d}=R_G i_{12d}+L_G\left(\frac{di_{12d}}{dt}-i_{12q}\right),$$

$$u_{1q}-u_{2q}=R_G i_{12q}+L_G\left(\frac{di_{12q}}{dt}-i_{12d}\right).$$

For the areas 3-4, 4-5 and 6-0 equations will have a similar form.

Equations for the transformer T_1:

$$u_{3d}=K_{T1}\left(u_{1q}+\sqrt{3}u_{1d}\right),\ u_{3q}=K_{T1}\left(\sqrt{3}u_{1q}-u_{1d}\right),$$

$$i_{34d}=K_{T1}\left(i_{12q}+\sqrt{3}i_{12d}\right),\ i_{34q}=K_{T1}\left(\sqrt{3}i_{12q}-i_{12d}\right).$$

Here K_{T1} is a transformation ratio. Equations for the transformer T_2 are treated similarly.

Equations of the first Kirchhoff's law for point 1:

$$i_{Gd}\cos(\theta-t)-i_{Gq}\sin(\theta-t)=i_{12d},$$

$$i_{Gd}\sin(\theta-t)+i_{Gq}\cos(\theta-t)=i_{12q}.$$

When an event corresponded to the fault occurs in HS, the voltage in point 4 is equated to zero $u_{4d}=u_{4q}=0$. In this case in the equivalent circuit two independent contours are formed. The equations for sections of the contours remain the same.

Plots of some state variables obtained in ISMA are shown in Figure 8. Calculation results correspond to theoretical statements and coincide with results obtained in MATLAB.

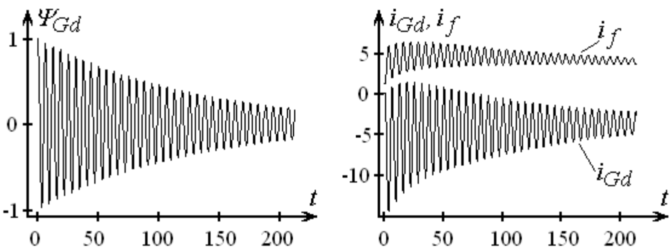

Fig. 8. Simulation results in ISMA

VII. CONCLUSIONS

In this paper the new class of hybrid systems within the ISMA instrumental environment, the modal behavior of which is defined by a system of ODE unresolved with respect to the derivative, is introduced. Architecture of instrumental environment is designed in accordance with CSSL standard. The new original method of switching point's localization is proposed. The algorithm easily complements the existing numerical solvers based on explicit and semi-explicit schemes including the proposed algorithm of implicit problem's analysis. Model of new HS system class is presented and studied in ISMA.

ISBN: 978-1-61804-251-4

REFERENCES

[1] Yu.V. Shornikov, “Numerical modeling of dynamic processes in electric power systems as a strategic management tool,” Yu.V. Shornikov, I.N. Tomilov, D.N. Dostovalov, M.S. Denisov, Scientific bulletin of the NSTU, vol. 4, no. 45, pp.129-134., 2011.

[2] E.A. Novikov, Yu.V. Shornikov, Computer simulation of stiff hybrid systems: monograph, Novosibirsk, Russia: Publishing house of NSTU, 2012.

[3] H.H. Rosenbrock, “Some general implicit processes for the numerical solution of differential equations,” Computer, vol. 5, pp. 329–330., 1963.

[4] Yu.V Shornikov, D.N. Dostovalov, M.S. Myssak “Simulation of hybrid systems with implicitly specified modal behavior in the ISMA environment “, Humanities and scinence university journal, no. 5, pp. 175-182, 2013.

[5] F. Breitenecker, N.Popper, “Classification and evaluation of features in advanced simulators,” Proceedings MATHMOD 09 Vienna, Full papers CD Volume, 2009.

[6] Yu.V. Shornikov, “Instrumental tools of computerized analysis (ISMA),” Yu.V. Shornikov, V.S. Druzhinin, N.A. Makarov, K.V. Omelchenko and I.N. Tomilov, Official registration license for computers 2005610126, Moscow, Rospatent, 2005.

[7] Yu.V. Shornikov, M.S. Myssak, D.N. Dostovalov et al, “Using ISMA Simulation Environment for Numerical Solution of Hybrid Systems with PDE”, Proc. Computer Modeling and Simulation, Sankt-Petersburg, Rusia, pp. 101-108, 2014.

[8] D.N. Dostovalov “Computer simulation and algortithms of numerical analysis of hybrid systems”, Control system and information technologies, vol. 53, no 3.1 pp. 128-133, 2013.

[9] Yu.V. Shornikov, D.N. Dostovalov, M.S. Myssak et al, “Specification and analysis of discrete behavior of hybrid systems in the workbench ISMA”, Open Journal of Applied Sciences, vol. 3, no. 2b, pp. 51-55, 2013.