THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 19, Number 4/2018, pp. 589–596

ON THE CONTROL OF A COOPERATIVELY ROBOTIC SYSTEM BY USING HIBRID LOGIC ALGORITHMS

Veturia CHIROIU, Ligia MUNTEANU, Cristian RUGINĂ

Institute of Solid Mechanics of the Romanian Academy - Ctin Mille 15, 010141 Bucharest, Romania
Corresponding author: Veturia CHIROIU, E-mail: veturiachiroiu@yahoo.com

Abstract. A hybrid logic algorithm is presented for extending the control of a cooperatively robotic system by including uncertain input conditions and the hard and soft stops. The method can be used in surgical interventions where the robot and the physician are working with the same surgical tool. The results show that the algorithm improves the trajectory tracking performance and preventing the surgeon-tool r to cross the critical boundaries and to move towards undesirable directions.

Key words: surgical robotics; differential logic control; motion trajectories.

1. INTRODUCTION

Hybrid systems are dynamic systems with discrete and continuous repetitive evolutions described by differential equations with discrete and continuous transient interactions [1, 2]. The discrete transitions are instantly changes in the system, while the continuous transitions are subjected to restrictions raised from the superposition between continuous dynamics and discrete control. The hybrid system integrates three components: the continuous behavior of the surgical instrument, the discrete robotic interventions and the control of the surgical tool-tip motion.

The control of a hybrid system is provided by *the dynamic differential logic* algorithm (dL) [3–10]. This control is concentrated on the avoiding collisions with the critical boundaries Γ which surround a safe work area Ω defined *ab initio*.

The paper discusses the control the cooperation between the surgeon and the robot during surgical procedures of the liver tumours. The hybrid algorithm is designed to prevent the tool-tip (SI) to reach and cross one or more critical boundaries of Ω. Usually, the surgeon manipulates freely the surgical tool in Ω without robotic interference, but, when SI, originally located at the distance d to Γ, reaches a critical area of all points located to a distance $D < d$ from the critical border, the robot starts to attenuate the speed of the SI proportionally to D. The $\bar{D} < D < d$ denotes the distance from which the SI begins to stop softly, to avoid the shocks (Fig. 1).

The control algorithm corrects the movement of SI through hybrid discrete and continuous logics for different inputs. A system with analog equipment is described by continuous mathematics, while a system with software devices that processes the data, is described by discrete logic programs. A hybrid program is an interface between these two systems because it incorporates both the discreet program and the continuous behavior of analog equipment. The control explores the domain geometry with eliminating the unexpected situations [12, 13].

We must mention the surgical robot built at the Johns Hopkins University Center for Integrated Surgical Systems and Technology Group [14,15]. This robot is composed of three components: Stealth Station navigation unit that follow the position and orientation of the optical markers on the rigid body, the 3DSlicer unit for viewing and analyzing imaging data, and a 6-degree Neuromate robotic arm equipped with a Food & Drug Administration (FDA). The robotic system has active constraints defined in three domains: safe, forbidden and critical boundary. The robot locks when the surgeon enters in the forbidden domain, preventing deeper intrusions. The flowchart of the algorithm is given in Fig. 2 [15, 16].

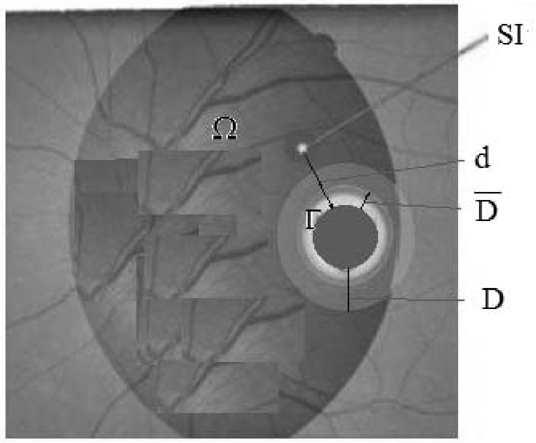

Fig. 1 – Cooperatively control to restrict a tool-tip (SI) to cross Γ .

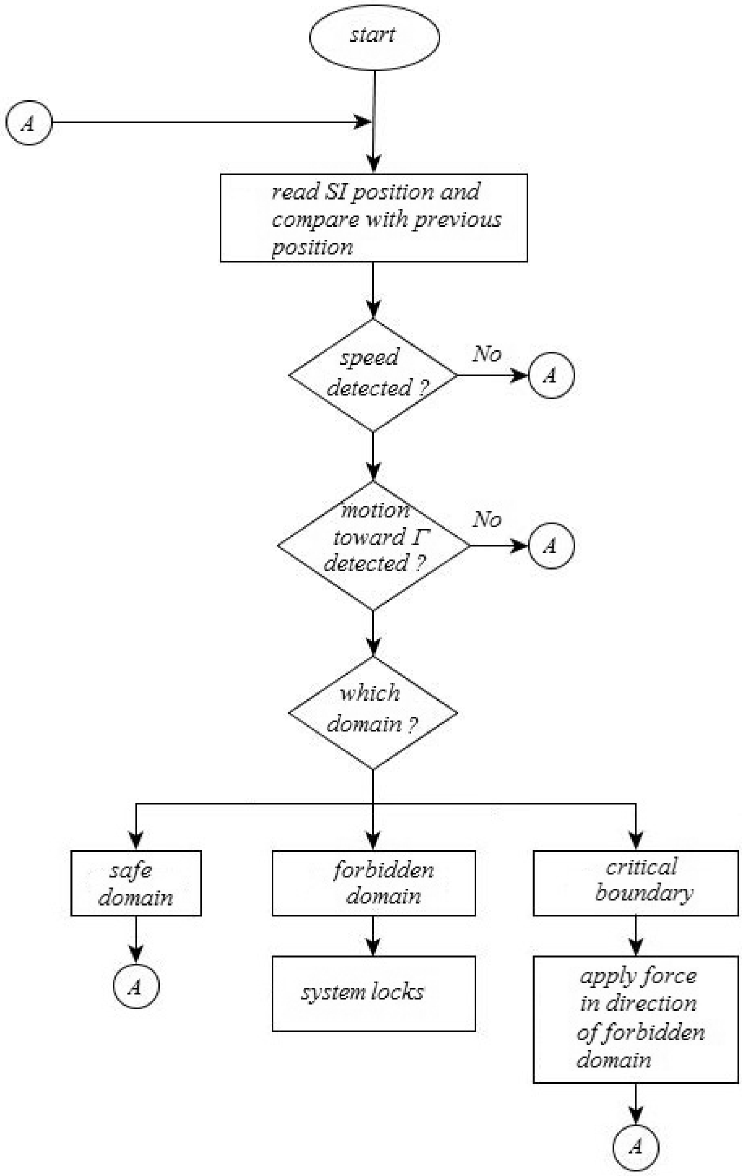

Fig. 2 – A general flowchart for the control algorithm.

2. KINEMATICS OF THE ROBOT

A system of coordinates (X, Y, Z) is attached to the base of the robotic system with a simulated SI as the top of a virtual joystick (red), Z is the vertical axis (Fig. 3). The working space Ω is defined by coordinates (x, y, z). The SI is located at the height h from the base of robot. The joint vector $q = (q_1, q_2, q_3)$ is defined as: q_1 is the rotation about *X*-axis (pitch angle), q_2 is the rotation about *Y*-axis (roll angle) and q_3 the rotation about the axis Z which is common with the joystick axis (yaw angle). The q_3 does not influence the SI position as required by the laser joystick.

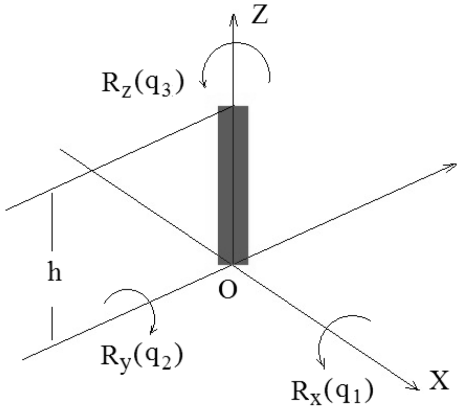

Fig. 3 – Scheme of the system of coordinates attached to the robotic system.

The mechanism has the following transformation matrix [16]

$$T_S^O = \begin{pmatrix} i_x / \sqrt{1-\cos^2(q_1)\cos^2(q_2)} & i_y & i_z / \sqrt{1-\cos^2(q_1)\cos^2(q_2)} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

where

$$i_x = \begin{pmatrix} \sin(q_2) \\ \cos(q_1)\sin(q_1)\cos(q_2) \\ \sin^2(q_1)\cos(q_2) \end{pmatrix}, \; i_y = \begin{pmatrix} 0 \\ \sin(q_1) \\ -\cos(q_1) \end{pmatrix}, \; i_x = \begin{pmatrix} -\sin(q_1)\cos(q_2) \\ \cos(q_1)\sin(q_2) \\ \sin(q_1)\sin(q_2) \end{pmatrix}. \tag{2}$$

The transformation matrix of roll angle is given by

$$T_J^S = \begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & 0 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

The position of SI is

$$T_J^O = T_S^O T_J^S. \tag{4}$$

The Jacobean matrix which transform the joint vector q into working space Ω is

$$J_L = \begin{pmatrix} -h\cos(q_1)\cos(q_2)\sin^2(q_2)/\alpha & h\sin(q_1)\sin(q_2)/\alpha & 0 \\ -h\sin(q_1)\sin(q_2)/\alpha & h\cos(q_1)\cos(q_2)\sin^2(q_1)/\alpha & 0 \\ h\cos(q_1)\sin^3(q_2)/\alpha & h\sin^3(q_1)\cos(q_2)/\alpha & 0 \end{pmatrix}, \tag{5}$$

with $\alpha = \sqrt{(1-\cos^2(q_1)\cos^2(q_2))^3}$. The inverse kinematics of the robot is given by

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \arctan 2(i_z, i_y) \\ \arctan 2(i_z, i_z) \\ \arctan 2\{-\sin(q_2)\sin(q_3)/\alpha, \sin(q_2)\cos(q_3)/\alpha)\} \end{pmatrix}. \tag{6}$$

In practice, the source of uncertainly of the robot may be the input, the variable state, imperfect measurements, parameter and output data. The uncertain is bounded in all cases and the control in this case is assessed by the discrete model with the help of solutions of the continuous model, in order to describe how error occurs and propagate through the model.

The hybrid language permits description of the interaction between a discrete evolution and the continuous action of SI. To represent these interactions, dL includes the non-determinism concept and the ability to describe the continued evolution of SI. Non-determinism is an uncertainty that is not described probabilistically.

3. DESCRIPTION OF dL

The language dL is composed by arithmetic operations, state variables (as example the force f which is applied to SI by the surgeon), a set of statements and assignments. The operations in dL are: logical and $a \wedge b$; logical or $a \vee b$; negation $\neg a$; existential and universal quantifications in R, $\exists x P(x)$ and $\forall x P(y)$, respectively; all running of a satisfying the condition ψ (box mode) $[a]\psi$ satisfying ψ $<a>\psi$. The language of modelling contains the statement in the continuous evolution is written as $\dot{x}_1 = \varphi_1, \dot{x}_2 = \varphi_2, \ldots \dot{x}_n = \varphi_n \,\&\, \psi$; an assumption is expressed as $?\psi$; the assignment is $x_i := \varphi_i$; the non-deterministic assignment of any value $x_i := *$; the sequentially running a and b, $a;b$; non-deterministic choise $a \cup b$; non-deterministic loop $a*$. Details on dL can be found in [3-10]. The state variables can be discrete and continuous.

A sequence of dL program that contains an arbitrary input of a non-deterministic value to f, followed by three non-deterministic choices [23]:

$$\begin{aligned} &\mathrm{ctrl} = (f := *; \\ &(\dot{r} = gf \,\&\, f >= 0) \cup \\ &(\dot{r} = gf \,\&\, (f <= 0) \wedge (r >= D)) \cup \\ &(\dot{r} = g(r/D) f \,\&\, (f <= 0) \wedge (r <= D)))* \end{aligned} \tag{7}$$

The goal of the algorithm requires that SI starting from a location $(r >= 0)$ continues to stay in a safe location in Ω at every moment of time for any input conditions. The safety property is described by $(r >= 0) \rightarrow [\mathrm{ctrl}(r >= 0)$. KeYmaera is an instrument that can check the safety property of the algorithm [10].

The constraints are modelled in linear or nonlinear inequalities over Boolean-valued variables

$$\begin{aligned} &formula ::= \{clause \wedge\} * clause \\ &clause ::= linear\_constaints \mid boolean\_var \rightarrow linear\_constraint \cup \\ &clause ::= nonlinear\_constaints \mid boolean\_var \rightarrow nonlinear\_constraint \end{aligned} \tag{8}$$

The algorithm permits the introduction of a large number of constraints by simple syntactic statements, and the surgeon decides to give up or not to a few of these constraints, or to solve any possible conflict between them. The performance of dL is measured by verification of all safety conditions of SI trajectories in Ω and the degree of performing the control task. The algorithm pays attention to causal relationships between variables by a full compatibility between continuous and discrete actions. The continuous evolution subjected to a constaint

4. CONTROL OF CROSSING A CRITICAL BOUNDARY BY THE TIP-TOOL IN A RANDOM MOTION

The robot and the surgeon manipulate the same surgical tool SI, so that when the surgeon applies a force f to SI, the robot allows it according to [6–8]

$$\frac{\mathrm{d}r}{\mathrm{d}t} = G(f), \tag{9}$$

where r is a positive value that describes the SI position in Ω, and G describes the discreet part of the system and it is a constant multiple of f. Equation (9) involves a negative feedback control flow with admittance control, that transforms forces and moments into velocities.

The analyze of the proposed *ab initio* trajectories of SI and the crossing of a critical boundary Γ can be done by applying the Greenwood and Novikov results [17–19].

Let us consider the 1D case. The safe domain Ω is bounded by a constant boundary $-g$, $g \geq 0$. The motion is described by

$$s_n = s_0 + \sum_{p=1}^{n} x_p, \tag{10}$$

where $x_p, p \geq 1$ are state variables. The time when SI reaches the critical border is denoted by t_g

$$t_g := \min\{n \geq 1 : s_n < -g\}. \tag{11}$$

The contact can be identified by checking the minimum distance between SI and critical border

$$\min\left(\frac{1}{2}(r_1 - r_2)^T (r_1 - r_2)\right), \tag{12}$$

where r_1 and r_2 are the position of SI and the border, respectively. The class $M \in R^2$ of given motions of SI can be defined as

$$M := \{0 < \mu < 1; |\nu| < 1\} \cup \{0 < \mu < 2; |\nu| \leq 1\} \cup \{\mu = 1, \nu = 0\} \cup \{\mu = 2, \nu = 0\}, \tag{13}$$

and can be be generated by a genetic algorithm [19] or by the modified Kronecker sequences implemented into quasi-Monte Carlo [22].

The overall speed $\dot{r}_1$ is given by the control law [23]

$$\dot{r}_1 = \dot{r} - \left(1 - \frac{d}{D}\right)(\dot{r} \cdot n_1) n_1, \tag{14}$$

where d is the distance from SI to Γ with the normal n_1.

The control of trajectories of SI is made by accounting every position of SI with respect to Γ and surrounding space Ω at every moment of time. The unsafe region is continuously monitored in a coordinate system fixed to SI and centred at the initial position of SI.

Most of the time, the surgeon manages freely the SI (green circle) without any robotic intervention (blue lines). Various situations with the surgeon's given trajectories in space are presented in Fig.4. The modified Kronecker sequence is applied by generalizing of the golden ratio by a metallic ratio $\varphi = \dfrac{a + \sqrt{a^2 + 4}}{2}$ with a a positive integer. The critical border Γ is the ends of Ω marked with red colour. When SI approaches Γ, the normal speed component to the border is attenuated by the robot and slowly cancelled. It is the case of the regions noted by A and B. It is possible that some SI trajectories not to be defined *ab initio,* and to be instantly changed depending on local working conditions. Or, the inputs can be unclear and the measurement imperfect. Such situations are handled by intervention of the robot and the surgeon accepting.

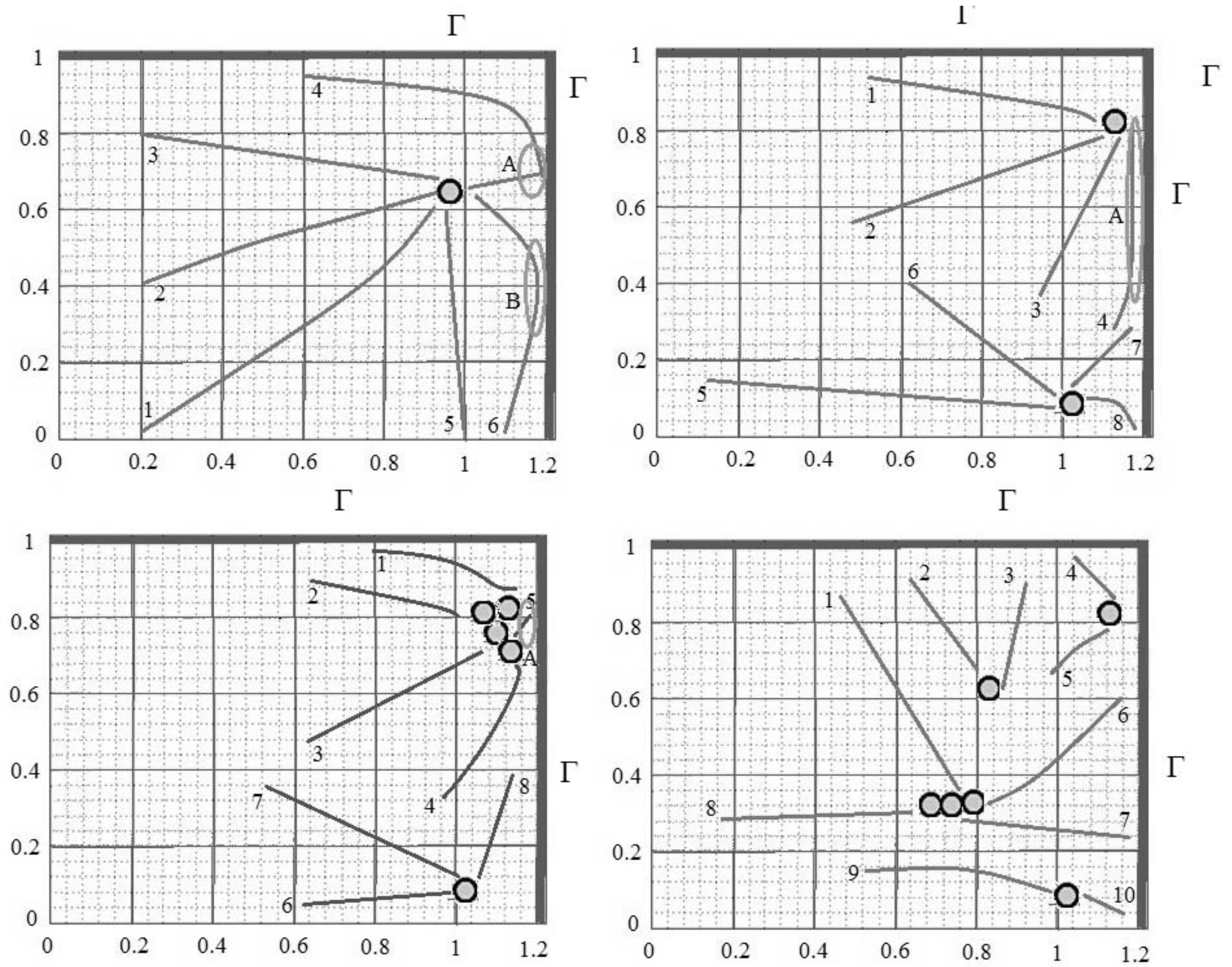

Fig. 4– Different SI trajectories in Ω.

A buffer can mitigate the stop with the progressive movements before a hard stop. Figure 5 presents different types of hard and soft stops.

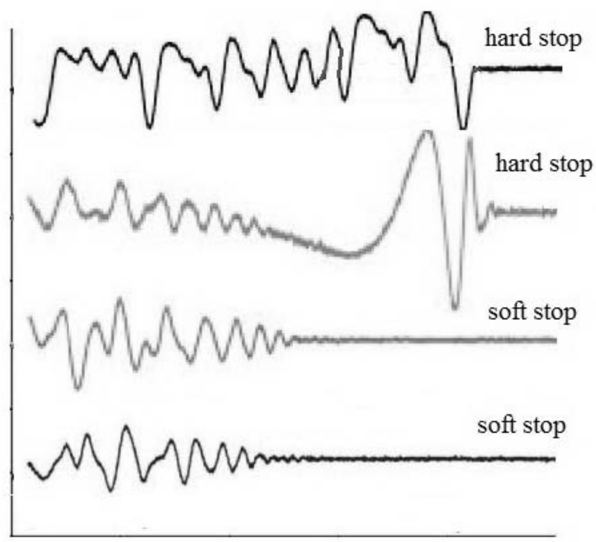

Fig. 5– Different hard and soft stops.

5. CONCLUSIONS

The paper discusses the control of a cooperatively robotic system, namely the cooperation surgeon-robot. The hybrid language is used to describe the interaction between the discrete evolution and the continuous action of the surgeon instrument (SI). The behaviour of the system includes uncertain input conditions for a class of random motion of SI and the hard and soft stops. The control algorithm corrects the movement of SI through hybrid discrete and continuous logics. The hybrid control program is an interface between a system with analog equipment described by continuous mathematics and a system with discrete software devices that processes the data. To represent the interactions surgeon-robot, the dL includes the non-determinism concept and the ability to describe the continued evolution of SI. Non-determinism is an uncertainty that is not described probabilistically. The control explores the domain geometry and eliminates the unexpected and conflictual situations.

ACKNOWLEDGEMENTS

This research was elaborated through the PN-III-P1-1.2-PCCDI-2017-022 project nr. 59/1 Martie 2018 of the National Authority for Scientific Research (ANCS, UEFISCDI), Romania. The authors acknowledge the similar and equal contributions to this article.

REFERENCES

1. FRANZLE, M., HERDE, C., *HYSAT: An efficient proof engine for bounded model checking of hybrid systems*, Formal Methods in System Design, **30**, *2*, pp. 179–198, 2007.
2. XIA, T., BAIRD, C., JALLO, G., HAYES, K., NAKAJIMA, N., *An integrated system for planning, navigation and robotic assistance for skull base surgery*, Int. J. of Med. Robot., **4**, *4*, pp. 321–330, 2008.
3. PLATZER, A., *Differential-algebraic dynamic logic for differential-algebraic programs*, J. Log. Comput., **20**, *1*, pp. 309–352, 2010.
4. PLATZER, A., *A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems*, Logical Methods in Computer Science, **8**, *4*, pp. 1–44, 2012.
5. PLATZER, A., *Differential dynamic logic for hybrid systems,* J. Autom. Reasoning, **41**, *2*, pp. 143–189, 2008.
6. HAREL, D., KOZEN, D., TIURYN, J., *Dynamic logic*, MIT Press, Cambridge, 2000.
7. KOUSKOULAS, Y., PLATZER, A., KAZANZIDES, P., *Formal methods for robotic system control software,* Johns Hopkins Appl. Technical Digest, **32**, *2*, pp. 490-498, 2013.
8. KOUSKOULAS, Y., RENSHAW, D., PLATZER, A., KAZANZIDES, P., *Certifying the safe design of a virtual fixture control algorithm for a surgical robot*, Proc. 16th International Conf. on Hybrid Systems: Computation and Control, 2013, pp. 263–272.
9. PLATZER, A., *Differential dynamic logics. automated theorem proving fir hybrid systems*, PhD Thesis, Carl Von Ossientzky Universitat Oldenburg, 2008.
10. PLATZER, A., QUESEL, J.D., *Keymaera: A hybrid theorem prover for hybrid systems,* in Automated Reasoning, 4th Int. Joint Conf., Vol. 5195 of LNCS, A. Armando, P. Baumgartner, and G. Dowek (eds.), Springer Verlag, Berlin Heidelberg, pp. 171–178, 2008.
11. PLATZER, A., *Logical analysis of hybrid systems- proving theorems for complex dynamics,* Springer Verlag, Berlin Heidelberg, 2010.
12. TOADER, A., SIRETEANU, T., CHIROIU, V., DUMITRIU, D., *Decentralized control of large scale systems*, in *Research Trends in Mechanics*, Vol. 4 (Eds. L. Munteanu, V. Chiroiu, T. Sireteanu), Edit. Academiei Române, pp. 340-368, 2010.
13. URSU, I., CHIROIU, V., MUNTEANU, H*ysteresis modeling and feedforward control of a five-story system*, Proceedings of the Romanian Academy – Series A: Mathematics, Physics, Technical Sciences, Information Science, **13**, *1*, pp. 55-61, 2012.
14. BECKERT, B., PLATZER, A., *Dynamic logic with non-rigid functions: A basis for object-oriented program varication*, in Ulrich Furbach and Natarajan Shankar (Editors), *Automated reasoning, Third International Joint Conference* (IJCAR 2006), Seattle, WA, USA, Proceedings, Vol. 4130 of LNCS, 266-280, Springer Verlag, Berlin Heidelberg, 2006.
15. ABBOTT, J., MARAYONG, P., OKAMURA, A., *Haptic Virtual Fixtures for Robot-Assisted Manipulation*, in *Robotics Research*, Vol. 28, pp. 49-64, S. Thrun, R. Brooks, and H. Durrant-Whyte (Eds.), Springer, Berlin, Heidelberg, 2007.
16. CHANG, CHIN-BOON, HO, YVONNE, CHUI, CHEE-KONG, *Automation of retinal surgery: A shared control robotic system for laser ablation,* Proc. of the 2015 IEEE Int. Conf. on Information and Automation, 1957-1962, Lijiang, China, August 2015.
17. GREENWOOD, P.E., NOVIKOV, A.A., *One-sided boundary crossing for processes with independent increments*, Theory Probab.. Appl., **31**, *2*, pp. 221-232, 1987.
18. NOVIKOV, A.A., *The martingale approach in problems on the time of the first crossing of nonlinear boundaries* (russian), Trudy Mat. Inst. Steklov., **158**, pp. 130-152, 1981.
19. NOVIKOV, A.A., *On the moment of crossing the one-sided nonlinear boundary by sums of independent random variables,* Theory Probab. Appl., **27**, pp. 688-702, 1983.

20. DENISOV, D., WACHTEL, V., *Exact asymptotic for the instant of crossing a curve boundary by an asymptotically stable random walk*, Preprint, arXiv:1110.1254, 2014.
21. GOLDBERG, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA:Addison- Wesley, 1989.
22. CHI, H, *Generation of parallel modified Kronecker sequences,* Monte Carlo Methods Appl., **19**, pp. 261-271, 2013.
23. KAZANZIDES, P., *Virtual Fixture Computation,* Note on Combining the Effects of Multiple Virtual Fixtures, 2011.

Received March 9 , 2018