Dumitru, S.I. and Chiroiu, V., 2017. Differential-dynamic logic for cooperative robotic systems.
Romanian Journal of Technical Sciences – Applied Mechanics, 62(2), pp.135–143.

DIFFERENTIAL-DYNAMIC LOGIC FOR COOPERATIVE ROBOTIC SYSTEMS

SIMONA IOANA DUMITRU, VETURIA CHIROIU

Abstract. Hybrid systems are dynamical systems with interacting discrete transitions and continuous evolutions described by differential equations. They combine the digital and continuous physical effects resulting from the cooperation between the man and the robot. The man and the robot handle the same tool, and the control algorithm has the role to prevent the man to cross the critical borders of a 2D safety workplace and to move into undesirable locations. The man manipulates the tool during his activity without interacting with the robot, but near the boundaries of the working area, the robot intervenes to guide the man towards safe trajectories. The scope of the paper is to provide a coherent logical analysis for such hybrid systems defined on 2D safe domains with critical straight lines or curved borders. The differential-dynamic logic (dL) is used to capture the logical content of the dynamics of such systems.

Key words: hybrid systems, differential-dynamic logic, cooperation man-robot, straight lines or curved borders.

1. INTRODUCTION

The scope of cooperative robotic systems is to obtain safety and efficiency in interaction between the robot and the man [1]. The control system prevents accidental events endangering the man by helping him to navigate during target procedures without crossing one or more straight lines or curved borders of a 2D safe working domain. The man freely manipulates the tool into the site without interfering with the robot. The closer the man approaches the boundaries of the working site the man enters an area defined by points that are at a critical border distance. The tool movement speed is controlled by the robot and is attenuated in proportion to the proximity to the border. When the tool reaches the border, the normal speed component is zero and the man is prevented to move further in that direction. The control algorithm is based on differential-dynamic logic (dL). The algorithm is called hybrid because it couples discreetly software behavior with the man and the physical behavior of the tool [2].

Institute of Solid Mechanics of the Romanian Academy, Bucharest

The aim of this paper is to develop the control that ensures the precision of the tool movement by avoiding the intersection with the critical borders for any inputs. The man and the robot handle the same tool. Figure 1 shows the free space where the man can manipulate the tool and the critical boundary which should not be intersected. Example of possible critical boundaries are shown in Fig. 2.

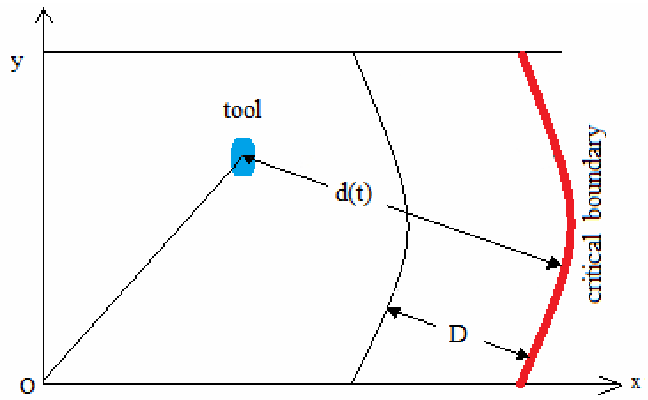

Fig. 1 – Free workspace and critical boundary which should not be intersected.

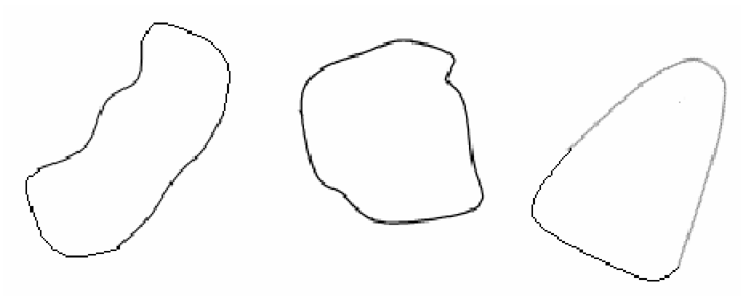

Fig. 2 – Examples of critical boundaries.

The man manipulates the tool and the robot allows the motion of the tool according to the differential equation

$$\frac{\mathrm{d}p}{\mathrm{d}t} = G(f), \tag{1}$$

where f is the force exerted by the man in the motion direction, $p > 0$ is the position of the tool, and G is the function that controls the discrete part of the system. The function G is the effect of the software of the system. The stop is set for $p = 0$. The stop can be hard or soft depending on the robotic mission. In the soft stop, the tool came to it and could not slide across it.

The overall velocity of the tool is $\dot{p}_1$ [6]

$$\dot{p}_1 = \dot{p} - \left(1 - \frac{d}{D}\right)(\dot{p} \cdot n)n, \tag{2}$$

where is the dot product of two vectors, *d* is the distance from the boundary, *n* is the unit normal to the boundary, and *D* is the safe distance from the boundary in the free zone where the man is moving without control.

The algorithm is controlling the interface between two domains, the analogue domain and the software domain, and to do this, it requires new logical approach. For an analogue component, the modeling is done by differential equations and it can be shown that it meets the given requirements. For software that processes data, the formal logic methods can be applied to obtain the properties of the software system. Hybrid control couples the analogue domain to the software domain by using a differential-dynamic logic (dL) [3–7]. The functionality of this control is extremely important in terms of safety and spatial separation of critical site boundaries during surgical or industrial manoeuvres, especially collision avoidance manoeuvres.

The approach is provocative due to the overlapping of the continuous dynamics involved and the non-rigorous discretionary control through strategic tasks of the cooperative robots [8, 9]. This approach is related to the modular walking robots with force position control [9] or to module parallel manipulator [10]. The conclusion is that the hybrid systems are difficult to analyse and verify [11].

2. THE CONTROL ALGORITHM

The novelty of this paper consists in development of dL for a hybrid system with curved critical boundaries. Such boundaries are described by standard (r, θ) parameterization of a disc. The medial axis is inspired by the circle's center point to boundaries with more complex boundaries. The medial axis of the figure is the locus of the centers of all maximal inscribed circles. A circle intersects the boundary at least two points, and there is a corresponding point on the medial axis for each boundary point. The boundary generation operator is given by [12, 13]

$$(1-t)\gamma(s) + tMA(\gamma(s)), \tag{3}$$

where $\gamma(s)$ is a point on the boundary curve, and $MA(\gamma(s))$ maps this point to the medial axis.

The algorithm (dL) is a first-order dynamic logic characterized by interaction between the robot and the man, working with real-valued functions for quantifying all possible values of system parameters and durations of evolution. The algorithm provides modal operators such as $[\alpha]$ or $<\alpha>$ that refer to transitions of hybrid system α [14–17]. The dL is working with operational models of hybrid systems, providing modular combination of arithmetic sentences describing hybrid transitions.

The algorithm works with discrete jump sets described by difference equations, the continuous evolution of the robotic system described by differential equations and the control of discrete and continuous transitions [18–22]. The discrete jump sets are instantaneous assignments of values of state variables. Continuous evolution of the system is described by differential equations. The interacting dynamics of the hybrid system is modelled as a sequence of statements representing the discrete and continuous transitions. The algorithm uses a set V of real logical variables and a signature S which is a finite set of real-valued functions and predicate symbols [23].

The variables in V can be quantified universally or existentially, and the variables in S can change their value by discrete jumps. The function takes the values of arguments and give back a function value. The predicate symbol represents as true for a vector of arguments or as false. They have the values of arguments and give either the truth-value "true" or the truth-value "false".

The set of all terms $\mathrm{trm}(S,V)$ is the smallest set such that if $x \in V$, then $x \in \mathrm{trm}(S,V)$ and if $F \in S$ is a function symbol of arity $n \geq 0$ and, $\phi_i \in \mathrm{trm}(S,V)$ for $1 \leq i \leq n$, then $F(\phi_1, \phi_2, ..., \phi_n) \in \mathrm{trm}(S,V)$.

The state variables are positions, velocities, and accelerations represented as real-valued function symbols of S of value 0. Unlike fixed symbols like 1, the state variables are flexible because their interpretation can change from state to state. There is not distinction between the discrete and continuous variables.

The language contains the dynamics of continuous evolution described by

$$x_1' = \phi_1,\ x_1' = \phi_2,\ \ldots,\ x_i' = \phi_i, \tag{4}$$

where $x_i = \phi_i$, $x_i = *$ is the nondeterministic assignment of any value, and the logical restriction R which is added $\phi_i \,\&\, R$. The sequence run is in order α and β with nondeterministic choice $\alpha \cup \beta$ and nondeterministic loop α*.

In the free zone where the man is moving without control, G is a constant multiple of f, and $\dot{p} = gf$ with g a constant. Close to the stop, a buffer attenuates the tool's motion. The stop can be hard or soft depending on the robotic mission. The directions of motion of the tool are not restricted. The equations $\dot{x} = v$, $\dot{v} = a$ define the motion of the tool with velocity v and acceleration a for $t \leq T$, where T is the biggest stop of the robot before acting.

The control combines discrete and continuous transitions using the Kleene algebras [23].

The set of hybrid programs $\mathrm{hp}(S,V)$ of the hybrid program with elements α, β is defined as the smallest set such that [11]:

1) if $x_i \in S$ is a state variable and $\phi_i \in \mathrm{trm}(S,V)$ for $1 \le i \le n$, then the discrete jump set $(x_1 = \phi_1,\ldots,\ x_n = \phi_n) \in \mathrm{hp}(S,V)$ is a hybrid program. The $x_1,\ldots,x_n$ are the state variables;

2) if $x_i \in S$ is a state variable and $\phi_i \in \mathrm{trm}(S,V)$ for $1 \le i \le n$, then $\dot{x}_i = \phi_i$ is a differential equation in which $\dot{x}_i$ is the time derivative of x_i. If χ is a first-order formula, then $(\dot{x}_1 = \phi_1,\ldots,\ \dot{x}_n = \phi_n\ \&\ \chi) \in \mathrm{hp}(S,V)$;

3) if χ is a first-order formula, then $(?\chi) \in (S,V)$ (the control constraints can be expressed using tests like $?\chi$);

4) if $\alpha,\beta \in \mathrm{hp}(S,V)$ then $(\alpha \cap \beta) \in \mathrm{hp}(S,V)$;

5) if $\alpha,\beta \in \mathrm{hp}(S,V)$ then $(\alpha;\beta) \in \mathrm{hp}(S,V)$;

6) if $\alpha \in \mathrm{hp}(S,V)$ then $(\alpha^*) \in \mathrm{hp}(S,V)$.

The safety of the control algorithm consists in that the tool must to remain in a safe place at every moment of time when the tool starts in a safe place $(p >= 0)$ for any input. The following logic predicates are defined to describe the relationship between the state variables [22]:

1) logical and $a \wedge b$;

2) logical or $a \vee b$;

3) negation $\neg a$;

4) existential quantification of reals $\exists x P(x)$;

5) universal quantification of reals $\forall x P(y)$;

6) all runs of α satisfy the condition χ $[\alpha]\chi$;

7) exists a run of α that satisfies the condition χ $<\alpha>\chi$.

The safety condition is written as

$$(p >= 0) \rightarrow [\mathrm{ctr}[(p >= 0). \tag{5}$$

To verify the temporal logic properties of the control algorithm, the model checking of successively exploring of all state transitions from the set of initial states until unsafe states, is used [18, 24–26] (Fig. 3).

Snapshots of the tool motion $d(t)$ with respect to time *t*, in which the robot must intervene to stop the advancing of the tool, is displayed in Fig. 4. When the tool is intersecting the boundary, the robot takes action to stop the movement of the tool. The tool starts to move from a safe location, and different trajectories correspond to different accelerations and velocities. When the tool is intersecting the boundary, the robot intervenes to ensure safety.

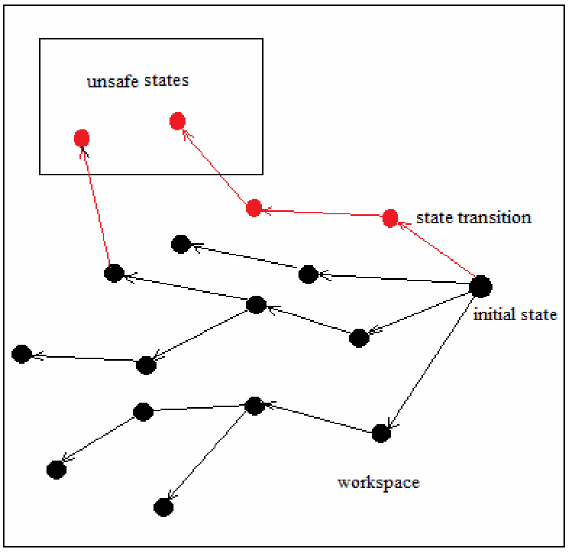

Fig. 3 – Exploring the state transitions
from the set of initial state until unsafe states.

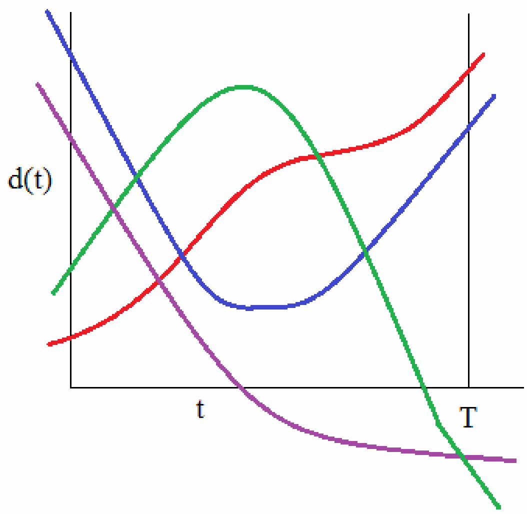

Fig. 4 – Snapshots of the tool motion.

A set of trajectories of the tool in the workspace is presented in Fig. 5. The black positions are safe, and red ones corresponds to unsafe positions that are corrected by the robot by straightening the movement into desirable locations.

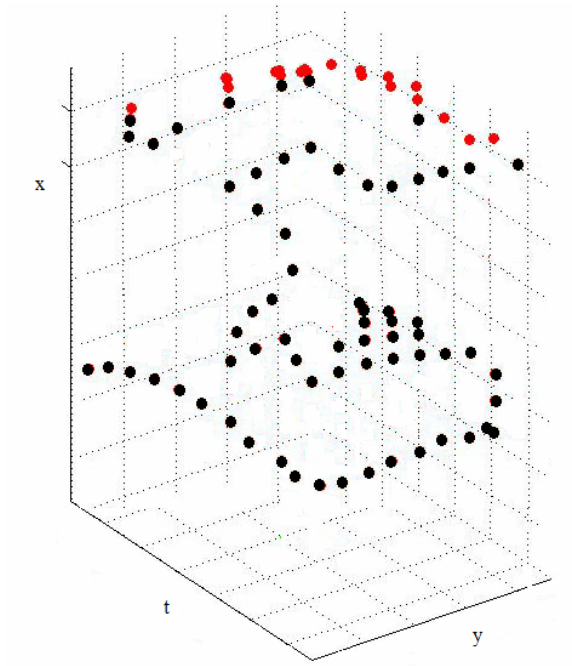

Fig. 5 – A set of safe/unsafe trajectories of the tool in the workspace.

5. CONCLUSIONS

The control algorithm for hybrid systems consists in preventing the man to cross the boundaries of a safety workplace and the moving into undesirable locations. The man manipulates the tool during his activity without control, but near the boundaries of the working area, the robot intervenes to guide the man towards safe trajectories.

The scope of the paper is to provide the differential-dynamic logic (dL) to capture the logical content of the dynamics of a robotic system where the man collaborates with the robot in a 2D working domain with straight or curved borders.

The control makes guarantees about the man and the robot behavior under all possible input conditions. The verification technique of the temporal logic properties is made by the model checking of successively exploring of state transitions from the set of initial states until unsafe states is used. Our main result proves that (dL) control captures and control the transition behavior of a hybrid system man-robot by preventing the accidental events endangering the man by intersecting one or more straight lines or curved borders of a 2D safe working domain.

Received on July 24, 2017

REFERENCES

1. VERL, A., SCHÄFFER, A.A., BROCK, O. (Eds.), *Soft robotics. Transferring theory to applications*, Springer-Verlag, 2015.
2. FRÄNZLE, 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.
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. 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.
7. KOUSKOULAS, Y., RENSHAW, D., PLATZER, A., KAZANZIDES, P., *Certifying the safe design of a virtual fixture control algorithm for a surgical robot*, Proc. of the 16th International Conf. on Hybrid Systems: Computation and Control, 2013, pp. 263–272.
8. PLATZER, A., QUESEL, J.-D., *Keymaera: a hybrid theorem prover for hybrid systems*, 4th International Joint Conference on Automated Reasoning, Vol. 5195 of Lecture Notes in Computer Science book series (Eds. A. Armando, P. Baumgartner and G. Dowek), Springer-Verlag, 2008, pp. 171–178.
9. VLĂDĂREANU, L., ION, I., *New approaches on modular walking robots with force-position hybrid control*, Rev. Roum. Sci. Techn. – Méc. Appl., **55**, *2*, pp. 167–182, 2010.
10. STAICU, St., *Kinematics of a translation-rotation hybrid parallel robot*, Ro. Techn. Sci. – Appl. Mech., **60**, *3*, pp. 171–183, 2015.
11. PLATZER, A., *Logical analysis of hybrid systems. Proving theorems for complex dynamics*, Springer-Verlag, 2010.
12. MARTIN, W., COHEN, E., *Surface completion of an irregular boundary curve using a concentric mapping*, Proceedings of the 5th International Conference on Curves and Surfaces, Saint-Malo, France, 2002, pp. 1–10.
13. ZHENG, J. J., BALL, A.A., *Control point surfaces over non-four-sided areas*, Comput. Aided Geom. Design, **14**, *9*, pp. 807–821, 1997.
14. BRANICKY, M.S., BORKAR, V.S., MITTER, S.K., *A unified framework for hybrid control: model and optimal control theo*ry, IEEE Trans. Automat. Contr., **43**, *1*, pp. 31–45, 1998.
15. CHAOCHEN, Z., JI, W., RAVN, A.P., *A formal description of hybrid systems*, in: *Hybrid Systems III*, Vol. 5195 of Lecture Notes in Computer Science book series (Eds. R. Alur, T.A. Henzinger, E.D. Sontag), Springer-Verlag, 1995, pp. 511–530.
16. CHUTINAN, A., KROGH, B.H., *Computational techniques for hybrid system verification*, IEEE Trans. Automat. Contr., **48**, *1*, pp. 64–75, 2003.
17. CLARKE, E.M., FEHNKER, A., HAN, Z., KROGH, B.H., OUAKNINE, J., STURSBERG, O., THEOBALD, M., *Abstraction and counterexample-guided refinement in model checking of hybrid systems*, Int. J. Found. Comput. Sci., **14**, *4*, pp. 583–604, 2003.
18. CLARKE, E.M., GRUMBERG, O., PELED, D.A., *Model checking*, MIT, Cambridge, MA, USA, 1999.
19. COLLINS, G.E., HONG, H., *Partial cylindrical algebraic decomposition for quantifier eliminatio*n, J. Symb. Comput., **12**, *3*, pp. 299–328, 1991.
20. COOK, S.A., *Soundness and completeness of an axiom system for program verification*, SIAM J. Comput., **7**, *1*, pp. 70–90, 1978.
21. DAMM, W., HUNGAR, H., OLDEROG, E.R., *Verification of cooperating travel agents*, Int. J. Control, **79**, *5*, pp. 395–421, 2006.

22. KOUSKOULAS, Y., PLATZER, A., KAZANZIDES, P., *Formal methods for robotic system control software*, Johns Hopkins Apl. Technical Digest, **32**, *2*, pp. 490-498, 2013.
23. KOZEN, D., *Kleene algebra with tests*, ACM Trans. Program. Lang. Syst., **19**, *3*, pp. 427–443, 1997.
24. EGERSTEDT, M., MISHRA, B. (eds.), *Hybrid systems: computation and control*, Proceedings of the 11th International Conference, HSCC 2008, St. Louis, MO, USA, Vol. 4981 of Lecture Notes in Computer Science (LNCS) book series, Springer-Verlag, 2008.
25. PLATZER, A., QUESEL, J.D., *European train control system: A case study in formal verification*, ICFEM, Vol. 5885 of Lecture Notes in Computer Science book series (eds. K. Breitman, A. Cavalcanti), 2009, pp. 246–265, Springer-Verlag.
26. PLATZER, A., QUESEL, J.D., RUMMER, P., *Real world verification*, CADE, Vol. 5663 of Lecture Notes in Computer Science book series (Ed. R.A. Schmidt), 2009, pp. 485–501, Springer-Verlag.