Introducing interval differential dynamic logic

Daniel Figueiredo
CIDMA – University of Aveiro, Portugal

Abstract Differential dynamic logic ($d\mathcal{L}$) is a dynamic logic with first-order features which allows us to describe and reason about hybrid systems. We have already used this logic to reason about biological models. Here we explore some variants of its semantics in order to obtain a simplified and more intuitive way of describing errors/perturbations, unavoidable in real-case scenarios. More specifically, we introduce interval differential dynamic logic which takes $d\mathcal{L}$ as its base and adapts its semantics for the interval setting.

Keywords: Interval differential dynamic logic, Differential dynamic logic, Interval arithmetics

1 Introduction

Differential dynamic logic ($d\mathcal{L}$) is a very expressive language which is able to describe properties of systems involving complex dynamics. Due to its specification of atomic programs, it is specially designed for *hybrid systems*, *i.e.*, those which admit continuous evolutions along with discrete events (also called *discrete jump sets*). A classic example of this is the bouncing ball (Example 1.2 from [1]) where the continuous evolution of the ball position and velocity is interrupted by a discrete event that is the bounce in the ground. $d\mathcal{L}$ was introduced by Platzer in [1], where a proof calculus was also proposed. The proof calculus is sound but not complete, which means that not all valid formulas admit a proof (a weaker version of completeness is proved in [1]); moreover the first-order structure embedded in this logic turns it undecidable. It has been applied in diverse contexts from railway, plane and automotive traffic [2, 3, 4, 5, 6] to autonomous robotics [7] and even surgical robots [8]. In all these works, the $d\mathcal{L}$ proof calculus has shown to be a powerful tool for the verification of hybrid systems correctness.

While $d\mathcal{L}$ is a really useful logic, we also look at some complementary ones. Gao has developed a verification tool for hybrid systems – dReach [9] – based on dReal [10], which evaluates δ-satisfiability for SMT formulas, *i.e.* checks satisfiability of a formula under a bounded error δ. dReach embeds the capacity of handling a bounded error and is used for reachability problems. The typical question that dReach can answer is: "Is it possible to move from an initial region to an unsafe region under a bounded error δ?". When compared to $d\mathcal{L}$ proof calculus, dReach has the advantage of being able to generally handle complex differential equations described by SMT formulas (including polynomials, trigonometric functions, exponential functions, Lipschitz-continuous ODEs, etc.). This is because dReach admits a bounded perturbation δ and it can handle numerical solutions for differential equations. However, this is also a drawback because dReach needs to compute all combinations between continuous evolutions and discrete reconfigurations, not being able to work symbolically. Since this is impossible in practice, differential dynamic logic is more suited for safety while dReach is preferred for reachability. This symbiosis has already been proposed in [11].

In this paper we introduce the basic notions for an interval version of differential dynamic logic – interval differential dynamic logic ($\mathcal{I}d\mathcal{L}$). The syntax and semantics of differential dynamic logic are adapted to interval paradigma and used in problems where variables are associated to uncertainty. Moreover, in previous works [12, 11], some examples of application of $d\mathcal{L}$ to microbiological contexts were presented. Due to the small scale of the components involved there is always some uncertainty involved and, therefore, the development of an interval version of $d\mathcal{L}$ logic is welcome. Although it was conceptually different, some related work can be found in [13, 14], where the interval paradigm was applied to dynamic logic. We assume that the reader is familiarized with $d\mathcal{L}$ notation and its basic properties.

Outline. Section 2 recalls interval arithmetics and introduces the interval paradigm. In Section 3 we introduce $\mathcal{I}d\mathcal{L}$ and illustrate the utility of this logic. Finally, we discuss some future work.

2 Interval paradigm

In this section we present the basis for interval paradigm and motivate its integration in differential dynamic logic. During the 60's, in his PhD thesis [15], Ramon Moore introduced and studied arithmetic for intervals. For a general function $f : \mathbb{R}^n \to \mathbb{R}^m$ where $f(a_1, ..., a_n) = (f_1(a_1, ..., a_n), ..., f_m(a_1, ..., a_n))$, we can obtain the function $f^{\mathcal{I}(\mathbb{R})} : \mathcal{I}(\mathbb{R})^n \to \mathcal{I}(\mathbb{R})^m$ as the function that, for closed intervals $A_1, ..., A_n$ with $A_1 \times ... \times A_n$ being contained in the domain of f,

$$f^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n) = (f_1^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n), ..., f_m^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n)) \text{ and, } \forall i \in 1, ...m$$
$$f_i^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n) = ([\min_{a_1 \in A_1, ..., a_n \in A_n} f_i(a_1, ..., a_n), \max_{a_1 \in A_1, ..., a_n \in A_n} f_i(a_1, ..., a_n)]$$

Note. In this context, we still consider a real number x as a degenerated closed interval $[x, x]$. Indeed, this notation was introduced by Moore in [15] and allows usual arithmetics to be embedded in interval arithmetics.

Example We present the generalization of some basic operations.

- $[a, b] + [c, d] = [a + c, b + d]$
- $-[a, b] = [-b, -a]$
- $[a, b] \cdot [c, d] = [\min(P), \max(P)]$ where $P = \{a \cdot c, a \cdot d, b \cdot c, b \cdot d\}$
- $[a, b]^{-1} = [\frac{1}{b}, \frac{1}{a}]$ provided that $0 \in]a, b[$

The development of interval arithmetics generates some questions about how to act on intervals. Hickey, Ju & Emden, in [16], propose some properties that interval arithmetic implementations should verify. We are interested in two:

- *Correctness*: $a_1 \in A_1$, ..., $a_n \in A_n \implies f(a_1, ..., a_n) \in f^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n)$;

- *Optimality*: $a \in f^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n) \implies \exists\, a_1 \in A_1,\, ...,\, a_n \in A_n,\; f(a_1, ..., a_n) = a$.

Correctness guarantees that an interval $f^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n)$ contains all results of pointwise evaluations of f based on point values that are elements of the argument intervals and optimality assures that $f^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n)$ generates an interval which is not wider than necessary.

Proposition If f is continuous, than the interval generalization $f^{\mathcal{I}(\mathbb{R})}$ is correct and optimal.

Proof. Since f is continuous, correctness and optimality can be trivially obtained from the Weierstrass and Bolzano theorems, respectively. □

Example Consider a function $f : \mathbb{R}^3 \to \mathbb{R}^2$ defined as $f(x, y, z) = (x + y, -xz)$. Considering $I = \{2\} \times [0, 1] \times [1, 2]$, we can compute $f^{\mathcal{I}(\mathbb{R})}(2, [0, 1], [1, 2])$ as:

$$([\min_{(x,y,z)\in I} x + y, \max_{(x,y,z)\in I} x + y], [\min_{(x,y,z)\in I} -xz, \max_{(x,y,z)\in I} -xz]) = ([2, 3], [-4, -2])$$

In a similar way, given an n-ary propositions P over reals, we define $P^{\mathcal{I}(\mathbb{R})}$, a proposition over $\mathcal{I}(\mathbb{R})$ such that $P^{\mathcal{I}(\mathbb{R})}(A_1, ..., A_n) = \mathit{true} \Leftrightarrow \forall a_1 \in A_1, ..., a_n \in A_n,\; P(a_1, ..., a_n) = \mathit{true}$

Example $A \leq^{\mathcal{I}(\mathbb{R})} B \Leftrightarrow \forall a \in A, \forall b \in B, a \leq b$

3 Interval functions for $d\mathcal{L}$

Although $d\mathcal{L}$ is able to reason about hybrid systems, in general, when one works with differential equations, it is known that small changes in initial conditions can lead to great changes in a continuous evolution. For instance, consider the following differential equation and its analytic solution:

$$\begin{cases} x' = y^2 - x \\ y' = y \end{cases} \qquad || \qquad \begin{cases} x(t) = \frac{y_0^2}{3}(e^{2t} - e^{-t}) + x_0.e^{-t} \\ y(t) = y_0.e^t \end{cases} \tag{1}$$

For $x = 0$ and y close to 0, a small change of y makes a great difference in the exact value of y because $x = y = 0$ is an unstable steady state and, for $\neq 0$, the value of y will evolve either to large positive or large negative values.

This turns to be a great issue in real life systems since it is virtually impossible to measure exact values for variables like distance and velocity. consider that a state variable x belonging to an interval ($x \in [a, b]$) is, sometimes, more useful than defining a exact value for x. For instance, piecewise-linear models in biological regulatory networks consider the behavior of a variable for certain intervals rather than the exact values of those variables [12]. To acommodate this, we propose an interval version of $d\mathcal{L}$. In our version, the variables are not evaluated as real values, but as closed intervals, leading to a methodological representation of uncertainty and experimental error. At this point, it is important to mention that $d\mathcal{L}$ syntax is also able to specify the evaluation of variables as intervals, namely $x = [a, b]$ is equivalent to $x \geq a \wedge x \leq b$.

Interval differential dynamic logic. $d\mathcal{L}$ is a dynamic logic with a first-order structure. The set of atomic programs is composed of two kinds of programs: discrete jump sets (discrete assignments) and continuous evolutions (directed by differential equations). We can obtain hybrid behavior by combining both kinds of atomic programs. The syntax of $d\mathcal{L}$ admits a set X of logical variables (which can be quantified) and a signature containing function and relation symbols as well as the set Σ_{fl} of state variables, which are variables whose interpretation is not fixed (contrary to the other symbols in Σ).

The syntax of $\mathcal{I}d\mathcal{L}$ are those of $d\mathcal{L}$ and the semantics are adapted. Real numbers are considered as degenerated intervals of the form $[a, a]$ for $a \in \mathbb{R}$. The semantics of $\mathcal{I}d\mathcal{L}$ consider a "strict" interpretation over closed intervals, *i.e.* a symbol like $+ \in \Sigma$ is interpreted as "interval sum", for instance. Also, logical and state variables are evaluated over $\mathcal{I}(\mathbb{R})$. Three functions are used to interpret formulas: an interpretation I – for rigid symbols in $\Sigma\backslash\Sigma_{fl}$; an assignment η – for logical variables in X; and a state v – for state variables in Σ_{fl}. A formula is said to be valid if it is true for every triple (I, η, v).The semantics of $d\mathcal{L}$ can be seen as a particular case of the one of $\mathcal{I}d\mathcal{L}$ since the interpretation of its formulas is done over the reals (the set of degenerated intervals). The semantics of $d\mathcal{L}$ (see [1]) is straightforwardly adapted to the interval version. However, the main difference is observed in continuous evolutions (differential equations constrained by a first-order formula χ). Given an initial state u, a system of differential equations $\overrightarrow{x}' = (f_1(\overrightarrow{x}), ..., f_n(\overrightarrow{x}))$ and a first-order formula χ, the set of reachable states is obtained by computing the solution $F(t) = (F_1(t), ..., F_n(t))$ of the differential equation f whose initial conditions are set by the state u. For each $\bar{t} \in \mathbb{R}_0^+$ we can define a reachable state v according to $F^{\mathcal{I}(\mathbb{R})}(\bar{t})$ and χ in such a way that $b \in \mathbb{R}^n \in F^{\mathcal{I}(\mathbb{R})}(\bar{t})$ if there is an initial state $a \in \mathbb{R}^n$ such that $F(\bar{t}) = b$ and $F(t)$ satisfies χ for every $t \in [0, \bar{t}]$. This definition verifies correctness and optimality because of the continuity of F.

We present two examples. One illustrating how continuous evolutions are evaluated and another one evaluating a formula of $\mathcal{I}d\mathcal{L}$.

Example Let us consider a system whose dynamics is described by the system of differential equations previously presented in Section 3. Also, for the purpose of this example, we consider a first-order condition stating the positivity of state variables $\chi \equiv x \geq 0 \wedge y \geq 0$. We obtain the following hybrid program to describe the dynamics of this system: $\alpha \equiv (x' = y^2 - x, y' = y \,\&\, x \geq 0 \wedge y \geq 0)$.

In this example, we desire to obtain the set $\rho_{I,\eta}(\alpha)$ – set of reachable states from α (see [1]). Note that I and η are not relevant because only state variables occur in α. Let us consider a state u such that $u(x) = [0, 1]$ and $u(y) = [0, 1]$). We describe the process to obtain the set of pairs (u, v) which are contained in $\rho_{I,\eta}(\alpha)$. Firstly, we need to obtain the analytical solution of the system of differential equations, which is shown in 1 with $x_0 = x(0)$ and $y_0 = y(0)$.

Let us denote $\min\limits_{(x_0,y_0)\in[0,1]^2} x(\bar{t})$ by $\underline{x(\bar{t})}$, $\max\limits_{(x_0,y_0)\in[0,1]^2} x(\bar{t})$ $\overline{x(\bar{t})}$, $\min\limits_{(x_0,y_0)\in[0,1]^2} y(\bar{t})$ by by $\underline{y(\bar{t})}$ and $\max\limits_{(x_0,y_0)\in[0,1]^2} y(\bar{t})$ by $\overline{y(\bar{t})}$. For each non-negative value of $\bar{t}$, we obtain an attainable reachable state $v_{\bar{t}} = u[x \to [\underline{x(\bar{t})}, \overline{x(\bar{t})}]][y \to [\underline{y(\bar{t})}, \overline{y(\bar{t})}]]$ for every $\bar{t}$ satisfying $val_{I,\eta}(v_t, \chi) = true$, for every $0 \leq t \leq \bar{t}$. For instance, if $v = u[x \to [0, \frac{5}{3}]][y \to [0, 2]]$, then $(u, v) \in \rho_{I,\eta}$, and $R = \ln(2)$ is the witness.

We end this section with an academic example of the evaluation of a formula of $\mathcal{I}d\mathcal{L}$.

Example

In this example, we will simply write a instead of $[a, a]$. Let us consider a system like the one from Example 3 but where the initial value of x and y is $[0, 1]$ and the value of y resets to 0 whenever it reaches the value of 5.

This is described by the following hybrid program:

$$\beta \equiv \Big((?(y \leq 5); (x' = y^2 - x, y' = y \,\&\, x \geq 0 \land 0 \leq y \leq 5)) \cup (?y = 5; y := 0) \Big)^*$$

Note that the hybrid program checks $y \leq 5$ before to proceed with the continuous evolution which never allows y to go above 5. When $y = 5$, the program can check again if $y = 5$ (because of the $*$ and $\cup$ operators) and proceed with the discrete jump set $y := 0$, setting y to $[0, 0]$. Then, the continuous evolution can resume (again, due to the $*$ operator). Note that, since we are considering the interval paradigm, when we check $y \leq 5$, we are checking if the upper limit of the interval is less or equal to 5. Finally, we evaluate the formula: $\varphi \equiv (x \leq [0, 1] \land y \leq [0, 1]) \to [\beta]x < 6$. Note that the choice of I, η is not important, since only state variables occur in this formula: $val_{I,\eta}(u, \varphi) = true \Leftrightarrow val_{I,\eta}(u, x \leq [0, 1]) = false$ or $val_{I,\eta}(u, y \leq [0, 1]) = false$ or $val_{I,\eta}(u, [\beta]x < 6) = true$.

If $u(x)(\leq)^{\mathcal{I}(\mathbb{R})}[0, 1]$ is false or $u(y)(\leq)^{\mathcal{I}(\mathbb{R})}[0, 1]$ is false, then the formula is *true*. Otherwise, we are in the same conditions as in Example 3. From Example 3, we know that is possible to reach a state $v_{\tilde{R}}$ with $v_{\tilde{R}}(y) = [0, 5]$ when $R = \ln(5)$ and, at the same state $v_{\tilde{R}}(x) = \frac{77}{15}$. Furthermore, this is the maximum value it takes for $R \in [0, \ln(5)]$ because the analytical solution for $x(r)$ is monotonically increasing. After the discrete jump set which sets y to $[0, 0]$, the continuous evolution can be run again. From the state $v_{\tilde{R}}$, the continuous evolution will permanently set y to $[0, 0]$ while $v_R(X) = [0, \frac{77}{15}.e^{-R}]$, for $R \geq \ln(5)$. Considering that $\frac{77}{15}.e^{-R}$ is a monotonically decreasing function, x will never take a value above 6. Because of this, we can conclude that $val_{I,\eta}(u, \varphi)$ is *true* for every I, η and u and thus it is valid.

4 Conclusions and future work

In this work we present an interval version of $d\mathcal{L}$ in order to make it more user-friendly in contexts where the use of intervals is more appropriated. The proof calculus of $d\mathcal{L}$ is compared to dReach, which is used to approach reachability problems in hybrid systems and that already admits a notion of δ-perturbation. Since $d\mathcal{L}$ is particularly designed for safety properties, we believe that $\mathcal{I}d\mathcal{L}$ can be an interesting complement to dReach, which is designed for reachability properties.

Although we highlight the connection between $d\mathcal{L}$ and $\mathcal{I}d\mathcal{L}$, we are interested in proving some properties relating both languages. Moreover, in the future, we intend to adapt the proof calculus of $d\mathcal{L}$ to $\mathcal{I}d\mathcal{L}$. Furthermore, in order to make $\mathcal{I}d\mathcal{L}$ more appealing, an interval version of KeYmaera (a semi-automatic prover for the $d\mathcal{L}$ proof calculus) could be developed to assist in the process of proving a $\mathcal{I}d\mathcal{L}$ formula.

Other interesting developments would be the inclusion of new interval operators and relations such as $\cap$ and $\subseteq$; the development of a fuzzy and interval logic; and study how the notion of differential equation in this paper relates with the one in [17].

Acknowledgements. This work was supported by ERDF - The European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project POCI-01-0145-FEDER-030947 and project with reference UIDB/04106/2020 at CIDMA.

References

[1] André Platzer. *Logical analysis of hybrid systems: proving theorems for complex dynamics*. Springer Science & Business Media, 2010.

[2] André Platzer and Jan-David Quesel. European train control system: A case study in formal verification. In *Formal Methods and Software Engineering*, pages 246–265. Springer, 2009.

[3] Stefan Mitsch, Sarah M. Loos, and André Platzer. Towards formal verification of freeway traffic control. In Chenyang Lu, editor, *ICCPS*, pages 171–180. IEEE, 2012.

[4] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In Michael Butler and Wolfram Schulte, editors, *FM*, volume 6664 of *LNCS*, pages 42–56. Springer, 2011.

[5] André Platzer and Edmund M Clarke. *Formal verification of curved flight collision avoidance maneuvers: A case study*. Springer, 2009.

[6] Jean-Baptiste Jeannin et al. A formally verified hybrid system for the next-generation airborne collision avoidance system. In Christel Baier and Cesare Tinelli, editors, *TACAS*, volume 9035 of *LNCS*, pages 21–36. Springer, 2015.

[7] Stefan Mitsch, Khalil Ghorbal, and André Platzer. On provably safe obstacle avoidance for autonomous robotic ground vehicles. In Paul Newman, Dieter Fox, and David Hsu, editors, *Robotics: Science and Systems*, 2013.

[8] Yanni Kouskoulas, David Renshaw, André Platzer, and Peter Kazanzides. Certifying the safe design of a virtual fixture control algorithm for a surgical robot. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 263–272. ACM, 2013.

[9] Soonho Kong, Sicun Gao, Wei Chen, and Edmund Clarke. dReach: δ-reachability analysis for hybrid systems. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 9035 of *LNCS*, pages 200–205. Springer, 2015.

[10] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In Maria Paola Bonacina, editor, *Automated Deduction CADE-24*, volume 7898 of *LNCS*, pages 208–214. Springer, 2013.

[11] Daniel Figueiredo. *Logic Foundations and Computational Tools for Synthetic Biology.* PhD thesis, Universities of Minho, Aveiro and Porto (joint doctoral program), 2020.

[12] Daniel Figueiredo, Manuel A. Martins, and Madalena Chaves. Applying differential dynamic logic to reconfigurable biological networks. *Mathematical Biosciences*, 291:10 – 20, 2017.

[13] Regivan Santiago, Benjamn Bedregal, Alexandre Madeira, and Manuel A. Martins. On interval dynamic logic: Introducing quasi-action lattices. *Science of Computer Programming*, 175:1 – 16, 2019.

[14] Regivan Santiago, Benjamín Bedregal, Alexandre Madeira, and Manuel A Martins. On interval dynamic logic. In *Lecture Notes in Computer Science, vol 10090*, pages 129–144. Springer, 2016.

[15] Ramon E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing.* PhD thesis, Stanford University, 1962.

[16] Timothy Hickey, Qun Ju, and Maarten H Van Emden. Interval arithmetic: From principles to implementation. *Journal of the ACM (JACM)*, 48(5):1038–1068, 2001.

[17] Mohadeseh Ramezanadeh, Mohammad Heidari, Omid S. Fard, and Akbar H. Borzabadi. On the interval differential equation: novel solution methodology. *Advances in Difference Equations*, 2015(1):338, 2015.