Logical Differential Constraints Based on Interval Boolean Tests

Julien Alexandre Dit Sandretto, Alexandre Chapoutot

Logical Differential Constraints Based on Interval Boolean Tests

Julien Alexandre dit Sandretto and Alexandre Chapoutot
ENSTA ParisTech, 828 bd des maréchaux, 91120 Palaiseau, France,

June 4, 2019

Abstract

Continuous-time dynamical systems play a crucial role in the study or the design of systems in various domains. Checking the satisfaction of properties on these systems is important in particular in robotics or control-command systems. Constraint satisfaction problems is a well-suited framework for this purpose and recent papers extend this framework to deal with differential constraints. This article proposes an improvement of constraint differential satisfaction framework by providing a new solving algorithm based on interval Boolean functions.

1 Introduction

Continuous-time dynamical systems play a crucial role in the study or the design of systems in various domains. In particular, prediction of behaviors of systems or satisfaction of properties can be obtained using numerical simulation methods applied to these mathematical models.

Recently, an extension of constraint satisfaction problem (CSP) has been proposed in [2] to deal with dynamical systems which can be used to check temporal properties. This framework named *Set-based Constraint Satisfaction Differential Problems* (SCSDP) has as main feature the use of set-based constraints in order to have *specification robustness* against bounded uncertainties and also to have *model robustness* against model approximation. In consequence, this framework increases reliability of the computed solutions with respect to the real system. These two concepts are explained in the following paragraphs.

Properties of systems, or a specification, are usually given with margins in order to ensure safety or to increase robustness. For example, for autonomous vehicle reaching a particular point x in a map is given up to a given precision δ as sensors produce approximate information on the environment. In consequence, system properties are usually defined over sets of admissible behaviors, *i.e.*, a vehicle is considered to have reached x if its position p is such that $x-\delta \leqslant p \leqslant x+\delta$. Nonetheless, the combination of inequalities in classical CSP framework can be used to model such properties but may also lead to equality relation as in $x \geqslant 0 \wedge x \leqslant 0$. Usually, such equality constraints require relaxation techniques in CSP solver, *i.e.*, $x \in [-\varepsilon, \varepsilon]$. This relaxation should not be taken place inside solving algorithm, which usually applies the same relaxation for all equality constraints,

but instead specification should be written in order to emphasize the margins that is important for the properties.

A system model M is usually only an approximation of the true system S, especially for continuous-time systems defined by ordinary differential equations (ODEs), where some unknown data have to be considered. For example, the position of a mobile robot is usually known up to a given precision depending on sensors. In the framework of bounded uncertainties, such data are represented by bounded sets and so the model M will be associated with a set of possible trajectories. Hence, properties on M should consider sets of values instead of a single value.

The main contribution of this article is to defined a new solving algorithm for SCSDP based on interval Boolean function. This improvement in regards to the solving algorithm presented in [2] allows for a simplification of the construction of complex constraints which may involve disjunctive logical operator.

2 Set-based Constraint Satisfaction Differential Problems

In [2], a general class of differential equations are considered which can represent ODEs, Differential Algebraic Equations (DAEs) of index 1, and a mix of these equations with additional constraints, *e.g.*, to model energy preservation. More precisely, differential systems are of the form

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{F}(t, \mathbf{y}(t), \mathbf{x}(t), \mathbf{p}), \\ 0 = \mathbf{G}(t, \mathbf{y}(t), \mathbf{x}(t)) \\ 0 = \mathbf{H}(\mathbf{y}(t), \mathbf{x}(t)) \end{cases} . \tag{1}$$

Non-linear functions $\mathbf{F} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$, $\mathbf{G} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$, $\mathbf{H} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $t \in [0, t_{\text{end}}]$, $\mathbf{y}(0) \in \mathscr{Y}_0$ and $\mathbf{p} \in \mathscr{P}$ are considered. More precisely, Initial Value Problems (IVP) for parametrized differential equations are considered over a finite time horizon $[0, t_{\text{end}}]$. Note that a bounded set of initial values and a bounded set of parameters are considered in this framework. This necessitates dealing with set of trajectories solution of Equation (1). We assume classical hypothesis on $\mathbf{F}$, $\mathbf{Q}$, and $\mathbf{H}$ to ensure the existence and uniqueness of the solution of Equation (1).

We denote by $\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P})$ the solution set

$$\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P}) = \{\mathbf{y}(t; \mathbf{y}_0, \mathbf{p}) \mid t \in \mathscr{T}, \mathbf{y}_0 \in \mathscr{Y}_0, \mathbf{p} \in \mathscr{P}\} . \tag{2}$$

Intuitively, $\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P})$ gathers all the points reached by the solution $\mathbf{y}(t; \mathbf{y}_0, \mathbf{p})$ of Equation (1) starting from all scalar initial values $\mathbf{y}_0$ and all scalar parameters $\mathbf{p}$. The proposed framework aims at checking if $\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P})$ fulfills some specification defined in terms of set-based constraints.

To avoid problematic issue due to equality constraints set-based constraints are considered. More precisely, *inclusion* and *intersection* operators are considered. More precisely, constraints of the form

$$\mathbf{g}(\mathscr{A}) \subseteq \mathscr{B} \qquad \text{and} \qquad \mathbf{g}(\mathscr{A}) \cap \mathscr{B} = \emptyset,$$

where $\mathscr{A}$ and $\mathscr{B}$ are real compact sets and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ is a non-linear function. The lifting of $\mathbf{g}$ to sets is defined as usual by $\mathbf{g}(\mathscr{X}) = \{\mathbf{g}(x) : x \in \mathscr{X}\}$.

Note that these constraints can be seen as Boolean functions but while, from a mathematical formulation, the truth value always exist, it may not be the case when they can be computed. The contribution of this article is to give a formulation of theses constraints in terms of *interval test function* considering *interval Boolean values*, see Section 3.

The handling of differential constraints here follows the approach given in [3] in the exception of the solution operator of Equation (1), which is here represented as a set of solution $\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P})$ in order to unify the objects manipulated into constraints which are also sets. Set-based Constraints Satisfaction Differential Problems (SCSDP) based on a set-membership constraints and embedding differential constraints can now be defined.

Definition 1 (SCSDP) *A SCSDP is a NCSP made of*

- *a finite set $\mathscr{S}$ of differential systems S_i as defined in Equation* (1)*.*
- *a finite set of variables $\mathscr{V}$ including the parameters of the differential systems S_i, i.e., $(\mathbf{y}_0, \mathbf{p})$, a time variable t and some other algebraic variables $\mathbf{q}$;*
- *a domain $\mathscr{D}$ made of the domain of parameters $\mathbf{p} : \mathscr{D}_p$, of initial values $\mathbf{y}_0 : \mathscr{D}_{y_0}$, of the time horizon $t : \mathscr{D}_t$, and the domains of algebraic variables $\mathscr{D}_q$;*
- *a set of constraints $\mathscr{C}$ which may be defined by inclusion or disjunction constraints over variables of $\mathscr{V}$ and special variables $\mathscr{Y}_i(\mathscr{D}_t, \mathscr{D}_{y_0}, \mathscr{D}_p)$ representing the set of the solution of S_i in $\mathscr{S}$.*

3 Interval Inclusion Test for SCSDP

Presented in [4], **inclusion tests** can be used to prove that all points in a box verify a property. These tests exploit the notion of interval Boolean values. The Boolean set is defined such that $\mathbb{B} = \{\texttt{false}, \texttt{true}\} = \{0, 1\}$. An interval Boolean is a subset of $\mathbb{B}$, *i.e.*, an element of the interval Boolean set $\mathbb{IB} = \{\emptyset, [0,0], [1,1], [0,1]\}$. The particular values $\emptyset$ and $[0,1]$, standing for the set $\{0,1\}$, mean, respectively, impossible and undetermined.

The operations on Booleans are extended to **interval Booleans**. If $[a] \in \mathbb{IB}$ and $[b] \in \mathbb{IB}$, the operations are defined such that:

- $[a] \wedge [b] = \{a \wedge b \mid a \in [a], b \in [b]\}$;
- $[a] \vee [b] = \{a \vee b \mid a \in [a], b \in [b]\}$;
- $\neg[a] = \{\neg a \mid a \in [a]\}$.

The behavior of undetermined interval of interval Boolean values is as follow: $0 \wedge [0,1] = 0$, $1 \wedge [0,1] = [0,1]$, $0 \vee [0,1] = [0,1]$, $1 \vee [0,1] = 1$.

A test function $\mathbf{t}$ maps $\mathbb{R}^n$ to $\mathbb{B}$. The interval extension of $\mathbf{t}$ is the inclusion test $[\mathbf{t}]$ mapping $\mathbb{IR}^n$ to $\mathbb{IB}$, such that for any $[\mathbf{x}] \in \mathbb{IR}^n$:

- $([\mathbf{t}]([\mathbf{x}]) = 1) \Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], \mathbf{t}(\mathbf{x}) = 1)$
- $([\mathbf{t}]([\mathbf{x}]) = 0) \Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], \mathbf{t}(\mathbf{x}) = 0)$

Example 1 *Consider the simple test* $t : \mathbb{R} \to \{0,1\}$ *such that*

$$t : x \mapsto x < 2$$

such that $t(x) = 0$ *if* $x \geq 2$ *and* $t(x) = 1$ *otherwise. The associated inclusion test is* $[t] : \mathbb{IR} \to \mathbb{IB}$ *such that*

$$[t] : [x] \mapsto [x] < 2$$

$[t]([x]) = 0$ *if* $\underline{x} \geq 2$, $[t]([x]) = 1$ *if* $\overline{x} < 2$, *and* $[t]([x]) = [0,1]$ *otherwise. For example,* $[t]([1,3]) = [0,1]$.

Inclusion tests for sets can also be defined. Let $\mathscr{A}$ be a subset of $\mathbb{R}^n$, an inclusion test $[\mathbf{t}_{\mathscr{A}}]$ for $\mathscr{A}$ is an inclusion test for the test $\mathbf{t}_{\mathscr{A}}(\mathbf{x}) \iff (\mathbf{x} \in \mathscr{A})$, *i.e.*, $[\mathbf{t}_{\mathscr{A}}]$ satisfies:

- $[\mathbf{t}_{\mathscr{A}}]([\mathbf{x}]) = 1 \Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], \mathbf{t}_{\mathscr{A}}(\mathbf{x}) = 1) \iff ([\mathbf{x}] \subset \mathscr{A})$;
- $[\mathbf{t}_{\mathscr{A}}]([\mathbf{x}]) = 0 \Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], \mathbf{t}_{\mathscr{A}}(\mathbf{x}) = 0) \iff ([\mathbf{x}] \cap \mathscr{A} = \emptyset)$;
- $[\mathbf{t}_{\mathscr{A}}]([\mathbf{x}]) = [0,1]$ (nothing can be determined) about the inclusion of $[\mathbf{x}]$ in $\mathscr{A}$.

Contribution: We propose to transpose the approach based on interval Booleans for the inclusion test to the SCSDP. Operations 2 on sets used to define SCSDP have been transposed to intervals in a previous paper [2], with evaluation still in Boolean set $\mathbb{B}$. In order to obtain a unified formalism, with easier soundness understanding and larger expressivity, we propose here to use interval Booleans.

Operations we consider in our SCSDP formalism is as follow:

- $[\mathbf{t}]_{\mathbf{g},\mathscr{A},\mathscr{B}} = 1 \Rightarrow (\forall \mathbf{x} \in \mathscr{A}, \mathbf{g}(\mathbf{x}) \in \mathscr{B}) \iff \mathbf{g}(\mathscr{A}) \subset \mathscr{B}$;
- $[\mathbf{t}]_{\mathbf{g},\mathscr{A},\mathscr{B}} = 0 \Rightarrow (\forall \mathbf{x} \in \mathscr{A}, \mathbf{g}(\mathbf{x}) \notin \mathscr{B}) \iff \mathbf{g}(\mathscr{A}) \cap \mathscr{B} = \emptyset$;
- $[\mathbf{t}]_{\mathbf{g},\mathscr{A},\mathscr{B}} = [0,1]$ nothing can be determined.

Example 2 *The following Boolean formula* ϕ *can be defined*

$$\phi \equiv \left([\mathbf{t}]_{\mathbf{g},\mathscr{A},\mathscr{B}} \vee [\mathbf{t}]_{\mathbf{f},\mathscr{A},\mathscr{C}}\right) \wedge [\mathbf{t}]_{\mathbf{h},\mathscr{C},\mathscr{B}}$$

Remark in SCSDP, the set $\mathscr{A}$ can be the set $\mathscr{Y}(\mathscr{T}, \mathscr{Y}_0, \mathscr{P})$ to define dynamical constraints. In that case, $\mathscr{A}$ is parametrized by t. Hence, quantification over time can be considered, *e.g.*, to check that the set of trajectories is inclued in a particuluar safety set $\mathscr{B}$ for all time or that the solution reaches a particular set at a given time.

Solving a problem as given in Example 2 means determining sets $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$ such that the formula ϕ is satisfied. With interval methods, it is common to enclose a set by an inner and an outer paving [4] with a branching algorithm. The algorithm presented in Algorithm 1 is able to solve a SCSDP assuming that *Formula* is an interval Boolean function.

4 Conclusion

An extension of the *Set-based Constraints Satisfaction Differential Problems* is proposed in this paper. The main idea is to equip the formalism with a full interval-based approach by considering interval of Boolean values. It provides a clearer soundness and a larger expressivity. Moreover, a basic branching algorithm can be used to solve a SCSDP, even if disjunctive logic operations appear in constraints. We are currently implementing the presented contribution in DynIbex [1]. As future work, experimentation will be performed.

Algorithm 1 Branching algorithm for SCSDP

Require: $Stack = X$, $Stack_{in}$, $Stack_{out}$, $Formula$
  while $Stack \neq \emptyset$ **do**
    Pop $[x]$ in $Stack$
    Solve differential equation with $[x]$
    if $Formula([x]) = [1,1]$ **then**
      Push $[x]$ in $Stack_{in}$
    else if $Formula([x]) = [0,0]$ **then**
      Push $[x]$ in $Stack_{out}$
    else if $width([x]) > \varepsilon$ **then**
      $[x] = [x_1] \cup [x_2]$
      Push $[x_1]$ and $[x_2]$ in $Stack$
    end if
  end while

References

[1] Julien Alexandre dit Sandretto and Alexandre Chapoutot. DynIBEX: a Differential Constraint Library for Studying Dynamical Systems. In *Conference on Hybrid Systems: Computation and Control*, 2016. Poster.

[2] Julien Alexandre Dit Sandretto, Alexandre Chapoutot, and Olivier Mullier. Constraint-Based Framework for Reasoning with Differential Equations. In *Cyber-Physical Systems Security*, pages 23–41. Springer International Publishing, 2018.

[3] Alexandre Goldsztejn, Olivier Mullier, Damien Eveillard, and Hiroshi Hosobe. Including ode based constraints in the standard CP framework. In *Principles and Practice of Constraint Programming*, volume 6308 of *LNCS*, pages 221–235. Springer, 2010.

[4] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. *Applied Interval Analysis*. Springer, 2001.