Combining Interval Arithmetic and Three-Valued Temporal Logics for the Verification of Analog Systems

R. Gentilini, K. Schneider
Reactive System Group, Computer Science Department, Kaiserslautern University (Germany)
{gentilin, Klaus.Schneider}@informatik.uni-kl.de

A. Dreyer
Fraunhofer-Institut für Techno-und Wirtschaftsmathematik
Fraunhofer-Platz 1 67663 Kaiserslautern (Germany)
alexander.dreyer@itwm.fraunhofer.de

Abstract

We deal with the problem of designing suitable languages for the modeling *and the* automatic verification *of properties over analog circuits. To this purpose, we suitably enrich classical temporal logics with basic formulæ allowing to model arbitrary functions relating analog variables. We show how to accomplish the task of automatically check the resulting* CTL_f *formulæ on analog circuits. To this purpose, we extend to the analog context a number of techniques for the abstraction and the verification of digital systems, based on three-valued temporal logics.*

1. Motivation

Analog circuit design is of great importance in nowadays microelectronics: Analog components play a growing role in major application fields like automotive industry and telecommunication, and the development of the corresponding analog integrated circuits is recognized as a severe task [6, 10]. In order to avoid a large number of defective prototypes, it is necessary to check significant properties of an analog circuit before manufacturing it. Traditionally, the above task relies largely on the cooperation of expert knowledge, manual calculation, and numerical circuit simulation [6]. A quite recent and active research field, named as *Symbolic Analysis* of analog circuits, aims at (1) endowing the verification process of some degree of *automation* (2) coping with *variability* of parameters and input signals [18, 10, 17]. The two above objectives are fullfilled by providing computer algebra algorithms to simplify the mathematical model of an analog circuit. Typically, such procedures work by detecting and deleting subformulæ whose influence on the evolution of the system is insignificant. Despite the promising developments, *system analysis* is still far from being a *completely automatic* method for the verification of analog circuits.

On the contrary, in the context of digital circuits, *model checking* allows to verify (1) entirely automatically (2) along all possible executions of the systems, properties that are *formally* stated in a temporal logic. A natural way to apply model checking to the context of analog circuits verification is the extraction of some finite digital abstraction from the corresponding analog model. Such an approach has been adopted in [9, 8], where the authors qualitatively proved its effectiveness by applying

it to some concrete (and well-understood) examples. As part of their work, the authors proposed an enrichment of classical temporal logic languages, recognizing the inadequacy of the latter to model analog properties. However, their extension is still not satisfactory: As an example, simple properties comparing the values of analog variables along their evolution can still not be dealt with in [9, 8]. Indeed, even if a constraint of the kind $x < y$ could be expressed in the language in [9, 8], a major problem to its verification would be the impossibility to define grid abstractions of the analog system's (infinite) state-space, such that on each grid box, the property is either true or false.

In this paper, we face the problems of (1) designing a suitable language for modeling analog properties, (2) providing tools to verify properties expressed in such a language over abstractions of analog models (3) formally state the relation of preservation of properties from the abstract model to the (concrete) analog system. The latter problem—fundamental in the spirit of *formal* verification techniques—is indeed not taken in consideration in [9, 8]. It can be proved that their framework allows to preserve only truth of universal quantified properties from the abstract to the concrete models [14]. Our technique, which combines three-valued temporal logic with interval arithmetic techniques, allows us to apply *multi-valued* model checking to analog systems (rather than classic 2-valued model checking). We show that in the context of analog circuit verification, the gain of using multivalued model checking is the same achieved for digital (infinite) systems [1]: Namely both truth and falsity of general branching[1] temporal formulæ are preserved from abstract to concrete models.

2. Preliminaries

This section contains some preliminary notions as well as the notation used in the rest of the paper.

2.1 Interval Arithmetic

Interval arithmetic assumes closed intervals over the reals as its fundamental computational object. We denote by $\mathbb{IR}$ the set of all such closed intervals of $\mathbb{R}$. Since $\mathbb{R}$ and $\emptyset$ are both open and closed, they both belong to $\mathbb{IR}$, and any element of $\mathbb{IR}$ can be written in one of the following form $[\underline{x}, \overline{x}]$, $]-\infty, \overline{x}]$, $[\underline{x}, +\infty[$, where $\underline{x} \leq \overline{x} \in \mathbb{R}$. Each elementary function on $\mathbb{R}$ (+,-,*,/,*sin*, *cos*, *log*, e^x, x^a) can be easily redefined in the context of $\mathbb{IR}$, by considering the bounds of the corresponding closed intervals. As an example, the rules for recasting each binary operator $\diamond \in \{+, -, *\}$ on the domain $\mathbb{IR}$ are given by:

$$\begin{array}{l} [\underline{x}, \overline{x}] \diamond [\underline{y}, \overline{y}] = [\underline{z}, \overline{z}] \text{ , for } \diamond \in \{+, -, *\} \text{ ,} \\ \text{with } \underline{z} = \min \{\underline{x} \diamond \underline{y}, \underline{x} \diamond \overline{y}, \overline{x} \diamond \underline{y}, \overline{x} \diamond \overline{y}\} \text{ ,} \\ \text{and } \overline{z} = \max \{\underline{x} \diamond \underline{y}, \underline{x} \diamond \overline{y}, \overline{x} \diamond \underline{y}, \overline{x} \diamond \overline{y}\} \text{ .} \end{array} \quad (1)$$

A *box* of dimension n is a subset of $\mathbb{R}^n$ that can be defined as the cartesian product of n closed intervals. We denote by $\mathbb{IR}^n$ the set of all boxes of dimension n.

Definition 1 (Inclusion Function). *The function $F : \mathbb{IR}^n \mapsto \mathbb{IR}^m$ is an inclusion function for the function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ if and only if:*

$$\forall b \in \mathbb{IR}^n (f(b) \subseteq F(b))$$

Let $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ be a function expressed as a finite composition of elementary real valued functions. The *natural inclusion function* of f is the inclusion function $[f] : \mathbb{IR}^n \mapsto \mathbb{IR}^m$, obtained by replacing each elementary function by its interval counterpart. In the rest of the paper we deal with functions obtained by composing elementary functions, and it is sufficient to assume the use of natural inclusion functions as the corresponding inclusion functions.

A more substantial introduction to the subject of interval arithmetic can be found in [13].

[1] i.e. combining universal and existential path quantifiers operators

2.2 Analog Circuits

Analog circuits can be formally described via a set of differential algebraic equations (DAE):

$$F(x(t), \dot{x}(t), y(t)) \tag{2}$$

where $x \in \mathbb{R}^m$ denotes the vector of system variables, $\dot{x} \in \mathbb{R}^m$ denotes the corresponding time derivatives, and $y \in \mathcal{R}^{m'}$ represents the vector of static variables, like input signals. However, most real-world circuits can be modelled in the so called *semi-explicit* form:

$$\begin{array}{c} \dot{x}(t) = F(x(t), y(t)) \\ 0 = g(x(t), y(t)) \,, \end{array} \tag{3}$$

where g denotes algebraic (i. e. non-differential) equations. In this paper, we refer to analog circuits models as *concrete models*[2], and we denote them by tuples of the form $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$, where:

- X^I is a set of *independent variables*.
- X^D is a set of *dependent variables*, and $X^D \cap X^I = \emptyset$.
- $\mathcal{D}$ is a differential rule describing the evolution of dependent variables, that can be written as $\mathcal{D} = \bigwedge_{1 \leq i \leq p} (\dot{x}_i = f_i(x_1, \ldots, x_n))$, where $p = |X^D|$ and $n = |X^D \cup X^I|$.
- $\mathcal{I}$ is an invariant on $X^D \cup X^I$.

Let $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ be a concrete system and let $n = |X^D \cup X^I|$. A *state* in $\mathcal{C}$ is a vector $v \in \mathbb{R}^n$ of real values. A *trajectory* of $\mathcal{C}$ is a function $\rho : \mathbb{R} \mapsto \mathbb{R}^n$, which describes a possible time evolution of dependent/independent variables according to the differential rule $\mathcal{D}$ and to the invariant condition $\mathcal{I}$.

3. CTL$_f$: A Temporal Logic for Specification of Analog Properties

In this section, we develop a suitable temporal logic for specification of the temporal evolution of analog and mixed signals. Our CTL$_f$ language can be viewed as an enrichment of the classical CTL language [3, 16], originally designed for the temporal analysis of digital finite systems. In particular, we allow CTL$_f$ 'basic formulæ' to state the membership in boxes of values given by arbitrary elementary functional relations. The evolution of these functional relations is then modeled using classical CTL temporal operators. More precisely, the CTL$_f$ language syntax is given in Definition 2.

Definition 2 (CTL$_f$ Syntax). *Let $X = \{x_1, \ldots, x_n\}$ to be a finite set of real valued variables. The set of CTL$_f$ formulæ is defined by the following grammar:*

$$\begin{array}{lll} \phi ::= f(x_{i_1}, \ldots, x_{i_m}) \triangleright I & & \textit{(basic formulæ testing membership in box } I\textit{)} \\ \quad \neg\phi | \phi \vee \phi & & \textit{(boolean combinators)} \\ \quad \mathsf{E}\phi\mathsf{U}\phi | \mathsf{A}\phi\mathsf{U}\phi & & \textit{(temporal combinators \& path quantifiers)} \end{array}$$

where $1 \leq i_1 \leq \ldots \leq i_m \leq n$, $f : \mathbb{R}^m \mapsto \mathbb{R}^p$ is an arbitrary composition of elementary functions, and I is a box in $\mathbb{IR}^p$.

Note that we do not include in our CTL$_f$ logic the usual neXt temporal operator [2]. In fact, since we want to use CTL$_f$ to model *analog* properties, the notion of *next* point in time is meaningless in our context. Other temporal operators (as for example, the operators stating that some property holds Globally or Finally on some path) can be encoded in our grammar (endowed of the only Until temporal operator) by standard definitions [3, 16]. Given a state $v = \{v_1, \ldots, v_n\}$ in an analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$, the value of CTL$_f$ formulæ on v is defined by considering the trajectories departing from v, as formally stated in Definition 3.

[2] In contrast to *abstract models* that will be introduced in further sections.

Definition 3 (**CTL**$_f$ **Concrete Semantics).** *Let* $v = (v_1, \ldots, v_n)$ *be a state in the concrete system* $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ *and let* ϕ *be a* **CTL**$_f$ *formula. Then, the two-shaped value* $[v \models \phi] \in \{\mathsf{tt}, \mathsf{ff}\}$ *is inductively defined as follows:*

- $[v \models f(x_{i_1}, \ldots, x_{i_m}) \triangleright I] = \mathsf{tt}$ *iff* $f(v_{i_1}, \ldots, v_{i_m}) \in I$.
- $[v \models \phi_1 \vee \phi_2] = \mathsf{tt}$ *iff* $[v \models \phi_1] = \mathsf{tt} \vee [v \models \phi_2] = \mathsf{tt}$.
- $[v \models \neg\phi_1] = \mathsf{tt}$ *iff* $[v \models \phi_1] = \mathsf{ff}$.
- $[v \models \mathsf{E}\phi_1\mathsf{U}\phi_2] = \mathsf{tt}$ *iff there exists a trajectory* ρ *departing from* v *and a point of time* t *such that* $[\rho(t) \models \phi_2] = \mathsf{tt}$ *and for all* $0 \leq t' \leq t$, $[\rho(t') \models \phi_1] = \mathsf{tt}$.
- $[v \models \mathsf{A}\phi_1\mathsf{U}\phi_2] = \mathsf{tt}$ *iff for each trajectory* ρ *departing from* v, *there exists a point of time* t *such that* $[\rho(t) \models \phi_2] = \mathsf{tt}$ *and for all* $0 \leq t' \leq t$, $[\rho(t') \models \phi_1] = \mathsf{ff}$.

The modelling of analog properties by means of **CTL**$_f$ is briefly discussed in the following example.

Example 1. Figure 1 depicts a nonlinear analog circuit consisting of four bipolar transistors. The circuit is know as *square root function block* since the output current I_0, which is measured through the independent voltage source VLOAD, is roughly given by the square root of the product of input currents I_I, I_B, and a constant factor. In particular, text books on analog circuit design [6] state the

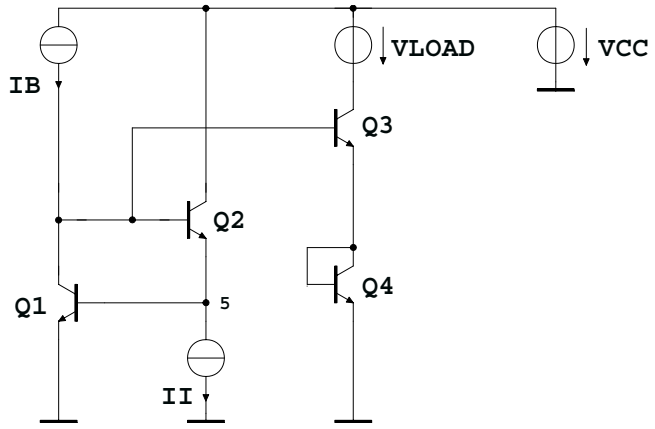

Fig. 1. Schematics of a square root function block

following approximate formula for the static input/output behavior):

$$I_0 = \beta \sqrt{I_I} \sqrt{I_B} \ , \tag{4}$$

where β is a constant related to parameters of the transistors $Q_1, \ldots, Q_4$. This formula was computed using manual approximations based on expertise of the circuit. It has been shown in [7], that this rule of thumb can also be derived from a fully-featured system of circuit equations using automated combined numeric/symbolic approximation methods. These are model reduction techniques, which successively remove unnecessary terms from the equations. Even tough this approach behaves well for practical examples, the validity of the reductions lack formal verification yet.

Property 4 can be naturally modeled by means of the **CTL**$_f$ language. Specifically, formula 5 below states that "Globally along the evolution of all trajectories, the relation $(I_0 - \beta \sqrt{I_I} \sqrt{I_B}\,) \in [-\delta, +\delta]$" holds:[3]

$$\mathsf{AG}((I_0 - \beta \sqrt{I_I} \sqrt{I_B}\,) \triangleright [-\delta, +\delta]) \tag{5}$$

Broadly speaking, our logic allows to naturally encode:

[3] Note, that in this case, smaller δ corresponds to I_0, which are closer to the idealized behaviour of $\beta \sqrt{I_I} \sqrt{I_B}$.

– general properties on the relative value of input/system variables of an analog model, while the latter evolves (as in Formula 5, above).
– general properties relating *the rate* at which system variables decrease/increase along their evolution, since the corresponding derivative functions can be explicitly stated in basic CTL_f formulæ. For example, consider the analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$, where $\mathcal{D} = \bigwedge_{1 \leq i \leq p} \dot{x}_i = f_i(x_1, \ldots, x_n)$. Formula 6 below states the existence of a trajectory that finally reaches a point, from where on variable x_1 decreases at least as fast than variable x_2

$$\mathsf{EFEG}((f_1(x_1, \ldots, x_n) - f_2(x_1, \ldots, x_n)) \rhd] - \infty, 0]) \tag{6}$$

3.1 CTL_f Three-Valued Abstract Semantics

It is well known [11, 12] that simple problems such as *reachability*[4] are generally computationally intractable for analog systems, or even not decidable for hybrid systems[5] endowed with rather trivial continuous dynamics. In our context, the systems motivating the analysis are characterized by complex analog behaviours. Moreover, it is likely that the analog properties that we want to check for validity can not be expressed in any decidable theory of the reals[6]. Hence, to reconcile the need of reasoning on general analog beheaviors with the spirit of *automatic* formal analysis, we develop here a notion of *three-valued abstract* semantics for our logic CTL_f. Specifically, we first define the concept of *box modal abstraction* structure (cf. Definition 4). On this ground, we use the power of interval arithmetic to evaluate CTL_f formulæ to one of the *three* values $\{\mathsf{tt}, \mathsf{ff}, \bot\}$. As we demonstrate in further sections, this framework built up by combining interval arithmetic and three-valued temporal logics, will allow us to:

– Automatically check the three-valued abstract semantics of CTL_f formulæ;
– Defining suitable *preorder relations* between analog models (concrete systems) and box modal abstractions (abstract systems), formally guaranteeing the *preservation* of the true/false value from abstract to concrete CTL_f semantics.

Definition 4, below, formally introduces the concept of box modal abstraction (BMA). Intuitively, the term *box* in BMA is reminding that we make use of boxes as states of our abstractions. The term *modal* instead, refers to the fact that, in order to be able to both *overapproximate* and *underapproximate* trajectories in the concrete systems, we accordingly endow our abstractions of two kinds of transitions (may/must transitions). Here, we are inspired by the work tracing back to [4, 5, 1], related to the abstraction of discrete (possibly infinite states) systems.

Definition 4 (Box Modal Abstraction (BMA)). *Let $X = \{x_1, \ldots, x_n\}$ be a set of n real valued variables. A* Box Modal Abstraction *on X is a tuple $\langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$, where:*

– *$\mathcal{B}$ is a finite collection of boxes over $\mathbb{R}^n$.*
– *$\overset{must}{\rightarrow}$ and $\overset{may}{\rightarrow}$ are binary relations on $\mathcal{B}$. We assume that $\overset{must}{\rightarrow} \subseteq \overset{may}{\rightarrow} \subseteq \mathcal{B} \times \mathcal{B}$ and that $\overset{may}{\rightarrow}$ is total (i.e. each box has at least one may transition departing from it).*

Definition 5 (Must & May Paths). *Let $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ be a* BMA *on $X = \{x_1, \ldots, x_n\}$. A may-path (must-path) in $\mathcal{A}$ is an infinite sequence of boxes $\{b_i\}_{i \in \mathbb{N}}$ such that for all $i \geq 0$ it holds that $b_i \overset{may}{\rightarrow} b_{i+1}$ ($b_i \overset{must}{\rightarrow} b_{i+1}$). We use the notation $\{b_i\}^{may}_{i \in \mathbb{N}}$ ($\{b_i\}^{must}_{i \in \mathbb{N}}$) to denote may-paths (must-paths). A may-subpath (must-subpath) is a finite prefix of a may-path (must-path).*

[4] The reachability problem seeks for reachability of a set of states S, from the initial states of an analog or mixed analog/discrete system. Reachability allows the modelling of *safety properties* and can easily expressed using (any extension) of the CTL temporal logic language [3, 16].
[5] Mixed analog/digital systems
[6] For example, we could be interested in formulæ involving exponentials, that can be expressed only in the theory of reals endowed with exponentiation, whose decidability is still an open problem.

We are now ready to define the three-valued abstract semantics of CTL_f formulæ.

Definition 6 (CTL_f **Abstract Semantics).** *Let* $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ *be a* BMA *on* $X = \{x_1, \ldots, x_n\}$. *Given* $b \in \mathcal{B}$ *and a* CTL_f *formula* ϕ, *we define* $[b \models_3 \phi]$ *to be one of the three values in* $\{\mathsf{tt}, \mathsf{ff}, \bot\}$, *according to the following rules.*

Basic Formulæ:

– *If* $\phi = f(x_{i_1}, \ldots, x_{i_m}) \rhd I$, *then* $[b \models_3 \phi]$ *is defined as:*

$$\begin{cases} \mathsf{tt}, \textit{ if } [f](b) \subseteq I; \\ \mathsf{ff}, \textit{ if } [f](b) \cap I = \emptyset; \\ \bot, otherwise. \end{cases}$$

Boolean Operators:

– *If* $\phi = \neg\phi_1$, *then* $[b \models_3 \phi]$ *is defined as:*

$$\begin{cases} \mathsf{tt} \textit{ if } [b \models_3 \phi_1] = \mathsf{ff}; \\ \mathsf{ff} \textit{ if } [b \models_3 \phi_1] = \mathsf{tt}; \\ \bot \, otherwise. \end{cases}$$

– *If* $\phi = \phi_1 \vee \phi_2$, *then* $[b \models_3 \phi]$ *is defined as:*

$$\begin{cases} \mathsf{tt} \textit{ if } [b \models_3 \phi_1] = \mathsf{tt} \vee [b \models_3 \phi_2] = \mathsf{tt}; \\ \mathsf{ff} \textit{ if } [b \models_3 \phi_1] = \mathsf{ff} \wedge [b \models_3 \phi_2] = \mathsf{ff}; \\ \bot \, otherwise. \end{cases}$$

Temporal Operators & Path Quantifiers:

– *If* $\phi = \mathsf{E}\phi_1\mathsf{U}\phi_2$, *then* $[b \models_3 \phi]$ *is defined as:*

$$\begin{cases} \mathsf{tt} \textit{ if there exists a must-subpath } \{b_i\}_{i=0..k}^{must} \textit{ departing from } b \textit{ such that:} \\ \quad ([b_k \models_3 \phi_2] = \mathsf{tt}) \textit{ and } \forall 0 \leq i \leq k ([b_i \models_3 \phi_1] = \mathsf{tt}); \\ \mathsf{ff} \textit{ if for all may-path } \{b_i\}_{i \in \mathbb{N}}^{may} \textit{ departing from } b \textit{ in } \mathcal{A} \textit{ and for all } i \geq 0 \textit{ it holds that:} \\ \quad ([b_i \models_3 \phi_2] \neq \mathsf{ff}) \rightarrow \exists j < i ([b_j \models_3 \phi_1] = \mathsf{ff}) \\ \bot \textit{ otherwise}. \end{cases}$$

– *If* $\phi = \mathsf{A}\phi_1\mathsf{U}\phi_2$, *then* $[b \models_3 \phi]$ *is defined as:*

$$\begin{cases} \mathsf{tt} \textit{ if for all may-path } \{b_i\}_{i \in \mathbb{N}}^{may} \textit{ departing from } b \textit{ it holds that:} \\ \quad \exists i \geq 0 ([b_i \models_3 \phi_2] = \mathsf{tt}) \textit{ and } \forall 0 \leq j \leq i ([b_i \models_3 \phi_1] = \mathsf{tt}); \\ \mathsf{ff} \textit{ if there exists a must-subpath } \{b_i\}_{i=1..k}^{must} \textit{ departing from } b \textit{ in } \mathcal{A} \textit{ such that:} \\ \quad ([b_k \models_3 \phi_1] = \mathsf{ff}) \textit{ and } \forall j < i ([b_j \models_3 \phi_2] = \mathsf{ff}) \\ \bot \textit{ otherwise}. \end{cases}$$

4 Relating Abstract and Concrete CTL_f Semantics: Preservation Results

An immediate motivation to the choice of defining a *three*-valued abstract semantics for our CTL_f logic is the naturalness of having discrete abstract states where it is not possible to determine a definitive true or false value for basic CTL_f formulæ.

In this section, we show that the above choice has a further fundamental advantage. In fact, it allows to apply 3-valued model checking techniques—rather than classical (2-valued) model checking—for the automated reasoning on analog circuits. In the context of digital systems, three-valued abstractions

and model checking [1] have the advantage of preserving *both truth and falsity* of totally branching temporal logic from so called partial Kripke structures (playing the rule of abstract systems) to complete Kripke structures (playing the role of concrete digital systems). On the contrary, classical model checking allows only the preservation of formulæ containing exclusively universal path quantifiers and evaluating to the truth over the abstraction. We prove that the same gain is obtained in the context of analog systems abstraction/verification, when using three-valued logics.

In order to formalize the above ideas, we introduce the notion of *preserving* box modal abstractions for a given analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$. The conditions ensuring a box modal abstraction to be preserving for $\mathcal{C}$ naturally extends the notion of *completeness preorder* in [1], to the case in which the concrete system evolves according to continuous trajectories (rather than following discrete paths). Let $\prec$ to be the partial relation on $\{\mathsf{tt}, \mathsf{ff}, \bot\}$ such that $\bot \prec \mathsf{tt}, \bot \prec \mathsf{ff} \wedge \forall a \in \{\mathsf{tt}, \mathsf{ff}, \bot\}(a \prec a)$.

Definition 7 (Preserving Box Modal Abstraction). *Consider an analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ and let $n = |X^D \cup X^I|$. A box modal abstraction $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ on $X = X^D \cup X^I$ is said* preserving *for $\mathcal{C}$ iff there exists a relation $\lessdot \subseteq \mathcal{B} \times \mathbb{R}^n$ such that, whenever $b \lessdot v$, the following rules hold:*

1. *For each basic CTL_f formula ϕ, $[b \models_3 \gamma] \prec [v \models \gamma]$.*
2. *If $\{b_i\}_{1 \leq i \leq k}^{must}$ is a must-subpath departing from b in $\mathcal{A}$, then $\mathcal{C}$ admits a trajectory ρ departing from v such that there exists an increasing sequence of real values $\{t_i\}_{0 \leq i \leq k}$ such that:*

$$\forall t \in \mathbb{R}(t_{i-1} \leq t \leq t_i \rightarrow b_i \lessdot \rho(t))$$

3. *If $\mathcal{C}$ admits a trajectory $\rho(t)$ departing from v, then $\mathcal{A}$ admits a may-path $\{b_i\}_{i \in \mathbb{N}}^{may}$ such that there exists an increasing sequence of real values $\{t_i\}_{i \in \mathbb{N}}$ such that:*

$$\forall t \in \mathbb{R}(t_i \leq t \leq t_{i+1} \rightarrow b_i \lessdot \rho(t))$$

Given a preserving box modal abstraction $\mathcal{A}$ of $\mathcal{C}$, the preserving concretization *for $\mathcal{A}$ and $\mathcal{C}$ is the maximum relation satisfying the three above conditions.*

Let $\mathcal{A}$ be a preserving box modal abstractions of the concrete system $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$. Assume that the state v in $\mathcal{C}$ is related to the box b via the corresponding preserving concretization relation. The following theorem ensures that if the abstract semantics of the CTL_f formula ϕ evaluates to true (false) on the box b, then also the concrete semantics of ϕ evaluates to true (false) against v. Hence, we can trust preservation of both truth and falsity of any CTL_f property from abstract to concrete systems.

Theorem 1. *Let $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ to be an analog model and let $n = |X^D \cup X^I|$. If $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ is a preserving box modal abstraction for $\mathcal{C}$, then:*

$$\forall b \in \mathcal{B}, \forall v \in \mathbb{R}^n, \forall \phi \in \mathsf{CTL_f}((b \lessdot v) \rightarrow ([b \models_3 \phi] \prec [v \models \phi]))$$

where $\lessdot \subseteq \mathcal{B} \times \mathbb{R}^n$ denotes the preserving concretization relation for $\mathcal{A}$ and $\mathcal{C}$.

Proof. Let $v \in \mathbb{R}^n, b \in \mathcal{B}$ and assume $b \lessdot v$. We proceed by structural induction on ϕ. If ϕ is a basic formula of CTL_f, then the claim follows directly from item 1 in the definition of $\lessdot$. If ϕ is obtained by any boolean composition operator, then the claim follows from inductive hypothesis and from monotonicity of boolean operators with respect to the partial order $\prec$ on $\{\mathsf{tt}, \mathsf{ff}, \bot\}$. The remaining interesting cases where $\phi = \mathsf{A}\phi_1\mathsf{U}\phi_2 | \mathsf{E}\phi_1\mathsf{U}\phi_2$ can be proved with symmetrical arguments. Hence we report here only a complete proof for the case in which $\phi = \mathsf{A}\phi_1\mathsf{U}\phi_2$.

– Let $\phi = \mathsf{A}\phi_1\mathsf{U}\phi_2$ and assume $[b\models_3\phi] = \mathsf{tt}$ and $[v \models \phi] = \mathsf{ff}$. Then, there exists a trajectory departing from v such that $\forall t > 0(\rho(t) \models \phi_1 \rightarrow (\exists t' \leq t(\rho(t') \models \neg\phi_2)))$. By item 3 in Definition 7, $\mathcal{A}$ admits a may path $\{b_i\}_{i\in\mathbb{N}}^{may}$ departing from b such that there exists an increasing sequence of real values $\{t_i\}_{i\in\mathbb{N}}$ for which it holds the assertion: $\forall t \in \mathbb{R}(t_i \leq t \leq t_{i+1} \rightarrow b_i \lessdot \rho(t))$. By inductive hypothesis we obtain that $\forall t \in \mathbb{R}(t_i \leq t \leq t_{i+1}([b_i\models_3\phi_1] \prec [\rho(t) \models \phi_1] \wedge [b_i\models_3\phi_2] \prec [\rho(t) \models \phi_2])$. By Definition 6, this contradicts our hypothesis $[b\models_3\phi] = \mathsf{tt}$.
To complete the proof, we should find a contradiction to the assumption $[b\models_3\phi] = \mathsf{ff}$ and $[v \models \phi] = \mathsf{tt}$. In the above case $\mathcal{A}$ admits a must-subpath $\{b_i\}_{i=1\ldots k}^{must}$ departing from b such that it holds $([b_k\models_3\phi_1] = \mathsf{ff})$ and $\forall j < i([b_j\models_3\phi_2] = \mathsf{ff})$. By item 2 in Definition 7 such a path induces a trajectory departing from v in $\mathcal{C}$ witnessing that $[v \models \phi] = \mathsf{ff}$ and contradicting our hypothesis.

□

5. CTL_f Model Checking

Let $\mathcal{C}$ be an analog model. The aim of this section is to show how to use interval arithmetic to define a preserving box modal abstraction for $\mathcal{C}$. Given a CTL_f formula ϕ, it is then possible to compute its abstract semantics over $\mathcal{A}$. Finally, (underapproximations of) sets of states in $\mathcal{C}$ against which ϕ evaluates to true/false can be obtained via the concretization relation for $\mathcal{C}$ and $\mathcal{A}$.

In detail, Figure 2 depicts a procedure to extract a preserving box modal abstraction from an analog model[7] $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$. The procedure BUILDBMA is composed by three macro steps, one for each component in the box modal abstraction $\mathcal{A} = \{\mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\}$, which is computed as output. In particular, the first macrostep uses an input parameter $k \in \mathbb{N}$, to perform a uniform subdivision in each component of the box of interest b_0 (also given as input). The remaining space in $\mathbb{R}^{|X|} \setminus b_0$ is covered by an arbitrary decomposition into (possibly unbounded) boxes. The second macro step of our procedure uses simple interval arithmetic techniques to define the set of may-transitions in the abstract model. Finally, in the third macro step, some of the previously defined may transitions are considered to be colored also as must transition. The following theorem proves that the box modal abstraction generated by this algorithm is CTL_f preserving for the input analog system model $\mathcal{C}$.

Theorem 2. *Given $k \in \mathbb{N}$ and an analog system model $\mathcal{C}$, the procedure* BUILDBMA$(\mathcal{C}, k)$ *computes a CTL_f preserving box modal abstraction for $\mathcal{C}$.*

Proof. Let $\mathcal{A} = \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle$ be the BMA computed by $\mathbf{BuildBMA}(\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle, k)$, where we denote $n = |X^D \cup X^I|$. We get to the thesis by proving that the relation over $\mathcal{B} \times \mathbb{R}^n$ such that $b \lessdot v$ if $v \in b$ is a concretization relation for $\mathcal{A}$ and $\mathcal{C}$, i.e. it respects the three conditions in Definition 7. For the first condition, let $\phi = f(x_{i_1}, \ldots, x_{i_m}) \rhd I$ be a basic CTL_f formula, and assume we have a box b and $v \in \mathbb{R}$ such that $b \lessdot v$. If $[b\models_3\phi] = \bot$, then $[v \models \phi]$ can assume any value to satisfy $[b\models_3\phi] \not\prec [v \models \phi]$. If $[b\models_3\phi] = \mathsf{tt}$, then $f(v) \in [f](b) \subseteq I$ which leads to $[v \models \phi] = \mathsf{tt}$ i.e. $[b\models_3\phi] \not\prec [v \models \phi]$. A similar argument holds in case $[b\models_3\phi] = \mathsf{ff}$.

In procedure BUILDBMA$(\mathcal{C}, k)$, the conditions allowing to connect two neighbour boxes with a may transition, are *necessary* conditions to have a trajectory traversing the two boxes. Similarly, the conditions allowing to establish a self may-loop on the box b, are necessary conditions to have a trajectory indefinetely evolving in b. Symmetrically, the condition allowing to define a must-transition connecting the boxes b_1, b_2 , are sufficient conditions ensuring that each trajectory traversing b_1 evolves towards b_2. The validity of item 2 and item 3 in Definition 7, is established on the ground of the above observations. □

[7] Here, we assume to deal with a input model such that input variables maintain a constant (arbitrary) value overall the execution of the system. However, a symmetrical extreme parameter model in which input variables are admitted to assume any value while the system evolves, could be also easily considered for the abstraction generation. Note that the same extreme input models were adopted in [8, 9].

BuildBMA$(\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle, b_0, k)$

Input: *a concrete analog model* $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$*, a box of interest* $b_0 \subseteq \mathbb{R}^{|X^D \cup X^I|}$, $k \in \mathbb{N}$ *defining the size of the uniform decomposition on* b_0
Output: *A* CTL_f *preserving box modal abstraction* $\mathcal{A} = \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle$ *for* $\mathcal{C}$

% *Step 1. Definition of the collection of boxes* $\mathcal{B}$ *in the box modal abstraction for* $\mathcal{C}$
(1) Let $\mathcal{B}_1$ be the set of boxes obtained by performing an uniform k-decomposition of b_0
(2) Let $\mathcal{B}_2$ be the set of (unbounded) boxes obtained by performing an arbitrary box decomposition in $\mathbb{R}^{|X^D \cup X^I|} \setminus b_0$
(3) $\mathcal{B} \leftarrow \mathcal{B}_1 \cup \mathcal{B}_2$

% *Step 2. Definition of the may transitions in the box modal abstraction of* $\mathcal{C}$
(4) $\overset{may}{\rightarrow} := \{\}$
(5) **for each** (pair p of neighbouring boxes in $\mathcal{B}$ having F as a common face) **do**
(6) Let i such that ($p = \langle b_1, b_2\rangle \wedge F$ is the i lower face of $b_1 \wedge F$ is the i upper face of b_2)
(7) **if** $(i \leq |X^D| \wedge [\dot{x}_i](F) \cap [0, +\infty] \neq \emptyset)$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b_1, b_2)\}$
(8) **if** $(i \leq |X^D| \wedge [\dot{x}_i](F) \cap [-\infty, 0] \neq \emptyset)$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b_2, b_1)\}$
(9) **for each** $(b \in \mathcal{B})$ **do**
(10) **if** $(\forall 1 \leq i \leq |X^D| (0 \in [\dot{x}_i](b)))$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b, b)\}$

% *Step 3. Definition of the must transitions in the box modal abstraction of* $\mathcal{C}$
(11) $\overset{must}{\rightarrow} := \{\}$
(11) **for each** $(b \in \mathcal{B})$ **do**
(12) **if** ($b \overset{may}{\rightarrow} b'$ is the only may transition sourcing from b) **then** $\overset{must}{\rightarrow} := \overset{must}{\rightarrow} \cup \{(b, b')\}$

Fig. 2. Computing a CTL_f Preserving Box Modal Abstraction.

LabelCTL$_f(\phi, \mathcal{A} = \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$

Input: *a* CTL_f *formula* ϕ *and a box modal abstraction* $\mathcal{A} = \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle$
Output: $\forall b \in \mathcal{B}$, $\forall$ *subformula* ϕ' *of* ϕ, b *is labelled with* $\langle \phi', [b \models_3 \phi']\rangle$

case ϕ **of**
$f(x_{i_1}, \ldots, x_{i_m}) \rhd I$: **for each** $b \in \mathcal{B}$ **do**
if $[f](b_{|_{i_1 \ldots i_m}}) \subseteq I$
then $L(b) \leftarrow L(b) \cup \langle \phi, \mathtt{tt}\rangle$
else if $[f](b_{|_{i_1 \ldots i_m}}) \cap I = \emptyset$
then $L(b) \leftarrow L(b) \cup \langle \phi, \mathtt{ff}\rangle$
else $L(b) \leftarrow L(b) \cup \langle \phi, \bot\rangle$

$\neg\phi_1$: **for each** $b \in \mathcal{B}$ **do**
LabelCTL$_f(\phi_1, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$;
ChecNot$(\langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle, \phi_1)$

$\phi_1 \wedge \phi_2$: **for each** $b \in \mathcal{B}$ **do**
LabelCTL$_f(\phi_1, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$; **LabelCTL**$_f(\phi_2, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$
ChecAnd$(\langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle, \phi_1, \phi_2)$

$A\phi_1 U \phi_2$: **for each** $b \in \mathcal{B}$ **do**
LabelCTL$_f(\phi_1, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$; **LabelCTL**$_f(\phi_2, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$
CheckAU$(\langle \mathcal{B}, I_{\mathcal{B}}, \rightarrow\rangle, \phi_1, \phi_2)$

$E\phi_1 U \phi_2$: **for each** $b \in \mathcal{B}$ **do**
LabelCTL$_f(\phi_1, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$; **LabelCTL**$_f(\phi_2, \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle)$
CheckEU$(\langle \mathcal{B}, I_{\mathcal{B}}, \rightarrow\rangle, \phi_1, \phi_2)$

Fig. 3. CTL_f Model Checking Algorithm

Assume that a box modal abstraction $\mathcal{A}$ has been generated from a concrete system $\mathcal{C}$ using the algorithm BUILDBMA$(\mathcal{C}, k)$, and let ϕ be a CTL_f formula. The problem of checking, for each box $b \in \mathcal{B}$, whether $b \models_3 \phi$ can be easily solved on the base of the three-valued classic CTL model checking for digital systems [1]. More precisely, such a problem can be solved using the CTL_f model checking algorithm reported in Figure 3 in which:

- According to Definition 6, interval arithmetic is used to label each box $b \in \mathcal{B}$ with the set of pairs:

$$\{\langle \gamma_i, [b \models_3 \gamma_i] \rangle \mid \gamma_i \text{ is a subformula of } \phi\}$$

- Each composed CTL_f formula, is manipulated using the corresponding subprocedure for three-valued classic CTL model checking on digital systems.

Of course, in case of a $\perp$ result, the BMA needs to be refined; An interesting problem, to this purpose, is that of using the information about $\perp$ evaluations of subformulæ to drive the refinement of the BMA. Also, constraint propagation techniques, as the ones proposed in [15], could be taken in consideration within the process of BMA refinement.

6 Conclusions and Future Work

We propose CTL_f, as a new three-valued temporal logic for the specification of analog properties. We deal with the problem of (multi-valued) model checking of the corresponding formulæ on analog systems abstractions, as well as on the quest about *preservation* of properties from (discrete) abstractions to (analog) concrete systems. We foresee a reasonable way to apply the ideas developed here, by combining them with symbolic analog systems simplification techniques. In particular, our approach could be used to automatically verify properties of interest on simplified analog models.

References

1. G. Bruns and P. Godefroid. Model checking partial state spaces with 3-valued temporal logics. In *Computer Aided Verification*, pages 274–287, 1999.
2. E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using brancing time temporal logic. In *Proc. of Workshop on Logic of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1982.
3. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 1999.
4. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Trans. Program. Lang. Syst.*, 19(2):253–291, 1997.
5. Patrice Godefroid, Michael Huth, and Radha Jagadeesan. Abstraction-based model checking using modal transition systems. *Lecture Notes in Computer Science*, 2154:426+, 2001.
6. P. R. Gray and R. G. Meyer. *Analysis and Design of Analog Integrated Circuits (3rd Edition)*. John Wiley & Sons Inc., 1993.
7. Thomas Halfmann and Tim Wichmann. Overview of symbolic methods in industrial analog circuit design. Technical Report 44, Fraunhofer ITWM, Kaiserslautern, Germany, 2003.
8. W. Hartong, L. Hedrich, and E. Barke. Model checking algorithms for analog verification. In *Proc. of the 39th conference on Design automation (DAC'02)*, pages 542–547, New York, NY, USA, 2002. ACM Press.
9. W. Hartong, L. Hedrich, and E. Barke. On discrete modeling and model checking for nonlinear analog systems. In *Proc. of the 14th International Conference on Computer Aided Verification (CAV'02)*, pages 401–413, London, UK, 2002. Springer-Verlag.
10. E. Hennig. *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits*. Shaker Verlag, 2000.
11. T.A. Henzinger. The theory of hybrid automata. In M.K. Inan and R.P. Kurshan, editors, *Verification of Digital and Hybrid Systems*, NATO ASI Series F: Computer and Systems Sciences 170, pages 265–292. Springer-Verlag, 2000.
12. T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373–382. ACM Press, 1995.
13. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer Verlag, 2001.
14. C. Myers. Formal verification of analog and mixed signal circuits. In *In Proc. 2nd Annual Utah Verification Workshop*, 2005.
15. S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In *HSCC*, pages 573–589, 2005.
16. K. Schneider. *Verification of Reactive Systems*. Springer Verlag, 2004.
17. R. Sommer, E. Hennig, M. Thole, T. Halfmann, and T. Wichmann. Analog insydes 2 - new features and applications in circuit design. In *Proc. 6th International Workshop on Symbolic Methods and Applications in Circuit Design (SMACD 2000)*, 2000.
18. R. Sommer, M. Thole, and E. Hennig. A generic circuit modelling strategy combining symbolic and numerical analisys. In *Proc. of the 5th International Workshop on on Symbolic Methods and Applications to Circuit Design (SMCAD'98)*, 1998.