Abstract— Hybrid systems modelers have become the cornerstone of embedded system development, with Simulink a de facto standard and Modelica a new player. Such tools still raise a number of issues that, we believe, require more fundamental understanding. In this paper we propose using non standard analysis as a semantic domain for hybrid systems — non standard analysis is an extension of classical analysis in which infinitesimals (the $\varepsilon$ and $\eta$ in the celebrated generic sentence $\forall \varepsilon \exists \eta \ldots$ in college maths) can be manipulated as first class citizens. This allows us to provide a denotational semantics and a constructive semantics for hybrid systems, thus establishing simulation engines on a firm mathematical basis. In passing, we cleanly separate the job of the numerical analyst (solving differential equations) from that of the computer scientist (generating execution schemes).

I. INTRODUCTION

Hybrid systems modelers have become in the last two decades the cornerstone of complex embedded system development, for computer controlled systems. Simulink\(^2\) has become the de facto standard for physical system modeling and simulation. Noticeably, by building on top of the success of Simulink, The Mathworks was able to take over the market of simulation. Noticeably, by building on top of the success of Simulink, The Mathworks was able to take over the market of simulation. Simulink, The Mathworks was able to take over the market of simulation.

This work was supported by the INRIA “Action d’envergure” SYNCHRONICS.

1This work was supported by the INRIA “Action d’envergure” SYNCHRONICS.

2http://www.mathworks.com/products/simulink/

3http://www-rocq.inria.fr/scicos/

4http://www.modelica.org/

Since simulations use a single, global, solver, the choice and tuning of the integration method is global to the system. This may cause undesirable interactions between sub-systems that seemingly should not interact [2].

Zero-crossings, which trigger mode changes, can involve a combination of complex operations whose scheduling may be delicate.

II. BACKGROUND ON NON-STANDARD ANALYSIS

The key difficulty in reconciling requirements (a)–(c) is to give proper semantics to the inherently continuous time statement: $\dot{y} = x$. In non-standard analysis, this statement means, by definition of the derivative of a function: $\forall \theta \simeq 0, \forall t \in \mathbb{R}_+, \frac{1}{\theta} (y_{t+\theta} - y_t) \simeq x_t$, where expression “$\simeq 0$” is a non-standard expression that reads: “$\theta$ is infinitesimal”.

Non-standard analysis has been proposed by Abraham Robinson in the 60’s to allow handling explicitly “infinitesimals” in analysis [1], [16]. Robinson’s approach is axiomatic, in that he proposes enriching the basic ZFC (Zermelo-Fraenkel) framework with three more axioms. There has been much debate in the community of mathematicians as to
whether it is worth considering non-standard analysis instead of sticking with the traditional one. We won’t enter this
debate. One important thing for us, however, is that it allows
using non-standard discretization of continuous dynamics "as
if" it was operational. To our surprise, such an idea is indeed
not new. Bludzne and Kroh [9], [8] have used non-standard
analysis as a mathematical support for defining a system
theory for hybrid systems. The formalization they propose
closely mimics that of Turing machines. The introduction to
non-standard analysis in [8] is very pleasant and we take the
liberty to borrow it. This presentation was originally due to
Lindström, see [15]. Its interest is that it does not require any
fancy axiomatic material but only makes use of the axiom of
choice — actually a weaker form of it.

The goal is to augment $\mathbb{R} \cup \{\pm \infty\}$ by adding, to each $x$ in
this set, a bunch of elements that are "infinitesimally close"
to it, call $^*\mathbb{R}$ the resulting set. Another requirement is that all
operations and relations defined on $^*\mathbb{R}$ should extend to $^*\mathbb{R}$. A
first idea is to represent such additional numbers as convergent
sequences of reals. For example, elements infinitesimally
close to the real number zero are the sequences $u_n = 1/n$,
$v_n = 1/\sqrt{n}$ and $w_n = 1/n^2$. Observe that the above three
sequences can be ordered: $v_n > u_n > w_n > 0$. How can this
be made systematic? We explain it next.

A. Ultrafilters, ultraproducts, and the Transfer Principle

For an arbitrary set, a filter $\mathcal{F}$ over $I$ is a family of subsets
of $I$ such that:
1) the empty set does not belong to $\mathcal{F}$,
2) $P,Q \in \mathcal{F}$ implies $P \cap Q \in \mathcal{F}$, and
3) $P \in \mathcal{F}$ and $P \subseteq Q \subseteq I$ implies $Q \in \mathcal{F}$.

Consequently, $\mathcal{F}$ cannot contain both a set $P$ and its com-
plement $P^c$. A filter that contains at least one of the two for
any subset $P \subseteq I$ is called an ultra-filter. At this point we
recall Zorn’s lemma, known to be equivalent to the axiom of
choice:

Lemma 1 (Zorn’s lemma): Any partially ordered set $(X, \leq)$
such that any chain in $X$ possesses an upper bound
has a maximal element.

It is easily seen that a filter $\mathcal{F}$ over $I$ is an ultra-filter if
and only if it is maximal with respect to set inclusion. By
Zorn’s lemma, any filter $\mathcal{F}$ over $I$ can be extended to an
ultra-filter over $I$. Now, if $I$ is infinite, the family of sets $\mathcal{F} = \{P \subseteq I \mid P^c \text{ is finite}\}$ is a free
filter, meaning it contains no finite set.

It can thus be extended to a free ultra-filter over $I$:

Lemma 2: Any infinite set has a free ultra-filter.

Every free ultra-filter $\mathcal{F}$ over $I$ uniquely defines, by setting
$\mu(P) = 1$ if $P \in \mathcal{F}$ then 1 else 0, a finitely additive measure
$\mu : 2^I \to [0,1]$ such that

$\mu(I) = 1$ and $\mu(P) = 0$ whenever $P$ is finite.

Indeed, the proposed construction bears some resemblance with the
construction of $\mathbb{R}$ as the set of equivalence classes of Cauchy sequences in $\mathbb{Q}$
modulo the equivalence relation $(u_n) \equiv (v_n)$ iff $\lim_{n \to \infty} (u_n - v_n) = 0$.

Observe that, as a consequence, $\mu$ cannot be sigma-additive (in contrast
to probability measures or Radon measures) in that it is not true that $\mu(\bigcup A_n) = \sum_n \mu(A_n)$ holds for an infinite denumerable sequence $A_n$
of pairwise disjoint subsets of $\mathbb{N}$.

Now, fix an infinite set $I$ and a finitely additive measure $\mu$ over
$I$ as above. Let $X$ be a set and consider the Cartesian product
$X^I = \{x_i\}_{i \in I}$. Say $(x_i) \sim (x'_i)$ iff $\mu(\{i \in I \mid x_i \neq x'_i\}) = 0$.
Relation $\sim$ is an equivalence relation whose equivalence
classes are denoted by $[x_i]$ and we define

$^*X = X^I / \sim$ (1)

$X$ is naturally embedded into $^*X$ by mapping every $x \in X$ to
the constant tuple such that $x_i = x$ for every $i \in I$.

Any algebraic structure over $X$ (group, ring, field) carries over to $^*X$ by almost pointwise extension. In particular, if $[x_i] \neq 0$, meaning that $\mu(\{i \mid x_i = 0\}) = 0$ we can define
its inverse $[x_i]^{-1}$ by taking $y_i = x_i^{-1}$ if $x_i \neq 0$ and $y_i = 0$
otherwise. This construction yields $\mu(\{i \mid y_i x_i = 1\}) = 1$, whence $[y_i][x_i] = 1$ in $^*X$. The existence of an inverse for any
non-zero element of a ring is indeed stated by the following first
order formula: $\forall x(x = 0 \lor \exists y(xy = 1))$. More generally:

Lemma 3 (Transfer Principle): Every first order formula is
true over $^*X$ iff it is true over $X$.

B. The sets $^*\mathbb{R}$ and $^*\mathbb{N}$ of non-standard reals and integers

We just apply the above general construction to $X = \mathbb{R}$ and
$I = \mathbb{N}$ and we denote by $^*\mathbb{R}$ the result, which is then a field
according to the transfer principle. By the same principle, $^*\mathbb{N}$
is totally ordered by $\{u_n\} \leq \{v_n\}$ if $\mu(\{n \mid v_n > u_n\}) = 0$.

For an arbitrary algebraic structure of real numbers, let $\lim(u) \subseteq \mathbb{R} =_{def} \mathbb{R} \cup \{\infty, -\infty, +\infty\}$ denote the (possibly empty) set of
all limit points of sequence $u$: $x \in \lim(u)$, let $v_k = u_{n_k}$
be a subsequence of $u$ converging to $x$. If $\lim(u) \neq \emptyset$, there
exists exactly one limit point $x \in \lim(u)$ such that $\mu(\{n_k\}) = 1$, and any other limit point yields a $\mu$-measure $0$ for the corresponding
subsequence. Call $x$ the standard part of $\{x_n\}$
and we write $x = st(\{x_n\})$. Infinite $x \in ^*\mathbb{R}$ have no standard
part in $\mathbb{R}$. It is also of interest to apply the general construction
for non-standard integers. $^*\mathbb{N}$ differs from $\mathbb{N}$ by the addition of infinite integers,
which are equivalence classes of sequences of integers whose
esential limit is $+\infty$.

C. Integrals and differential equations

Any sequence $(g_n)$ of functions $g_n : \mathbb{R} \to \mathbb{R}$ pointwise
defines a function $^*[g_n] : ^*\mathbb{R} \to ^*\mathbb{R}$ by setting

$^*[g_n](x_n) = [g_n(x_n)]$

A function $^*g : ^*\mathbb{R} \to ^*\mathbb{R}$ which can be obtained in this way is
called internal. Properties of and operations on ordinary functions
extend pointwise to internal functions of $^*\mathbb{R} \to ^*\mathbb{R}$. For
g : $\mathbb{R} \to \mathbb{R}$, its non-standard version is the internal function
$^*g = [g,g,g,\ldots]$. The same notions apply to sets. An internal
set $A = [A_n]$ is called hyperfinite if $\mu(\{n \mid A_n \text{ finite}\}) = 1$; the
cardinal $|A|$ of $A$ is defined as $[|A_n|]$.

$^*\mathbb{R}$ is an extension of $\mathbb{R}$ in which a certain amount of
non-standard analysis can be used. For instance, one can define
the non-standard derivative of a function $f : \mathbb{R} \to \mathbb{R}$ as the
function $D^\text{non-standard}(f) : ^*\mathbb{R} \to ^*\mathbb{R}$ by setting

$D^\text{non-standard}(f)(x_n) = \lim(x_{n} \to x) f(x_n)$

This function is analogous to the standard derivative, but it has
the advantage of being defined for every function $f$.
Now, consider an infinite number $N \in {}^*\mathbb{N}$ and the set

$$T = \{0, 1\}$$
— hence \( u \) is discrete — and, for every \( t \in \tau \), \( u_t = z_t \) holds, and \( u_t = u_0 \) for \( t < t_1 \), the first instant of \( \tau \). Note that we do not require that \( z \) is discrete.

Eq. 6: For \( y, x \) two signals, \( y_0 \) a value, and \( u \) a discrete signal, Eq. 6 states that ODE \( \dot{y}_t = x_t \) holds with initial condition \( y_0 \) and this ODE is reset to the value given by \( u \) at each instant of the discrete clock of \( u \).

As said before, hybrid systems are specified in SimpleHybrid via sets of equations of the form Eq. 1–Eq. 6, taken conjunctively. For example, composing ODE Eq. 6 with statement \( x = f(y, v) \) of the form Eq. 1, and reset Eq. 5, yields the ODE

\[
\dot{y} = f(y, v) \text { init } y_0 \text { reset } [z] \text { every } [\tau] \quad (11)
\]

which means that ODE \( \dot{y}_t = f(y_t, v_t) \) holds with initial condition \( y_0 \) and this ODE is reset to a value given by \( z_t \) each time zero-crossing \( \tau_t \) occurs. It is easily checked that generic form (11) for a SimpleHybrid equation is closed under parallel composition, and that SimpleHybrid allows to encode hybrid automata with their locations [2].

IV. A SEMANTICS OF SimpleHybrid

Throughout this section we fix a basic infinitesimal base step \( \partial \). Following [9], as our universal time base we replace \( \mathbb{R}^+ \) by the non-standard set

\[
\mathbb{T} = \{ t_n = n\partial \mid n \in \star \mathbb{N} \}
\]

For \( t \in \mathbb{T} \), define

\[
^*t = \max \{ s \mid s \in \mathbb{T}, s < t \}
\]

\[
t^* = \min \{ s \mid s \in \mathbb{T}, s > t \}
\]

(12)

We thus have \( ^*t_n = t_{n-1} \) and \( t^*_n = t_{n+1} \). The key fact about \( \mathbb{T} \) is that for every \( u \in \mathbb{R}^+ \) there exists a unique \( t \in \mathbb{T} \) such that \( ^*t < u \leq t \) and \( t - u \) is infinitesimal. Thus \( \mathbb{T} \) is, at the same time, dense in \( \mathbb{R}^+ \), and can still be handled as if it was discrete and totally ordered.

A. Non-standard semantics

We assume an underlying set \( \mathcal{T} \) of clock variables. Elements and subsets of \( \mathcal{T} \) are generically denoted by \( \tau \) and \( T \), respectively. We identify \( \tau \) with the boolean predicate it defines, see (10). We assume an underlying set \( \mathcal{X} \) of variables and, a domain \( D_x \) for every \( x \in \mathcal{X} \), finite, a state over \( \mathcal{X} \) is an element \( s \in D_{\mathcal{X}} \), where \( D_{\mathcal{X}} = \prod_{x \in \mathcal{X}} D_x \) and a behaviour over \( \mathcal{X} \) is an element \( \sigma \in \mathbb{T} \rightarrow D_{\mathcal{X}} \). We write \( x_t \) instead of \( \sigma \rightarrow \sigma(t, x) \) and \( \tau_t \) instead of \( \sigma \rightarrow \sigma(t, \tau) \).

A hybrid system is a tuple \( S = (X, T, \Sigma) \), where \( X \subseteq \mathcal{X} \) and \( T \subseteq \mathcal{T} \) are finite and \( \Sigma \) is a set of behaviours over \( X \cup T \). For \( Y \supseteq X \cup T \), we can lift \( \Sigma \) to \( Y \), written \( \Sigma^Y \), by taking all behaviours over \( Y \) whose projection over \( X \cup T \) are in \( \Sigma \). Then, for \( S_i = (X_i, T_i, \Sigma_i), i = 1, 2 \), we define the parallel composition

\[
S_1 \parallel S_2 = (X, T, \Sigma_1 \uparrow_{X \cup T} \Sigma_2)
\]
be abstracted away from the constructive semantics. As an example, the constructive semantics for statement 2 writes as "\( \cdot x \triangleright y \), and the constructive semantics for Eq. writes as "\( \text{on } \tau_u \text{ then } u \triangleright y \text{ else } \cdot x \triangleright y \)". Using the above notations, the constructive semantics is given in Table I, last column.

Avoiding causality circuits: Using the above abstraction, for each given clock configuration of \( S \), the transitive closure of relation \( \triangleright \) is a pre-order on \( X \) — by abuse of notation, we call it also \( \triangleright \). If \( S \) is such that \( \triangleright \) is a partial order for any reachable clock configuration (see (14) and below), this means that no causality circuit occurs in \( S \) and the different variables can be evaluated according to any order compatible with \( \triangleright \).

Since no clock occurs on any consequent part of a zero-time causality constraint, the only possible cause of circuits in relation \( \triangleright \) is via sets of statements of the form Eq. We thus formally justify here the rule that no delay-free, derivative-free, data flow circuit should exist in the considered program. If this condition is satisfied, topological sorting yields the due scheduling. We assume this condition to be in force in the remainder of this section.

**Single assignment condition**: Say that system \( S \) obeys the single assignment condition if no variable of \( S \) sits on the left hand side of two or more equations. The following holds:

**Lemma 4**: If \( S \) possesses no causality circuit and obeys the single assignment condition, then it is deterministic and partial order \( \triangleright \) at each clock configuration specifies all correct schedulings for the execution of \( S \).

### Table I

<table>
<thead>
<tr>
<th>Statement ( S )</th>
<th>Non-standard semantics of ( S )</th>
<th>([S]): constructive semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Eq_1 ): ( y = f(x, \dot{x}) )</td>
<td>( \forall t \in \mathbb{R}_+ \implies y_t = f(x_t, \dot{x}_t) )</td>
<td>( \text{on } \tau_y : x \triangleright y )</td>
</tr>
<tr>
<td>( Eq_2 ): ( y = \text{last}(x) )</td>
<td>( \forall t \in \mathbb{R}_+ \implies y_t = x^{\cdot t} )</td>
<td>( \text{on } \tau_y : \cdot x \triangleright y )</td>
</tr>
<tr>
<td>( Eq_3 ): ( \tau = \text{up}(z) )</td>
<td>( \tau_{\cdot t} = [z_{\cdot t} &lt; 0] \wedge [z_t \geq 0] )</td>
<td>( \text{on } \tau_y : z \triangleright \tau_{\cdot t} )</td>
</tr>
<tr>
<td>( Eq_4 ): ( y = \text{pre}(x) \text{ init } y_0 )</td>
<td>( \forall t &lt; \min(\tau_y) \implies y_t = y_0 ) ( \forall t \in \tau_y \implies y_t = x^{\cdot t} )</td>
<td>( \text{on } \tau_y : \tau_t \text{ discrete } ) ( \text{on } \tau_y : \cdot x \triangleright y )</td>
</tr>
<tr>
<td>( Eq_5 ): ( u = [z] \text{ every } [\tau] \text{ init } u_0 )</td>
<td>( \forall t &lt; \min(\bigcup_i \tau_{u_i}) \implies u_t = u_0 ) ( \forall t \in \tau_{u_i} \setminus \bigcup_{j&lt;i} \tau_{u_j} \implies u_t = z_{i,t} )</td>
<td>( \text{on } [\tau] : [z] \triangleright u )</td>
</tr>
<tr>
<td>( Eq_6 ): ( y = x \text{ init } y_0 \text{ reset } u )</td>
<td>( \forall t \notin \tau_u \implies y_t = y^{\cdot t} + \partial \cdot x^{\cdot t} ) ( \forall t \in \tau_u \implies y_t = u_t )</td>
<td>( \text{on } \tau_u \text{ then } u \triangleright y \text{ else } \cdot x \triangleright y )</td>
</tr>
<tr>
<td>( Eq_7 ): ( S_1 \parallel S_2 ) ( S_1 = (X_1, \Sigma_1) ) ( S_2 = (X_2, \Sigma_2) ) ( (X, \Sigma_1^{+X} \cap \Sigma_2^{+X}) ), ( X = X_1 \cup X_2 )</td>
<td>( [S_1] \parallel [S_2] )</td>
<td></td>
</tr>
</tbody>
</table>

Non-standard semantics (mid column) and constructive semantics (right column) of SIMPLEHYBRID.

## V. Related Work

Studies on hybrid systems modelers from a semantics point of view are not so numerous. We discuss the few we consider relevant for comparison. First of all, we recall the legacy work [3]. In fact, the agenda presented in that paper closely resembles the one we develop here. Except that, in [3] the tool of non-standard analysis was not used. Consequently, [3] suffers from some hand waving, as a careful reader can notice.

Perhaps the closest attempt similar to ours is the work of the Ptolemy group, by Ed Lee and Haiyang Zheng [13], [14], which studies the handling of discontinuities in hybrid systems modelers. Consider the system \( x = f(x, u) \) \( \text{reset } v \) \( \text{every up}(g(x) \leq 0) \). Then, in handling this resetting mechanism, the following landmark values for \( x \) must be considered: 1/ the first \( x \) where \( g(x) \leq 0 \) holds in the ODE; and 2/ the resetting value \( x^0 = h(v) \) at the same instant. From the mathematical viewpoint, the two values for \( x \) occur at the same time, but they are clearly causally ordered. Following the idea of tagged signals [12], this was solved in [13], [14] by tagging events with an extended time index taken from index set \( \mathbb{R}_+ \times \mathbb{N} \) equipped with the lexicographic order, and the above two values for \( x \) would get indexed as \( x_{t,0} \) and \( x_{t,1} = x^0 \), respectively. Tag set \( \mathbb{R}_+ \times \mathbb{N} \) is referred by the authors as the super-dense time. This type of multi-dimensional time set was considered earlier for discrete time systems models in the area of synchronous languages [5], [6]. Our approach avoids using super-dense time because non-standard index set \( \mathbb{T} \) is both discrete and dense. Existence of previous instant \( \cdot t \) and next instant \( t \) was
used in table I, replacing the multi-dimensional instants \((t, 0)\) and \((t, 1)\) of [13], [14]. On another aspect, the work [13], [14] is made complicated by issues of smoothness, Lipschitzness, existence and uniqueness of solutions, Zenoness, etc (see section 6 of [13] on “Ideal Solver Semantics” and section 7 of [14] on “Continuous Time Models”). In our approach those issues do not disappear from the whole process, but they are, sort of, postponed to run time, as wished in our introduction.

The work performed by P. Mosterman and his co-workers at The Mathworks [18] is also very interesting, in its attempt to establish the Simulink modeler on a solid semantic basis. The contribution of the paper is to show how (a restricted class of) variable step solvers can be given a functional stream semantics [11]. To achieve this, the class of solvers is first restricted to those relying on explicit schemes, as implicit ones cannot be put in explicit functional form. The second difficulty consists in the use of iterative solving in order to on-line adapt the variable step size. This mechanism, again, does not have a functional shape since several successive integrations with different step sizes are compared, for a same time interval, in order to select the appropriate step size. [18] proposes to re-cast the above procedure to a functional form by replacing a repeated integration with smaller step size, by its increment with respect to the previous integration. If explicit schemes are used, then an explicit form for this increment can be found and added to the previous integration. Observe that this technique requires using the mechanism of super-dense time since a same time interval is processed several times until adequate step size is found. While this indeed provides a hybrid systems modeler with a stream semantics, this semantics is extremely complex since it explicits the discretization method — in particular, changing the latter changes the semantics. This approach forbids using implicit schemes, although they are valuable from the numerical analysis point of view. We also believe that this method cannot easily support the kind of clock configuration dependent causality analysis such as the one provided by our constructive semantics.

VI. CONCLUSION

We have proposed a novel approach to give a semantics to hybrid systems modelers. In doing so, we wanted:

- To keep the choice of integration method totally free;
- To ensure that hybrid systems are a conservative extension of discrete time systems;
- To give semantic support for the following:
  - Scheduling the actions triggered by zero-crossings;
  - Using typing to separate discrete from continuous;
  - Rejecting programs with causality circuits;

More objectives are addressed in [2]. Achieving these objectives was made possible thanks to the use of non-standard analysis as our semantic domain. The key point is that non-standard semantics allows cleanly separating the tasks of the computer scientist (answering the above questions) from that of the numerical analyst (tuning the solvers). Also, we believe that non-standard semantics is not a fancy thing for math addicts. It is rather a very natural way of viewing continuous time and hybrid systems from the syntactic side, as the computer scientist usually likes. While the first author was aware of non-standard analysis since the mid eighties, it is only the presentation [15] by Lindström, as reported in [8], that allowed the authors to become familiar with the subject. In [2] we develop a small single-assignment language for hybrid systems modelers, with minimal type system to properly manage the discrete/continuous separation.

Next steps are the study of DAE compliant hybrid systems modelers, such as Modelica, with the same objectives.

ACKNOWLEDGEMENT: The authors are indebted to Raminre Nikoukhah and Sésabrien Furic for detailed discussions regarding Modelica, and to Daniel Krob and Simon Bludzde for comments on their work.

REFERENCES