Une sémantique formelle pour les modèles Simulink

Béatrice Bérard*‡, Yann Duplouy*†, Serge Haddad*
*LSV, ENS Paris-Saclay, CNRS, Inria, France †IRT SystemX, Paris-Saclay, France
‡Sorbonne Universités, UPMC Univ. Paris 06, CNRS UMR 7606, LIP6, Paris, France

***Résumé*—De nombreux projets industriels, notamment dans la construction automobile, font appel à la suite d'outils Simulink® pour la conception et la validation de composants critiques représentant des systèmes hybrides c'est-à-dire combinant des aspects discrets et continus. Cependant les formalismes associés ne disposent pas d'une sémantique formelle ce qui peut diminuer la confiance des ingénieurs vis-à-vis des résultats produits. Nous proposons ici une telle sémantique en procédant en deux étapes. Nous développons d'abord une sémantique exacte mais non exécutable. Puis nous l'enrichissons d'une sémantique opérationnelle approchée avec pour objectif une quantification de l'erreur issue de cette approximation.**

I. Introduction

Analyse des systèmes hybrides. La vérification des systèmes hybrides soulève de nombreuses questions partiellement résolues jusqu'à présent. La combinaison des caractères continu et discret de ces systèmes conduit très rapidement à l'indécidabilité de toutes les propriétés significatives [1]. Afin de pallier ce problème, plusieurs approches sont possibles : soit restreindre la puissance d'expression du modèle, soit concevoir des procédures de semi-décision, soit faire appel à la simulation. Nous nous situons ici dans ce dernier cadre.

Simulation des systèmes hybrides. Cependant, même l'approche par simulation se heurte à des difficultés majeures. De par leur nature, les systèmes hybrides nécessitent la résolution d'équations différentielles, ce qui n'est pas toujours possible de manière exacte. De la même façon, les changements de mode induits par la partie discrète impliquent la résolution d'équations $f(x) = 0$ pour des fonctions f arbitraires. L'outil Simulink®, extrêmement répandu dans l'environnement industriel, adopte un certain nombre de solutions à ces problèmes qui ne sont malheureusement pas explicitées à travers une sémantique formelle qui garantirait aux ingénieurs des résultats fiables. Nous proposons ici une telle sémantique en procédant en deux étapes. Nous développons d'abord une sémantique exacte mais non exécutable. Puis nous l'enrichissons d'une sémantique opérationnelle approchée avec pour objectif une quantification de l'erreur issue de cette approximation.

Application au véhicule autonome. Ce travail a été réalisé au sein du projet *Simulation pour la sécurité des Véhicules Autonomes* (SVA) dont l'un des axes consiste à élaborer des méthodes de simulation pour certifier la sûreté du véhicule autonome. Les partenaires du projet SVA fournissent des contrôleurs pour l'assistance aux conducteurs, généralement sous la forme de modèles Simulink®. Afin d'évaluer la qualité de ces contrôleurs, la sémantique approchée sera intégrée à l'outil de *model-checking* statistique COSMOS [2]. En particulier, nous chercherons à obtenir des garanties probabilistes sur les risques d'accident.

Travaux connexes. Des travaux ont déjà été menés pour associer une sémantique aux modèles Simulink®. Le premier, Chapoutot *et al* [3], définit une exécution symbolique ne reflétant que partiellement le modèle. Le second, Benveniste *et al* [4], propose une sémantique exacte basée sur l'analyse non-standard qui manipule des infinitésimaux et nécessite des versions approchées pour l'exécution. Nous adoptons une approche similaire à ce dernier travail, en adoptant le point de vue d'un contrôleur discret intervenant dans un environnement continu. Ceci nous permet d'éviter au maximum les infinitésimaux, pour une meilleure adaptation à l'outil de model-checking statistique.

Plan. Nous définissons d'abord la syntaxe des modèles Simulink® (Section II), puis une sémantique exacte pour ces modèles (Section III) suivie d'une version approchée (Section IV).

II. Syntaxe

Dans cette section, nous définissons les *SK*-modèles, qui constituent une syntaxe formelle des modèles Simulink®. Un tel modèle est représenté graphiquement comme un ensemble de boîtes reliées par des fils. Les boîtes sont appelées *blocs*, et les fils sont les supports de *signaux*. De manière plus précise, les blocs agissent comme des *opérateurs* transformant des signaux d'entrée en signaux de sortie.

A. Signaux et types

On associe des types aux signaux et aux opérateurs et on note $\mathcal{T}ype$ l'ensemble des types utilisés dans les *SK*-modèles.

Définition 1 (Types de base, constructeurs de types)**.** *Les types de base forment un sous-ensemble de $\mathcal{T}ype$, contenant des sous-ensembles de l'ensemble $\mathbb{R}$ des nombres réels :*

- *l'ensemble des booléens $\mathbb{B} = \{0, 1\}$;*
- *les entiers (non-)signés, représentés avec différents nombres de bits (`int8`, `uint8`, `int16`, `uint16`, `int32`, `uint32`), $\mathbb{N}$ ou $\mathbb{Z}$;*
- *les nombres flottants (`double`, `single`) ou $\mathbb{R}$.*

Un constructeur de type est une fonction $\mathcal{T}ype^n \to \mathcal{T}ype$ où $n \in \mathbb{N}$ est l'arité du constructeur.

Le travail de Yann Duplouy a été effectué dans le cadre des recherches menées au sein de l'IRT SystemX, Paris-Saclay, France, et a ainsi bénéficié d'une aide de l'État au titre du programme d'Investissements d'Avenir. Celui de Serge Haddad est soutenu par le projet ERC EQualIS (FP7-308087).

Remarque. *Bien que les types $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ ne puissent pas être réprésentés de manière exacte sur un ordinateur, ils sont utilisés pour définir la sémantique exacte des SK-modèles.*

Exemple. *Le type $Tuple_3(\mathbb{N}, \mathbb{R}, \mathbb{R})$ est construit avec le constructeur $Tuple_3$. Il permet de combiner trois signaux sous la forme d'un seul signal, ce qui correspond à un multiplexage.*

Le domaine de temps, noté $\mathbb{T}$, est un intervalle $[t_{\text{init}}, t_{\text{end}}]$ de $\mathbb{R}$.

Définition 2. *Un* signal *de type $\mathcal{T}p \in \mathcal{T}ype$ est une fonction $s : \mathbb{T} \to \mathcal{T}p$ continue à droite, $\mathcal{C}^\infty$ par morceaux et qui admet une limite à gauche pour tout $t \in \mathbb{T}$, notée $s(t^-)$. On note $\mathcal{S}ig_{\mathbb{T}}(\mathcal{T}p)$ l'ensemble des signaux de type $\mathcal{T}p$ définis sur $\mathbb{T}$.*

Un opérateur *est une fonction* $\text{op} : \mathcal{S}ig_{\mathbb{T}}(\mathcal{T}p_1) \times \ldots \times \mathcal{S}ig_{\mathbb{T}}(\mathcal{T}p_m) \to \mathcal{S}ig_{\mathbb{T}}(\mathcal{T}p)$ *telle que pour tout $t \in \mathbb{T}$ la valeur $\text{op}(s_1, \ldots, s_m)(t)$ ne dépend que de la restriction des signaux s_i sur $[t_{\text{init}}, t]$.*

B. Blocs

Un bloc contient un ensemble d'opérateurs générant des signaux de sortie à partir des signaux d'entrées. Il existe trois critères principaux de classification des blocs.

- Les signaux sont-ils évalués continûment ou échantillonnés ? Dans le second cas, le bloc est qualifié de *discret* et un *délai d'échantillonnage* doit être spécifié.
- Y a-t-il un *retard* sur l'évaluation des entrées ? Ce retard est-il *infinitésimal* ? Nous entendons ici par *infinitésimal* le cas d'un retard nul, ou tel que la valeur des signaux de sortie à l'instant t ne dépend que des valeurs des signaux d'entrée sur l'intervalle $[t_{\text{init}}, t[$ (comme, par exemple, pour l'intégration). Lorsque le retard est nul, le bloc est qualifié d'*immédiat*. Lorsque le retard n'est pas infinitésimal, il est qualifié de *positif*. Un retard *positif* ou *infinitésimal non-nul* est qualifié de *non nul*.
- La valeur du signal est-elle conditionnée au franchissement de seuils par un des signaux d'entrée (noté i_c et appelé *entrée critique*) ? Dans ce cas, le bloc est appelé *bloc à seuil* et les valeurs de ces seuils doivent également être spécifiées par une suite strictement croissante $(v_i)_{i \in I}$ ne comportant pas de point d'accumulation, indexée par l'ensemble fini ou dénombrable I.

Définition 3 (Type de bloc)**.** *Un* type de bloc *est un tuple* $\mathsf{BT} = (n, m, (\text{op}_i)_{1 \le i \le n}, b_c, b_l, b_i, b_s, \text{Param})$ *avec :*

- *n le nombre de signaux de sortie et m le nombre de signaux d'entrée ;*
- *$(\text{op}_i)_{1 \le i \le n}$ un tuple d'opérateurs dont les types sont compatibles avec les types des signaux d'entrée et de sortie, un par signal de sortie ;*
- *b_c un booléen indiquant si le bloc est discret ou continu ;*
- *b_l un booléen indiquant si le bloc est immédiat ou retardé ;*
- *b_i un booléen indiquant si le bloc a un retard infinitésimal ;*
- *b_s un booléen indiquant si le bloc comporte des seuils ;*
- *Param est un ensemble de paramètres, qui inclut le délai d'échantillonnage δ si le bloc est discret et le retard r si le bloc a un retard non-infinitésimal.*

Un type de bloc générique *sur un ensemble $\mathcal{G}$ est une famille $(\mathsf{BT}_g)_{g \in \mathcal{G}}$ de types de bloc.*

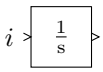

FIGURE 1

Le bloc de la 1 est un bloc Intégrateur. Il s'agit d'un bloc continu, à retard infinitésimal.

$$\text{op}(i)(t) = \int_{t_{\text{init}}}^{t} i(\tau)\mathrm{d}\tau$$

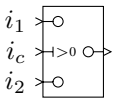

FIGURE 2

Le bloc de la FIGURE 2 est un bloc *Switch*. Il s'agit d'un bloc à seuil (sur l'entrée i_c) ayant un seul point de discontinuité. Ce bloc est continu, immédiat et sans retard.

$$\text{op}(i_1, i_c, i_2)(t) = \mathsf{si}(i_c(t) > 0, i_1(t), i_2(t))$$

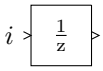

FIGURE 3

Le bloc de la FIGURE 3, *Unit Delay*, échantillonne son entrée tous les δ, avec un retard de δ. Il s'agit donc d'un bloc discret à retard non-infinitésimal δ.

C. SK-Modèles

Un *SK*-modèle définit une architecture dans laquelle les blocs sont des instances des types de blocs, dont les valeurs des paramètres sont spécifiées. Un exemple de *SK*-modèle est donné à la FIGURE 4.

Définition 4. *Un SK-modèle $\mathcal{M} = (\mathcal{B}, L)$ se compose :*

- *d'un ensemble $\mathcal{B} = \{B_k \mid 1 \le k \le K\}$ de K blocs. Chaque bloc B_k est défini par :*
  - *son type BT_k (avec m_k entrées et n_k sorties) ;*
  - *une instanciation de l'ensemble de ses paramètres (incluant le délai d'échantillonnage δ_k et le retard r_k si nécessaire).*

  On note $\text{O}(\mathcal{B})$ l'ensemble des ports de sortie et $\text{I}(\mathcal{B})$ l'ensemble des ports d'entrée, k^+ (resp. k^-) indique une sortie (resp. entrée) du bloc k :

$$\begin{aligned} \text{O}(\mathcal{B}) &= \{\langle k^+, o\rangle \mid 1 \le k \le K \wedge 1 \le o \le n_k\} \\ \text{I}(\mathcal{B}) &= \{\langle k^-, i\rangle \mid 1 \le k \le K \wedge 1 \le i \le m_k\} \end{aligned}$$

- *un ensemble de liens $L \subseteq \text{O}(\mathcal{B}) \times \text{I}(\mathcal{B})$ qui satisfait :*

$$\forall \langle k^-, i\rangle \in \text{I}(\mathcal{B}) : \exists! \langle k'^+, o\rangle \in \text{O}(\mathcal{B}) \text{ tq } (\langle k'^+, o\rangle, \langle k^-, i\rangle) \in L$$

On note $\text{src}(k^-, i)$ l'unique port $\langle k'^+, o\rangle$ qui y est relié.

Le graphe $\mathcal{G}_{\mathcal{M}}$ de $\mathcal{M} = (\mathcal{B}, L)$ est un multigraphe étiqueté défini de manière naturelle : $\mathcal{B}$ est l'ensemble des nœuds, et il existe un arc de B_i vers B_j pour chaque lien entre ces deux blocs, étiqueté par les numéros de ports.

Afin de justifier la restriction qui suit, nous anticipons sur la section III. Nous appelons *trajectoire* d'un *SK*-modèle l'ensemble des valeurs des signaux de sortie sur $\mathbb{T}$. L'application des opérateurs nécessaire pour obtenir cette trajectoire induit des contraintes de dépendance temporelle entre les blocs et, plus spécifiquement, entre les blocs immédiats. La définition suivante permet d'identifier des dépendances incorrectes :

Définition 5. *Un* cycle immédiat *d'un SK-modèle $\mathcal{M}$ est une suite de liens*

$$\begin{aligned} &(\langle k_1^+, o_1\rangle, \langle k_2^-, i_2\rangle), (\langle k_2^+, o_2\rangle, \langle k_3^-, i_3\rangle), \\ &\ldots (\langle k_{n-1}^+, o_{n-1}\rangle, \langle k_n^-, i_n\rangle) \text{ avec } k_n = k_1 \end{aligned}$$

telle que pour tout $i \in [\![1, n]\!]$, B_{k_i} est un bloc immédiat.

Définition 6. *Un SK-modèle est dit* correct *s'il ne possède pas de cycle immédiat.*

Proposition 1. *Le problème de la correction d'un SK-modèle est décidable en temps linéaire.*

La correction d'un *SK*-modèle se vérifie aisément à l'aide d'un tri topologique restreint aux blocs immédiats. Dans la suite, on ne considère que des modèles corrects.

III. Sémantique exacte

Nous décrivons maintenant une sémantique ne prenant pas en compte d'éventuelles difficultés d'implémentation et qui servira de base pour une sémantique approchée mais implémentable (section IV).

La trajectoire d'un *SK*-modèle, lorsqu'elle existe, s'obtient informellement comme suit. L'intervalle $[t_{\text{init}}, t_{\text{end}}]$ est découpé en une suite finie de sous-intervalles contigus. Les extrémités de ces intervalles – à l'exception de t_{init} et t_{end} – correspondent soit à un franchissement de seuil pour un des blocs à seuil du modèle, soit à l'échantillonnage d'un bloc discret. A l'intérieur de chaque intervalle, la trajectoire est la solution d'un système d'équations différentielles dont la spécification dépend des valeurs des signaux obtenues jusqu'à présent.

A. Equations différentielles d'un SK-modèle

Afin de spécifier l'équation différentielle liée à un bloc d'intégration B, on procède par une exploration arrière de $\mathcal{G}_{\mathcal{M}}$ à partir du bloc, noté B^-, dont l'une des sorties, notée o^-, est l'entrée de B. Cependant cette exploration doit comporter un test d'arrêt. A cette fin, on introduit les *blocs terminaux*. Un bloc terminal est soit un bloc sans entrée, soit un bloc avec un retard non nul. Les sous-graphes obtenus par exploration arrière sont ainsi définis par :

Définition 7 (Graphe arrière)**.** *Soit $\mathcal{M}$ un SK-modèle. Le* graphe arrière *$\mathcal{G}_B$ d'un bloc B est un graphe acyclique orienté enraciné en B défini inductivement par :*

- *si B est un bloc terminal alors $\mathcal{G}_B$ est réduit au bloc B sans arc ;*
- *sinon en notant $B_1, \ldots, B_m$ les blocs dont des sorties sont liées aux signaux d'entrées de B alors $\mathcal{G}_B$ est obtenu en ajoutant à $\cup_{i=1}^{m} \mathcal{G}_{B_i}$ les liens de $B_1, \ldots, B_m$ vers B.*

La spécification de l'équation différentielle à appliquer dépend aussi du *mode* des blocs à seuil. On note $\mathsf{Th}(\mathcal{M})$ l'ensemble des blocs à seuil de $\mathcal{M}$.

Définition 8 (Mode)**.** *Les discontinuités des blocs à seuil définissent une partition de $\mathbb{R}$ en intervalles dont les extrémités finies correspondent aux points de discontinuité $\{v_i\}_{i \in I}$. Un* mode *est le choix d'un intervalle pour chaque bloc à seuil.*

Afin de compléter la spécification du système d'équations différentielles, il reste à définir quelles sont les fonctions à substituer pour chacun des blocs terminaux. Il y a trois cas de bloc terminal B à considérer :

- B est un bloc sans entrée. Dans ce cas la fonction associée à chaque sortie de B est substituée dans l'équation différentielle ;
- B est un bloc d'intégration. Dans ce cas la variable d'intégration associée à la sortie de B est substituée dans l'équation différentielle ;
- B est un bloc à retard positif. Dans ce cas la fonction à substituer doit être fournie par un mécanisme extérieur ce qui nous conduit à la définition suivante.

On note $\mathsf{Lat}(\mathcal{M})$ l'ensemble des blocs à retard positif de $\mathcal{M}$. On supposera que chacun de ces blocs n'a qu'une seule sortie.

Définition 9 (Environnement)**.** *Un* environnement *lat est le choix d'une fonction $lat(B, o)$ pour chaque sortie o d'un bloc B de $\mathsf{Lat}(\mathcal{M})$.*

Nous définissons maintenant le système d'équations différentielles obtenu une fois fixé un mode et un environnement.

Définition 10 (Équations différentielles d'un modèle)**.** *Soit $mode$ un mode et lat un environnement, l'équation différentielle associée à un bloc d'intégration B, dont la variable est notée x_B, est obtenue en associant à chaque sortie o d'un bloc B' de $\mathcal{G}_{B^-}$ une expression $y_{B',o}$ définie inductivement par :*

- *Si B' est un bloc sans entrée alors $y_{B',o} = \text{op}_{B',o}$;*
- *Si B' est un bloc d'intégration alors $y_{B',o} = x_{B'}$;*
- *Si B' est un bloc à retard positif alors $y_{B',o} = lat(B', o)$;*
- *Si B' est un bloc non terminal ayant comme entrées $(B_1, o_1), \ldots, (B_m, o_m)$ alors $y_{B',o} = \text{op}_{B',o}(y_{B_1,o_1}, \ldots, y_{B_m,o_m})$.*

L'équation différentielle est alors $\dot{x_B} = y_{B^-,o^-}$.

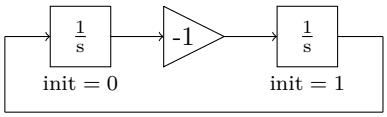

FIGURE 4: Un *SK*-modèle représentant une fonction sinusoïdale. Le bloc du centre multiplie son entrée par -1.

On considère le modèle de la FIGURE 4 dans lequel les blocs sont numérotés de gauche à droite avec x_i en sortie de B_i pour $i \in \{1, 2, 3\}$. Les équations sont $\dot{x_1} = x_3$ et $\dot{x_3} = -x_1$, ce qui permet de retrouver $\ddot{x_1} = -x_1$, l'équation caractéristique d'une fonction sinusoïdale.

B. Partition en sous-intervalles

La partition de l'intervalle de simulation en sous-intervalles peut être due à différents facteurs :

- Les blocs à échantillonnage modifient leur sortie en temps nul à chaque instant d'échantillonnage créant ainsi une discontinuité et imposant à cet instant un découpage de l'intervalle.
- Les signaux des blocs à retard positif nécessitent d'être connus durant le sous-intervalle où l'équation différentielle est fixée. Ainsi la longueur d'un sous-intervalle ne peut excéder le délai positif minimal. De

plus, les *SK*-modèles fixent un délai $\delta_{\max}$ bornant la longueur du sous-intervalle.
- Les valeurs des signaux i_c, en entrée des blocs à seuil, doivent satisfaire le mode spécifié par l'équation différentielle à résoudre. Un changement de mode entraine un nouveau découpage.

Les deux premiers cas conduisent à introduire la fonction de *pas suivant statique* $\text{next}_s(t)$ qui étant donné t, la borne inférieure d'un sous-intervalle, fournit la borne supérieure (maximale) du sous-intervalle satisfaisant ces conditions.

Définition 11. *Soit $\mathcal{M}$ un SK-modèle, Δ l'ensemble de ses délais d'échantillonnage et R l'ensemble des retards positifs. Soit $\delta_{\text{lat}} = \min(\delta_{\max}, \min(R))$. On définit la fonction de* pas suivant statique $\text{next}_s(t)$ *pour un temps $t < t_{\text{end}}$ par :*

$$\begin{array}{rcl} \delta_{\text{samp}}(t) & = & \min\{p\delta \mid \delta \in \Delta, p \in \mathbb{N}, p\delta > t\} \\ \text{next}_s(t) & = & \min(\delta_{\text{samp}}(t), t + \delta_{\text{lat}}, t_{\text{end}}) \end{array}$$

Au changement d'intervalle, les signaux de sortie de certains blocs à échantillonnage doivent être réévalués. Cette opération requiert de définir un ordre d'évaluation des blocs. Cet ordre d'évaluation est aussi nécessaire pour évaluer les signaux qui ne sont pas sortie d'un bloc intégrateur après la résolution d'une équation différentielle. Plusieurs ordres sont possibles mais tous conduisent à la même valeur de ces signaux.

Définition 12 (Ordre des blocs)**.** *L'*ordre des blocs BO *d'un SK-modèle $\mathcal{M}$ (correct) est un ordre total étendant l'ordre suivant sur les blocs :*

1) les blocs sans entrée ;
2) les blocs avec un retard non-infinitésimal par ordre de retard croissant ;
3) les blocs avec retard infinitésimal ;
4) les blocs immédiats dans l'ordre topologique de la proposition 1.

C. Existence d'une trajectoire

Nous sommes maintenant en mesure de caractériser l'existence d'une trajectoire d'un *SK*-modèle $\mathcal{M}$ sur l'intervalle $[t_{\text{init}}, t_{\text{end}}]$. Notons $\vec{w}$ le vecteur des signaux de sortie, $\vec{v}$ le sous-vecteur de $\vec{w}$ contenant les signaux en sortie des blocs intégrateurs et $\dot{X} = F_{mode,lat}(X)$ le système d'équations différentielles associé à $\vec{v}$, une fois fixés un mode et un environnement. On note $mode_0$ le mode initial des blocs à seuil et lat_0 les valeurs initiales des blocs à retard positif. Un modèle admet la trajectoire $\vec{w}$ si et seulement si il existe une suite $(t_i)_{i\in[\![0,N]\!]}$ contenant les instants d'échantillonnage – avec $t_0 = t_{\text{init}}$ et $t_N = t_{\text{end}}$ –, une suite $(mode_i)_{i\in[\![1,N-1]\!]}$ et $\varepsilon_{\mathbb{T}} > 0$ tels que pour tout i :

1. Si $0 < i < N$, d'après les hypothèses sur le découpage en sous-intervalles lat_i est défini sur l'intervalle $[t_i, t_{i+1}]$ par $\vec{w}$ dans l'intervalle $[t_0, t_i]$. Soit $mode_{temp}$ le mode défini par $\vec{w}$ à l'instant t_i : l'équation différentielle $\dot{X} = F_{mode_{temp},lat_i}(X)$ admet une solution $\vec{v}_{temp}$ dans l'intervalle $[t_i, t_i+\varepsilon_{\mathbb{T}}]$ et $\vec{w}_{temp}$ obtenue par évaluation des blocs immédiats satisfait $mode_i$ durant $]t_i, t_i + \varepsilon_{\mathbb{T}}]$.

2. Dans l'intervalle $[t_i, t_{i+1}]$, l'équation différentielle $\dot{X} = F_{mode_i,lat_i}(X)$ admet $\vec{v}$ comme solution et $\vec{w}$ est obtenue par évaluation des blocs restants suivant l'ordre indiqué ci-dessus. De plus le mode $mode_i$ est satisfait par $\vec{w}$ sur l'intervalle $]t_i, t_{i+1}[$.

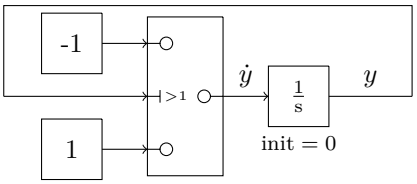

FIGURE 5: Un *SK*-modèle de sémantique fail

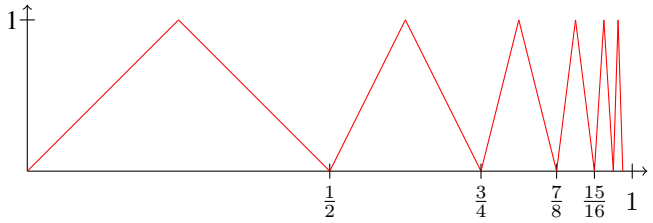

FIGURE 6: Une génération de trajectoire qui échoue par nombre de discontinuités infinies

Le *SK*-modèle de la FIGURE 4 possède une trajectoire sur $\mathbb{T} = [0, \pi]$: $x_1(t) = \sin(t)$, $x_2(t) = -\sin(t)$ et $x_3(t) = \cos(t)$.

S'il n'existe pas une telle trajectoire alors la sémantique du modèle est fail. Voici trois cas d'échec de génération de trajectoire :

- Le modèle contient une équation différentielle qui n'admet pas de solution sur $\mathbb{T}$ (comme par exemple $\dot{y} = y^2$ avec $y(0) = 1$ sur $[0, 1]$) ;
- Il n'est pas possible de trouver une suite de modes compatibles avec les solutions des équations différentielles. L'exemple de la FIGURE 5 sur l'intervalle $\mathbb{T} = [0, 2]$ illustre ce cas car, peu importe le choix de mode effectué sur le bloc *Switch*, la solution de l'équation différentielle obtenue violera la contrainte du bloc à seuil. En effet, $y(t) = t$ sur $[0, 1]$. Cependant, sur $]1, 1 + \varepsilon_{\mathbb{T}}]$: soit $\dot{y} = 1$ et alors $y(t) = t > 1$ ce qui ne satisfait pas le mode choisi. Soit $\dot{y} = -1$, alors $y(t) = 2 - t < 1$ ce qui ne satisfait pas non plus le mode choisi.
- Le nombre de discontinuités sur $\mathbb{T}$ est infini, comme dans le cas d'une trajectoire dont le début est illustré FIGURE 6.

Cette sémantique semble de prime abord non déterministe puisqu'elle dépend a priori du découpage de l'intervalle, de la valeur de $\varepsilon_{\mathbb{T}}$ et des solutions des équations différentielles successives. Cependant, compte tenu des hypothèses sur les opérateurs (qui sont $\mathcal{C}^\infty$ par morceaux), nous avons :

Proposition 2. *Si un SK-modèle admet une trajectoire, alors cette trajectoire est unique.*

Le point de vue que nous avons adopté pour la sémantique exacte des *SK*-modèles est celui d'un contrôleur discret intervenant dans un environnement continu. Ce contrôleur est limité par ses capacités de réaction, spécifiées implicitement par la valeur $\varepsilon_{\mathbb{T}}$. Ainsi lors des changements d'intervalle, le

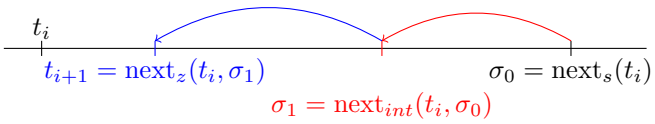

FIGURE 7: Recherche de t_{i+1}

contrôleur doit observer continuellement un mode pour décider de l'application de ce mode à l'équation différentielle suivante.

Il est bien connu que les problèmes associés aux trajectoires des systèmes hybrides sont souvent indécidables (*e.g.* dépassement d'un seuil). Ici le problème même de l'existence d'une trajectoire sur un intervalle fini est indécidable.

Proposition 3. *Le problème de l'existence d'une trajectoire pour un SK-modèle est indécidable.*

IV. SÉMANTIQUE APPROCHÉE

La sémantique exacte repose sur la résolution d'équations différentielles et la détermination des zéros de fonctions arbitraires nécessaires pour repérer les changements de mode. Il est bien connu que de telles opérations ne sont pas effectives. Aussi, nous définissons maintenant une *sémantique approchée* qui permettra la conception d'un moteur d'exécution des *SK*-modèles. De plus, il devrait être possible de fournir des garanties sur l'approximation vis-à-vis de la sémantique exacte.

Principe général. La sémantique approchée repose sur la construction itérative d'une partition en sous-intervalles de $[t_{\text{init}}, t_{\text{end}}] = \bigcup_{i=0}^{N-1}[t_i, t_{i+1}]$. Afin de contrôler les erreurs introduites par les approximations, on définit $\varepsilon_{\mathbb{T}}$ comme le *pas de temps minimal* (de manière analogue à la sémantique exacte) et ε_V comme la capacité de discernement du contrôleur ($|x| < \varepsilon_V \Rightarrow x \simeq 0$). La sémantique approchée remplace les trajectoires complètes par un tableau indiquant, pour chaque sortie o d'un bloc B_k, les valeurs $W_{k,o}[i]$ à chaque instant t_i, $0 \leq i \leq N$.

Pas d'itération. Supposons connues $t_0, \ldots, t_i$ ainsi que $W_{k,o}[j]$ pour tout k, o et $0 \leq j \leq i$. La première étape est de déterminer t_{i+1}. Cette valeur est obtenue en appliquant d'abord la fonction de pas suivant statique next_s : $\sigma_0 = \text{next}_s(t_i)$. Cette valeur sera ensuite éventuellement diminuée :

1) premièrement, par application d'une méthode d'intégration à pas variable – comme ODE45 [1] [5] – entre t_i et σ_0 sur l'ensemble des blocs intégration, obtenant ainsi $\sigma_1 = \text{next}_{int}(t_i, \sigma_0)$;
2) en détectant ensuite les changements de mode des blocs à seuil, par approximation linéaire. On obtient alors $t_{i+1} = \text{next}_z(t_i, \sigma_1)$.

Dans les deux cas, il est nécessaire de calculer une partie des valeurs aux points intermédiaires générés par ces méthodes : celles correspondant aux graphes arrières des blocs concernés.

Intégration à pas variable dans un *SK*-modèle. La méthode choisie, ODE45, procède en calculant simultanément deux approximations par les méthodes de Runge-Kutta d'ordre 4 et 5 (notées respectivement RK4 et RK5), comme décrit dans [5]. Si les deux méthodes renvoient des résultats trop éloignés, le pas de calcul est divisé par deux et on réitère le procédé.

Cependant, les méthodes de Runge-Kutta supposent connue (ou donnée de façon explicite) l'équation différentielle. Or, elle est ici spécifiée implicitement par le *SK*-modèle. Il faut donc adapter la résolution, qui se fera par l'évaluation des blocs des graphes arrières aux instants intermédiaires de la méthode de Runge-Kutta.

Remarque. *Il est possible qu'un des blocs à seuil change de* mode *entre deux étapes intermédiaires de calcul de l'approximation de l'intégrale. La détection des franchissements de seuil étant faite immédiatement après, l'application des méthodes de Runge-Kutta se fait à mode constant (celui de l'instant t_i).*

Franchissement de seuil. Soit $\sigma_1 = \text{next}_{int}(t_i, \sigma_0)$ (notations de la figure 7). On détermine s'il y a un changement de mode entre t_i et σ_1. Si tel est le cas, on calcule l'instant minimal de changement de mode parmi l'ensemble des blocs à seuil, en effectuant une interpolation linéaire de chaque entrée critique. Cette interpolation linéaire est évaluée en bornant la précision à ε_V.

Calcul des nouvelles valeurs. L'instant suivant $t_{i+1} = \text{next}_z(t_i, \text{next}_{int}(t_i, \text{next}_s(t_i)))$ étant déterminé par la procédure précédente, on calcule les valeurs des sorties des blocs à cet instant dans l'ordre des blocs. Pour les blocs intégrateur, la garantie de précision ayant déjà été fournie par ODE45, il suffit de calculer leur valeur en choisissant RK4. Pour le reste des blocs, on effectue le calcul standard conformément au mode courant (celui à l'instant t_i).

Ensuite, de manière similaire au déroulement de la sémantique exacte, on effectue le calcul des nouvelles valeurs en $t_{i+1} + \varepsilon_{\mathbb{T}}$ afin d'établir le mode en t_{i+1}. Si un changement de mode apparaît dans l'intervalle $]t_{i+1}, t_{i+1} + \varepsilon_{\mathbb{T}}[$ alors le résultat est fail.

Nous projetons de démontrer la conjecture suivante, qui exprime que la sémantique approchée fournit une bonne approximation de la trajectoire de la sémantique exacte :

Conjecture. Sous des hypothèses raisonnables et si la sémantique exacte fournit une trajectoire $\vec{w}$ alors pour tout ε, il existe des valeurs de $\varepsilon_{\mathbb{T}}$ et de ε_V telles que :

$$\forall t_i, \forall k, \forall o, |W_{k,o}[i] - w_{k,o}(t_i)| < \varepsilon$$

V. CONCLUSION

Nous avons défini les *SK*-modèles, une syntaxe pour les architectures Simulink®, et deux sémantiques. La sémantique exacte décrit son comportement idéal, indépendamment d'une réalisation par un outil tandis que la version approchée permet l'exécution d'un *SK*-modèle.

La prochaine étape de ce travail consistera à intégrer la sémantique approchée dans un outil afin de valider des contrôleurs destinés à l'assistance aux conducteurs. Il est prévu pour cela d'étendre COSMOS, un outil de *model checking* statistique spécifiant des indices de performances à l'aide de la logique HASL [2], pour obtenir l'exécution du contrôleur

1. encore appelée Runge-Kutta-Fehlberg, du nom de leurs auteurs

dans un environnement aléatoire, modélisé par un réseau de Petri statistique coloré.

RÉFÉRENCES

[1] T. A. Henzinger and J. Raskin, "Robust undecidability of timed and hybrid systems," in *Hybrid Systems : Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1790. Springer, 2000, pp. 145–159.

[2] P. Ballarini, B. Barbot, M. Duflot, S. Haddad, and N. Pekergin, "HASL : A new approach for performance evaluation and model checking from concepts to experimentation," *Perform. Eval.*, vol. 90, pp. 53–77, 2015.

[3] O. Bouissou and A. Chapoutot, "An Operational Semantics for Simulink's Simulation Engine," in *Proceedings of the 13th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, Tools and Theory for Embedded Systems*, ser. LCTES'12. ACM, 2012, pp. 129–138.

[4] A. Benveniste, T. Bourke, B. Caillaud, and M. Pouzet, "Non-standard semantics of hybrid systems modelers," *Journal of Computer and System Sciences*, vol. 78, no. 3, pp. 877–910, 2012.

[5] E. Fehlberg, *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, ser. NASA technical report. National Aeronautics and Space Administration, 1969.