ハイブリッドシステムモデリング言語HydLaの区間制約に基づく全解シミュレーション実行処理系

大谷 順司† 廣瀬 賢一† 石井 大輔†† 細部 博史‡ 上田 和紀††
† 早稲田大学大学院 †† 早稲田大学理工学術院 ‡ 国立情報学研究所

1 はじめに

時間の経過に伴って状態が連続変化したり，状態や方程式自体が離散変化したりする系をハイブリッドシステムと呼ぶ．様々な分野のモデルがハイブリッドシステムとして統一的に扱えるため，その記述やシミュレーション，検証手法が研究されている [1]．ハイブリッドシステムの記述形式としてハイブリッドオートマトンやHybridCCがあるが，複雑なモデルの制約を過不足なく記述するのは困難である．また，高信頼なシミュレーションや検証のためには不確実性を持つモデルの到達可能範囲をできるだけ正確に計算する必要があるが，既存ツールの中には対応していないものも多い．

我々はより簡潔な記述を目指したモデリング言語として，制約概念と制約階層に基づくモデリング言語HydLa[2] を提案，開発している．また，HydLaの数式処理によるシミュレーション実行処理系の実装が進んでいるが [3]，不確実性を持つモデルを扱うことはできず，不確実性を持ったHydLaモデルの処理手法は自明ではない．

そこで我々は，不確実性を持つモデルに対して解の確実な包囲を計算する新たなシミュレーション実行処理系を作成した．本論文では作成した新たな処理系について述べる．

2 HydLa

HydLaは制約概念に基づきハイブリッドシステムをモデリングするための宣言型言語である．“時刻の関数”である変数に関する非線形等式・不等式と常微分方程式を制約として記述することでモデリングする．また，制約が条件式を含意するか否かによって条件判断をおこなう．HydLaでは，制約システムをパラメタとして考え，記述可能な式のクラスをとくに定めないが，本論文では常微分方程式が解析的に求解可能なものを対象とする．さらに，HydLaは制約階層 [4] を持つ．制約間に優先度を設けることで，制約を簡潔かつ過不足なく与えることを目標としている．

HydLaプログラムの解は，HydLaプログラム中の関数変数の値を具体化した実ベクトル関数 $\mathbb{R}_{\geq 0} \to \mathbb{R}^n$($n$ は関数変数の数) である．不確実性を持つモデルでは解を1つに定めることはできず，モデルの到達可能範囲は無数の解を束ねたものとなる．

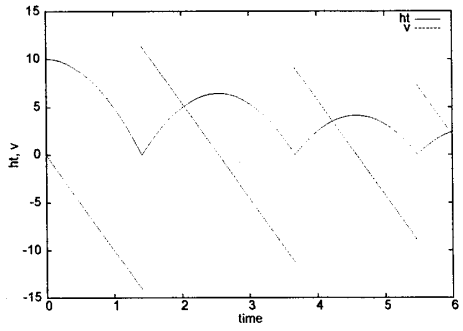

図1: ボールが自由落下し地面で弾むモデルの実行結果

例として物体が自由落下し地面で弾むモデルのHydLaプログラムを以下に示す．またモデルが描く軌道（解）の1つ，初期位置10としたものを図1に示す．

```
INIT   <=> 9 <= ht/\ht <= 11 /\ v = 0.
FALL   <=> [](ht' = v /\ v' = -10).
BOUNCE <=> [](ht- = 0 => v = -4/5*v-).
INIT, FALL << BOUNCE.
```

htが物体の位置（高さ），vが速度を表す関数変数である．1–3行目のINIT，FALL，BOUNCEは制約の定義で，INITは時刻0における初期値を，FALLは連続変化(自由落下)を，BOUNCEは離散変化(床との衝突)を記述している．9 <= ht /\ ht <= 11は不確実な初期位置を表し，このように変数の取りえる値の範囲（例えば $9 \leq \mathtt{ht} \leq 11$）が得られる制約を区間制約と呼ぶ．なお，[]は時刻0以降常に成り立つ制約であることを，=>は左辺を条件式とする条件文であることを，ht'はhtの導関数を，ht-は離散変化時刻におけるhtの左極限値を表す．4行目で制約の合成をおこなっている．FALLとBOUNCEについて，両者が矛盾した場合はBOUNCEを優先的に採用するように合成し，さらに優先順序をつけずにINITを合成している．HydLaの詳細な仕様については文献 [2] を参照してほしい．

3 実行処理系の設計

閉区間（開区間 $(-\infty, \infty)$ を含む）I の組 $(I_1, \ldots, I_n)$ を n 次元の box という．作成する実行処理系は，HydLa

An Interval-Based Simulator of Hybrid Systems Modeled in HydLa
†Graduate School of Sci. and Engr., Waseda University
††Faculty of Sci. and Engr., Waseda University
‡National Institute of Informatics

プログラムとシミュレーション時間を入力とし，時間までの HydLa プログラムの解全体（到達可能範囲）を包囲する box 集合を時刻 $t = 0$ から各離散変化，連続変化ごとに計算，出力する．

各離散変化，連続変化ごとの到達可能範囲は，常微分方程式を解析的に解くことで時間変数 t を含む非線形制約の解として表せる．本処理系では非線形制約の求解手法として BranchAndPrune アルゴリズム [5] を用いる．BranchAndPrune は入力として非線形制約 $f(x) \bullet 0$ (f は連続関数 $\mathbb{R}^n \to \mathbb{R}$, $\bullet \in \{=, \leq, \geq\}$)，$n$ 変数の組 x，各変数のとりうる範囲を格納した box D_0，最大区間幅 $\epsilon \in \mathbb{R}$ を受け取り，D_0 中の非線形制約の解が全て含まれている box の集合 $\mathcal{D}$ を出力する．各 $D \in \mathcal{D}$ の最大区間幅は ϵ 以下であり，各 $D \in \mathcal{D}$ は box-consistent [5] のため，解を漏れなくかつ指定した区間幅で包囲できる．また各 D 内に制約の唯一解の存在保証 $g \in \{true, false\}$ を付加できる．区間制約や条件式も非線形制約として BranchAndPrune へ入力可能である．

基本的には，各変化ごとに満たすべき制約を決定し，制約を BranchAndPrune で解くことで box 集合が得られる．しかし，各変化で制約を 1 つに決定できない場合が存在する．1 つは無矛盾極大な制約が複数ある場合で，HydLa ではどれかを満たすもの全てが解となるため，それぞれについて box 集合を求める．もう 1 つは制約間の含意関係 (entailment) が "不明" である場合である．区間制約により，含意関係に基づく条件判断は {真, 偽, 不明} の 3 値を考える必要がある．"不明" とは条件式を含意する区間としない区間両方を持つことを意味し，その場合はそれぞれの区間について条件判断に基づき制約を決定し解く．条件判断は BranchAndPrune で制約と条件式とを解くことで保守的に判断する．制約を S，条件式を g としたとき，$S \wedge g$ を解き $\emptyset$ であれば "偽"，$S \wedge \neg g$ を解き $\emptyset$ であれば "真"，それ以外ならば "不明" と判断する．

また，連続変化の処理では，条件式の 1 つに対して真偽が変化する時刻とその時刻における各変数の値が次の離散変化の開始状態となるため，条件式またはその否定を満たすべき制約に加えて解くことが必要である．ここで BranchAndPrune の出力に解の存在保証を付加することで，その条件式による次の離散変化が必ず起こるものかどうかを判断する．出力の全てに存在保証があれば離散変化は必ず起こるが，そうでないならば，他の条件式について同様に計算するか，連続変化が継続する軌道も考慮する必要がある．

4 実装と例題

以上の手法を確認するためのプロトタイプ実装をおこない，上記のボールの例題をシミュレーションした．

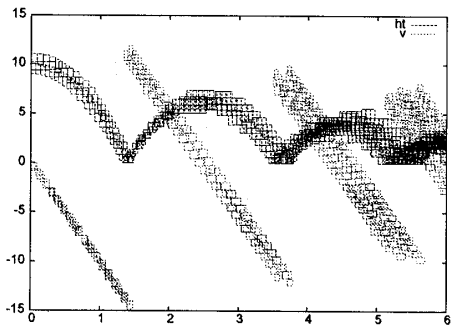

図 2: 提案手法実装による実行結果

実行結果を図 2 に示す．box の最大区間幅は 0.5 とした．実装は C++言語を用い，BranchAndPrune アルゴリズムの実装としてライブラリ RealPaver(ver. 1.0)[6] を使用した．

現在は数式処理による手法との統合処理系を開発中である．実装には同じく C++言語を用い，データ構造やいくつかの処理を共通化することで実行中に動的に手法を切り替えられることを目指している．

5 まとめ

本論文では，HydLa で記述された不確実性を持つモデルの解全体を box 集合で包囲する処理系を提案し，確認のための実装をおこなった．現在実装中の統合処理系を用いて，検証への具体的手法を検討することは今後の課題となる．

謝辞 本研究の一部は，科学研究費補助金（基盤研究（B）20300013）の補助を得ておこなった．

参考文献

[1] Lunze, J., Lamnabhi-Lagarrigue, F., Handbook of Hybrid Systems Control: Theory, Tools, Applications. Cambridge University Press, 2009.

[2] 上田和紀，石井大輔，細部博史，制約概念に基づくハイブリッドシステムモデリング言語 HydLa，第五回システム検証の科学技術シンポジウム，pp. 6–11, 2008.

[3] 廣瀬賢一, 大谷順司, 石井大輔, 細部博史, 上田和紀, 制約概念に基づくハイブリッドシステムモデリング言語 HydLa の実装, 第 11 回プログラミングおよびプログラミング言語ワークショップ, C3, 2009.

[4] Borning, A., Feldman-Benson, B., Wilson, M., Constraint Hierarchies. *Lisp and Symbolic Computation*, Vol. 5, No. 3, pp. 223–270, 1992.

[5] Van Hentenryck, P., Mcallester, D., Kapur, D., Solving Polynomial Systems Using a Branch and Prune Approach. In *SIAM Journal on Numerical Analysis*, Vol. 34(2), pp. 797–827, 1997.

[6] Granvilliers, L., Benhamou, F., Algorithm 852: Realpaver: An Interval Solver using Constraint Satisfaction Techniques. *ACM Trans. on Mathematical Software*, Vol. 32, pp. 138–156, 2006.