

## ハイブリッドシステムの高信頼シミュレーションのための区間に基づく制約伝播手法

石井大輔<sup>†1</sup> 上田和紀<sup>†1</sup> 細部博史<sup>†2</sup>

離散変化と時間に関する連続変化からなるハイブリッドシステムは、物理学をはじめ、さまざまな分野の問題を記述、シミュレーションするためのモデルとして注目を集めている。ハイブリッドシステムのシミュレーションにおいては、連続状態を扱う際に計算誤差が不可避であり、厳密な解軌道を得ることができないという問題がある。本研究の提案手法では、常微分方程式の区間解析に基づく求解手法と、非線形問題のための区間制約伝播手法を用いた求解手法を統合することにより、ハイブリッド軌道の指定精度による完全な区間包囲を効率良く求めることを可能にした。また提案手法を実装するとともに、実験により提案手法の有効性を評価した。提案手法により、ハイブリッドシステムのシミュレーションを、解の精度を保証しながら高信頼に行うことが可能になる。

### An Interval-based Consistency Technique for Reliable Simulation of Hybrid Systems

DAISUKE ISHII,<sup>†1</sup> KAZUNORI UEDA<sup>†1</sup>  
and HIROSHI HOSOBÉ<sup>†2</sup>

Hybrid systems are systems consisting of discrete changes and continuous changes over time. Problems in various fields such as physics can be modeled as hybrid systems. In a simulation of a hybrid system, it is difficult to obtain rigorous solution of the model because of computation errors in the handling of continuous states. Our proposed method integrates interval-based techniques for initial value problems for ordinary differential equations and consistency techniques for nonlinear problems. The method obtains complete interval enclosures of hybrid trajectories efficiently with a given accuracy. We implemented the method and evaluated its effectiveness. Using the method, we can simulate hybrid systems reliably with guaranteeing its accuracy.

### 1. はじめに

ハイブリッドシステムは、離散変化と時間に関する連続変化とからなる系を表す概念である。例として、図1に示すような重力に従って自由落下し、床に達すると跳ね返る質点がある。ハイブリッドシステムは、物理学、制御工学、分子生物学などの分野のさまざまな問題のモデリングに用いることができる。

MATLAB/Simulink<sup>15)</sup> や Dymola<sup>6)</sup> などの商用システムをはじめとして、ハイブリッドシステムのシミュレーションを行うための枠組みが広く利用されている。しかしハイブリッドシステムの連続状態を扱う際には、浮動小数点演算や近似数値計算を用いなければならず、既存の多くのシステムでは厳密な解軌道を得ることができなかった<sup>2)</sup>。1990年代以降、ハイブリッドシステムの連続変化を有限状態遷移系に抽象化し、モデルチェッキング手法により厳密に可到達範囲を求める手法が発展している<sup>20)</sup>。しかしこれらの手法では、連続変化やガード条件が非線形方程式により表される場合など、扱える問題に限られていた。

本論文では、区間解析<sup>16)</sup>に基づきハイブリッドシステムのシミュレーションを行う手法を提案する。図1に提案手法による求解結果の例を示す。区間解析は精度保証付き数値計算とも呼ばれ、真解を計算誤差とともに区間に包みながら計算を進めていく手法である。我々は、ハイブリッドシステムを区間を用いてモデリング、シミュレーションすることが可能なハイブリッド並行制約プログラミング (Hybrid Concurrent Constraint Programming, 以下「Hybrid CC」とする)<sup>3),10)</sup>を拡張し、区間解析に基づき厳密にシミュレーションを行うための手法<sup>12),13)</sup>を示した。

本論文の貢献は、(1) 上記手法を拡張し、ハイブリッドシステムの区間に基づく解軌道を、制約伝播手法により指定した精度で効率良く求めるための手法を提案することにある。そのために、(a) 区間に基づく常微分方程式の求解手法<sup>18)</sup>を拡張し、連続変化の区間包囲を任意のステップ幅で逐次計算する方法を示すと同時に、(b) 非線形方程式・不等式のための制約伝播手法<sup>1),21)</sup>を拡張し、(a)の方法と連携して離散変化の判定および連続変化の逐次計算を行う方法を示す。さらに本論文では、(2) 提案アルゴリズムについて区間解が真解を包囲することを示すと同時に、(3) 提案アルゴリズムを用いた求解系を実装し、(4) 実装

<sup>†1</sup> 早稲田大学理工学術院

Faculty of Science and Engineering, Waseda University

<sup>†2</sup> 国立情報学研究所

National Institute of Informatics

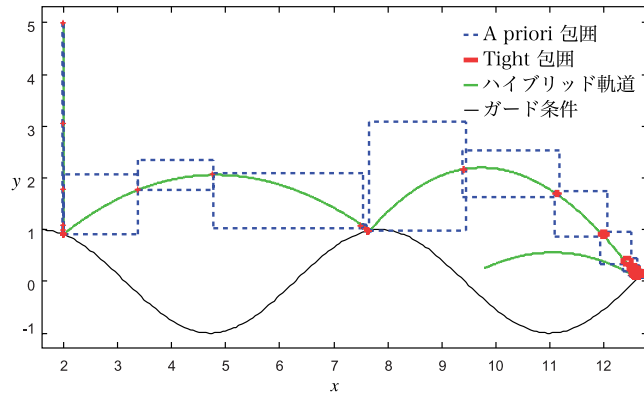


図 1 ハイブリッドシステムとその高信頼シミュレーション  
Fig. 1 Hybrid system and its dependable simulation.

を用いた実験により提案手法の有効性を評価する。

本論文の提案手法により、ハイブリッドシステムのシミュレーションを、ユーザが指定した許容誤差で解の精度を保証しながら、高信頼に行うことが可能になる。またシミュレーションの際には、解軌道の完全な区間包囲を効率良く求めることできる。

本論文の構成について述べる。まず 2 章で背景となる区間演算と常微分方程式の初期値問題について述べる。3 章では提案手法が対象とするモデルであるハイブリッドシステムについて述べる。4 章では提案手法について述べる。5 章で提案手法の実装について述べ、6 章で実装を用いた提案手法に関する実験結果を述べる。

## 2. 準備

本章では、提案手法について述べるにあたって必要となる諸概念について説明する。

### 2.1 区間演算

機械表現可能な浮動小数点数の集合を  $F$  で表す。区間  $I = [l, u]$  (ただし  $l, u \in F$ ) は以下のような実数の集合である：

$$I = \{r \in \mathbf{R} \mid l \leq r \leq u\}. \quad (1)$$

$I$  により区間の集合を表す。また box  $B$  は、区間の  $n$  組  $(I_1, \dots, I_n)$  である。ただし  $n \in \mathbf{N}$ ,  $I_1, \dots, I_n \in I$  とする。区間の例として  $[0.25, 0.5]$  と  $[1.0, 1.0]$ , box の例として  $([0.25, 0.5], [1.0, 1.0])$  があげられる。区間  $I$  について、下限を  $\text{lb}(I)$  で、

上限を  $\text{ub}(I)$  で、幅を  $w(I)$  で、内部を  $\text{int}(I)$  で表す。 $r \in \mathbf{R}$  について  $r$  の最近傍の浮動小数点数への下向き丸め値を  $\lfloor r \rfloor$  で、上向き丸め値を  $\lceil r \rceil$  で表す。 $\lceil r \rceil$  で  $\lfloor r \rfloor, \lceil r \rceil$  を表し、 $r_1, r_2 \in \mathbf{R}$  について  $[r_1, r_2]$  で  $\lfloor r_1 \rfloor, \lceil r_2 \rceil$  を表す。たとえば  $[0.1] = [0.0999999940395355224609375, 0.100000001490116119384765625]$  となる(端を単精度浮動小数点数とした場合)。 $R \subset \mathbf{R}$  について、 $R$  を包む区間  $I \in \mathbf{I}$  を  $\square R$  で表す。

関数  $f: \mathbf{R}^n \rightarrow \mathbf{R}$  について、以下を満たす  $F: \mathbf{I}^n \rightarrow \mathbf{I}$  を  $f$  の区間拡張という：

$$\forall I_1, \dots, I_n \in \mathbf{I} [\forall r_1 \in I_1, \dots, \forall r_n \in I_n (f(r_1, \dots, r_n) \in F(I_1, \dots, I_n))]. \quad (2)$$

また、制約  $c \subseteq \mathbf{R}^n$  について、以下を満たす  $C \subseteq \mathbf{I}^n$  を  $c$  の区間拡張という：

$$\forall I_1, \dots, I_n \in \mathbf{I} [\exists r_1 \in I_1, \dots, \exists r_n \in I_n ((r_1, \dots, r_n) \in c) \Rightarrow (I_1, \dots, I_n) \in C]. \quad (3)$$

たとえば  $f(x) = x - x$  の区間拡張として、 $F([l, u]) = [\lfloor l - u \rfloor, \lceil u - l \rceil]$  や  $F'([l, u]) = [0]$  などがある。基本演算の区間拡張を以下のように定義することができる ( $[l_1, u_1], [l_2, u_2] \in \mathbf{I}$ )：

$$\begin{aligned} [l_1, u_1] + [l_2, u_2] &= [\lfloor l_1 + l_2 \rfloor, \lceil u_1 + u_2 \rceil], \\ [l_1, u_1] - [l_2, u_2] &= [\lfloor l_1 - u_2 \rfloor, \lceil u_1 - l_2 \rceil], \\ [l_1, u_1] \cdot [l_2, u_2] &= [\lfloor \min\{l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2\} \rfloor, \lceil \max\{l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2\} \rceil], \\ [l_1, u_1] / [l_2, u_2] &= [l_1, u_1] \cdot [\lfloor 1/u_2 \rfloor, \lceil 1/l_2 \rceil], \quad 0 \notin [l_2, u_2]. \end{aligned} \quad (4)$$

たとえば  $I_1 = [0.75, 1.5]$ ,  $I_2 = [0.5, 0.75]$  とすると、上記基本演算の区間拡張により以下が成り立つ： $I_1 + I_2 = [1.25, 2.25]$ ,  $I_1 - I_2 = [0, 1]$ ,  $I_1 \cdot I_2 = [0.375, 1.125]$ ,  $I_1 / I_2 = [1, 3]$ 。

### 2.2 微分方程式の初期値問題とその区間拡張

以下のような微分方程式の初期値問題 (initial value problems for ordinary differential equations, 以下「IVP-ODE」と表記する) を考える：

$$\phi' = f(t, \phi), \quad \phi(t_0) = x_0. \quad (5)$$

ただし、 $\phi$  は未知関数  $\mathbf{R} \rightarrow \mathbf{R}^n$  とし、 $t, t_0 \in \mathbf{R}$ ,  $x_0 \in \mathbf{R}^n$ ,  $f: \mathbf{R} \times \mathbf{R}^n \rightarrow \mathbf{R}^n$  とする ( $f$  をベクトル場と呼び、その集合を  $\mathcal{VF}$  と表す)。

IVP-ODE の解を連続軌道と呼ぶ。連続軌道は関数  $\phi(t): \mathbf{R} \rightarrow \mathbf{R}^n$ , あるいは  $t_0$  での初期値  $x_0$  を始域に含め、関数  $\phi^*(t_0, x_0, t): \mathbf{R} \times \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n$  で表すことができる。

$\Phi: \mathbf{R} \rightarrow \mathbf{I}^n$ ,  $F: \mathbf{R} \times \mathbf{I}^n \rightarrow \mathbf{I}^n$ ,  $X_0 \in \mathbf{I}^n$ ,  $t, t_0 \in \mathbf{R}$  とするとき、上記の IVP-ODE に対して以下のような区間拡張を与える：

$$\Phi' = F(t, \Phi), \quad \Phi(t_0) = X_0. \quad (6)$$

ただし  $\Phi$  は  $\phi$  の区間拡張、 $F(\bullet, \Phi)$  は  $f(\bullet, \phi)$  の区間拡張とし ( $F$  をベクトル場の区間拡張と呼び、その集合を  $\mathcal{VF}$  で表す)、以下のような区間解を持つとする ( $\Phi$  を連続軌道の区間包囲ともいう)。区間解は関数  $\Phi(t): \mathbf{R} \rightarrow \mathbf{I}^n$  あるいは  $\Phi^*(t_0, X_0, t): \mathbf{R} \times \mathbf{I}^n \times \mathbf{R} \rightarrow \mathbf{I}^n$

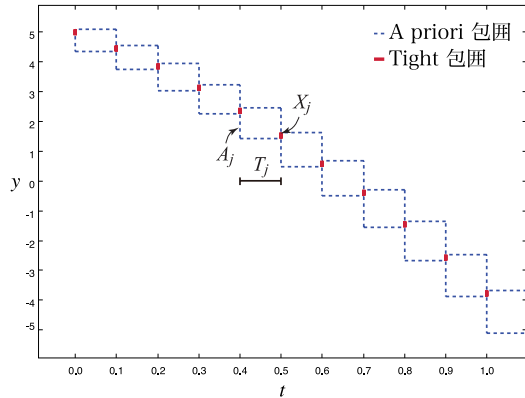


図 2 連続軌道の区間包囲

Fig. 2 An interval enclosure of a continuous trajectory.

である。ただし以下が成り立つ：

$$\forall x_0 \in X_0 (\phi^*(t_0, x_0, t) \in \Phi^*(t_0, X_0, t)). \quad (7)$$

区間解析に基づく IVP-ODE の求解手法は Moore<sup>16)</sup> をはじめとして現在まで研究が行われている<sup>4),5),14),18)</sup>。

例 1 係数  $k \in \mathbf{R}$  により空気抵抗を受けながら重力加速度  $g \in \mathbf{R}$  により落下する質点を、以下の IVP-ODE によりモデリングする ( $y \in \mathbf{R} \rightarrow \mathbf{R}^2, x \in \mathbf{R}^n$  について  $\langle x, n \rangle$  で  $x$  の第  $n$  成分を表す)：

$$y' = (\langle y(t), 2 \rangle, -g - k \langle y(t), 2 \rangle), \quad y(0) = (5, -5). \quad (8)$$

上記に対し、以下の区間拡張を考えることができる ( $G \in \mathbf{I}, g \in G, K \in \mathbf{I}, k \in K, Y : \mathbf{R} \rightarrow \mathbf{I}^2$ )：

$$Y' = (\langle Y(t), 2 \rangle, -G - K \langle Y(t), 2 \rangle), \quad Y(0) = ([4.9, 5.1], [-5]). \quad (9)$$

落下する質点の区間解の例を図 2 に示す。

図 2 の区間解は既存手法<sup>18)</sup> により求めた (手法については 4.1 節で述べる)。この手法では区間解を、a priori 包囲および tight 包囲と呼ばれる 2 種類の box の集合から構成する。box の集合の和は区間  $Y(0)$  中の任意の点を初期値とした問題の解軌道を包む。第  $j$  ステップの a priori 包囲  $A_j \in \mathbf{I} \times \mathbf{I}^n$  は、時間区間  $T_j = [t_{j-1}, t_j]$  上のすべての解を包囲する。すなわち  $\forall t \in T_j (\Phi^*(t_{j-1}, X_{j-1}, t) \subseteq A_j)$  が成り立つ。Tight 包囲  $X_j \in \mathbf{I}^n$  は時間  $t_j$  における解を包囲する。

### 3. ハイブリッドシステム

本章では、まずハイブリッドシステムの実数上における定式化を行ったあと、計算機で扱うために区間拡張を施したモデルについて述べる。

#### 3.1 実数上における定式化

ハイブリッドシステムは、状態が離散的に、あるいは時間に関して連続的に変化する系を表すモデルである。ハイブリッドシステムを、下記からなる組  $H = (Q, X, E, F, Init, G, R)$  として定義する：

- $Q$  は離散状態の集合、
- $X = \mathbf{R}^n$  は連続状態の集合、
- $E \subseteq Q \times Q$  は離散変化の集合、
- $F = \{f_q\}_{q \in Q}$  は各離散状態に関連づけられたベクトル場  $f_q \in VF$  の族、
- $Init \in Q \times X$  は初期状態、
- $G = \{g_e\}_{e \in E}$  はガード条件  $g_e \subseteq X$  の族、
- $R = \{r_e\}_{e \in E}$  はリセット写像  $r_e : X \rightarrow X$  の族。

例 2 図 1 に示すような正弦曲面上で跳ね返る質点をあげる。質点は 2 次元の座標  $(x, y)$  を持ち、各要素は速度  $x'$  および  $y'$  で連続変化する。質点は  $y$  方向に例 1 の IVP-ODE に基づく空気抵抗および重力を受ける。質点が落下する下には正弦曲面の床が広がっており、質点が床に達すると反発を受けて跳ね返る。質点のモデル  $H_{BB} = (Q_{BB}, X_{BB}, E_{BB}, F_{BB}, Init_{BB}, G_{BB}, R_{BB})$  の各要素を以下のように定義する：

$$\begin{aligned} Q_{BB} &= \{falling\}, & X_{BB} &= (x, x', y, y') \in \mathbf{R}^4, \\ E_{BB} &= \{bounce = (falling, falling)\}, \\ F_{BB} &= \{f_{falling} = (0, 0, y', -g - ky)\}, \\ Init_{BB} &= (falling, (2, 0, 5, -5)), \\ G_{BB} &= \{g_{bounce} = \{(x, x', y, y') \in \mathbf{R}^4 | y = \sin(x)\}\}, \\ R_{BB} &= \left\{ r_{bounce} = (x, x', y, y') \mapsto \left( x, \frac{(1 - e \cos(x)^2)x' + (1 + e) \cos(x)y'}{1 + \cos(x)^2}, y, \frac{(1 + e) \cos(x)x' + (-e + \cos(x)^2)y'}{1 + \cos(x)^2} \right) \right\}. \end{aligned} \quad (10)$$

ハイブリッドシステムの実行をハイブリッド軌道により表す。まず、時間区間  $T_j \subseteq \mathbf{R}$

の列  $\tau = \{T_j\}_{j \in \{1, \dots, n\}}$  を考える ( $n \in \mathbf{N}$ ). ただし  $T_j$  は  $[l_j, u_j)$  のような半开区間とし ( $l_j, u_j \in \mathbf{R}$ ),  $u_j = l_{j+1}$  が成り立つ. ハイブリッド軌道は, 上記  $\tau$  と, 各  $T_j$  の上で不変な離散状態の列,  $T_j$  の上で連続な関数の列から構成することができる. ハイブリッド軌道を組  $(q_0, x_0, \tau, q, \phi)$  により定義する. 各要素は, 初期離散状態  $q_0 \in Q$ , 初期連続状態  $x_0 \in X$ , 時間区間の列  $\tau$ , 離散状態の列  $q = \{q_j\}_{j \in \{1, \dots, n\}}$ ,  $q_j \in Q$ , 連続状態を与える写像 (連続軌道) の列  $\phi = \{\phi_j\}_{j \in \{1, \dots, n\}}$ ,  $\phi_j: T_j \rightarrow X$  からなる. またハイブリッド軌道は, すべての  $j \in \{1, \dots, n\}$  について以下を満たす:

- $q(T_0) = q_0$ ,
- $\phi_0(\text{lb}(T)_0) = x_0$ ,
- $(q_j, q_{j+1}) \in E$ ,
- $\phi_j$  はベクトル場を  $f_{q_j}$ , 初期値を  $r_{(q_{j-1}, q_j)}(\phi_{j-1}(\text{ub}(T)_{j-1}))$  とした IVP-ODE に関する連続軌道,
- $\phi_{j+1}(\text{lb}(T)_{j+1}) \in r_{(q_j, q_{j+1})}(\phi_j(\text{ub}(T)_j))$ ,
- $\phi_j(\text{ub}(T)_j) \in g_{(q_j, q_{j+1})}$ ,
- すべての  $t \in T_j$  について  $\phi_j(t) \notin g_{(q_j, q_{j+1})}$ .

例 2 のハイブリッドシステムの実行により得られるハイブリッド軌道を図 3 (a) に示した. ハイブリッド軌道は, 初期条件  $Init_{BB}$  とベクトル場  $f_{falling}$  に基づく連続変化と, ガード条件  $g_{bounce}$  とリセット写像  $r_{bounce}$  により質点の速度を瞬間的に更新する離散変化からなる. 各連続変化は時間軸上の区間  $T_1, T_2, T_3, \dots$  に対応し, 各離散変化は各区間の端点に対応する.

### 3.2 ハイブリッドシステムの区間拡張

本論文ではハイブリッドシステムの高信頼な実行のためにハイブリッドシステムの区間拡張およびハイブリッド軌道の区間包囲を導入する.

以下によりハイブリッドシステムの区間拡張を行う:

- $X$  を  $\mathcal{X} = I^n$  に置き換える,
- $F$  を  $\mathcal{F} = \{F_q\}_{q \in Q}$  に置き換える ( $F_q$  は  $f_q$  の区間拡張),
- $G$  を  $\mathcal{G} = \{G_e\}_{e \in E}$  に置き換える ( $G_e \in I^n$ ,  $g_e \in \mathcal{X}$ ),
- $R$  を  $\mathcal{R} = \{R_e\}_{e \in E}$  に置き換える ( $R_e: \mathcal{X} \rightarrow \mathcal{X}$  は  $r_e$  の区間拡張).

ハイブリッドシステム  $H$  の区間拡張を, 以下に述べるハイブリッド軌道の区間包囲を持つような組  $\mathcal{H} = (Q, \mathcal{X}, E, \mathcal{F}, Init, \mathcal{G}, \mathcal{R})$  と定義する.  $\text{ub}(T_j) \leq \text{lb}(T_{j+1})$  が成り立つ時間区間  $T_j \in I$  の列  $T = \{T_j\}_{j \in \{1, \dots, n\}}$  を考える ( $n \in \mathbf{N}$ ). ハイブリッド軌道の区間包

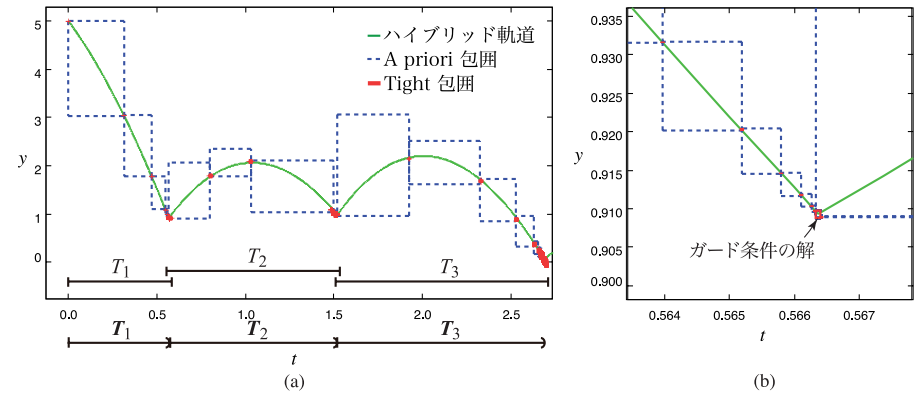


図 3 ハイブリッド軌道の例: (a)  $T_1, T_2, T_3$  上のハイブリッド軌道とその区間包囲, (b)  $j = 1$  におけるガード条件  $G_{BB}$  の解

Fig. 3 An example of a hybrid trajectory: (a) A hybrid trajectory and an interval enclosure over  $T_1, T_2, T_3$ , (b) A solution of a guard condition  $G_{BB}$  at  $j = 1$ .

を組  $(q_0, X_0, T, q, \phi)$  により表す. ハイブリッド軌道の区間包囲はハイブリッド軌道の第 2 要素を  $X_0 \in I^n$  に, 第 3 要素を  $T$  に, 第 5 要素を連続状態を与える写像 (連続軌道) の列  $\Phi = \{\Phi_j\}_{j \in \{1, \dots, n\}}$ ,  $\Phi_j: T_j \rightarrow \mathcal{X}$  に置き換えたものである. すべての  $x_0 \in X_0$  を初期状態とするハイブリッド軌道  $(q_0, x_0, \tau, q, \phi)$  に対し, 区間包囲  $(q_0, X_0, T, q, \Phi)$  はすべての  $j \in \{1, \dots, n\}$  について以下を満たす ( $n \in \mathbf{N}$ ):

- すべての  $T_j \in \tau$ ,  $T_j \in \tau$  について  $T_j \subseteq T_j$ ,
- すべての  $T_j \in \tau$ ,  $t \in T_j$ ,  $x_0 \in X$  について,  $\phi_j(t) \in \Phi_j(t)$  となる  $T_j \in T$ ,  $T \in T_j$ ,  $X_0 \in \mathcal{X}$  が存在する.

例 2 のハイブリッドシステムに区間拡張を施し, 実行することで得られるハイブリッド軌道の区間包囲を図 1 に示した. ハイブリッドシステムの区間拡張では, 連続状態を box の集合で包み, ベクトル場, ガード条件およびリセット写像を区間拡張する. 図 3 に例 2 の区間拡張を実行し得られたハイブリッド軌道の区間包囲を示す. 図中の (a) は  $j = 1, 2, 3$  に対応する連続変化の区間包囲を示している. (b) は  $j = 1$  の連続変化に対するガード条件の解の区間包囲を示している. 区間拡張されたモデルを実行する際には, ハイブリッド軌道中の連続変化については連続軌道の区間包囲を求め, 離散変化については, ガード条件およびリセット写像の評価を区間演算に基づき行う. これにより, 理論解を包む box の集合であるハイブリッドシステムの区間拡張を得る.

ハイブリッドシステムの区間包囲ではガード条件の判定が非決定的になる場合が生じる<sup>13)</sup>。本論文では非決定的な問題については考慮しないが、それらの問題にも提案手法は適用可能である。

#### 4. ハイブリッド軌道の区間包囲の求解手法

本章では、ハイブリッド軌道の区間包囲を求めるための手法を述べる。提案手法の概要を以下に示す：

- (1)  $q_j \in Q, j \in \{1, \dots, n\}$  について、初期状態  $X_{j-1}$  として、 $X_0$  を ( $j = 1$  の場合)、または  $R_{(q_{j-1}, q_j)}(X'_{j-1})$  を ( $j > 1$  の場合、 $X'_{j-1}$  は前フェーズで求めた状態) 考える。

例 2 において初期状態は、 $j = 1$  のときはモデル  $H_{BB}$  が与える  $X_0 = (2, 0, 5, -5)$ 、 $j = 2$  のときは図 3 (b) に示したガード条件の解にリセット写像  $R_{BB}$  を適用した値 ( $X_1 = ([2.00], [5.95], [0.91], [4.98])$ 、小数点以下第 3 位で四捨五入している) となる。

- (2) ベクトル場を  $F_{q_j}$ 、初期値を  $X_{j-1}$  とした IVP-ODE の区間拡張に基づき、 $T_j \in T$  上の連続軌道の区間包囲  $\Phi_j$  を求める。

例 2 では  $j = 1$  のとき、例 1 で述べた IVP-ODE の区間拡張を求解することに該当する。

- (3)  $\Phi_j$  とガード条件  $G_{q_j}$  を連立し離散変化を起こす状態  $X_j$  を求める。(1) に戻る。

例 2 では  $\Phi_j$  とガード条件  $G_{q_j} : \langle X_j, 3 \rangle = \sin(\langle X_j, 1 \rangle)$  を連立、求解する。

本論文の提案手法は、おもに上記の (2) と (3) の処理を、制約伝播に基づき効率良く行うことを目指したものである。

4.1 節では、IVP-ODE の求解手法<sup>18)</sup> を拡張した、連続軌道を任意のステップ幅で逐次計算する方法について述べる。

4.2 節では、Benhamou ら<sup>1),21)</sup> による区間領域の非線形制約を制約伝播に基づき求解する手法について述べる。この手法により、非線形のガード条件の判定が可能になる。代表的な BRANCHANDPRUNE アルゴリズムを示す。

4.3 節では、4.1 節と 4.2 節の方法を統合することにより、ハイブリッド軌道を求める枠組みを示す。とくに、4.1 節の方法を組み込むことで 4.2 節の枠組みを拡張し、IVP-ODE 求解と非線形制約の求解を基本操作とした BRANCHANDPRUNE アルゴリズムを実現する。このアプローチでは、連続変化の逐次計算回数をなるべく抑えながら、離散変化を引き起こすガード条件の解を高精度に求めることを目標にしている。

#### 4.1 連続軌道の求解

提案手法では、IVP-ODE の区間拡張の求解を行うことにより、ハイブリッドシステム中の連続軌道の区間包囲を求める。本研究では Nedialkov ら<sup>18)</sup> による既存手法を利用した (以下、この手法を「VNODE」とする)。本節では、VNODE の概要と提案手法のために追加した処理について述べる。

VNODE では、以下の計算を繰り返すことにより逐次的に連続軌道の区間包囲を計算する (ここでは  $j$  回目の計算を述べる)：

- (1) 初期時間  $t_{j-1} \in I$ 、初期値  $X_{j-1} \in I^n$ 、ベクトル場  $F \in \mathcal{VF}$ 、ステップ幅の見積り値  $h_{trial,j}$  ( $h_{trial,j} > 0$  とする) が入力される。 $h_{trial,j}$  をもとにステップ幅  $h_j$  を求め、a priori 包囲  $A_j$  を計算する。
- (2)  $A_j$  を用いて時点  $t_j$  における tight 包囲  $X_j$  を計算する。

上記 (1) においてステップ幅  $h_j$  を求める際には、Picard-Lindelöf の反復式に基づく PICARD 演算子を用いる。区間拡張のもととなる IVP-ODE

$$\phi' = f(t, \phi), \quad \phi(t_{j-1}) = x_{j-1} \in \text{int}(A'_j) \quad (11)$$

に対し ( $A'_j \in \mathbf{R}^n$ )、

$$\text{PICARD}(A'_j) = \Phi(t_{j-1}) + [0, h_j]F'([t_{j-1}, t_{j-1} + h_j], A'_j) \quad (12)$$

を計算する ( $F'$  は  $F$  の第 1 変数に関する区間拡張。  $h_j \in \mathbf{R}$  は適当に与える)。このとき  $\text{PICARD}(A'_j) \subseteq A'_j$  が成り立つならば、すべての  $t \in t_{j-1} + [0, 1]h_j$  および  $x_{j-1} \in A'_j$  について、解  $\phi(t_{j-1}, x_{j-1}, t) \in A'_j$  が唯一存在する (VNODE では  $f$  の高次のテイラー級数を用いて改良した PICARD 演算子を用いる)。このような検証に成功すれば、a priori 包囲を  $A_j = h_j \cdot A'_j$  としてよい。例 1 において下記のように PICARD 演算子を計算し、a priori 包囲  $A = (0.5, [-1, 6], [-10, -5])$  を得ることができる：

$$[4.9, 5.1] + [0, 0.5] \cdot [-10, -5] = [-0.1000003814697265625, 5.1000003814697265625] \subseteq [-1, 6], \quad (13)$$

$$[-5] + [0, 0.5] \cdot (-[9.8] - [0.3] \cdot [-10, -5]) = [-9.150000572204589843750, -5] \subseteq [-10, -5]. \quad (14)$$

ハイブリッド軌道の区間包囲の計算では、初期時間を幅を持った区間として与える必要があるが、VNODE ではこれが可能である。また提案手法では、4.3 節で説明する離散変化が起こる時点の探索のために、ステップ幅を任意に指定しながら逐次計算を行う機能が必要となる。VNODE では、tight 包囲については任意の時点のものを求めることができるものの、a priori 包囲については、できるだけ大きなステップ幅を自動算出する方針としており、

幅を指定することができない。

提案手法では、離散変化が起こる時点を探索するため、VNODE にステップ幅の上限  $h_{max,j} \in \mathbf{R}$  ( $h_{max,j} > 0$ ) を指定する処理を追加した。上記 (1) の処理を以下のように行う。

- (1) 上述の入力に加え、 $h_{max,j}$  が入力される。
- (2)  $h_{trial,j}$  と  $f$  のテイラー級数展開を利用して  $A'_j$  を求める。
- (3)  $\text{PICARD}(A'_j) \subseteq A'_j$  の検証を行い、成功したら  $h_j$  とし、失敗したら終了する。
- (4)  $h_{trial,j} \geq h_{max,j}$  ならば  $h_{trial,j} = h_{max,j}$  とする。
- (5) 次ステップのための  $h_{trial,j+1}$  を計算する。
- (6)  $h_{trial,j+1} \geq h_{max,j}$  ならば  $h_{trial,j+1} = h_{max,j}$  とする。

(1), (4), (6) が本研究で追加した処理である。これらの処理が区間解の完全性を保つのは、上述の PICARD の性質より明らかである。

#### 4.2 制約伝播手法による区間非線形制約の求解

提案手法では、離散変化を起こすガード条件を判定するために既存手法 Newton<sup>1),21)</sup> を利用した。Newton は  $\mathbf{R}$  上の非線形制約の区間拡張  $\mathcal{C}$  および box  $\mathcal{D}_0$  を入力とし、 $\mathcal{D}_0$  から制約を満たすドメインを求めることができる。

Newton は、BRANCHANDPRUNE アルゴリズムを用い、制約の変数の値を制約伝播と呼ばれる計算により削減していく。制約伝播は、変数間で保たれる整合性 (consistency) を適用することにより行われる。

制約伝播手法を連続領域を表した区間を変数値とする問題に適用するために、box 整合<sup>1),21)</sup> が提案された。以下にその定義を示す。まず、 $\mathcal{C}$  が  $v_1, \dots, v_n$  を変数とする区間制約であるとき、 $\mathcal{C}$  の  $i$  番目の射影として以下を満たす組  $\langle \mathcal{C}, i \rangle$  を考える：

$$\langle \mathcal{C}, i \rangle = \{I_i \in \mathbf{I} \mid \exists I_1 \in \mathbf{I}, \dots, \exists I_n \in \mathbf{I} (I_1, \dots, I_n \in \mathcal{C})\}. \quad (15)$$

$\mathcal{C}$  の射影  $\langle \mathcal{C}, i \rangle$  が以下を満たすときかつそのときに限り、 $\langle \mathcal{C}, i \rangle$  は  $(I_1, \dots, I_n) \in \mathbf{I}^n$  に関して box 整合であるという：

$$I_i = \square\{r_i \in I_i \mid (I_1, \dots, I_{i-1}, [r_i], I_{r+1}, \dots, I_n) \in \mathcal{C}\}. \quad (16)$$

図 4 に BRANCHANDPRUNE アルゴリズムを示す。1 行目でアルゴリズムは入力 box を PRUNE 手続きを呼び出すことで更新する。PRUNE 手続きは、制約の各射影について整合性を適用し、box の削減を行う。その結果が空ならばアルゴリズムは空集合を返す (11 行目)。結果が空でなく、十分具体的であれば、結果として返す (4 行目)。それ以外の場合、box の次数  $i \in \{1, \dots, n\}$  を選択し (6 行目)、box の第  $i$  射影に対して分割を行う BRANCH 手続きを呼び出す (7 行目)。アルゴリズムは分割した box について再帰呼び出しを行う

**Input:** 制約の集合  $\mathcal{C}$ , 初期ドメイン  $\mathcal{D}_0$ , 許容誤差  $\epsilon$

**Output:** ドメインの集合  $\mathcal{D}_i$  ( $1 \leq i \leq n$ )

```

1:   $\mathcal{D} := \text{PRUNE}(\mathcal{C}, \mathcal{D}_0)$ 
2:  if  $\mathcal{D} \neq \emptyset$  then
3:    if  $\mathcal{D}$  の精度が  $\epsilon$  以下 then
4:      return  $\{\mathcal{D}\}$ 
5:    else
6:       $i :=$  成分を選択する
7:       $(\mathcal{D}_1, \mathcal{D}_2) := \text{BRANCH}(\mathcal{D}, i)$ 
8:      return  $\text{BRANCHANDPRUNE}(\mathcal{C}, \mathcal{D}_1) \cup \text{BRANCHANDPRUNE}(\mathcal{C}, \mathcal{D}_2)$ 
9:    end if
10:  else
11:    return  $\emptyset$ 
12:  end if

```

図 4 BRANCHANDPRUNE アルゴリズム  
Fig. 4 BRANCHANDPRUNE algorithm.

(8 行目)。

#### 4.3 制約伝播手法によるハイブリッド軌道の計算

本節では 4.1 節と 4.2 節の手法を統合し、BRANCHANDPRUNE によりハイブリッド軌道の計算を行う方法を示す。

提案手法を、下記を入力とした BRANCHANDPRUNE アルゴリズムとして構成する。ここではハイブリッド軌道の第  $j$  ステップを計算する場合について述べる。入力は、ガード条件  $\mathcal{G}$  とベクトル場  $F_{q_j}$  からなる組  $\mathcal{C} = (\mathcal{G}, F_{q_j})$  と、tight 包囲  $X_{j-1} \in \mathbf{X}$  と a priori 包囲の初期値  $A_j \in \mathbf{I} \times \mathbf{I}^n$  (最初は未定義でよい) からなる組  $\mathcal{D}_0 = (X_{j-1}, A_j)$  である。上記のもと、 $T_j \in \tau$  上の連続軌道をできるだけ大きなステップ幅で計算しながら、ガード条件を満たす指定した区間幅の状態を求める。制約の変数値となる a priori 包囲は時間を成分として含む  $(T, I_1, \dots, I_n) \in \mathbf{I} \times \mathbf{I}^n$  のような box である。以上の定式化により、4.2 節で述べた box 整合が適用可能になる。

図 5 と図 6 に提案アルゴリズム  $\text{PRUNE}_{\text{hybrid}}$  および  $\text{BRANCH}_{\text{hybrid}}$  の実装を示す。 $\text{PRUNE}_{\text{hybrid}}$  は入力された a priori 包囲の時間以外の成分について (2 行目)、Newton

**Input:** ガード制約の集合  $\mathcal{G}$ , a priori 包囲  $A_j \in I \times I^n$ , インデックス  $i \in \{1, \dots, n\}$

**Output:** a priori 包囲  $A'_j \in I \times I^n$

- 1:  $A'_j := A_j$
- 2: **if**  $i \neq 1$  **then**
- 3:      $A'_j := \text{PRUNE}_{\text{Newton}}(\mathcal{G}, A'_j, i)$
- 4:     **return**  $A'_j$
- 5: **else**
- 6:     **return**  $A'_j$
- 7: **end if**

図 5 PRUNE<sub>hybrid</sub> アルゴリズム  
Fig. 5 PRUNE<sub>hybrid</sub> algorithm.

**Input:** ベクトル場  $F_{q_j} \in \mathcal{VF}$ , tight 包囲  $X_{j-1}, X_j \in I^n$ , a priori 包囲  $A_j \in I \times I^n$  からなる 3 つ組  $(X_{j-1}, A_j, X_j)$

**Output:** tight 包囲  $X_{j-1}, X'_j, X''_j$  と a priori 包囲  $A'_j, A''_j$  からなる 3 つ組の組  $((X_{j-1}, A'_j, X'_j), (X'_j, A''_j, X''_j))$

- 1:  $T_j := \langle A_j, 1 \rangle$
- 2:  $(A'_j, X'_j, A''_j, X''_j) := \text{IVP-ODE } \Phi' = F_{q_j}$  (初期値  $\Phi(\text{lb}(T_j)) = X_{j-1}$ ) を, ステップ幅を  $w(T_j)/2$  として, 2 ステップ分求解する
- 3:  $X''_j := X_j \cap X'_j$
- 4: **return**  $((X_{j-1}, A'_j, X'_j), (X'_j, A''_j, X''_j))$

図 6 BRANCH<sub>hybrid</sub> アルゴリズム  
Fig. 6 BRANCH<sub>hybrid</sub> algorithm.

を用いてガード条件を適用することにより削減し (3 行目), その結果を返す. アルゴリズムはガード条件が成り立つかどうかを, 結果が空であるか確認することにより判定することができる. BRANCH<sub>hybrid</sub> は, 入力されたベクトル場  $F_{q_j}$  と状態  $X_{j-1}, A_j$  から IVP-ODE  $\Phi' = F_{q_j}(t, \Phi)$ ,  $\Phi(t_0) = X_{j-1}$  を構築し, ステップ幅を  $w(\langle A_j, 1 \rangle)/2$  に改良し, 解きなおすことにより, a priori 包囲  $A_j$  を 2 つに分割する (2 行目). BRANCHANDPRUNE アルゴリズムは, ガード条件を成り立たせる状態がなくなるか, a priori 包囲  $A_j$  が指定された精度に達した (すべての  $i \in \{1, \dots, n\}$  について  $w(\langle A_j, i \rangle)$  が指定した区間幅以下になった)

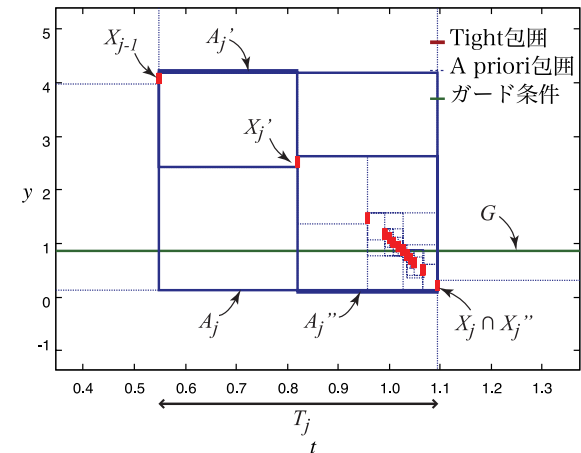


図 7 BRANCHANDPRUNE による計算  
Fig. 7 Computation by BRANCHANDPRUNE.

ならば, 停止する.

図 7 に提案手法により時間区間  $T_j \in T$  上の軌道を求める計算を示す. 提案する BRANCHANDPRUNE はガード条件  $G$  を含む集合  $\mathcal{G}$  とベクトル場  $F_{q_j} \in \mathcal{VF}$  を制約として, tight 包囲  $X_{j-1} \in I^n$  と a priori 包囲  $A_j \in I \times I^n$  を初期ドメインとして受け取る. まず, PRUNE<sub>hybrid</sub> が  $A_j$  中に  $G$  の解が含まれていることを確かめる. 確かめられたならば, BRANCH<sub>hybrid</sub> が,  $F_{q_j}$  と  $X_{j-1}$  から IVP-ODE を構成し  $T_j$  の 1/2 のステップ幅で解くことにより, tight 包囲  $X'_j, X''_j \in I^n$  と a priori 包囲  $A'_j, A''_j \in I \times I^n$  を得る. BRANCHANDPRUNE は  $A'_j$  と  $A''_j$  について同様の処理を再帰的に行う.

## 5. 実装

4.1 および 4.2 節で述べた VNODE と Newton は, それぞれ VNODE-LP<sup>17)</sup>, Elisa<sup>8),9)</sup> という C++ による実装が存在する. VNODE-LP と Elisa はともに制約求解のためのライブラリである. 区間非線形方程式および不等式求解のためのライブラリ Elisa では, 階層的なクラス群を用い, 制約の記述と, 求解方法の指定 (探索アルゴリズム, 制約伝播で用いる整合性など) を行うことができる.

本研究では Elisa の枠組みに独自のクラス群を実装, 追加することにより, 微分方程

式を記述し, VNODE-LP と連携して求解することを可能にした. まず IVP-ODE を表す ContConstraint クラスを実装した. ContConstraint クラスは, 時間を表す Time オブジェクト, 時間上の連続変数を表す ContVar オブジェクト, 常微分方程式を表す VNODE-LP のデータ構造をパラメータとする. ContConstraint, Time, ContVar は Elisa の基本クラス群を継承しているため, これらを用いて記述した連続制約に対し, ガード条件を Elisa のクラスを用いて記述し, 連立させることにより, ハイブリッドシステムを構築することができる.

本研究では独自の BRANCHANDPRUNE を実現するために, Elisa に対して 4.3 節で述べた PRUNE<sub>hybrid</sub> に対応する HybridReduction クラスと BRANCH<sub>hybrid</sub> に対応する HybridMakeNeighborhood クラスを追加した. これらのクラスは Elisa の基本クラス群を継承しており, Elisa の探索アルゴリズム実装を利用して求解を行うことができる. HybridMakeNeighborhood は呼び出しが行われると (関数オブジェクトになっている), ContConstraint が与える IVP-ODE を VNODE-LP を利用して以前の 1/2 のステップ幅で解きなおすことで解の探索の分岐を行う. HybridReduction は上記で IVP-ODE の求解して得られた a priori 包囲を伝播し, ドメインの削減を行う.

### 5.1 変数選択の方法

BRANCHANDPRUNE アルゴリズムの 6 行目では, ハイブリッドシステムの多次元の連続状態のある成分を表すインデックス  $i$  を選択している. アルゴリズムは連続状態を包む区間を選択した成分について分割する (成分が時間の場合 ( $i = 1$ ), HybridMakeNeighborhood を実行する). 例 2 では, 図 3 の時間軸で分割するか, 図 1 の  $x$  軸あるいは  $y$  軸で分割するかを選択する. 本研究の実装では変数選択は下記のように行われる:

- (1) 時間変数 Time の値が指定精度になるまでは Time を選択する.
- (2) Time が指定精度以下ならば, 他の変数を内部の格納順に選択していく (既存の SearchRoundRobin クラスと同様の処理).

現状の実装では上記 (1) により, IVP-ODE の計算を連続状態を 1 つの box で包みながら行っている. 連続状態を複数の box に区別して保持しながら IVP-ODE の計算を精度良く行う手法が提案されている<sup>5)</sup>. そのような手法を採用した場合, 上記の変数選択方法を変更するとともに, BRANCH の呼び出し回数を抑える必要がある.

## 6. 実験結果

提案手法の実験結果について述べる. 実験では, 3 章の例 2 で示した跳ね返る質点のモ

デルのシミュレーションを行った. また実験は以下のような環境で行った:

OS Windows XP SP2

CPU 2.4 GHz Core 2 Duo

メモリ 2 GB 667 MHz DDR2 SDRAM

### 6.1 床が平坦な場合

最初の実験では, 提案手法および可到達範囲検証ツールの 1 つである PHAVer<sup>7)</sup> によりハイブリッド解軌道の区間包囲の計算を行い, 時間計算量および解の精度について比較を行った.

PHAVer は, 多角形計算およびモデルチェック手法に基づきハイブリッドシステムの可到達範囲を計算することができる. 状態方程式としてアフィン系を, ガード条件として線形制約を入力として受け取り, 時間軸を抽象化した可到達範囲の多角形包囲を出力する.

本実験では, 以下の要素からなるモデル  $H_{FBB}$  のシミュレーションを行った:

$$\begin{aligned} Q_{FBB} &= \{falling\}, & X_{FBB} &= (y, y') \in \mathbf{R}^2, \\ E_{FBB} &= \{bounce = (falling, falling)\}, \\ F_{FBB} &= \{f_{falling} = (y', -g - ky')\}, \\ Init_{FBB} &= (falling, (5, 3)), \\ G_{FBB} &= \{g_{bounce} = \{(y, y') \in \mathbf{R}^2 \mid y = 0\}\}, \\ R_{FBB} &= \{r_{bounce} = (y, y') \mapsto (y, -ey')\}. \end{aligned} \quad (17)$$

このモデルでは, 例 2 のモデルについて, 状態を 2 次元にし, ガード条件およびリセット写像を平坦な床に対応するものに置き換えている. モデルの実行における最初の 3 回の落下および跳ね返りについて, ガード条件の解となる状態の区間包囲を求めた. 提案手法によるシミュレーションの処理は以下のように行った:

- (1) 初期条件をもとに提案手法 (BRANCHANDPRUNE) で次の衝突が起こる状態の区間包囲を求める.
- (2) (1) の結果をもとに離散変化の計算を区間演算により行い, 次の連続変化のための初期条件を求める. (1) へ.

提案手法では, Elisa の最大区間幅を 1 回目は  $10^{-9}$  でそれ以降は初期 box の幅 (各成分の平均値) に, VNODE-LP の最大区間幅を  $10^{-9}$  に設定した. PHAVer は, 多角形の分割を状態の第 1 および第 2 成分に沿って行い, 分割の最小幅が 0.2 および 0.8 となるように設定した. また USE\_CONVEX\_HULL パラメータに true を設定し, その他のパラメータはデフォルト設定とした. なお PHAVer では, 前述のマシン上で OS に Mac OS X 10.4.11 を使用



表 1 跳ね返る質点のシミュレーション結果 (床が平坦な場合)

Table 1 Simulation results of a bouncing particle on a flat ground surface.

跳ね返り回数	提案手法		PHAVer (0.2)		PHAVer (0.8)	
	実行時間	区間幅	実行時間	区間幅	実行時間	区間幅
1	0.46	$3.27 \cdot 10^{-9}$	78.1	0.190	17.6	0.756
2	0.78	$1.41 \cdot 10^{-7}$	2018.3	0.389	33.7	1.542
3	0.62	$2.93 \cdot 10^{-6}$	340.6	0.538	26.0	2.057
8	0.31	1.36	-	-	-	-

表 2 跳ね返る質点のシミュレーション結果 (床が正弦曲面の場合)

Table 2 Simulation results of a bouncing particle on a sinusoidal ground surface.

跳ね返り回数	実行時間	区間解	C&G による解
1	2.50	([2.00], [0.91])	(2.00, 0.91)
2	2.03	([7.67], [0.98])	(7.67, 0.98)
3	1.40	([12.65], [0.08])	(12.63, 0.06)
4	1.40	([9.47], [-0.05])	(9.39, 0.03)
5	1.25	([11.56], [-0.84])	(11.67, -0.79)
6	1.25	([10.02], [-0.56])	(9.90, -0.46)

した。

実験結果を表 1 に示す。表の各列は、各システムが求解に要した実行時間 (単位は 100 ミリ秒) と、得られた区間解の第 2 成分の幅 (衝突時の質点の速度) を示している。

提案手法により、PHAVer よりも高精度な解を高速に求めることができた。PHAVer の時間計算量が大きいのは、ハイブリッド軌道全体を最小幅で多角形分割するためであるが、最小幅の設定を大きくすると実行時間は短くなるものの、解の区間幅が大きくなる。

区間に基づく手法では計算にともなう区間幅の増大が不可避であるが、実験では離散変化の計算を行うごとに区間幅が指数関数的に増大することを確認した。

## 6.2 床が正弦曲面の場合

次の実験では、例 2 の跳ね返る質点のモデルについて提案手法により 6 回の落下と跳ね返りをシミュレーションする実験を行った。提案手法によるシミュレーションは前節と同様の方法で行った。本実験では、Hybrid CC の既存実装<sup>3)</sup> (以下「C&G」と呼ぶ) でもシミュレーションを行い、結果を比較した。なお、このモデルはガード条件が非線形方程式で与えられるため、PHAVer では扱うことができない。

実験結果を表 2 に示す。表の各列は、提案手法の実行時間、提案手法により得た質点の座標  $(x, y)$  の区間解、C&G により得た解を示している。各浮動小数点数は小数点以下第 3

位で四捨五入して表記した (区間の両端が等しくなれば点区間として表記した)。

提案手法が真解を包んだ区間を得るのに対し、C&G は異なった求解結果を得ている。C&G では IVP-ODE 求解に 4 次のルンゲクッタ法、離散変化判定に指定閾値 (本実験では  $10^{-9}$  とした) に基づいた方法などを用い、近似的に計算を行っているためである。また、前節での実験を含め、提案手法の実行時間が初期 box の大きさに依存しないことが分かった。

## 7. 関連研究

ハイブリッドシステムを扱う既存の枠組みは、シミュレーションのためのものと到達範囲の検証のためのものとに大別できる。サーベイとして文献 2) を参照されたい。厳密な求解を行うために、区間解析に基づくアプローチ、離散系に抽象化するアプローチ、数式処理により記号的に扱うアプローチなどが研究されている。

ハイブリッドシステムのシミュレーションを行う枠組みの多くは浮動小数点数や近似解法に基づいており、解の精度保証を可能とする研究は少ない。文献 19) では定数パラメータ切替えシステムに問題を限定し、VNODE および数式処理手法を利用してシミュレーションを厳密に行う手法を提案している。切替え時点の基本的な求解手法などを示しているものの、効率の良い実装方法や非線形のガード条件を扱う方法は示していなかった。

1990 年代以降、モデルチェック手法に基づきハイブリッドシステムの安全性などを厳密に検証するための手法がさかんに研究されている<sup>7),11)</sup> (サーベイとして文献 20) がある)。これらの手法では、ハイブリッドオートマトンで記述したモデルを検証したい性質を保つ範囲で離散遷移系に抽象化し、モデルチェック手法を適用する。非線形の連続軌道については、box や多角形による包囲で近似することにより扱う。文献 7) では連続軌道の到達範囲を多角形で包囲する手法を提案している。また文献 11) では、区間解析に基づき到達可能性検証を行う手法を提案している。

ハイブリッドシステムを簡潔に記述しシミュレーションを行うための Hybrid CC<sup>3),10)</sup> の枠組みが提案された。Hybrid CC の実装では区間制約の概念を用いており、実行が区間演算により計算される。しかし区間は記述性を高めるために導入されたもので、真解を厳密に包む実装になっていない。また、区間に基づきハイブリッド軌道を計算するのに必須である、幅を持った時間区間での衝突を扱うことができないという問題があった。

## 8. まとめ

本論文では、区間に基づく IVP-ODE 求解手法 VNODE および非線形問題の求解手法

Newton を BRANCHANDPRUNE アルゴリズムに組み込み、ハイブリッド軌道の区間包囲を効率良く求める方法を示した。

今後は、より実際的な問題に対して提案手法がどの程度有用か検討するとともに、たとえば離散変化の逆向き求解などによるより高度な PRUNE 処理や、数式処理と組み合わせる方法などを開発し、実用的な枠組みに発展させていきたいと考えている。

謝辞 各査読者からのコメントは論文の改善に非常に有用であった。感謝の意を表す。本研究は、早稲田大学上田研究室描画班を卒業した大野善之、西澤亮太、西村光弘の各氏、および現在在籍中の大谷順司、笹嶋唯、露崎浩太、廣瀬賢一各氏との関連諸技術に関する討論に支えられながら進めてきた。本研究の一部は、早稲田大学特定課題研究助成費（2007A-876 および 2007B-127）の補助を得て行った。

### 参 考 文 献

- 1) Benhamou, F., McAllester, D. and Van Hentenryck, P.: CLP (intervals) revisited, *Proc. 1994 International Symposium on Logic programming*, pp.124–138, MIT Press (1994).
- 2) Carloni, L.P., Passerone, R., Pinto, A. and Angiovanni-Vincentelli, A.L.: Languages and tools for hybrid systems design, *Foundations and Trends in Electronic Design Automation*, Vol.1, No.1/2, pp.1–193, Now Publishers (2006).
- 3) Carlson, B. and Gupta, V.: Hybrid cc with Interval Constraints, *Proc. HSCC 1998*, LNCS, Vol.1386, pp.80–95, Springer (1998).
- 4) Cruz, J. and Barahona, P.: Constraint satisfaction differential problems, *Proc. CP 2003*, LNCS, Vol.2833, pp.259–273, Springer (2003).
- 5) Deville, Y., Janssen, M. and Van Hentenryck, P.: Consistency techniques in ordinary differential equations, *Constraints*, Vol.7, No.3/4, pp.289–316, Kluwer (2002).
- 6) Dynasim AB: Dymola.
- 7) Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech, *Proc. HSCC 2005*, LNCS, Vol.3414, pp.258–273, Springer (2005).
- 8) Granvilliers, L. and Benhamou, F.: RealPaver: An interval solver using constraint satisfaction techniques, *ACM Trans. Mathematical Software*, Vol.32, No.1, pp.138–156, ACM Press (2006).
- 9) Granvilliers, L. and Sorin, V.: Elisa (2005). <http://sourceforge.net/projects/elisa>
- 10) Gupta, V., Jagadeesan, R., Saraswat, V.A. and Bobrow, D.: Programming in hybrid constraint languages, *Hybrid Systems II*, LNCS, Vol.999, pp.226–251, Springer (1995).
- 11) Henzinger, T.A., Horowitz, B., Majumdar, R. and Wong-Toi, H.: Beyond HyTech:

Hybrid systems analysis using interval numerical methods, *Proc. HSCC 2000*, LNCS, Vol.1790, pp.130–144, Springer (2000).

- 12) Ishii, D., Ueda, K. and Hosobe, H.: An interval-based approximation method for discrete changes in Hybrid cc, *Trends in Constraint Programming*, pp.245–255, ISTE (2007).
- 13) 石井大輔, 上田和紀, 細部博史: ハイブリッド並行制約プログラミングにおける分岐を含む軌道の区間包囲の求解手法, 日本ソフトウェア科学会第 24 回大会論文集 (CD-ROM), 6C-1 (2007).
- 14) Lohner, R.J.: Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems, *Computational Ordinary Differential Equations*, pp.425–435, Oxford University Press (1992).
- 15) MathWorks, T.: MATLAB/Simulink (2008).
- 16) Moore, R.E.: *Methods and applications of interval analysis*, Studies in Applied Mathematics, Vol.2, SIAM (1979).
- 17) Nediaklov, N.S.: VNODE-LP: A validated solver for initial value problems in ordinary differential equations, Technical Report CAS-06-06-NN, McMaster University (2006).
- 18) Nediaklov, N.S., Jackson, K.R. and Corliss, G.F.: Validated solutions of initial value problems for ordinary differential equations, *Applied Mathematics and Computation*, Vol.105, No.1, pp.21–68, Elsevier (1999).
- 19) Nediaklov, N.S. and von Mohrenschildt, M.: Rigorous simulation of hybrid dynamic systems with symbolic and interval methods, *Proc. 2002 American Control Conference*, Vol.1, No.1, pp.140–147 (2002).
- 20) Silva, B.I., Stursberg, O., Krogh, B.H. and Engell, S.: An assessment of the current status of algorithmic approaches to the verification of hybrid systems, *Proc. 40th IEEE Conference on Decision and Control*, Vol.3, No.3, pp.2867–2874 (2001).
- 21) Van Hentenryck, P., McAllester, D. and Kapur, D.: Solving polynomial systems using a branch and prune approach, *SIAM Journal on Numerical Analysis*, Vol.34, No.2, pp.797–827 (1997).

(平成 20 年 2 月 6 日受付)

(平成 20 年 3 月 26 日再受付)

(平成 20 年 4 月 29 日採録)



石井 大輔（学生会員）

2004年早稲田大学大学院理工学研究科修士課程修了。同年（株）東芝に入社。現在、早稲田大学大学院理工学研究科博士課程に在学中。また、同大学基幹理工学部助手を兼任。区間解析および制約プログラミングを用いたハイブリッドシステムのシミュレーション・検証に興味を持つ。



上田 和紀（正会員）

1978年東京大学工学部計数工学科卒業。1986年同大学大学院情報工学専攻博士課程修了，工学博士。1983年NEC入社。1985～1992年（財）新世代コンピュータ技術開発機構出向。1993年より早稲田大学理工学部情報学科。現在，同理工学術院情報理工学科教授。専門はプログラミング言語，並行・並列処理等。第7回日本IBM科学賞等受賞。情報処理学会論文誌：プログラミング初代編集委員長，日本ソフトウェア科学会編集委員長等歴任。



細部 博史（正会員）

1993年東京大学理学部情報科学科卒業。1995年同大学大学院理学系研究科情報科学専攻修士課程修了。1998年同専攻博士課程修了。博士（理学）。文部省学術情報センター助手，国立情報学研究所助手等を経て，現在，国立情報学研究所准教授。2005年仏ナント大学計算機科学研究所客員教員。制約プログラミング，ユーザインタフェース，情報可視化等に興味を持つ。2003年日本ソフトウェア科学会高橋奨励賞受賞。