社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

平行体計算を用いた非線形ハイブリッドシステムのシミュレーション

石井　大輔†　　Alexandre GOLDSZTEJN††

† 東京工業大学, 〒 152-8552 東京都目黒区大岡山 2-12-1-W8-67
†† CNRS, LINA (UMR-6241), Nantes, France
E-mail: †dsksh@acm.org, ††alexandre.goldsztejn@gmail.com

あらまし　非線形常微分方程式，非線型ガード制約，非線型リセット関数で構成されるハイブリッドシステムを区間計算にもとづき高信頼にシミュレーションする手法を提案する．提案手法は，平行体計算を用いて区間シミュレーションにおけるラッピング効果を抑え，ハイブリッドシステムの軌道の精度のよい区間包囲を求めることができる．複数の例題について，提案手法の実装を用いてシミュレーションを実施した実験結果を示す．
キーワード　区間制約プログラミング，射影手法，ハイブリッドシステム，平行体計算

Simulation of Nonlinear Hybrid Systems using a Parallelotope Method

Daisuke ISHII† and Alexandre GOLDSZTEJN††

† Tokyo Institute of Technology, 2-12-1-W8-67 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
†† CNRS, LINA (UMR-6241), Nantes, France
E-mail: †dsksh@acm.org, ††alexandre.goldsztejn@gmail.com

Abstract　We propose an interval-based method for reliable simulation of hybrid systems that consist of nonlinear ODEs, nonlinear guard constraints, and nonlinear reset functions. Our proposed method precisely encloses trajectories of a hybrid system by suppressing the wrapping effect in interval-based simulation with a parallelotope computation. We report experimental results in which we have simulated several examples with a tool that implements the proposed method.
Key words　interval constraint programming, projection methods, hybrid systems, parallelotope methods

1. はじめに

ハイブリッドシステム (2. 節) は状態が連続変化と離散変化の振る舞いをするシステムである．制御工学，計算生物学，物理学等の問題を直截に記述するためのモデルとして用いられている．ハイブリッドシステムのシミュレーションや検証では連続変化を記述した常微分方程式と離散変化の判定条件を連立して解く必要があるが，高信頼な計算方法は未だ確立されていない．数値計算に基づく方法では，計算誤差が深刻な問題を引き起こす場合があることが知られている [1].

本研究では，区間計算 (1. 2 節) 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．近年，区間シミュレーションにもとづくハイブリッドシステムの形式的検証手法が活発に研究されており [2]〜[6]，高精度かつ高速な区間シミュレーションの必要性が高まっている．

一般に区間計算では，ラッピング効果と呼ばれる box (区間ベクトル) の回転等による区間幅の増大を抑えることが重要である．ハイブリッドシステムの区間シミュレーションにおいては，常微分方程式の求解，離散変化の判定・適用等，複数の計算を連携して行う必要がある．既存手法 [7], [8] では前記計算を個別に行い，各計算結果の受け渡しは box で行っており，ラッピング効果の抑制を考慮していなかったため，問題によっては容易に計算結果の区間が爆発してしまっていた．そこで提案手法では，box に線形変換を施すことで得られる平行体 (3. 節) を導入し，ハイブリッドシステムの連続状態を平行体で包囲することにより，ハイブリッドシステム固有のラッピング効果を抑制し，高精度にシミュレーションを実施可能なアルゴリズムを提案する (4. 節).

5. 節では，提案手法の実装について述べる．6. 節では，ラッピング効果をはらむ線形および非線形のハイブリッドシステムの例題を示すとともに，提案手法を用いてラッピング効果を抑え，高精度にシミュレーションができることを示す.

1.1 関連研究

前記の通り，既存手法 [7], [8] では，離散変化の前後の状態を包囲する box を明示的に計算するため，ラッピング効果が発生する問題がある (6. 節において例を示す)．また，離散変化前の

連続状態をサポート関数や多角形等の近似手法を用いて精度よく包囲する手法 [9], [10] が提案されているが，本稿の手法との比較は今後の課題とする．

提案手法は，上記手法が離散変化が生起する時間区間でシミュレーションの計算を区切るのと異なり，ハイブリッドシステムの軌道を途中で離散変化を含むような 1 つの写像として考える点を特徴とする．このような形式化は [11] においても (数値シミュレーション手法の形式化として) なされている．

区間計算やその他の近似手法を用いてハイブリッドシステムの到達可能性解析を行う手法 [3], [12], [13] が提案されている．これらの手法は，連続状態の集合を時間軸を除いて計算する点がシミュレーション手法と異なる．モデルに時間変数を含めることで，シミュレーションと同様の計算が可能だが，非線形モデルを扱う場合，計算量が膨大になり非現実的である．

1.2 区間計算とその表記

区間計算の詳細については文献 [14] 等を参照されたい．区間 $[x] = [\underline{x}, \overline{x}]$ により集合 $\{r \in \mathbb{R} \mid l \leqq r \leqq u\}$ を表す．区間の集合を $\mathbb{I}$ で表す．区間 $[x]$ の下限と上限を $\underline{x}$ と $\overline{x}$ で表し，中心値を $\mathrm{mid}\,[x]$ で表す．集合 $S \subset \mathbb{R}$ について，$\square S$ でその区間包囲を表す．本稿では $\mathbf{x}$ と $[\mathbf{x}]$ のように，太字で実ベクトルと区間ベクトル (*box*) を表す．また，区間から区間への関数を $[f](\cdot)$，区間ベクトルから区間ベクトルへの関数を $[\mathbf{f}](\cdot)$ のように表す．

2. ハイブリッドシステム

本研究では，基本的かつ汎用的なハイブリッドシステムのモデルとして，ハイブリッドオートマトンから非決定性を取り除いたものを考える．

ハイブリッドシステム *HS* は以下の構成要素からなる:

- 各時刻における状態 ($\in \mathbb{R}^n$) を表す**変数** $\mathbf{x}$．時刻 t における状態を $\mathbf{x}(t)$ または $\mathbf{x}_-(t)$ で表す (後者は離散変化直前の値を表すのに用いる)．
- **ロケーション**の有限集合 $\mathcal{L} = (l_1, \ldots, l_p)$．
- ロケーションに紐づけられた**常微分方程式** (*ODE*) の族 $\{\dot{\mathbf{x}}(t) = \mathbf{f}_l(\mathbf{x}(t), t)\}_{l \in \mathcal{L}}$．ロケーション l における $\mathbf{x}(0)$ の値を $\tilde{\mathbf{x}}$ とした初期値問題の解を関数 $\phi_l(\tilde{\mathbf{x}}, t) : \mathbb{R}^n \times [0, t_{max}] \to \mathbb{R}^n$ で表す (ここでは，ハイブリッドシステム実行のグローバルな時間軸とは別に，ロケーション到達時を時刻 0 と考える．また本稿では各 ODE について一意な解の存在を仮定する)．
- **ガード制約**の族

$\{grd_{l,l'}(\mathbf{x}) \equiv h_{l,l'}(\mathbf{x}) = 0 \wedge g_{l,l'}(\mathbf{x}) \leqq 0\}_{(l,l') \in \mathcal{L} \times \mathcal{L}}$.
ただし，$h_{l,l'}(\cdot), g_{l,l'}(\cdot) : \mathbb{R}^n \to \mathbb{R}$.

- **リセット関数**の族 $\{\sigma_{l,l'} : \mathbb{R}^n \to \mathbb{R}^n\}_{(l,l') \in \mathcal{L} \times \mathcal{L}}$.
- **初期条件** $(l_0, [\mathbf{x}(0)]) \in \mathcal{L} \times \mathbb{I}^n$.

本稿では，簡単のため，あるロケーション間に 1 つのガード制約・リセット関数を割り当てている．

提案シミュレーション手法では，ハイブリッドシステムとステップ数の上限 k が与えられると，時間 $t \in [0, t_k]$ から状態 $(l, \mathbf{x}(t)) \in \mathcal{L} \times \mathbb{R}^n$ への写像である**軌道**を計算する．時刻 t の状態 $(l, \mathbf{x}(t))$ を，下記のように初期状態 $(l_0, \mathbf{x}(0))$ $(\mathbf{x}(0) \in [\mathbf{x}(0)])$ から前向きに求めることができる:

$$
\begin{aligned}
(l_0, \mathbf{x}_-(t_1)) &:= \phi_{l_0}(\mathbf{x}(0), t_1) \\
&\quad \text{s.t. } \forall \tilde{t} \in [0, t_1) \forall \tilde{l} \in \mathcal{L}\ \neg grd_{l_0,\tilde{l}}(\mathbf{x}(\tilde{t})) \wedge grd_{l_0,l_1}(\mathbf{x}_-(t_1)), \\
(l_1, \mathbf{x}(t_1)) &:= \sigma_{l_0,l_1}(\mathbf{x}_-(t_1)), \\
(l_1, \mathbf{x}_-(t_2)) &:= \phi_{l_1}(\mathbf{x}(t_1), t_2 - t_1) \\
&\quad \text{s.t. } \forall \tilde{t} \in [t_1, t_2) \forall \tilde{l} \in \mathcal{L}\ \neg grd_{l_1,\tilde{l}}(\mathbf{x}(\tilde{t})) \wedge grd_{l_1,l_2}(\mathbf{x}_-(t_2)), \\
&\qquad \vdots \\
(l, \mathbf{x}(t_j) &:= \sigma_{l_{j-1},l}(\mathbf{x}_-(t_j)), \\
(l, \mathbf{x}(t)) &:= \phi_l(\mathbf{x}(t_j), t - t_j), \\
&\quad \text{s.t. } \forall \tilde{t} \in [t_j, t) \forall \tilde{l} \in \mathcal{L}\ \neg grd_{l_j,\tilde{l}}(\mathbf{x}(\tilde{t})).
\end{aligned} \tag{1}
$$

ここでロケーション $l_1, \ldots, l_{j-1} \in \mathcal{L}$ および時刻 $t_1, \cdots, t_j \in (0, t)$ は適当にとるものとする．また本稿ではこれらを一意にとれるものとする．

3. 平行体に基づく写像の計算

一般に，単純な区間計算を用いて区間から区間への写像を繰り返し計算していくと，ラッピング効果 (*wrapping effect*) により区間サイズが爆発する．これを防ぐために多角形で値を包囲する計算手法が提案されている [15]．平行体を用いる手法 (例: [16]) もその 1 つであり，平行体は box に線形変換を施すのみで計算できるため効率がよいという利点がある．

n 次元の**平行体** (parallelotope) を 3 つ組 $\langle A, [\mathbf{u}], \mathbf{v} \rangle$，あるいは $\langle \mathbf{x} \rangle$ のように表す．ここで A は $n \times n$ 行列，$[\mathbf{u}]$ は n 次元区間ベクトル，$\mathbf{v}$ は n 次元実ベクトルである．上記 3 つ組は集合 $\{A\tilde{\mathbf{u}} + \mathbf{v} \mid \tilde{\mathbf{u}} \in [\mathbf{u}]\}$ と解釈する．n 次元の平行体の集合を $\mathbb{P}^n$ で表す．

ある写像 $\mathbf{f}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ について，$\mathbf{f}(\cdot)$ の**平行体拡張** $\langle \mathbf{f} \rangle(\cdot) : \mathbb{P}^n \to \mathbb{P}^n$ を考える．任意の n 次元平行体 $\langle A, [\mathbf{u}], \mathbf{v} \rangle$ について以下がなりたつ:

$$
\forall \mathbf{a} \in \langle A, [\mathbf{u}], \mathbf{v} \rangle\ (f(\mathbf{a}) \in \langle \mathbf{f} \rangle(\langle A, [\mathbf{u}], \mathbf{v} \rangle)).
$$

本研究では平行体拡張 $\langle \mathbf{f} \rangle(\langle A, [\mathbf{u}], \mathbf{v} \rangle)$ の値 $\langle A', [\mathbf{u}'], \mathbf{v}' \rangle$ を以下のようにして計算する:

- $\mathbf{v}' := \mathrm{mid}\,[\mathbf{f}](\mathbf{v})$.
- $A' := \mathrm{mid}\,[J]A$，あるいは QR 分解法で $\mathrm{mid}\,[J]A$ の各列を直交化してもよい．ただし，$[J] = [D\mathbf{f}](\square\langle A, [\mathbf{u}], \mathbf{v} \rangle)$ である．前者は $\mathbf{f}(\cdot)$ が線形写像であれば厳密な像を与える．後者は A が特異行列に近いときに有効である．
- 上記 $\mathbf{v}'$ と A' が求まれば，$[\mathbf{u}']$ は $\{A'^{-1}(\mathbf{f}(A\tilde{\mathbf{u}} + \mathbf{v}) - \mathbf{v}') \mid \tilde{\mathbf{u}} \in [\mathbf{u}]\}$ のように表せる．そこで区間計算により $[\mathbf{u}'] := A'^{-1}([\mathbf{f}](A[\mathbf{u}] + \mathbf{v}) - \mathbf{v}')$ を計算する．さらに次のように中間値形式 ([14] を参照) に変形して精度よく計算することが可能である:

$$
[\mathbf{u}'] := (A'^{-1}[J]A)[\mathbf{u}] + A'^{-1}([\mathbf{v}'] - \mathrm{mid}\,[\mathbf{v}']). \tag{2}
$$

上記のような平行体拡張 $\langle \mathbf{f} \rangle(\cdot)$ に十分小さな平行体 $\langle \mathbf{x} \rangle$ を与えれば，$\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \langle \mathbf{x} \rangle\}$ の精度のよい近似を得ることができる．

4. ハイブリッドシステムのシミュレーション手法

あるハイブリッドシステムにおける，時刻 t_i の状態 $(l_i, \mathbf{x}(t_i))$ から，$t_{i+1} - t_i$ 秒間連続変化した後ガード制約を満たし，リ

Input: ハイブリッドシステム HS, ステップ数 $k \in \mathbb{N}$
Output: フローパイプ FP

1: $FP := \emptyset$;
2: $t := 0$; $l := l_0$; $\langle \mathbf{x} \rangle := \langle I_n, [\mathbf{x}_0] - \mathrm{mid}\,[\mathbf{x}_0], \mathrm{mid}\,[\mathbf{x}_0] \rangle$;
3: **for** $i \in \{1, \ldots, k\}$ **do**
4: $\quad [t_c] := [t_{max}]$;
5: $\quad$ **for** $l' \in \mathcal{L}$ **do**
6: $\quad\quad [t'_c] := \mathsf{SearchZero}([\phi_l](\langle \mathbf{x} \rangle, \cdot), [h_{l,l'}](\cdot), [g_{l,l'}](\cdot))$;
7: $\quad\quad$ **if** $[t'_c] \neq \emptyset \wedge \mathsf{IsEarlier}([t'_c], [t_c])$ **then** $[t_c] := [t'_c]$; $l_c := l'$;
8: $\quad$ **end for**
9: $\quad$ **if** $[t_c] = [t_{max}]$ **then return** $\emptyset$;
10: $\quad FP := FP \cup \{(\langle \mathbf{x} \rangle, [\mathbf{x}](\cdot), \langle \phi_l \rangle(\langle \mathbf{x} \rangle, \bar{t}_c), [t, \bar{t}_c])\}$;
11: $\quad \langle \mathbf{x} \rangle := \mathsf{Jump}(\langle \mathbf{x} \rangle, [\mathbf{x}]([t_c]), [t_c], \phi_l(\cdot, \cdot), \phi'_{l_c}(\cdot, \cdot)), \sigma_{l,l_c}(\cdot), h_{l,l_c}(\cdot))$;
12: $\quad t := t + \bar{t}_c$; $l := l_c$;
13: **end for**
14: **return** FP;

図 1　シミュレーションアルゴリズム

セット関数の適用により離散変化し，状態 $(l_{i+1}, \mathbf{x}(t_{i+1}))$ へ至る振る舞いについて，$\mathbf{x}(t_i) \mapsto \mathbf{x}(t_{i+1})$ と対応づけるような写像を $\omega_{l_i,l_{i+1}}(\cdot)$ で表す．提案手法はその平行体拡張 $\langle \omega_{l_i,l_{i+1}} \rangle(\cdot)$ を計算するアルゴリズムを与える．ハイブリッドシステムの初期状態 $(l_0, \langle \mathbf{x}_0 \rangle)$ から式 (1) のように k ステップ分 $\langle \omega_{l_i,l_{i+1}} \rangle(\cdot)$ による像を計算していけば，平行体計算に基づくシミュレーションを実施することができる．

シミュレーションのアルゴリズムを図 1 に示す．アルゴリズムは HS を初期状態から k ステップ分シミュレートし，フローパイプ (flowpipe) を出力する．フローパイプは平行体と box による軌道の包囲であり，$(\langle \mathbf{x} \rangle, [\mathbf{x}](\cdot), \langle \mathbf{x}' \rangle, [t]) \in \mathbb{P}^n \times \mathcal{M}([t], \mathbb{I}^n) \times \mathbb{P}^n \times \mathbb{I}_{\geqq 0}$ のような 4 つ組を要素とする集合である ($\mathcal{M}([t], \mathbb{I}^n)$ は関数 $[t] \to \mathbb{I}^n$ の集合を表す)．4 つ組のはじめの 3 要素が時間区間 $[t]$ 上の連続状態の包囲となっており，$\langle \mathbf{x} \rangle$ と $\langle \mathbf{x}' \rangle$ が状態 $\mathbf{x}(\underline{t})$ と $\mathbf{x}(\bar{t})$ の平行体包囲，任意の時間区間 $[\tilde{t}] \subseteqq [t]$ について，関数 $[\mathbf{x}]([\tilde{t}])$ が状態集合 $\{\mathbf{x}(\tilde{t}) \mid \tilde{t} \in [\tilde{t}]\}$ の区間包囲を与える．

提案手法は計算結果の平行体および box について唯一解の存在保証を行うため，アルゴリズムがフローパイプを返したならば，各初期連続値 $\mathbf{x}_0 \in [\mathbf{x}_0]$ について，唯一の k ステップ長の軌道が存在し，軌道はフローパイプに包囲されていることが保証される．また，アルゴリズムが空集合を返したならば，初期条件について k ステップ長の軌道は存在しないことが保証される．上記が保証できない場合，計算はエラーとなる．

アルゴリズムの 3–13 行目のループにおいて，現在の時刻 t，ロケーション l，連続状態の平行体包囲 $\langle \mathbf{x} \rangle$ をステップ毎に更新していく．4–8 行目で各ロケーション $l' \in \mathcal{L}$ について，l, l' 間の離散変化が生起しうるかどうかを確認している．手続き SearchZero は離散変化のガード制約の充足可能性を調べ，充足可能ならば，ガード制約をみたす時間 $[t_c]$ (現在ロケーション到達時刻を 0 とする)，充足可能でないならば空集合を返す．引数の $[\phi_l](\langle \mathbf{x} \rangle, \cdot)$ は区間計算にもとづく ODE 求解処理 (ref)，$[h_{l,l'}](\cdot)$ と $[g_{l,l'}](\cdot)$ はガード制約の左辺の区間拡張とする．本稿

Input: $\langle \mathbf{x} \rangle = \langle A, [\mathbf{u}], \mathbf{v} \rangle \in \mathbb{P}^n$, $[\mathbf{x}_c] \in \mathbb{I}^n$, $[t_c] \in \mathbb{I}_{\geqq 0}$, $\phi(\cdot, \cdot), \phi'(\cdot, \cdot) : \mathbb{R}^n \times [0, t_{max}] \to \mathbb{R}^n$, $\sigma(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $h(\cdot) : \mathbb{R}^n \to \mathbb{R}$
Output: $\langle A', [\mathbf{u}'], \mathbf{v}' \rangle \in \mathbb{P}^n$

1: $[\mathbf{v}'] := [\sigma]([\phi](\mathbf{v}, \mathrm{mid}\,[t_c]))$;
2: $[\mathbf{v}'] := [\phi']([\mathbf{v}'], \bar{t}_c - \mathrm{mid}\,[t_c])$;
3: $[Dt] := -\dfrac{[Dh]([\mathbf{x}_c])[D_{\mathbf{x}}\phi](\langle \mathbf{x} \rangle, [t_c])}{[Dh]([\mathbf{x}_c])[D_t\phi](\langle \mathbf{x} \rangle, [t_c])}$; **{** 式 (5)**}**
4: $[D_{\mathbf{x}}\sigma] := [D\sigma]([\mathbf{x}_c])\big([D_{\mathbf{x}}]([t_c]) + [D_t\phi](\langle \mathbf{x} \rangle, [t_c])[Dt]\big)$; **{** 式 (4)**}**
5: $[J] := [D_{\mathbf{x}}\phi']\left([\sigma]([\mathbf{x}_c]), \bar{t}_c - [t_c]\right)[D_{\mathbf{x}}\sigma] - [D_t\phi']\left([\sigma]([\mathbf{x}_c]), \bar{t}_c - [t_c]\right)[Dt]$; **{** 式 (3)**}**
6: $A' := \mathrm{mid}\,[J]A$; **{** あるいは QR 分解を用いる．**}**
7: $[\mathbf{u}'] := (A'^{-1}[J]A)[\mathbf{u}] + A'^{-1}([\mathbf{v}'] - \mathrm{mid}\,[\mathbf{v}'])$; **{** 式 (2)**}**
8: **return** $\langle A', [\mathbf{u}'], \mathrm{mid}\,[\mathbf{v}'] \rangle$

図 2　Jump アルゴリズム

では SearchZero の実装方法は省略する．既存手法 [7], [8] のアルゴリズムを用いることができる．時間区間の比較手続き IsEarlier を用い，最も早く生起する離散変化の時間区間を 1 つ求める．複数時間が重なっており比較が困難な場合は IsEarlier がエラーとなる．10 行目でフローパイプを出力している．$\langle \phi_l \rangle(\langle \mathbf{x} \rangle, \cdot)$ は平行体計算にもとづく ODE 求解処理 (5. 節を参照) とする．11 行目の手続き Jump (4. 1 節) は離散変化後の状態の平行体包囲 $\langle \omega_{l,l'} \rangle(\langle \mathbf{x} \rangle)$ を計算する．

4. 1　離散変化の計算

1 回の離散変化を平行体にもとづき計算するアルゴリズム Jump を図 3 に示す．アルゴリズムは，時刻 0 の連続状態の平行体包囲 $\langle A, [\mathbf{u}], \mathbf{v} \rangle$，離散変化が起こる時間区間 $[t_c]$，$[t_c]$ 上の連続状態の包囲 $[\mathbf{x}_c]$ から，時刻 $\bar{t}_c$ における離散変化後の連続状態の平行体包囲 $\langle A', [\mathbf{u}'], \mathbf{v}' \rangle$ を計算する．

まず 1–2 行目において，ODE と σ の区間計算により $[\mathbf{v}']$ を計算する．

次に 3. 節で述べたように A' を計算するために，平行体拡張する関数 $\omega(\cdot)$ のヤコビ行列 $J(\cdot)$ を用いる．そこで提案手法では $\omega(\cdot)$ の導関数を求め，区間計算する．初期状態 $\mathbf{x} \in \mathbb{R}^n$ の離散変化時刻を関数 $t(\mathbf{x})$，離散変化前の連続状態を関数 $\mathbf{x}_-(\mathbf{x}) = \phi(\mathbf{x}, t(\mathbf{x}))$ で表すとすると，関数 ω は以下のように定義できる:

$$\omega(\mathbf{x}) := \phi'(\sigma(\mathbf{x}_-(\mathbf{x}), \bar{t}_c - t(\mathbf{x}))).$$

$\omega(\cdot)$ の導関数は連鎖律により次のように計算できる:

$$D\omega(\mathbf{x}) = D_{\mathbf{x}}\phi'\left(\sigma(\mathbf{x}_-(\mathbf{x})), \bar{t}_c - t(\mathbf{x})\right) D_{\mathbf{x}}\sigma(\mathbf{x}_-(\mathbf{x})) - D_t\phi'\left(\sigma(\mathbf{x}_-(\mathbf{x})), \bar{t}_c - t(\mathbf{x})\right) Dt(\mathbf{x}). \quad (3)$$

ただし，$D_{\mathbf{x}}\sigma(\mathbf{x}_-(\mathbf{x}))$ と $Dt(\mathbf{x})$ は次のように計算する:

$$D_{\mathbf{x}}\sigma(\mathbf{x}_-(\mathbf{x})) := D\sigma(\mathbf{x}_-(\mathbf{x})) \left(D_{\mathbf{x}}\phi(\mathbf{x}, t(\mathbf{x})) + D_t\phi(\mathbf{x}, t(\mathbf{x}))Dt(\mathbf{x})\right), \quad (4)$$

$$Dt(\mathbf{x}) := -\frac{Dh(\phi(\mathbf{x}, t(\mathbf{x})))D_{\mathbf{x}}\phi(\mathbf{x}, t(\mathbf{x}))}{Dh(\phi(\mathbf{x}, t(\mathbf{x})))D_t\phi(\mathbf{x}, t(\mathbf{x}))}. \quad (5)$$

アルゴリズムの 3–5 行目において上式 (3)–(5) の区間拡張を計

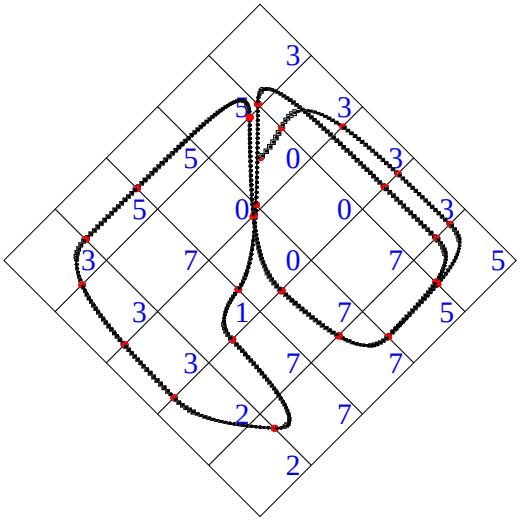

図 3　Navigation の計算結果 ($(\mathbf{x}_1, \mathbf{x}_2)$ を図示).

算している.

$[\mathbf{v}']$ と $[J]$ をもとに 3. 節で述べたように A' と $[\mathbf{u}']$ を計算し，平行体 $\langle A', [\mathbf{u}'], \mathrm{mid}\,[\mathbf{v}']\rangle$ を得ることができる.

5. 実　　装

提案手法を C/C++および Ocaml で実装した．実装にあたっては，区間演算ライブラリ Filib++と，ODE 求解ライブラリ CAPD(注1) を用いた．CAPD を利用し，ODE の解軌道 $\phi(\mathbf{x}, t) : \mathbb{R}^n \times [0, t_{max}] \to \mathbb{R}^n$ について，提案アルゴリズムで必要な $\langle\phi\rangle(\langle\mathbf{x}\rangle, t) : \mathbb{P}^n \times [0, t_{max}] \to \mathbb{P}^n$，$[\phi](\langle\mathbf{x}\rangle, [t]) : \mathbb{P}^n \times \mathbb{I}_{\geq 0} \to \mathbb{P}^n$，$[D_{\mathbf{x}}\phi](\langle\mathbf{x}\rangle, [t]) : \mathbb{P}^n \times \mathbb{I}_{\geq 0} \to \mathbb{P}^n$ を実装することができる．また CAPD は連続軌道のテイラー展開を用いた計算を `Curve` と呼ばれるクラスで隠蔽しており，アルゴリズム中での連続軌道の計算にはこれを用いた.

図 3 のアルゴリズムの 6 行目について，A' について条件数を計算し，それに応じて 2 つの方法を切り替えるようにした.

6. 実　　験

ハイブリッドシステムの例を 2 つ挙げ，それぞれについて提案手法による計算結果を述べる.

6.1 Navigation

1 つ目の例として，$\sqrt{2} \times \sqrt{2}$ の大きさの矩形領域が $n \times n$ 個並んだ平面上を動くオブジェクトをモデリングした問題 navigation [17] のシミュレーションを実施する．本稿では，元の問題を変更し，矩形領域を 45° 傾けたモデルを考える．平面とシミュレーション結果を図 4 に示す．モデルでは，4 次元の変数 $\mathbf{x}$ でオブジェクトの位置 $(\mathbf{x}_1, \mathbf{x}_2)$ と速度 $(\mathbf{x}_3, \mathbf{x}_4)$ を表す．各矩形領域に応じてロケーションを用意する．隣接するロケーション間に，境界への到達判定を表すガード制約と，恒等写像としたリセット関数を設定する．各ロケーションには下記 ODE を設定する:

(注1)：`http://capd.ii.uj.edu.pl/`.

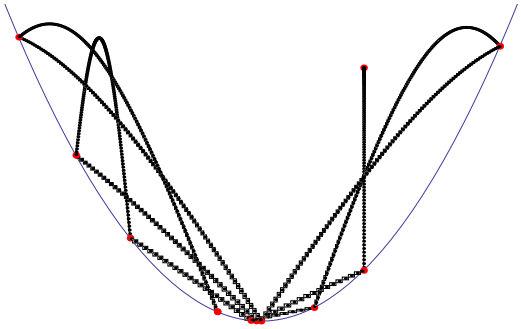

図 4　Bouncing ball ($m = 3$) の計算結果 ($(\mathbf{x}_1, \mathbf{x}_3)$ を図示).

$$
\begin{aligned}
(\dot{\mathbf{x}}_1(t), \dot{\mathbf{x}}_2(t)) &= (\mathbf{x}_3(t), \mathbf{x}_4(t)),\\
(\dot{\mathbf{x}}_3(t), \dot{\mathbf{x}}_4(t))^T &= A\left((\mathbf{x}_3(t), \mathbf{x}_4(t))^T - \mathbf{v}_{d,l}\right).
\end{aligned}
$$

本実験では下記の A と $\mathbf{v}_{d,l}$ を用いた:

$$
A = \begin{pmatrix} -1.2 & 0.1 \\ 0.2 & -1.2 \end{pmatrix},\ \mathbf{v}_{d,l} = \begin{pmatrix} \sin(i_l \pi/4) \\ \cos(i_l \pi/4) \end{pmatrix}.
$$

ただし各ロケーションに $i_l \in \{0, \ldots, 7\}$ があらかじめ図 4 のように設定されているものとする．初期条件はロケーションが 3 段目中央の矩形領域，連続状態が $(0, 7, 1, 1)$ とした (下部の頂点が $(0, 0)$).

図 4 には実装を用いて 100 ステップシミュレーションした結果得られたフローパイプを示している．図中の連続軌道の box 包囲は最大時間幅 0.1 で計算した．計算には約 4 秒を要した.

6.2 2 次曲面上の bouncing ball

2 つ目に，非線形 ODE と非線形ガード制約，非線形リセット関数からなる例題である，2 次曲面上で跳ね返る質点のモデルを考える．ここでは m 次元の位置座標をもつ質点のモデルを $n = 2m$ 個の変数をもつハイブリッドシステムで表す．ハイブリッドシステムは 1 ロケーションからなり，同一ロケーション間の離散変化で跳ね返りを表す．各要素は次のように設定する:

$$
\begin{aligned}
\mathbf{x} &= \{\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{v}_1, \ldots, \mathbf{v}_m\},\\
\dot{\mathbf{x}}(t) &= (\mathbf{v}_1(t), \ldots, \mathbf{v}_m(t), 0, \ldots, 0, -g + f\mathbf{x}_m^2(t)),\\
h_{l,l}(\mathbf{x}) &= \mathbf{x}_1^2 + \cdots + \mathbf{x}_{m-1}^2 - \mathbf{x}_m,\\
g_{l,l}(\mathbf{x}) &= \mathbf{x}_m - 2\mathbf{x}_1\mathbf{v}_1 - \cdots - 2\mathbf{x}_m\mathbf{v}_m,\\
\sigma_{l,l}(\mathbf{x}) &= \Big(\mathbf{x}_1, \ldots, \mathbf{x}_m,\\
&\qquad \mathbf{v}_1 + \frac{4\mathbf{x}_1(\mathbf{v}_m - 2\mathbf{x}_1\mathbf{v}_1 - \cdots - 2\mathbf{x}_{m-1}\mathbf{v}_{m-1})}{1 + 4\mathbf{x}_1^2 + \cdots + 4\mathbf{x}_{m-1}^2},\\
&\qquad \vdots\\
&\qquad \mathbf{v}_{m-1} + \frac{4\mathbf{x}_{m-1}(\mathbf{v}_m - 2\mathbf{x}_1\mathbf{v}_1 - \cdots - 2\mathbf{x}_{m-1}\mathbf{v}_{m-1})}{1 + 4\mathbf{x}_1^2 + \cdots + 4\mathbf{x}_{m-1}^2},\\
&\qquad \mathbf{v}_n + \frac{2(\mathbf{v}_m - 2\mathbf{x}_1\mathbf{v}_1 - \cdots - 2\mathbf{x}_{m-1}\mathbf{v}_{m-1})}{1 + 4\mathbf{x}_1^2 + \cdots + 4\mathbf{x}_{m-1}^2}\Big).
\end{aligned}
$$

実験ではパラメタ値 $g = 1, f = 0.1$ を設定した.

$m = 3$ のインスタンスについて，初期値を $\mathbf{x}(0) = (1, 1, 10, 0, 0, 0)$ として，10 ステップシミュレーションした結果を図 5 に示す.

多ステップ (10229 ステップ) のシミュレーションを実施した結果について，図 6 と図 7 に示す．ここでは，$m = 2$ 次元のインスタンスに，2 ステップでほぼ周期軌道となる初期値

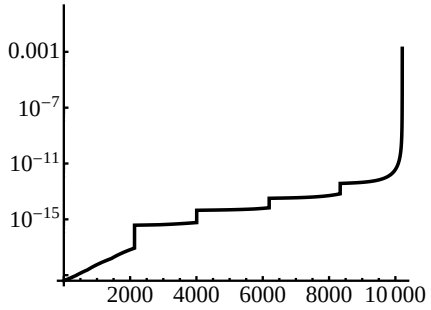

図 5　ステップ数 (横軸) と平行体の体積 (縦軸).

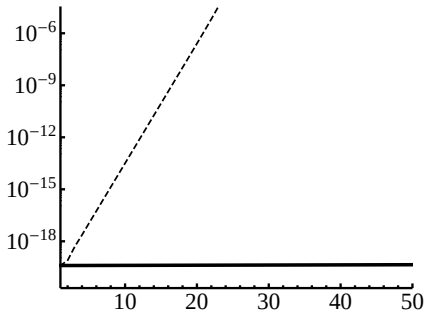

図 6　提案手法 (太線) と box 手法 (点線).

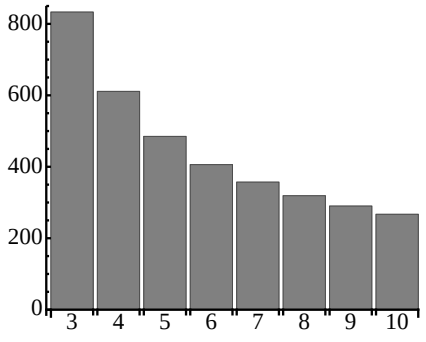

図 7　次元 (横軸) とシミュレーション可能なステップ数 (縦軸).

$\mathbf{x}(0) = (-1, 1, 1.4, 0.7)$ を設定し，シミュレーションしている．プロット線に段差ができているのは，図 3 のアルゴリズム 6 行目の 2 種類の計算が切り替わっているためだと考えられる．図 7 は離散変化後の連続状態を box で包囲する手法との比較結果である．Box 手法ではラッピング効果により急速に box が膨張し，22 ステップしかシミュレーションできなかった．

$m = 3$–10 次元のインスタンスを同様に多ステップシミュレーションした結果について，図 8 に示す．3 次元以上では 2 次元のインスタンスに較べ大幅にステップ数が少なくなったが，両者の間には離散変化の非線形性に本質的な差異があるものと考えられる．

7. ま と め

写像の平行体拡張スキームをハイブリッドシステムの軌道の計算に適用することでラッピング効果を抑えた区間シミュレーションを実現した．今後の課題として，提案手法の現実的な問題への適用や，形式的検証への応用が挙げられる．

謝辞　本研究の一部は科研費 25880008 の補助を得て行った．

文　献

[1] T. Park and P.I. Barton, “State event location in differential-algebraic models,” ACM Transactions on Modeling and Computer Simulation, vol.6, no.2, pp.137–165, 1996.

[2] A. Eggers, M. Fränzle, and C. Herde, “SAT Modulo ODE : A Direct SAT Approach to Hybrid Systems,” Proc. of ATVA, pp.171–185, *LNCS*5311, 2008.

[3] P. Collins and A. Goldsztejn, “The Reach-and-Evolve Algorithm for Reachability Analysis of Nonlinear Dynamical Systems,” Electronic Notes in Theoretical Computer Science, vol.223, no.639, pp.87–102, 2008.

[4] D. Ishii, K. Ueda, and H. Hosobe, “An interval-based SAT modulo ODE solver for model checking nonlinear hybrid systems,” International Journal on Software Tools for Technology Transfer (STTT), vol.13, no.5, pp.449–461, 2011.

[5] A. Eggers, N. Ramdani, N. Nedialkov, and M. Franzle, “Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods,” Software & Systems Modeling, 2012.

[6] S. Gao and E.M. Clarke, “Satisfiability Modulo ODEs,” Proc. of FMCAD, pp.105–112, 2013.

[7] D. Ishii, K. Ueda, H. Hosobe, and A. Goldsztejn, “Interval-based solving of hybrid constraint systems,” Proc. of the third IFAC Conference on Analysis and Design of Hybrid Systems (ADHS), pp.144–149, 2009.

[8] N. Ramdani and N.S. Nedialkov, “Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques,” Nonlinear Analysis: Hybrid Systems, vol.5, no.2, pp.149–162, 2011.

[9] X. Chen, E. Abraham, and S. Sankaranarayanan “Taylor Model Flowpipe Construction for Non-linear Hybrid Systems,” Proc. of RTSS, pp.183–192, 2012.

[10] O. Bouissou, A. Chapoutot, S. Mimram “Computing Flowpipe of Nonlinear Hybrid Systems with Numerical Methods,” arXiv preprint, 2013.

[11] P. Thota and H. Dankowicz, “TC-HAT ($\widehat{TC}$): A Novel Toolbox for the Continuation of Periodic Trajectories in Hybrid Dynamical Systems,” SIAM Journal on Applied Dynamical Systems, vol.7, no.4, pp.1283–1322, 2008.

[12] G. Frehse, “PHAVer: algorithmic verification of hybrid systems past HyTech,” STTT, vol.10, no.3, pp.263–279, 2008.

[13] G. Frehse, C.L. Guernic, A. Donz, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, “SpaceEx : Scalable Verification of Hybrid Systems,” Proc. of CAV, pp.379–395, *LNCS*6806, 2011.

[14] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, 1990.

[15] O. Stursberg and B.H. Krogh, “Efficient Representation and Computation of Reachable Sets for Hybrid Systems,” Proc. of HSCC, pp.482–497, *LNCS*2623, 2003.

[16] A. Goldsztejn and L. Granvilliers, “A New Framework for Sharp and Efficient Resolution of NCSP with Manifolds of Solutions,” Constraints, vol.15, no.2, pp.190–212, 2010.

[17] A. Fehnker and F. Ivančić, “Benchmarks for Hybrid Systems Verification,” Proc. of HSCC, pp.326–341, *LNCS*2993, 2004.