1D1-3

数式処理・Quantifier Eliminationを用いたハイブリッドシステムのZeno状態の導出手法

A Method of Deriving Zeno States in Hybrid Systems using Formula Manipulation and Quantifier Elimination

大野 善之[*1] 石井 大輔[*1] 上田 和紀[*2]
Yoshiyuki Ohno Daisuke Ishii Kazunori Ueda

[*1]早稲田大学大学院理工学研究科情報・ネットワーク専攻
Graduate School of Computer Science, Waseda University

[*2]早稲田大学理工学術院
Faculty of Science and Engineering, Waseda University

Hybrid systems are systems that exhibit both continuous and discrete dynamic behaviors. Some hybrid system models have a behavior called Zenoness, and the such models are called Zeno models. A Zeno model is a model with an execution that takes an infinite number of discrete transitions in a finite time interval. So, simulations and verifications of Zeno models often indicate an unexpected result, or don't terminate. In this paper, we propose a method to detect Zeno states using formula manipulation and quantifier elimination.

1. はじめに

時間の経過に伴って状態が連続変化したり，状態や方程式系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．ハイブリッドシステムのモデルには，有限時間内に離散変化が無限回起こり時間が進まなくなる性質 **Zenoness** を持つものがある．Zenoness を持つモデル (Zeno モデル) をシミュレーション実行すると，シミュレーションが進まなくなったり，予期せぬ結果を返したりするという問題が発生する．

例えば，ハイブリッドシステムの例としてボールが自由落下して地面で撥ね返るという事象を考える．ボールは障害物の無い限り重力加速度にしたがって落下を続ける (連続変化)．そして，地面に衝突したときに速度が離散的に変化する (離散変化)．そして，再び重力加速度にしたがって連続的に変化を続ける．この事象をモデル化して考える．ボールの位置を y とすると，ボールは通常 $y'' = -9.8$ という微分方程式で連続的に落下し，地面と衝突した瞬間 ($y = 0$ のとき) に，その時点の速度 y'_d が $y'_d \mapsto -0.5y'_d$ という離散変化をするというモデルとする．このモデルでは，衝突が起こるたびに，衝突と次の衝突の時間の間隔が徐々に短くなっていく．そのため，数値計算で解析すると無限の処理が必要になるため，有限時間での計算が不可能になる．

そこで本研究では，モデルが Zenoness を持つかどうかの可能性，およびモデル内のどのような状態において Zeno 実行を示すのかを導出する手法を構築することを目指す．これらの導出をすることができるようになれば，モデルの修正により Zenoness を解消することができるようになる．

本研究では 2 つのアイデアを提案した．Zenoness は無限に離散変化が起こるという性質のため，Zenoness を持つモデルには，ある離散状態から有限回の離散変化を経て同一の離散状態に戻るようなループを持つという特徴がある．したがって，そのようなループを抜き出して解析することで，それ以降 Zeno 実行を示すような状態 (**Zeno 状態**) を導出できると考えられる．そこで，モデルが持つループ状態を，数式処理と Quantifier Elimination を用いて導出する手法を提案した．

2 つ目として，導出したループ状態が Zeno 状態であるかを判定する条件を定めた．この条件も，数式処理および Quantifier Elimination を用いて満たすかどうか判定することができる．

提案手法を用いることで，モデルが陥りうる Zeno 状態を導出することができるようになる．これにより，Zeno 状態から抜け出すような離散変化を追加したり，Zeno 状態へ到達しないような初期値を定めるなどといったモデルの改良で Zenoness を回避することができるようになる．

連絡先: 大野 善之，早稲田大学大学院理工学研究科情報・ネットワーク専攻，〒 169-8555 新宿区大久保 3-4-1 63 号館 5 階 02 号，y.ohno(at)ueda.info.waseda.ac.jp

2. ハイブリッドシステムと Zenoness

本節では，ハイブリッドシステムおよび Zenoness について簡単に述べる．詳細については，文献 [1, 2] を参照されるとよい．

2.1 ハイブリッドシステム

時間の経過に伴って状態が連続変化したり，状態や方程式系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．ハイブリッドシステムの定義を以下に記す．

定義 2.1.1 (ハイブリッドシステム)
ハイブリッドシステム $\mathscr{H}$ は以下の組からなる．

$$\mathscr{H} = (\Gamma, X, G, R, F)$$

$\Gamma = (Q, E)$ は有限有向グラフであり，Q は離散状態の集合，E は 2 つの離散状態をつなぐ辺を表す．また，辺 $e \in E$ は始点 $s : E \to Q$ と終点 $t : E \to Q$ をもつ．X は連続状態 $\boldsymbol{x} \in \mathbb{R}^n$ の集合である．ハイブリッドシステムの状態は，離散状態と連続状態の直積 $(q, \boldsymbol{x})$ である．$G = \{G_e \subseteq \mathbb{R}^n \mid e \in E\}$ は，ガードの集合であり，離散状態 $s(e)$ にて連続状態 $\boldsymbol{x}$ と G_e の積が空でないとき，離散状態 $t(e)$ に遷移する．$R = \{R_e : X \to X \mid e \in E\}$ はリセットマップの集合であり，離散状態が $s(e)$ から $t(e)$ に変化するとき，連続状態 $\boldsymbol{x}$ が $R_e(\boldsymbol{x})$ に書き換わる．$F = \{f_q : X \to \mathbb{R}^n \mid q \in Q\}$ はベクトル場の集合であり，離散状態 q では $\boldsymbol{x}' = f_q(\boldsymbol{x})$ にしたがって，連続状態 $\boldsymbol{x}$ が連続

的に変化する．この微分方程式の初期値を $\boldsymbol{x}_0$ としたときの解を $\phi_q(t, \boldsymbol{x}_0)$ と表記する．

ハイブリッドシステムは初期状態 $(q_0, \boldsymbol{x}_0) : q_0 \in Q, \boldsymbol{x}_0 \in X$ を与えることで実行する．ハイブリッドシステムの実行を以下に定義する．

定義 2.1.2 (ハイブリッドシステムの実行)
ハイブリッドシステムの実行 χ は以下の組からなる．

$$\chi = (\rho, \tau, \xi, \eta)$$

$\rho = \{\rho_i \in Q\}_{i\in\mathbb{N}}$ は，i 番目の離散状態 ρ_i の列である．$\tau = \{\tau_i \in \mathbb{R}\}_{i\in\mathbb{N}}$ は，i 番目の離散変化時間 τ_i の列であり，$0 = \tau_0 < \tau_1 < \cdots < \tau_j < \ldots$ である．$\xi = \{\xi_i\}_{i\in\mathbb{N}}$ は，i 番目の離散状態 ρ_i における初期状態 $\xi_i \in X$ の列である．$\eta = \{\eta_i\}_{i\in\mathbb{N}^+}$ は，i 番目の離散変化 $\eta_i \in E$ の列であり，$\rho_i = t(\eta_i) = s(\eta_{i+1})$ を満たす．

さらに，実行 χ は以下の条件を満足する．

- $\tau_{i+1} = min\{t \in \mathbb{R}^+ > \tau_i : \phi_{\rho_i}(t - \tau_i, \xi_i) \in G_{\eta_{i+i}}\}$
- $\forall t \in \mathbb{R}^+ \forall G_e \in G\{\tau_{i+1} - \tau_i > t > 0 \wedge s(G_e) = s(G_{\eta_{i+1}}) \Rightarrow \phi_{\rho_i}(t, \xi_i) \notin G_{\eta_{i+i}}\}$
- $\xi_{i+1} = R_{\eta_{i+1}}(\phi_{\rho_i}(\tau_{i+1} - \tau_i, \xi_i))$

第 *1* および第 *2* 条件で，ガードと連続状態が空でないときすぐに離散変化が起こることを表し，第 *3* 条件は離散変化の際に変数がリセットマップで書き換わることを表している．

また，実行のうち有限の離散変化列を抜き出したものをパスと呼ぶ．

$$\rho_j \xrightarrow{\eta_{j+1}} \rho_{j+1} \xrightarrow{\eta_{j+2}} \rho_{j+2} \xrightarrow{\eta_{j+3}} \cdots \xrightarrow{\eta_k} \rho_k$$

上のようなパスを，$\langle \rho_j : \eta_{j+1}, \eta_{j+2}, \eta_{j+3}, \ldots, \eta_k : \rho_k \rangle$ と表記することとする．さらに，パスの最初の離散状態 ρ_j と最後の離散状態 ρ_k が同一状態のものをサイクルパスと呼ぶ．

2.2 Zenoness

Zenoness は，有限時間内に離散変化が無限回起こる性質である．以下に，Zenoness の定義を記す．

定義 2.2.1 (Zenoness)
ハイブリッドシステムのある実行が，

$$\lim_{i\to\infty} \tau_i = \sum_{i=0}^{\infty} (\tau_{i+1} - \tau_i) = \tau_\infty$$

であるような定数 $\tau_\infty \in \mathbb{R}^+$ を持つとき，そのハイブリッドシステムは *Zenoness* を持つという．このときの実行を *Zeno* 実行といい，定数 τ_∞ を *Zeno* 時刻と呼ぶ．

3. 前提知識

本節では，我々の提案手法で必要となる前提知識を紹介する．

3.1 数式処理

数式処理は数式を記号的に代数処理する技術あり，代数方程式や微分方程式の求解をすることができる．Maxima や Maple といったソフトウェアが存在する．

3.2 Quantifier Elimination

quantifier elimination(限定記号消去,QE) は，与えられた形式理論について，限定記号付きの式 (一階述語論理式) を入力とし，等価で限定記号の無い式を出力する技術である．QEPCAD や REDLOG といったソフトウェアが存在する．

3.3 数式処理に基づく連続システムの可到達範囲の計算 [3]

線形連続システムは，以下のような微分方程式でモデル化される．

$$x' = Ax + u(t)$$

$x(t) \in \mathbb{R}^n$ は時間 t でのシステムの状態を表す．A は n 次正方行列であり，$u(t)$ は連続関数 $u : \mathbb{R} \to \mathbb{R}^n$ である．初期値 $x_0 = x(0)$ を与えると，

$$x(t) = \Phi(x_0, u, t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}u(\tau)d\tau$$

で解を導出することができる．

状態集合 $Y \subseteq \mathbb{R}^n$ が与えられたとき，状態 Y へと到達する全ての前状態 $Pre(Y) \subseteq \mathbb{R}^n$，および，状態 Y から到達しうる全ての後状態 $Post(Y) \subseteq \mathbb{R}^n$ は次式で表現できる．

$$Pre(Y) = \{x \in \mathbb{R}^n \mid \exists y \exists u \exists t : y \in Y \wedge t \geq 0 \wedge \Phi(x, u, t) = y\}$$
$$Post(Y) = \{x \in \mathbb{R}^n \mid \exists y \exists u \exists t : y \in Y \wedge t \geq 0 \wedge \Phi(y, u, t) = x\}$$

以下の条件が満たされれば，$Pre(Y)$ と $Post(Y)$ は数式処理および QE により計算可能である．

- 行列 A が冪零行列：$\exists m \in \mathbb{N}[A^m = \mathbf{0}]$
- u が t の多項式：$u(t) = \sum b_k t^k$

まず，解 $\xi(t) = \Phi(x, u, t)$ を数式処理システムにて求めることができる，次に，解 ξ を QE システムの入力とし，$Pre(Y)$ や $Post(Y)$ を求めることができる．

この技術は，ハイブリッドシステムに応用することができる．状態集合 (q, X) のうち有限時間の連続変化の後に辺 e で離散変化が起こる領域 $To((q, X), e)$ や，その離散変化後の状態 $Post((q, X), e)$ などの以下の式を計算をすることができる．

$$\begin{aligned} To((q,X),e) = \{(q,x_0) \mid \forall t \forall e_2 \exists \tau \exists x : \\ x_0 \in X \wedge \tau > 0 \wedge x = \phi(\tau, x_0) \wedge x \in G_e \\ \wedge (0 < t < \tau \Rightarrow \phi(t, x_0) \notin G_{e_2} \wedge s(e) = s(e_2))\} \end{aligned}$$

$$\begin{aligned} Post((q,X),e) = \{(t(e),x_1) \mid \forall t \forall e_2 \exists \tau \exists x \exists x_0 : \\ x_0 \in X \wedge \tau > 0 \wedge x = \phi(\tau, x_0) \wedge x \in G_e \\ \wedge (0 < t < \tau \Rightarrow \phi(t, x_0) \notin G_{e_2} \\ \wedge s(e) = s(e_2) \wedge x_1 = R_e(x)\} \end{aligned}$$

4. 提案手法

本節では，我々が提案するハイブリッドシステムの Zeno 状態の導出手法について述べる．本提案手法を用いることで，Zenoness のうち Regular Zenoness の判定ができるようになる．

4.1 Regular Zenoness

Zenoness は，Regular Zenoness と Irregular Zenoness にわけることができる．Zeno 実行は有限時間内に無限回の離散変化が起こるということで，その実行パスにはサイクルパスが存在する．ある離散状態以降は常に同一のサイクルパスが続くような Zeno 実行をするモデルを，Regular Zenoness を持つモデルと呼ぶ．また，それ以外の Zenoness を Irregular Zenoness と呼ぶ．

Input: $\mathscr{H} = ((Q, E), X, G, R, F)$
Output: $\mathscr{H}$ が持つループ状態
1: $P = \phi$
2: $I = \phi$
3: $i = 0$
4: $X_0 = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n\}$
5: **for all** $q \in Q$ **do**
6: $i = i + 1$
7: enqueue((q, X_0), I)
8: P に根 (q, i) を追加
9: $A[i] = X_0$
10: $B[i] = false$
11: **end for**
12: **while** キュー I が空でない **do**
13: $((q, Y), j) :=$ dequeue(I)
14: $(p, k) = (q, j)$ の親
15: **for** $B[k] =$ true **do**
16: **if** $(q, j) \subseteq (p, k)$ **then**
17: **Print** $(p, A[k])$ はループ状態である.
18: **end if**
19: $(p, k) = (p, k)$ の親
20: **end for**
21: **for all** $\exists e \in E$ s.t.$s(e) = q$ **do**
22: **if** $To((q, Y), e) \equiv (q, Y)$ **then**
23: $B[j] =$ true
24: **else**
25: $B[j] =$ false
26: **end if**
27: **if** $To((q, Y), e) \neq \phi$ **then**
28: $i := i + 1$;
29: enqueue($Post((q, Y), e), i$)
30: P 内の木のノード (q, j) の子ノードに $(t(e), i)$ を追加
31: $A[i] := Post((q.Y), e)$. 領域部
32: **end if**
33: **end for**
34: **end while**

図 1: アルゴリズム：Loop Search

4.2 ループ状態の導出

定義 4.2.1 (ループ状態)
ハイブリッドシステムが取りうる全状態の部分集合 $(q, D) \subseteq Q \times X$ が以下の条件を満たすとき，(q, D) をループ状態と呼ぶ.

1. (q, D) のどの要素 $(q, \boldsymbol{x})$ を初期値としても，その実行は，全て同じ離散変化列 $\langle q : \eta_1, \eta_2, \ldots \rangle$ を持つ.
2. *1* の離散変化列の中にサイクルパス $\langle q : \eta_1, \eta_2, \ldots, \eta_i : q \rangle$ を持つ.
3. 状態 $(q, \boldsymbol{x}) \in (q, D)$ を初期値とするような実行の i 回目の離散変化後の状態 $Post^i_E(q, \boldsymbol{x})$ が (q, D) の要素である.

図 1 に，ハイブリッドシステムが持つループ状態を探索するアルゴリズム Loop Search を記す．図 2 は Loop Search の基本アイデアをあらわしている．モデルが取りうる全状態 $((q_1, \mathbb{R}^n), (q_2, \mathbb{R}^n), (q_3, \mathbb{R}^n))$ から，到達しうる状態を計算していき，ループとなる状態の導出をしている．例えば，離散状態 q_1 では，$A[1]$ のうち D_2 は，有限時間後に離散状態 q_3 に離散変化し，離散変化後は $A[5]$ の要素となる．さらに，離散状態 q_3 では，$A[5]$ の全ての要素が，有限時間後に同一の離散状態 q_3 に離散変化する．そして離散変化直後は $A[6]$ の要素となる．もし，$A[6] \subseteq A[5]$ であれば，以後，辺 e_{33} で離散変化が続くので，$(q_3, A[5])$ がループ状態であることがわかる．Loop Search は，このようなアイデアでループ状態を導く.

Loop Search では，実行とともに複数の木を構成していく (P:構成される木の集合)．P が持つ木の各ノードは，離散状態 q とインデックス i のペア (q, i) を情報として持っている．木の構成と同時に，表 A,B を構成する ($A[i] \subseteq \mathbb{R}^n$，$B[i] =$ {true or false})．これらの要素は，以下の意味を持つ.

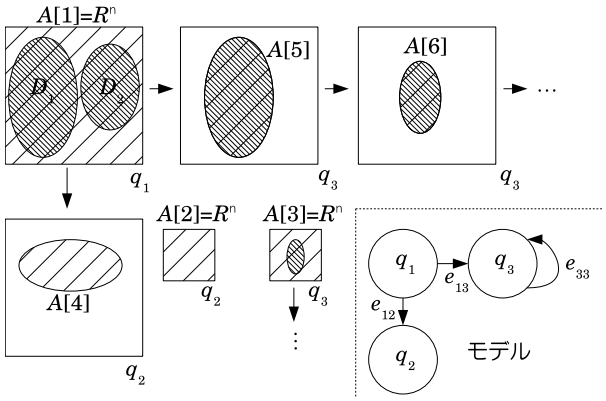

図 2: Loop Search の基本アイデア

- ノード (q, i) が子ノード (r, j) を持つとき，$(q, A[j])$ のある要素は，一回の離散変化で $(r, A[j])$ に到達する.
- 状態集合 $(q, A[i])$ から一回の離散変化で到達しうる状態すべてが，ノード (q, i) の子ノードとなる.
- 状態集合 $(q, A[i])$ の全状態のうち，どのような状態をとっても，離散変化が起き，それは全て同一の辺で起きているとき，$B[i]$ は true である.

もし，状態 $(q, A[i])$ がループ状態であるとき，ノード (q, i) および，全て子孫ノード (r, j) では，表 B の値は true になっている．さらに，ノード (q, i) の子孫ノードには，離散状態を同じくするノード (q, k) があり，$A[k] \subseteq A[i]$ である．これは，ループ状態の定義より明らかであることはわかる.

Loop Search は，木の集合 P と表 A,B を構成していく際に，新しいノードが作られるたびに，その親ノードを辿っていきながら，先祖ノードがループ状態であるかどうかのチェックをしている (14–20 行目)．もし，先祖ノードがループ状態であれば，ループ状態を出力するアルゴリズムになっている.

4.3 Zeno 状態であるかどうかの判定

ループ状態 (q, X) が得られたときに，そのループ状態に対応するサイクルパスが $\langle q : \eta_1, \eta_2, \ldots, \eta_i : q \rangle$ であったとする．このサイクルパス一回の実行をまとめて，以下の関数として表現可能である.

- $k : X \to X$
  状態 (q, X) の要素 $(q, \boldsymbol{x})$ を初期値として，ハイブリッドシステムを実行した場合，$\eta_1, \eta_2, \ldots, \eta_i$ と遷移して状態 $(q, k(\boldsymbol{x}))$ になる.
- $l : X \to \mathbb{R}$
  状態 (q, X) の要素 $(q, \boldsymbol{x})$ を初期値として，ハイブリッドシステムを実行した場合，$\eta_1, \eta_2, \ldots, \eta_i$ と遷移して状態 $(q, k(\boldsymbol{x}))$ になるまでの時間 $l(\boldsymbol{x})$.

定理 4.3.1
ハイブリッドシステム $\mathscr{H}$ にループ状態 (q, X) があるとする．このループ状態に対応するサイクルパス一回の実行を表す関数 k，l が，

$$\forall \boldsymbol{x} \in X \left(\frac{l(k(k(\boldsymbol{x})))}{l(k(\boldsymbol{x}))} \leq \frac{l(k(\boldsymbol{x}))}{l(\boldsymbol{x})} < 1 \right)$$

という条件を満たすとき，このハイブリッドシステムは，*Regular Zenoness* を持つ.

証明
ループ状態 (q, X) に対応するサイクルパスが，k 回の離散変化からなるとする．初期状態を $(q, \boldsymbol{x}) \in (q, X)$ としてハイブリッドシステム $\mathscr{H}$ を実行する．このとき，i 回目の離散変化時刻 τ_i は，

$$\tau_{nk} - \tau_{(n-1)k} = l(k^n(\boldsymbol{x}))$$

という関係を満たす．したがって，

$$\lim_{i\to\infty} \tau_i = \sum_{i=0}^{\infty}(\tau_{i+1} - \tau_i) = \sum_{n=0}^{\infty}(\tau_{(n+i)k} - \tau_{nk}) = \sum_{n=0}^{\infty} l(k^{n+1}(\boldsymbol{x}))$$

となる．この無限級数は収束条件

$$\lim_{i\to\infty} \frac{a_{i+1}}{a_i} < 1$$

を満足するので収束する．したがって，ハイブリッドシステム $\mathscr{H}$ は *Zenoness* を持つ． (証明終)

ループ状態が導かれたとき，そのループに対応する関数 k, l を数式処理で求めて，上記の条件を検査することで，ループ状態が Zeno 状態であるかどうかを判定することができる．

5. 例題

本節では例題を用い，提案手法の効果を見る．モデルは，文献 [1] にある Water Tank を用いる．

[Model:Water Tanks]

- $Q = \{q_1, q_2\}$, $E = \{e_{12}, e_{21}\}$
  - $s(e_{12}) = q_1,\ t(e_{12}) = q_2,\ s(e_{21}) = q_2,\ t(e_{21}) = q_1$
- $X = \{(x, y) \in \mathbb{R}^2\}$
- $G = \{G_{e_{12}}, G_{e_{21}}\}$
  - $G_{e_{12}} = \{(x, y) \in \mathbb{R}^2 \mid x = 1\}$
  - $G_{e_{21}} = \{(x, y) \in \mathbb{R}^2 \mid y = 1\}$
- $R = \{R_{e_{12}}, R_{e_{21}}\}$
  - $R_{e_{12}}(x, y) = R_{e_{21}}(x, y) = (x, y)$
- $F = \{f_{q_1}, f_{q_2}\}$
  - $f_{q_1}(x, y) = (-8, 7),\ f_{q_2}(x, y) = (-4, 5)$

このモデルを，Loop Search アルゴリズムに適用して得られる木を図 3 に記す．太丸で表されているノードが $B[i] = true$ であるノードである．これより，$(q_1, A[5])$ および $(q_2, A[6])$ がループ状態であることが分かる．

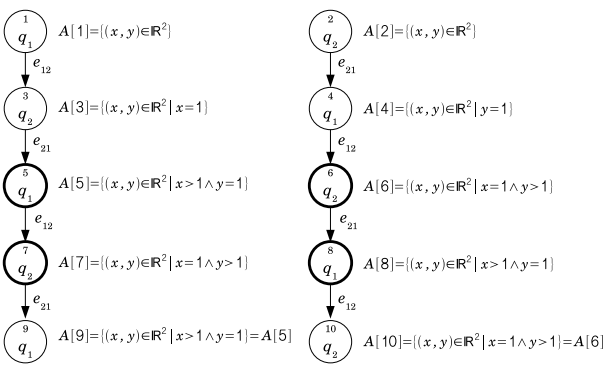

図 3: Loop Search で構成される木 (Water Tank)

ループ状態 $(q_1, A[5])$ について，サイクルパス $\langle q_1 : e_{12}, e_{21} : q_1\rangle$ の実行関数を求めると，

$$k(x, y) = \left(\frac{7x - 8y - 15}{10}, 1\right),\ l(x, y) = \frac{3x + 2y - 5}{10}$$

となる．領域 $A[5]$ では，これらの関数 k 及び l は Zeno 状態判定条件を満たし，Zeno 状態であることが示せた．同様に $(q_2, A[6])$ についても，同様に Zeno 状態であることを示せる．

また，文献 [4] にあるような，離散状態数が 3 つ以上あるモデルや，1 離散状態から複数の状態遷移の可能性があるようなモデルについても提案手法の適用ができた．

6. 関連研究

モデルを限定して Zenoness を判定する手法は，文献 [4] でも提案されている．この手法では，ベクトル場が定数であり，離散変化時の変数の書き換えが無いという条件を定め，グラフ上の閉路を指定して，指定した閉路を構成するベクトル場の定数からなる行列を解析することで，モデルが Zenoness を持つかどうかの判定を行っている．我々の提案手法では，ベクトル場は一階の連立線形微分方程式に広げられ，離散変化時の変数の書き換えについても，多項式で表現できる書き換えまで認め，扱えるクラスが広くなっているといえる．一方で，[4] では，ベクトル場を定数から区間値へ拡張することで，判定できるクラスを広げることができることも示している．これを踏まえて，本提案手法においても，ベクトル場の関数が複雑である場合であっても，複雑な関数を包囲するような複数の関数を考えることで，扱えるクラスの拡大ができるのではないかと考えられる．

7. まとめと今後の課題

本研究では，数式処理と Quantifier Elimination を用いて，ハイブリッドシステムの Zeno 状態を導出する手法を提案した．提案手法により，Regular Zenoness の判定ができるようになる．また，いくつかのモデルについて提案手法の適用を試み，各モデルが陥りうる Zeno 状態の導出ができたことを確認した．

提案手法では，Irregular Zenoness の判定ができないので，今後の課題として不規則に変化する Irregular Zenoness に対しての対応を考察する必要がある．

参考文献

[1] H. Y. Zheng: Simulating Zeno Hybrid Systems Beyond Their Zeno Points, Technical Report No. UCB/EECS-2006-114.

[2] H. Y. Zheng, E. A. Lee and A. D. Ames: "Beyond zeno: Get on with it!", "Hybrid Systems: Computation and Control, Proceedings", 3927, pp. 568–582, 2006.

[3] G. Lafferriere, G.J. Pappas and S. Yovine: Symbolic Reachability Computations for Families of Linear Vector Fields, Journal of Symbolic Computation, 32(3), pp. 231–253, 2001.

[4] M. Heymann, F. Lin, G. Meyer and S. Resmerita: Analysis of Zeno Behaviors in Hybrid Systems, IEEE Transactions on Automatic Control, 50(3), pp. 376–384, 2005.