

多数の離散変化をともなうハイブリッドシステムに対する ループ検出を用いた解析

Analysis of hybrid systems involving numerous discrete changes using loop detection

別納健市^{*1}
Kenichi BETSUNO

松本翔太^{*2}
Shota MATSUMOTO

若槻祐彰^{*2}
Yoshiaki WAKATSUKI

上田和紀^{*2}
Kazunori UEDA

^{*1}早稲田大学大学基幹理工学部情報理工学科
School of Computer Science and Engineering, Waseda University

^{*2}早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻
Graduate School of Fundamental Science and Engineering, Waseda University

Hybrid systems are dynamical systems that include both continuous and discrete changes. Sliding mode, which can be expressed as a hybrid system, makes progress with a large or infinite number of discrete changes within an infinitesimal-width region. It is difficult to analyze hybrid systems with sliding mode through typical simulation, since the time required for simulation increases as the number of discrete changes increases. In particular, the simulation of an infinite number of discrete changes require infinite time. In this paper, we propose a method to analyze such models involving numerical discrete changes by detecting loops of the models. We implemented the method partly into our hybrid system simulator HyLaGI, and conducted analysis of these models.

1. はじめに

ハイブリッドシステム [1] は、離散変化と連続運動の両方を扱う動的システムを指す。ハイブリッドシステムのうち、スライディングモードという特殊な運動をともなうモデルでは、離散変化の回数が増えるため、通常のシミュレーションによる解析が困難となる。本研究は、このスライディングモードを含むハイブリッドシステムに対する解析手法の提案を目的とする。提案手法では、スライディングモードにおけるシステムの運動を一種のループとみなし、そのループの検出を通して、与えられた目標仕様に満たされるかどうかの検証を行う。

2. 背景

2.1 ハイブリッドシステム

時間経過にともない状態や方程式が離散変化、連続変化するシステムをハイブリッドシステムという。ハイブリッドシステムの例としては、撃力を含む物理運動や、スイッチ切替を含む回路システムを挙げることができる。この場合、撃力やスイッチの切替によってシステムの状態が瞬間的に変化する現象は離散変化として表現される。他方、微分方程式にしたがって状態が連続的に変化する現象は連続変化に対応している、こうした表現能力により、物理学や制御工学をはじめとする広範な領域のモデルをハイブリッドシステムとして扱うことができる。

2.2 スライディングモード

本研究では、微小幅の領域内で離散変化を繰り返しつつ進行する挙動をスライディングモードと呼ぶ。ハイブリッドシステムには、スライディングモードを含むモデルが存在する。制御工学のモデルであれば、スライディングモードは一般にチャタリングという形で現れる。また、制御工学には、スライディングモード制御 (SMC) [2] と呼ばれる制御手法が存在し、この手法による制御モデルは、スライディングモードをともなう

ハイブリッドシステムとして記述することができる。ハイブリッドシステムの観点からは、スライディングモードは離散変化が極めて短い周期で発生する運動とみなすことができる。そのようなシステムに対してシミュレーションを試みると、短時間に多数の離散変化が発生するため、シミュレーションに要する実行時間が増大してしまい、場合によってはシミュレーションが終了しない。こうした特徴から、スライディングモードを含むモデルに対しては、通常のシミュレータによる解析が困難となる。

2.2.1 例:スライディングモード制御によるブレーキ制御問題

スライディングモードを含むハイブリッドシステムの例題として、スライディングモード制御を用いて自動車のブレーキ制御を行うモデルについて述べる。このモデルでは、ある初速度のもとで走行している自動車を目的位置で停止させるために制御を行う。自動車に対する制御入力としては、ブレーキを作動させる、あるいは何もしないの 2 通りを想定する。この 2 通りの制御入力は、自動車の位置 x 、速度 x' の関数であるスイッチング関数 $S(x, x')$ の正負に応じて切り替えられる。また、スイッチング関数 $S(x, x') = 0$ により定義される位相平面上の面を切替面とよぶ。このシステムの運動を位相平面上に図示したものを図 1 に示す。

ここでは、自動車の初速度を 100、目的距離を 100 とする。また、制御関数は $S(x, x') = x + x' - 100$ 、ブレーキによって自動車に働く加速度は -50 とする。以下、図 1 上に付された ① から ⑦ により示されるシステムの運動について説明を行う。

- ① システムの初期状態
- ② $S < 0$ となるためブレーキ OFF、すなわち、自動車は等速運動を行う
- ③ システムの状態が切替面に到達し、制御入力が切り替わる。すなわち、ブレーキが ON になる
- ④ 自動車にはブレーキが掛けられているため、等加速度運動を行う

連絡先: 別納健市, 早稲田大学基幹理工学部情報理工学科,
〒169-8555 新宿区大久保 3-4-1 63 号館 5 階 02 号,
03-5286-3340, betsuno(at)ueda.info.waseda.ac.jp

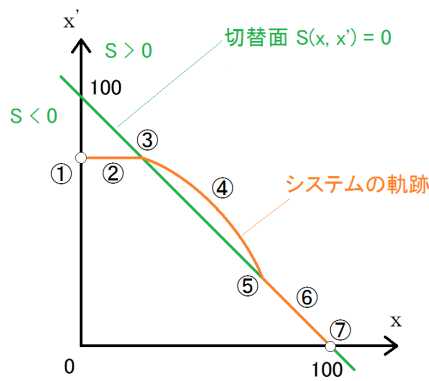


図 1: スライディングモード制御によるブレーキ制御の軌跡

- ⑤ 再び切替面に到達し，制御入力切り替わる
- ⑥ スライディングモードにしたがって運動する
- ⑦ 自動車が目的距離で停止する

⑥において，システムがスライディングモードに入る（切替面に拘束される）のは，切替面のいずれの側においても，位相平面上のベクトルが切替面に近づく向きとなっているためである．図 2 は，このスライディングモードにおける位相平面上のベクトルを図示したものである．

このとき，スライディングモードでは制御入力の切り替えが無数の周波数で行われることになるが，現実のモデルでは制御入力の切り替えの速さは無限ではない．このことをモデルに反映するために，切替面を表わす制御関数から微小値だけ離れた 2 つの関数 S_1, S_2 を設定し，それらによって定義される領域の縁で制御入力の切替が行われるようにする．以下，この領域をスライディングモード領域とよぶ．この場合，スライディングモードにおけるシステムの振る舞いは図 3 のようになる．このとき，微小値 $\epsilon > 0$ を用いて， $S_1 = S, S_2 = S + \epsilon$ としている．

制御関数は，状態変数（の部分集合）から実数への関数である．すなわち，各状態変数からなるベクトルを値とする軌道を $X : T \rightarrow \mathbb{R}^n$ ， X の値域を $A \subset \mathbb{R}^n$ としたとき，制御関数は $S : A \rightarrow \mathbb{R}$ と書ける．そこで， X と S の合成関数 $S \circ X : T \rightarrow \mathbb{R}$ を考えると， $S \circ X$ は（時刻の関数としての）状態変数と同様に扱うことができる．以降，この $S \circ X$ を一種の変数として扱い，制御変数と呼ぶ．また， S_1, S_2 に対応する制御変数をそれぞれ s_1, s_2 と表記する．

3. 提案手法

本研究では，こうしたスライディングモードを含むハイブリッドシステムに対する検証手法の提案を行う．提案手法では，スライディングモードにおけるシステムの挙動を一種のループとみなし，そのループの検出を通してスライディングモードの発生を検出する．本手法の目標は，対象となるシステムがスライディングモードに沿って運動した結果，与えられた目標仕様を満たすことを検証することである．目標仕様は論理式の形で与えられる．以下に，提案する検証手法の全体像を示す．

1. シミュレーションを行い，システムのループを検出する（制御変数に関するループ検査）

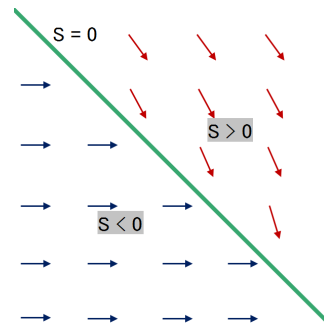


図 2: ブレーキ制御：スライディングモード時の位相平面ベクトル

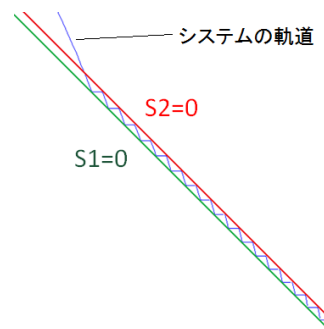


図 3: ブレーキ制御：切替時間を考慮したスライディングモードにおける振る舞い

2. 検出されたループにおいて，システムがスライディングモード領域の内部に存在しつづけることを検証する（ループ不変条件の検証）
3. ループにおいて，いずれ停止条件が満たされることを検証する（ループ停止の検証）
4. システムがスライディングモード領域内部に存在し，かつ停止条件を満たしているならば，目標仕様が満たされることを検証する（モデル仕様の検証）

以降，各ステップに関して解説する．

3.1 制御変数に関するループ検査

ハイブリッドシステムのループ検出の既存手法では，ある時点におけるシステムの状態が過去の状態の繰り返しであることを検出することによりループの検出を行う [3, 4]．ここで，過去の状態の繰り返しとは，新しい状態を過去の状態と比較した際に

1. システムが同じ微分方程式にしたがって運動する
2. 微分方程式の初期値の組のとり得る領域が包含されている

という 2 つの条件を満たすことをいう．

スライディングモードにおけるループでは，条件 2 が一般に満たされない．そこで手法の拡張を行う，条件 2 において，初期値の組のうち制御変数 s_1, s_2 とその導関数の組に関してのみ，とり得る領域が包含されていることを確認することにより，ループの検出を行う（制御変数 s_1, s_2 は，それぞれ制御関数 S_1, S_2 に対応する）．この手法により検出されるループ

ブを制御変数に関するループと呼ぶ。このループにおいては、制御変数の変化は過去の変化の繰り返しとなる。

3.2 ループ不変条件の検証

検出されたループにおいて、 $s_1 \leq 0 \leq s_2$ が不変条件であることを確認する。 $s_1 \leq 0 \leq s_2$ は、システムがスライディングモード領域内に存在することの必要十分条件である ($s_1 + \epsilon = s_2$ のとき)。したがって、 $s_1 \leq 0 \leq s_2$ がループ不変条件であるならば、ループ全体を通してシステムがスライディングモード領域内に存在することが示される。 $s_1 \leq 0 \leq s_2$ がループ不変条件であることを調べるためには、ループ 1 周期分に注目し、その 1 周期において、 $s_1 \leq 0 \leq s_2$ が成立しつづけるか否かを調べる。成立していた場合、制御変数に関するループの性質から、 $s_1 \leq 0 \leq s_2$ がそのループ全体で満たされる、すなわちループ不変条件であることが示される。

3.3 ループ停止の検証

検出されたループにおいて、1 周期ごとにシステムが停止条件に近づき、かつその差分の和が収束しないことを検証する。

3.4 モデルの仕様の検証

このステップでは、システムがスライディングモード領域内部に存在し、かつシステムの状態変数がループ停止条件を満たしている場合に、必ず目標仕様が満たされるということを確認する。このステップは、システムの記述を静的に解析することによって行われる。システムがスライディングモード領域の内部に存在するのは、2 つの制御変数の符号が異なるか、一方が 0 となる場合である。これは、 $s_1 + \epsilon = s_2$ のとき、 $s_1 \leq 0 \leq s_2$ という論理式で表わされる。したがって、このステップでは ($s_1 \leq 0 \leq s_2$) (ループ停止条件) (目標仕様) という論理式が恒真であることを検証する。

4. HydLa 処理系への実装

提案手法は前述した 4 つのステップからなるが、その基本となるループ検出ステップについて、ハイブリッドシステムのシミュレータである HyLaGI [5] に対する実装を行った。HyLaGI は、ハイブリッドシステムの処理系であり、入力としてハイブリッドシステムのモデリング言語 HydLa [6] のプログラムを受け取る。HyLaGI は、ソルバ部分に数式処理を採用することにより、精度保証付きのシミュレーションや変数にパラメータを含むモデルに対するシミュレーションを実現している。

4.1 状態包含検査

HyLaGI には、与えられたプログラムに対してシミュレーションを行い、そのモデルに対応するハイブリッドオートマトンを出力する機能が存在する [3, 4]。その一環として 2 つの状態間の包含関係を検出するためのアルゴリズムが実装されている。このアルゴリズムは、本稿で提案したスライディングモードにおけるループ検出手法の実装のベースとなるものであるため、以下にその解説を行う。

HyLaGI では、ハイブリッドシステムにおける離散変化と連続運動をまとめてフェーズと呼ぶ。HyLaGI によるシミュレーション実行の流れにおいては、それまで得られた各フェーズについて、採用される制約、成立するガード条件 (条件付き制約の前件に対応する論理式)、および各変数に対応する数式の情報が保存されている。システムのダイナミクスは、そのフェーズにおいて有効となる制約によって決定される。したがって、システムの状態変数がしたがう微分方程式は、採用される制約の集合および成立するガード条件の集合によって確定する。2 つのフェーズに注目したとき、状態変数がしたがう微分方程式

が等しいための必要十分条件は、それらのフェーズにおいて採用される制約の集合および成立するガード条件の集合が一致することである。HyLaGI における包含関係の判定は、新しいフェーズを過去のフェーズと比較した際に

1. 採用される制約の集合が等しい
2. 成立するガード条件が等しい
3. 微分方程式の初期値の組のとり得る領域が包含されている

という条件が満たされるかどうかをチェックすることにより行われる。今回実装した制御変数に関するループ検査においては、上記の 3. について、任意の状態変数ではなく、制御変数についてのみ、初期値の組の包含関係を比較するように変更を行った。

5. 実験

前述の実装を用い、スライディングモードをとまなうモデルに対してシミュレーションを通したループの検出を行った。使用したモデルは、スライディングモード制御によるブレーキ制御の問題である。シミュレーションの結果、システムの振る舞いは表 1 に示すように初速度に応じて 3 ケースに分かれた。

表 1: SMC を用いたブレーキ制御問題：初速度による分岐

初速度 x' の値	ループの有無
$(0, 2^{1/2} * (-35) + 50)$	ループを形成せず
$2^{1/2} * (-35) + 50$	ループを形成せず
$(2^{1/2} * (-35) + 50), 80)$	ループを形成した

全ケースのうち、自動車の初速度が $(0, 2^{1/2} * (-35) + 50)$ を満たす 2 つのケースでは、最初のブレーキによって自動車が停止したため、システムはループを形成することがなかった。一方、1 回目のブレーキで自動車が停止しなかったケースにおいては、1 回目と 2 回目のブレーキ開始時において、制御変数に関する包含関係が成立したため、ループが検出された。この場合における状態遷移図を図 4 に示す。

図 4 において、Phase 1 は初期状態に対応し、その後 Phase 2 で自動車は等速運動を行い、Phase 3 において 1 回目のブレーキを掛けている。Phase 4 はブレーキを掛けた状態での自動車の走行 (連続変化)、Phase 5 はブレーキを離れた瞬間 (離散変化) に対応している。Phase 12 から Phase 3 にかけて遷移の循環が見てとれるが、これは Phase 12 (ブレーキを離れた状態での自動車の走行) から次のフェーズ (2 回目のブレーキを掛ける瞬間) に移る際に、新しいフェーズが過去のフェーズ (1 回目のブレーキを掛けた Phase 3) に包含されていることが検出されたことを示している。ここで成立したのは、任意の変数に対する包含関係ではなく、制御変数に対する包含関係であり、従来の手法では検出することができないパターンである。

本モデルでは、シミュレーションの際に自動車の初速度を区間値 $(0, 80)$ として抽象化した。これは、本モデルにおいては、ブレーキを掛けることに制御変数の 1 階微分の値が減少していくため、初速度をある特定の値に設定した場合に包含関係が成立せず、ループの条件が満たされないためである。例として、自動車の初速度を 80 に設定した場合の 1 回目と 2 回目のブレーキ開始時における変数表を表 2 に示す。

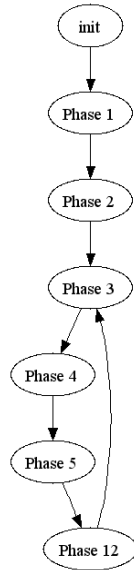


図 4: SMC を用いたブレーキ制御問題におけるループ図

表 2: SMC を用いたブレーキ制御問題：初速度 80 の場合

変数名	1 回目	2 回目
s1	1/2	1/2
s2	0	0
s1'	80	$50+38^{(1/2)*(-5)}$
s2'	80	$50+38^{(1/2)*(-5)}$
x	41/2	$(101+10*38^{(1/2)})*1/2$
x'	80	$(38^{(1/2)}+(-10))*(-5)$

この表において、制御変数 (s1, s2) の 1 回微分の値はそれぞれ特定の値をとっているが、1 回目と 2 回目のブレーキ開始時で異なる値をとるため、包含関係が成立しないことがわかる。このような場合は、スライディングモードにおけるループの検出がされないため、モデルを適宜抽象化する必要がある。

6. まとめと今後の課題

本研究では、スライディングモードを含むハイブリッドシステムに対して、ループ不変条件に着目した解析手法を提案した。また、ハイブリッド処理系である HyLaGI に拡張を施し、スライディングモードに付随するループを検出する機能を実装した。今後の課題として、検出されたループがいずれ停止することを検証するアルゴリズムの設計が挙げられる。この検証に関しては、個々のモデルの複雑さに大きく依存する。したがって、自動検証が難しいモデルでは、ユーザから補助的な情報を求めるといったインタラクティブな手法が考えられる。また、モデルによっては、ループを検出するためにシステムの初期状態の抽象化が必要となる場合がある。現時点では、抽象化は手動で行う必要があるが、手動で適切な抽象化を行うことが必ずしも容易であるとは限らない。そのため、初期値を適切に抽象化する自動抽象化によって、検証がより容易になるものと考えられる。その他、複数の切替面を含むモデルに対する拡張や、区間計算アルゴリズム [7, 8] との併用が挙げられる。

参考文献

- [1] Lunze, J. : Handbook of Hybrid Systems Control: Theory, Tools, Applications, Cambridge University Press, 2009.
- [2] 野波健蔵, 田宏奇: スライディングモード制御, コロナ社, 1994.
- [3] 竹口輝, 和田亮, 松本翔太, 細部博史, 上田和紀: ハイブリッド制約言語プログラムのハイブリッドオートマトンへの変換アルゴリズム, 日本ソフトウェア科学会第 29 回大会 2A-3, 2012.
- [4] 若槻祐彰, 松本翔太, 上田和紀: ハイブリッド制約処理系 HyLaGI における LTL モデル検査, 人工知能学会第 30 回全国大会, 1F3-1, 2016 (発表予定).
- [5] 松本翔太, 上田和紀: ハイブリッド制約言語 HydLa の記号実行シミュレータ Hyrose, コンピュータソフトウェア, Vol.30, No.4 (2013), pp.18-35.
- [6] 上田和紀, 細部博史, 石井大輔: ハイブリッド制約言語 HydLa の宣言的意味論, コンピュータソフトウェア, Vol.28, No.1 (2011), pp.306-311.
- [7] 和田努, 松本翔太, 上田和紀: ハイブリッド制約言語 HydLa 処理系における数式処理と区間計算を組み合わせたシミュレーション実行, 人工知能学会第 29 回全国大会, 1E3-3, 2015.
- [8] 松本翔太, 上田和紀: ハイブリッドシステムのシミュレーションにおける精度保証数値計算と数式処理との連携, 電子情報通信学会 ソフトウェアサイエンス研究会 (SS2015-60), Vol. 115, No.419 (2016), pp. 93-98.