3O1-3in

ハイブリッド制約言語HydLaを用いたハイブリッドシステムの解析

Analyses of Hybrid Systems with the Hybrid Constraint Language HydLa

松本翔太[*1] 河野文彦[*1] 上田和紀[*2]
Shota MATSUMOTO Fumihiko KONO Kazunori UEDA

[*1]早稲田大学大学院基幹理工学研究科情報理工学専攻
Graduate School of Fundamental Science and Engineering, Waseda University

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

Hybrid systems are dynamical systems with continuous changes and discrete changes of states. Hybrid systems are appliable to diverse fields including physics. HydLa is a declarative language for describing hybrid systems using constraints. We have developed tools for HydLa, Hyrose and HIDE, for the purpose of analysis and verification of hybrid systems. Hyrose is a symbolic simulator of HydLa and it calculates trajectories symbolically. Hyrose can also handle hybrid systems with parameters. HIDE is an integrated development environment of HydLa that can visualize trajectories calculated by Hyrose. In this presentation, we introduce examples of analyses of hybrid systems using HydLa and its tools.

1. はじめに

ハイブリッドシステム [5] とは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システムを指す．ハイブリッドシステムは物理学や制御工学，生命工学などさまざまな分野に適用可能な概念である．HydLa はハイブリッドシステムを制約によって記述する宣言型言語であり，システムを表現する数式や論理式をそのまま制約としてプログラムに記述できるのが特徴である．我々は HydLa を用いたハイブリッドシステムの解析を目指し，HydLa の記号実行シミュレータ Hyrose[6] や，統合開発環境 HIDE を開発してきた．本論文ではこれらのツールを用いたハイブリッドシステムの解析を紹介する．

2. HydLa

HydLa は制約を用いてハイブリッドシステムのモデリングを行う宣言型言語であり，その宣言的意味は文献 [8] において与えられている．HydLa ではシステムが満たすべき個々の条件を，等式・不等式・微分方程式を原子論理式とする時相論理式で記述することでモデリングを行う．HydLa プログラムは，このように時相論理式で記述した制約（これを「制約モジュール」と呼ぶ）の集合である．各制約モジュールの間には優先順位を設けることができる．システムの例外的な動作に，通常時の動作より高い優先度を設けることで簡潔な振る舞いの記述を行うことを目標としている．HydLa プログラム中に出現する変数は，時刻についての関数変数であり，それらの変数の振る舞いのうち，プログラムの仕様を満たすものがプログラムの宣言的意味となる．

ハイブリッドシステムのモデリングツールは，ハイブリッドオートマトン [4] を用いてシステムを表現するものが多い [1] ．一方，制約による記述はオートマトンによる記述と比較して，システムの全状態を列挙する必要が無いため，記述量を減らしやすい．しかし HydLa のように制約を用いてハイブリッドシステムを扱う言語としては，他に Hybrid cc [2] があるが，HydLa では制約間に優先順位をつけることが可能となっている点が異なる．

連絡先: 松本翔太, 早稲田大学大学院基幹理工学研究科情報理工学専攻, 〒 169-8555 新宿区大久保 3-4-1 63 号館 5 階 02 号, 03-5286-3340, matsusho(at)ueda.info.waseda.ac.jp

3. HydLa の記号実行シミュレータ Hyrose

Hyrose [*1] は HydLa プログラムのシミュレータであり，ハイブリッドシステムの性質の理解や検証を目指して開発されている．開発言語は C++であり，現在の総コード行数は 3 万行程度の規模になっている．現在の Hyrose の特徴として，以下が挙げられる．

- 数値計算ではなく数式処理を用いることで，シミュレーションの結果から誤差を排除している．(制約求解のバックエンドとして，数式処理ソフトウェアである Mathematica [9] と REDUCE[3] のいずれか一つを利用する)
- シミュレーションの記号実行を行うことができ，パラメータを持つシステムを扱うことが可能である．
- パラメータの値によってシステムの振る舞いが定性的に変化する場合，自動場合分けに基づく全解探索実行を行う．
- 有限時間におけるシステムの性質検証を自動で行うことができる (5.4 節).

現実のシステムを適切にシミュレーションするためには，モデリングにともなう観測誤差や，システムの特性の環境や経年による変化等を考慮する必要がある．Hyrose はこうした不確定性も記号実行の枠組みの中でパラメータとして扱うことで，現実のハイブリッドシステムの高信頼なシミュレーションを可能とすることを目標としている．

4. HydLa 統合開発環境 HIDE

HIDE は Hyrose をバックエンドとして動作する HydLa の統合開発環境であり，HydLa ユーザへのグラフィカルな補助

*1 http://www.ueda.info.waseda.ac.jp/hydla/

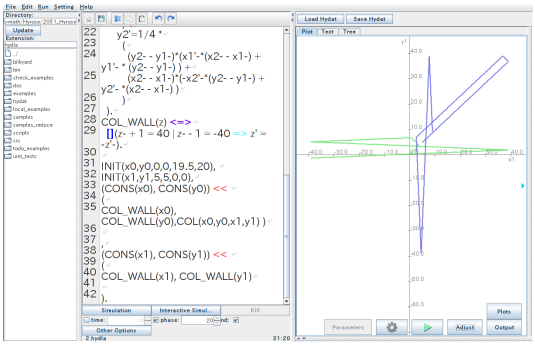

図 1: HIDE のスクリーンショット

```
INIT <=> x = 0 & y = 1 & x' = 3 & 0 <= y' <= 10.
FALL <=> [](y'' = -10).
BOUNCE <=> [](y- = 0
    => y' = -0.8 * y'- & x' = 0.8 * x'-).
WIND <=> [](x'' = -0.1 * y).

INIT, (FALL, WIND) << BOUNCE.
```

図 2: 例題の HydLa プログラム

を行うことを目的として開発されている．実装言語は Java であり，現在は総コード行数 12000 行程度の規模になっている．

図 1 に HIDE のスクリーンショットを示す．HydLa の専用エディタ（画面の左部分）としての機能のほか，Hyrose によるシミュレーション結果の可視化（画面の右部分）を行うことができるのが主な特徴である．シミュレーション結果の可視化に関しては，2 次元グラフによる解軌道の描画のほか，時間経過に従ったアニメーションの描画も可能になっており，ハイブリッドシステムの振る舞いの理解の補助を行うことができる．

5. ハイブリッドシステムの解析例

本節では，HydLa，Hyrose，HIDE を用いたハイブリッドシステムの解析例を示す．例題として向かい風が吹いている空間で，ボールを斜方投射するというモデルを考える．

5.1 HydLa プログラム

図 2 に HydLa プログラムを示す．図 2 のプログラムにおいて，x はボールの水平方向の位置を，y はボール垂直方向の位置を示す変数になっている．x' のように，変数値に「'」記号がついたものは変数の時間微分を表しており，x' と y' はそれぞれボールの水平方向の速度と垂直方向の速度を表している．

このプログラム内では，INIT，FALL，WIND，BOUNCE の 4 つの制約が定義されており，INIT はボールの初期位置と初期速度を表している．垂直方向の初期速度に関しては $0 < y' < 10$ という不等式で記述してあり，モデルに不確定性を与えている．FALL はボールの垂直方向の加速度を -10 とする制約であり，自由落下を意味している．BOUNCE は地面とボールとの衝突を意味する条件付き制約である．衝突時にボールの水平方向の速度が 4/5 倍，垂直方向の速度が $-4/5$ 倍になることを表している．x-のように，変数の後に-を加えた場合，その変数の左極限値を意味する．WIND はボールの水平方向の加速度に関する制約であり，向かい風による水平方向の速度の減少を意味している．本例題の場合，ボールの位置が高ければ高いほど向かい風が強く影響するものとしている．最後にプログラムの

```
---------IP 2---------
{FALL$0, WIND$0, BOUNCE$0}
time      : 0->(parameter[y, 1, 1]
              +(20+parameter[y, 1, 1]^2)^(1/2))*1/10
x         : t*(360+5*t^3+t*(-6)
              +t^2*(-2)*parameter[y, 1, 1])*1/120
y         : 1+t^2*(-5)+t*parameter[y, 1, 1]
x'        : (180+10*t^3+t*(-6)
              +t^2*(-3)*parameter[y, 1, 1])*1/60
y'        : t*(-10)+parameter[y, 1, 1]
x''       : (5*t^2+(-1)
              +t*(-1)*parameter[y, 1, 1])*1/10
y''       : -10
```

図 3: Hyrose による出力例

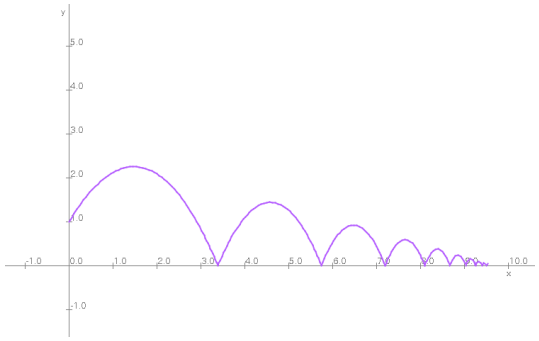

図 4: 図 2 の軌道例

最下部において，BOUNCE に FALL や WIND より強い優先度を与えて宣言することで，BOUNCE が地面との衝突時に優先的に採用されるよう記述している．

5.2 Hyrose による出力

以上のプログラムについて，フェーズ数は 30，モデルの時刻は 10 までという条件で Hyrose によるシミュレーションを行った．図 3 に，Hyrose による例題実行時の出力の一部を示す．この出力は，Hyrose のシミュレーションの 1 フェーズに対応しており，最上部の IP 2 は，この出力が ID2 のインターバルフェーズに対応することを意味している．続く {FALL$0, WIND$0, BOUNCE$0} は，このフェーズで採用された制約集合を意味しており，制約名の後の$0 は同名の制約を複数宣言した場合，それらの識別に利用するための番号である．time は IP 2 の時間を表しており，IP 2 は時刻 0 から (parameter[y, 1, 1] + (20 + parameter[y, 1, 1]^2)$^{1/2}$) * 1/10 まで続くことを意味している．parameter[y, 1, 1] は，「y」の「1」階微分の第「1」フェーズにおける値を表す記号定数である．x 以下は IP 2 での各変数の振る舞いを示しており，t はモデルの時刻に対応する．以上のようなフェーズごとの出力を，各場合ごとに並べたものが Hyrose の出力となっている．

5.3 HIDE によるグラフ化

HIDE を用いて Hyrose の実行結果をグラフ化したものを図 4 と図 5 に示す．これらのグラフの横軸と縦軸がそれぞれボールの水平方向と垂直方向の位置に対応している．

図 4 は y' の初期値を 5 とした場合の描画結果であり，解軌道の概形を見ることができる．図 5 は y' の初期値を 0 から 10 まで 1 刻みで変化させて解軌道を描画しており，初期値が解軌道全体に与える影響を表したものになっている．なお描画

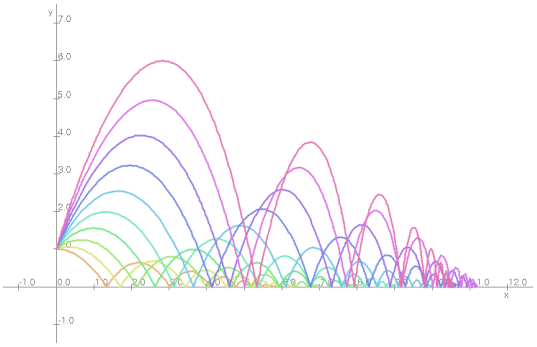

図 5: 図 2 の軌道群

の都合上，HIDE によるグラフ化は有限数の解軌道しか描画できないが，Hyrose による解軌道の出力自体は，0 から 10 までの範囲の任意の実数を初期値とした場合の無限個の解軌道を意味している．HIDE の機能としては以上のグラフ化の他，アニメーション機能も存在している．アニメーションを用いると，時刻の経過に従った変数値の振る舞いをより直感的に理解できるようになり，図 5 のような複数の解軌道が重なりあう場合でも，各軌道ごとの振る舞いを理解しやすくなる．

5.4 有界モデル検査によるパラメータ解析

前述のモデルについて，水平方向の位置が 10 を超えるための初期値の条件を求めることを考える．Hyrose は有界モデル検査機能を有しており，これを用いることでそのような条件を自動的に計算することができる．Hyrose の有界モデル検査機能は，シミュレーションによって求めた解軌道が指定された条件を満たすかどうかを調べる機能である．プログラム内に `ASSERT(y > 0)` のようにシステムの満たすべき条件を記述し，シミュレータを実行することで検査が行われる．条件に違反した場合はその旨を出力し，条件に違反するまでにシステムがどのような振舞いをしていたかをあわせて出力する．もし記号定数の値によって条件に違反するかどうかが変化する場合は，違反する場合と違反しない場合とに分け，違反しない場合についてはシミュレーションを続行する．

本例題の場合は，プログラム内に `ASSERT(x <= 10)`．という 1 行を含めることで `x` が 10 を超える場合を Hyrose が検出するようになる．実際に図 2 のプログラムに `ASSERT` 文を加えて実行すると，初期値が 5.43 を超えるとボールの水平方向の位置が 10 を超えるという事がわかった．

6. まとめと今後の課題

HydLa とその処理系 Hyrose および統合開発環境 HIDE を用いることで，数式処理に基づいたハイブリッドシステムのシミュレーションや有界モデル検査に加え，パラメータ解析やその結果の可視化を行うことができる．

今後の課題としては，数式処理と区間演算とを組み合わせることによる Hyrose のシミュレーション能力の向上が挙げられる．また，HIDE によるパラメータによる場合分けの可視化や，3 次元グラフの描画など，シミュレーション結果の理解を補助する機能も充実させていきたい．Hyrose を用いた無限時間に対するモデル検査機能 [7] も考えられており，今後はその活用や可視化を行うことも視野に入れている．

謝辞 本研究を行うにあたり，貴重な意見を頂いた上田研究室 HydLa 班の皆様に感謝する．また，本研究の一部は，科学研究費補助金 (基盤研究 (B) 23300011) の補助を得て行った．

参考文献

[1] Carloni, L., Passerone, R., Pinto, A. and Sangiovanni-Vincentelli, A. L. : Languages and Tools for Hybrid Systems Design, Foundations and Trends in Design Automation, Vol. 1 No. 1, 2006, pp. 1–204.

[2] Gupta, V., Jagadeesan, R., Saraswat, V. and Bobrow, D.: Programming in Hybrid Constraint Languages, in Hybrid Systems II, LNCS 999, Springer, 1995, pp. 226–251.

[3] Hearn, A. C., REDUCE, http://reduce-algebra.com/.

[4] Henzinger, T. : The Theory of Hybrid Automata, in Proc. LICS '96, IEEE Computer Society Press, 1996, pp. 278–292.

[5] Lunze, J. : Handbook of Hybrid Systems Control: Theory, Tools, Applications, Cambridge University Press, 2009.

[6] 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. 11，pp.18-35.

[7] 竹口輝：HydLa プログラムの抽象実行によるハイブリッドシステムのモデル検査, 早稲田大学大学院基幹理工学研究科, 修士論文, 2014.

[8] 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.

[9] Wolfram Research, Inc., Mathematica, http://www.wolfram.co.jp/products/mathematica/index.html.