

ハイブリッドシステムの制御と混合整数計画問題

Control of Hybrid Dynamical Systems and Mixed Integer Programming Problems

井村 順一
Jun-ichi Imura

東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan
imura@mei.titech.ac.jp

概要

ハイブリッドダイナミカルシステムとは離散変数と連続変数が混在したダイナミカルシステムのことをいう。近年、制御工学の分野で、このようなシステムに対するモデリング、解析、制御系設計に関する研究が活発になされている。解析や制御問題の多くは一般に混合整数計画問題に帰着されるが、実用化にあたり課題は多い。本稿では、ハイブリッドダイナミカルシステムの制御に関する基本的な考え方を概説した後、制御において混合整数計画問題を解く際の課題について整理する。また、これらの課題に対する筆者らの試みについても紹介する。

Keywords: ハイブリッドダイナミカルシステム (hybrid dynamical system), 混合整数計画問題 (mixed integer programming problem), モデル予測制御 (model predictive control), 可制御性問題 (controllability problem)

1 はじめに

ハイブリッドダイナミカルシステム（以下、略してハイブリッドシステムと呼ぶ）とは、離散ダイナミクスと連続ダイナミクスが混在したシステムのことである。離散変数と連続変数が混在したシステム、あるいは、より踏み込んで、事象駆動システムと時間駆動システムが混在したシステムなどと説明されることもある。「ハイブリッドシステム」という用語および概念は新しいものではなく、1966年に Witsenhausen[1] が既に用いている。ところが、それ以降の20数年間は研究の進展はなく、1990年代に入り、それまで離散事象システムを研究してきたコンピュータ科学分野の研究者が、実世界とのインターラクティブを有する、すなわち外界からの連続信号を含む離散事象システム（リアクティブシステム）や、時間概念を取り入れた時間付きオートマタなどの検証問題を研究し始めたのを契機に、そして、それと同調するようにシステム制御分野でも、ロボスタ制御の隆盛を横目に、新たなシステム論のブレイクスルーとして、ハイブリッドシステムのモデリングや制御に興味を抱くようになって生まれた新しい学問である。2000年頃を境に非常に多くのジャーナルで特集が組まれ、多くの研究者が精力的に研究するようになり、現在では、システム制御分野において欠かすことのできない研究テーマの一つになっている。

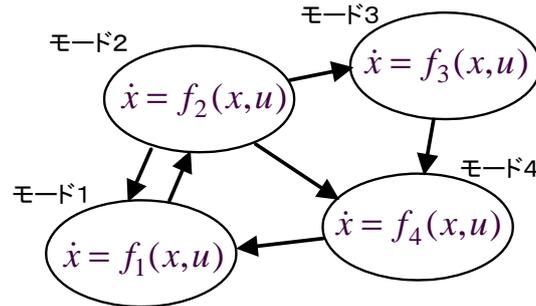


図 1: ハイブリッドシステムのイメージ

本稿では、こうしたハイブリッドシステムに関する研究について、特に制御問題を中心に解説する。ハイブリッドシステムのモデルについて簡単に説明した後、その制御問題が混合整数計画問題に帰着されることを紹介し、制御において現われる混合整数計画問題を解く際の課題を整理する。また、それらの課題に対して、筆者らが近年挑んでいるアプローチを具体的に紹介し、今後の方向性について言及する。

2 ハイブリッドダイナミカルシステム

2.1 ハイブリッドシステムのイメージ

ハイブリッドシステムとは、図1に示すように、オートマトンの各ノードに常微分方程式を割り当てたものであると考えると理解しやすい。オートマトンによって離散ダイナミクス（モード1ならばモード2に遷移できるといったようにモード遷移に記憶を要するのでダイナミクスである）を表現し、各ノードに割り当てられた常微分方程式によって連続ダイナミクスを表現している。離散ダイナミクスの状態を離散状態（モード）、連続ダイナミクスの状態を連続状態と呼ぶ。車のマニュアルシフトのように外的要因により、あるいは、温度や位置などの連続状態があるしきい値を超えるなどの内的要因によりモードが切り替わり、それに伴い、常微分方程式が切り替わったり、あるいは、その解がジャンプしたりする。

連続ダイナミクスは、力学系の運動方程式や化学反応式などに相当する。一方、離散ダイナミクスは、不連続な物理現象により生じるもの（物理的離散事象）とコンピュータプログラムなどにより実現される論理によるもの（論理的離散事象）の2種類がある。物理的離散事象を含むシステムとして、例えば、ロボットの2足歩行やハンドによる物体操りが挙げられる。歩行では、1本足と2本足の立位相が交互に切り替わり、さらに1本足から2本足への遷移の際に足裏の速度が瞬時に0になるので、解のジャンプ現象が生じる。これを図1のように表現するのは容易であろう。論理的離散事象を含むシステムは、コンピュータプログラムによって実現される論理が実システムと混在している状況において見られる。例えば、化学プラントにおける反応炉を想像してみよう。立ち上げモード、定常モード、緊急モードなど様々なモードが存在し、それらによりオートマトン（論理的関係）が構成され、一方、各モードごとに、液体注入用のバルブが開閉したり、あるいはヒーターがON/OFFしたりすることによって微分方程式も切り替わる。さらに制御入力としては、タンクへの液量のような連続値だけでなく、ヒーターなどのON/OFFスイッチのような離散値も考えられる。

センサ出力も同様に、タンクの水位や温度などの連続値だけでなく、温度などを very low, low, normal, high, very high のようにシンボル（離散値）として表現する場合も考えられる。

このように、不連続な物理現象やコンピュータによる論理を含む実システムは、自動車、航空機、ロボットなどの機械システム、電気機器、鉄鋼や化学プラント、通信ネットワーク、生体ネットワークなど身近に多数存在する。このようなシステムの制御系設計では、これまでは、システムを局所的に捉え、連続か離散のどちらかの観点のみに注目し、他方は無視するか、あるいは直感や経験に基づいてヒューリスティックに設計してきたのが現状である。連続と離散が混在した系として如何に適切にモデリングするか？、如何に解析するか？、そして如何に最適な制御系を設計するか？が研究課題である。しかしながら、このような課題が現段階で完全に解かれているわけではない。ハイブリッドシステムの理論は、その核が漸く見え始めてきた段階と言って良いであろう。詳細は文献 [2, 3, 4] などを参照されたい。

2.2 ハイブリッドシステムのモデル

一般的なハイブリッドシステムモデルは

$$\begin{cases} \dot{x}(t) = f(I(t), x(t), u(t)) \\ I(t) = g(I(t_-), x(t_-), u(t)) \\ x(t) = h(I(t_-), x(t_-), u(t)) \\ y(t) = \psi(I(t), x(t), u(t)) \end{cases} \quad (1)$$

で与えられる [5, 6]. ここで、 $t \in \mathcal{R}$ は連続時間、 $I \in \mathcal{N}^\ell$ は、有限集合 \mathcal{N} の直積上に定義された離散状態（モード）、 $x \in \mathcal{R}^n$ は連続状態、 $u \in \mathcal{N}^{m_1} \times \mathcal{R}^{m_2}$ は制御入力で、 $y \in \mathcal{N}^{p_1} \times \mathcal{R}^{p_2}$ は出力である。また、 f, g, h, ψ は、 $\mathcal{N}^\ell \times \mathcal{R}^n \times \mathcal{R}^m$ から、それぞれ \mathcal{R}^n , \mathcal{N}^ℓ , \mathcal{R}^n , $\mathcal{N}^{p_1} \times \mathcal{R}^{p_2}$ への写像である。 (I, x) の組で、単に「状態」、もしくは「ハイブリッド状態」などと呼ばれ、 \mathcal{N}^ℓ の要素数は「モード数」と呼ばれる。

(1) 式において、第 1 式は、 $I \in \mathcal{N}^\ell$ ごとに通常の常微分方程式であり、連続ダイナミクスを表す。また、第 2 式は離散状態の遷移（モードの切換え）を表す。ここで t_- は時刻 t でモードが切換わる直前を意味し、 $x(t_-) := \lim_{\tau \rightarrow t-0} x(\tau)$ などと定義する。第 2 式は、例えば、 $I_+ \in \mathcal{N}^\ell$ と $\mathcal{L}_{I_+} \subseteq \mathcal{N}^\ell \times \mathcal{R}^n \times \mathcal{R}^m$ によって定義される if-then ルール

$$I(t) = \underbrace{I_+}_{g(I(t_-), x(t_-), u(t))} \quad \text{if } (I(t_-), x(t_-), u(t)) \in \mathcal{L}_{I_+} \quad (2)$$

が挙げられる。第 3 式は連続状態の不連続変化（状態のジャンプ）を定義し、第 4 式は出力方程式である。ここで、モードの切換えと状態のジャンプは、連続状態 x によるものと入力 u によるものに区別される。前者は、それぞれ「自律切換え (autonomous switching)」, 「自律ジャンプ (autonomous jump)」と呼ばれ、後者は「制御切換え (controlled switching)」, 「制御ジャンプ (controlled jump)」と呼ばれる。なお、(1) 式に対応する離散時間モデルも同様に定義され、その場合、第 1 式を差分方程式に置き換えるなどすればよい。

より具体化したモデルとしては、区分的アファイン（動的）システム (Piecewise Affine System)[7, 8], 混合論理動的システム (Mixed Logic Dynamical System)[9], 相補性（動的）システム (Complementarity Systems)[10] などが挙げられる。以下では、区分的アファインシ

システム（以下、PWA システムという）と混合論理的動的システム（以下、MLD システムという）について簡単に紹介する。

下記の離散時間 PWA システムは、ハイブリッドシステムモデルの中で最も標準的でありこれに対して様々な研究がなされている。

$$x(t+1) = A_{I(t)}x(t) + a_{I(t)} + B_{I(t)}u(t), \quad \text{if } (x(t), u(t)) \in \mathcal{S}_{I(t)} \quad (3)$$

ここで、 $t \in \{0, 1, \dots\}$ は離散時間、 $x \in \mathcal{R}^n$ と $I \in \mathcal{N}^\ell$ はそれぞれ連続状態と離散状態（モード）であり、 $u \in \mathcal{R}^m$ は連続入力、 $\mathcal{S}_I \subseteq \mathcal{R}^n \times \mathcal{R}^m$ はモード I を割り当てる集合である。なお、 \mathcal{S}_I は連続状態 x に関して与えられるのが一般的である。 $(x(t), u(t))$ の値によってモード I が切り変わり、従って連続ダイナミクスも切り替わる。なお、出力方程式は必要に応じて表記することにする。

一方、離散時間 MLD システムは

$$x(t+1) = Ax(t) + B_1u(t) + B_2z(t) + B_3\delta(t) \quad (4)$$

$$Cx(t) + D_1u(t) + D_2z(t) + D_3\delta(t) \leq E \quad (5)$$

で与えられる。ここで、 $x \in \mathcal{R}^{n_c} \times \{0, 1\}^{n_d}$ は状態、 $u \in \mathcal{R}^{m_{1c}} \times \{0, 1\}^{m_{1d}}$ は入力、 $z \in \mathcal{R}^{m_2}$ 、 $\delta \in \{0, 1\}^{m_3}$ はそれぞれ連続値と離散値の補助変数である。不等式 (5) は p 本、すなわち $C \in \mathcal{R}^{p \times n}$ 、 $D_i \in \mathcal{R}^{p \times m_i}$ ($i = 1, 2, 3$)、 $E \in \mathcal{R}^p$ であり、 \leq は要素ごとの大小関係を表す。補助変数 z と δ は、混合整数不等式 (5) において x と u が与えられたら一意に定まるとする。すなわち、そもそも、 z と δ は x と u の（論理条件などから決まる）非線形関数の値として $(z, \delta) = h(x, u)$ で与えられているものであり、この関数を混合整数不等式で表現していると解釈できる。このシステム表現の特徴は、等式も不等式も（バイナリ変数であることを除けば）すべて線形の形式で表現されていることにある。下記の簡単な例でこの表現方法を説明する。

例 2.1 有界閉集合 $\mathcal{X} \subseteq \mathcal{R}^n$ 上で定義されたモード数 2 の離散時間 PWA システム

$$x(t+1) = \begin{cases} A_0x(t) + a + Bu(t) & \text{if } x(t) \in \mathcal{S}_0 \\ A_1x(t) + a + Bu(t) & \text{if } x(t) \in \mathcal{S}_1 \end{cases} \quad (6)$$

を MLD システムとして表現する。ただし、 $a \in \mathcal{R}^n$ 、 $B \in \mathcal{R}^{n \times m}$ 、 $\mathcal{S}_0 := \{x \in \mathcal{X} | Cx < 0\}$ 、 $\mathcal{S}_1 := \{x \in \mathcal{X} | Cx \geq 0\}$ 、 $C \in \mathcal{R}^{1 \times n}$ とする。

補助変数 $\delta \in \mathcal{R}$ を

$$\delta(t) = \begin{cases} 0 & \text{if } Cx(t) < 0 \\ 1 & \text{if } Cx(t) \geq 0 \end{cases} \quad (7)$$

とおくと、(6) 式はつぎのように表現できる。

$$x(t+1) = A_0x(t) + \delta(t)(A_1 - A_0)x(t) + a + Bu(t) \quad (8)$$

そこで $z(t) = \delta(t)x(t)$ の変数変換、および (7) 式を 0-1 を含む不等式で表現することによって、(7) 式と (8) 式は、つぎの近似的に等価な表現に変換できる。

$$\begin{cases} x(t+1) = A_0x(t) + (A_1 - A_0)z(t) + a + Bu(t) \\ \xi_{\min}(1 - \delta(t)) \leq Cx(t) \leq \xi_{\max}\delta(t) - (\delta(t) - 1)\varepsilon \\ \zeta_{\min}\delta(t) \leq z(t) \leq \zeta_{\max}\delta(t) \\ x(t) - \zeta_{\max}(1 - \delta(t)) \leq z(t) \leq x(t) - \zeta_{\min}(1 - \delta(t)) \end{cases} \quad (9)$$

ここで、 ε は近似の度合いを表す十分小さな正数である。また、 $\xi_{min} := \min_{x \in \mathcal{X}} Cx$, $\xi_{max} := \max_{x \in \mathcal{X}} Cx$, $\zeta_{min} := \min_{x \in \mathcal{X}} x$, $\zeta_{max} := \max_{x \in \mathcal{X}} x$ である。(9) 式の第2式は (7) 式を任意の精度で近似している。第3, 4式は、 $z(t) = \delta(t)x(t)$ の等価表現である。これを (4),(5) 式の形式で表したものが、(6) 式の MLD システム表現となる。□

MLD システムは、このようなモードの切換えだけでなく、解のジャンプ現象や離散値入力の場合なども同様に記述できる。また、数理計画分野で良く知られているように、命題論理を 0-1 変数の線形型の不等式で表現することによって MLD システムに組み込める [11]。MLD システムの様々な表現例は、文献 [9, 12] などを参考にされたい。

3 混合論理動的システムのモデル予測制御と混合整数計画問題

本節では、MLD システム表現に基づく制御手法としてモデル予測制御手法を概説する。また、この制御手法において現れる混合整数計画問題の特徴を整理し、研究課題について述べる。

3.1 モデル予測制御

MLD システム (4),(5) において

$$v(t) = \begin{bmatrix} u(t) \\ z(t) \\ \delta(t) \end{bmatrix} \in (\mathcal{R}^{m_{1c}} \times \{0, 1\}^{m_{1d}}) \times \mathcal{R}^{m_2} \times \{0, 1\}^{m_3}$$

$$B = [B_1 \ B_2 \ B_3], \quad D = [D_1 \ D_2 \ D_3]$$

とおくと

$$x(t+1) = Ax(t) + Bv(t) \tag{10}$$

$$Cx(t) + Dv(t) \leq E \tag{11}$$

と表現できる。これは v や x に離散値が含まれるが、形式上は拘束線形システムと呼ばれる動的システムと同じである。したがって、拘束線形システムに対する有効な制御系設計法の一つであるモデル予測制御手法が以下に示すように、MLD システムに対しても適用できる [9]。

議論を簡単にするため、原点を目標値とし、評価関数の終端条件を省略した、つぎの最適制御問題を考える。

問題 3.1 (有限時間最適制御問題) 現在時刻 k での状態 $x(k)$ におけるシステム (10), (11) に対して、評価関数

$$J(x(k), v) = \sum_{t=k+1}^{k+T} x^T(t) Q x(t) + \sum_{t=k}^{k+T-1} v^T(t) R v(t) \tag{12}$$

を最小にする入力 $v^*(t)$, $t = k, k+1, \dots, k+T-1$, を求めよ。ただし、 $Q \geq 0$ (半正定), $R > 0$ (正定) であり、 T は予測ステップ数と呼ばれる。

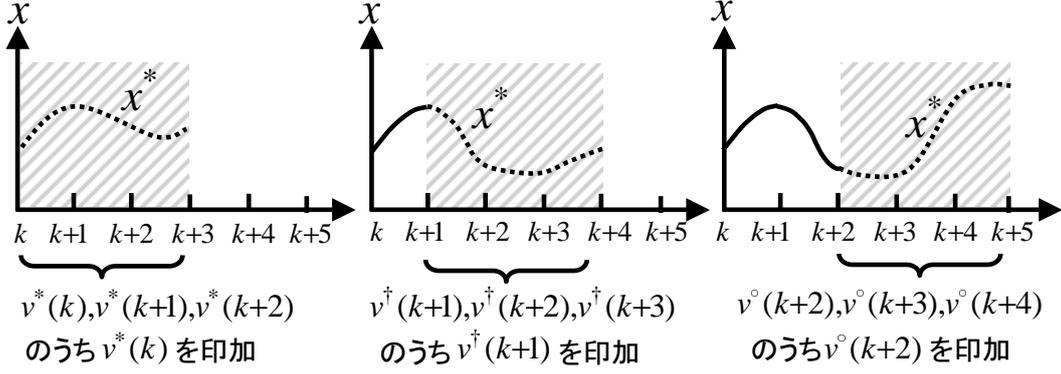


図 2: モデル予測制御 (v^*, v^\dagger, v° は各区間での最適解, x^* は最適軌道を示す)

評価関数 J の第 1 項は状態 x の過渡的な振る舞いに相当する量を, 第 2 項は入力 v のエネルギーを表す.

このとき, モデル予測制御とは, 図 2 に示すように, 状態に依存して毎時刻ごとに最適化を行う逐次最適化手法である. 具体的には, J を最小にする入力 v^* を最適入力列と呼ぶものとする, この最適入力列 $v^*(t)$, $t = k, k+1, \dots, k+T-1$ のうち, $v^*(k)$ のみを現在時刻 k において制御対象に印加し, つぎに, それにより得られたつぎの時刻 $t = k+1$ の状態 $x(k+1)$ に対して, 上記の問題において k を $k+1$ と置き換えた問題を考え, 同様に, 最適入力列の 1 ステップ分の入力のみ印加する, といったように繰り返す手法である.

このように, モデル予測制御を適用するには, 上記の有限時間最適制御問題を解けばよい. この問題は, 式 (10) の漸化式を解く, すなわち $x(k+1), \dots, x(k+T)$ を現在状態 $x(k)$ と入力列 $v(k), \dots, v(k+T-1)$ で表現することによって, つぎの混合整数 2 次計画問題 (MIQP 問題と略す) に帰着される.

$$\begin{aligned} \min_{\bar{v}_k \in \mathcal{V}} \quad & \bar{v}_k^T M_1 \bar{v}_k + \bar{v}_k^T M_2 x(k) \\ \text{s.t.} \quad & L_1 \bar{v}_k \leq L_2 x(k) + L_3, \end{aligned} \quad (13)$$

ここで入力集合 \mathcal{V} は $\mathcal{V} = (\mathcal{R}^{m_1 T} \times \{0, 1\}^{m_1 d T}) \times \mathcal{R}^{m_2 T} \times \{0, 1\}^{m_3 T}$ であり

$$\bar{v}_k = \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+T-1) \end{bmatrix}$$

である. また M_i, L_i は適当な次元の行列である.

この問題は $x(k)$ の値によっては可解でなくなる可能性もあり得る. 常に可解であることは, ある種の可制御性問題を予め解くことによって確かめられる. これに関しては次節で扱う. また, 評価関数を 2 次形式でなく, 1 次形式で与え, 混合整数線形計画問題に帰着する場合も同様である.

3.2 リアルタイム最適化

モデル予測制御問題の特徴は、現在時刻 k において、動的システムに基づいて状態 $x(k)$ が与えられたとき、上記の MIQP 問題 (13) をリアルタイムで解くことにある。リアルタイム性の要求度は制御対象である動的システムの応答の速さに依存して決まるが、一般に、制御入力の変更は、機械システムの場合で数ミリ秒～数百ミリ程度、プロセスの場合で数分～数時間が要求される。一方、MIQP 問題 (13) の入力集合 \mathcal{V} の次元は、連続変数が $(m_{1c} + m_2)T$ 、離散変数が $(m_{1d} + m_3)T$ となり、大雑把には、連続値入力数 m_{1c} と補助変数 z の次元 m_2 の和を 10 以下、離散値入力数 m_{1d} と補助変数 δ の次元 m_3 の和も 10 以下とすると、連続変数、離散変数の数は、それぞれ予測ステップ数 T のオーダーもしくは、 T の 10 倍程度になる。一般には、 T は数十程度（大きいほど制御性能は良くなるが）あればよいので、全体の変数の数は連続変数、離散変数それぞれ数百程度と大雑把には見積もれる。

例えば、例 2.1 の 2 モードの 2 次元 PWA システムに対する制御では、連続変数は $4T$ 、バイナリ変数は T となり、ILOG 社の CPLEX ソルバ (Xenon 2.0GHz, メモリ 2.0GB) で $T = 20$ で平均 0.1 秒、 $T = 30$ で平均 10 秒を要し、機械システムの場合には、数十程度の変数を含む極小規模なシステムの制御問題にしか適用できない、というのが現状である。

一方、制御分野では、2000 年頃より、オフラインで (13) 式の最適解 \bar{v}_k^* を $x(k)$ の関数として陽に求めておき、オンラインでは、求めておいた関数の値を計算することで制御入力を生成するアプローチも精力的に行われてきた [13, 14]。すなわち、 $x(k)$ をパラメータとして扱い、この関数として最適解を求めるマルチパラメトリック MIQP (M-MIQP) 問題の解法が提案され、ソルバできている。しかしながら、容易に予想できるように解構造の複雑さから、一般には、オンラインで解く場合よりもさらに小規模の問題しか解けていないのが現状である。

まとめると、ハイブリッドシステムの最適制御問題に生じる MIQP 問題の特徴は

1. 数百程度以下の変数を含む小・中規模程度の問題
2. 計算時間は、機械システムの場合で数ミリ～数百ミリ秒、プロセス系の場合で数分～数時間以内
3. 毎時刻状態 $x(k)$ がダイナミクスに依存して変化する問題
4. 予測ステップ数 T が計算量に大きく依存する

である。これらの特徴を考慮して、より効率のよい MIQP 問題の解法を見つけることが大きな課題である。

例えば、 $x(k)$ はあるダイナミクスに依存して変化するの、その特徴を活かした解法が考えられる。離散変数がない場合、すなわち (10),(11) 式において $v(k) \in \mathcal{R}^m$ の場合を考える。この場合、MIQP 問題は単に QP 問題になるのでモデル予測制御の解は、KKT 条件より得られるラグランジュ定数 $\lambda(k)$ を用いて $v(k) = Kx(k) + F\lambda(k)$ (K, F は問題から定まる) の形式で与えられる。これより、制御対象 (10) と上記の制御入力の閉ループ系は、制御分野では線形相補性システムと呼ばれる、つぎのシステムで与えられる。

$$\begin{aligned}x(k+1) &= (A + BK)x(k) + BF\lambda(k) \\y(k) &= Mx(k) + N\lambda(k) + L \\y(k) &\geq 0, \lambda(k) \geq 0, y^T(k)\lambda(k) \geq 0\end{aligned}$$

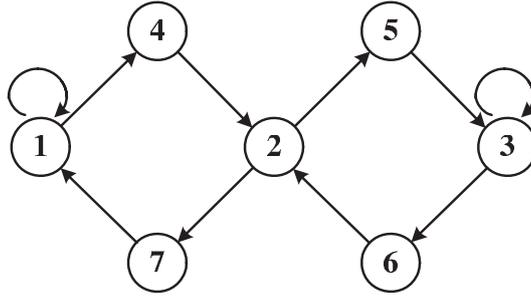


図 3: 有限オートマトンの例

ここで K, F, M, N, L は予め与えられる行列である。このシステムの解を追跡するには、毎時刻、相補性条件を満たす $\lambda(k)$ を求める必要がある。この際、 $x(k)$ の変化分をもとに $\lambda(k)$ を追跡する方法（連続変形法など）が考えられるが [15]、MIQP 問題の場合は離散値を含むため、直接適用することは難しいように思われる。

近次解を求めることもアプローチの一つである。例えば、制約された時間内の暫定解を使用することが考えられる。あるいは、 T に依存することを考慮して、入力変数を適当な時間区間で一定値となる制約条件を加えることで、実際に求めるべき入力変数の数を減らす方法が考えられている（入力ブロッキング法）。いずれにせよ、これらの場合には、近似精度に関して実用レベルの、何らかの理論的な保証が求められるが、筆者の知る限り、未解決のままである。

4 2つの試み

本節では、上記の計算の困難さを緩和する別のアプローチとして、筆者らが提案しているオートマトンモデリングについて紹介する。また、モデル予測制御問題の可解性の解を与える可制御性問題に対して、筆者らの確率的アプローチについて簡単に紹介する。

4.1 モデリングからのアプローチ

有限オートマトン、すなわち離散ダイナミクスを効率よく表現することによって、MIQP 問題の解を得るまでの計算時間を短縮することができる。本節では、こうした視点から研究を始めた有限オートマトンの表現方法について概説し、今後の課題について述べる。詳細は文献 [16] を参照されたい。

簡単のため、図 3 の有限オートマトンの例を用いて説明する。ここで、図 3 の各ノードの数字はモード（離散状態）を表す。各ノードに連続値からなる離散時間線形状態方程式を割り当てることで、システム全体はハイブリッドシステムとなるが、線形状態方程式表現とは独立に議論できるので、ここではオートマトンのみに注目する。

0-1 変数 δ_{ij} をノード i からノード j への辺に割り当て、各ノードの入出力関係を表現する。例えば、ノード 1 の入出力関係は $\delta_{14}(t+1) + \delta_{11}(t+1) = \delta_{11}(t) + \delta_{71}(t)$ により表現できる。他のノードの入出力関係も同様に表現することにより、図 3 の有限オートマトンは離

散時間線形インプリシットシステムと呼ばれるシステム

$$E\xi(t+1) = F\xi(t), \quad e^T \xi(0) = 1 \quad (14)$$

により表現できる. ここで

$$\xi = [\delta_{11} \ \delta_{14} \ \delta_{25} \ \delta_{27} \ \delta_{33} \ \delta_{36} \ \delta_{42} \ \delta_{53} \ \delta_{62} \ \delta_{71}]^T,$$

$$E = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

であり, $e = [1 \ 1 \ \dots \ 1]^T \in \mathcal{R}^{10}$ である. これを直接 MIQP 問題の拘束条件として用いるのではなく, さらに, ある正則行列 V による線形変数変換 $\xi(t) = V [x_d^T(t) \ u_d^T(t)]^T$ によって, インプリシットシステム (14) をつぎのように状態方程式と線形不等式に書き換える.

$$\begin{cases} x_d(t+1) = A_d x_d(t) + B_d u_d(t), \\ 0_{7 \times 1} \leq x_d(t) \leq e, \quad e^T x_d(0) = 1 \end{cases} \quad (15)$$

ここで, $u_d(t) \in \{0, 1\}^3$, $x_d(t) \in \mathbf{R}^7$ ((15) 式の不等式, 初期値の条件, および $u_d \in \{0, 1\}^3$ から自動的に $x_d(t) \in \{0, 1\}^7$ となる), $e = [1 \ 1 \ \dots \ 1]^T \in \mathcal{R}^7$ であり各係数行列は

$$A_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

である. $u_d(t) \in \{0, 1\}^3$ は自由変数であり, 入力に相当する変数は 3 変数となることに注意されたい. 有限オートマトンを表す状態 x_d は MIQP 問題の変数には現われず, 0-1 入力数 (u_d の次元) \times ステップ数が MIQP 問題の離散変数となることを考えると, 有限オートマトンを 3 つの自由変数を用いて簡潔に表現できたとと言える. この有限オートマトンの各ノードに異なる 2 次の 1 入力線形システムを割り当てたハイブリッドシステムの場合, $T = 50$ で 1 秒以内 (Pentium 4, 2.53GHz, メモリ: 2GB, CPLEX ソルバ) で解くことができる.

以上が基本的なアイデアである. ここでの研究動機は, 有限オートマトンを最小数の入力変数を有する状態方程式で系統だって表現する手法を導出することにある. もちろん, MIQP 問題の解法にも依存するため, 離散変数の数と計算時間とは直接的な関係はないが, 正攻法の理論を構築することは難しいため, 離散変数の数を計算量の一つの目安として捉えている. また, 与えられたオートマトンを線形形式で表現できる離散変数の最小数を理論的に解析したり, こうした状態方程式表現によって離散ダイナミクス自体の制御における様々な理論的な展開も期待できる.

さて, 一般にこの問題はつぎのように与えられる.

問題 4.1 エッジ数が l , ノード数が $n (\leq l)$ の有限オートマトンを表すインプリシットシステム

$$\Sigma : E\xi(t+1) = F\xi(t), \quad \xi(t) \in \{0, 1\}^n, \quad \xi(0) = \xi_0 \in \mathcal{A}_0$$

が与えられているとする．ここで $E, F \in \mathcal{R}^{m \times n}$, $\mathcal{A}_0 \subseteq \{0, 1\}^n$ である．このとき，これを状態方程式と線形不等式

$$\Sigma_S : \begin{cases} x_d(t+1) = Ax_d(t) + Bu_d(t), \\ Cx_d(t) + Du_d(t) \leq E, \\ x_d(t) \in \mathcal{R}^p, u_d(t) \in \{0, 1\}^r, x_d(0) = x_0 \in \mathcal{X}_0, u_d(0) = u_0 \in \mathcal{U}_0 \end{cases}$$

に変換する正則な座標変換 $\xi = V[x_d^T \ u_d^T]^T$ を求めよ．ここで， $x_d(t)$ は離散状態， \mathcal{X}_0 は初期状態の集合， $u_d(t)$ は 0-1 変数からなる自由変数（以下では，入力変数と呼ぶ）， \mathcal{U}_0 は初期入力の集合， A, B, C, D は適当なサイズの行列である．

この問題のポイントは，インプリシットシステムから状態方程式への変換において，入力変数のバイナリ性を保存する座標変換を見つけることにある．この種の問題の場合，有限体上で座標変換を考えることもできるが，その後，前節のモデル予測制御で論じたように MIQP 問題に帰着するため，有限体上で得られた状態方程式を実数体上に変換しなければならない．この変換は一般に非線形であるため，問題は簡単ではない．そこでここでは，実数体上のまま，上記の座標変換を求めることを考える．

この座標変換は大雑把には， $EP = \begin{bmatrix} I & \tilde{E} \end{bmatrix}$ (I は単位行列， \tilde{E} は適当な行列) を満たす置換行列 P を用いて

$$\xi = V \begin{bmatrix} x_d \\ u_d \end{bmatrix}, \quad V := P \begin{bmatrix} I & -\tilde{E} \\ 0 & I \end{bmatrix},$$

で与えられ，適当な行列 A, B を用いて， $x_d(t+1) = Ax_d(t) + Bu_d(t)$ と表現できる．しかし，一般には， $\text{rank} B$ が行フルランクでないため，さらに u_d のバイナリ性を保存した形で座標変換を必要とし，実際の変換は少々複雑なのでここでは割愛する．さらにこの座標変換を用いて得られた 0-1 の自由変数（入力変数）の数は，適当な線形変換のクラスの中で最小になることが示せる．詳細は文献 [16] を参照されたい．

4.2 可制御性問題における確率的アプローチ

モデル予測制御を適用するためには，各時刻 k で定まる $x(k)$ に対して MIQP 問題が可解でなければならない．しかし $x(k)$ は入力が定まらないと決まらないため，適当な有界閉集合 \mathcal{X}_0 を予め設定し，すべての $x(k) \in \mathcal{X}_0$ に対して MIQP 問題の可解性をオフラインで調べておくことを考える．この可解性問題は可制御性問題（入力の存在性判別問題）に含まれる．具体的には，つぎのように与えられる [17]．

問題 4.2 (可制御性判別問題) 状態空間 \mathcal{X} 上で定義された MLD システム ($x \in \mathcal{R}^n, u \in \mathcal{R}^m$ とする) に対し，終端時刻 $t = T \in \{1, 2, \dots\}$ と，初期時刻 $t = 0$ ，終端時刻 $t = T$ の連続状態集合 $\mathcal{X}_0 \subseteq \mathcal{X}$, $\mathcal{X}_T \subseteq \mathcal{X}$ が任意に与えられているものとする．このとき，すべてのおのおの $x(0) \in \mathcal{X}_0$ に対し， $x(t) \in \mathcal{X}$ ($t = 1, 2, \dots, T-1$), $x(T) \in \mathcal{X}_T$ を満たす入力列 $U := [u(0), u(1), \dots, u(T-1)]^T \in \mathcal{U}$ が存在するかを判別せよ．ここで， $\mathcal{X}, \mathcal{X}_0, \mathcal{X}_T, \mathcal{U}$ は凸多面体とする．

この問題の難しさは、任意の $x(0) \in \mathcal{X}_0$ に対して可制御性判別をしなければならないことにあり、確定的な手法では膨大な計算量を要するため、極小規模の問題しか解けない。そこで、筆者らは、この問題に対して、モンテカルロ法と射影法を用いた確率的なアプローチを提案した [17]。以下ではその概要を述べる。

上記の問題は、結局、可制御状態集合（終端状態条件や入力条件を満たす入力が存在する初期状態の集合）を求める問題に置き換えることができる。そこで、連続状態 (x) の次元を n 、離散状態の次元（モード数）を M 、入力 (u) の次元を m とおく。さらに、モード列を $\mathcal{I} = [I(0), I(1), \dots, I(T)]$, $I(k) \in \{1, 2, \dots, M\}$ とおくと、モード列 \mathcal{I} に依存した適当な行列 $E_{\mathcal{I}}, F_{\mathcal{I}}, G_{\mathcal{I}}$ によって、可制御状態集合 $\mathcal{X}_c \subseteq \mathcal{R}^n$ は、大雑把には

$$\mathcal{X}_c = \bigcup_{\mathcal{I}} \mathcal{X}_c^{\mathcal{I}}, \quad \mathcal{X}_c^{\mathcal{I}} = \{x_0 \in \mathcal{X}_0 \mid \exists U \in \mathcal{U} \text{ s.t. } E_{\mathcal{I}}x_0 + F_{\mathcal{I}}U \leq G_{\mathcal{I}}\}$$

と表現できる。ここで $\mathcal{X}_c^{\mathcal{I}}$ はモード列を \mathcal{I} に固定した際の可制御状態集合を表し、すべての \mathcal{I} に対する $\mathcal{X}_c^{\mathcal{I}}$ の和集合が可制御状態集合 \mathcal{X}_c となる。このとき、 $\mathcal{X}_c = \mathcal{X}_0$ ならば \mathcal{X}_0 上で可制御であると判定できる。

さて、モード列 \mathcal{I} を固定した際の可制御状態集合 $\mathcal{X}_c^{\mathcal{I}}$ は、集合

$$\mathcal{Z}^{\mathcal{I}} := \{(x_0, U) \in \mathcal{X}_0 \times \mathcal{U} \mid E_{\mathcal{I}}x_0 + F_{\mathcal{I}}U \leq G_{\mathcal{I}}\}$$

の \mathcal{X}_0 上への射影であると言い換えることができる。これより、射影アルゴリズムとして Vertex enumeration 法, Fourier-Motzkin elimination 法, Block elimination 法などが適用できるが、いずれの方法も計算量は n, T, M, m に関して指数的に増大する。そこで、文献 [17] では、射影空間 \mathcal{X}_0 上での包 (hull) に基づくアルゴリズムを構築することによって、 T, M, m に関しては多項式時間の射影アルゴリズムを提案している。しかしながら、 n に関しては指数時間オーダとなる。

こうして、上記の射影アルゴリズムにより、i.i.d. サンプルした \mathcal{I} に対して求めた可制御状態集合 $\mathcal{X}_c^{\mathcal{I}}$ の和が \mathcal{X}_0 と一致するならば「可制御である」と判定できる。一方、適当に与えた正数 $\gamma, \varepsilon \in (0, 1)$ に対して $\ln \frac{1}{\gamma} / \ln \frac{1}{1-\varepsilon}$ より大きな整数 N の i.i.d. サンプル \mathcal{I} に対して、 $\mathcal{X}_c^{\mathcal{I}}$ の和が \mathcal{X}_0 と一致しないならば、

$$\text{Prob}\{\text{Prob}\{\mathcal{I} \in \mathcal{M}(x_0)\} \leq \varepsilon\} \geq 1 - \gamma$$

$\mathcal{M}(x_0)$: 初期状態 x_0 からの解が入力によって原点に到達することができるモード列の集合

の意味で不可制御であると判定できる。すなわち ε, γ が十分に小さいならば、ある x_0 を原点に到達させるモード列がほとんどないと判定できる。ここで、サンプル数 N は終端時刻 T やシステムの次元 n には依存せず、 ε と γ の多項式関数で抑えられる点に注意されたい。こうして全体として、 n 以外の T, M などに関して多項式時間アルゴリズムとなっている。このアルゴリズムをここでは P アルゴリズム (Positive-sided error algorithm) と呼ぶ。

同様に、初期状態 $x_0 \in \mathcal{X}_0$ を i.i.d. サンプルし、与えられた $x_0 \in \mathcal{X}_0$ に対して、ある MILP 問題を解いて x_0 に対する可制御性を判別することができる。これにより、ある $x_0 \in \mathcal{X}_0$ で相当する MILP 問題が可解でないならば、不可制御であると判定できる。一方、適当に与えた正数 $\gamma, \varepsilon \in (0, 1)$ に対して $\ln \frac{1}{\gamma} / \ln \frac{1}{1-\varepsilon}$ より大きな整数 N の i.i.d. サンプルのいずれに対しても可解であるならば

$$\text{Prob}\{\text{Prob}\{x_0 \in \mathcal{X}_0 - \mathcal{X}_c\} \leq \varepsilon\} \geq 1 - \gamma$$

表 1: 3つのアルゴリズムの関係

		Pアルゴリズム	Nアルゴリズム	Tアルゴリズム
アルゴリズムのランダム変数		\mathcal{I}	x_0	\mathcal{I} かつ x_0
アルゴリズムの解	肯定的結果	可制御である	確率的な意味で可制御	確率的な意味で可制御
	否定的結果	確率的な意味で不可制御	不可制御	確率的な意味で不可制御
最悪ケースの計算量	T について	多項式オーダー	指数オーダー	多項式オーダー
	n について	指数オーダー	多項式オーダー	多項式オーダー
	他の変数について	多項式オーダー	多項式オーダー	多項式オーダー

の意味で可制御であると判定できる. これをNアルゴリズム (Negative-sided error algorithm) と呼ぶことにする. さらに, この2つのサンプリングを同時に実行した場合 (Tアルゴリズム, Two-sided error algorithm) も考えられる. 表1に, これら3つのアルゴリズムの計算量と判定結果の関係を示す. これらは相補的な関係にあり, 併用することで, 確定的な解を得ることも十分可能である.

以下に, 簡単な例を示す. 例えば, 5モード2次元1入力システムの場合で, $\mathcal{X} = [-20, 20] \times [-20, 20]$, 終端時刻が $T = 10$ と $T = 50$, 精度が $\varepsilon = 0.001$, $\delta = 0.001$ のとき, Pアルゴリズム, Nアルゴリズムともサンプル数は $N = 7000$ となる (実際には, 事前処理により, この精度のもとでそれぞれ $N = 719$, $N = 5178$ で十分であることが示せる). Tアルゴリズムの場合は $N = 7000^2$ (実際には $N = 719 \times 5178$) となる. このとき判定までに要した計算時間を表2, 表3に示す (Pentium 4, 2.53GHz, メモリ: 2GB).

$T = 50$ の場合でも, Nアルゴリズムを除いて, 1時間以内で解を得ることができるのがわかる. Nアルゴリズムは計算量が T に関して指数オーダーであるため, 解けなかった. 同様に, Pアルゴリズムは, T の値次第ではあるが, n が数次元 (< 10) のシステムの場合しか計算量の観点から適用できない. このアルゴリズムは, 単にモデル予測制御の可解性を解くためだけでなく, 例えば, バイオ系の可制御状態集合を求める問題を解く際にも使える. ただ, バイオ系の場合一般に数十次元以上のシステムを扱う必要があり, 今後はシステムの構造を利用した射影アルゴリズムの改良などが大きな課題である. 詳細は文献 [18, 19] を参照されたい.

5 おわりに

本稿では, ハイブリッドシステムの制御において混合整数計画問題に関連する話題を取り上げ, 筆者らの研究を中心に, その要点と問題点を解説した. ハイブリッドシステムの制御に関する理論はかなり整備されてきており, 実用化に向けた試みも数多くなされるようになってきている. しかしながら, 上述したように, ハイブリッドシステムの最適制御を考える際には, 混合整数計画問題をリアルタイムで解くことが実用化の上で不可欠な要請となっているが, 現状では満足のいく域に達していない. これには, 制御問題の特殊な構造に特化した最適化アルゴリズム等を考える必要があるのかもしれない. 斬新なブレイクスルーが望まれる.

表 2: 確率的可制御性解析 : $T := 10$ の場合

システム例		例 1			例 2		
アルゴリズム		P	N	T	P	N	T
可制御性		N*	N	N*	Y	Y*	Y*
計算時間 [秒]	最大	133.2	1,600.4	7.9	5.2	154.8	156.7
	平均	121.7	1,404.4	7.2	3.9	124.7	136.1
	最小	114.4	1,274.0	6.2	1.9	102.0	121.3

表 3: 確率的可制御性解析 : $T := 50$ の場合

システム例		例 1			例 2		
アルゴリズム		P	N	T	P	N	T
可制御性		N*	?	N*	Y	Y*	Y*
計算時間 [秒]	最大	2785.1		564.5	12.2	1227.7	1101.9
	平均	2690.0	—	515.3	9.7	1057.7	896.5
	最小	2639.2		444.2	6.7	851.1	762.4

Y: 可制御, Y*: 確率的な意味で可制御, N: 不可制御, N*: 確率的な意味で不可制御, ?: 判定不能

謝辞: 第 4 章の研究成果は東俊一氏 (京都大学), 小林孝一氏 (北陸先端科学技術大学院大学) との共同研究によるものである. また, 4.1 節は, 松井知己先生 (中央大学) とのディスカッションが契機となって展開できた研究成果である. 記して感謝します.

参考文献

- [1] H.S. Witsenhausen. A class of hybrid-state continuous-time dynamical systems. *IEEE Transactions on Automatic Control*, Vol.11, No.2, pp.161–167, 1966.
- [2] 井村. ハイブリッドシステムの制御における未解決問題. 計測と制御, Vol.42, No.2, pp.113–119, 2003.
- [3] 井村. ハイブリッドシステムのリアルタイム最適制御, 計測と制御, Vol. 44, No. 7, pp.464–470 (2005)
- [4] 井村, 東. ハイブリッドシステムの制御 - I: 総論, システム/制御/情報, Vol.51, No.5, pp.230–237 (2007)
- [5] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automatic Control*, Vol.43, No.1, pp.31–45, 1998.
- [6] A.J. van der Schaft and J.M. Schumacher. An introduction to hybrid dynamical systems. *Lecture Notes in Control and Information Sciences*, 251, Springer, 2000.

- [7] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control*, Vol.43, No.4, pp.555–559, 1998.
- [8] J. Imura and A.J. van der Schaft. Characterization of well-posedness of piecewise linear systems. *IEEE Trans. on Automatic Control*, Vol.45, No.9, pp.1600–1619, 2000.
- [9] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, Vol.35, pp.407–427, 1999.
- [10] W.P.M.H. Heemels, J.M. Schumacher, and S. Weiland. Linear complementarity systems. *SIAM J. Applied Mathematics*, Vol.60, No.4, pp.1234–1269, 2000.
- [11] H.P. Williams. 数理計画モデルの作成法 (小林訳). 産業図書, 1995.
- [12] 井村. ハイブリッドシステムの制御 - II : モデリング, システム/制御/情報, Vol.51, No.7, pp.306–312, 2007.
- [13] A. Bemporad, F. Borrelli, and M. Morari. Optimal controllers for hybrid systems: stability and piecewise linear explicit form. *Proc. of IEEE Conference on Decision and Control*, pp.1810–1815, 2000.
- [14] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-parametric toolbox(MPT). *International Conference on Hybrid Systems: Computation and Control (HSCC'07)*, LNCS 2993, Springer-Verlag, pp.448–462, 2004.
- [15] Y. Saito and J. Imura, A difference based efficient approximate algorithm for model predictive control of input constrained linear systems, *Proc. of IFAC World Congress*, Th-E05-TP/1 (2005)
- [16] K. Kobayashi and J. Imura. Minimality of finite automata representation in hybrid systems control. *Int. Conf. on Hybrid Systems: Computation and Control (HSCC'07)*, LNCS 4416, Springer-Verlag, pp.343–356, 2007.
- [17] S. Azuma and J. Imura. Polynomial-time probabilistic controllability analysis of discrete-time piecewise affine systems. *to appear in IEEE Trans. on Automatic Control*.
- [18] S. Azuma, E. Yanagisawa, and J. Imura. Probabilistic approach to controllable state set problem of hybrid systems and its application to luminescence bacterium. *Proc. 17th Int. Symposium on Mathematical Theory of Networks and Systems*, pp.375–380, 2006.
- [19] D. Sugiyama and J. Imura. Controllability measure of piecewise affine systems and its applications to the Luminescence Bacterium. *to appear in IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*.