非線形なベクトル場をもつハイブリッドオートマトンの 切り替え条件の最適設計

—PLC を用いた制御システムへの適用—

小 中 英 嗣*

Optimal Design of Guard Conditions of Nonlinear Hybrid Automata
—Application to PLC-based Control Systems—

Eiji KONAKA*

Programmable Logic Controllers (PLCs) have been widely used in industry. This class of systems is characterized by its use of low-resolution (typically ON/OFF) sensors and actuators. In PLC-based control systems, the design of sensor parameters is crucial since the sensor parameters as well as the outputs of the plant determine the sensors' ON/OFF states, with the parameters finally affecting overall control performance.

In this paper, first, a method for transforming PLC-based control systems into hybrid automata is presented. Here, the sensor parameter design problem can be formulated as the optimal design of guard conditions in hybrid automata. Secondly, the optimal method for designing of guard conditions is proposed. This method is basically based on a graph search method, and can be applied to nonlinear plants. Finally, the usefulness of the proposed method is demonstrated through a numerical example.

Key Words: optimization, nonlinear switching system, PLC(Programmable Logic Controller)

1. はじめに

プログラマブルロジックコントローラ (Programmable Logic Controller, 以下 PLC) は現在産業界で幅広く用いられている制御器である。PLC の特徴として, ON/OFF 型のセンサ・アクチュエータの使用を想定していることが挙げられる。ラダー図, SFC(Sequential Function Chart) など, PLC のプログラミング言語として古くから用いられているものの記述法は, リレー接点の ON/OFF を図的に表現したものである^{1), 2)}。

近年では連続値入出力を備え, PID 制御プログラムなどを実装できる高機能な PLC の開発も進んでいるが, PLC を用いた制御の大半は離散入出力を用いた単純なものである³⁾。本研究では, センサ・アクチュエータとして ON/OFF 型のものを扱う PLC のみを考察の対象とする。

PLC を用いて制御される制御対象として, 機械系, 化学プラントなどさまざまな種類のものが考えられるが, その多くは連続値の状態変数や出力変数をもっており, その挙動は微

Fig. 1 PLC-based control system

分方程式で表わされる。センサやアクチュエータは, それら連続的な変数と ON/OFF 値を変換する装置となる。

Fig. 1 に本論文で考察する制御系を示す。 x, u および y はそれぞれ制御対象の状態, 入力および出力である。制御対象の出力 y は ON/OFF 型センサにより観測され, 論理値 (ON/OFF) の信号 Y に変換される。センサには調整可能なパラメータ θ があり, これにより ON/OFF の条件を調節することができるとする。PLC は Y を受け取り, 実装されている制御論理に従ってアクチュエータへの ON/OFF 信号 U

* 名城大学理工学部 名古屋市天白区塩釜口 1-501

* School of Science and Technology, Meijo University, 1-501
Shiogamaguchi, Tenpaku-ku, Nagoya
(Received August 31, 2006)
(Revised February 21, 2007)

を決定する。アクチュエータは U を実際の物理的な入力 u へ変換し、 u が制御対象の状態 x を制御する。

PLC を用いた制御系を設計する場合、制御論理の設計と共にセンサのパラメータ θ の設計も重要な問題となる。なぜなら、センサのパラメータ θ は制御対象の状態 x とともにセンサの ON/OFF の条件を決定し、最終的には制御系全体の制御性能に影響を及ぼす(いい換えると、 θ をうまく設計することにより、制御系全体の制御性能の改善が期待できる)と考えられるからである。このようなセンサパラメータの例として、位置制御システムにおけるリミットスイッチの設置位置や、温度制御システムにおけるサーモスタットの設定温度などが挙げられる。しかしながら、ON/OFF 型のセンサ・アクチュエータと PLC の実システムの適用例として污水处理施設⁴⁾、点火システム⁵⁾、二足歩行ロボット⁶⁾、電源自動切換スイッチ (Automatic Transfer Switch, ATS)⁷⁾などが報告されているが、いずれもセンサパラメータの調節はあらかじめ済んでいるという立場のものであり、その調節による制御性能の向上については言及されていない。

これらのパラメータの設計方法としてまず挙げられるのは、ON/OFF の切り替えを非線形要素と見なし、閉ループ全体を非線形制御系とした後、位相面解析や記述関数法などを用いて設計する方法である⁸⁾。しかしながら、位相面解析は基本的に 2 次の制御系の設計手法であること、記述関数法は近似的解法であり、その適用にはさまざまな仮定が必要であることなどが知られている。さらに、本研究で扱う PLC を用いた制御系では、センサ・アクチュエータ以外にも制御プログラム自体に非線形性があり、それらをすべて非線形制御の枠組みの中で取り扱うことは実用上難しい。

もう一つの設計方法として、最も直感的で幅広く用いられているものは、試行錯誤およびシミュレーション・実験によるものである。この方法では、パラメータの候補の値をいくつか選び、そのパラメータをシミュレーションや実験により検証し、選んだ中で最もよいパラメータを採用する。しかしながら、この方法では制御系の知識をもったオペレータがその経験を活かしながらパラメータの調整を行なうことが多く、系統的な設計手法に対する議論が十分に成されていない。また、選ばれたパラメータよりさらに良いパラメータが存在するかどうか判断できないこともこの手法の特徴として挙げられる。特に、この問題はパラメータの変化に関して挙動が大きく変わる非線形な制御対象で頻繁に起こるものである。

PLC を用いた制御系に対し、混合論理動的システム (Mixed Logical Dynamical System, MLD) 形式による制御システムの表現と数値計画法を組み合わせた動作検証手法が報告されている^{9), 10)}。文献 10) では、パラメータがモデル中に定数項として現れる場合、それらのパラメータの検証に対して提案されている動作検証手法が拡張可能であることが述べられている。また文献 11) では、MLD と数値計画法を用いて最適なセンサパラメータを設計する手法を提案している。しかし、これらの手法には以下の制約がある。

- 制御対象のダイナミクスが区分線形であるか、区分線形関数で近似できるかのいずれかである。
- 制御対象のダイナミクスが離散時間の差分方程式で表わされているか、差分方程式で近似できるかのいずれかである。
- パラメータはモデル中に定数項として現れる。

したがって、これらの手法を非線形なダイナミクスをもつ制御対象に適用する場合、さまざまな近似が必要になる。元の非線形なダイナミクスを区分線形関数で近似するためには補助二値変数を導入する必要がある、近似の精度を上げるためには補助二値変数の数を増やす必要がある。さらに、時間の離散化によっても誤差が生じ、近似の精度を上げるためには時間間隔を短く取る必要がある。しかしながら、MLD と数値計画法を用いた手法では二値変数を含む最適化問題を解くことになり、その計算時間は二値変数の数に対して指数的に増加しうることが知られている^{12), 13)}。したがって、これらの方法では、非線形な制御対象に対して正確な近似と短い計算時間を両立するのが困難である。

それに対し、本論文では、非線形な制御対象をもつ制御システムに対し、区分線形関数への近似や時間の離散化を必要としない、センサパラメータの設計手法を提案する。さらに、提案手法では状態変数の係数として現れるパラメータも設計可能であることを示す。

本論文の構成を以下に示す。第 2 章では、本研究で対象とする PLC を用いた制御系について述べる。第 3 章では、PLC を用いた制御系をハイブリッドオートマトン^{14), 15)}へ変換する手法を述べる。第 4 章では、センサパラメータの設計問題をハイブリッドオートマトンの切り替え条件の設計問題として定式化する。また、この問題に対し、グラフ探索に基づく手法を提案する。最後に、第 5 章で提案手法の有効性を数値例により示す。

2. PLC を用いた制御系

この章では、PLC を用いた制御系における各構成要素について簡単に述べる。

2.1 PLC, SFC(Sequential Function Chart)

PLC はセンサからの入力とアクチュエータへの出力として二値 (ON/OFF) 信号を想定しているものが多く、それに対応したプログラミング言語が多数提案されている。本論文では、PLC に実装される制御論理は SFC により記述されているものとする。

SFC はベトリネット^{16), 17)}から派生したプログラミング言語であり、制御順序 (直列, 並列) を直感的に理解できる図的表現を備えているのをその特徴としている。

Fig. 2 に SFC の例を示す。SFC の構成要素は以下のとおりである。

- ステップ (長方形)：制御の各段階を表わす。制御の段階があるステップにあるとき、そのステップが活性であるという。初期ステップは二重の四角で表わされる。

ステップが活性かどうかを表わすベクトル $X \in$

Fig. 2 Example of SFC

$\{0, 1\}^{N_X}$, (N_X はステップの数) を SFC の論理状態ベクトルと呼ぶ(注1)。

- アクション(楕円): ステップに割り当てられ, その制御の段階でどのような動作(どのアクチュエータを ON にするか, など)を行なうかを表す。あるシステムで, 取得するアクチュエータの ON/OFF の組み合わせの数を N_A とすると, その SFC でのすべてのアクションの動作をまとめて写像 $f_A: \{0, 1\}^{N_X} \rightarrow \{0, 1, \dots, N_A - 1\}$ で表わすことができる。

- トランジション(太い横線): 制御の段階の切り替えを表わす。実際に制御の段階の切り替えが起こることを, トランジションが発火するという。また, あるトランジションの前(後)につながっているステップをそのトランジションに対する前置(後置)ステップと呼ぶ。

トランジションとステップの繋ぎ方で並列/選択分岐を表現できる。図中に示した二重線は並列分岐を表わしており, tr_2 が発火すると st_2, st_4 が同時に活性になる。

- 発火条件: トランジションに割り当てられ, どのような条件が成り立てば(どのセンサが ON になれば, など)そのトランジションが発火可能になるかを表わす。 i 番目のトランジションに対応する発火条件は写像 $f_{C_i}: \{0, 1\}^{N_Y} \rightarrow \{0, 1\}$, (N_Y はセンサの数) で表わされる(センサについては 2.3 節で述べる)。Fig. 3 に発火条件の例を示す。この図は, 発火条件 f_{C_i} がセンサ出力 Y_1, Y_2, Y_3 を用いて $f_{C_i}(\mathbf{Y}) = \bar{Y}_2 \wedge (Y_1 \vee Y_3)$ となっていることを示している。

Fig. 3 Example of firing condition

SFC はトランジションの発火により動作する。トランジション

(注1) ベトリネットでの呼び方にならって, マーキングベクトルと呼ぶこともある。

ンが発火するのは,

- (1) すべての前置ステップが活性
- (2) すべての後置ステップが非活性
- (3) 割り当てられている発火条件が真

の三つの条件を満たすとき(かつそのときのみ)である。トランジションの発火を示すベクトル $\mathbf{V} = \{0, 1\}^{N_V}$, (N_V はトランジションの数) を SFC の発火ベクトルと呼ぶ。トランジションが発火すると, すべての前置ステップが非活性, すべての後置ステップが活性となり, 制御の段階が遷移する。

SFC はサンプル周期に同期してセンサからのデータの取り込み, トランジションの発火の評価とつぎの時刻での論理状態の計算を行なう。トランジションの発火と論理状態の遷移の動作を状態方程式で表わすと次式となる。

$$\mathbf{X}((k+1)T_c) = \mathbf{X}(kT_c) + \mathbf{D}\mathbf{V}(kT_c) \quad (1)$$

ここで k は離散時刻, T_c は PLC の制御周期, $\mathbf{D}$ は接続行列と呼ばれる SFC の構造を表わす行列である。制御時刻間での $\mathbf{X}$ の値はゼロ次ホールド

$$\mathbf{X}(t) = \mathbf{X}(kT_c), \quad t \in [kT_c, (k+1)T_c) \quad (2)$$

により与えられるものとする。ここで t は連続時刻である。

SFC からの出力は, 活性となっているステップに割り当てられたアクションに従って定められる, アクチュエータへの ON/OFF 信号である。これについては次節で述べる。

[例] Fig. 2 の SFC には七つのステップと五つのトランジションがある。したがって, $N_X = 7, N_V = 5$ である。

ステップ st_1 のみが活性であり(つまり, X_1 のみが 1) fc_2 が真となると, トランジション tr_2 が発火して論理状態が $X_1 = 0, X_2 = 1, X_4 = 1$ に遷移する。

アクション ac_i が i 番目のアクチュエータの ON に対応している場合, この SFC で取得するアクチュエータの ON/OFF すべての組み合わせは $\{0, \{1\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{6\}\}$ となる。したがって, $N_A = 7$ となる。

2.2 アクチュエータと制御対象

PLC を用いた制御システムでは, 二値 (ON/OFF) のアクチュエータが制御対象に取り付けられており, その ON/OFF の切り替えにより制御対象の状態が駆動される。本論文では, 二値アクチュエータを備えた制御対象を次式の微分方程式でモデル化する。

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}_i(\mathbf{x}(t)), \\ i &= f_A(\mathbf{X}(kT_c)), \quad t \in [kT_c, (k+1)T_c) \end{aligned} \quad (3)$$

ここで, $\mathbf{x} \in \mathbb{R}^n$ は制御対象の状態であり, $f_i: \mathbb{R}^n \rightarrow \mathbb{R}^n$, ($i = 0, \dots, N_A - 1$) は制御対象のダイナミクスを表わすベクトル場である。また, N_A はアクチュエータの ON/OFF が取得する組み合わせの数であり, (3) はアクチュエータの ON/OFF によりベクトル場が切り替わる制御対象を表わしている。

2.3 センサ

制御対象の出力は各制御時刻ごとに二値 (ON/OFF) のセンサにより観測される。ここで、センサに調整可能なパラメータ $\theta \in \mathbb{R}^{n_\theta}$ がある場合を考える。この場合、センサが ON となる条件は、制御対象の出力のみではなく、センサパラメータにも依存する。出力が入力の関数であることと合わせると、センサは入力とセンサパラメータの関数 $f_{Y_i} : \mathbb{R}^n \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}$ が負となるときに ON になる、とすれば良い。これを式で表わすと、

$$[Y_i(kT_c) = 1] \Leftrightarrow [f_{Y_i}(\mathbf{x}, \theta) < 0] \quad (4)$$

となる。ここで、 f_{Y_i} は θ に対して線形な関数であると仮定する。

[例] $f_Y(\mathbf{x}, \theta) = \|\mathbf{x}\| - \theta$ の場合、このセンサは状態のノルムが θ より小さいときに ON になる。距離センサなどはこの形の関数で表わされる。

[例] Fig. 3 の発火条件 f_{C_i} を f_{Y_i} を用いて表わすと次式になる。

$$[f_{C_i}(\mathbf{Y}) = 1] \Leftrightarrow [f_{Y_2} \geq 0] \wedge [[f_{Y_1} < 0] \vee [f_{Y_3} < 0]] \quad (5)$$

3. ハイブリッドオートマトンによる制御系のモデル化

前章では制御系の各部分について述べたが、本章では制御系全体をハイブリッドオートマトンで表わす方法について述べる。

3.1 ハイブリッドオートマトン

ハイブリッドオートマトン^{14), 15)}は、オートマトンの各離散状態に、ダイナミクスを表わす連続値の微分方程式や差分方程式を割り当てることで、離散事象システムと連続時間システムが混在した動作を表わすものであり、ハイブリッドシステムの主要なモデル化手法として知られている。

本論文では、次式で定義されるハイブリッドオートマトン H を考える。

$$H = (Q, X, E, F, G, q_0, \mathbf{x}_0). \quad (6)$$

ここで、 $Q = \{q_0, \dots, q_{N-1}\}$ はオートマトンの離散状態の集合 (このうち、 q_0 は初期離散状態)、 $X \subseteq \mathbb{R}^n$ は連続状態空間を表わす。また、離散状態を $q(t) \in Q$ 、連続状態を $\mathbf{x}(t) \in X (t \geq 0 \text{ は時刻})$ で表わす。さらに、 $q_0, \mathbf{x}_0$ はそれぞれ離散初期状態と連続初期状態である。

$E \subset Q \times Q$ は離散状態間のアークの集合を表わす。また、 $F = \{f_0, \dots, f_{N-1}\}$ はベクトル場の集合で、 $f_i : X \rightarrow X$ は離散状態 q_i に割り当てられているベクトル場を表わす。 $G = \{G_{i,j} | (q_i, q_j) \in E\}$ は切り替え条件の集合で、 $G_{i,j} \subset X \times \mathbb{R}^{n_\theta}$ は離散状態 q_i から q_j への切り替え条件を表わす。また、それぞれの q_i に対し、 $G_{i,j}$ の共通部分は無いとする (遷移先を一意に決定するため)。

このハイブリッドオートマトンでの状態 $(q(t), \mathbf{x}(t))$ の遷移は次式で表わされる。

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f_i(\mathbf{x}(t)), \text{ iff } q(t) = q_i, \\ q((k+1)T_c) &= q_j, \text{ iff } (\mathbf{x}, \theta) \in G_{i,j}, \\ q(t) &= q(kT_c), t \in [kT_c, (k+1)T_c). \end{aligned} \quad (7)$$

3.2 ハイブリッドオートマトンへの変換手法

第2章で示した制御システムをハイブリッドオートマトンへ変換する方法を述べる。

(1) SFC の初期状態から到達可能な $\mathbf{X}$ を数え上げ、それを集めた集合を $W = \{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_{N_W-1}\}$ とする。ここで、 N_W は到達可能な $\mathbf{X}$ の数である。

(2) 離散状態と連続状態をそれぞれ、

$Q = \{q_0, q_1, \dots, q_{N_W-1}\}, X \subseteq \mathbb{R}^n$ とする。また、ベクトル場の集合を $F = \{f_0, \dots, f_{N_A-1}\}$ とし、離散状態 q_i に対してベクトル場 $f_j, j = f_A(\mathbf{X}_i)$ を対応させる。

(3) つぎに離散状態間のアークの集合 E を定める。元の制御系で、トランジションの発火により状態が $\mathbf{X}_i$ から $\mathbf{X}_j$ に変化する可能性がある場合、 $(q_i, q_j) \in E$ とし、またそのトランジションの発火条件を $f_{G_{i,j}}$ とする。

(4) 最後に離散状態間の切り替え条件の集合 G を定める。 $(q_i, q_j) \in E$ に対し、

$$[f_{G_{i,j}}(\mathbf{Y}) = 1] \Leftrightarrow [(\mathbf{x}, \theta) \in G_{i,j}], \quad (8)$$

を満たす $G_{i,j}$ を離散状態 q_i から q_j への切り替え条件とすると、 $G = \{G_{i,j} | (q_i, q_j) \in E\}$ が切り替え条件の集合となる。

[例] Fig. 4 は、Fig. 2 で示した SFC を用いた制御系を、ここで述べた方法でハイブリッドオートマトンに変換したものである。ステップの活性/非活性は 7 通りあるので、離散状態の数も 7 個になっており、それぞれの離散状態 q_i に連続ベクトル場 f_i を割り当てる。また、図中にはそれぞれの離散状態が、元の SFC でどのステップが活性となっている状態と対応しているかを記してある。 q_0 は $\{st_0\}$ が、 q_1 は $\{st_1\}$ が活性な状態にそれぞれ対応しており、その間の切り替え条件は $f_{C_1}(\mathbf{Y}) = 1$ である。(4) と (8) より、 st_0 から st_1 への切り替え条件は $G_{0,1} = \{(\mathbf{x}, \theta) | f_{C_1}(\mathbf{Y}) = f_{G_{0,1}}(\mathbf{Y}) = 1\}$

Fig. 4 Example of hybrid automata

となる。

4. センサパラメータの最適化

本章では、前章で示したハイブリッドオートマトンを用いて最適なセンサパラメータを設計する手法を述べる。

まず、制御要求に従い、最適化したい評価関数 J を次式とする。

$$\begin{aligned} J &= \sum_{k=0}^{k_f-1} \int_{kT_c}^{(k+1)T_c} h(q(t), x(t)) dt, \quad h \geq 0 \\ &= \int_0^{k_f T_c} \tilde{h}(q(t), x(t)) dt, \quad \tilde{h} \geq 0 \end{aligned} \quad (9)$$

ここで、 $k_f T_c$ は最適化を行なう最終時刻である。

J を最適にするパラメータ θ を求めるアルゴリズムを以下に示す。

(1) 初期状態 $(q(0), x(0))$ は既知であるとし、評価関数 J と最適化を行なう最終時刻 $k_f T_c$ を与える。また、評価関数の最適値の上限 J_{sup} を十分大きな値とし、時刻 kT_c までの評価関数の値を

$$J_{mid}(kT_c) = \int_0^{kT_c} \tilde{h}(q(t), x(t)) dt \quad (10)$$

とおく。 $J_{mid}(kT_s)$ は区間 $[0, kT_c]$ での評価関数の値を示している。

(2) $k \leftarrow 0$ とする。初期状態を根ノードとし、これを未評価のノードの集合 $\mathcal{N}_u$ に追加する。

各ノード $node$ は以下の組で構成されているとする。

$$node = (k, q(kT_c), x(kT_c), J_{mid}(kT_c), \Theta) \quad (11)$$

ここで $\Theta \subseteq \mathcal{R}^{n_\theta}$ は、そのノードに対して取り得るパラメータの範囲を表わす集合である。初期ノードでは、 $\Theta = \mathcal{R}^{n_\theta}$ とする。

(3) (a) $\mathcal{N}_u$ にノードが一つ以上ある場合、どれか一つを選択し、それを $\mathcal{N}_u$ から削除して (4) に進む。

(b) $\mathcal{N}_u$ にノードが一つも無い場合、これまでで得られている J_{sup} が最適 (最小) 値であり、それに対応する Θ が最適なパラメータの集合となる。もし $\Theta = \emptyset$ ならば、初期状態から制約を満たす入力系列が存在しない。

これら場合、アルゴリズムを終了する。

(4) 以下の三つの条件のいずれかを満たす場合、(3) に戻る。そうでない場合、(5) に進む。

(a) 選択したノードで $J_{mid}(kT_c) > J_{sup}$ が成り立つならば、 J_{mid} の単調増加性 ($\tilde{h} \geq 0$) よりこのノードより下のノードで最適解が現れることは無い。

(b) $\Theta = \emptyset$ の場合、そのノードに対応するパラメータが無い。

(c) 選択したノードに対し、 $k = k_f$ ならばこれ以降子ノードを生成する必要が無い。ここで、 $J_{mid}(k_f T_c) < J_{sup}$ ならば、 $J_{sup} \leftarrow J_{mid}(k_f T_c)$ とする。

(5) 選択したノードの離散状態を $q(kT_c) = q_i$ とし、 $(q_i, q_j) \in E$ となる q_j を列挙する。それぞれの q_j に対し、新しいノードを生成し、それらの要素はつぎのように決める。

$$\begin{aligned} k &\leftarrow k+1 \\ q &\leftarrow q_j = q((k+1)T_c) \\ x &\leftarrow x((k+1)T_c) \\ &= x(kT_c) + \int_{kT_c}^{(k+1)T_c} f_i(x(t)) dt \\ J_{mid} &\leftarrow J_{mid}((k+1)T_c) \\ &= J_{mid}(kT_c) + \int_{kT_c}^{(k+1)T_c} \tilde{h} dt \\ \Theta &\leftarrow \Theta \cap G_{i,j}|x(kT_c) \end{aligned} \quad (12)$$

$(q((k+1)T_c), x((k+1)T_c))$ および $J_{mid}((k+1)T_c)$ の計算には (3), (9) を用いる。解析的に積分が行えない場合は、数値解法によりこれらの計算を行なう。

$G_{i,j}|x(kT_c)$ は $x(kT_c)$ の条件付の切り替え条件の集合を表わしており、いい換えると、これは状態 $x(kT_c)$ で q_i から q_j への切り替えがおりうるパラメータの集合を表わしている。

このアルゴリズムを実行することにより、最適なセンサパラメータの集合 Θ_o を得ることができる。そして、得られた Θ_o の中から $\theta_o \in \Theta_o$ となる θ_o を選び、(4) を用いて個別のセンサ Y_i の特性を設計できる。

ここで、 $G_{i,j}$ の表現方法について述べる。仮定より、センサの ON/OFF 条件を決める関数 f_{Y_i} はパラメータ θ に関する線形関数であり、トランジションの発火条件 f_{C_i} は Y_i に関する論理関数である。そのため、 $G_{i,j}|x(kT_c)$ は $\mathcal{R}^{n_\theta}$ 中の凸多面体の和で表現できる。各凸多面体は、 $f_{G_{i,j}}$ を加法標準形で表わしたときの項に対応する。

〔例〕 Fig. 3 の発火条件を加法標準形で表わすと、

$$\begin{aligned} [f_{C_i}(\mathbf{Y}) = 1] &\Leftrightarrow [[f_{Y_2} \geq 0] \wedge [f_{Y_1} < 0]] \\ &\vee [[f_{Y_2} \geq 0] \wedge [f_{Y_3} < 0]] \end{aligned} \quad (13)$$

となる。 f_{Y_i} は θ に関する線形関数なので、各項は凸多面体と対応する。

探索中は Θ を構成する各凸多面体の係数を保存しておく。連立不等式が凸多面体を構成するかどうか、いい換えると連立不等式が解をもつかどうかは、単体法を用いた線形計画問題として実行できる¹⁸⁾。したがって、アルゴリズムの (3)(4) での $\Theta = \emptyset$ の判定には線形計画問題を用いる。一度空集合と判定された不等式の集合は、それ以降の探索には必要ないため、その時点で破棄する。

また、アルゴリズムの (5) で $\Theta \cap G_{i,j}|x(kT_c)$ を計算しているが、これはこの時点で Θ を表わす係数の集合に、 $G_{i,j}|x(kT_c)$ を表わす係数の集合を追加することで得られる。このとき、より細かな凸多面体への分割が必要であればそれを行なう。

また、この時点で新しく追加された不等式が冗長な場合がある。この場合も、着目する不等式が冗長であるかどうか確

認する問題が線形計画問題として定式化できる。

Θ をこの方法で表現すると、探索が進むにつれ Θ を構成する凸多面体の数およびそれを表わす不等式の数が指数的に増加する恐れがある。繰り返し同じ発火条件についての条件をチェックする制御系（ループをもつ制御系）では、新たに不等式を追加する際、

- 空集合となる：以前のセンサ ON/OFF と新たな ON/OFF が矛盾する状況。

例) $[\theta + 1 < 0]$ に $[\theta - 1 > 0]$ を追加する。

- 冗長な不等式となる：以前のセンサ ON/OFF に新たな ON/OFF が含まれる状況。

例) $[\theta + 1 < 0]$ に $[\theta + 2 < 0]$ を追加する。

となる場合があるため、凸多面体の数およびそれを表わす不等式の数の増加は抑えられると予測される。しかし、計算効率を考慮した Θ の表現方法は今後の課題である。

また、このアルゴリズムの (3) でどのようにノードを選択するかで、探索が幅優先か深さ優先かを選ぶことができる（最初に追加したノードを選択する場合は幅優先探索に、最後に追加したノードを選択する場合は深さ優先探索になる）。このアルゴリズムでは、(5) で J_{sup} をより小さな値に更新することで、(4) での枝刈りの効果が増すため、時間を進める方向に探索が進む深さ優先探索のほうが効率が良いと考えられる。

4.1 計算量の削減

前節で示したアルゴリズムでは、4. で J の単調増加性を用いて探索木の枝刈りを行なうことで探索範囲を小さくしている。ここで、枝刈りをより効率的に実行できれば、計算量の削減につながる。

計算量の削減のためには、各ノードで評価関数の最良値の下限 J_{inf} を求め、 $J_{mid}(kT_c)$ の代わりに J_{inf} と J_{sup} を比べることで枝刈りを行なえばよい。

本節では、評価関数が次式で与えられる場合での、計算量の削減手法を提案する。

$$J = \int_0^{k_f T_c} \|Q(r(t) - x(t))\|^2 dt \quad (14)$$

ここで、 $r(t) : \mathcal{R} \rightarrow X$ は連続な目標値関数、 Q は重みを表わす正方行列である。また、 $r(t)$ の劣微分¹⁹⁾を $\partial r(t)$ とする。ここで、

$$v_m = \sup_{X, \xi \in \partial r, i} \{\|Q\xi\| + \|Qf_i\|\} \quad (15)$$

とおくと、 v_m は $Qx(t)$ と $Qr(t)$ が近づく速度の上限を表わす。

あるノードでの連続状態とそこまでの評価値が $x(kT_c)$ 、 $J_{mid}(kT_c)$ であるとする。このとき、 $Qx(kT_c)$ から $Qr(t)$ まで速度 v_m で移動した場合の J が評価値の最良値の下限 J_{inf} となる。目標値へ到達するまでにかかる時間は $t_{inf} = \frac{\|Q(r - x(kT_c))\|}{v_m}$ なので、

$$J_{inf} = J_{mid}(kT_c) +$$

$$\int_{kT_c}^{kT_c + t_{inf}} \|Qx(kT_c) + \frac{t}{t_{inf}} (Q(r - x(kT_c)))\|^2 dt \quad (16)$$

となる。この式は多項式の積分であるので、容易に計算できる。前節で示したアルゴリズムの 4. で $J_{mid}(kT_c)$ の代わりに J_{inf} を用いると、 $J_{mid} \leq J_{inf}$ なので枝刈りの効率が上がり、計算量を削減できる。

5. シミュレーション

提案手法の有効性をシミュレーションにより確認した。計算機環境として、Pentium IV 3.0[GHz]、メモリ 512[MB] の PC 上で Matlab6.5 を用いた。

Fig. 5, 6 に、シミュレーションに用いた SFC と、それをハイブリッドオートマトンに変換したものを示す。

Fig. 5 SFC used in simulation

Fig. 6 HA used in simulation

Fig. 6 より、 $N_W = 2$ 、 $q(0) = q_0$ 、 $E = \{(q_0, q_0), (q_0, q_1), (q_1, q_0), (q_1, q_1)\}$ となる。

ベクトル場として次式の非線形関数を考える。

$$\dot{x}(t) = a_i x(t) - \frac{a_i}{b_i} x^2(t), i = 0, 1 \quad (17)$$

制御対象のパラメータは $a_0 = 3, a_1 = 0.15, b_0 = 1, b_1 = 0.1$ 、初期条件は $x(0) = 0.1$ とした。Fig. 7 に二つのベクトル場での $x = 0.1, 1.5$ からの状態の軌跡を示す。(17) 式と図より、 $x = b_i$ が平衡点であることがわかる。

さらに、センサ Y_1, Y_2 と切り替え条件の関係は

$$\begin{aligned} [Y_1 = 1] &\Leftrightarrow [g_{0,1}(x, \theta) < 0] \\ [Y_2 = 1] &\Leftrightarrow [g_{1,0}(x, \theta) < 0] \end{aligned} \quad (18)$$

であり、それぞれの切り替え条件は $g_{i,j}(x, \theta) = \theta_{i,1}x + \theta_{i,2}$ 、($i = 0, 1$) でパラメトライズされているとする。

Fig. 7 State trajectories

参照入力として, $r_1(t)$: 周期 0.6[s], 振幅 0.1, オフセット 0.5 の三角波, $r_2(t)$: 周期 0.4[s], 振幅 0.1, オフセット 0.5 の三角波の二つを用いた. また, サンプル時間 $T_s = 0.0625$ [s], 評価区間を $k_f = 40$, 評価関数を

$$J = \int_0^{k_f T_s} (r(t) - x(t))^2 dt \quad (19)$$

とした.

5.1 結果と考察

まず, パラメータを試行錯誤的に変化させてそれに対応する J を求めた. $r_1(t)$ に対し, $\theta_{0,1} = -1, \theta_{1,1} = 1$ とし, $\theta_{0,2}, \theta_{1,2}$ を 0.01 刻みでそれぞれ 40 通り (合計 1600 通り) 変化させた場合の結果を Fig. 8 に示す. 図より, 評価関数は

Fig. 8 Parameters and performance index

パラメータに対し複雑な非線形関数となることがわかる. 二つの参照入力に対する結果を Table 1 に示す.

Table 1 Simulation result (trial-and-error method)

参照入力	Θ_o	Time[s]	min J
$r_1(t)$	$\theta_{0,1} = -1,$ $\theta_{0,2} = 0.28, 0.29, \dots, 0.33,$ $\theta_{1,1} = 1, \theta_{1,2} = -0.5$	450.9	0.0464
$r_2(t)$	$\theta_{0,1} = -1, \theta_{0,2} = 0.45,$ $\theta_{1,1} = 1, \theta_{1,2} = -0.47$	453.2	0.0447

これに対し, 提案手法を用いて J を最小化するパラメータの集合 Θ_o を求めた. 最適化の結果を Table 2, 計算量削

減の効果を Table 3, 最適状態の軌跡を Fig. 9, 10 に示す. $\theta_{i,j}$ は範囲 Θ_o 内でいくつかの値を取り得るが, 図には Θ_o の中の一つの値を示してある. また, パラメータの値は x の係数が ± 1 となるように正規化してある. それぞれの図から, 状態が目標値に近づき, 評価関数を最小にするパラメータが設計されていることがわかる. また, 二つの図を比較すると, 目標値関数の変更に対応して切り替え条件が設計されているのがわかる.

さらに, Table 3 より, 提案した計算量削減手法により計算時間・ノード数が共に削減できていることがわかる.

この結果を試行錯誤的な設計手法と比較すると, 得られた評価関数の最小値に違いがあることがわかる. これは, 評価関数がパラメータに関して複雑な非線形関数となるため, あらかじめ刻み幅を固定してパラメータを与えてしまうとその間の値で評価値がより小さくなりうることを示している.

また, 計算時間については双方とも大差が無いように見える. しかし, 試行錯誤の手法では $\theta_{0,1}, \theta_{1,1}$ の符号を固定したり, パラメータの取り得る範囲をあらかじめ絞り込む ($\theta_{0,2}, \theta_{1,2}$ の動く範囲を決めておく) などを行なった結果の計算時間であり, これらを考慮しない場合計算時間はさらに増大する ($\theta_{0,1}, \theta_{1,1}$ の符号の組み合わせをすべて試すと, 計算時間は 4 倍になる). それに対し提案手法では, パラメータの符号や取り得る範囲を絞りこむ機能が備わっているため, それらを事前に考慮する必要が無い. この点からも, 提案手法がパラメータの設計に有効であることが示された.

Table 2 Simulation result (proposed method)

参照入力	Θ_o	Time[s]	min J
$r_1(t)$	$\theta_{0,1} = -1,$ $0.446 < \theta_{0,2} < 0.450,$ $\theta_{1,1} = 1,$ $-0.460 < \theta_{1,2} < 0.462$	380.9	0.0441
$r_2(t)$	$\theta_{0,1} = -1,$ $0.448 < \theta_{0,2} < 0.450,$ $\theta_{1,1} = 1,$ $-0.476 < \theta_{1,2} < -0.470$	444.4	0.0422

Table 3 Computational amount

$r_1(t)$			
Method	Time[s]	Nodes	min J
No reduction	500.6	7133	0.0441
With reduction	380.9	5339	0.0441
$r_2(t)$			
Method	Time[s]	Nodes	min J
No reduction	569.2	7843	0.0422
With reduction	444.4	6157	0.0422

6. まとめと今後の課題

本論文では, PLC を用いた制御系でのセンサパラメータの

Fig. 9 Simulation result ($r(t) = r_1(t)$) Fig. 10 Simulation result ($r(t) = r_2(t)$)

設計問題に対し、非線形のダイナミクスをもつ制御系にも適用でき、センサパラメータの種類を選ばない(状態との積で現れるパラメータでも適用可能な)手法を提案した。具体的には、まず制御系をハイブリッドオートマトンに変換する手法について述べ、つぎにセンサパラメータの設計問題をハイブリッドオートマトンの切り替え条件の最適化問題に帰着した。最後に、グラフ探索に基づく手法により最適なセンサパラメータの範囲を求める手法を提案し、提案手法の有効性をシミュレーションにより検証した。

今後の課題として、評価関数の最良値の下限の推定をより精度良く行なう方法や、問題に適したノード選択の方法の開発などが挙げられる。

参考文献

- 1) International Electrotechnical Commission (IEC): *Programmable controllers—Part 3: Programming languages* (2003)
- 2) A. Falcione and B. H. Krogh: Design recovery for relay ladder logic, *IEEE Control Systems Magazine*, **13-2**, 90/98 (1993)
- 3) I. Bell: The future of control, *IEE Manufacturing Engineer*, **84-4**, 36/39 (2005)
- 4) M. C. Zhou and E. Twiss: Design of industrial automated systems via relay ladder logic programming and petri nets, *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, **28-1**, 137/150 (1998)
- 5) H. D. Dongargaonkar and R. G. Jamkar: PLC based ig-

- 6) G. Figliolini, M. Ceccarelli and M. T. Llorens Aymerich: Logical sensors and control system programming for an autonomous biped walking robot, In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1346/1351 (2000)
- 7) H. Ashour: Automatic transfer switch (ATS) using programmable logic controller (PLC), In *Proceedings of the IEEE International Conference on Mechatronics 2004*, 531/535 (2004)
- 8) Jean-Jacques E. Slotine and W. Li: *Applied Nonlinear Control*, Prentice-Hall (1991)
- 9) S. Kowalewski, S. Engell, J. Preussig and O. Stursberg: Verification of logic controllers for continuous plants using timed condition/event-systemmodels, *Automatica*, **35-3**, 505/518 (1999)
- 10) A. Bemporad, F. D. Torrisi and M. Morari: Discrete-time hybrid modeling and verification of the batch evaporator process benchmark, *European Journal of Control*, **7-4**, 382/399 (2001)
- 11) E. Konaka, T. Mutou, T. Suzuki and S. Okuma: Optimal design of sensor parameters in PLC-based control system using mixed integer programming, *IEICE Trans. Fundamentals*, **E88-A-4**, 818/824 (2005)
- 12) R. Raman and I. E. Grossmann: Relation between MILP modeling and logical inference for chemical process synthesis, *Computers & Chemical Engineering*, **15-2**, 73/84 (1991)
- 13) L. A. Wolsey: *Integer Programming*, Wiley-Interscience (1998)
- 14) T. A. Henzinger: The theory of hybrid automata, In *Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS96)*, 278/292 (1996)
- 15) M. S. Branicky, V. S. Borkar and S. K. Mitter: A unified framework for hybrid control: Model and optimal control theory, *IEEE Trans. on Automatic Control*, **43-1**, 31/45 (1998)
- 16) J. Peterson: *Petri net theory and the modeling of systems*, Prentice-Hall (1992)
- 17) 村田忠夫: ペトリネットの解析と応用, 近代科学社 (1992)
- 18) 今野 浩: 線形計画法, 43/45, 日科技連 (1987)
- 19) 福島雅夫: 非線形最適化の基礎, 第2章, 朝倉書店 (2001)

[著者紹介]

小中英嗣(正会員)

1978年生まれ。2005年名古屋大学大学院博士後期課程電子情報学専攻修了。工学博士。同年名城大学理工学部講師。2007年同助教。現在に至る。信号・記号ハイブリッドシステム論とその実システムへの応用に関する研究に従事。電気学会、電子情報通信学会、IEEEの各会員。