Silverman アルゴリズムを用いた 実数有限オートマトンの最小実現

渡辺 浩司

(メディア情報文化学科)

有限オートマトン (FA) を離散時間動的システムと捉えることで, 動的システム論 (現代制御路理論) の分野における実数 R 上の線形システムと類似した FA の状態空間モデルが B$(=\{0,1\})$ 上で構成でき, 可到達性・可観測性といった線形システムに対する概念が FA に導入できる。この表現に基づいて論文 [1] では現代制御理論で良く知られる Silverman のアルゴリズムを用いた決定性 FA の最小実現法を提案した。FA の状態空間モデルは B 上, Silverman のアルゴリズムは R 上の線形システムに対するものであるため、アルゴリズムにいくつかの拡張を加えている。本報告ではまず FA の状態空間モデルを自体を R 上に拡張した実数有限オートマトンを導入し、それに対して Silverman のアルゴリズムの適用を試みている。

キーワード:実数有限オートマトン, 状態空間モデル, Silverman アルゴリズム

1 まえがき

有限オートマトン (FA) は記号入力により内部状態を変える離散時間動的システムである. このような動的システムに対して, 動的システム論 (現代制御理論) の分野では実数 R 上のシステムに対しての状態空間モデルによる表現と解析が確立している.

これに対しオートマトン理論では, FA は状態推移を定義する表, 図, および関数等による取り扱いが行なわれているのみであり, 動的システムとしての状態空間モデル表現やアプローチは存在していない.

しかし, 状態と記号を B $(=\{0,1\})$ 上でベクトル化し, 状態推移関数, 受理および初期状態をシステム行列 $(\{A_k\}, \boldsymbol{c}, \boldsymbol{x}_0)$ でパラメータ化することにより, FA はブール半環 $\boldsymbol{B}$ $(=(B,+,\cdot))$ 上の双線形離散時間動的システムとして定式化でき, 状態空間モデルが得られ, その結果, R 上の線形システムと同様に, FA の状態空間モデル表現に対し, 可到達性, 可観測性, 正準分解といった諸概念が定義でき, $\boldsymbol{B}$ 上の状態空間モデルにおいても実現理論が展開できる.

動的システムの入出力データから状態空間モデルのシステムパラメータを求める実現理論は Kalman[2] によって始められ, R 上の線形システムにおける可到達性, 可観測性と最小次元のシステムを求める最小実現との関係が明らかにされた. 現在, 最小実現を行なう種々のアルゴリズムが存在するが, それらは R 上のモデルに対する理論であるため $\boldsymbol{B}$ 上のモデルである FA に用いる場合には何らかの拡張を行なう必要がある.

論文 [1] では最小次元のシステムのシステムパラメータが直接に解として得られる Silverman の実現理論 [2, 4] による決定性 FA (DFA) の最小実現のアルゴリズムを導出した。

本報告では FA の状態空間モデル自体を R 上に拡張した実数有限オートマトンを導入し, 拡張したモデルに対する Silverman のアルゴリズムの適用について検討した.

実現される FA は最小 DFA より少ない状態数の R 上のシステムパラメータを持つ FA となることが明らかとなった.

2　有限オートマトンの状態空間モデル

2.1　状態空間モデル

n 状態, m 入力の FA は次のような B 上の状態空間モデルで表現できる.

$$\begin{cases} \boldsymbol{x}(t+1) = \displaystyle\sum_{k=1}^{m} u_k(t) A_k \boldsymbol{x}(t) \\ y(t) = \boldsymbol{c}^t \boldsymbol{x}(t) \end{cases} \tag{1}$$

$\boldsymbol{x}(\in B^n)$ は状態, $y(\in B)$ は出力, $u_k(\in B)$, $A_k(\in B^{n\times n})$ は記号 a_k $(\in \Sigma:$ 記号集合$)$ の入力とそれによる状態推移, $\boldsymbol{c}(\in B^n)$ は出力ベクトルを表す. 初期状態を $\boldsymbol{x}_0$ とし, $\boldsymbol{x}(0) = \boldsymbol{x}_0$ である. 出力は $y(t) = 1/0$ となり, それぞれ受理/非受理を表す. なお, 式 (1) に対する FA のパラメータ表現を $(\{A_k\}, \boldsymbol{c}, \boldsymbol{x}_0)$ $(k = 1, \cdots, m)$ と表す.

2.2　可到達性および可観測性

入力記号列 $w = a_{k_{t-1}} a_{k_{t-2}} \cdots a_{k_0}$ $(\in \Sigma^*)$ を FA に入力するとき, 右の文字から入力されるものとし, w に対応する状態推移行列 $A(w)$ を次のように定義する.

$$A(w) = A_{k_{t-1}} A_{k_{t-2}} \cdots A_{k_1} A_{k_0} \tag{2}$$

ただし, a_{k_t} $(\in \Sigma)$ は時刻 t における入力記号, A_{k_t} $(\in \{A_1, \cdots, A_m\})$ は a_{k_t} に対応する状態推移行列である.

次の行列 $R_{n,\infty}$ を可到達行列という.

$$\begin{aligned} R_{n,\infty} = [\ & \boldsymbol{x}_0, A_1\boldsymbol{x}_0, \cdots, A_m\boldsymbol{x}_0, A_1A_1\boldsymbol{x}_0, \\ & \cdots, A_mA_1\boldsymbol{x}_0, \cdots, A(w)\boldsymbol{x}_0, \cdots]\ \ (w \in \Sigma^*) \end{aligned} \tag{3}$$

この $R_{n,\infty}$ の i 行目は状態 q_i に対応し, i 行目が零行ベクトルのとき, q_i は不可到達であるといい, 初期状態からその状態へ推移する入力記号列は存在しない. また $R_{n,\infty}$ に零行ベクトルが存在しないとき $(\{A_k\}, \boldsymbol{x}_0)$ は完全可到達であるという.

次の行列

$$O_{\infty,n} = \begin{pmatrix} \boldsymbol{c}^t \\ \vdots \\ \boldsymbol{c}^t A(w) \\ \vdots \end{pmatrix} \quad (w \in \Sigma^*) \tag{4}$$

を可観測行列という. この可観測行列の i 列目は状態 q_i に対応し i 列目が零列ベクトルであるとき, q_i は不可観測であるといい, その状態から受理状態へ推移する入力

記号列は存在しない．また同じ列が存在する時，それらに対応する状態は出力側から見て等価な状態であり，識別不可能と呼ぶ．また $O_{\infty,n}$ に零列ベクトルが存在しないとき $(\{A_k\}, \boldsymbol{c})$ は完全可観測であるという．

2.3　特性応答とハンケル行列

初期状態が $\boldsymbol{x}_0$ で入力記号列が w のとき，状態空間モデル (1) の一般解は次のように書ける．

$$\boldsymbol{x}(t) = A(w)\boldsymbol{x}_0 \quad , \quad y(t) = \boldsymbol{c}^t A(w)\boldsymbol{x}_0 \tag{5}$$

次に状態空間モデル (1) の入出力関係を表す特性応答を次のように定義する．

$$\{h(\varepsilon), h(a_1), \cdots, h(w), \cdots\} \quad ({}^{\forall} w \in \Sigma^*) \tag{6}$$

ここで ε は空記号列とし，$A(\varepsilon)$ は n 次単位行列とする．$h(w)$ は入力記号列が w のときの FA の出力 (応答) であり，次式のように書ける．

$$h(w) = \boldsymbol{c}^t A(w)\boldsymbol{x}_0 \tag{7}$$

この $h(w)$ を用いて，ハンケル行列と呼ばれる無限行列 $\mathcal{H}_{\infty,\infty}$ を次のように構成する．

$$\mathcal{H}_{\infty,\infty} \quad = \quad \begin{array}{c c} & \begin{array}{cccc} \varepsilon & \cdots & v & \cdots \end{array} \\ \begin{array}{c} \varepsilon \\ \vdots \\ r \\ \vdots \end{array} & \begin{pmatrix} h(\varepsilon) & \cdots & h(v) & \cdots \\ \vdots & & \vdots & \\ h(r) & \cdots & h(rv) & \cdots \\ \vdots & & \vdots & \ddots \end{pmatrix} \end{array} \tag{8}$$

ここで $r\ (\in \Sigma^*)$ は行ラベル，$v\ (\in \Sigma^*)$ は列ラベルであり，いずれも辞書順に並べられる．

ハンケル行列 $\mathcal{H}_{\infty,\infty}$ は可観測行列 $O_{\infty,n}$ と可到達行列 $R_{n,\infty}$ の積に分解できる．

$$\mathcal{H}_{\infty,\infty} \quad = \quad O_{\infty,n} R_{n,\infty} \tag{9}$$

ある DFA が完全可到達かつ完全可観測で全状態が識別可能 (完全可識別) であるとき，すなわち，$R_{n,\infty}$ に n 個の単位ベクトルがすべて存在し，$O_{\infty,n}$ の n 本の列ベクトルが全て非零で互いに異なるとき，その DFA の状態空間モデルは最小次元になっている．このとき $\mathcal{H}_{\infty,\infty}$ はその列ベクトルが $O_{\infty,n}$ の n 個の異なる非零列ベクトルと零ベクトルだけからなっている．よって，あるハンケル行列が得られたとき，その列ベクトルのうち異なる非零列ベクトルの数 $\overline{n}$ が最小 DFA の状態数を表していることが分かる．

3　決定性有限オートマトンの最小実現

Silverman のアルゴリズムでは与えられたハンケル行列のランクを求めることが必要である．しかし FA のハンケル行列は減算の定義されていない半環 $\boldsymbol{B}$ 上の行列であり実数 R 上のランクの定義を用いることができない．そこでハンケル行列の異なる非零列ベクトルの数が最小実現される DFA の状態数であるということから, 異なる非零列ベクトルの数をハンケル行列のランクとし rank_D で表す．さらにハンケル行列の異なる非零行ベクトルの数を行ランクと呼び, rank_{row} で表す.

最小実現アルゴリズム

1. ハンケル行列のランクを $\mathrm{rank}_D\mathcal{H}_{1,1}$, $\mathrm{rank}_D\mathcal{H}_{2,2}$, $\cdots$ と順に求め,

$$\mathrm{rank}_D\mathcal{H}_{c,c} = \mathrm{rank}_D\mathcal{H}_{c+1,c+1} = \cdots = \overline{n} \tag{10}$$

   となる最小の c を求める．$\overline{n}$ はハンケル行列の列ランクであり最小実現される DFA の状態数となる．次に

$$\mathrm{rank}_{row}\mathcal{H}_{r,c} = \mathrm{rank}_{row}\mathcal{H}_{r+1,c} = \cdots = \overline{n} \tag{11}$$

   を満たす最小の r を求める．ただし, 行ランクがランクに満たない場合は r が求められず、本アルゴリズムの適用はできない.

2. 部分ハンケル行列 $\mathcal{H}_{r,c}$ を構成する.
3. $\mathcal{H}_{r,c}$ から異なる非零列ベクトルを左から順に抜き出し $P_{r,\overline{n}}$ を構成する.
4. $\mathcal{H}_{r,c}^{(k)}$ を $\mathcal{H}_{r,c}$ の r_i 行 c_j 列の値を入力記号列 $r_i a_k c_j$ に対する出力とした行列とし, この $\mathcal{H}_{r,c}^{(k)}$ から $P_{r,\overline{n}}$ に対応する列を抜き出し $P_{r,\overline{n}}^{(k)}$ を構成する.
5. $P_{r,\overline{n}}$ から異なる行を抜き出し $\overline{P}_{\overline{n},\overline{n}}$ を構成する.
6. $P_{r,\overline{n}}^{(k)}$ から $\overline{P}_{\overline{n},\overline{n}}$ に対応する行を抜き出し $\overline{P}_{\overline{n},\overline{n}}^{(k)}$ を構成する.
7. 次式より最小 DFA のシステムパラメータ $(\{\overline{A}_k\}, \overline{\boldsymbol{c}}, \overline{\boldsymbol{x}}_0)$ が得られる.

$$\overline{A}_k = \overline{P}_{\overline{n},\overline{n}}^{-1}\overline{P}_{\overline{n},\overline{n}}^{(k)} \tag{12}$$

$$\overline{\boldsymbol{c}}^t = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} P_{r,\overline{n}} \tag{13}$$

$$\overline{\boldsymbol{x}}_0 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^t \tag{14}$$

R 上の行列ではこの行ランクはランクと一致するが $\boldsymbol{B}$ 上の行列のランクを上のように定義した場合は両者は一致しないことがある．そのため上述のアルゴリズムを直接適用できないハンケル行列が存在する。このような場合の最小実現法はここでは省略する。

4　実数有限オートマトンの最小実現

4.1　FA の状態空間モデルの実数上への拡張

FA のシステム行列 $(\{A_k\}, \boldsymbol{c}, \boldsymbol{x}_0)$ を R 上の行列とするモデルを実数有限オートマトン (RFA) と呼ぶ. RFA の状態空間モデルでは B 上の値をとる出力変数 $y(t)$ に関する方程式にしきい値関数 $g(\cdot)$

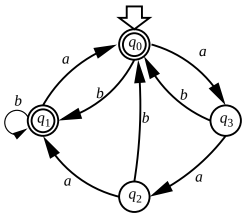
$$g(\alpha) = \begin{cases} 1 & (\alpha \geq \theta) \\ 0 & (\alpha < \theta) \end{cases} \quad (\theta \text{ はしきい値}) \tag{15}$$

を導入し, $y(t) = g(\boldsymbol{c}^T \boldsymbol{x}(t))$ とする.

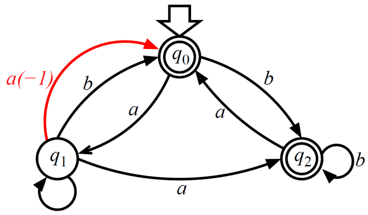
4.2　RFA の最小実現

FA の状態空間モデルを R 上に拡張することで, Silverman のアルゴリズムを $\boldsymbol{B}$ 上への拡張なしに FA に適用できる. ここでは実際の最小実現例を挙げておく (紙面の都合により実現の過程は省略する).

以下の DFA(最小 DFA)



を RFA と捉え、Silverman アルゴリズムを適用すると以下の最小 RFA が得られる.



この最小 RFA のシステムパラメータは

$$\widetilde{A}_a = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \widetilde{A}_b = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{16}$$

$$\widetilde{\mathbf{c}}^T = (1 \quad 0 \quad 1) \tag{17}$$

$$\widetilde{\mathbf{x}}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{18}$$

である。

最小 DFA $\overline{M}$ の状態数 $\overline{n}$ と, Silverman アルゴリズムにより得られた最小 RFA $\widetilde{M}$ の状態数 $\widetilde{n}$ には次の関係が存在する.

$$\overline{n} \geq \widetilde{n} \tag{19}$$

これは $\overline{n}$ がハンケル行列の非零の互いに異なる列数, $\widetilde{n}$ がハンケル行列の R 上のランクであることから明らかである. つまり, RFA を導入することによって, より少ない次数での FA の構成が可能となる. 上述の例でも 4 状態の最小 DFA が 3 状態の最小 RFA として実現されている。また, FA のハンケル行列は B 上の行列であることから, FA $\widetilde{M}$ は実際には有理数上のシステムパラメータを持つ FA(QFA) となる.

5 むすび

本報告では実数有限オートマトン (RFA) を導入し, 線形システム論で良く知られる Silverman アルゴリズムによる RFA の最小実現について検討した。今後は RFA の言語受理能力については検討し他のアナログ計算機械との比較を行いたいと考えている.

参考文献

[1] 渡辺,"Silverman アルゴリズムに基づく決定性有限オートマトンの最小実現", 福山大学人間文化学部紀要, Vol11, pp.1–8(2011).

[2] R. E. Kalman, P. L. Falb and M. A. Arbib, "Topics in Mathematical System Theory", McGraw-Hill(1969).

[3] S. Attasi, "Modelling and Recursive Estimation for Double Indexed Sequences", in R. K. Mehra and D. G. Lainiotis ed. "System Identification Advances and Case Studies", pp.289-348 (1976).

[4] 相良, 秋月, 中津, 片山, "システム同定", 計測自動制御学会 (1981).

Minimal Realization for Real Finite Automata using Silverman's Algorithm

Koji WATANABE

Regarding finite automata (FAs) as discrete time dynamical systems, they can be represented as state space models over $B(=\{0,1\})$ similar to the representation method of linear systems over the real numbers (R) in the field of dynamical systems and controls. Based on this representation, we first propose a minimal realization method for deterministic FAs using Silverman's algorithm which is well-known method in the field of dynamical systems and controls. Since state space models of FAs are bilinear over B, and Silverman's algorithm is, on the other hand, one for linear system over R, we add some extensions to their algorithm. We next extend state space models of FAs to those over R, and call them real finite automata. We then apply Silverman's algorithm to them.

real finite automata, state space model, Silverman's algorithm