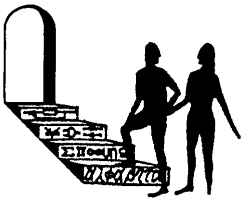
システム/制御/情報，Vol. 56, No. 5, pp. 263–270, 2012



講　座



# 離散事象システムのスーパバイザ制御理論—II
## —最大可制御部分言語の計算とその応用

高井　重昌*

## 1. はじめに

前回第1回目の講座 [1] では，離散事象システムのスーパバイザ制御問題 [2] を定式化し，与えられた制御仕様がスーパバイザ制御のもとで満足されるために，仕様を表す言語が満足すべき可制御性の概念を紹介した．そして，その可制御性の概念を用いて，制御仕様を満足させるスーパバイザが存在するための必要十分条件を示した．その必要十分条件が満足されない場合には，制御仕様を満足する事象列の生起をできるだけ多く許容する最大許容スーパバイザ [2] を構成することになる．

そこで，第2回目の本稿では，まず最大許容スーパバイザの構成において重要な役割を果たす最大可制御部分言語 [2] を定義し，その存在性および計算方法 [3] について説明する．そして，最大許容スーパバイザの存在性，および最大可制御部分言語を用いたその構成法について述べる．さらに，事象が生起するたびに，最大可制御部分言語の計算をオンラインで実行する先読み戦略を用いたスーパバイザ制御 [4] を紹介する．

## 2. 準備

### 2.1 オートマトンと言語

本稿で対象とする離散事象システムは，オートマトン $G=(Q,\ \Sigma,\ \delta,\ q_0,\ Q_m)$ でモデル化されるとする．ここで，$Q$ は状態の集合，$\Sigma$ は事象の有限集合，$\delta: Q\times\Sigma\to Q$ は状態遷移関数，$q_0\in Q$ は初期状態，$Q_m\subseteq Q$ はマーク状態の集合である．

$\Sigma^*$ は空列 $\varepsilon$ を含む，$\Sigma$ の要素で構成されるすべての有限列の集合とする．状態遷移関数 $\delta$ はその定義域を $Q\times\Sigma^*$ へ拡張することができる [1]．任意の $q\in Q$ と任意の $s\in\Sigma^*$ に対して，$\delta(q,s)!$ は $q$ から $s$ による状態遷移が定義されることを表す．また，$\delta(q,s)!$ の否定を $\neg\delta(q,s)!$ で表す．オートマトン $G$ の生成言語 $L(G)$ は

$$L(G)=\{s\in\Sigma^* \mid \delta(q_0,s)!\}$$

と定義される [1]．また，$G$ のマーク言語 $L_m(G)$ は

$$L_m(G)=\{s\in L(G) \mid \delta(q_0,s)\in Q_m\}$$

と定義される [1]．任意の言語 $L\subseteq\Sigma^*$ に対して，その要素のすべての接頭語 [1] からなる集合を $\overline{L}$ で表す．つまり，

$$\overline{L}=\{t\in\Sigma^* \mid \exists u\in\Sigma^*: tu\in L\}$$

である．$L=\overline{L}$ が成り立つとき，$L$ は（接頭語に関して）閉じているという．

本稿で用いる言語上の演算をまとめておく．任意の言語 $L_1,L_2\subseteq\Sigma^*$ に対して，その連接 $L_1L_2\subseteq\Sigma^*$ は

$$L_1L_2=\{st\in\Sigma^* \mid s\in L_1,\ t\in L_2\}$$

差集合 $L_1-L_2$ は

$$L_1-L_2=\{s\in\Sigma^* \mid s\in L_1,\ s\notin L_2\}$$

商集合 $L_1\backslash L_2$ は

$$L_1\backslash L_2=\{s\in\Sigma^* \mid \exists t\in L_2: st\in L_1\}$$

と定義される．また，任意の言語 $L\subseteq\Sigma^*$ と任意の事象列 $s\in\Sigma^*$ に対して，$s$ に続いて生起することで $L$ の要素となる事象列の集合を $L/s$ とする．つまり，

$$L/s=\{t\in\Sigma^* \mid st\in L\}$$

である．$L/s$ を商集合と混同しないよう注意されたい．さらに，任意の非負整数 $N$ に対して，$L$ の長さ $N$ 以下の要素の集合を $L|_N$ とする．つまり，

$$L|_N=\{s\in L \mid |s|\le N\}$$

である．ただし，任意の $s\in\Sigma^*$ に対して，その長さを $|s|$ で表す[1]．なお，空列 $\varepsilon$ は長さが 0，すなわち $|\varepsilon|=0$ であることに注意されたい．

### 2.2 スーパバイザ制御

事象の集合 $\Sigma$ を二つの部分集合 $\Sigma_c$ と $\Sigma_{uc}$ に分割する [1]．ここで，$\Sigma_c$ はスーパバイザによりその生起が禁止できる可制御事象の集合，$\Sigma_{uc}$ はその生起が禁止できない不可制御事象の集合である．形式的に，スーパバイザは関数 $S: L(G)\to\Gamma$ で表現できる [1]．ここで，$\Gamma=\{\gamma\subseteq\Sigma \mid \Sigma_{uc}\subseteq\gamma\}$ である．各 $s\in L(G)$ に対して，$S(s)\in\Gamma$ は，スーパバイザが $s$ の生起を観測した際に，

* 大阪大学 大学院 工学研究科

***Key Words***: discrete event system, supervisory control, controllability, supremal controllable sublanguage, limited lookahead policy.

[1]任意の有限集合 $A$ に対しては，$|A|$ はその要素数を表すとする．

つぎに生起を許容する事象の集合である．スーパバイザ $S$ によって制御されるシステム $G$ を $S/G$ で表す．$S/G$ によって生成される言語 $L(S/G)$ は，つぎのように帰納的に定義される [1].

- $\varepsilon \in L(S/G)$
- $\forall s \in L(S/G), \forall \sigma \in \Sigma$:

$$s\sigma \in L(S/G) \Leftrightarrow [s\sigma \in L(G) \wedge \sigma \in S(s)]$$

$S/G$ によってマークされる言語 $L_m(S/G)$ を

$$L_m(S/G) = L(S/G) \cap L_m(G)$$

と定義する [1]．$\overline{L_m(S/G)} = L(S/G)$ が成り立つとき，スーパバイザ $S$ はノンブロッキングであるという [1].

言語 $K \subseteq L(G)$ が

$$\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$$

を満足するとき，$K$ は（$L(G)$ と $\Sigma_{uc}$ に関して）可制御であるという [1]．また，$K \subseteq L_m(G)$ が

$$\overline{K} \cap L_m(G) = K$$

を満足するとき，$K$ は $L_m(G)$ に関して閉じているという [1].

制御仕様として与えられた言語に対し，スーパバイザが存在するための必要十分条件として，つぎの命題が成り立つ [1].

**【命題 1】** 任意の空でない部分言語 $K \subseteq L(G)$ に対して，$L(S/G) = K$ なるスーパバイザ $S: L(G) \to \Gamma$ が存在するための必要十分条件は，$K$ が閉じた可制御言語となることである．

**【命題 2】** 任意の空でない部分言語 $K \subseteq L_m(G)$ に対して，$L_m(S/G) = K$ なるノンブロッキングなスーパバイザ $S: L(G) \to \Gamma$ が存在するための必要十分条件は，$K$ が可制御かつ $L_m(G)$ に関して閉じていることである．

## 3. 最大可制御部分言語

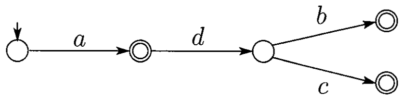
**命題 1**，**命題 2** にあるように，スーパバイザの存在性において，制御仕様として与えられた言語の可制御性が重要な役割を果たす．そこで，本章では，最大許容スーパバイザの構成において有用な最大可制御部分言語の存在性および計算方法を示す．

### 3.1 最大可制御部分言語の存在性

つぎの定理に示すように，可制御性は言語の和のもとで閉じているという望ましい性質をもつ [2].

**【定理 1】** $I$ を添字集合とし，任意の $i \in I$ に対して，言語 $K_i \subseteq L(G)$ は可制御であるとする．このとき，$\bigcup_{i \in I} K_i$ も可制御である．

一方，つぎの例に示すように，可制御性は一般に言語の交わりのもとでは閉じていない．




第 1 図 オートマトン $G$

**【例題 1】** 第 1 図に示されるオートマトン $G$ を考える．なお，初期状態は始点のない矢印で，マーク状態は二重丸で示している．$\Sigma_c = \{a,b,c\}$, $\Sigma_{uc} = \{d\}$ とおくと，二つの言語 $K_1 = \{a,adb\} \subseteq L(G)$ と $K_2 = \{a,adc\} \subseteq L(G)$ は可制御である．しかし，これらの交わり $K_1 \cap K_2 = \{a\}$ に対して，

$$ad \in \overline{K_1 \cap K_2}\Sigma_{uc} \cap L(G) - \overline{K_1 \cap K_2}$$

であるから，可制御ではない．

しかし，閉じた言語に限定すれば，つぎの定理に示すように，可制御性は言語の交わりのもとで閉じている [5].

**【定理 2】** $I$ を添字集合とし，任意の $i \in I$ に対して，$K_i \subseteq L(G)$ は閉じた可制御言語とする．このとき，$\bigcap_{i \in I} K_i$ も閉じた可制御言語である．

任意の言語 $K \subseteq L(G)$ に対して，その可制御な部分言語の集合を $\mathcal{C}(K)$ とおく．つまり，

$$\mathcal{C}(K) = \{K' \subseteq K \mid \overline{K'}\Sigma_{uc} \cap L(G) \subseteq \overline{K'}\}$$

とする．空集合 $\emptyset$ に対して，$\emptyset = \overline{\emptyset}$ かつ

$$\emptyset = \emptyset\Sigma_{uc} \cap L(G) \subseteq \emptyset$$

より，$\emptyset$ は可制御，すなわち $\emptyset \in \mathcal{C}(K)$ である．よって，$\mathcal{C}(K)$ は空集合ではない．$\mathcal{C}(K)$ の集合の包含関係のもとでの最大要素を $\sup\mathcal{C}(K)$ と書き，$K$ の最大可制御部分言語とよぶ．つまり，$\sup\mathcal{C}(K)$ は

$$\forall K' \in \mathcal{C}(K): K' \subseteq \sup\mathcal{C}(K)$$

を満足する $\mathcal{C}(K)$ の要素である．**定理 1** より，$\mathcal{C}(K)$ のすべての要素の和 $\bigcup_{K' \in \mathcal{C}(K)} K'$ は可制御であり，しかも $\bigcup_{K' \in \mathcal{C}(K)} K' \subseteq K$ が成り立つ．よって，$\bigcup_{K' \in \mathcal{C}(K)} K' \in \mathcal{C}(K)$ となることがわかる．さらに，任意の要素 $K'' \in \mathcal{C}(K)$ に対して，$K'' \subseteq \bigcup_{K' \in \mathcal{C}(K)} K'$ となるため，

$$\sup\mathcal{C}(K) = \bigcup_{K' \in \mathcal{C}(K)} K' \tag{1}$$

が成り立つ．つまり，任意の言語 $K \subseteq L(G)$ に対して，最大可制御部分言語 $\sup\mathcal{C}(K)$ は $\mathcal{C}(K)$ のすべての要素の和と一致し，常に存在する．

また，最大可制御部分言語 $\sup\mathcal{C}(K)$ について，つぎの命題が成り立つ [5].

**【命題 3】** 閉じた任意の言語 $K \subseteq L(G)$ に対して，$\sup\mathcal{C}(K)$ もまた閉じている．

【命題 4】 $L_m(G)$ に関して閉じた任意の言語 $K \subseteq L_m(G)$ に対して，$\sup\mathcal{C}(K)$ もまた $L_m(G)$ に関して閉じている．

### 3.2 最大可制御部分言語の計算

$\mathcal{C}(K)$ の要素数は一般に有限とは限らないため，(1) 式は $\sup\mathcal{C}(K)$ の計算の観点からは有用ではない．そこで，オートマトンに基づく $\sup\mathcal{C}(K)$ の計算アルゴリズムを示す．なお，$K=\emptyset$ ならば，明らかに $\sup\mathcal{C}(K)=\emptyset$ であるから，ここでは，$K\neq\emptyset$ の場合を考える．

アルゴリズム中の反復が有限回で終了することを保証するため，$G=(Q, \Sigma, \delta, q_0, Q_m)$ は状態集合 $Q$ が有限である有限オートマトンであり，$K$ は正規言語 [1] であるとする．$K$ は正規言語であるから，$L_m(G_K)=K$ である trim[1] な有限オートマトン $G_K$ が存在する．$G_K$ は trim であるから，$L(G_K)=\overline{K}$ である．このとき，$\sup\mathcal{C}(K)$ はつぎのアルゴリズムにより，計算できる [3].

ステップ 1 有限オートマトン $G=(Q, \Sigma, \delta, q_0, Q_m)$ において，マーク状態の集合 $Q_m$ を $Q$ で置き換えたものを $G'=(Q, \Sigma, \delta, q_0, Q)$ とし，$G_K=(R, \Sigma, \xi, r_0, R_m)$ を $L_m(G_K)=K$ である trim な有限オートマトンとする．

ステップ 2 $G'$ と $G_K$ の同期合成 [1]$G' \| G_K$ を構成し，それに対して Trim 演算 [1] を行った結果のオートマトンを $H_0$ とおく．つまり，

$$
\begin{aligned}
H_0 &= (Z_0, \Sigma, \xi_0, (q_0,r_0), Z_{0,m}) \\
&:= Trim(G' \| G_K)
\end{aligned}
$$

とする．ここで，$Trim(\cdot)$ は Trim 演算 [1] を表し，$Z_0 \subseteq Q\times R$, $Z_{0,m} \subseteq Q\times R_m$ である．そして，$i:=0$ とおく．

ステップ 3 まず，

$$
Z'_i := \{(q,r)\in Z_i \mid \forall\sigma\in\Sigma_{uc}: \delta(q,\sigma)! \Rightarrow \xi_i((q,r),\sigma)!\}
$$

$$
Z'_{i,m} = Z_{i,m} \cap Z'_i
$$

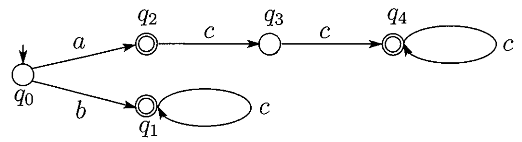
とし，$\xi'_i: Z'_i\times\Sigma\to Z'_i$ を，任意の $z\in Z'_i$ と任意の $\sigma\in\Sigma$ に対して，

$$
\xi'_i(z,\sigma) = \begin{cases} \xi_i(z,\sigma), & \text{if} \quad \xi_i(z,\sigma)\in Z'_i \\ \text{undefined}, & \text{otherwise} \end{cases}
$$

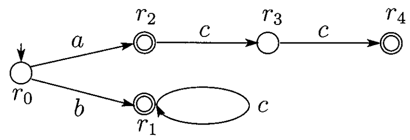
と定義する．そして，$H_{i+1}$ をつぎのように構成する．

$$
H_{i+1} := Trim(Z'_i, \Sigma, \xi'_i, (q_0,r_0), Z'_{i,m})
$$

ステップ 4 もし，$H_{i+1}=H_i$ もしくは $H_{i+1}$ が状態集合が空集合である空オートマトン [1] ならば，$H_{i+1}$ を出力し，アルゴリズムは終了．




第 2 図 オートマトン $G$




第 3 図 オートマトン $G_K$

そうでなければ，

$$
H_{i+1} = (Z_{i+1}, \Sigma, \xi_{i+1}, (q_0,r_0), Z_{i+1,m})
$$

とおき，$i:=i+1$ とし，ステップ 3 へ．

このアルゴリズムにより出力された有限オートマトン $H_{i+1}$ に対して，

$$
L_m(H_{i+1}) = \sup\mathcal{C}(K)
$$

$$
L(H_{i+1}) = \overline{\sup\mathcal{C}(K)}
$$

が成り立ち，$\sup\mathcal{C}(K)$ が計算できる．つまり，$H_{i+1}$ は $\sup\mathcal{C}(K)$ をマークする trim な有限オートマトンである．もし，$H_{i+1}$ が空オートマトンならば，$\sup\mathcal{C}(K)=\emptyset$ である．

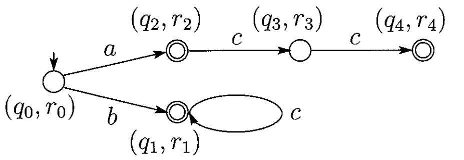
（注意 1） オートマトン $H_i$ の各状態 $(q,r)\in Z_i$ は，$G$ の状態 $q\in Q$ と $G_K$ の状態 $r\in R$ の組である．ステップ 3 では，$G$ において $q$ では生起可能であるが，$H_i$ において $(q,r)$ では生起可能でない不可制御事象が存在するならば，$(q,r)$ とそれに付随する状態遷移を $H_i$ から取り除く．そして，その結果に対して Trim 演算を行うことで，オートマトン $H_{i+1}$ を得る．

（注意 2） ステップ 2 で構成する $H_0=Trim(G'\|G_K)$ は，状態数がたかだか $|Q|\cdot|R|$ の有限オートマトンである．そして，ステップ 3 において，$H_{i+1}$ は $H_i$ からいくつかの状態とそれらに付随する状態遷移を取り除くことによって得られる．よって，たかだか $|Q|\cdot|R|$ 回の反復でこのアルゴリズムは終了し，その計算量は $O(|Q|^2\cdot|R|^2\cdot|\Sigma_{uc}|)$ である．

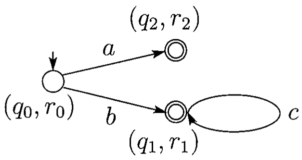
【例題 2】 第 2 図に示す有限オートマトン $G$，および第 3 図に示す trim な有限オートマトン $G_K$ でマークされる言語 $K\subseteq L(G)$ を考える．ここで，$\Sigma_c=\{a,b\}$, $\Sigma_{uc}=\{c\}$ とする．

まずステップ 2 において，マーク状態の集合 $Q_m$ を $Q$ で置き換えた $G'$ と $G_K$ の同期合成に対して，Trim 演算を行い $H_0$ を求めると，第 4 図のようになる．

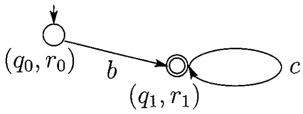
ステップ 3 において，$\delta(q_4,c)!$ かつ $\neg\xi_0((q_4,r_4),c)!$ であるため，




第 4 図 オートマトン $H_0$




第 5 図 オートマトン $H_1$




第 6 図 オートマトン $H_2$

$$Z_0' = \{(q_0,r_0),(q_1,r_1),(q_2,r_2),(q_3,r_3)\}$$

$$\begin{aligned}Z_{0,m}' &= \{(q_1,r_1),(q_2,r_2),(q_4,r_4)\}\cap Z_0' \\ &= \{(q_1,r_1),(q_2,r_2)\}\end{aligned}$$

となる．そして，Trim演算により $H_1$ を求めると，第 5 図のようになる．

ステップ 4 において，$H_1$ は空オートマトンではなく，$H_1 \neq H_0$ であるため，$i := 1$ としてステップ 3 へ戻る．

ステップ 3 において，$\delta(q_2,c)!$ かつ $\neg\xi_1((q_2,r_2),c)!$ であるため，

$$Z_1' = \{(q_0,r_0),(q_1,r_1)\}$$

$$\begin{aligned}Z_{1,m}' &= \{(q_1,r_1),(q_2,r_2)\}\cap Z_1' \\ &= \{(q_1,r_1)\}\end{aligned}$$

である．そして，Trim演算により $H_2$ を求めると，第 6 図のようになる．

ステップ 4 において，$H_2$ は空オートマトンではなく，$H_2 \neq H_1$ であるため，$i := 2$ として再びステップ 3 へ戻る．

ステップ 3 において，

$$Z_2' = \{(q_0,r_0),(q_1,r_1)\}$$

$$Z_{2,m}' = \{(q_1,r_1)\}\cap Z_2' = \{(q_1,r_1)\}$$

であり，Trim 演算により $H_3$ を求めると，第 6 図のオートマトンとなり，$H_3 = H_2$ である．

ステップ 4 において，$H_3 = H_2$ であるから，$H_3$ が出力され，アルゴリズムが終了する．そして，得られた $H_3$ に対して，

$$L_m(H_3) = \sup\mathcal{C}(K)$$

$$L(H_3) = \overline{\sup\mathcal{C}(K)}$$

が成り立つ．

与えられた言語 $K \subseteq L(G)$ が閉じている場合には，$\sup\mathcal{C}(K)$ は $K$ に関して，つぎのように陽に表現することができる [6].

**【定理 3】** $K \subseteq L(G)$ は任意の閉じた言語とする．このとき，

$$\sup\mathcal{C}(K) = K - \{(L(G)-K)\backslash\Sigma_{uc}^*\}\Sigma^* \qquad (2)$$

が成り立つ．

(2) 式において，商集合の定義より，

$$\begin{aligned}&(L(G)-K)\backslash\Sigma_{uc}^* \\ &\quad = \{s\in\Sigma^* \mid \exists t\in\Sigma_{uc}^* : st\in L(G)-K\}\end{aligned}$$

であるから，これは，ある不可制御事象列が続いて生起した結果が $K$ の要素ではない事象列の集合である．$K$ の可制御性は，任意の $s\in K$ において，不可制御事象 $\sigma\in\Sigma_{uc}$ が続いて生起したとしても，$s\sigma\in K$ となることを要求するため，$(L(G)-K)\backslash\Sigma_{uc}^*$ は $K$ の可制御性が満足されないことの要因となる事象列の集合とみなせる．よって，これらの事象列とそれに続く事象列を $K$ から取り除くことによって，$\sup\mathcal{C}(K)$ が得られることを (2) 式は意味している．

### 3.3 最大可制御部分言語のモジュラ計算

離散事象システムの制御問題における制御仕様では，複数の制約を同時に満足することが要求される場合がある．そこで，$I$ を添字集合とし，言語 $K \subseteq L(G)$ が

$$K = \bigcap_{i\in I} K_i \qquad (3)$$

のように，複数の言語の交わりで表現される場合を考える．

$\bigcap_{i\in I} K_i$ の最大可制御部分言語 $\sup\mathcal{C}\left(\bigcap_{i\in I} K_i\right)$ を，各 $K_i$ $(i\in I)$ の最大可制御部分言語 $\sup\mathcal{C}(K_i)$ の交わり $\bigcap_{i\in I}\sup\mathcal{C}(K_i)$ により計算する方法を，モジュラ計算法という．このようなモジュラ計算法は，制御仕様を表す言語 $K$ が (3) 式で与えられる場合に，各 $K_i$ ごとにスーパバイザを構成するモジュラ設計において有用である．このようなスーパバイザのモジュラ設計については，紙面の都合上省略するが，興味のある読者は参考文献 [5] を参照されたい．

$$\sup\mathcal{C}\left(\bigcap_{i\in I} K_i\right) \subseteq \bigcap_{i\in I}\sup\mathcal{C}(K_i) \subseteq \bigcap_{i\in I} K_i$$

が成り立つため，もし $\bigcap_{i\in I}\sup\mathcal{C}(K_i)$ が可制御ならば

$$\sup\mathcal{C}\left(\bigcap_{i\in I}K_i\right)=\bigcap_{i\in I}\sup\mathcal{C}(K_i) \tag{4}$$

が成り立ち，$\sup\mathcal{C}\left(\bigcap_{i\in I}K_i\right)$ のモジュラ計算が可能である．しかし，**3.1** で示したように，一般に可制御性は言語の交わりのもとで閉じていないため，$\bigcap_{i\in I}\sup\mathcal{C}(K_i)$ は可制御であるとは限らず，(4) 式は必ずしも成立しない．そこで，可制御言語の交わりが可制御となるための十分条件として，言語の無競合性という概念が提案されている [5].

**【定義 1】** $I$ を添字集合とする．このとき，言語の集合 $\{K_i\subseteq\Sigma^*\,|\,i\in I\}$ に対して，

$$\overline{\bigcap_{i\in I}K_i}=\bigcap_{i\in I}\overline{K_i}$$

が成り立つならば，$\{K_i\subseteq\Sigma^*\,|\,i\in I\}$ は無競合であるという.

$\overline{\bigcap_{i\in I}K_i}\subseteq\bigcap_{i\in I}\overline{K_i}$ は常に成り立つため，$\{K_i\subseteq\Sigma^*\,|\,i\in I\}$ が無競合であることは，

$$\bigcap_{i\in I}\overline{K_i}\subseteq\overline{\bigcap_{i\in I}K_i}$$

が成り立つこと，つまり，すべての $K_i$ の接頭語である事象列は $\bigcap_{i\in I}K_i$ の接頭語であることを要求している．

**【例題 3】** **例題 1** で考えた二つの言語 $K_1=\{a,adb\}$ と $K_2=\{a,adc\}$ に対して，

$$\overline{K_1}\cap\overline{K_2}=\{\varepsilon,a,ad\}\not\subseteq\{\varepsilon,a\}=\overline{K_1\cap K_2}$$

より，$\{K_1,K_2\}$ は無競合ではない．

**【命題 5】** $I$ を添字集合とし，任意の $i\in I$ に対して，$K_i\subseteq L(G)$ は可制御言語であるとする．このとき，$\{K_i\subseteq L(G)\,|\,i\in I\}$ が無競合であるならば，$\bigcap_{i\in I}K_i$ は可制御である．

**命題 5** より，$\sup\mathcal{C}\left(\bigcap_{i\in I}K_i\right)$ のモジュラ計算に関して，つぎの結果が成り立つ [5].

**【定理 4】** $I$ を添字集合とし，各言語 $K_i\subseteq L(G)$ $(i\in I)$ の最大可制御部分言語からなる集合 $\{\sup\mathcal{C}(K_i)\,|\,i\in I\}$ が無競合ならば，(4) 式が成り立つ．

各 $K_i$ が閉じた言語である場合，**定理 2** と**命題 3** より，$\bigcap_{i\in I}\sup\mathcal{C}(K_i)$ は可制御である．したがって，つぎの定理が成り立つ [5].

**【定理 5】** $I$ を添字集合とし，任意の $i\in I$ に対して，$K_i\subseteq L(G)$ は閉じた言語とする．このとき，(4) 式が成り立つ．

**定理 5** より，各 $K_i$ $(i\in I)$ が閉じているならば，$\sup\mathcal{C}\left(\bigcap_{i\in I}K_i\right)$ のモジュラ計算が可能である．

## 4. 最大許容スーパバイザ

本章では，制御仕様として与えられた言語 $K\subseteq L(G)$ が**命題 1**，**命題 2** のスーパバイザの存在条件を満足しない場合を考える．そして，最大許容スーパバイザを定義し，その存在性や構成法について述べる．

### 4.1 制御仕様が生成言語に関して与えられる場合

制御仕様が生成言語 $L(G)$ に関して，$K\subseteq L(G)$ で与えられる場合，**命題 1** より，$L(S/G)=K$ なるスーパバイザ $S:L(G)\to\Gamma$ が存在するための必要十分条件は，$K$ が閉じた可制御言語となることである．よって，もし $K$ が閉じた可制御言語でないならば，$L(S/G)=K$ なるスーパバイザ $S$ は存在しない．このような場合には，つぎの二条件を満足するスーパバイザ $S:L(G)\to\Gamma$ を構成する問題を考える．

$$L(S/G)\subseteq K \tag{5}$$

$$\begin{aligned}&\forall S':L(G)\to\Gamma:\\&L(S'/G)\subseteq K\Rightarrow L(S'/G)\subseteq L(S/G)\end{aligned} \tag{6}$$

(5) 式の条件は，制御仕様として与えられた $K$ の要素である事象列のみが，$S/G$ で生起することを要求する．さらに，(6) 式は，(5) 式の条件を満たすスーパバイザのなかで，生成言語 $L(S/G)$ が集合の包含関係の意味で最大となることを要求する．(5) 式と (6) 式の条件を満足するスーパバイザ $S$ は最大許容スーパバイザとよばれる [2]. そのような最大許容スーパバイザ $S$ は

$$\begin{aligned}&\forall S':L(G)\to\Gamma:\\&L(S'/G)\subseteq K\Rightarrow L(S'/G)\subseteq L(S/G)\subseteq K\end{aligned}$$

を満足することにより，$L(S/G)$ はスーパバイザ制御のもとで生成される $K$ の部分言語のなかで，$K$ の最もよい近似であるとみなせる．

つぎに，(5) 式と (6) 式の条件を満足する最大許容スーパバイザの存在性について考える．言語 $K\subseteq L(G)$ に対して，その閉じた可制御部分言語の集合を $\mathcal{PC}(K)$ とおく．つまり，

$$\mathcal{PC}(K)=\{K'\subseteq K\,|\,K'=\overline{K'},\ K'\Sigma_{uc}\cap L(G)\subseteq K'\}$$

である．$\emptyset\in\mathcal{PC}(K)$ が成り立つため，$\mathcal{PC}(K)$ は空集合ではない．閉じた言語の和をとると，その結果もまた閉じた言語となる．すなわち，$I$ を添字集合とし，任意の $i\in I$ に対して，$K_i=\overline{K_i}$ ならば，

$$\bigcup_{i\in I}K_i=\overline{\bigcup_{i\in I}K_i}$$

が成り立つ．よって，**定理 1** より，$\bigcup_{K' \in \mathcal{PC}(K)} K' \in \mathcal{PC}(K)$ となる．したがって，集合の包含関係のもとでの $\mathcal{PC}(K)$ の最大要素 $\sup\mathcal{PC}(K)$ が存在し，

$$\sup\mathcal{PC}(K) = \bigcup_{K' \in \mathcal{PC}(K)} K'$$

が成り立つ．$\sup\mathcal{PC}(K)$ は $K$ の閉じた最大可制御部分言語とよばれる．

(5) 式と (6) 式の条件を満足する最大許容スーパバイザの存在性について，つぎの結果が成り立つ．

**【定理 6】**　任意の言語 $K \subseteq L(G)$ に関して，以下の三つの項目は等価である．

(1) (5) 式と (6) 式の条件を満足する最大許容スーパバイザ $S: L(G) \to \Gamma$ が存在する．

(2) $\sup\mathcal{PC}(K) \neq \emptyset$ が成り立つ．

(3) $\Sigma_{uc}^* \cap L(G) \subseteq K$ が成り立つ．

**定理 6** より，(5) 式と (6) 式の条件を満足する最大許容スーパバイザが存在することは，$K$ の閉じた最大可制御部分言語が空でないことと等価であることがわかる．さらに，不可制御事象のみからなる事象列が生起可能な場合，それが必ず $K$ の要素であることとも等価となる．

言語 $K \subseteq L(G)$ に対して，(5) 式と (6) 式の条件を満足する最大許容スーパバイザ $S: L(G) \to \Gamma$ が存在する場合，

$$L(S/G) = \sup\mathcal{PC}(K)$$

が成り立つ．参考文献 [1] の **5.3** で述べたように，最大許容スーパバイザ $S$ は，

$$L(G_{\sup\mathcal{PC}(K)}) = L_m(G_{\sup\mathcal{PC}(K)}) = \sup\mathcal{PC}(K)$$

なるオートマトン $G_{\sup\mathcal{PC}(K)}$ により実現できる．

(5) 式と (6) 式の条件を満足する最大許容スーパバイザを構成するためには，$\sup\mathcal{PC}(K)$ を計算する必要がある．そこで，$K$ の閉じた部分言語の集合を $\mathcal{P}(K)$ とおく．つまり，

$$\mathcal{P}(K) = \{K' \subseteq K \mid K' = \overline{K'}\}$$

とする．$\emptyset \in \mathcal{P}(K)$ より，$\mathcal{P}(K)$ は空でなく，集合の包含関係のもとでのその最大要素 $\sup\mathcal{P}(K)$ は常に存在し，

$$\sup\mathcal{P}(K) = K - (\Sigma^* - K)\Sigma^* \tag{7}$$

が成り立つ [7]．そして，$\sup\mathcal{PC}(K)$ の計算について，**命題 3** より，以下の結果が成り立つ．

**【定理 7】**　任意の言語 $K \subseteq L(G)$ に対して，

$$\sup\mathcal{PC}(K) = \sup\mathcal{C}(\sup\mathcal{P}(K))$$

が成り立つ．

**定理 7** より，まず (7) 式により $\sup\mathcal{P}(K)$ を計算し，そして **3.2** で示した最大可制御部分言語の計算アルゴリズムにより $\sup\mathcal{C}(\sup\mathcal{P}(K))$ を計算することで，$\sup\mathcal{PC}(K)$ が得られる．

さらに，$K$ が閉じた言語の場合，$\sup\mathcal{P}(K) = K$ であるから，つぎの系が成り立つ．

**【系 1】**　任意の閉じた言語 $K \subseteq L(G)$ に対して，

$$\sup\mathcal{PC}(K) = \sup\mathcal{C}(K)$$

が成り立つ．

### 4.2　制御仕様がマーク言語に関して与えられる場合

制御仕様がマーク言語 $L_m(G)$ に関して，$K \subseteq L_m(G)$ で与えられる場合，**命題 2** より，$L_m(S/G) = K$ なるノンブロッキングなスーパバイザ $S: L(G) \to \Gamma$ が存在するための必要十分条件は，$K$ が可制御かつ $L_m(G)$ に関して閉じていることである．$K$ が可制御かつ $L_m(G)$ に関して閉じた言語でないならば，つぎの二条件を満足するノンブロッキングな最大許容スーパバイザ $S: L(G) \to \Gamma$ を構成する．

$$L_m(S/G) \subseteq K \tag{8}$$

$$\begin{aligned}&\forall S': L(G) \to \Gamma:\\&[\overline{L_m(S'/G)} = L(S'/G) \wedge L_m(S'/G) \subseteq K]\\&\ \Rightarrow L_m(S'/G) \subseteq L_m(S/G)\end{aligned} \tag{9}$$

言語 $K \subseteq L_m(G)$ に対して，その $L_m(G)$ に関して閉じた可制御部分言語の集合を $\mathcal{LC}(K)$ とおく．つまり，

$$\begin{aligned}\mathcal{LC}(K) = \{K' \subseteq K \mid\ &\overline{K'} \cap L_m(G) = K',\\&\overline{K'}\Sigma_{uc} \cap L(G) \subseteq \overline{K'}\}\end{aligned}$$

である．$\emptyset \in \mathcal{LC}(K)$ が成り立つため，$\mathcal{LC}(K)$ は空集合ではない．そして，$I$ を添字集合とし，任意の $i \in I$ に対して，$\overline{K_i} \cap L_m(G) = K_i$ とすると，

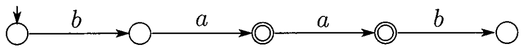
$$\overline{\bigcup_{i \in I} K_i} \cap L_m(G) = \bigcup_{i \in I} K_i$$

が成り立ち，$\bigcup_{i \in I} K_i$ もまた $L_m(G)$ に関して閉じている．よって，**定理 1** より，$\bigcup_{K' \in \mathcal{LC}(K)} K' \in \mathcal{LC}(K)$ となる．したがって，集合の包含関係のもとでの $\mathcal{LC}(K)$ の最大要素 $\sup\mathcal{LC}(K)$ が存在し，

$$\sup\mathcal{LC}(K) = \bigcup_{K' \in \mathcal{LC}(K)} K'$$

が成り立つ．$\sup\mathcal{LC}(K)$ を $K$ の $L_m(G)$ に関して閉じた最大可制御部分言語という．

(8) 式と (9) 式の条件を満足するノンブロッキングな最大許容スーパバイザの存在性について，つぎの結果が




第 7 図　オートマトン $G$

得られる．

【**定理 8**】　任意の言語 $K \subseteq L_m(G)$ に関して，(8) 式と (9) 式の条件を満足するノンブロッキングな最大許容スーパバイザ $S: L(G) \to \Gamma$ が存在するための必要十分条件は，$\sup\mathcal{LC}(K) \neq \emptyset$ となることである．

（注意 3）つぎの例に示すように，$K$ が生成言語に関して与えられる場合の**定理 6** とは異なり，ノンブロッキングな最大許容スーパバイザが存在することと $\Sigma_{uc}^* \cap L(G) \subseteq K$ が成り立つことは，等価ではないことに注意されたい．

【**例題 4**】　第 7 図に示すオートマトン $G$ において，$\Sigma_c = \{a\}$, $\Sigma_{uc} = \{b\}$ とする．このとき，言語 $K = \{ba, baa\}$ に対して，$\sup\mathcal{LC}(K) = \{ba\}$ となり，(8) 式と (9) 式の条件を満足するノンブロッキングな最大許容スーパバイザ $S: L(G) \to \Gamma$ が存在し，$L_m(S/G) = \sup\mathcal{LC}(K)$ となる．しかし，

$$b \in (\Sigma_{uc}^* \cap L(G)) - K$$

であるため，$\Sigma_{uc}^* \cap L(G) \subseteq K$ は成り立たない．

言語 $K \subseteq L_m(G)$ に対して，(8) 式と (9) 式の条件を満足するノンブロッキングな最大許容スーパバイザ $S: L(G) \to \Gamma$ は

$$L_m(S/G) = \sup\mathcal{LC}(K)$$

$$L(S/G) = \overline{\sup\mathcal{LC}(K)}$$

を満足する．したがって，参考文献 [1] の **5.3** より，ノンブロッキングな最大許容スーパバイザ $S$ は，

$$L(G_{\sup\mathcal{LC}(K)}) = L_m(G_{\sup\mathcal{LC}(K)}) = \overline{\sup\mathcal{LC}(K)}$$

なるオートマトン $G_{\sup\mathcal{LC}(K)}$ により実現できる．

(8) 式と (9) 式の条件を満足するノンブロッキングな最大許容スーパバイザを構成するには，$\sup\mathcal{LC}(K)$ の計算が必要である．そこで，

$$\mathcal{L}(K) = \{K' \subseteq K \mid \overline{K'} \cap L_m(G) = K'\}$$

とおく．つまり，$\mathcal{L}(K)$ は $K$ の $L_m(G)$ に関して閉じた部分言語の集合である．$\emptyset \in \mathcal{L}(K)$ であり，集合の包含関係のもとでのその最大要素 $\sup\mathcal{L}(K)$ は常に存在し，

$$\sup\mathcal{L}(K) = K - (L_m(G) - K)\Sigma^* \tag{10}$$

が成り立つ [7]．そして，$\sup\mathcal{LC}(K)$ の計算について，**命題 4** より，つぎの結果が得られる．

【**定理 9**】　任意の言語 $K \subseteq L_m(G)$ に対して，

$$\sup\mathcal{LC}(K) = \sup\mathcal{C}(\sup\mathcal{L}(K))$$

が成り立つ．

**定理 9** より，$\sup\mathcal{LC}(K)$ を得るには，まず (10) 式により $\sup\mathcal{L}(K)$ を計算し，そして **3.2** の最大可制御部分言語の計算アルゴリズムを用いて $\sup\mathcal{C}(\sup\mathcal{L}(K))$ を計算すればよいことがわかる．

さらに，$K$ が $L_m(G)$ に関して閉じている場合，$\sup\mathcal{L}(K) = K$ となるため，つぎの系が得られる．

【**系 2**】　$L_m(G)$ に関して閉じた任意の言語 $K \subseteq L_m(G)$ に対して，

$$\sup\mathcal{LC}(K) = \sup\mathcal{C}(K)$$

が成り立つ．

## 5.　先読み戦略を用いたスーパバイザ制御

**4.** で述べた最大許容スーパバイザを構成するためには，制御仕様として与えられた言語の最大可制御部分言語の計算が必要である．**3.2** で示したように，最大可制御部分言語の計算は，制御対象である離散事象システムと制御仕様の有限オートマトンモデルを用いて行われるが，大規模・複雑なシステムを対象とする場合には，モデルの状態数が増加し，最大可制御部分言語の計算が困難になるという問題がある．そこで，Chung らは，事象が生起するたびに，つぎに生起を許容する事象を将来の $N$ ステップの振舞いに基づきオンラインで決定する，スーパバイザ制御を提案している [4]．ここで，$N$ は自然数である．このようなスーパバイザ制御法は先読み戦略とよばれている．本章では，参考文献 [4] で提案された，先読み戦略を用いたスーパバイザ制御を紹介する．

制御仕様として与えられた言語 $K \subseteq L_m(G)$ は $L_m(G)$ に関して閉じていると仮定する[1]．任意の事象列 $s \in L(G)$ に対して，将来の $N$ ステップの振舞いは

$$L(G)/s|_N = \{t \in \Sigma^* \mid st \in L(G),\ |t| \leq N\}$$

で表すことができる．そして，$L(G)/s|_N$ に対する制御仕様を考える．このとき，長さが $N$ である $\overline{K}/s$ の要素，すなわち $\overline{K}/s|_N - \overline{K}/s|_{N-1}$ の要素の扱いが重要となる．任意の $t \in \overline{K}/s|_N - \overline{K}/s|_{N-1}$ に対し，$N$ ステップ以降の振舞いにおいて，$stu \in K$ となる事象列 $u \in \Sigma^*$ が存在する一方，ある不可制御事象列 $v \in \Sigma_{uc}^*$ の生起により，$stv \in L(G) - \overline{K}$ となり，制御仕様を満足しなくなることも考えられる．したがって，$N$ ステップの振舞いの情報 $L(G)/s|_N$ だけでは，$t$ を許可すべき望ましい事象列か否かを判断することができない．そこで，参考文献 [4] では，$\overline{K}/s|_N - \overline{K}/s|_{N-1}$ の要素の扱いについて，保守的方針と楽観的方針の二つの方針が提案されている．保守的方針では，$\overline{K}/s|_N - \overline{K}/s|_{N-1}$ の要素はすべて望まし

[1] もし $K \subseteq L_m(G)$ が $L_m(G)$ に関して閉じていないならば，$L_m(G)$ に関して閉じた最大部分言語 $\sup\mathcal{L}(K)$ を考えればよい．

くない事象列とみなし，$L(G)/s|_N$ に対する制御仕様を

$$K/s|_N - (\overline{K}/s|_N - \overline{K}/s|_{N-1}) = K/s|_{N-1}$$

とする．一方，楽観的方針では，$\overline{K}/s|_N - \overline{K}/s|_{N-1}$ の要素はすべて望ましい事象列とみなし，$L(G)/s|_N$ に対する制御仕様を

$$K/s|_N \cup (\overline{K}/s|_N - \overline{K}/s|_{N-1})$$

とする．楽観的方針を採用した場合，制御仕様を満足する事象列のみが生起することは一般に保証されないため，ここでは，保守的方針のみを考え，$L(G)/s|_N$ に対する制御仕様を $K/s|_{N-1}$ とする．そして，$K/s|_{N-1}$ の $L(G)/s|_N$ と $\Sigma_{uc}$ に関する最大可制御部分言語を $\sup\mathcal{C}(K/s|_{N-1})_{L(G)/s|_N}$ と書く．

$N$ ステップの先読み戦略を用いたスーパバイザ $S^N: L(G) \to \Gamma$ を，任意の $s \in L(G)$ に対して，

$$\begin{aligned} &S^N(s) \\ &= (\overline{\sup\mathcal{C}(K/s|_{N-1})_{L(G)/s|_N}}|_1 - \{\varepsilon\}) \cup \Sigma_{uc} \end{aligned} \qquad (11)$$

と構成する．つまり，$S^N$ は事象列 $s \in L(G)$ に対して，最大可制御部分言語 $\sup\mathcal{C}(K/s|_{N-1})_{L(G)/s|_N}$ の1ステップ目の事象と不可制御事象の生起を許容する．

$N$ ステップの先読み戦略を用いたスーパバイザ $S^N: L(G) \to \Gamma$ について，以下の結果が得られている [4].

**【定理 10】** 任意の空でない部分言語 $K \subseteq L_m(G)$ に対して，(11)式で与えられるスーパバイザ $S^N: L(G) \to \Gamma$ を考える．もし $\sup\mathcal{C}(K|_{N-1})_{L(G)|_N} \neq \emptyset$ ならば，$S^N$ は $L_m(S^N/G) \subseteq K$ なるノンブロッキングなスーパバイザである．ここで，$\sup\mathcal{C}(K|_{N-1})_{L(G)|_N}$ は，$K|_{N-1}$ の $L(G)|_N$ と $\Sigma_{uc}$ に関する最大可制御部分言語である．

（注意 4） (11)式で与えられたスーパバイザ $S^N: L(G) \to \Gamma$ を用いたスーパバイザ制御では，あらかじめすべての $s \in L(G)$ に対して $S^N(s)$ をオフラインで計算しておく必要はなく，事象が生起するたびに，将来の $N$ ステップの振舞いに関する最大可制御部分言語の計算をオンラインで行うことで，$S^N(\cdot)$ を計算すればよい．

## 6. おわりに

本稿ではまず，最大可制御部分言語を定義し，その存在性および計算のためのアルゴリズムを示した．そして，制御仕様として与えられた言語がスーパバイザの存在条件を満足しない場合において，最大許容スーパバイザの概念を紹介し，最大可制御部分言語の計算を用いたその構成法について述べた．さらに，事象が生起するたびに，将来の $N$ ステップの振舞いに関する最大可制御部分言語の計算をオンラインで実行する先読み戦略についても紹介した．

前回と今回の講座では，スーパバイザは対象システムで生起した事象がすべて観測できる完全観測を仮定していた．事象の部分的な観測のもとでのスーパバイザ制御については，次回の第3回目で詳しく解説する．

(2012年1月5日受付)

## 参 考 文 献


[1] 高井：離散事象システムのスーパバイザ制御理論—I— スーパバイザと可制御性；システム／制御／情報, Vol. 56, No. 3, pp. 144–151 (2012)

[2] P. J. Ramadge and W. M. Wonham: Supervisory control of a class of discrete event processes; *SIAM J. Contr. Optim.*, Vol. 25, No. 1, pp. 206–230 (1987)

[3] W. M. Wonham and P. J. Ramadge: On the supremal controllable sublanguage of a given language; *SIAM J. Contr. Optim.*, Vol. 25, No. 3, pp. 637–659 (1987)

[4] S.-L. Chung, S. Lafortune and F. Lin: Limited lookahead policies in supervisory control of discrete event systems; *IEEE Trans. Autom. Contr.*, Vol. 37, No. 12, pp. 1921–1935 (1992)

[5] W. M. Wonham and P. J. Ramadge: Modular supervisory control of discrete event systems; *Math. Contr. Signals Syst.*, Vol. 1, No. 1, pp. 13–30 (1988)

[6] R. D. Brandt, V. Garg, R. Kumar, F. Lin, S. I. Marcus and W. M. Wonham: Formulas for calculating supremal controllable and normal sublanguages; *Syst. Contr. Lett.*, Vol. 15, No. 2, pp. 111–117 (1990)

[7] R. Kumar and V. K. Garg: *Modeling and Control of Logical Discrete Event Systems*, Kluwer Academic Publishers (1995)


## 著 者 略 歴


高井（たかい） 重昌（しげまさ） (正会員)

学会誌 Vol. 56, No. 3, p. 151 参照.