

Title	時間付き離散事象システムの言語安定性の研究
Author(s)	盧, 吉錫
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1348
Rights	
Description	Supervisor: 平石 邦彦, 情報科学研究科, 修士

修 士 論 文

時間付き離散事象システムの言語安定性の研究 Language Stability of Timed Discrete Event Systems

指導教官 平石 邦彦 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

盧 吉錫

2000 年 2 月 15 日

目次

1	はじめに	1
2	準備	4
2.1	システムの記述	4
2.2	制御機構	8
3	離散事象システムの安定性	10
3.1	従来の離散事象システムに対する安定性	10
3.1.1	Brave-Hyemann の安定性	10
3.1.2	Ozveren-Willsky-Antsaklis の安定性	11
3.1.3	Kumar-Garg-Marcus の安定性	11
3.2	離散事象システムに対する新しい安定性の概念の提案	11
3.2.1	準備	12
3.2.2	離散事象システムの新しい安定性	12
3.2.3	従来の離散事象システムとの比較	13
4	時間付き離散事象システムの安定性	14
4.1	準備	14
4.2	従来の時間付き離散事象システムに対する安定性	14
4.2.1	望山-平石の安定性と安定度	14
4.3	時間付き離散事象システムに対する新しい安定性の概念の提案	15

4.3.1	TDES の安定性と安定度	15
4.4	安定性の解析	16
4.4.1	循環的な事象列	16
4.4.2	状態集合 $u(q)$	16
4.4.3	TDES の安定性定理	17
4.5	安定性判別アルゴリズム	17
4.5.1	目標事象までの最長時間	18
5	安定化スーパーバイザ	22
5.1	可安定性	22
5.1.1	定義	22
5.1.2	TDES の可安定性定理	22
5.2	最適安定化スーパーバイザ	24
5.2.1	最適安定化スーパーバイザ	24
5.2.2	言語 K からの経過時間	24
5.2.3	スーパーバイザ V_* のもとでの目標事象までの最長時間	25
5.2.4	最適安定化スーパーバイザ	25
5.3	最適安定化スーパーバイザの計算アルゴリズム	26
5.4	数値例	27
6	まとめ	32

第 1 章

はじめに

事象が非同期的・離散的に生起することによって状態が変化するシステムは，離散事象システム (discrete event system, DES) と呼ばれる．コンピュータネットワーク，Flexible Manufacturing System (FMS)，オペレーティングシステム，通信システム，データベースシステム，シーケンス制御などがその典型的な例である．これらのシステムを効果的に制御したいという要求から，様々な角度から離散事象システムに対する研究がなされてきた．特に，これらのシステムに対するモデリングや解析に関しては多くの研究がなされてきた．その一方で，その制御に関しては，排他制御や並行処理制御などの個々の問題に対しては多くの手法が提案されてきたが，統一的な方法論は少なかった．しかしながら，コンピュータ応用技術の発展にともない，大規模な離散事象システムが現われ，また，その応用技術も複雑になってきた．そのため，離散事象システムに対するシステム理論的な研究が急速に注目されるようになった．

Ramadge と Wonham は離散事象システムの振る舞いと制御目的を形式言語を用いて記述することにより，離散事象システムに対するシステム理論を構築しようと試み，スーパーバイザ制御を提案した．Ramadge と Wonham によって提唱された離散事象システム解析の枠組みは，それまでの場当たりの解析手法の流れを変え，離散事象システムに対する制御理論的なアプローチを与えた．この枠組みは，現在でも離散事象システムの理論的解析手段の起点となっている．

本研究の目的は，Ramadge-Wonham 流の枠組みのもとで，特に離散事象システムの安

定性に焦点を当て、効果的な安定性の概念を模索し、それを達成する手法を提示することにある。

離散事象システムに対するこれまでの安定性の概念は、大きく2つに分けることができる。1つは状態集合に関する安定性であり、もう1つは事象列に関するものである。

状態集合に関する安定性とは、あらかじめ定められた望ましい状態 (legal states) の集合に注目して安定性を定義するものである。Brave, Heymann らは、離散事象システムが安定であることを、任意の状態から有限回の状態遷移で望ましい状態に到達し、以後、その望ましい状態に留まり続けることと定義した [2]。他方、Ozveren, Willsky, Antsaklis らは、システムの状態が無限の頻度で望ましい状態に訪れるとき、離散事象システムは安定であると定義した [3]。これは、Brave, Heymann らのそれよりも、緩い安定性の定義である。

これらの状態集合に関する安定性に対して、事象列に関する安定性とは、あらかじめ定められた望ましい事象列 (legal behavior) に注目して安定性を定義するものである。Kumar, Garg, Marcus らは、システムが最終的に、与えられた望ましい事象列を生起し続けるとき、離散事象システムは安定であると定義した [4]。

以上に挙げた安定性は、事象の生起とともに状態遷移が起こる離散事象システムに対して議論されている。しかし実際には、事象の生起とともに状態遷移が起こるとは限らず、時間遅れやデッドラインなどが存在する場合が多い。この問題を解決するために、近年、事象の生起に対する時間的な制約を考慮に入れた時間付き離散事象システム (Timed Discrete Event Systems, TDES) と呼ばれるモデルが提案された。このモデルに関して、Ozveren らによる安定性を TDES に拡張した研究が望山らによって行われている [6] が、状態集合から安定性を定義しているので、その記述能力は十分とは言えない。

本稿では、新しく離散事象システムの安定性を提案し、それを基礎として、時間付き離散事象システムにおける新しい安定性の概念を提案する。時間付き離散事象システムにおいては、事象列の時間というものをを用いて、それを安定性の尺度としている。本稿で提案する安定性の特徴は、時間付き離散事象システムに対し、言語を用いた安定性の概念を与

えている点である．また，提案する安定性を判別するためのアルゴリズムを示し，さらには，制約最小の意味で最適なスーパーバイザとそれを用いたオンライン制御アルゴリズムを与える．最後に，状態からの安定性と言語からの安定性の記述能力について考察する．

第 2 章では，離散事象システムと時間付き離散事象システムを定義する際に必要になる，基本的なシステムの記述や制御機構を与える．第 3 章では，離散事象システムに対する従来の安定性について考察し，新しい離散事象システムの安定性の概念を提案する．第 4 章では，時間付き離散事象システムに対する従来の安定性について考察し，時間付き離散事象システムの新しい安定性の概念を提案する．第 5 章では，時間付き離散事象システムの可安定の定義について述べ，システムが可安定であるための条件を示す．最後に，第 6 章では，本研究から得られた結果について考察し，今後の課題についてまとめる．

第 2 章

準備

本章では，制御対象である時間付き離散事象システムと，その制御機構について述べる．

2.1 システムの記述

(時間なし) 離散事象システム (DES) は，つぎのオートマトンで表現される．

$$G_{act} = (\Sigma_{act}, A, \delta_{act}) \quad (2.1)$$

ここで， A はアクティビティと呼ばれる状態の有限集合， Σ_{act} は事象の有限集合，部分関数 $\delta_{act} : \Sigma_{act} \times A \rightarrow A$ は状態遷移関数である．状態遷移関数 δ_{act} は部分関数であり，全ての $\sigma \in \Sigma_{act}$ と $a \in A$ に対して $\delta_{act}(\sigma, a)$ が定義されているとは限らない．定義されているときは， $\delta_{act}(\sigma, a)!$ と書くことにする．もし， $\delta_{act}(\sigma, a)!$ であるとき， σ は生起可能であるという．

各事象 $\sigma \in \Sigma_{act}$ には，時間の下限 $l_\sigma \in N$ と時間の上限 $u_\sigma \in N \cup \{\infty\}$ が与えられている．ただし， N は非負整数の集合であるとする．3 項組 $(\sigma, l_\sigma, u_\sigma)$ を時間事象と呼ぶ．時間の下限 $l_\sigma \in N$ は遅延を表わし，時間の上限 $u_\sigma \in N \cup \{\infty\}$ はデッドラインを表わす．

Σ_{act} を以下の 2 つの部分集合に分ける．

$$\Sigma_{act} = \Sigma_{spe} \cup \Sigma_{rem} \quad (2.2)$$

Σ_{spe} をプロスペクティブ (prospective) な事象と呼び， Σ_{rem} をリモート (remote) な事象

と呼ぶ．プロスペクティブな事象とは，時間の上限 u_σ が有限 ($0 \leq u_\sigma < \infty$) な事象のことであり，リモートな事象とは，時間の上限が無限 ($u_\sigma = \infty$) な事象のことである．

各事象 σ に対して時間間隔 T_σ を次のように定義する．

$$T_\sigma = \begin{cases} [0, u_\sigma] & \text{if } \sigma \in \Sigma_{spe} \\ [0, l_\sigma] & \text{if } \sigma \in \Sigma_{rem} \end{cases} \quad (2.3)$$

ここで， $[j, k]$ は j 以上 k 以下である整数の集合を表わす．

状態集合 Q を，以下のように記述する．

$$Q = A \times \prod \{T_\sigma \mid \sigma \in \Sigma_{act}\} \quad (2.4)$$

ただし， $\prod \{S_1, \dots, S_i\}$, $i \in N$ は集合要素の直積 $S_1 \times \dots \times S_i$ を意味する．また，状態集合 Q の要素である q を，以下のように記述する．

$$q = (a, \{t_\sigma \mid \sigma \in \Sigma_{act}\}) \quad (2.5)$$

時間に関する上限，下限を持つ事象によって駆動する DES G_{act} を用いて，時間付き離散事象システム (TDES) G を

$$G = (Q, \Sigma, \delta) \quad (2.6)$$

により定義する．ここで， Q は状態集合を表わし，また， Σ は事象の有限集合で，

$$\Sigma = \Sigma_{act} \cup \{tick\} \quad (2.7)$$

である．ただし， $tick$ は単位時間の経過を表わす特別な事象である．

状態遷移関数 $\delta : \Sigma \times Q \rightarrow Q$ は部分関数であり， $\delta(\sigma, q)$ のように記述され，もし状態 q において σ が生起可能であるならば $\delta(\sigma, q)!$ と書くことにする．

状態遷移関数 δ の定義を以下に与える．この中で， $\delta(\sigma, q) = q'$ に対し，

$$q = (a, \{t_\tau \mid \tau \in \Sigma_{act}\}) \quad (2.8)$$

$$q' = (a', \{t'_\tau \mid \tau \in \Sigma_{act}\}) \quad (2.9)$$

であるとする .

状態遷移関数 $\delta(\sigma, q)$ は , 以下の 3 つの条件のいずれかを満たすときに限り定義される .

- $(\sigma = tick) \wedge (\forall \tau \in \Sigma_{spe}(t_\tau > 0))$
- $(\sigma \in \Sigma_{spe}) \wedge (\delta_{act}(\sigma, a)!) \wedge (0 \leq t_\sigma \leq u_\sigma - l_\sigma)$
- $(\sigma \in \Sigma_{rem}) \wedge (\delta_{act}(\sigma, a)!) \wedge (t_\sigma = 0)$

さらに , 状態遷移後の状態 $q' = \delta(\sigma, q)$ を以下のように定義する .

- $\sigma = tick$ のとき

$a' := a$ かつ ,

– もし , $\tau \in \Sigma_{spe}$ ならば ,

$$t'_\tau := \begin{cases} t_\tau - 1 & \text{if } \sigma_{act}(\tau, a)! \wedge t_\tau > 0 \\ u_\tau & \text{if not } \sigma_{act}(\tau, a)! \end{cases} \quad (2.10)$$

– もし , $\tau \in \Sigma_{rem}$ ならば ,

$$t'_\tau := \begin{cases} t_\tau - 1 & \text{if } (\delta_{act}(\tau, a)!) \wedge (t_\tau > 0) \\ 0 & \text{if } (\delta_{act}(\tau, a)!) \wedge (t_\tau = 0) \\ l_\tau & \text{if not } \delta_{act}(\tau, a)! \end{cases} \quad (2.11)$$

- $\sigma \in \Sigma_{act}$ のとき

$a' := \delta_{act}(\sigma, a)$ かつ ,

– もし $\tau \neq \sigma$ かつ $\tau \in \Sigma_{spe}$ ならば ,

$$t'_\tau := \begin{cases} t_\tau & \text{if } \delta_{act}(\tau, a')! \\ u_\tau & \text{if not } \delta_{act}(\tau, a')! \end{cases} \quad (2.12)$$

– もし $\tau = \sigma$ かつ $\tau \in \Sigma_{spe}$ ならば ,

$$t'_\tau := u_\sigma \quad (2.13)$$

– もし $\tau \neq \sigma$ かつ $\tau \in \Sigma_{rem}$ ならば ,

$$t'_\tau := \begin{cases} t_\tau & \text{if } \delta_{act}(\tau, a')! \\ l_\tau & \text{if not } \delta_{act}(\tau, a')! \end{cases} \quad (2.14)$$

– もし $\tau = \sigma$ かつ $\tau \in \Sigma_{rem}$ ならば ,

$$t'_\tau := l_\sigma \quad (2.15)$$

$q_0 \in Q$ は初期状態であり , 次式で定義する .

$$q_0 = (a_0, \{t_{\sigma_0} \mid \sigma \in \Sigma_{act}\}) \quad (2.16)$$

ただし , t_{σ_0} は既定値と呼ばれ ,

$$t_{\sigma_0} := \begin{cases} u_\sigma & \text{if } \sigma \in \Sigma_{spe} \\ l_\sigma & \text{if } \sigma \in \Sigma_{rem} \end{cases} \quad (2.17)$$

である .

状態遷移関数 δ の定義域をつぎのように $\Sigma^* \times Q$ に拡張する .

1. $\forall q \in Q : \delta(\varepsilon, q) := q;$
2. $\delta(s\sigma, q) := \delta(\sigma, \delta(s, q)) .$

単位時間の間に無限回の状態遷移が起こることを避けるために , つぎの仮定を置く .

仮定 Activity-Loop-Free 条件

$$(\forall q \in Q)(\forall s \in \Sigma_{act}^* - \{\epsilon\})\delta(s, q) \neq q \quad (2.18)$$

ただし, ϵ は空列 (長さゼロの系列) を表す.

また, 任意の状態において少なくとも 1 つは発生可能な事象が存在する. これは時間の経過を表わす *tick* の発生は抑制できないからである.

言語 L の接頭語の閉包 (prefix closure) を $\bar{L}$ を以下のように定義する.

$$\bar{L} := \{s \in \Sigma^* \mid \exists t \in \Sigma^*(st \in L)\} \quad (2.19)$$

また, 言語 L の接尾語の閉包 (suffix closure) を $\underline{L}$ を以下のように定義する.

$$\underline{L} := \{t \in \Sigma^* \mid \exists s \in \Sigma^*(st \in L)\} \quad (2.20)$$

2.2 制御機構

ここでは, 本論文で用いる時間付離散事象システムに対する制御機構の説明を行う.

事象がその生起を抑制できるとき, プロヒビティブル (prohibitible) である呼び, その集合を Σ_{hib} と表わす. プロヒビティブルな事象はリモートである. すなわち, $\Sigma_{hib} \subseteq \Sigma_{rem}$ である.

次に, 不可制御な事象の集合 Σ_{unc} を次式で定義する.

$$\begin{aligned} \Sigma_{unc} &:= (\Sigma_{act} - \Sigma_{hib}) \cup \{tick\} \\ &= \Sigma_{spe} \cup (\Sigma_{rem} - \Sigma_{hib}) \cup \{tick\} \end{aligned} \quad (2.21)$$

また, その補集合として可制御な事象の集合 Σ_{con} を次式で定義する.

$$\begin{aligned} \Sigma_{con} &:= \Sigma - \Sigma_{unc} \\ &= \Sigma_{hib} \end{aligned} \quad (2.22)$$

TDES G における状態 $q \in Q$ において生起可能な事象の集合を q におけるエリジブル (eligible) な事象の集合と呼び, $Elig_G(q)$ と表わし, 次式で定義する.

$$Elig_G(q) := \{\sigma \in \Sigma \mid \delta(\sigma, q)!\} \quad (2.23)$$

初期状態が $q \in Q$ ($a \in A$) であるような TDES G ($\text{DES } G_{act}$) を, $\langle G, q \rangle$ ($\langle G_{act}, a \rangle$) と記述する .

TDES $\langle G, q \rangle$ のふるまいは以下に定義する接頭語について閉じた言語 $L(\langle G, q \rangle)$ で記述される .

$$L(\langle G, q \rangle) := \{s \in \Sigma^* \mid \delta(s, q)!\} \quad (2.24)$$

任意の事象列 $s \in L(\langle G, q \rangle)$ に対して以下の条件を満足する写像 $V : L(\langle G, q \rangle) \rightarrow \Gamma$ をスーパーバイザという .

$$V(s) \cap \text{Elig}_G(\delta(s, q)) \neq \emptyset \quad (2.25)$$

$$V(s) \supseteq \Sigma_{unc} \quad (2.26)$$

ただし Γ は, $\Gamma := \{\gamma \mid \Sigma_{unc} \subseteq \gamma \subseteq \Sigma\}$ であり, 制御パターンの集合という .

スーパーバイザは今までに発生した事象列を基につぎに発生を許可する事象の集合を指示するものである . スーパーバイザにより制御されたシステムを閉ループシステムとよび, $V/\langle G, q \rangle$ と表わす .

$V/\langle G, q \rangle$ により生成される言語 $L(V/\langle G, q \rangle)$ は, つぎのように再帰的に定義される .

1. $\epsilon \in L(V/\langle G, q \rangle)$;
2. もし $s \in L(V/\langle G, q \rangle)$, $\sigma \in V(s)$, $s\sigma \in L(\langle G, q \rangle)$ ならば, $s\sigma \in L(V/\langle G, q \rangle)$;
3. それ以外の列は $L(V/\langle G, q \rangle)$ には含まれない .

第 3 章

離散事象システムの安定性

本章では、時間を考慮しない離散事象システムに対して提案されたいくつかの安定性について整理する。その後、新しい安定性の概念を提案し、従来の安定性と新しい安定性との比較を行う。

3.1 従来の離散事象システムに対する安定性

3.1.1 Brave-Hyemann の安定性

DES $G_{act} = (A, \Sigma_{act}, \delta_{act})$ は、以下の条件が成り立つとき、 $(A_0, \hat{A})$ に関して安定である。

- すべての $\hat{a} \in \hat{A}$, $s \in L(\langle G_{act}, \hat{a} \rangle)$ に関して $\delta_{act}(s, \hat{a}) \in \hat{A}$ が成り立つ。
- $|s| \geq |A - \hat{A}|$ であるような、すべての $a_0 \in A_0$, $s \in L(\langle G_{act}, \hat{a} \rangle)$ に関して、 $\delta_{act}(s, a_0) \in \hat{A}$, $|s'| \leq |A - \hat{A}|$ であるような $s' \in \overline{\{s\}}$ が存在する。

ここで、 A_0 , $\hat{A}$ はそれぞれ、初期状態の集合、望ましい状態の集合を表わす。また、 $|\cdot|$ は系列ならその長さ、集合ならばその要素数を意味する。

定義の前半部分では $\hat{A}$ に留まり続ける性質を述べており、後半部分では A_0 から $\hat{A}$ への可到達性を述べている。すなわち、この安定性の定義では A_0 から出発した状態の軌跡が $\hat{A}$ へ到達し、以後そこに留まり続けることを要求している。

3.1.2 Ozveren-Willsky-Antsaklis の安定性

DES $G_{act} = (A, \Sigma_{act}, \delta_{act})$ は、以下の条件が成り立つとき、 $(A_0, \hat{A})$ に関して安定である。

$|t| \geq |A - \hat{A}|$ であるような、すべての $a_0 \in A_0$ 、 $s, st \in L(\langle G_{act}, a_0 \rangle)$ に対して、
 $\delta_{act}(st', a_0) \in \hat{A}$ であるような $t' \in \overline{\{t\}}$ が存在する。

この安定性の定義では、状態の軌跡は $\hat{A}$ 内に留まり続けている必要はなく、有限回の状態遷移で $\hat{A}$ に訪れることが保証されていればよい。

3.1.3 Kumar-Garg-Marcus の安定性

DES $G_{act} = (A, \Sigma_{act}, \delta_{act})$ に対し、初期状態の集合 A_0 および接頭語について閉じた Σ_{act} 上の言語 K を与える。任意の $a_0 \in A_0$ に対し以下の条件が成り立つとき、 G_{act} は (A_0, K) に関して安定である。

$L(\langle G_{act}, a_0 \rangle) \subseteq \Sigma_{act}^{\leq n} K$ であるような $n \in N$ が存在する。

ここで、 Σ_{act}^n は長さが $n \leq N$ であるような Σ_{act} 上の列の集合であり、 $\Sigma_{act}^{\leq n'}$ は $\bigcup_{n \leq n'} \Sigma_{act}^n$ ($n' \in N$) である。

この安定性の定義では、状態の軌跡が最終的に言語 K に一致する必要がある。

3.2 離散事象システムに対する新しい安定性の概念の提案

ここでは、時間付き離散事象システムにおける安定性の概念を提案する前の準備として、離散事象システムにおける安定性の概念を提案する。ここで提案する離散事象システムにおける安定性は、このあとに提案する時間付き離散事象システムにおける安定性の基礎となるものである。

3.2.1 準備

K を接頭語について閉じた Σ_{act} 上の言語とする．言語 K および事象列 $s \in K$ に対して，事象の集合 $next(s, K)$ を以下のように定義する．

$$next(s, K) = \{\sigma \in \Sigma \mid s \cdot \sigma \in K\} \quad (3.1)$$

この関数は，ある事象列 s を生起した後に，次にどの事象を生起すれば言語 K の要素になるのかということを表わすものである．

言語 K および事象列 $s \in K$ に対して， $s//K \in K$ を以下のように定義する．

$$\lambda//K = \lambda \quad (3.2)$$

$$(s' \cdot \sigma)//K = \begin{cases} (s'//K) \cdot \sigma & (\sigma \in next(s'//K, K)) \\ s'//K & (otherwise) \end{cases} \quad (3.3)$$

この関数は，ある事象列 s からそれに含まれる言語 K の要素を取り出すためのものである．例えば， $s = abcdefg$, $K = \{a, af, afb\}$ に対して $//$ を適用した結果は，

$$s//K = af \quad (3.4)$$

となる．

3.2.2 離散事象システムの新しい安定性

DES $G_{act} = (A, \Sigma_{act}, \delta_{act})$ に対し初期状態の集合 A_0 および接頭語について閉じた Σ 上の言語 K を与える． $a_0 \in A_0$ について以下の条件が成り立つとき， G_{act} は (A_0, K) に関して安定である．

任意の $s \in L(\langle G_{act}, a_0 \rangle)$ に対して，その後に有限長の事象列 $w \in \Sigma^*$ を生起し

たのちに，必ず，ある $\sigma \in next(s//K, K)$ を生起する．

$$\begin{aligned} & (\exists n \in \mathbb{N})(\forall s_1, s_2 := s_1 w \in L(\langle G_{act}, a_0 \rangle))(\exists \sigma \in next(s_1//K, K)) \\ & (|w| \geq n \Rightarrow (\exists w' \cdot \sigma \in \overline{\{w\}} - \{\epsilon\}) s_1 \cdot w' \cdot \sigma \in L(\langle G_{act}, a_0 \rangle)) \end{aligned} \quad (3.5)$$

この安定性は，システムの動作が与えられた言語 K の周辺に存在しつづけることを表わす．

3.2.3 従来の離散事象システムとの比較

Brave-Hyemann の安定性や Ozveren-Willsky-Antsaklis の安定性のように，安定性を判断するための要素としてシステムにおける状態を用いたものでは，安定であるためには常にある決められた状態に遷移しなくてはならず，動的なシステムを考えたとき，その記述能力が不足する．また，Kumar-Garg-Marcus の安定性においては，安定であるためには最終的にある決められた事象列を遷移し続けなくてはならず，これも安定性を表現する上での記述能力に乏しいと言える．

今回提案する安定性においては，安定性を判断するものとして言語を用いることにより，従来の安定性と比べて安定性を表現する上での記述能力に大きな違いがある．事象列内に望ましい言語の列が部分列として含まれてさえいれば良い．

第 4 章

時間付き離散事象システムの安定性

ここでは、すでに提案した離散事象システムの安定性を基礎として、時間付き離散事象システムの安定性を提案する。そのために、まず準備として、ここで使われる用語の定義を行い、従来研究における時間付き離散事象システムの安定性を概観し、そのあと、新しい時間付き離散事象システムの安定性を定義する。そして、それが終わったら、その新しい安定性を実現するアルゴリズムを提案し、そのアルゴリズムの計算量について考察する。

4.1 準備

つぎの記法を定義する。

- 事象列 s に含まれる事象 $tick$ の数を s の時間といい、 $\|s\|_{tick}$ で表す。
- 事象列 w に含まれる全ての事象の集合を $\Sigma(w)$ と表す。

4.2 従来の時間付き離散事象システムに対する安定性

4.2.1 望山-平石の安定性と安定度

TDES G に対し、初期状態の集合 Q_0 および望ましい状態の集合 $\hat{Q}$ を与える。以下を満たすとき、TDES G は $(Q_0, \hat{Q})$ に関して安定である。

ある整数 $\hat{k} \in N$ が存在して、すべての $q_0 \in Q_0$ および $\|t\|_{tick} \geq \hat{k}$ を満足するすべての $s_1, s_2 := s_1 t \in L(\langle G, q_0 \rangle)$ に対して、 t の接頭語 t' が存在して $\delta(s_1 t', q_0) \in \hat{Q}$ を満足するとき、すなわち、

$$\begin{aligned} & (\exists \hat{k} \in N)(\forall q_0 \in Q_0)(\forall s_1, s_2 := s_1 t \in L(\langle G, q_0 \rangle)) \\ & (\|t\|_{tick} \geq \hat{k} \Rightarrow (\exists t' \in \overline{\{t\}} - \{\epsilon\}) \delta(s_1 t', q_0) \in \hat{Q}) \end{aligned} \quad (4.1)$$

条件を満たす最小の $\hat{k}$ を安定度と呼ぶ。

この時間付き離散事象システムの安定性は、Ozveren らによる離散事象システムの安定性 [3] を基礎とし、時間付きの場合に拡張したものである。安定性の記述能力を高めるためには、言語を指標とする方法が考えられる。この点を考慮に入れた新しい安定性の概念を次に提案する。

4.3 時間付き離散事象システムに対する新しい安定性の概念の提案

4.3.1 TDES の安定性と安定度

TDES G に対し、初期状態の集合 Q_0 および接頭語について閉じた Σ 上の言語 K を与える。以下を満たすとき、TDES G は (Q_0, K) に関して安定である。

ある整数 $\hat{k} \in N$ が存在し、 $\|w\|_{tick} \geq \hat{k}$ を満足するすべての $q_0 \in Q_0$ および $s_1, s_2 := s_1 w \in L(\langle G, q_0 \rangle)$ に対して、 $\sigma \in next(s//K, K)$ 、 w の接頭語 $w' \cdot \sigma$ が存在して、 $s_1 \cdot w' \cdot \sigma \in L(\langle G, q_0 \rangle)$ を満足する。すなわち、

$$\begin{aligned} & (\exists \hat{k} \in N)(\forall q_0 \in Q_0)(\forall s_1, s_2 := s_1 w \in L(\langle G, q_0 \rangle)(\exists \sigma \in next(s//K, K)) \\ & (\|w\|_{tick} \geq \hat{k} \Rightarrow (\exists w' \cdot \sigma \in \overline{\{w\}} - \{\epsilon\}) s_1 \cdot w' \cdot \sigma \in L(\langle G, q_0 \rangle)) \end{aligned} \quad (4.2)$$

条件を満たす最小の $\hat{k}$ を安定度と呼ぶ。

4.4 安定性の解析

安定性を判別するための条件およびそれを判定するアルゴリズムについて述べる．ここでは，言語 K は正規言語であると仮定する．なお，議論を簡単にするために，初期状態の集合はただ 1 つの要素をもつ ($Q_0 = \{q_0\}$) と仮定する．また，すべての状態は初期状態から到達可能であるとする．

4.4.1 循環的な事象列

状態 q ，事象列 s に対して $\delta(s, q) = \delta(s', q)$ となるような s の真の接頭語 s' が存在するとき，事象列 s は循環的である．すなわち，ある状態 q から出発する $s \in L(G)$ が循環的であるとは，

$$(\exists s' \in \overline{\{s\}} - \{s\}) \quad \delta(s, q) = \delta(s', q) \quad (4.3)$$

が成り立つことである．また， $\delta(s, q) = \delta(s', q) = q'$ となるような状態 q' を，循環点， $s = s'w$ としたときの部分列 w を循環部分列と呼ぶ．

4.4.2 状態集合 $u(q)$

言語 K は正規言語なので，決定性有限オートマトンによって生成される．この決定性有限オートマトン (DFA) を $M = (\Sigma, X, \xi, x_0)$ とする．

TDES $\langle G, q_0 \rangle = (\Sigma, Q, \delta, q_0)$ において事象の生起により状態が遷移するとき，同時に DFA M においても同時に状態を遷移させることを考える．

$K \subseteq L(\langle G, q_0 \rangle)$ なので，一般には $\langle G, q_0 \rangle$ で状態が遷移可能であっても M で遷移できない場合がある．このときは， $\langle G, q_0 \rangle$ の状態のみを遷移させ， M の状態は変えないようにする．このような状態遷移を考えたとき， $\langle G, q_0 \rangle$ において状態が $q \in Q$ にあるとき，DFA M における状態として取りうる状態の集合を $u(q)$ と定義する．

$x \in u(q)$ のとき，ある $s \in L(\langle G, q_0 \rangle)$ に対し， $\delta(s, q_0) = q$ かつ $next(s//K, K) = Elig_M(x)$ である．

4.4.3 TDES の安定性定理

TDES G が言語 K に関して安定であるための必要十分条件は以下のように得られる .

初期状態 $q_0 \in Q_0$ から生起する任意の循環的な事象列 s について , その循環点を q , 循環部分列を w とする . このとき , $u(q)$ に含まれる各 x に対し $Elig_M(x)$ の少なくとも 1 つの事象が w に含まれる . すなわち ,

$$\begin{aligned} & (\forall q_0 \in Q_0)(\forall s_1, s_2 := s_1 w \in L(\langle G, q_0 \rangle)) \\ & (\delta(s_1, q_0) = \delta(s_2, q_0) = q \Rightarrow (\forall x \in u(q))(\exists \sigma \in Elig_M(x)) \sigma \in \Sigma(w)) \end{aligned} \quad (4.4)$$

「証明」

(必要性) TDES G が Q_0 および言語 K に関して安定であるとき , 初期状態 Q_0 の要素から生起する循環的な事象列の中で , 循環点 q における $u(q)$ のある状態 x に対する $Elig_M(x)$ の事象を生起しないものが存在すると仮定する . ここで , ある $s \in L(\langle G, q_0 \rangle)$ に対し , $\delta(s, q_0) = q$ かつ $next(s//K, K) = Elig_M(x)$ である . 状態 q から $next(s//K, K)$ の事象を生起することなしに w を任意回繰り返すことができる . このとき , Activity-Loop-Free 条件より 1 回の繰り返しごとに時間は増加する . これは , G が安定であることに矛盾する .

(十分性) いま , $\hat{k} = |Q|$ とおく . そして , $s_2 = s_1 w$ で , $\|w\|_{tick} \geq \hat{k}$ となる任意の $s_1, s_2 \in L(\langle G, q_0 \rangle)$ を考える . ここで , $|w| \geq \|w\|_{tick} \geq \hat{k} = |Q|$ であり , $|Q| < \infty$ であるから , $\delta(s_1 w_1, q_0) = \delta(s_1 w_2, q_0)$ となるような w の接頭語 w_1, w_2 (ただし , $|w_1| < |w_2|$) が存在する . 循環的ならば , 必ず少なくとも 1 つの $\sigma \in next(s_1//K, K)$ を生起するので , $s_1 \cdot w' \cdot \sigma \in L(\langle G, q_0 \rangle)$ かつ $|w_1| < |w'| < |w_2|$ であるような w' が存在する . これは , TDES G が安定度 $\hat{k}$ 以下で言語 K に関して安定であることを意味する .

4.5 安定性判別アルゴリズム

ここでは前節での結果を用いて , TDES G の安定性を判別し , さらに , 安定であるならばシステムの安定度を求めるアルゴリズムを示す .

前節の定理から，安定性を判別するためには，初期状態から出発する全ての事象列に対して，もし事象列が循環事象列であるならば，その循環点 q における $u(q)$ の要素の生起可能事象が循環部分列の中に含まれている否かを調べればよいことが解る．

よって，まず $\langle G, q_0 \rangle$ の各状態 q に対し $u(q)$ を計算する．つぎに，各状態 q を根として深さ優先探索を行い，各 $x \in u(q)$ に対する $Elig_M(x)$ の事象が出現するかを調べる．もし，それらが出現しないような循環事象列が発見されるならば安定ではない．また，安定度は，以下に定義する目標事象までの最長時間を計ることにより計算することができる．

4.5.1 目標事象までの最長時間

TDES $\langle G, q_0 \rangle$ において状態 q から全ての $x \in u(q)$ に対し少なくとも1つの $\sigma \in Elig_M(x)$ を生起するまでにかかる最長の時間を $T_{rest}(q)$ とする．すなわち，

$$T_{rest}(q) = \max_{w \in L'(q)} \|w\|_{tick} \quad (4.5)$$

ここで，

$$L'(q) = \{w \in L(\langle G, q \rangle) \mid (\exists x \in u(q))(\forall \sigma \in Elig_M(x))(\sigma \notin \Sigma(w))\} \quad (4.6)$$

である．

この T_{rest} を用いれば，安定度 $\hat{k}$ は以下のように表わすことができる．

$$\hat{k} = \max_{q \in Q} T_{rest}(q) \quad (4.7)$$

安定性判別アルゴリズムの流れは，以下のようになる．

「安定性判別アルゴリズム」

1. TDES $\langle G, q_0 \rangle$ の各状態 $q \in Q$ に対する $u(q)$ を計算する．
2. TDES $\langle G, q_0 \rangle$ の各状態 q から深さ優先探索を行い， $u(q)$ の各要素 x に対し $Elig_G(x)$ の事象が生起するかどうかを調べる．このとき，同時に $T_{rest}(q)$ の値も記憶する．

3. もし途中で, $Elig_G(x)$ の事象が生起しないまま循環事象列であることが判明した場合, ”不安定” と表示し, 終了する.
4. ”不安定” となることなく, すべての状態においてチェックが完了した場合, そのシステムは安定であるとし, 安定度 $\hat{k}$ を計算して出力する.

```

Input:
TDES  $G (Q, \Sigma, \delta)$ , TDES  $G$  における初期状態集合  $Q_0$ 
DFA  $M (X, \Sigma, \xi)$ , DFA  $M$  における初期状態  $x_0$ 
Output:
 $u(q_i) (q_i \in Q)$ 
Main:
begin
  for  $\forall q_i \in Q$  do
     $u(q_i) = \phi$ ;
  end-for;
  check_1( $q_0, x_0$ );
   $u(q_i) (q_i \in Q)$  を出力;
end.
procedure check_1( $q_i, x_k$ )
begin
   $u(q_i) \leftarrow u(q_i) \cup \{x_k\}$ ;
  for  $\forall \sigma_j \in Elig_G(q_i)$  do
     $q_l = \delta(\sigma_j, q_i)$ ;
    if  $\exists \sigma_j \in Elig_M(x_k)$  then  $x_n = \xi(\sigma_j, x_k)$ 
      if  $x_n \notin u(q_l)$  then check_1( $q_l, x_n$ ) end-if;
    else
      if  $x_k \notin u(q_l)$  then check_1( $q_l, x_k$ ) end-if;
    end-if;
  end-for;
end.

```

図 4.1: $u(q)$ を計算するアルゴリズム

```

Input:
TDES  $G (Q, \Sigma, \delta)$ , TDES  $G$  における初期状態集合  $Q_0$ 
DFA  $M (X, \Sigma, \xi)$ , DFA  $M$  における初期状態  $x_0$ 
Output:
システムが安定ならば安定度  $\hat{k}$  を出力し , そうでなければ “不安定” と表示する .
Main:
begin
   $u(q_i)$  ( $q_i \in Q$ ) を計算するアルゴリズムを実行する;
  for  $\forall q_i \in Q$  do
     $q_i$  を探索済みでないとする;
     $t_{rest}(q_i) \leftarrow 0$ ;
  end-for;
  for  $\forall q_i \in Q$  do
     $w \leftarrow \epsilon$ ;
    check_2( $q_i, q_i$ );
  end-for;
  安定度  $\hat{k}$  として ,  $\max_{q_i \in Q} t_{rest}(q_i)$  を出力する;
end.

```

図 4.2: 安定性判別アルゴリズム (Main)

```

procedure check_2( $q_i, q_l$ )
begin
   $q_l$  を探索済みとする;
  for  $\forall x_k \in u(q_i)$  do
     $Seiki(x_k) = 0$ ;
  end-for;
  for  $\forall \sigma_j \in Elig_G(q_l)$  do
    if  $(\exists x_k \in u(q_i)) \sigma_j \in Elig_M(x_k)$  かつ、 $Seiki(x_k) = 0$ 、かつ、
     $\|w\|_{tick} > t_{rest}(q_i)$  then
       $t_{rest}(q_i) \leftarrow \|w\|_{tick}$ ;
       $Seiki(x_k) = 1$ ;
    end-if;
     $w \leftarrow w\sigma_j$ ;
     $q_r := \delta(\sigma_j, q_l)$ ;
    if  $q_r$  が未探索である then check_2( $q_i, q_r$ );
    else  $w'$  を  $w$  の循環部分とする
      if  $(\exists x_k \in u(q_i))(\forall \sigma_j \in Elig_M(x_k)) \sigma_j \notin \Sigma(w')$  then
        “不安定” と出力し、強制終了する;
      end-if;
    end-if;
  end-for;
end.

```

図 4.3: 安定性判別アルゴリズム (check_2)

第 5 章

安定化スーパーバイザ

本章では，TDES をスーパーバイザにより安定化するための条件について考察する．

5.1 可安定性

5.1.1 定義

任意の $q_0 \in Q_0$ に対し，あるスーパーバイザ V_{q_0} が存在して，閉ループシステム $V_{q_0}/\langle G, q_0 \rangle$ が言語 K に関して安定となるとき，すなわち，

$$\begin{aligned} & (\exists \hat{k} \in N)(\forall q_0 \in Q_0)(\exists V_{q_0} : L(\langle G, q_0 \rangle) \rightarrow \Gamma) \\ & (\forall s_1, s_2 := s_1 w \in L(V/\langle G, q \rangle))(\exists \sigma \in next(s_1//K, K)) \\ & (\|w\|_{tick} \geq \hat{k} \Rightarrow (\exists w' \cdot \sigma \in \overline{\{w\}} - \{\epsilon\}) s_1 \cdot w' \cdot \sigma \in L(G)) \end{aligned} \tag{5.1}$$

が成り立つとき，TDES G は (Q_0, K) に関して可安定であるという．

5.1.2 TDES の可安定性定理

TDES G が言語 K に関して可安定であるための必要十分条件は以下のように得られる．

初期状態 $q_0 \in Q_0$ から生起する任意の循環的な事象列 s について，その循環点を q ，循環部分列を w とする．このとき，ある $x \in u(q)$ に対し $Elig_M(x)$ のすべての事象が w に含まれないならば， s は少なくとも 1 つの可制御な事象を含む．すなわち，

$$\begin{aligned}
& (\forall q_0 \in Q_0)(\forall s_1, s_2 := s_1 w \in L(\langle G, q_0 \rangle)) \\
& (\delta(s_1, q_0) = \delta(s_2, q_0) = q \wedge (\exists x \in u(q))(\forall \sigma \in Elig_M(x))\sigma \notin \Sigma(w)) \Rightarrow \Sigma(s_2) \cap \Sigma_{con} \neq \emptyset
\end{aligned} \tag{5.2}$$

「証明」

(必要性) TDES G が言語 K に関して可安定であるとき, (5.2) が成立しないと仮定する. このとき,

$$\begin{aligned}
& (\exists q_0 \in Q_0)(\exists s_1, s_2 := s_1 w \in L(\langle G, q_0 \rangle))(\exists q \in Q) \\
& (\delta(s_1, q_0) = \delta(s_2, q_0) = q \wedge (\exists x \in u(q))(\forall \sigma \in Elig_M(x))\sigma \notin \Sigma(w)) \wedge \Sigma(s_2) \cap \Sigma_{con} = \emptyset
\end{aligned} \tag{5.3}$$

である. $\Sigma(s_2) \cap \Sigma_{con} = \emptyset$ であることから, この s_1, s_2 は任意のスーパーバイザ V による影響を受けず, $L(V/\langle G, q_0 \rangle)$ の要素である. したがって,

$$\begin{aligned}
& (\exists s_1, s_2 := s_1 w \in L(V/\langle G, q_0 \rangle))(\exists q \in Q) \\
& (\delta(s_1, q_0) = \delta(s_2, q_0) = q \wedge (\exists x \in u(q))(\forall \sigma \in Elig_M(x))\sigma \notin \Sigma(w))
\end{aligned} \tag{5.4}$$

が成り立つ. これは TDES $V/\langle G, q_0 \rangle$ が安定でないことを意味しており, 矛盾である.

(十分性) (5.2) が成立すると仮定する. 次のようなスーパーバイザ V_* を考える.

$$V_*(s) = \Sigma - \Sigma_{con} \tag{5.5}$$

つまり, 全ての可制御な事象を抑制するような制御を施すことを考える. このとき, この V_* によって制御された閉ループ系の生成言語 $L(V_*/\langle G, q_0 \rangle)$ は, Σ_{con} の要素を含まないから, 任意の $s_2 \in L(V_*/\langle G, q_0 \rangle)$ に対して,

$$\Sigma(s_2) \cap \Sigma_{con} = \emptyset \tag{5.6}$$

である. よって, (5.2) より,

$$\begin{aligned}
& (\forall s_1, s_2 := s_1 w \in L(V_*/\langle G, q_0 \rangle)) \\
& (\delta(s_1, q_0) = \delta(s_2, q_0) = q \Rightarrow (\forall x \in u(q))(\exists \sigma \in Elig_M(x))\sigma \in \Sigma(w))
\end{aligned} \tag{5.7}$$

であり, V_* は $\langle G, q_0 \rangle$ を安定化する.

5.2 最適安定化スーパーバイザ

可安定性定理の十分性の証明から明らかなように，もしシステムが可安定ならば，最も抑制的で静的なスーパーバイザ V_* によって，システムを安定化することができる．しかし，実際のシステムでは，制御すること，すなわち，事象の生起を抑制することは最後の手段であり，必要以上に事象の生起に抑制をかけることはコスト面でも，本来のシステム的能力を損ないたくないという点からも好ましくない．そこで，指定された安定性の度合いを保ちつつ，抑制を最小限に留めるという意味で最適な安定化スーパーバイザを見つけることが必要となる．

ここでも議論を簡単にするために $Q_0 = \{q_0\}$ を仮定する．

5.2.1 最適安定化スーパーバイザ

TDES G は，言語 K に関して安定度 $\hat{k}$ で可安定であるとする．TDES G を言語 K に関して目標安定度 $\hat{k}^* \geq \hat{k}$ で安定化するスーパーバイザのクラスを $C(\hat{k}^*)$ とする．このとき，任意の $V \in C(\hat{k}^*)$ に対して

$$L(V/G) \subseteq L(V^*/G) \quad (5.8)$$

を満足する安定化スーパーバイザ V^* を最適安定化スーパーバイザとよぶ．

5.2.2 言語 K からの経過時間

事象列 $s \in L(\langle G, q_0 \rangle)$ に対し，言語 K の事象を最後に生起してから現在に至るまでに経過した時間 $T_{off}(s) \in N$ を次式で定義する．

$$T_{off}(s) = \max_{t \in L''(s)} \|t\|_{tick} \quad (5.9)$$

ここで，

$$L''(s) = \{t \in \underline{\{s\}} \mid s = s't \wedge \Sigma(t) \cap next(s'//K, K) = \emptyset\} \quad (5.10)$$

である．

TDES G が言語 K に関して安定度 $\hat{k}$ で安定であるための必要十分条件は , TDES G に
よって生成される任意の事象列 $s \in L(G)$ が言語 K から離れている時間が $\hat{k}$ 以下になる
ことである . すなわち ,

$$(\forall q_0 \in Q_0)(\forall s \in L(\langle G, q_0 \rangle))(T_{off}(s) \leq \hat{k}) \quad (5.11)$$

が成り立つことである .

5.2.3 スーパーバイザ V_* のもとでの目標事象までの最長時間

最も抑制的なスーパーバイザ V_* による閉ループシステムを考える .

事象列 $s \in L(V_*/\langle G, q_0 \rangle)$ に対し , つぎに $next(s//K, K)$ の事象が生起するまでの最長
時間 $T_{*rest}(s) \in N$ を次式で定義する .

$$T_{*rest}(s) = \max_{t \in L'''(s)} \|t\|_{tick} \quad (5.12)$$

ここで ,

$$L'''(s) = \{t \in L(V_*/\langle G, \delta(s, q_0) \rangle) \mid \Sigma(t) \cap next(s//K, K) = \emptyset\} \quad (5.13)$$

5.2.4 最適安定化スーパーバイザ

TDES G が言語 K に関して安定度 $\hat{k}$ で可安定であるとする . ここで , 次の動的なスー
パーバイザ $V^* : L(G) \rightarrow \Gamma$ を考える .

$$V^*(s) = \Sigma - \{\sigma \in Elig_G(\delta(s, q_0)) \cap \Sigma_{con} \mid T_{off}(s\sigma) + T_{*rest}(s\sigma) > \hat{k}^*\} \quad (5.14)$$

ただし , $q_0 \in Q_0$ は初期状態 , $\hat{k}^* \geq \hat{k}$ は目標安定度である .

V^* は TDES G を言語 K に関して安定度 $\hat{k}^*$ で安定化し , さらにそのようなスーパーバイ
ザの中で閉ループ系の生成言語を包含関係に関して最大にするという意味で最適なもの
である .

5.3 最適安定化スーパーバイザの計算アルゴリズム

図 5.1 に最適安定化スーパーバイザに基づくオンライン制御アルゴリズムを示す．最適安定化スーパーバイザを計算するためには， T_{off} と T_{rest} の 2 つの値を計算する必要がある．

T_{off} の値はオンラインで計算する必要がある． T_{*rest} の値はもし言語 K が正規言語ならば， G の状態 q および $u(q)$ の状態 x に対して一意に決まるので，オフラインで計算することができる．

```

Main:
begin
   $s \leftarrow \epsilon$ 
   $T_{off} \leftarrow 0$ 
  while 制御を終えるまで
     $V \leftarrow \Sigma_{unc}$ 
    for  $\forall \sigma \in Elig_G(q) \cap \Sigma_{con}$  do
       $T \leftarrow T_{off} + \|\sigma\|_{tick}$ 
      if  $T + T_{*rest}(s\sigma) > \hat{k}$  then
         $V \leftarrow V \cup \{\sigma\}$ 
      end-if
    end-for
    発生可能な事象集合  $V$  に基づいて制御し，実際に発生した事象  $\nu$  を観測する．
    if  $\nu \in next(s//K, K)$  then
       $T_{off} \leftarrow 0$ 
    else
       $T_{off} \leftarrow T_{off} + \|\nu\|_{tick}$ 
    end-if
     $s \leftarrow s\nu$ 
  end-while
end

```

図 5.1: 最適安定化スーパーバイザに基づいたオンライン制御アルゴリズム

5.4 数値例

図 5.2 の有向グラフで表わされる，時間付き離散事象システムを考える．状態集合は $\{0, 1, \dots, 12\}$ である．また，言語 K を生起する決定性有限オートマトンを図 5.3 に示す．ここでの状態集合は $0, 1$ である． $\alpha, \beta \in \Sigma_{spe}, \gamma \in \Sigma_{rem}$ である．そして， $\gamma \in \Sigma_{con}$ ，つまり事象の生起を抑制できる可制御な事象である．ここでは，初期状態 $q_0 = 0$ ，目標安定度 $\hat{k}^* = 50$ として最適安定化問題について考える．このシステムに対して，まず，安定性判別アルゴリズムを適用する．すると，このシステムにおいて， $u(q)$ に含まれる各状態 x における $Elig_G(x)$ の事象を生起しないような循環的な事象列が存在することが解る．図 5.2 における状態列 q_0, q_1, q_2, q_3, q_1 から，ここで言う循環的な事象列が生起される．よって，このシステムはこのままでは”不安定”である．このとき，このシステムはその循環的な事象列内に可制御な事象 γ を含むので，TDES の可安定性定理より可安定であることが解る．よって，このシステムを安定化するために，スーパーバイザ制御を用いることを考える．つまり，このシステムにおける安定性を阻害している状態遷移，

$$3 \xrightarrow{\gamma} 1 \quad (5.15)$$

を抑制することを考えるのである．

システムを安定化するために，すべての可制御な事象を抑制するような制御を施すスーパーバイザ V_* を用いることも考えられるが，スーパーバイザ V_* を用いた制御は，システムから生起する事象に必要以上の抑制をかけることになり，好ましくないことはさきに述べた通りである．そこで，可制御な事象に目標安定度に応じて制御をかける，最適安定化スーパーバイザ V^* を用いるのである．このシステムに対する最適安定化スーパーバイザの計算アルゴリズムを適用すると，まず，可制御な事象をすべて抑制した $V_*/\langle G, q_0 \rangle$ に対して，安定性判別アルゴリズムを実行する．この結果として，可制御な事象がすべて抑制された図 5.4 と，各状態 $q \in Q$ における $T_{rest}(q)$ を示した表 5.1 の値が得られる．このとき，このシステムは，

$$\hat{k} = \max_{q \in Q} T_{rest}(q) = 3 \quad (5.16)$$

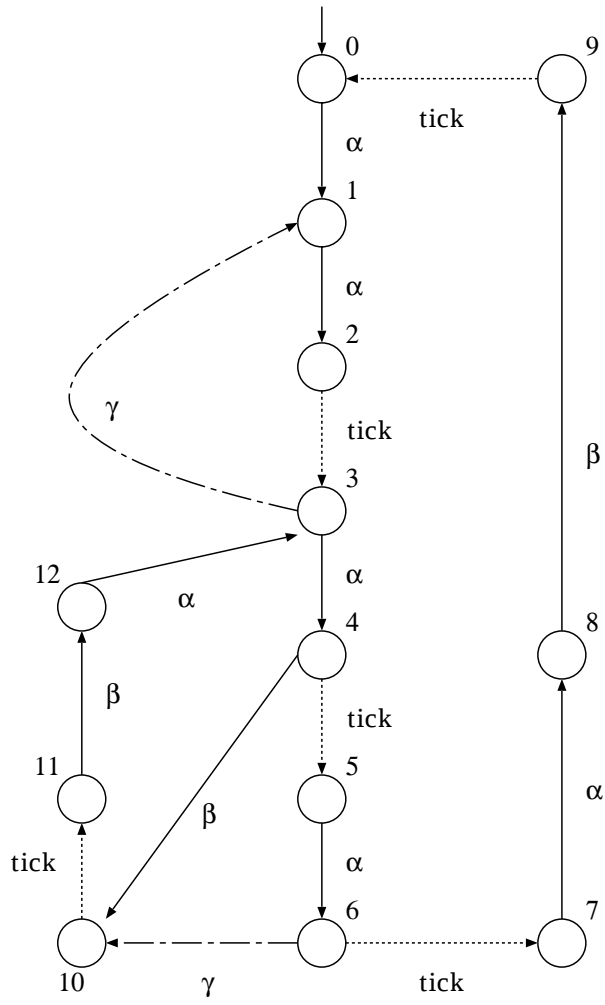
で可安定であることが解る．これらの結果を用いて，最適安定化スーパーバイザ V^* を用いて目標安定度に応じて制御をかけるのである． V^* は，現在，目標事象から離れている時間 T_{off} を計って， $T_{off}(s\sigma) + T_{*rest}(s\sigma)$ の値が目標安定度 $\hat{k}^*$ を越えてしまう場合だけ抑制をかける．この例では， $\hat{k}^* = 50$ であり，問題の状態遷移に対して $T_{*rest}(s\sigma) = 3$ である（ここで， $q_1 = \delta(s\sigma, q_0)$ より， $T_{*rest}(s\sigma) = T_{rest}(q_1) = 3$ が求まる）

よって，もし状態 3 にあるときに，

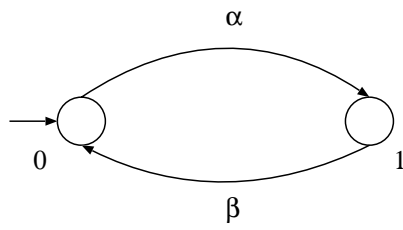
$$T_{off} > 50 - 3 = 47 \quad (5.17)$$

であったなら， γ の生起を抑制する．

ここで注目して欲しいところは，上記以外のような場合のときには，つねに監視はしているが，まったく制御（抑制）をかけていない点である．必要以上に制御を施さないの
で，システムの能力における損失を最小限度に留めることができるのである．



☒ 5.2: An example of TDES G



☒ 5.3: An example of DFA

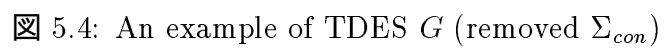


表 5.1: 各状態から , 目標事象が生起するまでの最長時間

TDES G の状態	目標事象が生起するまでの最長時間 T_{rest}
q_0	0
q_1	3
q_2	3
q_3	2
q_4	2
q_5	1
q_6	1
q_7	0
q_8	0
q_9	1
q_{10}	1
q_{11}	1
q_{12}	0

第 6 章

まとめ

本稿では，時間付き離散事象システムに対して，時間を考慮に入れた言語安定性の概念を提案した．また，本稿では提案した安定性を判別するアルゴリズムを示し，制約最小の意味で最適な安定化スーパーバイザとその計算アルゴリズムを示した．

本稿では，時間付き離散事象システム論において重要な役割を果たすフォーシブルな事象が存在しないこと前提として時間事象システムの安定性について述べてきた．フォーシブルな事象とは，もしエリジブルならば，事象 *tick* にとって代わることをできる事象のことである．このフォーシブルな事象の存在により，事象 *tick* は状態に依存して，可制御，不可制御のどちらにもなり得るのであるが，この性質を用いて，事象 *tick* は，それがエリジブルな状態のとき，同時にフォーシブルな事象がエリジブルならば，間接的に生起を抑制できる．つまり，フォーシブルな事象が存在する場合には，事象 *tick* が状態によって可制御にもなりうるのである．このような性質をもつフォーシブルな事象の存在を考慮した安定性の概念を提案することが，今後の課題である．

参考文献

- [1] Peter J. G. Ramadge and W. Murray Wonham : The Control of Discrete Event Systems, Proc. of the IEEE, Vol.77, No.1, 81/98, (1989)
- [2] Y. Brave and M. Heymann : Stabilization of Discrete-Event Processes, Int. J. Control, Vol.51, No.5, 1101/1117, (1990)
- [3] C. M. Ozveren and A. S. Willsky and P. J. Antsaklis : Stability and Stabilizability of Discrete Event Systems, J. ACM, Vol.38, No.3, 730/752, (1991)
- [4] R. Kumar, V. Garg and S. I. Marcus : Language Stability and Stabilizability of Discrete Event Dynamical Systems, SIAM J. contril and Optimization, Vol.31, No.5, 1294/1320, (1993)
- [5] B. A. Brandin and W. M. Wonham : Supervisory Control of Timed Discrete-Event Systems, IEEE Trans. on Automatic Control, Vol.39, No.2, 329/342, (1994)
- [6] 望山 洋, 平石 邦彦 : 時間付き離散事象システムの時間付き安定性, 北陸先端科学技術大学院大学 情報科学研究科 リサーチレポート, IS-RR-97-0043F, ISSN 0918-7553, 22 pages, 1997.
- [7] 望山 洋, 平石 邦彦 : フォーシブルな事象を持つ時間付き離散事象システムの時間付き安定性, 北陸先端科学技術大学院大学 情報科学研究科 リサーチレポート, IS-RR-98-0027F, ISSN 0918-7553, 11 pages, 1998.