

Title	通信遅延を考慮した離散事象システムに対する分散制御方式についての研究
Author(s)	野原, 猛史
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1475">http://hdl.handle.net/10119/1475</a>
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

# 修士論文

## 通信遅延を考慮した離散事象システムに対する 分散制御方式についての研究

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

野原 猛史

2001年2月15日

# 目次

1	はじめに	1
2	スーパーバイザ制御方式	3
2.1	集中制御方式	3
2.2	分散制御方式	4
2.3	コントローラが行う推定	5
2.3.1	通信を行う分散制御方式での推定	5
3	通信を行う分散制御方式	6
3.1	諸定義	6
3.2	通信を行う分散コントローラ	8
3.2.1	通信を行う分散コントローラの必要十分条件の証明	8
3.3	通信を行う myopic 分散コントローラ	9
3.4	通信を行う myopic 分散制御コントローラの必要十分条件の証明	10
3.5	通信を行う分散コントローラと 通信を行う myopic 分散コントローラ	11
4	$DELAY_1$ コントローラ	13
4.1	通信を行う分散コントローラ (以下 $DELAY_0$ ) との比較	14
5	$DELAY_{\leq 1}$ コントローラ	17
6	遅延が 2 以上の定数のときの推定	21
7	おわりに	23

# 第 1 章

## はじめに

離散事象システム (Discrete Event System) は、イベントが非同期的、離散的に発生することによって状態が変化するシステムである。離散事象システムのモデリングや解析については多くの研究が行われてきたが、制御に関する一般的な方法論は少なかった。

Ramadge と Wonham は、一般に理論的解析が困難である評価モデルではなく、扱いやすい形式言語とオートマトンを用いてシステムの動作をモデル化する、スーパーバイザ制御方式を提案した [1]。Ramadge と Wonham は、1 つの制御対象を 1 つのスーパーバイザ (以下、コントローラ) で制御する集中制御方式を提案し、さらにそれを発展させ、複数のコントローラで 1 つの制御対象を制御する分散制御方式 (decentralized control scheme) を提案した。Ramadge と Wonham が提案した分散制御方式は、コントローラ間の通信を全く考慮していなかったが、近年、コントローラ間の通信を行う分散制御方式についての研究が行われている。Barrett と Lafortune による結果 [4] は、その 1 つである。

Barrett と Lafortune が提案したモデルでは、各コントローラによって観測できるイベントと観測できないイベントが存在する。各コントローラはイベントの発生を観測したとき、他のコントローラに送信し、その発生を伝える。各コントローラは、イベントの発生に関する情報を用いて制御対象の制御を行う。彼らは、このモデルで、集中制御方式と同じ制御仕様を満たすための必要十分条件を示した。また、コントローラ間の通信を行うモデルと行わないモデルとの中間のモデルとして、通信で得られる情報を制限した myopic コントローラを提案し、そのモデルにおいて制御仕様を満たすための必要十分条件も示した。

本論文では、分散制御方式におけるコントローラ間の通信に遅延を導入したモデルを提案し、その性質について解析を行った。

2 章では、Ramadge と Wonham によって提案された集中制御方式、分散制御方式につ

いて紹介する。また、通信を行わない分散制御方式と通信を行う分散制御方式との比較からコントローラが行う軌跡の推定の性質を示す。3章では、Barrett と Lafortune によって提案された通信を行う分散制御方式について紹介し、通信を行う myopic 分散コントローラに関する結果を紹介する。制御仕様を満たす必要条件是、可制御性と可観測性であり、可制御性はシステムから比較的簡単に示すことができるが、可観測性を導くには、コントローラの軌跡の推定を求める必要がある。言い換えれば、任意のシステムにおいて、正しい軌跡の推定を示すことができれば、制御仕様を満たすための必要十分条件を導くことができることを示す。4章では、通信の到着が1イベント発生後となる  $DELAY_1$  システムを提案し、 $DELAY_0$  と比較を行いその性質について明らかにする。5章では、通信の到着が1以下のイベント発生後となる  $DELAY_{\leq 1}$  システムを提案し、 $DELAY_1$  との比較を行い、 $DELAY_1$  では制御仕様を満たせるが  $DELAY_{< 1}$  では満たせない例を示した。6章では、通信の到着が2以上の整数  $K$  に対して、軌跡の推定が指数で増加することを示した。

## 第 2 章

# スーパーバイザ制御方式

この章では、この章では、Ramadge と Wonham らによって提案されたスーパーバイザ制御方式について、集中制御方式と分散制御方式、および制御仕様を満たすための必要十分条件について紹介する。

### 2.1 集中制御方式

集中制御方式では、1つの制御対象を1つのコントローラを用いて制御を行う。コントローラを  $Z$ 、制御対象を  $G$  とする。また、 $\mathcal{L}(G)$  は  $G$  の言語であり、 $\mathcal{L}(H)$  は制御仕様を表す言語であり  $\mathcal{L}(H) \subseteq \mathcal{L}(G)$  である。また、 $\mathcal{L}(Z/G)$  は  $Z$  によって  $G$  が制御されたときの言語である。

制御対象で発生するイベントの集合  $\Sigma$  の中でコントローラが観測できるイベント (可観測事象) の集合を  $\Sigma_o$  とし、イベントの発生を抑止できるイベント (可制御事象) の集合を  $\Sigma_c$  とする。また、コントローラが観測できないイベント (不可観測事象) の集合を  $\Sigma_{uo} = \Sigma \setminus \Sigma_o$  とし、イベントの発生を抑止できないイベント (不可制御事象) の集合を  $\Sigma_{uc} = \Sigma \setminus \Sigma_c$  とする。

集中制御方式において、与えられた制御仕様を満たすための必要十分条件は、可制御性と可観測性で与えられる。可制御性は次のように定義される。言語  $\mathcal{L}(H)$  は、 $\mathcal{L}(H)\Sigma_{uc}^* \cap \mathcal{L}(G) \subseteq \mathcal{L}(H)$  のとき、かつそのときに限って  $\mathcal{L}(G)$  と  $\Sigma_{uo}$  に関して可制御である。また、 $\mathcal{L}(H)$  は prefix closed である。この条件は、コントローラが観測できないイベントが発生したとしても、 $\mathcal{L}(H)$  から外れることはない事を示している。

また、可観測性は次のように定義される。言語  $\mathcal{L}(H)$  は、もし  $(\forall s, s' \in \mathcal{L}(H))(\forall \sigma \in \Sigma) : \mathcal{P}(s) = \mathcal{P}(s'), s\sigma \in \mathcal{L}(H), s'\sigma \in \mathcal{L}(G) \Rightarrow s'\sigma \in \mathcal{L}(H)$  ならば、 $\mathcal{L}(G), \Sigma_o, \Sigma_c$  に関して

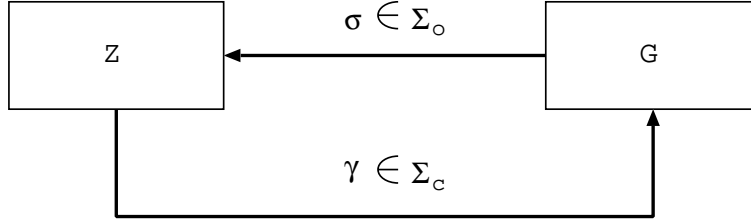


図 2.1: 集中制御方式

可観測である。ここで、 $\mathcal{P}$  は  $\sigma \in \Sigma_o$  ならば  $\mathcal{P}(\sigma) = \sigma$  であり、その他の場合は  $\mathcal{P}(\sigma) = \varepsilon$  である。また、 $\mathcal{P}^{-1}(\omega) = \{s \in \Sigma^* | \mathcal{P}(s) = \omega\}$  である。この条件は、次のように書き換えることができる。

$$(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c):$$

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow [\mathcal{P}^{-1}(\mathcal{P}(s))\sigma \cap \mathcal{L}(H) = \emptyset].$$

これは、 $\mathcal{L}(H)$  から飛び出させるイベントを全て抑止しても  $\mathcal{L}(H)$  の要素が発生しうることを示している。以上の 2 つの条件から、望ましい振る舞いを示す  $\mathcal{L}(H)$  が達成される。すなわち、 $\mathcal{L}(H) = \mathcal{L}(Z/G)$ 。

## 2.2 分散制御方式

分散制御方式は、1 つの制御対象を複数のコントローラを用いて制御を行う。コントローラの集合を  $Z = \{Z_1, Z_2, \dots, Z_n\}$  とする。図 2.2 は  $n = 2$  の場合の分散システムの例である。 $\Sigma_{c,i}, \Sigma_{o,i}$  は、それぞれのコントローラの発生を抑止できるイベントの集合と観測できるイベントの集合であり、 $\Sigma_{uc,i}, \Sigma_{uo,i}$  は、それぞれ発生を抑止できないイベントの集合と観測できないイベントの集合である。

分散制御方式における可制御性と可観測性は以下ようになる。

可制御性 言語  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uo}$  に関して可制御であり、かつ

可観測性  $(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c):$

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow$$

$$(\exists i \in Z) [\mathcal{P}_{\Sigma_{o,i}}^{-1}(\mathcal{P}_{\Sigma_{o,i}}(s))\sigma \cap \mathcal{L}(H) = \emptyset] \wedge [\sigma \in \Sigma_{c,i}].$$

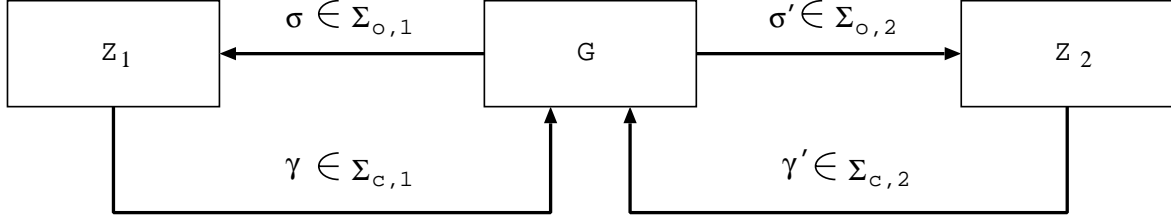


図 2.2: 分散制御方式 ( $Z = \{Z_1, Z_2\}$  の場合)

可制御性は集中制御方式と同じである。しかし、可観測性の条件は各コントローラが観測できるイベントが  $\mathcal{P}_{\Sigma_{o,i}}$  によっ

て制限されるため、

$$\mathcal{P}^{-1}(\mathcal{P}(s)) \subseteq \mathcal{P}_{\Sigma_{o,i}}^{-1}(\mathcal{P}_{\Sigma_{o,i}}(s))$$

となり、さらに  $\mathcal{P}_{\Sigma_{o,i}}^{-1}(\mathcal{P}_{\Sigma_{o,i}}(s)) \cap \mathcal{L}(H)$  が空のとき、 $\sigma$  の発生を抑止しなければならず、集中制御方式と比べて厳しい条件となる。

## 2.3 コントローラが行う推定

通信の無い分散制御方式では、各コントローラは直接観測できるイベントのみからシステムの軌跡を推測しなければならない。すなわちあるコントローラが  $\alpha, \beta, \gamma$  の順にイベントを観測したとすると、初期状態から現在までの軌跡の推定は、観測したイベント間に自分自身で観測できないイベントが挟まる形になり、 $\Sigma_{u_{o,i}}^* \alpha \Sigma_{u_{o,i}}^* \beta \Sigma_{u_{o,i}}^* \gamma$  となる。これは、システム全体の軌跡を  $s$  とすると、 $\Phi = \mathcal{P}_{\Sigma_{o,i}}^{-1}(\mathcal{P}_{\Sigma_{o,i}}(s))$  と一致する。また、 $\gamma$  の次にイベント  $\sigma$  を観測するまでの軌跡の推定は、 $\Sigma_{u_{o,i}}^*$  である。

### 2.3.1 通信を行う分散制御方式での推定

通信を行う分散制御方式では、コントローラはイベントを観測するだけでなく、他のコントローラの推定も更新に用いることができる。通信を行う分散制御方式での初期状態から現在までの軌跡の推定は、

$$\bigcup_{j \in Z} \Sigma_{o,j} = \Sigma_o$$

より、イベント間に  $\Sigma_{u_o}^*$  を挟む必要があり、 $\Phi = \mathcal{P}^{-1}(\mathcal{P}(s))$  と一致する。また次のイベント  $\sigma$  が発生するまでの軌跡の推定も同様に  $\Sigma_{u_o}^*$  である。



# 第 3 章

## 通信を行う分散制御方式

Ramadge と Wonham によって示された分散制御方式は、コントローラ間の通信は考慮していなかったが、近年、通信を行う分散制御方式のモデルの研究が行われている。これらのモデルでは、各コントローラは、観測できるイベントと他のコントローラから送られてくる情報を用いて制御対象の軌跡を推定し、制御仕様に従った正しい制御動作を決める。

### 3.1 諸定義

Barrett と Lafortune が提案したモデルでは、コントローラが通信および制御に利用することができる情報は、情報構造 (information structure)

$$\mathcal{I} := \{T, (\Sigma_{o,i}, \Xi_i, \Psi_i) : i = 1, \dots, n\}$$

によって表される。ここで、 $T$  はシステムで発生し得る全ての軌跡を示し、一般的に

$$T \subseteq (\Sigma[\mathcal{C}_{\Xi^{n \times n}}][\mathcal{C}_{\Xi^{n \times n}}])^*$$

と表す。ここで、 $\Sigma$  は制御対象で起こり得るイベントの集合であり、最初の  $\mathcal{C}_{\Xi^{n \times n}}$  は、イベント発生直後にコントローラ間で送信される情報であり、最後の  $\mathcal{C}_{\Xi^{n \times n}}$  は、コントローラ間で受信された情報を表す。Barrett と Lafortune のモデルでは単純化のために、通信の損失や遅延が無いと仮定し  $T$  を

$$T \subseteq (\Sigma[\mathcal{C}_{\Xi^{n \times n}}])^*$$

としている。また、 $\Xi_i$  はコントローラ  $i$  が通信に使うことができる記号の集合であり、 $\Psi_i$  は、

$$\Psi_i : T_i \rightarrow \rho(\mathcal{L}(Z/G))$$

で与えられる推論マップである。ここで、 $T_i$  は、コントローラ  $i$  が観測できたイベントと通信を含めた軌跡の集合であり、 $\rho(\mathcal{L}(Z/G))$  は、 $\mathcal{L}(Z/G)$  のべき集合を表す。推論マップは、各コントローラの制御方針、通信方針に利用され、制御方針は  $\Gamma = \{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$  であり、制御マップは

$$\Gamma_i : \rho(\mathcal{L}(Z/G)) \rightarrow \rho(\Sigma_{c,i})$$

である。また、通信方針は  $\Theta = \{\Theta_1, \Theta_2, \dots, \Theta_n\}$  であり、通信マップ

$$\Theta_i : \rho(\mathcal{L}(Z/G)) \rightarrow \prod_{k=1}^n \rho(\Xi_i)$$

によって示される。

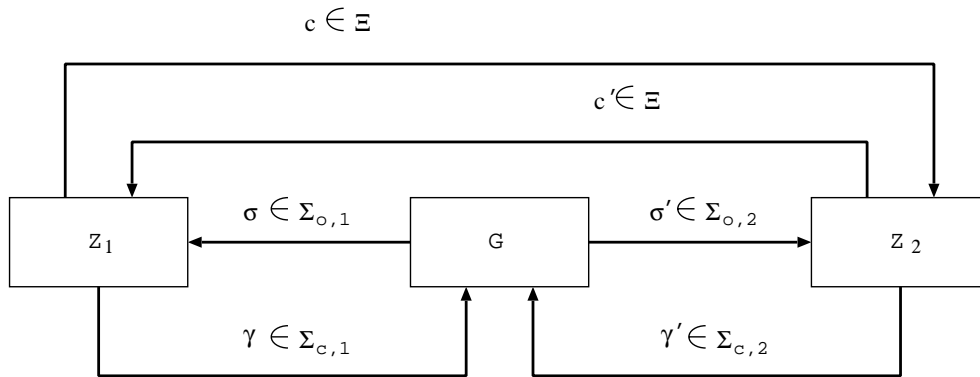


図 3.1: 通信を行う分散制御方式

Barrett と Lafortune は、問題 (P) に解を与える通信方針と制御方針を示し、通信を行う分散制御コントローラと通信を行う myopic 分散制御コントローラの 2 つのコントローラを提案し、それぞれについて  $\mathcal{L}(Z/G) = \mathcal{L}(H)$  となるに制御仕様を満たすための必要十分条件を示している。

(P)  $\mathcal{L}(G)$  を生成する制御対象  $G$ 、言語  $\mathcal{L}(H) \subseteq \mathcal{L}(G)$  を生成するオートマトン  $H$  によってモデル化された制御仕様、そして、コントローラの集合  $Z = \{Z_1, Z_2, \dots, Z_n\}$  が与えられたとき、コントローラのための制御方針  $\Gamma = \{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$  と通信方針  $\Theta = \{\Theta_1, \Theta_2, \dots, \Theta_n\}$  をそれぞれ  $\mathcal{L}(Z/G) = \mathcal{L}(H)$  となるように構成する。

1. 通信を行う分散制御コントローラ: コントローラが  $T_i$  の軌跡に沿った通信の受信を予測でき、かつ、予測された通信に基づく制御対象の軌跡を推定することができ

る。すなわち、イベントが発生したとき、そのイベントを観測したコントローラが他のコントローラに更新した推定を送信し、受信したコントローラは受信した情報を元に推定を更新する。

2. 通信を行う myopic 分散コントローラ: コントローラが制御対象の軌跡を推定するときに、通信を行う分散制御コントローラのような通信の予測を行わない。すなわち、自分自身でイベントを観測するまでその情報を使わない。

## 3.2 通信を行う分散コントローラ

通信を行う分散コントローラにおいて、以下の可制御性と可観測性の 2 つの条件を満たすとき、問題 (P) に解を与える情報構造が存在し、制御仕様を満たすことが示されている。

1.  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uc}$  に関して可制御。
2.  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$ ,  $\Sigma_o$  と  $\Sigma_c$  に関して可観測。

制御仕様を表す言語  $\mathcal{L}(H)$  が、集中制御方式によって制御仕様を満たすための必要十分条件と一致するため、集中制御方式で実現できる制御仕様が、この条件を満たす情報構造を用いることにより、通信を行う分散制御方式によって実現することができる。2 の条件は次のように書き換えることができる。

$$(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c):$$

$$\begin{aligned} & [s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow \\ & (\exists i \in Z) \left[ \mathcal{P}^{-1}(\mathcal{P}(s))\sigma \cap \mathcal{L}(H) = \emptyset \right] \wedge [\sigma \in \Sigma_{c,i}]. \end{aligned}$$

### 3.2.1 通信を行う分散コントローラの必要十分条件の証明

Barrett と Lafortune によって示された証明は以下のようなになる。

1.  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uc}$  に関して可制御。
2.  $(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c):$

$$\begin{aligned} & [s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow \\ & (\exists i \in Z) \left[ \mathcal{P}^{-1}(\mathcal{P}(s))\sigma \cap \mathcal{L}(H) = \emptyset \right] \wedge [\sigma \in \Sigma_{c,i}]. \end{aligned}$$

必要条件

- (a) 軌跡の推定を、イベントが発生する毎に送信する。
- (b) 制御マップを次のように定義する。

$$\Gamma_i(\mathcal{P}^{-1}(\mathcal{P}(s))) = \{\sigma \in \Sigma_{c,i} \mid \mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo}^*\sigma \cap \mathcal{L}(H) = \phi\}$$

もし、 $s\sigma \in \mathcal{L}(H)$  ならば、全てのコントローラも  $\sigma$  を抑止し、また、 $s\sigma \notin \mathcal{L}(H)$ 、 $\sigma \in \Sigma_c$ 、 $s\sigma \in \mathcal{L}(G)$  ならば (b) の制御マップと 2 の可観測性より  $\sigma$  が抑止される。

もし、 $s \in \mathcal{L}(H)$ 、 $s\sigma \in \mathcal{L}(G)$ 、 $\sigma \in \Sigma_{uc}$  ならば、1 の可制御性は  $s\sigma \in \mathcal{L}(H)$  を示し、どのコントローラも  $\sigma$  を抑止しない。

従って、問題 (P) は、通信を行う分散コントローラで解くことができる。

十分条件

もし、問題 (P) が、通信を行う分散コントローラ  $Z$  で解くことができるならば、集中制御コントローラでも解くことができる。したがって、1 可制御性は満たされる。

ここで、2 の可観測性が満たさないと仮定すると、

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \wedge [(\forall i \in Z)[\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo}^*\sigma \cap \mathcal{L}(H) \neq \phi] \vee [\sigma \notin \Sigma_{c,i}]]$$

が成り立つような  $s \in \mathcal{L}(H)$  が存在する。このとき、 $\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo}^*\sigma \cap \mathcal{L}(H) \neq \phi$  より、任意のコントローラ  $i \in Z$  が、 $s$  の後で  $\sigma$  を抑止しなければならない場合と、 $s'$  の後で  $\sigma$  の発生を許さなければならない場合があることを示す。このことから、(b) の制御マップより  $\sigma$  の発生を許さなければならず、制御仕様  $\mathcal{L}(H)$  を満たすことはできない。ゆえに、2 の可観測性は満たされる。

### 3.3 通信を行う myopic 分散コントローラ

通信を行う myopic 分散コントローラにおいて、以下の可制御と可観測の条件を満たすとき、問題 (P) に解を与える情報構造が存在し、制御仕様を満たすための必要十分条件が示されている。

1.  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uc}$  に関して可制御。
2.  $(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c)$ :

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow$$

$$(\exists i \in Z)[\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo,i}^*\sigma \cap \mathcal{L}(H) = \emptyset] \wedge [\sigma \in \Sigma_{c,i}].$$

### 3.4 通信を行う myopic 分散制御コントローラの必要十分条件の証明

Barrett と Lafortune によって示された証明は以下のようなになる。

1.  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uc}$  に関して可制御。
2.  $(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c)$ :

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow$$

$$(\exists i \in Z) [\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo,i}^*\sigma \cap \mathcal{L}(H) = \emptyset] \wedge [\sigma \in \Sigma_{c,i}].$$

必要条件

- (a) 軌跡の推定を、イベントが発生する毎に送信する。
- (b) 制御マップ

$$\Gamma_i(\mathcal{P}^{-1}(\mathcal{P}(s))) = \{\sigma \in \Sigma_{c,i} | \mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo,i}^*\sigma \cap \mathcal{L}(H) = \emptyset\}$$

もし、 $s\sigma \in \mathcal{L}(H)$  ならば、全てのコントローラも  $\sigma$  を抑止しない。また、 $s\sigma \notin \mathcal{L}(H)$ 、 $\sigma \in \Sigma_c$ 、 $s\sigma \in \mathcal{L}(G)$  ならば (b) の制御マップと 2 の可観測性より  $\sigma$  が抑止される。

もし、 $s \in \mathcal{L}(H)$ 、 $s\sigma \in \mathcal{L}(G)$ 、 $\sigma \in \Sigma_{uc}$  ならば、1 の可観測性は  $s\sigma \in \mathcal{L}(H)$  を示し、全てのコントローラが  $\sigma$  を抑止しない。

ゆえに、問題 (P) は、通信を行う myopic 分散コントローラで解くことができる。

十分条件

もし、問題 (P) が、通信を行う myopic 分散制御コントローラ  $Z$  で解くことができるならば、集中制御コントローラでも解くことができる。ゆえに 1 の可制御性は満たされる。

ここで、2 の可観測性が満たされないと仮定すると、

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \wedge [(\forall i \in Z) [\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo,i}^*\sigma \cap \mathcal{L}(H) \neq \emptyset] \vee [\sigma \notin \Sigma_{c,i}]]$$

が成り立つような  $s \in \mathcal{L}(H)$  が存在する。このとき、 $\mathcal{P}^{-1}(\mathcal{P}(s))\Sigma_{uo,i}^*\sigma \cap \mathcal{L}(H) \neq \emptyset$  より、任意のコントローラ  $i \in Z$  が、 $s$  の後で  $\sigma$  を抑止しなければならない場合と、 $s'$  の後で  $\sigma$  の発生を許さなければならない場合があることを示す。このことから、(b) の抑止写像より  $\sigma$  の発生を許さなければならない場合、制御仕様  $\mathcal{L}(H)$  を満たすことはできない。ゆえに、2 の可観測性は満たされる。

### 3.5 通信を行う分散コントローラと

#### 通信を行う myopic 分散コントローラ

通信を行う分散制御コントローラは、イベントの発生を観測したとき、または、他のコントローラからイベントを受信したとき、得られた情報から推定を更新し、新しい推定を送信し、受信したコントローラは受信した情報を基に推定を更新する。

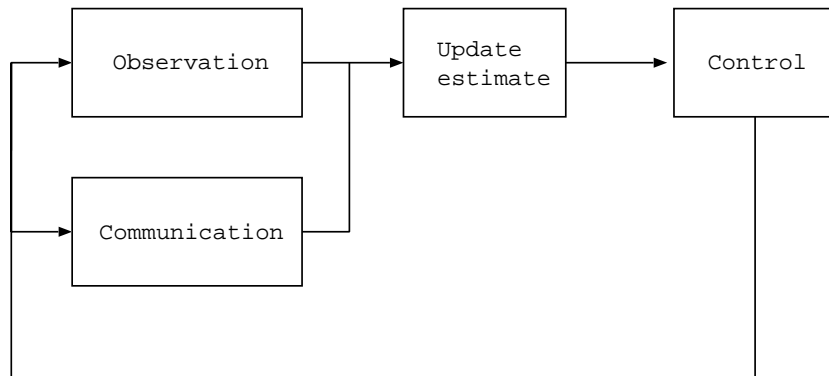


図 3.2: 通信を行う分散制御コントローラ

通信を行う myopic 分散制御コントローラは、通信は受け取るが、自分自身でイベントを観測するまではその情報を使わないコントローラである。

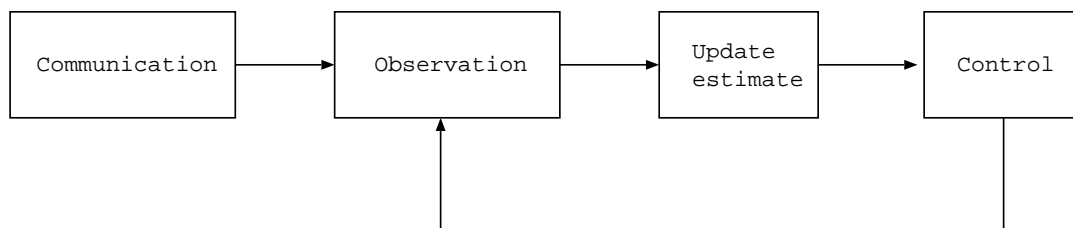


図 3.3: 通信を行う myopic 分散制御コントローラ

上記のような、通信を行う分散制御コントローラと通信を行う myopic 分散コントローラから得られる必要十分条件の違いは、通信の無い分散制御コントローラと通信を行う分散制御コントローラとの比較で述べたように、可観測性の条件において、初期状態から現在までの軌跡の推定は変わらず、現在から次にイベントが発生するまでの推定のみが異

なっている。すなわち、それぞれの推定は

通信のある分散制御コントローラ:  $\Phi \Sigma_{uo}^*$

myopic 分散制御コントローラ:  $\Phi \Sigma_{uo,i}^*$

となる。

以上のことから可観測性は、任意のシステムにおいて、そのシステムのコントローラが行う推定を求めることにより導くことができ、それにより、制御仕様を満たすための必要十分条件を導くことができる。

# 第 4 章

## DELAY<sub>1</sub> コントローラ

DELAY<sub>1</sub> コントローラは、図 4.1 のように、通信の到着が 1 イベント発生後となるシステムである。問題を単純にするために、 $\sigma \in \Sigma_{o,1}, \Sigma_{o,2}$  のように、複数のコントローラが観測できるイベントは存在しないと仮定する。これは、コントローラが観測したイベントと受信したイベントが連続してしまうためである。

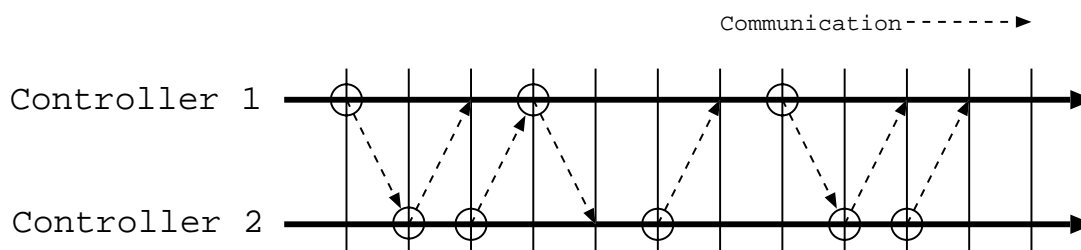


図 4.1: 通信遅延が 1 イベントの場合

通信にこのような遅延を導入した場合、

1. イベントを観測
2. 情報を受信
3. イベントを観測すると同時に情報を受信

の 3 つの場合について考えなければならない。現在から次にイベントが発生もしくは情報を受信するまでの推定の定義より、推定は次のイベントまでの全ての軌跡を含みかつ、余分な軌跡を含んではならない。このことから、上記の 3 つの場合について、それぞれ



表 4.1: DELAY<sub>1</sub> の推定の構成

イベントを観測したとき	
$\Sigma_{uo}^* \Sigma_{uo} \Sigma_{uo}$	×
$\Sigma_{uo}^* \Sigma_{uo} \Sigma_{o,i}$	○
$\Sigma_{uo}^* \Sigma_{uo} \{\Sigma_o \setminus \Sigma_{o,i}\}$	×
情報を受信したとき	
$\Sigma_{uo}^* \Sigma_{o,i} \Sigma_{uo}$	×
$\Sigma_{uo}^* \Sigma_{o,i} \Sigma_{o,i}$	×
$\Sigma_{uo}^* \Sigma_{o,i} \{\Sigma_o \setminus \Sigma_{o,i}\}$	×
イベントを観測すると同時に情報を受信したとき	
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{uo}$	○
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{o,i}$	○
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \{\Sigma_o \setminus \Sigma_{o,i}\}$	○
発生直前までの推定	$\Sigma_{uo}^* \{\Sigma_{uo} + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$
発生までの推定	$\Sigma_{uo}^* \{\Sigma_{o,i} + \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma\}$

さらに上記の 3 つの場合が起るまでの軌跡を求める。○ の軌跡は推定に含まれ、× の軌跡は推定に含まれない。

#### 4.1 通信を行う分散コントローラ (以下 DELAY<sub>0</sub>) との比較

$A, B, C$  をそれぞれ以下のようなイベントとする。

$$A \in \Sigma_{uo}$$

$$B \in \Sigma_{o,i}$$

$$C \in \Sigma_o \setminus \Sigma_{o,i}$$

また、 $B \in \Sigma_c$  を  $[B]$  のように表す。

$\text{DELAY}_0$  の推定を  $s$ 、 $\text{DELAY}_1$  の推定を  $s'$  とすると、情報が通信で送られてきたとき、 $\text{DELAY}_1$  では必ず  $\Sigma$  が発生し、送られてきた  $C$  と  $\Sigma$  の間にはイベントが発生しないことを示す。遅延が  $k$  の場合には  $\Sigma_{uo}^{k-1}$  となる。この性質は、連続して  $C$  と  $B$  が同時に送られてくる場合では、 $\text{DELAY}_0$  のようにイベントとイベントの間に  $\Sigma_{uo}^*$  が入らない。

$$s = \Sigma_{uo}^* C \Sigma_{uo}^* B \Sigma_{uo}^*$$

$$s' = \Sigma_{uo}^* C \Sigma_{uo}^{k-1} \Sigma \Sigma_{uo}^*$$

次の定理において、 $\mathcal{L}(G)$  は、制御対象の言語であり、 $\Sigma_{o,i}, \Sigma_{c,i}$  はコントローラ  $i = 1, \dots, n$  のそれぞれに与えられる。 $\mathcal{C}_{\mathcal{L}, \text{DELAY}_0}$  は、 $\mathcal{L}(G), \text{DELAY}_0$  に関する可観測の言語のクラスである。 $\mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$  は、 $\mathcal{L}(G), \text{DELAY}_1$  に関する可観測の言語のクラスである。

定理 5.1  $\mathcal{C}_{\mathcal{L}, \text{DELAY}_0} \not\subseteq \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$

証明  $\text{DELAY}_1$  で制御可能であるが、 $\text{DELAY}_0$  では制御できないシステムが存在することを示す。

図 4.2 に示すように、 $C$  が発生した後に、 $\text{DELAY}_0$  であれば、 $\Sigma_{uo}^* C \Sigma_{uo}^*$  となり、状態 2 もしくは 3 を推定し制御することができなくなる。それに対して、 $\text{DELAY}_1$  では、 $\Sigma_{uo}^* C B \Sigma_{uo}^*$  もしくは、 $\Sigma_{uo}^* C \Sigma_{uo, i} \Sigma_{uo}^*$  となり、後者であれば、明らかに状態 3 であることが推定でき、制御することができる。 ■

定理 5.2  $\mathcal{C}_{\mathcal{L}, \text{DELAY}_0} \not\subseteq \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$

証明  $\text{DELAY}_0$  で制御可能であるが、 $\text{DELAY}_1$  では制御できないシステムが存在することを示す。

図 4.3 では、明らかに  $\text{DELAY}_1$  では、状態 2 を推定することができないので、制御できない。 ■

以上のことから、 $\text{DELAY}_0$  と  $\text{DELAY}_1$  では、互いに含まない言語が存在する。

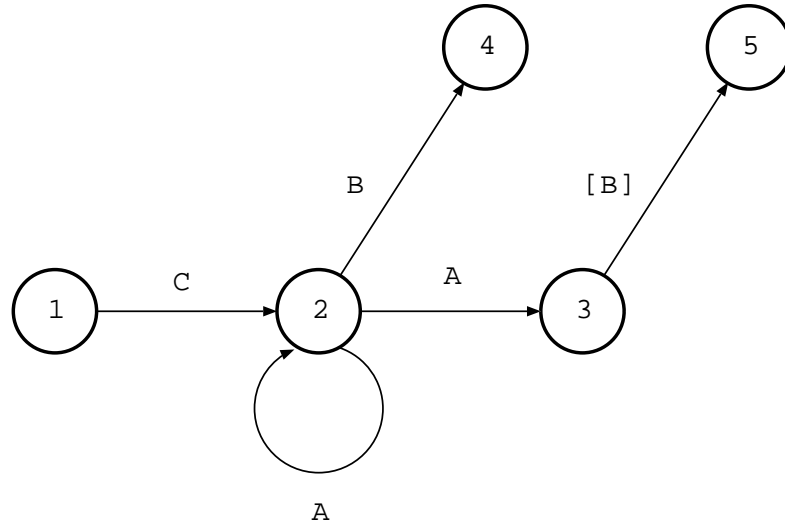


図 4.2:  $DELAY_0$  では制御できない例

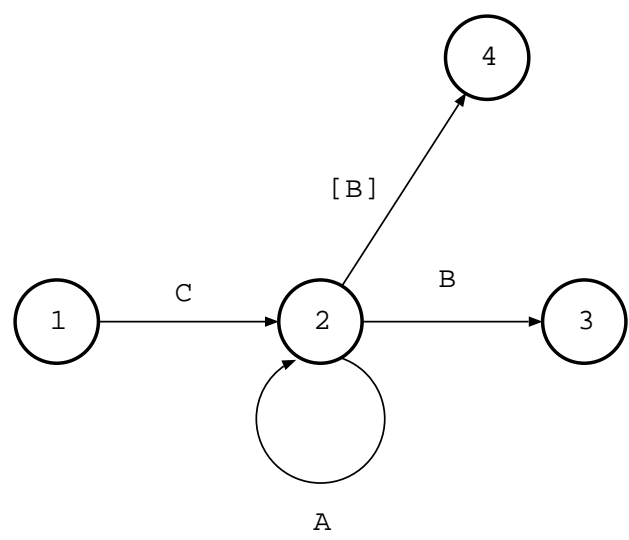


図 4.3:  $DELAY_1$  では制御できない例

## 第 5 章

### DELAY<sub>≤1</sub> コントローラ

DELAY<sub>≤1</sub> コントローラは、図 5.1 のように通信の到着が、1 個以下のイベント発生後となるシステムある。問題を単純にするために、このシステムでは、遅延が 0 の通信と、遅延が 1 の通信が同時に到着するような通信は考慮しない。これは、同時に受信した場合にイベント間に入れ替りが発生するのを避けるためである。

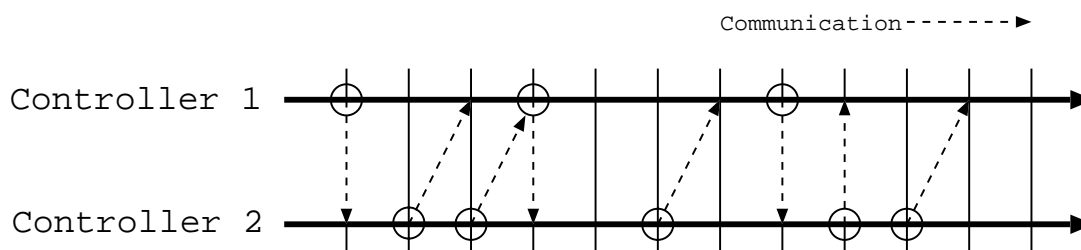


図 5.1: 通信遅延が 1 イベント以下の場合

DELAY<sub>1</sub> の推定と同様に、全ての場合について推定を求めて和集合を取る。

表 5.1:  $\text{DELAY}_{\leq 1}$  の推定の構成

	$\text{DELAY}_1$	$\text{DELAY}_0$	$\text{DELAY}_{\leq 1}$
<b>イベントを観測したとき</b>			
$\Sigma_{uo}^* \Sigma_{uo} \Sigma_{uo}$	×	×	×
$\Sigma_{uo}^* \Sigma_{uo} \Sigma_{o,i}$	○	○	○
$\Sigma_{uo}^* \Sigma_{uo} \{\Sigma_o \setminus \Sigma_{o,i}\}$	×	○	○
<b>情報を受信したとき</b>			
$\Sigma_{uo}^* \Sigma_{o,i} \Sigma_{uo}$	×	×	×
$\Sigma_{uo}^* \Sigma_{o,i} \Sigma_{o,i}$	×	×	×
$\Sigma_{uo}^* \Sigma_{o,i} \{\Sigma_o \setminus \Sigma_{o,i}\}$	×	×	×
<b>イベントを観測し 情報を受信したとき</b>			
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{uo}$	○	×	○
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{o,i}$	○	×	○
$\Sigma_{uo}^* \{\Sigma_o \setminus \Sigma_{o,i}\} \{\Sigma_o \setminus \Sigma_{o,i}\}$	○	×	○
<b>直前まで</b>	$\Sigma_{uo}^* \{\Sigma_{uo} + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$	$\Sigma_{uo}^*$	$\Sigma_{uo}^* \{\Sigma_{uo} + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$
<b>発生まで</b>	$\Sigma_{uo}^* \{\Sigma_{o,i} + \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma\}$	$\Sigma_{uo}^* \Sigma_o$	$\Sigma_{uo}^* \{\Sigma_o + \{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma\}$

定理 6.1  $\mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}} \subseteq \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$

証明 (P)  $\in \mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}} \Rightarrow (P) \text{in} \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$  を示すと、定理 6.1 の関係が成り立つことを示す。

$\Phi \Sigma_{uo}^* \{\varepsilon + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$  は  $\text{DELAY}_{\leq 1}$  コントローラの推定であり、 $\Phi' \Sigma_{uo}^* \{\varepsilon + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$  は  $\text{DELAY}_1$  コントローラの推定である。(P)  $\in \mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}}$  が成り立つ必要十分条件は、

可制御性 言語  $\mathcal{L}(H)$  は、 $\mathcal{L}(G)$  と  $\Sigma_{uo}$  に関して可制御であり、かつ

可観測性  $(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c)$ :

$$[s\sigma \notin \mathcal{L}(H)] \wedge [s\sigma \in \mathcal{L}(G)] \Rightarrow$$

$$(\exists i \in Z) [\Phi \Sigma_{uo}^* \{\varepsilon + \{\Sigma_o \setminus \Sigma_{o,i}\}\} \sigma \cap \mathcal{L}(H) = \emptyset] \wedge [\sigma \in \Sigma_{c,i}].$$

である。可制御性は  $\text{DELAY}_1$  コントローラについても同様に成り立つ。 $\Phi_{d0}$  を  $\text{DELAY}_0$  の推定、 $\Phi_{d1}$  を  $\text{DELAY}_1$  の推定とすると、

$$\Phi = \Phi_{d0} \cap \Phi_{d1}$$

$$\Phi' = \Phi_{d1}$$

$$\Rightarrow \Phi' \subseteq \Phi$$

よって、

$$\Phi \Sigma_{uo}^* \{\varepsilon + \{\Sigma_o \setminus \Sigma_{o,i}\}\} \sigma \cap \mathcal{L}(H) = \emptyset$$

ならば

$$\Phi' \Sigma_{uo}^* \{\varepsilon + \{\Sigma_o \setminus \Sigma_{o,i}\}\} \sigma \cap \mathcal{L}(H) = \emptyset$$

すなわち、 $\text{DELAY}_1$  の可観測性が満たされる。

以上から、(P)  $\in \mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}} \Rightarrow (P) \text{in} \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$  ■

定理 6.2  $\mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}} \neq \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$

証明  $\text{DELAY}_1$  で制御可能であるが、 $\text{DELAY}_{\leq 1}$  では制御できないシステムが存在することを示す。

$\text{DELAY}_1$  と  $\text{DELAY}_{\leq 1}$  では、推定が  $\Sigma_{uo}^* \{\Sigma_{uo} + \{\Sigma_o \setminus \Sigma_{o,i}\}\}$  であることがわかる。しかし、次の場合に於て  $\text{DELAY}_1$  では制御可能であるが  $\text{DELAY}_{\leq 1}$  では制御できない例が存在する。

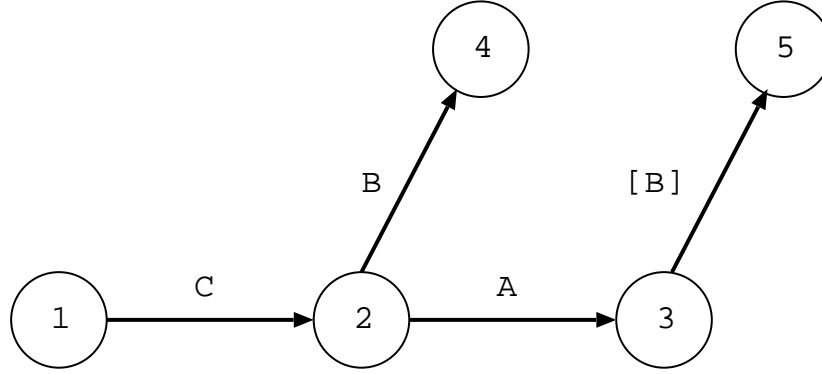


図 5.2:  $\text{DELAY}_{\leq 1}$  では制御できない例

図 5.2 のオートマトンでの、 $\text{DELAY}_1$  と  $\text{DELAY}_{\leq 1}$  の推定の更新を考える。初期状態 1 での推定は、どちらのシステムにおいても  $\Sigma_{uo}^* \{ \varepsilon + \Sigma_o \setminus \Sigma_{o,i} \}$  である。この推定のとときに  $C$  を受信した場合を考える。 $\text{DELAY}_1$  の場合は、明らかに推定は  $\Sigma_{uo}^* C \Sigma_{uo,i} \Sigma_{uo}^* \{ \varepsilon + \Sigma_o \setminus \Sigma_{o,i} \}$  であり、軌跡が  $CA$  であることを推測できる。それに対して、 $\text{DELAY}_{\leq 1}$  の場合は、さらに、通信の遅延が 0 の場合について考えなければならない。遅延が 0 のときの推定は  $\Sigma_{uo}^* C \Sigma_{uo}^* \{ \varepsilon + \Sigma_o \setminus \Sigma_{o,i} \}$  である。 $\text{DELAY}_{\leq 1}$  は、遅延は 0 か 1 のどちらかであり、受信した通信がどちらの遅延で到着したかは知ることができない。そのため、全ての軌跡を含む推定は、 $\text{DELAY}_0$  の推定と  $\text{DELAY}_1$  の推定の和集合をとり、 $\Sigma_{uo}^* C \{ \varepsilon + \Sigma_{uo,i} \} \Sigma_{uo}^* \{ \varepsilon + \Sigma_o \setminus \Sigma_{o,i} \}$  となる。このとき、 $\text{DELAY}_{\leq 1}$  の軌跡の推定は  $C$  もしくは  $CA$  となり、 $B$  のいずれかを抑止することができなくなる。ゆえに、図 5.2 は、 $\text{DELAY}_{\leq 1}$  では制御仕様を満たすことはできない。 ■

定理 6.1 と 定理 6.2 より、以下の系が導かれる。

系

$$\mathcal{C}_{\mathcal{L}, \text{DELAY}_{\leq 1}} \subsetneq \mathcal{C}_{\mathcal{L}, \text{DELAY}_1}$$

## 第 6 章

### 遅延が 2 以上の定数のときの推定

通信の到着が 2 以上の正数  $K$  イベント発生後となるシステムでは、通信を受信したとき、受信した瞬間のイベントと受信したイベントの間には  $K - 1$  イベントが既に発生している。このため、初期状態から現在までの推定  $\Phi$  は、通信の遅延が 1 イベント以下の定数の場合と異なり、図 6.1 のように、受信したイベントがこれ以上挟まれない部分と、通信が送られてきた場合に新たに更新しなければならない  $K - 1$  の部分に分けられる。

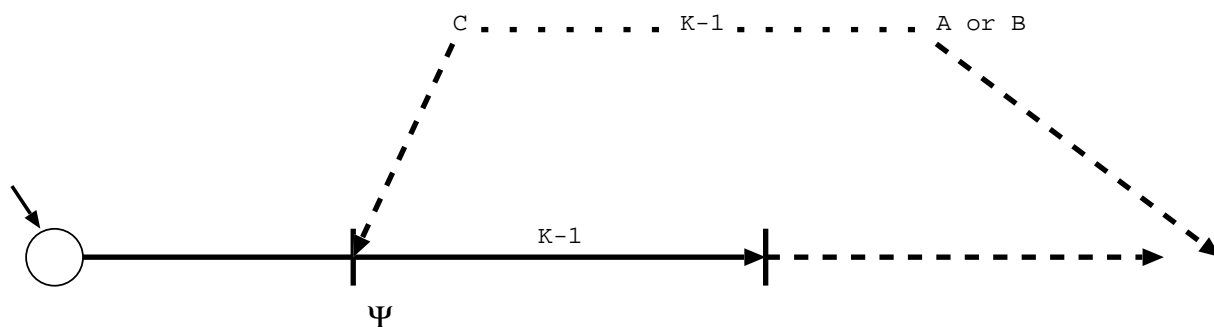


図 6.1: 遅延が 2 以上の定数のときの推定更新

通信を受信した場合について考えると、通信を受信した瞬間は  $\Sigma_o$  を観測した場合と、 $\Sigma_{uo,i}$  が発生した場合に分けることができる。すなわち、 $\Phi$  に挟み込まなければならない軌跡は、 $\{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{uo}^K - 1 \Sigma_o$  もしくは  $\{\Sigma_o \setminus \Sigma_{o,i}\} \Sigma_{uo}^K$  である。また、受信したイベントを挟み込む時は、次の 2 つの条件を満たさなければならない。

1. 全ての観測したイベントの順番は入れ替わらない
2. 全ての受信したイベントの順番は入れ替わらない



以上の条件を満たした上で、イベントが最も入れ替わる場合について考える。遅延が  $K$  のとき、 $C$  が  $K$  個連続して発生し、その後  $B$  が  $K$  個発生した場合、イベントの差し込みによって生じる現在までの軌跡の推定数は、

$$\sum_{n=0}^{K-1} \binom{K-1}{n} = 2^{K-1}$$

で表すことができる。明らかに、この部分での軌跡の推定は遅延  $K$  の指数個存在する。

# 第 7 章

## おわりに

本稿では、通信のある分散制御方式に、通信の到着が 1 イベント発生後となる  $\text{DELAY}_1$  を提案した。 $\text{DELAY}_1$  の言語クラスは  $\mathcal{C}_{\mathcal{L},\text{DELAY}_1}$  であり、通信を行わない分散制御方式の言語クラスは  $\mathcal{C}_{\mathcal{L},\text{nocomunication}}$  であり、通信を行う myopic 分散制御方式の言語のクラスは、 $\mathcal{C}_{\mathcal{L},\text{myopic}}$  であり、従来の遅延のない通信を行う分散制御方式の言語のクラスを  $\mathcal{C}_{\mathcal{L},\text{DELAY}_0}$  とする。 $\mathcal{C}_{\mathcal{L},\text{DELAY}_1}$  は、それぞれの言語のクラスに対してそれぞれ、

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\subseteq \mathcal{C}_{\mathcal{L},\text{nocomunication}}$$

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\supseteq \mathcal{C}_{\mathcal{L},\text{nocomunication}}$$

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\subseteq \mathcal{C}_{\mathcal{L},\text{myopic}}$$

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\supseteq \mathcal{C}_{\mathcal{L},\text{myopic}}$$

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\subseteq \mathcal{C}_{\mathcal{L},\text{DELAY}_0}$$

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_1} \not\supseteq \mathcal{C}_{\mathcal{L},\text{DELAY}_0}$$

となることを示した。また、 $\text{DELAY}_1$  において、常にイベントの観測と同時に通信が送られてくる場合は、観測もしくは受信したイベントの間には  $\Sigma_{u_0}$  が含まれず、 $\text{DELAY}_1$  は通信の遅延の制約があるにも関わらず、 $\text{DELAY}_0$  よりも能力が高いケースが存在する。 $\text{DELAY}_{\leq 1}$  の推定は、 $\text{DELAY}_1$  と  $\text{DELAY}_0$  の推定の和集合で得られることを示し、さらに、 $\text{DELAY}_1$  の言語のクラスは  $\text{DELAY}_{\leq 1}$  の言語のクラスに真に含まれる。すなわち、

$$\mathcal{C}_{\mathcal{L},\text{DELAY}_{\leq 1}} \subsetneq \mathcal{C}_{\mathcal{L},\text{DELAY}_1}$$

を示した。遅延が 2 以上の整数  $K$  の遅延の場合には、受信したイベントと受信した瞬間に発生したイベント間には  $K - 1$  個のイベントを挟むため、軌跡の推定が指数的に増加することを明らかにした。

$DELAY_0$  と  $DELAY_1$  との比較から、 $DELAY_1$  には、 $DELAY_0$  言語クラスに含まない要素が存在し、また  $DELAY_0$  に  $DELAY_1$  言語クラスに含まない要素も存在する。今後の課題としては、 $DELAY_N$  と  $DELAY_{N+1}$  との比較を行った場合に  $DELAY_0$  と  $DELAY_1$  の関係と同様の事が言えるか検討する必要がある。また、 $DELAY_{\leq 1}$  と  $DELAY_1$  との関係も、 $DELAY_{\leq N}$  と  $DELAY_N$  としたときに成り立つかの検討が今後の課題である。

# 謝辞

最後に本研究を進めるにあたり、指導教官であり、北陸先端科学技術大学院大学情報科学研究科の平石 邦彦助教授には、終始様々な助言、ご指導を頂きました。また、同研究室の宋 少秋助手、高島 康裕助手にも終始様々な助言、ご指導を頂きました。

ここに深く感謝の意を表します。

## 参考文献

- [1] P.J.Ramadge and W.M.Wonham, *The control of discrete event systems*, Proc. IEEE: Special Issue on Discrete Event Systems, 77 (1989), pp. 81-98.
  
- [2] 潮 俊光, “離散事象システムにおける制御問題とスーパーバイザ”, システム/制御/情報, Vol.34, No.9, pp.531-538, 1990
  
- [3] J.H.van Schuppen, *Decentralized supervisory control with information structures*, Proc. International Workshop on Discrete Event Systems (Cagliari, Italy), August 1998, pp.36-41.
  
- [4] George Barrett and Stéphane Lafortune, *Decentralized Supervisory Control with Communication*, The University of Michigan, System science and engineering division department of Electrical engineering and Computer science, Report No.CGR-98-12, October 1998.