システム制御情報学会論文誌, Vol. 16, No. 3, pp. 118–124, 2003

論　文

強化学習を用いた離散事象システムのスーパバイザ制御*

山崎　達志†・潮　俊光†

Supervisory Control of Discrete Event Systems based on a Reinforcement Learning*

Tatsushi YAMASAKI† and Toshimitsu USHIO†

This paper proposes a synthesis method of a supervisor based on a reinforcement learning. In discrete event systems, a supervisor controls disabling of controllable events to satisfy control specifications given by formal languages. However a precise description of the specifications is needed to construct the supervisor. In the proposed method, the specifications are given by rewards, and the optimal supervisor is derived under uncertain environments by considering rewards for occurrence of events and control patterns through learning. By computer simulation, we examine an efficiency of the proposed method.

1. はじめに

事象の非同期的，並行的な生起により状態が遷移する離散事象システム（DES; Discrete Event System）は，データベースシステム，通信システム，生産システム，オペレーティングシステムなど様々なシステムに見ることができ，離散事象システムに対する制御手法の研究が活発に行われている [1,2].

離散事象システムに対する論理的な制御法として，RamadgeとWonhamによって提案されたスーパバイザ制御 [3] がある．これは，制御仕様を満足する事象列のみを生起させるように，システムの可制御な事象の生起を許容，禁止する制御器（スーパバイザ）を構築してシステムを制御する．生起を許容する事象の集合を制御パターンという．

一方，近年注目を集めている学習手法のひとつに強化学習がある．強化学習は，環境から受け取る報酬をもとに，学習者たるエージェントが試行錯誤を通じてより良い制御規則を学習していく枠組みを提供している [4–6].

スーパバイザ制御では，制御仕様を満足するという制約のもとに，システムの生成言語が最大になるという意味で最適な制御パターンを指定する．これは，できる限りシステム本来の挙動を制限せずに多くの事象の生起を許容することが望ましいという考えに基づいている．このとき，あらかじめ制御対象および制御仕様が，厳密に形式言語あるいはオートマトンで記述されている必要がある．しかし多くの場合，“こうなって欲しい”という自然言語的に要求される仕様から直接に精確な形式言語での表現を求めるのは簡単ではない．また，スーパバイザ制御は論理的な制御の枠組みであるため，これにより望ましくない状態に至らないように制御をすることはできるが，事象の生起や禁止のためのコストは考慮していない．一方，多くの強化学習では，学習者の受け取る割引報酬の総和の期待値が最大となるという意味で最適な行動政策を，試行錯誤を通じて学習していく．

本論文ではスーパバイザの設計を強化学習を用いて行うことにより，事象の生起および禁止のコストを考慮したスーパバイザを獲得する手法を提案する．制御仕様に報酬という比較的与えやすい指標を導入し，制御仕様の詳細は学習によって獲得することにより，スーパバイザの設計の簡単化，自動化を図る．学習は，どの事象を生起させるかではなく，どの事象を生起させてもよいかという制御パターンの与え方に対して行う．ここでは，制御パターンを小さく制限するほど，制御対象の挙動が制限され，仕様を満たしやすくなるが，事象の生起の禁止によるコストもかかるため，過度の制限もまた望ましく

* 原稿受付　2002年6月17日

† 大阪大学　大学院　基礎工学研究科　Graduate School of Engineering Science, Osaka University; 1-3 Machikaneyama-cho, Toyonaka city, Osaka 560-8531, JAPAN

Key Words: discrete event systems, supervisory control, reinforcement learning, Markov decision process, optimal control.

ないという状況を想定している．そして，仕様および事象の生起と禁止に伴うコストを報酬を通じて考慮した中で生成言語を最大とするスーパバイザを構成することを目指す．さらにいくつかの仮定を導入することにより複数のQ値を同時更新でき，学習速度が向上することを示す．

鈴木らは，スーパバイザが構成されたシステムに対し，強化学習によりコストが最小となるタスク系列を学習するという手法を提案している[9]．また，あらかじめ構成されたスーパバイザから，事象の生起と禁止に関してのコストの総和を最小化するように生起事象を指定する最適スーパバイザ制御も提案されている[7,8]．これらに対し，本論文ではスーパバイザそのものの構成の段階から強化学習を用いることで，制御仕様とコストの両面を考慮しての最適な制御パターンを求めている．

以下，まず**2.**では準備としてスーパバイザ制御と強化学習について簡単に述べる．**3.**ではシステムの数理モデルを示し，スーパバイザ構成のアルゴリズムを提案する．**4.**で計算機実験により提案手法の有効性を示し，最後に**5.**でまとめと今後の課題を述べる．

2. 準備

2.1 スーパバイザ制御

離散事象システムにおけるスーパバイザは，FIFO処理やデッドロックの回避など，論理的に与えられた制御仕様を満たすように制御対象を制御する．ここでの制御とは，どの事象の生起を禁止，許容するかということである．このとき，事象にはスーパバイザが生起を禁止できる可制御事象と，できない非可制御事象があると仮定する．たとえば，タスクの開始要求や，自身の移動方向の決定は可制御な事象である．また，いつタスクが終了するかや割り込みの要求などは非可制御な事象の例といえる．

制御対象である離散事象システムとスーパバイザの関係は，Fig. 1のようにとらえることができる．スーパバイザは，制御パターンを選んで制御対象に伝える．このとき，非可制御事象は生起を禁止できないため，必ず制御パターンの中に含まれる．制御対象は，制御パターンの中から事象を生起させ次の状態へ移る．制御パターンの中からの生起事象の選択はスーパバイザ側から干渉はできない．スーパバイザは，制御対象で生起した事象を観測し，次の状態へ移る．以降，このサイクルが繰り返される．制御仕様を満たした上で，できる限り多くの事象の生起を許容するスーパバイザを構成する，多項式時間アルゴリズムが示されている[3]．また，このときシステムが生成する言語を最大可制御言語とよぶ．しかし，このアルゴリズムを適用するためには制御仕様を形式言語で厳密に記述する必要がある．これには大きな計算量を必要とし，たとえば複数個の仕様を同時に満足するスーパバイザを構成するために必要な計算量は$\mathcal{NP}$困難

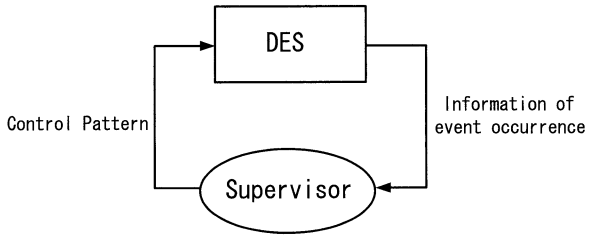


Fig. 1 The Discrete Event System (DES) controlled by the supervisor

となる[11]．すなわち，制御仕様を求めるところに多くの労力が必要となる．

2.2 強化学習

強化学習は，環境から受け取る報酬をもとに学習者が，より良い行動指針を見つけ出す学習手法である[4–6]．試行錯誤を通じた学習を行うため，環境に不確実さがある場合や，未知のパラメータがある場合にも，制御規則を自動的に獲得することができる．また，環境の変化に対しても自律的に対応することができる．

代表的な環境同定型の強化学習のアルゴリズムである，Q-learning[10]について述べる．Q-learningでは，Q値とよばれる，状態と行動の組に対する評価値を，経験をもとに更新していくことにより学習を進める．学習者が状態xで行動aを選択し，状態x'に遷移して，報酬rを受け取ったとするとき，以下の式によって，Q値を更新する．

$$Q(x,a)\leftarrow Q(x,a)+\alpha\big[r+\gamma\max_{a'}Q(x',a')-Q(x,a)\big] \tag{1}$$

ここで，$Q(x,a)$は状態xで行動aを行ったときの期待収益（以後に獲得する割引報酬の総和の期待値）の推定値，αは学習率$(0<\alpha\leq 1)$，γは報酬の割引率$(0\leq\gamma<1)$である．各状態で，期待収益が最大となるように，最大のQ値を持つ行動を選択していくことが，その時点で最適な行動政策となる．

$\alpha_k(x,a)$を，状態xで行動aがk回目に選択されたときの学習率とする．マルコフ環境下では$\alpha_k(x,a)$について，

$$\sum_{k=1}^{\infty}\alpha_k(x,a)=\infty \quad \text{かつ} \quad \sum_{k=1}^{\infty}\alpha_k(x,a)^2<\infty \tag{2}$$

の条件を満たすとき，すべての行動に対して十分な回数の学習を行うことにより，$Q(x,a)$は確率1で真値に収束する[10]．

3. スーパバイザの構成

3.1 システムの数理モデル

最初に，Fig. 1のスーパバイザで制御されるシステムの数理モデルを考える．ここでいうシステムは制御対象である離散事象システムおよびスーパバイザの全体を

含んでいる．ここではシステムが有限マルコフ決定過程（MDP; Markov Decision Process）であると仮定する．一般には，対象とする問題において必ずしもマルコフ性の仮定が成り立つとはいえないが，そのような場合でも，問題によっては近似的にマルコフ性が成り立つと考えてMDPのもとで構築された強化学習の手法を適用することができる．なお，強化学習の一般的な枠組みからすれば，スーパバイザがエージェント，離散事象システムが環境に相当する．

このとき，以下のBellman最適方程式が成り立つ．

$$Q^*(x,\pi)=\sum_{x'\in \mathrm{X}}\mathcal{P}(x,\pi,x')\left[\mathcal{R}(x,\pi,x')+\gamma\max_{\pi'\in\Pi(x')}Q^*(x',\pi')\right] \quad (3)$$

各記号の意味は，以下の通りである．

- X：制御対象である離散事象システム（DES）の状態の集合．なお，本論文ではDESの状態および，生起する事象が，スーパバイザ側から完全に観測できる状況のみを扱う．
- Σ：事象の集合．
- Σ^{c}：可制御な事象の集合．
- Σ^{u}：非可制御な事象の集合．
  （ここで，$\Sigma=\Sigma^{\mathrm{c}}\cup\Sigma^{\mathrm{u}}$, $\Sigma^{\mathrm{c}}\cap\Sigma^{\mathrm{u}}=\phi$.）
- $\mathrm{F}(x)$：状態$x\in\mathrm{X}$において生起可能な事象の集合．
- $\Pi(x)$：状態$x\in\mathrm{X}$における制御パターンの集合．各制御パターン$\pi\in\Pi(x)$は，状態xで生起を許容する事象の集合である．(ただし，$\forall\pi\in\Pi(x)$について，$\mathrm{F}(x)\cap\Sigma^{\mathrm{u}}\subseteq\pi\subseteq\mathrm{F}(x)\subseteq\Sigma$を満たす．すなわち，生起可能な非可制御事象がある場合，制御パターンの中に，これは必ず含まれる．よって，各状態における制御パターンの数は最大で$2^{|\mathrm{F}(x)\cap\Sigma^{\mathrm{c}}|}$個になる．)
- $\mathcal{P}(x,\pi,x')$：状態$x\in\mathrm{X}$で制御パターン$\pi\in\Pi(x)$を選択したときに，状態$x'\in\mathrm{X}$になる確率．
- $Q^*(x,\pi)$：状態$x\in\mathrm{X}$で制御パターン$\pi\in\Pi(x)$を選択し，以後は，各状態で最大のQ値を持つ制御パターンを選択するときの期待収益．
- $\mathcal{R}(x,\pi,x')$：状態$x\in\mathrm{X}$で制御パターン$\pi\in\Pi(x)$を選択し，状態$x'\in\mathrm{X}$に遷移するときに受け取る報酬の期待値．
- γ：報酬の割引率$(0\leq\gamma<1)$.

ここで，制御対象はスーパバイザによって与えられた制御パターンの中から生起事象を選択することから，

$$\mathcal{P}(x,\pi,x')=\sum_{\sigma\in\pi}\mathcal{P}_1(x,\pi,\sigma)\mathcal{P}_2(x,\sigma,x')$$

が成り立つ．ただし，

- $\mathcal{P}_1(x,\pi,\sigma)$は，状態$x\in\mathrm{X}$で，スーパバイザが制御パターン$\pi\in\Pi(x)$を選択したとき，事象$\sigma\in\pi$が制御対象によって選択される確率，
- $\mathcal{P}_2(x,\sigma,x')$は，状態$x\in\mathrm{X}$で，事象$\sigma\in\mathrm{F}(x)$が生起したとき，状態$x'\in\mathrm{X}$に遷移する確率，

である．

3.2 仮定

提案手法においては，制御対象に対して以下の二つの仮定を設ける．

(1) 各状態$x\in\mathrm{X}$について，事象の選ばれやすさを表すパラメータとして，$\eta^*(x,\sigma)$を導入する．このとき，

$$\mathcal{P}_1(x,\pi,\sigma)=\frac{\eta^*(x,\sigma)}{\sum_{\sigma'\in\pi}\eta^*(x,\sigma')} \quad (4)$$

$$\forall\sigma\in\mathrm{F}(x)\ ,\ \eta^*(x,\sigma)>0\ ,\ \sum_{\sigma\in\mathrm{F}(x)}\eta^*(x,\sigma)=1$$

の関係が成り立っているとする．すなわち，スーパバイザが選択した制御パターンの中から制御対象は生起事象を選択するが，このとき事象を選択する比率は，与えられた制御パターンに依存せず一定である．ただし，η^*の真値をスーパバイザは知らない．

(2) 報酬$\mathcal{R}(x,\pi,x')$について，

$$\mathcal{R}(x,\pi,x')=\mathcal{R}_1(x,\pi)+\mathcal{R}_2(x,\sigma,x') \quad (5)$$

という構造をもつとする．ここで，$\mathcal{R}_1$，$\mathcal{R}_2$の意味は以下の通りである．

- $\mathcal{R}_1(x,\pi)$：状態xで制御パターンπを選んだことによる報酬の期待値．制御パターンの与え方に依存して決定される．直観的には制御パターンに含まれない事象を禁止したことに伴うコストを表している．
- $\mathcal{R}_2(x,\sigma,x')$：状態$x$で事象$\sigma$が生起され，状態$x'$に遷移したときの報酬の期待値．直観的には事象の生起に伴うコストおよび，タスクの出来不出来に伴うコストを表している．

このとき(3)式は，

$$\begin{aligned}
&Q^*(x,\pi)\\
&=\sum_{x'\in\mathrm{X}}\left(\sum_{\sigma\in\pi}\frac{\eta^*(x,\sigma)}{\sum_{\sigma'\in\pi}\eta^*(x,\sigma')}\mathcal{P}_2(x,\sigma,x')\right)\\
&\quad\left[\mathcal{R}_1(x,\pi)+\mathcal{R}_2(x,\sigma,x')+\gamma\max_{\pi'\in\Pi(x')}Q^*(x',\pi')\right]\\
&=\mathcal{R}_1(x,\pi)+\sum_{\sigma\in\pi}\frac{\eta^*(x,\sigma)}{\sum_{\sigma'\in\pi}\eta^*(x,\sigma')}\sum_{x'\in\mathrm{X}}\mathcal{P}_2(x,\sigma,x')\\
&\quad\left[\mathcal{R}_2(x,\sigma,x')+\gamma\max_{\pi'\in\Pi(x')}Q^*(x',\pi')\right]\\
&=\mathcal{R}_1(x,\pi)+\sum_{\sigma\in\pi}\frac{\eta^*(x,\sigma)}{\sum_{\sigma'\in\pi}\eta^*(x,\sigma')}T^*(x,\sigma) \quad (6)
\end{aligned}$$

となる．ここで，$T^*(x,\sigma)$は，状態xで事象σが生起し，以後は，各状態で最大のQ値を持つ制御パターンを選択

するときの期待収益を表しており，

$$T^*(x,\sigma)=\sum_{x'\in X}\mathcal{P}_2(x,\sigma,x')\left[\mathcal{R}_2(x,\sigma,x')+\gamma\max_{\pi'\in\Pi(x')}Q^*(x',\pi')\right] \quad (7)$$

で定義される．$T^*(x,\sigma)$ の中には状態 x で制御パターン π を選択したことによる報酬は含まれていない．

3.3 提案アルゴリズム

本論文で提案する強化学習を用いた離散事象システムのスーパバイザ制御の概念図を Fig. 2 に示す．学習アルゴリズムを Fig. 3 に示す．Fig. 3 において，エピソード (episode) は，初期状態 (initial state) から始まり，終端状態 (terminal state) で終わるひと続きの系列を表す．学習の目的は，どの事象を生起させるかを学習することではなく，制御パターンの与え方を学習することにある．

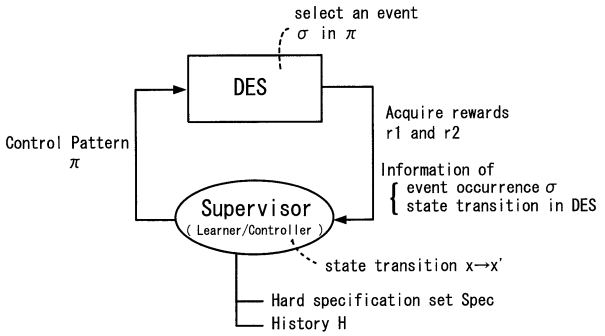


Fig. 2 The Discrete Event System (DES) controlled by the learning supervisor

ある時刻において状態が $x\in X$ であるとする．ここで，学習者たるスーパバイザは制御パターン $\pi\in\Pi(x)$ として，生起を許容する事象の集合を制御対象に提示する．これにより，DES の状態に対して制御パターンを決定するという，状態フィードバック制御となる．今回，制御パターンの選択には，ϵ-greedy 選択を用いるとした．これは，確率 $1-\epsilon$ で最大の Q 値を持つ制御パターンを，確率 ϵ でランダムにいずれかの制御パターンを選択する．

制御対象の側では，生起させる事象 $\sigma\in\pi$ を選択する．この選択は，学習者であるスーパバイザではなく制御対象側が π の中から選択するが，前節仮定 (1) にしたがった選択であるとする．制御対象での事象の生起により，状態 x が x' に遷移し，報酬 r_1，r_2 を獲得する．ただし，r_1 は制御パターン π に対する報酬，r_2 は生起事象 σ に対する報酬である．提案アルゴリズムは Q-learning の更新式に基づいているが，前節の仮定のもとで学習効率の改善を図っている．

(6) 式より，Q^* は，T^*, $\mathcal{R}_1$, η^* を用いて求めることができる．そこで，提案アルゴリズムでは，

$$T(x,\sigma)\leftarrow T(x,\sigma)+\alpha[r_2+\gamma\max_{\pi'\in\Pi(x')}Q(x',\pi')-T(x,\sigma)] \quad (8)$$

$$R_1(x,\pi)\leftarrow R_1(x,\pi)+\beta[r_1-R_1(x,\pi)] \quad (9)$$

$$\text{For all}\quad \sigma'\in\pi$$

$$\eta(x,\sigma')\leftarrow\begin{cases}(1-\delta)\,\eta(x,\sigma') & (\text{ if } \sigma'\neq\sigma)\\ \eta(x,\sigma')+\delta\left[\sum_{\sigma''\in\pi}\eta(x,\sigma'')-\eta(x,\sigma')\right] & (\text{ if } \sigma'=\sigma)\end{cases} \quad (10)$$

として，T, R_1, η を推定する．α, β, δ は学習率である．これらを用いて，実際に選択した制御パターン π のみではなく，π に含まれる事象を含む全制御パターンに対し同時に Q 値の更新を行うことができる．すなわち，

$$\text{For all}\quad \pi'\in\Pi(x)\ s.t.\ \pi'\cap\pi\neq\emptyset$$

$$Q(x,\pi')\leftarrow R_1(x,\pi')+\sum_{\sigma''\in\pi'}\frac{\eta(x,\sigma'')}{\sum_{\sigma'''\in\pi'}\eta(x,\sigma''')}T(x,\sigma'') \quad (11)$$

として間接的に Q 値を推定する．複数の Q 値の同時更新による学習効率の向上が期待できる．また，与えられた制御パターンの中から，いずれかの生起事象が選択されるという構成になっているため，単に制御パターンを行動ととらえて学習しただけでは受け取る報酬の分散が大きくなり学習性能が劣化すると考えられるが，提案手法では，その影響を抑えることができる．

今回，制御パターンの選択には ϵ-greedy 選択を用いるとしており，その場合は，T, R_1, η を保持しておけば，Q 値については，最大の Q 値の情報のみを保持することで，メモリ効率を改善させることができる．また，制御パターンの選択に Boltzman 選択のように，すべての Q 値を必要とする手法を用いた場合には，計算時間を優先するならば全 Q 値を保持する方式が，メモリ優先の場合には，必要になった時点で毎回 Q 値を計算する方式が考えられる．

エピソードの各ステップの最後に，現在の状態が最低限の要求仕様として定める状態の集合 Spec を満たしているかどうかをチェックする．Spec は，報酬という形で与えられる仕様とは別のものである．もしも満たされていない場合は，最後に生起した可制御事象 $\sigma^c\in\Sigma^c$ と，そのときの状態 w を特定し，σ^c を w における生起可能な事象の集合 $F(w)$ から削除する．また，状態 w における η の和が 1 となるように，各 η の値を更新し，Q 値の再計算を行う．これにより，強制的に Spec を満たさない事象列を取り除くという，一種の枝刈りを行う．そのために，各エピソードでは，生起した事象列および状態の列を履歴 H として記憶しておく．

提案するアルゴリズムは，ある厳密な仕様 Spec のも

1. Initialize $T(x,\sigma)$,$R_1(x,\pi)$, and $\eta(x,\sigma)$ at each state.
2. Calculate the initial Q value at each state by

$$Q(x,\pi) \leftarrow R_1(x,\pi)+\sum_{\sigma\in\pi}\frac{\eta(x,\sigma)}{\sum_{\sigma'\in\pi}\eta(x,\sigma')}T(x,\sigma)$$

3. Repeat (for each episode):
   (a) Clear history H.
   (b) $x \leftarrow$ *initial state*.
   (c) Repeat until x is *terminal state* (for each step of episode) :
      i. Select a control pattern $\pi \in \Pi(x)$ based on the Q value by the supervisor.
      ii. Observe event occurrence $\sigma \in \pi$ and state transition in DES.
      iii. Acquire rewards r_1 and r_2.
      iv. Make transition $x \xrightarrow{\sigma} x' \in$ X.
      v. Add (x,σ) to history H.
      vi. Update $T(x,\sigma)$, $\mathcal{R}_1(x,\pi)$, and $\eta(x,\sigma')$:

$$T(x,\sigma) \leftarrow T(x,\sigma)+\alpha[r_2 +\gamma \max_{\pi'\in\Pi(x')} Q(x',\pi')-T(x,\sigma)]$$

$$R_1(x,\pi) \leftarrow R_1(x,\pi)+\beta[r_1-R_1(x,\pi)]$$

For all $\sigma' \in \pi$

$$\eta(x,\sigma') \leftarrow \begin{cases} (1-\delta)\eta(x,\sigma') & (\text{if } \sigma' \neq \sigma) \\ \eta(x,\sigma')+\delta\left[\sum_{\sigma''\in\pi}\eta(x,\sigma'')-\eta(x,\sigma')\right] & (\text{if } \sigma' = \sigma) \end{cases}$$

      vii. Update the Q values

For all $\pi' \in \Pi(x)$ *s.t.* $\pi' \cap \pi \neq \emptyset$

$$Q(x,\pi') \leftarrow R_1(x,\pi') + \sum_{\sigma''\in\pi'}\frac{\eta(x,\sigma'')}{\sum_{\sigma'''\in\pi'}\eta(x,\sigma''')}T(x,\sigma'')$$

      viii. If $x' \notin$ Spec
         A. Search the latest controllable event $\sigma^c \in \Sigma^c$ and the corresponding state $w \in$ X from the history H.
         B. Remove σ^c from the feasible event set F(w).
         C. Normalize $\eta(w,\sigma')$ so as to satisfy $\sum_{\sigma'\in F(w)}\eta(w,\sigma')=1$ and update the Q values at the state w
      ix. $x \leftarrow x'$

Fig. 3 The proposed algorithm for construction of a supervisor

とで，期待収益の点で最適となる制御パターンを学習していくことを通じ，制御仕様とコストを考慮した中で，生成言語を最大とするスーパバイザを獲得するアルゴリズムであるといえる．

強化学習との関連でとらえると，制御パターンのそれぞれを行動と考えQ-learningを用いて最適な制御パターンを求める方法に対し，提案手法では仮定を利用することによって，複数のQ値の同時更新を行い，学習速度の向上を図っている．その上で，枝刈りの仕組みを付け足したものと位置付けることができる．あるいは，$T(x,\sigma)$を従来考えていたQ値ととらえると，スーパバイザ制御を導入するために，R_1, ηというパラメータを用いて，従来のQ-learningの枠組みを拡張したものととらえることもできる．また，提案手法では，最終的にはQ値を推定しているので，他のQ値を学習させるタイプの強化学習法と親和性が良く，それらと組み合わせての利用も考えられる．

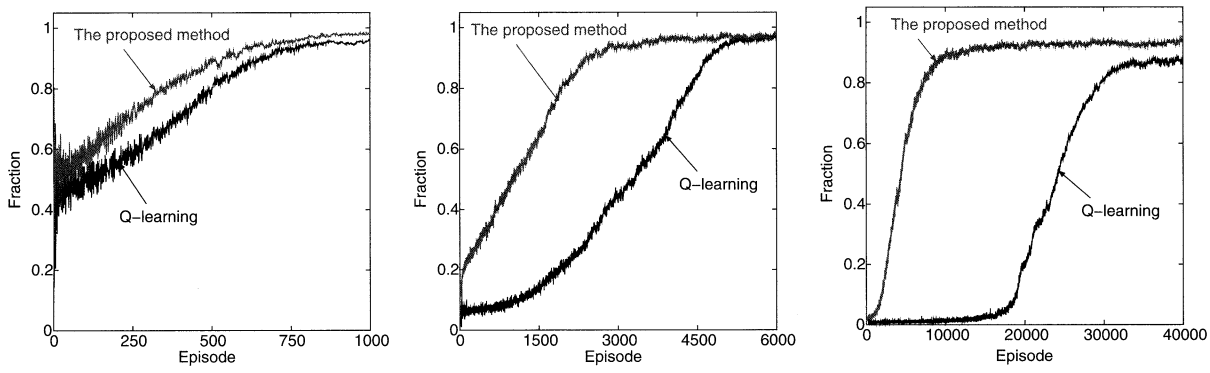
4. 実験

2種類の例題に対し，計算機実験を行った．なお，今回は，報酬に基づく学習の部分の評価に重点を置いており，Specによる枝刈りは用いていない．

4.1 n本腕バンディット問題

最初の例題として，n本腕バンディット問題を考える[5]．制御対象はn種類の行動から一つを選択し報酬を受け取る．報酬は，行動ごとに定められた平均値を持つ正規分布にしたがって返される．通常のn本腕バンディット問題では，期待報酬が最大となる行動をn個の行動選択肢の中から学習することを目的とするが，今回，すべての行動を可制御とし，2^n-1個の制御パターンの与え方の中から学習する．

Fig. 4は，それぞれ$n=2$, $n=5$, $n=8$の場合について提案手法と通常のQ-learningとで性能の差を比較したものである．これは制御パターン数で考えると，それぞれ3個，32個，255個となる．グラフの横軸はエピソード数を，縦軸は期待収益が最大となる制御パターンが求まった割合を示している．本例題では，1エピソードは1回の制御パターンの選択で終了する．それぞれについて10個の異なる問題を生成し，さらに各問題について100回の学習を行った結果を平均した．問題の生成にあたっては，各事象の生起に対する報酬は0から-100の範囲でランダムに設定した．制御パターンの与え方に対する報酬は，各事象の生起の禁止に対する報酬を0から-20の範囲でランダムに設定し，制御パターンで生起を禁止された事象に関してこの和を取ることで決定した．各$\eta*$は平均$1/n$，分散1の正規分布にしたがって生成した乱数をもとに，$\sum\eta^*=1$になるように調整した．ただし，最低でも各η^*が0.05以上となるようにしている．また，受け取る報酬は，真値を平均とし分散を0.1とした正規分布にしたがって返すとした．いずれの学習率も0.01に固定し，Q値の初期値は0とした．制御パターンの選択には，$\epsilon=0.1$のϵ-greedy選択を用いた．グラフより，提案手法はQ-learningよりも学習速度が速く，制御パターン数が多くなるほど，その差が顕著になっていることがわかる．この結果から，制御パターン数が少ない場合は計算が少なくてすむ通常のQ-learningを用い，



(a) Case $n=2$ (b) Case $n=5$ (c) Case $n=8$

Fig. 4 Relation between the number of episodes and the fraction that the supervisor found the optimal control pattern in n-armed bandit problem

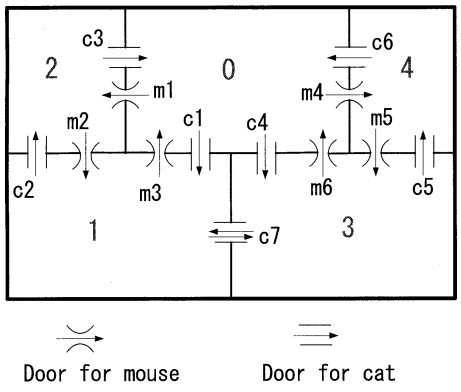


Fig. 5 Maze for cat and mouse

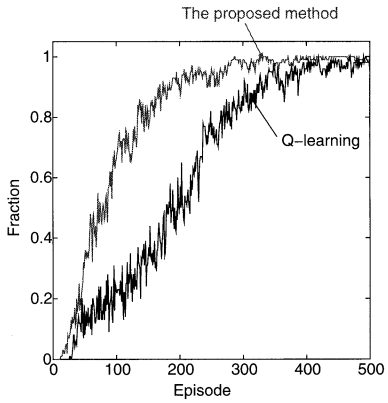


Fig. 6 Relation between the number of episodes and the fraction that the supervisor found the optimal control pattern in the maze for cat and mouse problem

制御パターンが多いところでは提案手法を用いて学習するといった使い分けも考えられる．なお，最終的に確率1で最適解を学習できていないのは，学習率が(2)式の条件を満たしておらず，生成した問題のいくつかで最適解と準最適解との差が僅少であるためと考えられる．実際，学習率を徐々に減らしていった場合，時間はかかるものの両手法とも最適解を求めることができた．

4.2 猫とねずみの迷路の問題

前節の例題では，1回の行動で一つのエピソードが終了するので，報酬の時間遅れという要素がなく，また，すべての事象が可制御であった．ここでは，スーパバイザ制御の例題としてしばしば用いられる猫とねずみの迷路の問題[3]に対して，提案手法を適用する．Fig. 5で示す，ドアで区切られた五つの部屋がある．猫とねずみが存在し，矢印の方向にのみ移動が許された専用のドアが用意されている．Fig. 5において，c1～c7が猫用のドア，m1～m6がねずみ用のドアである．部屋1-3間のドアはc7のみ非可制御なドアで，他は可制御なドアである．このとき，猫とねずみが同じ部屋に入ることのないようにドアの開閉を制御することを学習させたい．初期状態では，猫は部屋2に，ねずみは部屋4にいるとし，報酬として，ドアを閉じるにはドア一つにつき平均-1，分散0.1の正規分布にしたがうコストがかかるとした．また，ひとつのエピソードは20回移動を終えた場合か制御仕様を満たさなくなった場合，すなわち，猫とねずみが同じ部屋になった場合（-100の報酬を与える）に終了するとした．学習率はいずれも0.1を用い，$\gamma=0.9$とした．制御パターンの選択には，$\epsilon=0.1$のϵ-greedy選択を用いた．

Fig. 6は，横軸にエピソードを，縦軸に文献[3]で示されたスーパバイザが求まった割合を示している．結果は100回の学習の平均である．最終的には両手法ともこのスーパバイザが求まっているが，提案手法の方がQ-learningよりも速く，求まる確率が1に収束している．また，Fig. 7は，学習の結果得られた制御パターンを状態遷移図で表したものである．円内の数値の10の位が猫のいる部屋を，1の位がねずみのいる部屋を表しており，遷移の矢印が，各状態での生起を許可する事象を示している．猫とねずみが同じ部屋に居ないという論理的な制御仕様を満たす制御パターンを提示するスーパバイザが学習を通じて獲得されている．

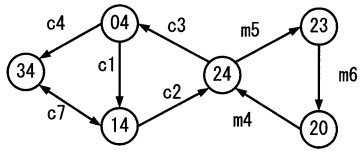


Fig. 7 The representation of the learned control pattern by the transition diagram

5. おわりに

本論文ではスーパバイザの制御機構の特性を利用した，Q-learningに基づくスーパバイザの構成法を提案した．スーパバイザ制御を応用するとき，精確に制御仕様を記述することが重要な問題点となっている．提案手法では，制御仕様の詳細は報酬を元に学習を通じて獲得するとした．スーパバイザの設計を強化学習を用いて行うことにより，環境が不確かであったり，変化する場合に対してスーパバイザ制御を適用するための一つの方法論を示した．

今後の課題としては，より詳細なアルゴリズムの検討と，性能向上が挙げられる．特に，事象の数が増えると，それに伴い制御パターンの数が急激に増大するので，さらに効率的な学習方法の提案が必要である．また，事象の生起について部分観測な場合など，より複雑な状況を考慮したアルゴリズムの開発も必要である．最大可制御言語を生成するスーパバイザを提案手法によって求められるかを理論的に明らかにすることも重要な課題である．

参考文献

[1] 児玉，潮：離散事象システム理論—来し方と現在；システム/制御/情報，Vol. 42, No. 8, pp. 415–420 (1998)

[2] C. G. Cassandras and S. Lafortune: *Introduction to Discrete Event Systems*, Kluwer Academic Pub. (1999)

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

[4] D. P. Bertsekas and J. N. Tsitsiklis: *Neuro-Dynamic Programming*, Athena Scientific (1996)

[5] R. S. Sutton and A. G. Barto: *Reinforcement Learning*, MIT Press (1998)

[6] 木村，宮崎，小林：強化学習システムの設計指針；計測と制御，Vol. 38, No. 10, pp. 618–623 (1999)

[7] R. Sengupta and S. Lafortune: An optimal control theory for discrete event systems; *SIAM J. Control Optim.*, Vol. 36, No. 2, pp. 488–541 (1998)

[8] R. Kumar and V. K. Garg: Optimal supervisory control of discrete event dynamical systems; *SIAM J. Control Optim.*, Vol. 33, No. 2, pp. 419–439 (1995)

[9] 鈴木，残間，稲葉，大熊：組立/分解作業手順の学習制御 - 離散事象システム論からのアプローチ - ；日本ロボット学会誌，Vol. 14, No. 7, pp. 1042–1052 (1996)

[10] C. J. C. H. Watkins and P. Dayan: Q-learning; *Machine Learning*, Vol. 8, pp. 279–292 (1992)

[11] P. Gohari and W. M. Wonham: On the complexity of supervisory control design in the RW framework; *IEEE Trans. Syst., Man, Cybern. B*, Vol. 30, No. 5, pp. 643–652 (2000)