컨벡스 클러스터링 기법을 이용한 영역기반 Q-learning

論 文
46~6~21

Region-based Q-learning using Convex Clustering Approach

金 載 顯* · 徐 一 弘**
(Jae-Hyun Kim · Il-Hong Suh)

Abstract - In this paper, for a continuous state space applications, a novel method of Q-learning is proposed, where the method incorporates a region-based reward assignment being used to solve structural credit assignment problem and a convex clustering approach to find a region with the same reward attribution property. Our learning method can estimate a current Q-value of an arbitrarily given state by using effect functions, and has the ability to learn its actions similar to that of Q-learning. Thus, our method enables robots to move smoothly in a real environment. To show the validity of our method, the proposed Q-learning method is compared with conventional Q-learning method through a simple two dimensional free space navigation problem, and visual tracking simulation results involving an 2-DOF SCARA robot are also presented

Key Words : Q-leanring, region-based, convex clustering

1. 서 론

강화 학습은 로보트가 미지의 환경에서 적절히 행동하는 것을 경험을 통해 학습하기 위해 개발된 방법으로 환경에 대한 충분한 지식 없이도 주어진 환경에 적절한 행위를 학습할 수 있는 효과적인 방법으로 알려져 있다. 이러한 강화 학습에 대한 대부분의 연구들은 불연속 상태 공간과 불연속 행위 공간을 기반으로 하여 이루어졌다[1-7]. 따라서 불연속 상태 공간으로 모델링 되는 미지의 환경과 상호작용을 통하여 불연속적인 행위들 중 최적의 행위를 학습할 수 있었다. 이러한 방법들 중 하나인 Q-learning은 현재 행위에 대한 평가를 위해 현재의 행위가 최적의 행동책략(policy)을 따른다고 가정할 때 현재 행위에 대한 행위값(Q-value)을 극대화하는 방법으로 이러한 행위값들은 각 상태에서 최적의 행위를 수행할 수 있는 근거가 된다. 이러한 Q-learning을 실질적인 작업에 적용하기 위해서 많은 노력이 진행되어 왔다[8-10]. 예로써, Berenji등에 의해 개발된 Fuzzy Q-learning방법은 Q-value 갱신식에 상황에 대한 제약을 첨가하여 현재의 상황을 반영하고자 하였다[9]. 그러나 이러한 Q-learning을 비롯한 대부분의 강화 학습 방법들은 불연속 상태 공간과 행위 공간을 학습 환경의 모델로 사용하기 때문에 실질적인 로보트 응용 분야에서 이러한 알고리즘을 적용하기에는 다음과 같은 여러 문제점을 극복하여야만 한다. (1)먼저, 너무 많은 기억 공간을 필요로 하고, (2)모든 상태에 대해 학습을 수행해야 하므로 학습 시간이 길며, (3)또한, 출력(행위)이 불연속이라는 제한을 갖게 된다. 이러한 문제점을 해결하기 위해 퍼지의 Q-table을 새로운 퍼지 Rule들로 대체하여 각 Q 값들을 퍼지 추론에 의하여 생성하고, 각 Rule의 후건부 파라메타들을 Steepest descent 방법으로 조정하고자 하는 새로운 시도가 이루어졌다[10]. 그러나 이 경우 초기 Rule들의 생성이 어렵고, Steepest descent 방법을 사용함으로써 국부 극소 점에 빠지는 경우 원하는 목표 상태로의 수렴을 보장 할 수 없다는 문제점이 있다. 특히, Q-value를 생성하는 각 rule들의 후건부 파라메타들을 조정할 때 현재의 Q-value과 각 rule로부터 생성되는 새로운 Q-value와의 차를 최소화하도록 하는 방법으로 비록 Q-value를 계산할 때 기존의 Q-learning의 Q-value 갱신 공식을 사용할 지라도 Steepest descent 방법으로 갱신된 rule들이 어떠한 통계적인 내용도 포함하지 않게 된다. 따라서 이산 공간에서 정의되었던 Q-learning을 연속 공간으로 확장한 방법이라고 할 수 없다. 본 논문에서는 이산 상태 공간 및 행위 공간 뿐만 아니라, 연속적인 상태 및 행위 공간을 포함하는 새로운 Q-learning방법을 제안하고자 한다.

여러 가지의 강화 학습들을 특성화 하는 기본적인 논제들 중 credit assignment problem은 "*일련의 sensor-action-feedback 으로부터 어떻게 최적의 행위를 배울 것인가*"로 정의되며, 각 강화 학습에서 풀어야 할 기본적인 문제로 대두되었다. 이러한 credit assignment problem중 structural credit assignment problem은 "*현재 받은 reward가 상태 공간 내의 각 상태들에게 어떻게 영향을 끼칠 것인가*"로 정의 된다. 이러한 관점에서, 기존의 Q-learning은 point-based credit assignment 방법이라고 정의 내릴 수 있다. 따라서 이러한 상태 집에 기반으로 보답를 할당하는 Q-learning을 일반화하기 위해 특정한 상태 영역에 보답을 할당하는 영역 기반(Region-based) Q-learning (RQ-learning)을 제안하고자 한다. 제안하고자 하는 방법에서, 현재상태의 보답은 현재상태의 주변 상태로 전파되고, 전파된 각 주변 상태의 보답을 기반으로 주변 상태에서의 Q-value가 기존의 Q-value 갱신 식에 의해 수정되며, 수정된 Q-value에 의해 최적의 행위가 조절된다. 여기서, Q-value의 갱신을 보다 빠르게 수행하며 생성된 행위의 연속성을 보장하기 위해서 삼각형 형태의 Q-value 분포 모델을 사용한다. 또한, 제안된 방법을 사용하는 경우, 학습 횟수가 증가함에 따라 기존의 Q-learning방법에서와 같이, 각 상태에서 최적으로 산정된 행위가 실질적인 최적의 행위로 수렴함을 보이고자 한다.

* 正 會 員 : 漢陽大 大學院 電子工學科 博士課程
** 正 會 員 : 漢陽大 工大 電子工學科 敎授 · 工博
接受日字 : 1997年 1月 29日
最終完了 : 1997年 4月 3日

2. Q-learning 알고리즘

2.1 Q-learning 알고리즘 소개

Q-learning은 현재상태에서 로보트가 수행한 행위를 평가하여 최적의 행위를 수행할 수 있도록 한다. Q-learning에서는 기본적으로 로보트와 환경과의 상호작용을 이산 반복 공정(Discrete Time Cyclic Processes)에서 동작하는 유한 상태를 갖는 두개의 대행자들(환경과 로보트)로 모델링 한다. 이러한 상호작용은 다음과 같다. 먼저 로보트는 환경에 대한 현재상태를 감지하고 적절한 행위를 선택하여 이를 수행한다. 다음으로, 환경은 현재 상태와 수행된 행위에 근거하여 새로운 상태로 전이되고 수행된 행위에 대한 보답을 발생시키며, 이를 로보트에게 되돌려 준다. 이러한 상호작용을 통해 로보트는 각 상태에 대한 적절한 행위를 배우게 된다. 이러한 관계를 정리하여 이론화한 Q-learning 알고리즘은 다음과 같다. 여기서, 로보트는 서로 다른 유한 상태들의 집합 S를 감지할 수 있고, 유한 행위들의 집합 A를 행할 수 있으며, 또한 외부 환경은 목표 상태로(Goal State) 수렴성이 보장되는 Markov Process로써 모델링 할 수 있다고 가정한다.

2.1.1 Q-learning 알고리즘

(1) 초기화

a. 난수 혹은 사전 정보를 이용한 Q-table, $Q(s,a)$ 초기화

b. 초기화된 Q-table를 근거로 Policy f_i 초기화

$$f(s)\leftarrow a \quad \text{such that} \quad Q(s,a)=\max_{b\in A} Q(s,b) \tag{1}$$

여기서 t는 tth iteration을, i는 현재상태를 각각 나타낼때, 는 다음 iteration의 현재 상태 i에서 수행할 행위 a에 대한 행위값을 나타낸다. 또한, A는 현재 정의된 행위 집합을 각각 나타내며, 따라서 f_i는 현재 수립된 최적의 행위계획(policy)을 나타낸다.

c. 여러 파라메타 (γ, α, ρ)의 초기화

(2) 현재 상태를 받아들임 ($s \leftarrow$ 현재상태)

(3) Policy Table로부터 현재상태에 해당하는 행위 a를 수행하거나 ρ만큼의 비율로 임의의 행위를 수행. 여기서 랜덤 행위를 수행하는 것은 최적의 policy를 구하기 위한 필요조건이 된다.

(4) 환경으로부터 현재 수행된 행위에 대한 보답 r를 받음.

(5) 다음 식 (2)를 이용하여 현재 상태에서 수행한 행위값(Q-value) $Q(s,a)$을 갱신한다.

$$Q_i^a(t+1)=\alpha\, Q_i^a(t)+(1-\alpha)(r_1^a+\gamma \max_{b\in A}\{Q_{i+1}^b(t)\}) \tag{2}$$

여기서 $\alpha(0<\alpha<1)$는 학습 속도를 나타내며, γ는 미래 행위에 대한 보답에 대한 감쇠 상수이다.

(6) 식 (1)를 이용하여 Policy f_i 갱신.

(7) 2 단계로 복원.

3. 영역기반(Region-based) Q-learning

2장에서 기술하였듯이, 기존의 Q-learning을 실제 환경에 적용하기 위해서는 너무 많은 기억 공간과 학습 시간이 필요하게 된다. 또한, 기존의 Q-learning은 불연속 상태 및 행위 공간에서 사용되기 때문에 출력되는 행위가 부드럽지 못하다. 이러한 제한을 극복하기 위해, 본 논문에서는 먼저 기존의 Q-learning을 영역 기반으로 보답을 할당하는 영역 기반 Q-learning (Region-based Q-learning)을 개발하였다. 이러한 영역 기반 Q-learning방법은 기존의 현재 상태에만 보답을 할당하는 방법(point-wise Q-learning)을 포함하는 일반화된 방법이라고 할 수 있다. RQ-learning에서는 상태 공간내의 모든 상태에 대해 학습할 필요가 없다. 즉, 단지 미리 설정한 특정한 상태들(주변 상태)에 대해서만 학습을 수행하며 최적의 행위도 이러한 주변 상태로부터 생성하게 된다. 본 장에서는 RQ-learning의 설명을 위해, 주변 상태의 보답을 할당하는 방법 및 이의 수렴성의 증명을 3.1에 기술하였고, 이를 기반으로, 연속 행위 공간에서 학습 속도와 기억 공간을 줄이기 위해 삼각 형태의 Q-value 모델을 이용한 주변 상태들의 Q-value 갱신 방법과 최적 행위 생성 방법을 각각 3.2, 3.3에서 기술하였다.

3.1 현재 상태의 행위값을 통한 주변 상태(neighboring states)의 행위값(Q-value) 결정

N-차 상태 공간을 이루는 각 상태 축들이 l 개의 분해능을 갖는다고 가정하자. 이러한 상태 공간 구조에서 주변 상태 (*neighboring state*)란 그림 1에서와 같이 현재상태가 포함되어있는 hyperbox의 각 꼭지점에 위치한 상태로 정의 하고자 한다.

이때, 임의의 hyperbox내의 임의의 위치에 있을 수 있는 현재 상태 s_i에서 얻은 보답을 r_i 라고 정의하고, 현재 상태의 j번째 주변 상태 $s_{i,j}$ 의 보답을 r_j 라 정의한다. 현재 상태의 보답과 주변 상태로 전파되는 보답과의 관계를 *effect function*, $\mu_{i,j}(s_i,s_{i,j})$으로 정의한다면 현재 상태의 보답으로부터 주변 상태로 전파되는 보답은 식 (3)과 같이 정의 될 수 있다.

$$r_j=\mu_{i,j} r_i \tag{3}$$

그림 1에서 볼 수 있는 것과 같이, 특정 주변 상태의 보답은 $\mu_{i,j}(s_i,s_{i,j})$와 r_i을 곱함으로써 얻을 수 있다. 따라서 최적의 policy를 따를 때 정의되는 $s_{i,j}$에 전달되는 보답의 감쇠 합인 Q-value는 다음 식 (4)와 같이 쓰여질 수 있다.

$$Q_j^{a_1}=\sum_{n=0}^{\infty}\gamma^n \mu_{1+n,j}\, r_{1+n} \tag{4}$$

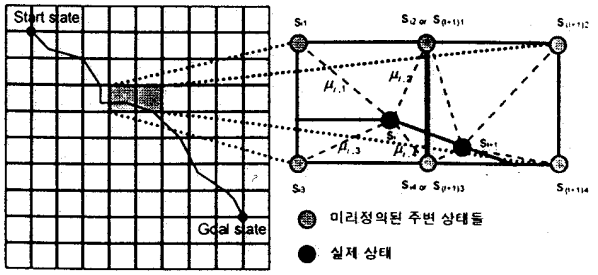


그림 1 hyperbox내의 현재 상태 s_i 의 주변 상태 정의 및 주변 상태 $s_{i,j}$ 로의 영역 기반 보답 할당

Fig. 1 Region-based reward assignment in two state cells

이러한 새롭게 정의된 Q-value에 대해 다음의 *Theorem 1*이 성립한다.

Theorem 1

현재 상태에서, 식 (3)에서 처럼 $r_j = \mu_{imj} r_i$에 의해 정의된 보답을 사용하는 식 (4)의 Q_j^{ai}을 기존의 Q-value 갱신식에 의해 갱신 시키면, iteration이 증가함에 따라 최대의 Q_j^{ai}는 최적 행위로 수렴하게 된다.

Proof

$$Q_j^{ai}(t) = \sum_{n=0}^{\infty} \gamma^n u_{i+n,j} r_{i+n} \tag{5}$$

$$= \mu_{i,j} r_i + \sum_{n=1}^{\infty} \gamma^n \mu_{i+n,j} r_{i+n} \tag{6}$$

$$= \mu_{i,j} r_i + \gamma \sum_{n=1}^{\infty} \gamma^{n-1} \mu_{i+n} r_{i+n} \tag{7}$$

$$= \mu_{i,j} r_i + \gamma \max_{b \in A} \{ Q_j^b(t) \} \tag{8}$$

여기서, $Q_j^{a1}(t+1)$를 나타내는 기존의 Q-learning 갱신식인 식 (2)은 식 (8)을 이용하여 다음 식 (9)로 다시 쓸 수 있다.

$$Q_j^{a1}(t+1) = \alpha Q_j^{a1}(t) + (1-\alpha)[\mu_{i,j} r_i + \gamma \max_{b \in A} \{ Q_j^b(t) \}] \tag{9}$$

또한, 식 (3)에서 처럼 $r_j = \mu_{i,j} r_i$이므로, $Q_j^{a1}(t+1)$은 다시 식 (10)과 같이 쓰일 수 있다.

$$Q_j^{a1}(t+1) = \alpha Q_j^{a1}(t) + (1-\alpha)[r_i + \gamma \max_{b \in A} \{ Q_j^b(t) \}] \tag{10}$$

따라서, 식 (10)은 완전히 식 (2)와 등가의 식임을 알 수 있다. 식 (2)에 의해 얻은 Q-value는 최적의 행위로 수렴한다는 사실은 이미 Watkins[5]에 의해 증명되었으므로, 식 (10) 역시 최적의 행위로 수렴하게 된다. 즉, $r_j = \mu_{i,j} r_i$에 의해 정의된 보답을 사용하는 Q_j^{a1}을 기존의 Q-value 갱신식에 의해 갱신 시키더라도, iteration이 증가함에 따라 최대의 Q_j^{a1}는 최적 행위로 수렴하게 된다. □

만일 각 hyperbox 의 모든 꼭지점들이 hyperbox의 중앙으로 집중된다면, l 만큼 등 간격으로 떨어진 점 형태의 hyperbox을 얻을 수 있다. 이러한 경우, 임의의 hyperbox에 대한 $s_{i,j}$ and $s_{i,j+1}$사이의 유클리디안 거리를 나타내는 $d(s_{i,j}, s_{i,j+1})$은 hyperbox의 용적이 0이기 때문에 0이 되어야 함을 알 수 있다. 따라서, 마찬가지의 경우에 현재상태에 기인하는 보답은 모든 주변 상태에서도 역시 동일한 양만큼의 영향을 주게 된다. 결국, 모든 존재할 수 있는(고려되어지는) 상태들은 이산 상태 공간으로 정의되어 지고, 이러한 상태들에서의 최적 행위 생성을 위한 Q-value들은 기존의 Q-learning의 Q-value 갱신식 (2)에 의해 구할 수 있게 된다. 위의 고찰을 통해, 식 (4)에서 정의된 Q_j^{a1}에 대한 정의가 기존의 Q-learning에 의해 얻은 Q_j^{a1}값을 포함하는 일반적인 형태가 되기 위해서는 $\mu(s_i, s_{i,j})$함수에 대해 다음과 같은 특성이 성립해야 한다.

$$\lim_{d(s_i, s_{i,j}) \to} \mu_{i,j} \to 1 \tag{11}$$

여기서, 각 hyperbox에 대해 독립적인 *effect function*이 존재할 수 있으므로, *effect function*들의 갯수는 최대 주변 상태 수 *(Nl-1)n* 만큼 존재하게 된다. 위의 특성들을 만족하는 모든 *effect function*들을 구한다는 것은 매우 힘든 일이므로, 우리는 모든 hyperbox에 대해 다음 식 (12)와 같은 현상태로부터 멀어짐에 따라 보답의 영향이 단조 감소하는 종형의 *effect functions*를 사용하고자 한다.

$$\mu_{i,j}(s_i, s_{i,j}) = \exp(-\lambda \cdot d^2(s_i, s_{i,j})) \tag{12}$$

여기서, $d(x,y)$ 는 상태x와 상태y간의 유클리디안 거리를 나타내며, λ은 함수의 형태를 결정한다. 간단한 계산으로 식 (12)가 식(11)을 만족함을 알 수 있을 것이다. 이와 같이, 현재상태의 보답과 주변 상태의 보답간의 관계를 사용하여 갱신된 주변 상태의 현재 행위에 대한 Q-value은 주변 상태에 존재하는 삼각형의 Q-value 모델을 갱신하도록 영향을 준다.

3.2 삼각 형태의 Q-value 모델

Q-learning에서, 가능한 모든 상태에서 가능한 모든 행위들의 행위값을 나타내는 것이 Q-table이었다. 따라서, 만일 연속된 상태와 행위 공간에서 Q-learning을 수행하기 위해서는 이론적으로는 행위 수(무한대)와 상태 수(무한대)의 곱만큼의 기억 공간이 Q-table을 구성하기 위해 필요하다. 따라서 이를 해결하기 위해 이러한 Q-table을 특정 형태로 모델링하는 것이 필요하게 된다. Q-table의 설립 목적은 모든 행위 중 가장 큰 Q값을 갖는 행위를 찾아 이를 새로운 Policy의 요소로 등록하기 위해 사용되므로, 특정 행위에서 최고치를 갖고 특정 행위와 관계(거리)가 멀 수록 일정하게 Q-value가 작아지는 형태의 Q-table를 고려해 볼 수 있다. 행위간의 관계를 단지 행위 벡터 공간 내의 유클리디안 거리로 정의하고, 현재상태의 Q-table내의 Q-value들을 특정 행위에서 최고의 Q값을 갖고 거리에 비례적으로 단순 감소하는 특성이 있는 함수로 모델링하면, 특정 상태에서 특정 행위축에 대한 모든 행위들의 Q-value들을 그림 2와 같이 Cone-shaped function으로 나타낼 수 있으며, 이를 특정 상태에서 Q-value model이라고 정의한다.

일반적으로, Q-learning에서는 학습이 수렴된 후에 하나의 행위에 대해서만 최대의 Q값을 갖게 되므로, RQ-learning에서

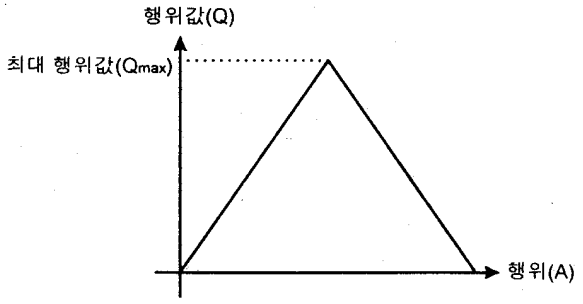


그림 2 행위값의 삼각 모델링

Fig. 2 Triangle-type modeling of Q-values

는 미리 하나의 최대의 Q값을 갖는 Q-value model을 위와 같이 정의하고 이러한 Q-value model의 최대 Q값 및 최대 Q값을 갖는 행위를 찾는 방법을 모색하였다.

3.3 Q-value model로부터 최대 Q값 및 최적 행위 생성

삼각형 Q-value 모델을 사용하여 식 (13)에서와 같이 현재상태에 대한 최적 행위는 현재 상태에서 가능한 모든 행위들의 Q-value들 중 최대치로 결정된다.

$$a_i = \arg\left(\max_{\forall a}\left\{\sum_{j=1}^{N}\mu_{i,j}Q^{a}_{i,j}\right\}\right) \tag{13}$$

이러한 최대Q값을 구하는 것은 삼각형의 최대와 최소점들만 고려하면 된다. 우선, 정의에 의해 삼각형의 최대와 최소점 사이의 $Q^{a}_{i,j}$은 선형직선을 이루고, $\mu_{i,j}$값은 상수이므로 $\mu_{i,j}Q^{a}_{i,j}$ 선형직선 역시 선형직선이 된다. 또한, 각 삼각형의 최대와 최소점들을 순서대로 나열하여 이루어진 점들의 집합에서 이웃하는 2점사이의 사이에 존재하는 각 삼각곡선의 선분들은 선형이므로 이들의 합은 선형직선이고 따라서 집합 내의 모든 점들 사이에는 선형 선분이 존재한다. 이러한 여러 삼각형의 합으로 이루어진 곡선의 최대와 최소점은 각 삼각형의 최대 및 최소점들만 고려하여 쉽게 얻을 수 있다. 따라서, 식 (13)의 해를 구할 수 있다.

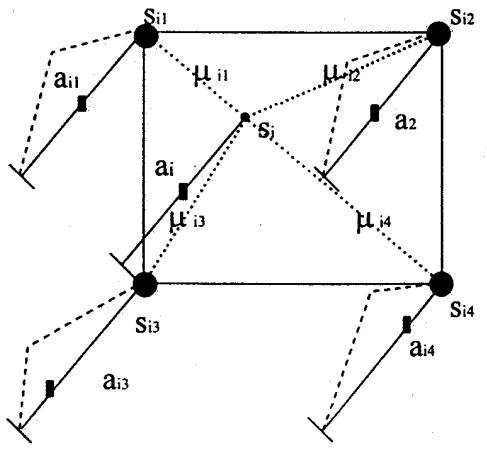


그림 3 주변 상태를 이용한 현재상태에서의 최적 행위 생성
Fig. 3 Current action estimation by using its neighboring states

식 (13)을 이용하여 구한 주변 상태의 최적 행위와 현재 상태의 최적 행위간의 관계를 그림 3에 나타내었다.

그림 4는 식 (13)에 의해 생성된 현재 행위의 예를 보여준다. 즉, 현재 행위를 수행한 후, 현재상태는 다음 상태로 변하고, 현재상태에서 수행된 행위에 대한 보답을 받게 된다. 다음으로 Q-value 갱신식에 의해 다음 iteration에서 사용하게 될 현재상태의 Q-value가 계산된다. 이렇게 계산된 Q-value에 근거하여 삼각형의 Q-value모델을 좌우 혹은 위쪽으로 조정하여 현재 행위에 대한 Q-value가 Q-value 모델에 의해 표현될 수 있도록 한다. 따라서 현재상태에서의 최적의 행위도 역시 이에 따라 조정되어 진다. 즉, 갱신된 Q-value 에 의해 Q-value 모델

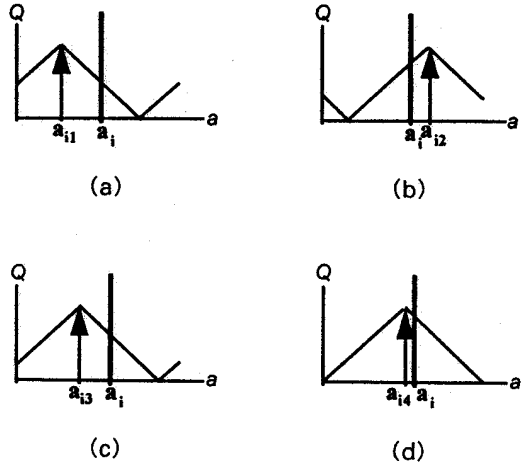


그림 4 현재 행위 a_i와 주변상태의 모든행위, $a_{i1}, a_{i2}, a_{i3}, a_{i4}$, 와의 관계
Fig. 4 An example of the current action a_i and the actions, a_{i1}, a_{i2}, a_{i3}, a_{i4}, of its neighboring states, S_{i1}, S_{i2}, S_{i3}, S_{i4}

을 고치는 것은 2가지 형태가 존재할 수 있다. 첫번째로, 갱신된 Q-value가 현재 가장 큰 Q-value보다 작으면, 현재 행위의 Q-value가 Q-value모델의 Q-value보다 낮은 경우는 현재 행위에서 멀어지는 방향으로, 또는 현재 행위의 Q-value가 Q-value모델의 Q-value보다 높은 경우는 현재 행위에 가까워지는 방향으로 조정하여 Q-value모델에서 현재 행위의 Q-value가 실제로 갱신된 Q-value를 만족할 수 있도록 한다. 이러한 조정은 각 행위 축별로 수행되며, 다음 식 (14)로 나타낼 수 있다.

$$a^{k}_{i,j}(t+1) = a^{k}_{i,j}(t) + \eta\, sgn(a^{k}_{i,j}(t) - a^{k}_{i}(t)) \tag{14}$$

여기서, $\eta = \dfrac{2a^{k}_{\max}}{Q_{\max}}\left|dQ^{a_1}_{i,j}(t+1)\right|$

식 (16)에서는 k번째 행위축의 갱신할 행위의 변위를 결정하기 위해 사용된다. 또한 sgn() 행위의 변위의 방향을 결정한다. 두 번째로, 만일 갱신된 Q-value가 현재 상태의 Q-value 모델의 최대 Q-value보다 크면, 식 (15)에서와 같이 최대 Q-value를 의미하는 현재상태의 최적의 행위는 현재 행위로 대치된다.

$$a^{k}_{i,j}(t+1) = a^{k}_{i}(t) \tag{15}$$

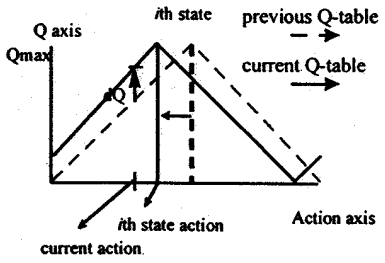
이러한 Q-value 모델의 갱신은 그림 5에 나타내었다.

앞에서 기술한 Q-value 모델에 근거한 최적 행위 갱신을 포함한 전체적인 RQ-Learning Algorithm은 다음과 같이 요약할 수 있다.

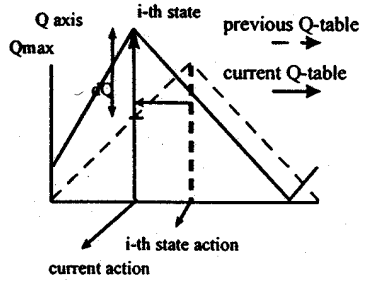
RQ-learning 알고리즘

(1) 초기화 : 여러 파라메타 (γ, α, ρ)의 초기화
(2) 현재 상태를 받아들임 (s ← 현재상태)
(3) 상태 s와 주변 상태와의 관계를 식 (12)를 이용하여 구한다.
(4) 상태 s에서 실행해야 할 행위는 식 (13)에 의해 구한다. (때로는 Random action 수행)

(5) 결정된 행위를 수행하고, Reward를 받는다.
(6) 주변 상태들에서의 Q값은 식 (10)을 사용하여 구한다.
(7) 주변 상태들에서의 최적의 행위 및 Q값을 식 (14), (15)를 사용하여 갱신한다.



(a)



(b)

그림 5 행위값(Q-value) 갱신 : (a) 좌우 조정을 통한 행위값 갱신 (b) 높이 조정을 통한 행위값 갱신
Fig. 5 The Q-value update scheme : (a) width modification and (b) height modification

4. Cluster-based Q-learning(CQ-learning)

앞에서 사용된 *effect function*은 현재상태와 그의 주변 상태들 사이의 보답 할당 관계를 나타내었다. 그러나 모든 hyper-box들 내의 보답 할당을 미리 설정한 하나의 보답 할당 함수에 의해 설명한다는 것은 매우 힘든 일이다. 따라서, 이러한 보답 할당 함수에 의해 정의될 수 있는 실제의 상태 영역을 구한다는 것은 많은 실험을 통해서만이 이룰 수 있을 것이다. 이러한 문제를 해결하기 위해 본 논문에서는 컨벡스 클러스터링 기법[11]을 사용하여 유사한 보답 할당 영역을 컨벡스 클러스터들로 표현하고, 영역기반 Q-learning을 수행하는 cluster-based Q-learning 기법을 제안하고자 한다.

4.1 CQ-learning의 구조

제안하고자 하는 Q-learning기법의 전반적인 구조는 그림 6과 같이 크게 2가지의 기능 모듈, 즉 상태 클러스터링 모듈과 영역기반 Q-learning모듈로 나뉘어진다. 상태 클러스터링 모듈에서는 비슷한 보답 할당이 이루어질 수 있는 영역은 단일의 클러스터로 표현한다. 또한, 영역기반 Q-learning 모듈에서는 발생된 클러스터들을 기반으로 현재상태에서 최적의 행위를 학습하게 된다.

그림 6에서 환경에 대한 실질적인 센서 정보가 CQ-learning을 위해 입력된다. 다음, 유용한 특징값들이 이들로 부터 계산되어, 현재상태를 만들어 낸다. 기존의 클러스터들 중에 현재상태를 포함하는 클러스터가 존재하는 지를 알아낸다. 만일 현재상태를 포함하는 클러스터가 존재하면, 클러스터의 각 꼭지점들은 RQ-learning의 주변 상태의 역할을 하기 위해 RQ-learning모듈에 제공된다. 만약 현재상태가 어떤 클러스터에도 속하지 않다면, 미리 설정된 특정한 행위(*default action*)를 수행하고 보답 *r*을 얻는다. 이러한 특정한 행위에 대한 실제 보답*r*은 현재상태 *si* 와 주변상태 *si,j* 사이의 유사도를 얻기 위해 사용된다. 즉, 받은 보답은 클러스터링 모듈로 입력되고, 이 모듈에서는 현재상태에서 받은 특정한 행위에 대한 보답과 클러스터의 각 꼭지점(주변상태)에 정의된 특정한 행위에 대한 보답들 사이의 유사도를 측정하는 *similarity function*을 사용하여 현재상태에서 얻은 보답을 가장 잘 표현할 수 있는 클러스터를 찾게 된다. 즉, 클러스터내의 모든 꼭지점들에 대해 유사도가 특정 문턱값 이상인 유사도들중 가장 유사한 클러스터를 구하고, 해당 클러스터로 현재 상태를 클러스터링 하게 된다. 따라서, 이러한 과정을 통해 유사한 보답 할당 영역이 만들어지게 된다. 따라서, 본 시스템은 현재상태를 포함하는 클러스터의 존재성 여부와 받은 보답에 따라 4가지의 동작이 가능하게 된다.

(1) 현재상태를 포함하는 클러스터가 존재하는 경우 RQ-learning을 통해 현재상태에서의 최적 행위를 수행하고, 주변상태의 최적행위를 3장에서와 같이 갱신한다.
(2) 현재상태를 포함하는 클러스터가 존재하지 않으며, 받은 reward가 가장 유사한 클러스터의 주변 상태들에 내재된 effect function에 의해 산정된 보답값과 유사한 경우 컨벡스 클러스터링이 수행된다.
(3) 현재상태를 포함하는 클러스터가 존재하지 않지만, 받은 reward가 가장 유사한 클러스터의 주변 상태들에 내재된 effect function에 의해 산정된 보답값과 유사하지않은 경우, 클러스터링이 수행되지 않고 새로운 클러스터가 현재상태에 할당된다.

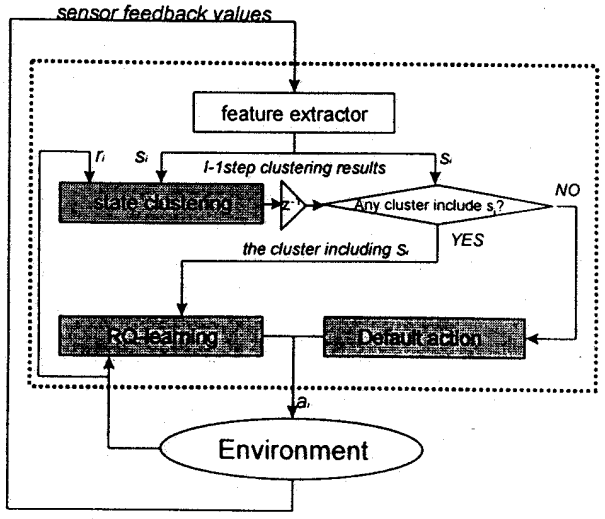


그림 6 CQ-learning 시스템의 기능 블록도
Fig. 6 Functional Architecture of CQ-learning system

4.2 State Clustering(SC) 알고리즘

를 r_{id}를 $\hat{r}_{id}^{j}$, $j=1,2,...,M$,들을 식(18)에서와 같이 구하게 된다. 사용하여 현재 상태에 대해 각 클러스터의 꼭지점의 갯수(M)만큼 산정된 보답값러스터링 모듈로의 입력값으로 현재상태를 나타내는 백터와 스칼라 보답값이 있다. 만일 입력된 현재상태가 클러스터에 속하면, RQ-learning에 의해 최적의 행위가 수행되고 받은 보답값에 의해 현재 클러스터의 주변상태의 최적행위가 갱신된다. 그러나 현재상태가 어떠한 클러스터에도 속하지않는 다면, 그러나 만일 입력된 상태가 어떠한 클러스터에도 포함되지않으면, 비슷한 보답 분포 영역을 구하기 위해 컨벡스 클러스터링이 수행된다. 컨벡스 클러스터링을 수행하기 위해 j번째 주변상태에서 특정한 행위에 대한 보답

$$\hat{r}_{id}^{j} = \mu_{i,j}(s_i, s_{i,j}) \cdot r_{jd}, \quad j=1,2,\ldots,M, \tag{16}$$

여기서, $\hat{r}_{id}^{j}$는 $\hat{r}_{id}^{j}$과 r_{id}을 비교하여 유사도를 얻기 위해 식

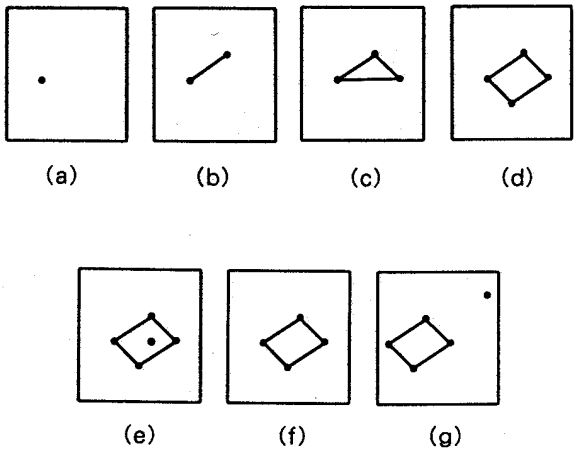


그림 7 SC 알고리즘의 수행 예. (a)먼저, 임의의 현재상태가 입력된다. (b),(c),(d) 가장 유사한 클러스터들이 현재의 입력상태를 포함하기 위해 현재의 입력상태 위치로 확장하는 컨벡스 클러스터링을 수행한다. (e) 클러스터내에 새로운 상태가 입력 된다. (f) RQ-learning에 의해 최적의 행위가 수행되고 클러스터에는 아무런 영향을 끼치지 못한다. (g) 현재상태가 모든 클러스터의 외부에 있고, 유사도가 모든 클러스터에 대해 정의된 문턱값보다 작으므로 새로운 클러스터가 생성된다.

Fig. 7 Operation flow of SC Algorithm. (a) First, an input state arrives. (b), (c), (d) The most similar cluster expand to the current input state. (e) A new state is arrived in a the input state. (f) A current action with the highest Q-value is performed by RQ-learning. (h) A new state is arrived, and since a new state is not similar to the nearest cluster, a new cluster is generated at the input state position

(17)를 사용한다. 현재상태에서의 보답값 모든 클러스터에 대해 수행되어지고, 식 (16)에서 구한 모든 클러스터의 모든 꼭지점(주변상태)에 대한 보답 추정값들 이러한

$$\xi(r_i, \hat{r}_i) = \frac{1}{1+d^2(r_1, \hat{r}_i)} \tag{17}$$

한 클러스가 어느 특정한 값(threshold value) 이상이면 이는 클러스터의 각 주변 상태에 있는 보답으로 부터 현재의 보답을 추정할 수 있고 이의 오차는 설정한 유사도의 문턱값에 반비례한다는 것을 의미한다. 따라서 본 연구에서는 클러스터링을 수행할 클러스터의 후보로서 클러스터 내의 모든 주변상태에 대해 이러한 유사도 문턱값보다 큰 유사도를 나타내는 클러스터들을 고르고, 이들 중 평균 유사도가 가장 큰 클러스터를 현재상태로 확장할 클러스터로 선택하도록 하였다. 여기서 확장의 구체적인 방법은 이전의 컨벡스 클러스터링 방법[11]에서의 확장 방법을 사용한다. 이와 같은 상태 클러스터링(State Clustering)은 그림 7과 같이 수행 된다. 터내의 모든 꼭지점에 대해 비슷한 정도 한편 클러스터 A와 B가 겹쳐지는 영역내의 특정한 상태는 두개의 클러스터가 나타내는 보답 분포 영역 특성을 공통적으로 소유하고 있으므로, 기존의 컨벡스 클러스터링 방법[11]에서와는 달리 클러스터링을 수행하는 도중 클러스터 사이의 겹침은 고려하지 않기로 한다. 컨벡스 클러스터링의 자세한 사항은 저자의 논문[11]을 참고하기 바란다.

4.3 Action generation

비슷한 보답 분포 영역을 클러스터링을 통해 구하기 위해, 본 연구에서는 두가지의 행위생성 방법을 제안하였다. 우선, 현재상태가 임의의 클러스터 내에 속하는 경우로, 이 경우는 3장에서의 RQ-learning을 통해 최적 행위를 수행하게 된다. 이와는 반대 경우로 현재상태가 임의의 클러스터 내에 속하지 않는 경우는 미리 설정한 특정한 행위를 수행하게 되고, 이는 클러스터링이 일어나 현재상태로 주변상태가 확장하게 되면 클러스터의 새로운 주변상태가 될 현재상태에 대해, 특정한 행위에 대한 보답을 기억 시켜두고자 함이며, 이러한 값들을 이용하여 앞 절에서 설명한 것과 같이 현재상태 대한 보답값이 산정되어진다.

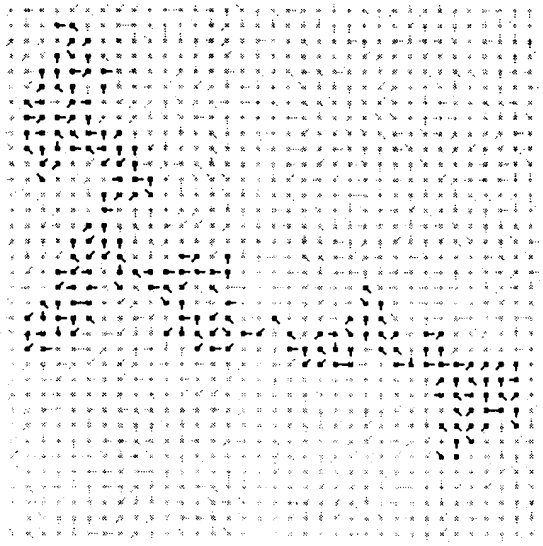
5. 시뮬레이션 결과

초기 위치를 (50,50)으로 하고 최종 위치(320,320절한 $\gamma(\alpha,\rho)$의 적절한 선택이 선행되어야 한다.)라 할 때, Q-learning의 경우 위치 상태 공간을 각 축마다 35개의 분해능으로 나누어야 목표 지점에 도달할 수 있다. 그림 8(a)(c)은 아무런 사전 정보 없이 이동하는 초기 iteration에서는 여러 방향으로 탐색하는 과정을 보여준다. 그러나 그림 8(b)에서와 같이 Q의 경우 약 400번의 iteration을 수행한 뒤에 원하는 상태 근처로 수렴하는 반면에, FQ의 경우 그림 8(d)에서와 같이 50번의 iteration후에 수렴하는 것을 볼 수 있다. 그러나 Q, CQ 양쪽 모두, 많은 iteration을 수행한 후에도 수렴하지 않는 경우가 있는데 이는 보답과 파라메터 (γ,α,ρ)설정이 잘못된 경우이다. 예로써, 상태 s에서 현재 행위 $a1$에 대한 reward를 $r1$이라고 하고, 최적행위, $a2$의 reward를 $r2$라고 하자. Optimal 행위 $a2$가 policy에 등록되기 위해서는 Q value가 가장 커야 하므로 식 (12)에서와

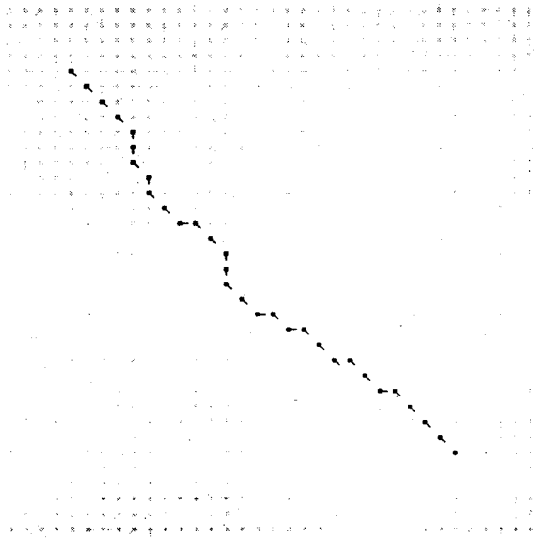
표 1 시각 추적 작업을 위한 상태 및 행위 변수

Table 1 The state and action variables for a visual tracking task

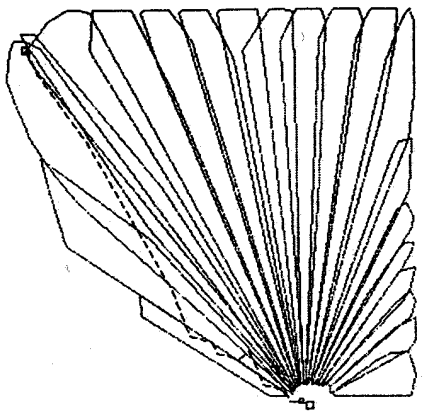
상태 및 행위변수 (범위)	변수에 대한 설명
(-30 ~ 210 degree)	로보트 1축의 각도
$\triangle P_x$(-10 ~ +10 cm)	화면 좌표계의 중앙 위치와 물체 위치와의 x축 방향으로 차이
θ^{12} (-30 ~ 210 degree)	로보트 2축의 각도
$\triangle P_y$(-10 ~ +10 cm)	화면 좌표계의 중앙 위치와 물체 위치와의 y축 방향으로 차이
a_{I1}(-5 ~ +5 degree)	로보트 1축의 각속도
a_{I2}(-5 ~ +5 degree)	로보트 2축의 각속도



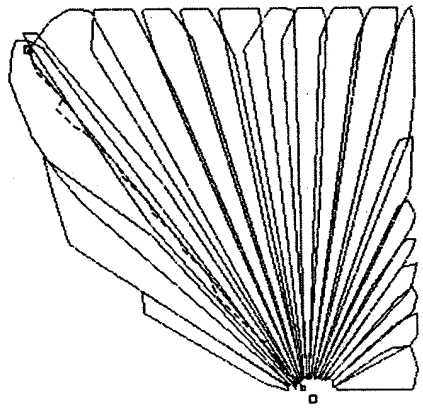
(a)



(b)



(c)



(d)

그림 8 Q 및 CQ-learning 시뮬레이션 결과 :
(a) Q-learning의 첫번째 iteration 후의 모습
(b) Q-learning의 400번째 iteration 후의 모습
(c) CQ-learning의 30번째 iteration을 수행한 후
(d) CQ-learning의 50번째 iteration을 수행한 후

Fig. 8 Q and CQ-learning simulation results :
(a) First iteration by Q-learning
(b) 400th iteration by Q-learning
(c) Most clusters are found after 30th Iteration by CQ-learning
(d) Each cluster converges to an optimal action after 50th iteration in CQ-learning

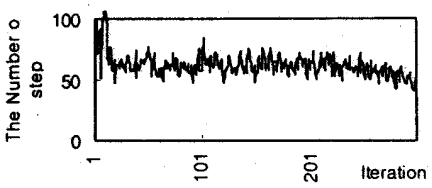
같이 갱신을 반복하는 과정에서 현재 Q값의 차이를 γ, α 혹은 ρ를 사용하여 극복하여야 한다. 즉, $\gamma(\alpha,\rho)$를 제외한 나머지 파라메터를 상수라고 하면 두 행위에 대한 Q값 갱신 속도는 $\gamma(\alpha,\rho)$에 비례하여 결정된다. 따라서 적절한 $\gamma(\alpha,\rho)$의 적절한 선택이 선행되어야 한다. 또한 Q와 FQ의 iteration 별 step 수는 약 40번 내외로 수렴됨을 알 수 있었다. RQ-learning 시뮬레이션 결과 우수한 수렴 속도와 Q-learning보다 부드러운 행위 집합을 학습함을 알 수 있다. 마지막으로 그림 9, 10에서 각 알고리즘의 iteration별 평균 step수를 보였다.

이와 더불어, 2 자유도(DOF)를 갖는 SCARA 로보트에 대해

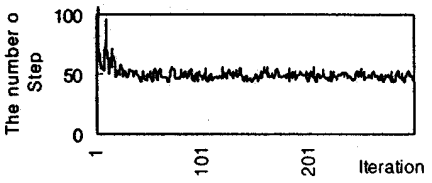
시각 추적 작업을 실시하였다. 이를 위해 4개의 특징들로(로보트 좌표계에서 본 각 Robot 팔의 각도. 화면 좌표계에서 본 물체의 x, y 속도 성분) 이루어진 4-D의 상태 공간이 정의되었다. 또한 행위 공간은 각각 Robot 팔의 각속도로 정의 하였다. 자세한 사항은 표 1에 나타내었다.

또한, 보답은 다음 식 (18)와 같이 정의 되었다.

$$r=\frac{(\|x_i+1-x_g\| \ -\|x_i-x_g\| \ )}{K} \tag{18}$$



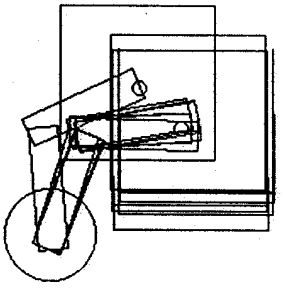
(a)



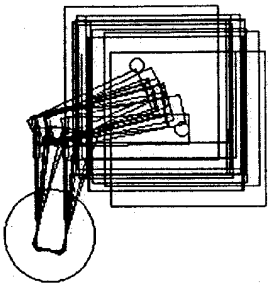
(b)

그림 9 Q과 CQ-learning의 iteration당 스텝 수 : (a) Q-learning 경우, (b) CQ-learning 경우

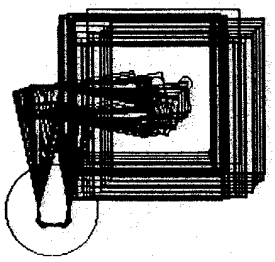
Fig. 9 The number of steps of Q, and CQ-learning : (a) Q-learning case, (b) CQ-learning case



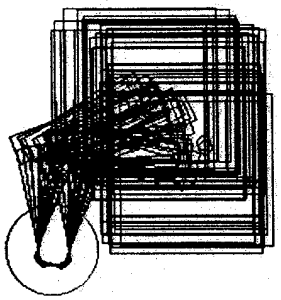
(a) 사용된 각속도 영역 : -15 ~ +15 degree/step
(a) Angular velocity Range(-15~+15 degree/sampling time)



(b) 사용된 각속도 영역 : -10 ~ +10 degree/step
(b) Angular velocity Range(-10~+10 degree/sampling time)



(c) 사용된 각속도 영역 : -5 ~ +5 degree/step
(c) Angular velocity Range(-5~+5 degree/sampling time)



(d) 사용된 각속도 영역 : -10 ~ +10 degree/step
(d) Angular velocity Range(-10~+10 degree/sampling time)

그림 10 제안된 CQ-learning방법을 사용한 2-DOF SCARA 로보트의 시각 추적 동작(Visual tracking): (a),(b)와 (c)는 서로 다른 두 지점에 있는 물체를 지정된 각속도 범위에서 학습한 시각 추적한 경우이며 (d)는 여러 개의 임의 위치에 위치한 물체를 특정한 각속도 범위에서 시각 추적을 수행한 경우이다.

Fig. 10 Visual tracking of a 2-DOF SCARA robot by our proposed Cluster-based Q-learning technique : (a), (b), and, (c) Visual tracking an object at two different positions, (d) Visual tracking an object at arbitrary different positions

여기서, x_i 와 x_{i+1} 는 각각 현재상태와 다음 상태를 나타내며, x_g 는 최종 목표 상태를, K는 Robot팔이 최대 각속도로 움직이어 변할 수 있는 상태 변이를 나타낸다. 본 시뮬레이션에서 목표 물체는 무작위로 움직이도록 하였다. 매 샘플링 시에 로보트는 본 논문에서 제안하였던 CQ-learning을 이용하여 이동하는 목표 물체 따라 움직이는 것을 학습하게 된다. 이러한 학습 후의 모습이 그림 10에 나타나있다. 본 시뮬레이션에서는 평균 1500~2000정도의 iteration을 거친 후 로보트가 최적의 행위를 수행함을 알 수 있었다.

6. 결 론

본 논문에서는 Structural Credit Assignment 문제를 해결하기 위해 도입된 영역 기반 Q-value 할당 기법과 비슷한 Q-value할당영역을 찾기 위해 도입된 컨벡스 클러스터링 기법을 사용하여 연속 상태 공간 및 행위 공간에서 최적의 행위를 학습할 수 있는 새로운 Q-learning방법을 제안하였다. 제안한

방법의 우수성을 보이기 위해, 기존의 Q-learning과의 2-D 자유 공간에서의 자율 주행 비교를 수행 하였으며, 그 결과, 본 CQ-learning이 기존의 Q-learning보다 더 적은 iteration후에도 최적의 행위로 수렴함을 보였다. 또한, 본 알고리즘이 연속적인 행위 및 상태 공간이 고려되어져야 하는 실질적인 적용에 보다 적합함을 보이기 위해, 가상적인 2-DOF 스카라 로보트의 시각 추적 작업 시뮬레이션을 수행하였다. 기존의 Q-learning과 마찬가지로 파라메타의 설정에 따라 최적행위로의 학습 속도가 달라지기 때문에 현재는 본 알고리즘의 여러 파라메터를 학습 상태에 따라 적절히 바꾸어 주는 자동 파라메타 조절에 관한 연구를 진행 중에 있다.

참 고 문 헌

[1] M. A. Salichs, E. A. Puente, D. Gachet, and J. R. Pementel, Learning behavioral control by reinforcement for an autonomous mobile robot, *Proc. IEEE Conference on R&A*, Vol. 1, pp. 1436-1441, 1993.

[2] H. Berenji and P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcement, *IEEE Trans. On Neural Networks*, Vol. 3, No. 5, Sept. 1992.

[3] L. J. Lin, Programming robots using reinforcement learning and teaching, *Proc. the Ninth National Conference on Artificial Intelligence*, 1991.

[4] G. Tesauro, *Practical issues in temporal difference learning*, Machine Learning, 1992.

[5] C. Watkins and P. Dayan, Q-learning, technical note, *Machine Learning*, Vol. 8, pp. 279-292, 1992.

[6] C. Watkins, Learning from delayed rewards, Ph.D. Thesis, University of Cambridge, England, 1989.

[7] A. Minoru, U. Eiji, and H. Koh, Behavior Coordination for a Mobile Robot Using Modular Reinforcement Learning, Proc. IEEE/RSJ Conference on Intelligent Robots and Systems, Vol. 3, pp. 1329-1336, 1996.

[8] P. Y. Glorennec, Fuzzy Q-learning and dynamical fuzzy Q-learning, Proc. IEEE Conference on R&A, Vol. 1, pp. 474-479, 1994.

[9] H. R. Berenji, Fuzzy Q-learning : A new approach for fuzzy dynamic programming, Proc. IEEE Conference on R&A, Vol. 1, pp. 486-491, 1994.

[10] A. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi, Fuzzy interpolation-based Q-learning with continuous states and actions, Proc. IEEE Conference on Fuzzy Systems, Vol. 1, pp. 594-600, 1996.

[11] I. H. Suh, J. H. Kim, and F. J. H. Lee, "Fuzzy Clustering involving Convex Polytopes," Proc. 5nd IEEE Conference on Fuzzy System, New Orleans, LA, Vol. 2, pp. 1013-1019, 1996.

저 자 소 개

김 재 현 (金 載 顯)

1969년 1월 16일생. 1991년 한양대 공대 전자공학과 졸업. 1993년 동 대학원 전자공학과 졸업(석사). 1993년~현재 동 대학원 전자공학과 박사과정 재학중

서 일 홍 (徐 一 弘)

1955년 4월 16일생. 1977년 서울대 공대 전자공학과 졸업. 1982년 한국과학기술원 졸업. 1982년~1985년 대우중공업 기술연구소 근무. 1987년~1988년 미국 미시간대 객원 연구원. 현재 한양대 공대 전자공학과 교수