지능형 로보트 시스템을 위한 영역기반 Q-Learning

Region-based Q-Learning for Intelligent Robot Systems

김 재 현, 서 일 홍
(Jae Hyun Kim, Il Hong Suh)

Abstract : It is desirable for autonomous robot systems to possess the ability to behave in a smooth and continuous fashion when interacting with an unknown environment. Although Q-learning requires a lot of memory and time to optimize a series of actions in a continuous state space, it may not be easy to apply the method to such a real environment. In this paper, for continuous state space applications, to solve problem and a triangular type Q-value model? This sounds very ackward. What is it you want to solve about the Q-value model. Our learning method can estimate a current Q-value by its relationship with the neighboring states and has the ability to learn its actions similar to that of Q-learning. Thus, our method can enable robots to move smoothly in a real environment. To show the validity of our method, navigation comparison with Q-learning are given and visual tracking simulation results involving an 2-DOF SCARA robot are also presented.

Keywords : Q-learning, region-based reward assignment, Q-value model, neighboring states, visual tracking

I. 서론

최근 들어, 제어(Control), 계획(Planning), 상황 결정(Decision making)등을 실시간으로 처리하고자 하는 지능형 시스템(Intelligent system)에 관한 연구가 많은 관심을 끌게 되었다[1-3]. 이러한 지능형 로보트 시스템을 개발하기 위해서는, 로보트가 미지의 환경에서 적절히 행동하는 것을 경험을 통해 학습하는 것이 매우 중요하다. 강화 학습은 이러한 목적을 달성하기 위해 개발된 학습 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. 따라서 불연속 상태 공간으로 모델링되는 미지의 환경과 상호작용을 통하여 불연속적인 행위들 중 최적의 행위를 학습할 수 있었다. 이러한 방법들 중 Q-learning은 가장 널리 사용되는 방법들 중 하나이다. Q-learning은 미래의 행위에 대한 보답에 감쇠를 고려한 평가척도 Q값을 추정하여 주어진 현재상태로부터 최적의 행위를 찾아내는 알고리즘이다. 이러한 Q-learning에서 현재 행위에 대한 평가를 위해 행위값(Q-value)을 정의하여 사용하게 되는데 이러한 행위값들은 각 상태에서 최적의 행위를 수행할 수 있는 근거 자료 역할을 수행하게 된다. 이러한 Q-learning을 실질적인 작업에 적용하기 위해서 많은 노력이 진행되어 왔다[7-8]. 예로써, Berenji등에 의해 개발된 Fuzzy Q-learning방법은 Q-value 갱신식에 상황에 대한 제약을 첨가하여 현재의 상황을 반영하고자 하였다[7]. 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. 예로써 Q-learning의 경우 각 상태에서 각 행위값을 기록하기 위해 Q-table이 필요한데, 2-D 상태 공간(state space)에서 각 상태 축이 1000개의 분해능(resolution)을 갖고, 1000개의 가능한 행위를 가정하더라도 이를 위해 필요한 Q-table의 크기는 1000로 계산될 수 있으므로 1,000,000,000개의 기억 공간이 필요하게 된다. 또한 이러한 기억 공간이 확보되었다 할지라도 1,000,000만큼의 상태를 학습하기 위해서는 많은 시간이 요구되며, Q-learning은 불연속 출력을 학습하는 방법이므로 연속적이지 못한 행위를 생성하고, 생성된 최적의 행위에는 항상 해당 분해능에 반비례하여 학습 오차가 존재하게 된다. 이러한 문제점을 해결하기 위해 퍼지의 Q-table을 새로운 퍼지 Rule들로 대체하여 각 Q값들을 퍼지 추론에 의하여 생성하고, 각 Rule의 후건부 파라메타들을 Steepest descent 방법으로 조정하고자 하는 새로운 시도가 이루어졌다[9]. 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-learning의 기본적인 개념인 통계적인 Q값 갱신을 통하여 최적의 행위를 찾고자 하는 방법과는 달리, Q값을 생성하는 각 Rule들의 후건부 파라메타를 조정할 때 현재의 Q값과 각 rule로부터 생성되는 새로운 Q 값과의 차(Q)를 최소화하도록 하는 방법으로 비록 Q를 계산할 때 기존의 Q-learning의 Q공식을 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. 이러한 문제점을 해결하기 위해, 연속적인 상태 및 행위 공간에서 최적의 행위를 학습하는 새로운 Q-learning 방법이 요구된다.

여러 가지의 강화 학습들을 특성화 하는 기본적인 논제들 중 credit assignment problem은 *일련의 sensor-action-feedback으로부터 어떻게 최적의 행위를 배울 것인가*로 정의되며, 각 강화 학습에서 풀어야 할 기본적인 문제이다. 이러한 credit assignment problem중 structural credit assignment problem은 "*현재 받은 reward가 상태 공간 내의 각 상태들에게 어떻게 영향을 끼칠 것인가*"로 정의된다. 이러한 관점에서, 기존의 Q-learning은 point-based credit assignment 방법이라고 정의 내릴 수 있다.

접수일자 : 1997. 1. 20., 수정완료 : 1997. 3. 2.
김재현, 서일홍 : 한양대학교 전자공학과

따라서 이러한 상태 점에 기반으로 보답(reward)를 할당하는 Q-learning을 일반화하기 위해 특정한 상태 영역에 보답(reward)를 할당하는 영역 기반(Region-based) Q-learning (RQ-learning)을 제안하고자 한다. 제안하고자 하는 방법에서, 현재상태의 보답(reward)은 현재상태의 주변 상태로 전파되고, 전파된 각 주변 상태의 보답을 기반으로 주변 상태에서의 Q-value가 기존의 Q-value갱신식에 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.

II. 이산 상태 공간에서의 Q-learning

강화 학습에서는 기본적으로 로보트와 환경과의 상호작용을 이산 반복 공정(Discrete time cyclic processes)에서 동작하는 유한 상태를 갖는 두개의 대행자들(환경과 agent)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 보답(Reward)을 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 모델링할 수 있다고 가정한다.

Q-learning 알고리즘

[초기화]

1. 초기화 :
   (1) 난수 혹은 사전 정보를 이용한 Q-table Q(s,a) 초기화
   (2) 초기화된 Q-table를 근거로 Policy fi 초기화

$$f_i \leftarrow a \text{ such that } Q_i^a(t+1) = \max_{b \in A} \{Q_i^b(t)\}, \quad (1)$$

여기서 t는 tth iteration을, i는 현재상태를, A는 현재 정의된 행위 집합을 각각 나타내며, 따라서 fi는 현재 상태에서 최적의 행위계획(policy)을 나타내고, $Q_i^a(t+1)$는 다음 iteration의 현재 상태 i에서 수행할 행위 a에 대한 행위값을 나타낸다.

   (3) 여러 파라메타(γ, α, ρ)의 초기화

[반복]

2. 현재 상태을 받아들임 (s ← 현재상태)
3. Policy Table로부터 현재상태에 해당하는 행위 a를 행하거나 ρ 만큼의 비율로 임의의 행위를 수행. 여기서 랜덤 행위를 수행하는 것은 최적의 policy를 구하기 위한 필요조건이 된다.
4. 환경으로부터 수행된 행위에 대한 보답(Reward)r를 받음.
5. 다음 (2)를 이용하여 현재 상태에서 수행한 행위값 (Q-value) Q(s,a)를 갱신.

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

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

6. (1)를 이용하여 Policy fi 갱신.

여기서, 최적 행위에 대한 Q-value는 (3)과 같이 보답치 r과 상태 전이 확률 T 및 다음 상태에서의 Q-value 등으로 정의 된다.

$$Q_i^a(t+1) = r(t) + \gamma \sum_{i+1 \in S} T(i,a,i+1) \max_{b \in A}\{Q_{i+1}^b(t)\}, \quad (3)$$

여기서, $Q_i^a(t+1) = Q_i^a(t)$ 이므로, (2)가 (4)와 같이 바꾸어 쓸 수 있다. iteration이 증가할 수록, (2)에서의 Q-value 갱신식이 (3)으로 수렴해 간다는 것을 다음에서와 같이 알 수 있다. 즉, 수렴 후에는

$$Q_i^a(t) = r_i^a(t) + \gamma \max_{b \in A}\{Q_{i+1}^b(t)\}, \quad (4)$$

만일 모든 i와 a에 대해 T(i,a,i+1)=1를 만족하면 (3)이 (4)가 됨을 알 수 있다. 이러한 측면에서, Q-learning은 (2)에서의 T-matrix가 현재 상태에서 특정한 행위에 의해 특정한 다음 상태로 전이되는 확률이 1이라는 가정하에서 이루어진 방법임을 의미한다. 즉, 동일한 상태에서 특정 행위에 의해 나타날 수 있는 상태는 오직 하나라는 것을 의미하며, 따라서, 이러한 환경에서 행위 궤적내의 전체 행위값의 합을 최대화 시키고자 하는 응용 분야에 적용될 수 있다는 것을 의미한다.

III. 영역 기반(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 제한를 극복하기 위해, 본 논문에서는 먼저 기존의 Q-learning 을 영역 기반으로 보답(reward)을 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 형태의Q-value 모델을 이용한 주변 상태들의 Q-value 갱신 방법과 최적 행위 생성 방법을 각각 3-2, 3-3 기술하였다.

1. 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.

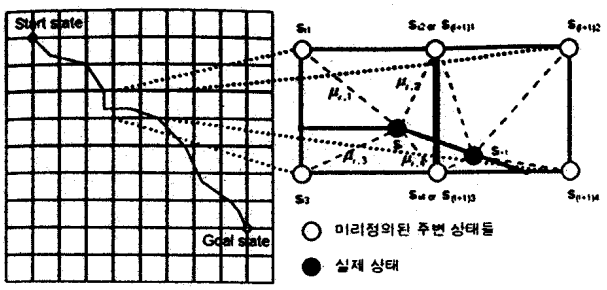


그림 1. hyperbox내의 현재 상태 *si* 의 주변 상태 정의 및 주변 상태 *si,j* 로의 영역 기반(Region-based) 보답 할당.

Fig. 1. The definition of current state and Region-bosed reward assignment.

이때, 임의의 hyperbox내의 임의의 위치에 있을 수 있는 현재 상태 s_i에서 얻은 보답(reward)을 r_i 라고 정의하고, 현재 상태의 j번째 주변 상태 s_{ij} 의 보답을 r_j 라 정의한다. 현재 상태의 보답과 주변 상태로 전파되는 보답과의 관계를 *effect function* $\mu_{i,j}(s_i,s_{ij})$ 로 정의한다면 현재 상태의 보답으로부터 주변 상태로 전파되는 보답은 (5)와 같이 정의 될 수 있다.

$$r_j = \mu_{i,j} r_i \tag{5}$$

그림 1에서 볼 수 있는 것과 같이, 특정 주변 상태의 보답은 $\mu_{i,j}(s_i,s_{ij})$와 ri을 곱함으로써 얻을 수 있다. 따라서 최적의 policy를 따를 때 정의되는 s_{ij}에 전달되는 보답의 감쇠 합인 Q-value는 다음 (6)과 같이 쓰여질 수 있다.

$$Q_j^{a_i} = \sum_{n=0}^{\infty} \gamma^n \mu_{i+n,j} r_{i+n} \ , \tag{6}$$

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

증명 : 현재 상태에서, (5)에서 처럼 $r_j = \mu_{i,j} r_i$ 에 의해 정의된 보답을 사용하는 (6)의 $Q_j^{a_i}$ 을기존의 Q-value 갱신식에 의해 갱신시키면, iteration이 증가함에 따라 최대의 $Q_j^{a_i}$ 는 최적 행위로 수렴하게 된다.

$$Q_j^{a_i}(t) = \sum_{n=0}^{\infty} \gamma^n u_{i+n,j} r_{i+n} \tag{7}$$

$$= \mu_{i,j} r_i + \sum_{n=0}^{\infty} \gamma^n \mu_{i+n,j} r_{i+n} \tag{8}$$

$$= \mu_{i,j} r_i + \gamma \sum_{n=1}^{\infty} \gamma^{n-1} \mu_{i+n,j} r_{i+n} \tag{9}$$

$$= \mu_{i,j} r_i + \gamma \max_{b \in A} \{Q_j^b(t)\} \tag{10}$$

여기서, $Q_j^{a_i}(t+1)$ 를 나타내는 기존의 Q-learning 갱신식인 (2)는 (10)을 이용하여 다음 (11)로 다시 쓸 수 있다.

$$Q_j^{a_i}(t+1) = \alpha Q_j^{a_i}(t) + (1-\alpha)\left[\mu_{i,j} r_i + \gamma \max_{b \in A} \{Q_j^b(t)\}\right] \tag{11}$$

또한, (5)에서 처럼 $r_j = \mu_{i,j} r_i$ 이므로, $Q_j^{a_i}(t+1)$ 은 다시 식 (12)와 같이 쓰일 수 있다.

$$Q_j^{a_i}(t+1) = \alpha Q_j^{a_i}(t) + (1-\alpha)\left[r_j + \gamma \max_{b \in A} \{Q_j^b(t)\}\right] \tag{12}$$

따라서, (12)는 완전히 (2)와 등가의 식임을 알 수 있다. (2)에 의해 얻은 Q-value는 최적의 행위로 수렴한다는 사실은 이미 Watkins[5]에 의해 증명되었으므로, (12) 역시 최적의 행위로 수렴하게 된다. 즉, $r_j = \mu_{i,j} r_i$ 에의해 정의된 보답을 사용하는 $Q_j^{a_i}$ 을 기존의 Q-value 갱신식에 의해 갱신 시키더라도, iteration이 증가함에 따라 최대의 $Q_j^{a_i}$ 는 최적 행위로 수렴하게 된다. ■

만일 각 hyperbox 의 모든 꼭지점들이 hyperbox의 중앙으로 집중된다면, 1 만큼 등 간격으로 떨어진 점인 hyperbox을 얻을 수 있다. 이러한 경우, 임의의 hyperbox에 대한 s_{ij} and s_{ij-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, (6)에서 정의된 $Q_j^{a_i}$ 에대한 정의가 기존의 Q-learning에 의해 얻은 $Q_j^{a_i}$ 값을 포함하는 일반적인 형태가 되기 위해서는 $\mu(s_i, s_{i.j})$ 함수에 대해 다음과 같은 특성이 성립해야 한다.

$$\lim_{d(s_i, s_j) \to 0} \mu_{i.j} \to 1 \tag{13}$$

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 (14)와 같은 현상태로부터 멀어짐에 따라 보답의 영향이 단조 감소 하는 종형의 *effect function*들을 사용하고자 한다.

$$\mu_{i.j}(s_i, s_{i.j}) = \exp(-\lambda \cdot d^2(s_i, s_{i.j})) \tag{14}$$

여기서, $d(x,y)$ 는 상태x와 상태y간의 유클리디안 거리를 나타내며, λ은 함수의 형태를 결정한다. 간단한 계산으로 (14)가 (13)을 만족함을 알 수 있을 것이다. 이와같이, 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.

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 구성

하기 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-값이 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.

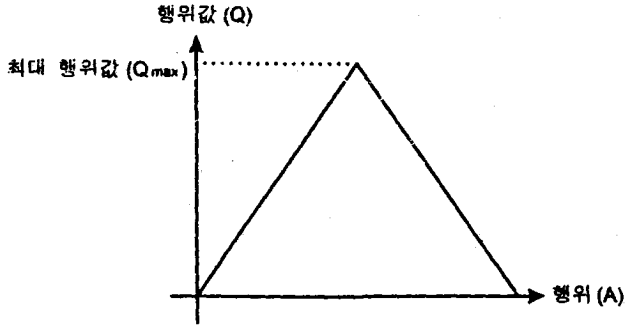


그림 2. 행위값의 삼각 모델링.

Fig. 2. Triangle-type modeling the Q-value.

일반적으로, Q-learning에서는 학습이 수렴된 후에 하나의 행위에 대해서만 최대의 Q값을 갖게 되므로, RQ-learning에서는 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. Q-value model로부터 최대 Q값 및 최적 행위 생성

삼각형 Q-value 모델을 사용하여 (15)에서와 같이 현재상태에 대한 최적 행위는 현재 상태에서 가능한 모든 행위들의 Q-value들 중 최대치로 결정된다.

$$a_i = \arg(\max_{\forall a}\{\sum_{j=1}^{N}\mu_{i,j}Q^a_{i,j}\}) \tag{15}$$

이러한 최대Q값을 구하는 것은 삼각형의 최대와 최소점들만 고려하면 된다. 우선, 정의에 의해 삼각형의 최대와 최소점사이의 $Q^a_{i,j}$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 삼각직선의선분들은 선형이므로 이들의 합은 선형직선이고 따라서 집합내의 모든 점들사이에는 선형선분이 존재한다. 이러한 여러 삼각형의 합으로 이루어진 곡선의 최대와 최소점은 각 삼각형의 최대최소 점들만 고려하여 쉽게 얻을 수 있다. 따라서, (15) 의 해를 구할 수 있다. (15)을 이용하여 구한 주변 상태의 최적 행위와 현재 상태의 최적 행위간의 관계를 그림 3에 나타내었다. 즉, 현재 행위를 수행한 후, 현재상태는 다음 상태로 변하고, 현재상태에서 수행된 행위에 대한 보답을 받게 된다.

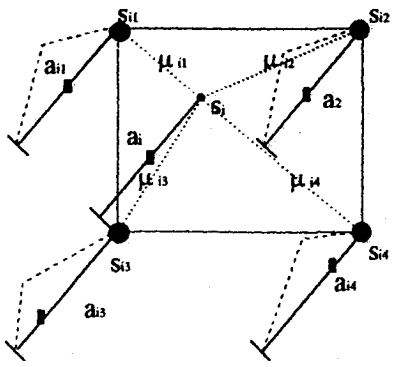


그림 3. 주변 상태를 이용한 현재상태에서의 최적 행위 생성.

Fig. 3. Optimal action estimation in current state by using it's neighboring state.

그림 4는 (15)에 의해 생성된 현재 행위의 예를 보여준다.

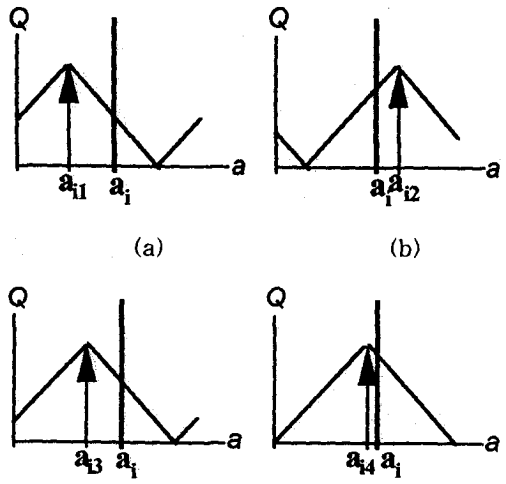


그림 4. 현재 행위 *ai*와 주변상태의 모든 행위 *ai1*, *ai2*, *ai3*, *ai4*, 와의 관계.

Fig. 4. An example of the current action *ai* and the actions, *ai1*, *ai2*, *ai3*, *ai4*, of it's neighboring states.

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. 이 갱신된 Q-value 에 의해 Q-value 모델을 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

도록 한다. 이러한 조정은 각 행위 축별로 수행되며, 다음 (16)로 나타낼 수 있다.

$$a^k_{i,j}(t+1) = a^k_{i,j}(t) + \eta \,\mathrm{sgn}(a^k_{i,j}(t) - a^k_i(t)) \quad (16)$$

여기서,

$$\eta = \frac{2a^k_{\max}}{Q_{\max}} \left| dQ^{a_i}_{i,j}(t+1) \right|$$

(16)에서 η는 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, (17)에서와 같이 최대 Q-value를 의미하는 현재상태의 최적의 행위는 현재 행위로 대치된다.

$$a^k_{i,j}(t+1) = a^k_i(t) \quad (17)$$

이러한 Q-value 모델의 갱신은 그림 5에 나타내었다.

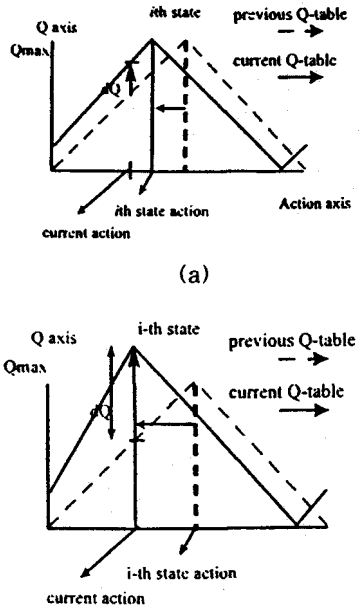


그림 5. 행위값(Q-value) 갱신. (a) 좌우 조정을 통한 행위값 갱신 (b) 높이 조정을 통한 행위값 갱신.

Fig. 5. The Q-value update. (a) width modification and (b) hight modification.

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

RQ-learning 알고리즘

[초기화]

1. 초기화 : 여러 파라메타(γ, α, ρ)의 초기화

[반복]

2. 현재 상태를 받아들임 ($s \leftarrow$현재상태)
3. 상태 s와 주변 상태와의 관계를 (14)를 이용하여 구한다.
4. 상태 s에서 실행해야 할 행위는 (15)에 의해 구한다. (때로는 Random action 수행)
5. 결정된 행위를 수행하고, Reward를 받는다.
6. 주변 상태들에서의 Q값은 (12)를 사용하여 구한다.
7. 주변 상태들에서의 최적의 행위 및 Q값을 (16),(17)을 사용하여 갱신한다.

IV. 시뮬레이션 결과

초기 위치를 (50,50)으로 하고 최종 위치(320,320)라 할 때, Q-learning의 경우 position 상태 공간을 각 축마다 35개의 resolution으로 나누어야 목표 지점에 도달할 수 있다. 그림 6(a), 7(a)은 아무런 사전 정보 없이 이동하는 초기 iteration에서는 여러 방향으로 탐색하는 과정을 보여준다. 그러나 그림 6(b)에서와 같이 Q의 경우 약 400번의 iteration을 수행한 뒤에 원하는 상태 근처로 수렴하는 반면에, FQ의 경우 그림 7(b)에서와 같이 47번의 iteration후에 수렴하는 것을 볼 수 있다. 그러나 Q, RQ 양쪽 모두, 많은 iteration을 수행한 후에도 수렴하지 않는 경우가 있는데 이는 reward와 파라메터 (γ, α, ρ)설정이 잘못된 경우이다. 예로써, 상태 s에서 현재 행위 a1에 대한 reward를 r1이라고 하고, optimal 행위, a2의 reward를 r2라고 하자. Optimal 행위a2가 policy에 등록되기 위해서는 Q value가 가장 커야 하므로 (12)에서와 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 선택이 선행되어야 한다. 또한 Q와 RQ의 iteration 별 step수는 약 40번 내외로 수렴됨을 알 수 있었다. RQ-learning 시뮬레이션 결과 우수한 수렴 속도와 Q-learning보다 부드러운 행위 집합을 학습함을 알 수 있다. 이와 더불어 RQ-learning의 또 다른 특징인 적은 상태를 정의하고도 비슷한 성능을 발휘함을 그림 8에서 알 수 있다. 즉, 그림 8에서는 RQ-learning의 경우 적은 상태(각 축 당 17상태)을 정의하고도 원하는 상태로 수렴됨을 보여주고 있다. 이 경우에서도 iteration당 수렴 step수는 약 100에서 150정도에서 수렴함을 알 수 있었다. 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 표1에 나타내었다.

또한, 보답은 다음 (18)와 같이 정의 되었다.

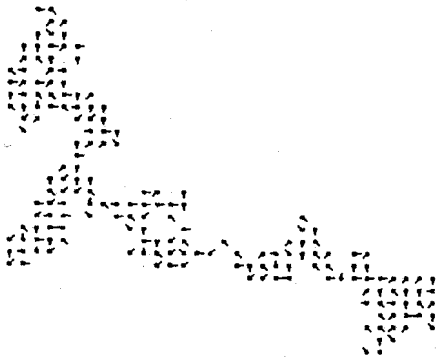
$$r = \frac{(\|x_{i+1} - x_g\| - \|x_i - x_g\|)}{K} \quad (18)$$

여기서, x_i 와 x_{i+1} 는 각각 현재상태와 다음 상태를 나타내며, x_g 는 최종 목표 상태를, K는 Robot팔이 최대 각속도로 움직여 변할 수 있는 상태 변이를 나타낸다. 본 시뮬레이션에서 목표 물체는 무작위로 움직이도록 하였다. 매 샘플링 시에 로보트는 본 논문에서 제안하였던 RQ-learning을 이용하여 이동하는 목표 물체 따라 움직이는

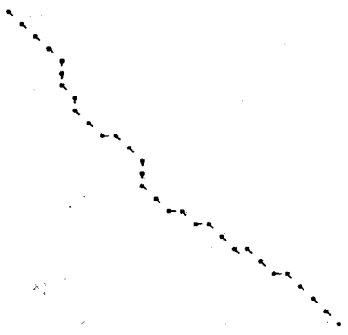
것을 학습하게 된다. 이러한 학습 후의 모습이 그림 11에 나타나있다. 본 시뮬레이션에서는 평균 1500~2000정도의 iteration을 거친 후 로보트가 최적의 행위를 수행함을 알 수 있었다.

표 1. 시각 추적 작업을 위한 상태 및 행위 변수.
Table 1. State and action variables for visual tracking task.

상태 및 행위 변수 (범위)	변수에 대한 설명
θ_{l1} (-30 ~ 210 degree)	로보트 1축의 각도
ΔP_x (-10 ~ +10 cm)	화면 좌표계의 중앙 위치와 물체 위치와의 x축 방향으로 차이
θ_{l2} (-30 ~ 210 degree)	로보트 2축의 각도
ΔP_y (-10 ~ +10 cm)	화면 좌표계의 중앙 위치와 물체 위치와의 y축 방향으로 차이
a_{l1} (-5 ~ +5 degree)	로보트 1축의 각속도
a_{l2} (-5 ~ +5 degree)	로보트 2축의 각속도

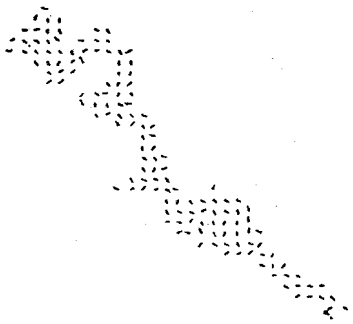


(a)

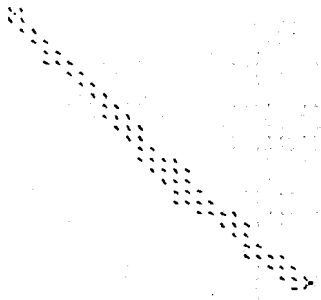


(b)

그림 6. The Q-learning 시뮬레이션 결과.
(a) 1st iteration, (b) 400th iteration.
Fig. 6. The Q-learning simulation result.
(a) 1st iteration, (b) 400th iteration.



(a)



(b)

그림 7. RQ-learning 시뮬레이션 결과.
(a) 1st iteration, (b) 47th iteration.
Fig. 7. The RQ-learning simulation result.
(a) 1st iteration, (b) 47th iteration.

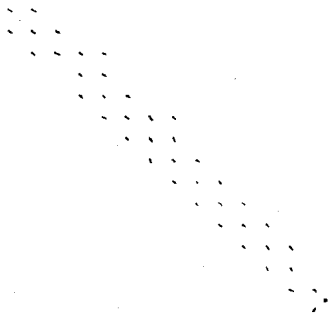


그림 8. 상태당 18 분해능을 사용한 RQ-learning 시뮬레이션 결과.
Fig. 8. The RQ-learning results using 18 resolution for each axis.

그림 9. Q-learning의 Iteration 별 평균step수.
Fig. 9. The number of averaging steps of Q-learning.

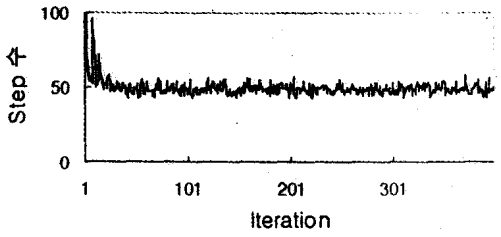


그림 10. RQ-learning의 Iteration별 평균step 수.
Fig. 10. The number of averaging steps of RQ-learning.

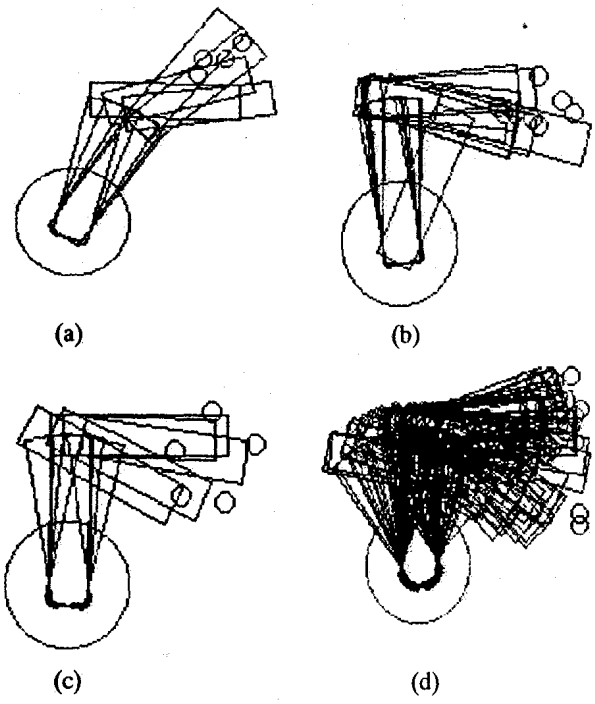


그림 11. Region-based Q-learning를 이용한 이동 물체 2-DOF SCARA 로보트의 시각 추적 시뮬레이션.
Fig. 11. Visual tracking of a 2-DOF SCARA robot by our proposed Region-based Q-learning technique.

V. 결 론

본 논문에서는 연속 상태 공간에서 각 상태에서의 적절한 연속된 행동 양식을 학습하기 위해서 Region-based Q-learning 알고리즘을 제안 하였다. 또한 이의 효율성을 증명하기 위해 기존의 Q-learning 알고리즘과의 비교 모의 실험을 수행하였으며 실제 상황과 동일하게 모델링된 로보트의 시각 추적(visual tracking) 시뮬레이션을 수행하였다. 시뮬레이션 결과, 본 논문에서 제안 하였던 방법은 연속 상태와 행위 공간을 가정하더라도 비슷한 수준의 수렴 속도를 나타내며, 특히, 환경에 대한 정보가 부족한 실제 상황, 즉 연속 공간에 대한 연속된 행위를 수행하여야 하는 시각 추적(visual tracking)과 같은 응용 분야에 효과적으로 적용될 수 있음을 보였다. 그러나 보답에 대한 영향이 동일한 함수에 의해 표현 되므로, 실제로 이러한 함수가 적절히 표현될 수 있는 영역을 온라인으로 구하는 방법에 대한 연구가 더 필요하고, 더 적절한 파라메터 조합을 이루기 위해서는 learning 상태에 알맞는 파라메터 설정 알고리즘에 대한 연구가 추가되어야 할 것이다.

참고문헌

[1] M. A. Salichs, E. A. Puente, D. Gachet and J. R. Pementel, "Learning behavioral control by reinforcent for an autonomous mobile robot," *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. of 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] P. Y. Glorennec, "Fuzzy Q-learning and dynamical fuzzy Q-learning," *IEEE Conference on R&A*, vol. 1, pp. 474-479, 1994.

[8] H. R. Berenji and Fuzzy Q-learning, "A new approach for fuzzy dynamic programming," *IEEE Conference on R&A*, vol. 1, pp. 486-491, 1994.

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

김 재 현

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

서 일 홍

1977년 서울대 공대 전자공학과 졸업. 1982년 한국 과학 기술원 졸업. 1982년 ~ 1985년 대우중공업 기술연구소 근무. 1987년 ~ 1988년 미국 미시간 대 객원 연구원. 현재 한양대 공학대 전자공학과 교수.