퍼지 보간 기법을 이용한 Fuzzy Q-learning

論 文
46~4~24

Fuzzy Q-learning using Fuzzy Interpolation Technique

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

Abstract - It is desirable for autonomous robot systems to posses the ability to behave in a smooth and continuous fashion when interacting with an unknown environment. Since Q-learning is normally used for optimizing a series of discrete actions, it may be undesirable when applied to a real environment which involves continuous states and actions. In this paper, we propose a new method of Q-learning that incorporates a fuzzy interpolation technique which is used to approximate a continuous state. Our learning method can estimate a current state by its neighboring states and has the ability to learn its actions similar to that of Q-learning. Thus, our method can enable robots to react smoothly in a real environment. Simulation results involving an autonomous robot are given to show the validity of our method.

Key Words : Autonomous robot system, Q-learning, fuzzy interpolation technique

1. 서 론

최근 들어, Control, Planning, Decision making 등을 실시간으로 처리하고자 하는 지능형 시스템(Intelligent system)에 관한 연구가 많은 관심을 끌게 되었다[1-6]. 이러한 여러 연구들에서 강화 학습(Reinforcement learning)과 같은 새로운 알고리즘의 개발과 Dynamic Programming과 같은 고전적인 알고리즘[7]의 개선이 동시에 이루어져 왔다. 특히, 강화 학습은 환경에 대한 충분한 지식 없이 주어진 환경에 적절한 행위를 학습할 수 있는 효과적인 방법으로 알려져 있다. 이러한 강화 학습 중 Tesauro는 Sutton의 Temporal Difference(TD) 방법을 개선하여 Backgammon- Playing Program에 적용하였다[8]. 특히, 이러한 TD 알고리즘과 밀접한 관련이 있는 Wakin의 Q-learning 알고리즘[10]은 최근 들어 집중적으로 연구되어 Lin은 이와 Backpropagation을 이용하여 로보트 제어에 적용하였다[11]. 한편, Watkin과 Dayan은 Q-learning의 convergence에 관한 결과를 발표하였으며[9] Moore와 Atkeson은 Q-learning의 수렴 속도를 높이기 위한 방법으로 Prioritized Sweeping technique을 개발하였다. 이와 더불어, 기존 Q-learning과 퍼지 이론을 접목시키고자하는 노력이 추진 되어왔다[12, 13]. 이러한 대부분의 기존 연구들은 불연속 상태 공간(discrete state space)과 불연속 행위 집합(discrete action set)을 기본으로 진행되어 왔다. 그러나, 실질적인 로보트 응용분야에서 이러한 알고리즘을 적용하기에는 여러 가지 한계를 극복하여야만 한다. (1)먼저, 너무 많은 기억공간을 필요로 하기 때문에 적용상 많은 어려움이 따르게 된다. (2)또한, 출력(행위)이 불연속이라는 제한을 갖게 된다. 예로써 Q-learning의 경우 각 상태에서 각 행위의 가치를 기록하기 위한 Q-table이 필요한데, 2-D 상태공간(state space)에서 각 상태축이 100개의 분해능(resolution)을 갖고, 100개의 가능한 행위를 가정하더라도 이를 위해 필요한 Q-table의 크기는 1,000,000개 이상의 기억 공간이 필요하게 된다. 따라서, 이러한 기억공간의 문제를 극복하면서도 기존의 Q-learning과 대등한 성능(즉, iteration time, learning rate)을 갖는 새로운 알고리즘이 요구된다. 또한 이러한 기억공간이 확보되었다 할지라도 Q-learning은 불연속된 출력을 학습하는 방법이므로 항상 해당 분해능에 반비례하여 최적의 행위에 대한 학습 오차가 존재하게 된다. 이러한 문제점을 해결하기 위해 퍼지의 Q-table을 새로운 퍼지 규칙들로 대체하여 각 Q값들을 퍼지추론에 의하여 생성하고, 각 규칙의 후건부 파라메타들을 Steepest descent 방법으로 조정하고자 하는 새로운 시도가 이루어졌다[14]. 그러나 이 경우 초기 규칙들의 생성이 어렵고, Steepest descent 방법을 사용함으로써 국부 극소점에 빠지는 경우 원하는 목표상태로의 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 규칙들의 후건부 파라메타를 조정할 때 현재의 Q값과 각 규칙으로 부터 생성되는 새로운 Q 값과의 차(Q)를 최소화하도록 하는 방법으로 비록 Q를 계산할 때 기존의 Q-learning의 Q 공식을 사용할지라도 Steepest descent 방법으로 갱신된 rule들이 어떠한 통계적인 내용도 포함하지 않으므로 이산공간에서 정의되었던 Q-learning을 연속공간으로 확장한 방법이라고 할 수 없다.

따라서, 본 논문에서는 기존 Q-learning을 연속공간으로 확장하면서도, 기존 Q-learning과 수렴속도 및 수행시간 등에서 대등한 성능(혹은 더 나은 성능)을 갖는 알고리즘을 개발 하고자 한다. 본 알고리즘에서는 먼저 각 상태에 대해 1개의 행위에 대해서만 최고치를 갖는 Q값 모델을 가정하고 퍼지보간(Fuzzy Interpolation)개념을 도입하여 연속된 상태와 연속된 행위간의 적절한 대응을 학습하게 된다. 즉, 기존의 Q-learning에서는 학습된 특정 상태에 가장 어울리는 행위에 관한 지식이 Policy Table에 기록되고, 실질적인 행위는 이러한 policy에 근

*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 : 1996年 10月 11日
最終完了 : 1997年 3月 5日

거하여 이루어졌지만, Fuzzy Q-learning에서는 Q-table에서 Policy Table을 얻는 기존의 방법을 연속된 Q값 모델에서 최대 Q값을 갱신하는 방법으로 대체하였다. 본 알고리즘은 다음과 같이 요약될 수 있다. 첫째, 현재 및 다음 상태의 Q값은 각 상태의 주변상태들로 부터 퍼지 소속도을 기반으로 하여 각각 구해지고, 둘째, 현재상태의 Q값 갱신에 Q-learning에서의 Q값 갱신식이 그대로 적용되며, 마지막으로, 현재상태에서의 Q가 각 주변상태들에 대한 Q에 대해 퍼지 소속도만큼 고려되어 각 주변상태들로 분배된다. 따라서, 주변상태의 Q값 보간(Interpolation)으로 현재상태의 Q값이 될 수 있다면, Fuzzy Q-learning에 의해 생성된 주변상태의 Q-table은 통계적인 Q값들을 포함할 수 있게 된다. 본문의 내용은 다음과 같다. 우선 Markovian Decision Process에 근거한 Watkin의 Q-learning을 2장에서 소개하고, 본 논문에서 제안하고자 하는 Fuzzy Q-learning을 3장에서 제시하고자 한다. 또한, Q-learning과 Fuzzy Q-learning의 비교 분석을 4장의 실험을 통해 보이고, 마지막으로 본 논문의 결론 및 향후 과제를 5장에서 제시하고자 한다.

2. Q-learning 알고리즘

2.1 Q-learning 알고리즘 소개

강화 학습은 최근 들어 아무런 선험 지식 없이도 학습할 수 있고, 높은 reactive 특성과 적응성이 뛰어난 행위를 생성하는 로보트 학습 방법으로서 많은 관심을 끌고있다. 그림 1은 기본적인 로보트와 환경과의 상호작용 모델을 나타낸다. 즉, Discrete Time Cyclic Processes에서 동작하는 유한 상태를 갖는 두개의 대행자들(환경과 로보트)로 모델링 될 수 있다. 먼저 로보트는 환경에 대한 현재상태를 감지하여 적절한 행위를 선택하여 이를 수행한다. 현재 상태와 수행된 행위에 근거하여 환경은 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. 우선 로보트는 서로 다른 유한 상태들의 집합 S를 감지할 수 있고, 유한 행위들의 집합 A를 행할 수 있으며, 또한 외부 환경은 목표상태(Goal State)로 수렴성이 보장되는 Markov Process로써 모델링할 수 있다고 가정하자. 여기서 $T(s, a, s)$를 현재 상태 s에 있는 로보트가 행위 a를 수행할 때 현재상태 s와 행위 a 사이의 관계로부터 현재상태 s가 다음 상태 s로 변하게 될 상태전이확률 이라 하며, 이 때, 각 상태와 행위에 대한 보답(reward)을 $r(s, a)$라 하자. 일반적인 강화 학습은 시간축에 걸쳐 얻어지는 보답의 합을 극대화하는 행위들의 책략(Policy)을 찾는 것으로 정의된다. 이러한 f는 S로부터 A로의 단순한 대입을 의미한다. 여기서 보답들의 합을 다음 식 (1)과 같이 정의한다.

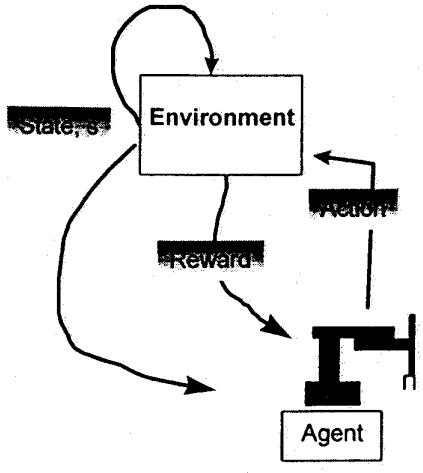


그림 1 로보트와 외부 환경과의 상호 작용 모델
Fig. 1 An Interaction model of robot and environment

$$\sum_{n=0}^{\infty} \gamma^n r_{p+n} \tag{1}$$

로보트가 상태를 감지하고 행동하는 1 반복공정(Cyclic Process)를 1 스텝(Step)으로 정의하고 시작상태로부터 출발하여 목표상태로 도달할 때까지의 과정을 1 이터레이션(iteration)이라 정의한다면, r_p는 로보트가 상태 s로부터 출발하여 행동책략 f를 따라 진행할 경우 어떤 스텝 p에서 받을 보답이라고 정의된다. 식 (1)에서 γ는 시간 축에 따라 감소하는 감쇠상수이며, 미래의 보답이 행동책략에 얼마만큼의 영향을 끼칠 것인가를 정하기 위해 사용된다. 대개는 1 이하의 값으로 정의된다. 이 때 만일 상태전이확률들과 각 상태전이에 대한 보답분포(Reward Distribution)를 미리 알 수 있다면 잘 알려진 Dynamic Programming[3]에 의해 최적의 행동책략(Policy)을 구할 수 있을 것이다. 그러나 이러한 정보를 미리 알 수 없으므로 Wakin은 통계적으로 적절한 행위를 배울 수 있는 Q-learning을 개발하게 되었다. $Q(s, a)$는 어떤 상태 s에서 행동 a를 취하고 이 후에 최적의 행동책략(Optimal Policy) f를 따르기 위한 응답값(Return Value) 혹은 행위값(Action Value)이라 하고, 다음과 같이 정의된다.

$$Q(s, a) = r(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \max_{a' \in A} Q(s', a')) \tag{2}$$

Wakin은 초기에는 T와 r에 대해 아는 바가 없으므로, 최적의 Q값으로 점증적으로 접근해가기 위해 온라인(on-line)으로 Q값을 산정함으로써 식 (2)의 Q값을 구하고자 하였다. 이러한 Q값의 갱신은 다음 식 (3)과 같이 정의되었다.

$$Q(s, a) = \alpha Q(s, a) + (1-\alpha)(r(s, a) + \gamma \max_{a' \in A} Q(s', a') \tag{3}$$

여기서 보답 r은 상태 s에서 행위 a를 수행하는 것에 대한 실질적인 보답값(Reward Value)이고 s는 다음상태를, $(0 < \alpha < 1)$는 학습속도를 각각 나타낸다. 여기서 Q값이 수렴함에 따라[5], 최종적인 수렴 상태에서 우변의 현재상태의 Q값과 좌변의 현재상태 Q값이 같으므로, 위의 Q값 산정 방법은 아래의 식 (4)로 풀어 쓸 수 있다.

$$Q(s, a) = r(s, a) + \gamma \max_{a' \in A} Q(s', a') \tag{4}$$

식 (4)는 Q-learning은 식 (2)에서의 T-matrix가 현재상태에서 다음상태로 전이되는 확률이 1이라는 가정 하에서 이루어진

방법임을 의미한다. 즉, 동일한 상태에서 특정 행위에 의해 나타날 수 있는 상태는 오직 하나라는 것을 의미며, 따라서, 현재 상태에서 어떤 행위에 의해 다음 상태로의 전이될 확률이 1인 환경에서 전체 행위값의 합을 최대화시키고자 하는 응용 분야에 적용될 수 있다는 것을 의미한다. 이러한 Q-learning은 다음과 같이 요약할 수 있다. 위에서 일단 Q값이 갱신되면 현재 상태에서의 각 상태에 대한 모든 행위의 Q값을 저장하고 있는 Q-table에서 해당 Q값을 식 (3)에 의해 교정하고, 교정된 Q-table를 바탕으로 Policy Table을 식 (5)와 같이 교정해야 한다.

$$f(s) \leftarrow a \quad such\ that \quad Q(s, a) = \max_{b \in A} Q(s, b) \qquad (5)$$

위에서 정의한 Q value 갱신 규칙을 기본으로 한 Q-learning 알고리즘은 다음과 같다.

Q-learning 알고리즘

(1) 초기화 :
  ① 난수 혹은 사전정보를 이용한 Q-table $Q(s, a)$ 초기화
  ② 초기화된 Q-table를 근거로 Policy Table 초기화
  ③ 여러 파라메타(γ, α, P)의 초기화
(2) 현재상태를 받아들임($s \leftarrow$ 현재상태).
(3) Policy Table로부터 현재상태에 해당하는 행위 a를 수행 (혹은 P만큼의 비율로 임의의 행위 수행)
(4) 수행된 행위에 대한 보답 r를 받음.
(5) 식 (3)를 이용하여 Q-table $Q(s, a)$ 갱신
(6) 식 (5)를 이용하여 Policy table $f(s)$ 갱신
(7) Goto step 2

3 Fuzzy Q-learning 알고리즘

Fuzzy Q-learning은 앞장에서 제시한 Q-learning을 연속 공간상으로 확장한 학습 방법이다. Q-learning에서처럼 공간상의 모든 상태에 대해 학습할 필요가 없고 몇 개의 대표적인 상태들에 대한 최적의 행위들을 학습하여 이의 Fuzzy Interpolation으로 임의의 현재상태에서의 행위를 추론하는 방법이다. 앞으로의 설명을 위해 다음 표 1과 같은 표기를 사용하기로 한다.

표 1 표기법
Table 1 Notation

$m_{c(n)i}$	현재(다음)상태가 i번째 주변상태로 소속될 소속도
a	현재(다음)상태에서 최대의 Q값을 갖는 행위벡터
$a_{c(n)i}$	현재(다음)상태의 i번째 주변상태에서 최대Q값을 갖는 행위벡터
$a_{jc(n)i}$	현재(다음)상태의 i번째 주변상태에서 최대Q값을 갖는 행위벡터의 j번째 축의 행위값
$s_{c(n)}$	현재(다음)상태
$s_{c(n)i}$	현재(다음)상태에서 i번째 주변상태벡터
$s_{jc(n)i}$	현재(다음)상태에서 i번째 주변상태벡터의 j번째 축의 상태값
$Q_{c(n)i}$	현재(다음)상태의 i번째 주변상태에서 최대의 Q값

3.1 연속 상태 공간에서의 Q-table

Q-learning에서, 가능한 모든 상태에서 가능한 모든 행위들의 행위값을 나타내는 것이 Q-table이었다. 따라서, 만일 연속된 상태와 행위공간에서 Q-learning을 수행하기 위해서는 이론적으로는 행위 수(무한대)의 Q-table이 상태 수(무한대) 만큼 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, 특정행위에서 최고치를 갖고 특정행위와 관계(거리)가 멀 수록 일정하게 Q-값이 작아지는 형태의 Q-table를 고려해 볼 수 있다. 행위간의 관계를 단지 행위 벡터 공간내의 유클리디안 거리로 정의하고, 현재상태의 Q-table를 특정행위에서 최고의 Q값을 갖고 거리에 비례적으로 단순 감소하는 특성이 있는 함수로 모델링하면, 특정상태에서 1차원의 모든 행위들에 대한 Q-table를 그림 2와 같이 Cone-shaped function으로 나타낼 수 있다.

일반적으로, Q-learning에서는 최대의 Q값을 갖는 행위가 여러 개인 경우, 이들 중 임의의 행위를 선택하여 학습하게 된다. 이와 같이, Fuzzy Q-learning에서도 최대의 Q값을 갖는 행위는 오직 하나라고 가정할 때, 특정 상태에 대한 최적의 특정 행위로의 대응이 가능하게 된다.

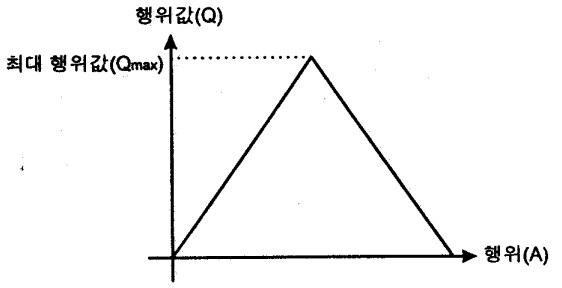


그림 2 행위값의 삼각 모델링
Fig. 2 A triangle model of action values

3.2 Fuzzy Interpolation을 이용한 Q-learning

상태공간 특성상 중간적인 상태에 대한 정의가 애매하므로 현재상태를 주변상태들의 Fuzzy Interpolation으로 정의 한다면, 주변에 존재하는 상태들로의 소속도에 의해 그림 3처럼 현재상태가 정의된다.

여기서, 소속도를 나타내는 소속 함수는 여러 함수가 적용될 수 있지만 자연적인 Interpolation을 수행하기 위해 종형 함수로 식 (6)과 같이 정의한다.

$$m_{cl} = e^{\alpha |s_{ci} - s_{cl}|} \qquad (6)$$

(여기서, α는 scale상수이고, sci는 i번째 주변상태벡터이며, sc는 현재상태벡터이다)

또한, 식 (6)을 이용하여 현재(다음)상태를 식 (7)로 정의할 수 있다.

$$s_c = \sum_{i=1}^{N} m_{ci} s_{cl} \qquad (7)$$

(여기서, N은 주변상태의 수)

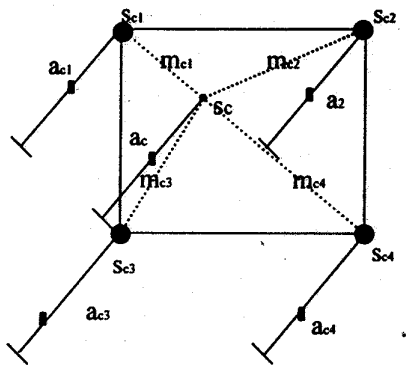


그림 3 주변상태들을 이용한 현재상태 추정
Fig. 3 A generation of a current state by using neighboring states

또한, 현재(다음)상태에서의 Q값, Qc(Qn)를 현재(다음)상태 및 현재(다음 step에)취해야 할 행위를 추정하는 것과 같이 Fuzzy Interpolation으로 정의한다면, 다음 식 (8)와 같이 나타낼 수 있다.

$$Q_c = \sum_{i=1}^{N} m_{ci} Q_{ci} \tag{8}$$

위의 Q값 정의를 이용하여 현재 상태에서 취해야 할 행위는 Q값을 최대화하는 값으로 정의하므로 식 (9)와 같이 나타낼 수 있다.

$$a_c = \max_{\forall a} \left(\sum_{i=1}^{N} m_{ci} Q_{ci}^{a} \right) \tag{9}$$

그러나 위의 식 (9)는 계산하기에 좀 까다로운 면을 갖추고 있다. 따라서 더욱 빠른 응답 속도를 요하는 시스템인 경우 최적은 아니지만 좀 더 직관적인 식 (10)을 사용할 수도 있다.

$$a_c = \sum_{i=1}^{N} m_{ci} a_{ci} \tag{10}$$

그림 4는 4개의 주변상태에서 나타난 최적의 행위와 이를 이용하여 현재상태에서 식 (10)에 의해 추론된 현재행위를 나타내고 있다.

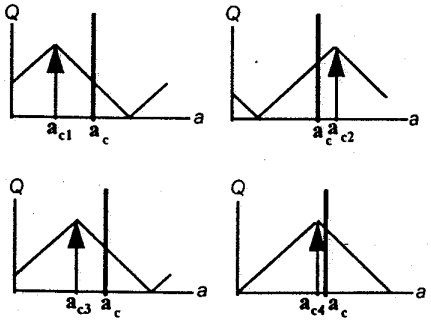


그림 4 현재상태의 적절한 행위 생성
Fig. 4 A generation of a current action

현재행위를 수행한 후, 환경은 현재상태에서 다음상태로 바뀌게 되고, 로보트는 외부로부터 보답(Reward)을 수신하게 된다. 수신된 보답을 이용한 Q값의 갱신은 다음 식 (11)과 같이 수행된다.

$$Q(t+1) = (1-\alpha)Q(t) + \alpha(r(t) + \gamma Q_n(t)) \tag{11}$$

(여기서, t는 t번째 iteration을 나타낸다.)

$$Q_c(t) = r(t) + \gamma Q_n(t) \tag{12}$$

수렴 후에 식 (11)은 식 (3)과 마찬가지로 식 (12)로 변환 할 수 있다. 따라서 임의의 상태에 대한 Q값은 보답들의 누적치를 나타냄을 알 수 있다. 식 (11)과 기존의 Q-learning에서의 Q값 학습 식(3)과의 차이는, 첫 번째로 Fuzzy Q-learning에서는 주변상태의 Q값을 통해 현재상태의 Q값을 구한다는 것과, 두 번째로 Fuzzy Q-learning에서는 다음상태의 모든 주변상태의 최대 Q값을 통해 추론된 다음상태의 최대 Q값을 의미하기 때문에 Q-learning에서 처럼 다음 상태에서의 최대 Q값을 따로 구할 필요가 없다는 것이다. 식 (9)혹은 식 (8)를 식 (11)에 대입하여 다음 iteration에서의 최대 Q값을 산정할 수 있다. 여기서 본 알고리즘의 핵심인 주변상태의 최적 행위 수정을 통한 현재상태의 행위 추측을 위해, 역으로 식 (11)에 의해 갱신된 현재상태에서 Q값 갱신 양을 기본으로 주변 상태의 Q값들 각각을 얼마만큼 갱신 시켜야 하는지를 알아낸다. 따라서 식 (13)에 의해 i번째 주변상태에 대한 Q값 갱신 정도를 현재상태에서의 Q값 갱신 정도에 소속도를 고려하여 추정한 후, 식 (14)을 이용하여 i번째 주변상태에서의 현재행위 a에 대한 Q값을 갱신한다.

$$dQ_{ci}^{a}(t+1) = m_{ci}(Q_c(t+1) - Q_c(t)) \tag{13}$$

$$Q_{ci}^{n}(t+1) = Q_{ci}^{n}(t) + dQ_{ci}^{a}(t+1) \tag{14}$$

이러한 갱신은 2가지 방식으로 이루어진다. 첫 번째로, (14) 식에 의해 갱신된 Q값이 현재의 최대 Q값 보다 작고 현재행위의 Q값보다 크면 최대 Q값이 현재의 행위로 이동하되 현재행위의 갱신된 Q값이 만족되는 행위위치로 이동한다. 또한 만일 교정된 Q값이 현재 Q값보다 작으면 최대 Q값이 현재의 행위의 반대쪽으로 이동하되 역시 갱신된 Q값이 만족되는 행위위치로 이동한다. 그림 6(가)에서 보는 바와 같이 갱신된 Q값이 현재의 최대 Q값 보다 작고 좋은 보답(> 0)을 받은 경우, 갱신된 Q값에 대해 적응하기 위해 Cone모양의 Q-table Model이 왼쪽으로 움직이는 Width Modification을 나타내며, 이는 식 (15)으로 표현된다.

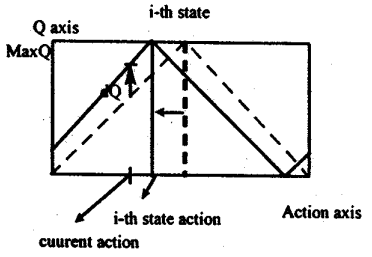
$$a_{ci}^{f}(t+1) = a_{ci}^{i}(t) + sgn(\cdot) \left\| 2 \frac{a'^{j}_{\max}}{Q_{\max}} dQ \right\| \tag{15}$$

여기서, $sgn(\cdot) = sgn(a_{ci}^{i}(t) - a_{i}^{j}(t))$dQ는 (13)식에 의해 구함

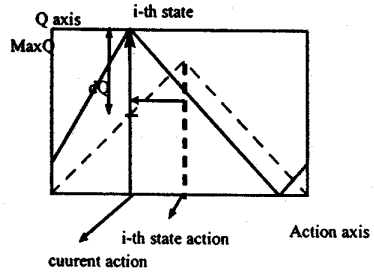
두 번째로, (14) 식에 의해 갱신된 Q값이 현재의 Maximum Q값 보다 큰 경우, 현재의 Action을 최대 Q값을 갖는 Action으로 수정하는 것이다. 그림 6(나)은 갱신된 Q값이 현재의

Maximum Q값 보다 큰 경우, 갱신된 Q값이 Maximum Q값이 되며, 이를 위해 Q-table Model의 대표행위가 현재행위로 움직이는 Height Modification(식 (15))을 나타낸다. 이 경우, 주변 상태에서의 최적행위는 현재행위가 된다. 이러한 행위의 교정은 N-차 공간 내에 정의되는 행위 벡터 공간의 각 행위축에 대해 정의되며, 각 행위축에 대해 이루어진다.

$$a_{\alpha}^{i}(t+1)=a_{c}^{i} \tag{16}$$



(a)



(b)

그림 5 Q값 갱신 방법
(a) 현재상태의 i번째 주변상태의 갱신된 Q값이 최대 Q값 보다 작은 경우의 폭조정
(b) 현재상태의 i번째 주변상태의 갱신된 Q값이 최대 Q값 보다 큰 경우의 높이조정

Fig. 5 Q-value modification method
(a) Width modification in case that the Q-value of ith neighboring state is lower than(or equal to) the maximum Q-value
(b) Height modification in case that the Q-value of ith neighboring state is larger than the maximum Q-value

이와 같이 갱신된 주변 상태들의 Q값에 근거하여 다음 행위를 취하게 된다. Fuzzy Q-learning 알고리즘은 다음과 같이 정리할 수 있다.

Fuzzy Q-learning 알고리즘

(1) 초기화 : 여러 파라메타초기화
(2) 현재상태을 받아들임(s←현재상태).
(3) 상태 s가 주변상태에 어느 정도 속하는지를 알 수 있도록 한다(Membership (mi)를 계산한다).
(4) 상태 s에서 실행해야 할 행위는 식 (9)에 의해 구한다 (때로는 Random action 수행).
(5) 결정된 행위를 수행하고, Reward를 받는다.
(6) 현재상태에서 Q값은 식 (11)를 사용하여 구한다.
(7) 주변상태들에서의 Q값은 식 (12), (13)를 사용하여 구한다.
(8) 현재상태에서의 최적의 행위를 식 (14), (15)을 사용하여 갱신한다.
(9) Goto Step 2

Fuzzy Q-learning은 Q-learning에 비해 다음과 같이 특징을 갖는다.

(1) 연속상태공간을 대상으로 한다.
(2) 모든 상태들을 정의할 필요가 없다(일정 수의 상태만을 정의하고 미정의 상태들은 주변상태들로의 소속 정도를 이용하여 추론한다).
(3) 연속된 행위 모델을 사용하여 정의된 상태에서는 최적의 행위만을 기억한다.
(4) 매번 행위를 선택할 때마다 모든 행위에 대하여 Maximum Q 값을 계산하여 그 Q 값에 해당하는 행위를 실행

4 모의 실험

4.1 Q-learning과 FQ-learning 알고리즘의 비교 모의실험

Q-learning과 Fuzzy Q-learning의 성능을 비교해보기 위해 자유 공간상에서 현재 위치에서 원하는 위치까지 도달하기 위한 policy를 구하는 모의실험을 실시하였다. 먼저 모의실험의 환경은 다음 표 2과 같다.

표 2 Simulation 환경 설정
Table 2 Simulation specification

Row state axis	0 - 350 state space
Column state axis	0 - 350 state space
Action	8방향 vector (Q) 연속 vector (FQ)
Sampling Time	0.032 sec

본 모의실험에서 사용된 reward r은 Move to goal을 기준으로 다음 식 (17)과 같이 정의한다

$$r=\|x-g\|-\|y-g\| \tag{17}$$

(여기서, 현재 position = x, 다음 position = y, 최종 position = g)

초기 위치를 (50, 50)으로 하고 최종 위치(320, 320)라 할 때, Q-learning의 경우 position 상태 공간을 각 축마다 35개의 분해능(resolution)으로 나누어야 목표 지점에 도달할 수 있다. 그림 6은 아무런 사전 정보 없이 이동하는 초기 수행에서는 여러 방향으로 탐색하는 과정을 보여준다. 그러나 그림 7, 8에서와 같이 Q의 경우 약 400번 반복 수행한 뒤에 원하는 상태 근처로 수렴하는 반면에, FQ의 경우 그림 9, 10, 11에서와 같이 47번의 수행후에 수렴하는 것을 볼 수 있다. 그러나 Q, FQ 양쪽

모두, 많은 학습 후에도 수렴하지 않는 경우가 있는데 이는 보답(reward)와 파라메터(Υ, α, P) 설정이 잘못된 경우이다. 예로써, 상태 s에서 현재행위 $a1$에 대한 보답를 $r1$이라고 하고, 최적행위, $a2$의 보답을 $r2$라고 하자. 최적행위 $a2$가 policy에 등록되기 위해서는 Q값이 가장 커야 하므로 식 (12)에서와 같이 갱신을 반복하는 과정에서 현재 Q값의 차이를 Υ, α 혹은P를 사용하여 극복하여야 한다. 즉, $\Upsilon(\alpha, P)$를 제외한 나머지 파라메터를 상수라고 하면 두 행위에 대한 Q값 갱신 속도는 $\Upsilon(\alpha, P)$에 비례하여 결정된다. 따라서 적절한 $\Upsilon(\alpha, P)$의 적절한 선택이 선행되어야 한다. 또한 Q와 FQ의 iteration 별 step수는 약 40번 내외로 수렴됨을 알 수 있었다. Fuzzy Q-learning 모의실험 결과 우수한 수렴 속도와 Q-learning보다 부드러운 행위 집합 을 학습함을 알 수 있다. 이와 더불어 Fuzzy Q-learning의 또 다른 특징인 적은 상태를 정의하고도 비슷한 성능을 발휘함을 그림 12에서 알 수 있다. 즉, 그림 12에서는 Fuzzy Q-learning의 경우 적은 상태(각 축당 17상태)를 정의하고도 원하는 상태로 수렴됨을 보여주고 있다. 이 경우에서도 이터레이션당 수렴 step수는 약 100에서 150정도에서 수렴함을 알 수 있었다. 마지막으로 그림 13, 14에서 각 알고리즘의 각 이터레이션별 평균 스텝(step)수를 보였다.

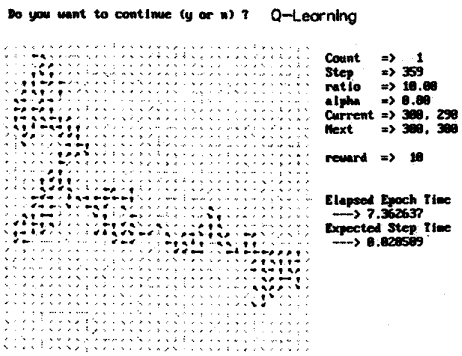


그림 6 1번째 iteration
Fig. 6 1'st iteration

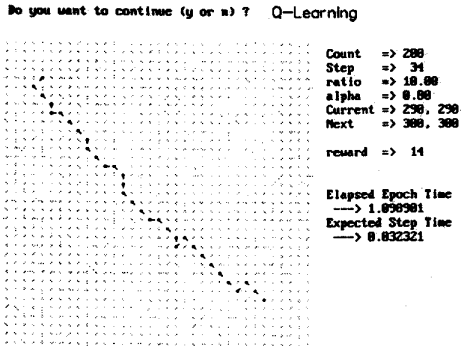


그림 7 200번째 iteration
Fig. 7 200th iteration

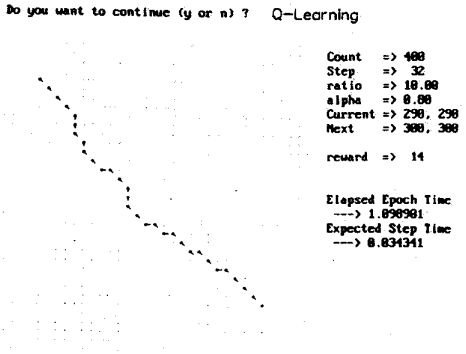


그림 8 400번째 iteration
Fig. 8 400th iteration

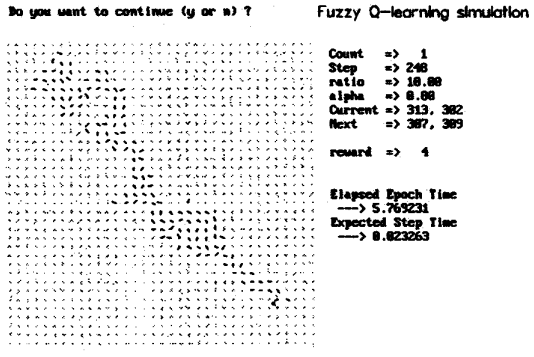


그림 9 1번째 iteration
Fig. 9 1'st iteration

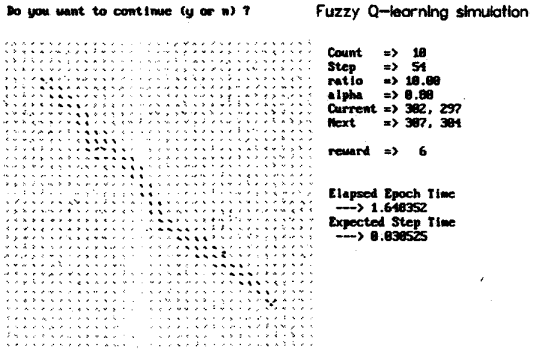


그림 10 10번째 iteration
Fig. 10 10th iteration

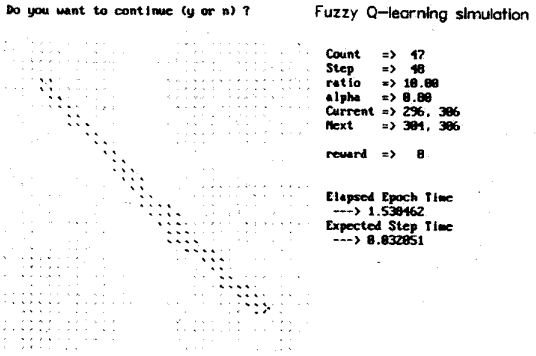


그림 11 47번째 iteration
Fig. 11 47th iteration

4.2 Box Canyon 예제

Fuzzy Q-learning을 적용하기 위한 일반적인 응용 분야는 상당히 다양할 수 있지만, 일반적으로 많이 알려져 있는 Box Canyon 문제를 모의실험 대상으로 사용하여 개발된 알고리즘의 유용성을 보이고자 한다. 먼저, 사용된 Reward는 다음과 같다.

$$\text{Reward} = \text{식(17)} - \text{Move step size} \quad (1-e^{-2Visit}) \qquad (18)$$

여기서 visit은 동일 상태를 탐색한 횟수를 내타냄.

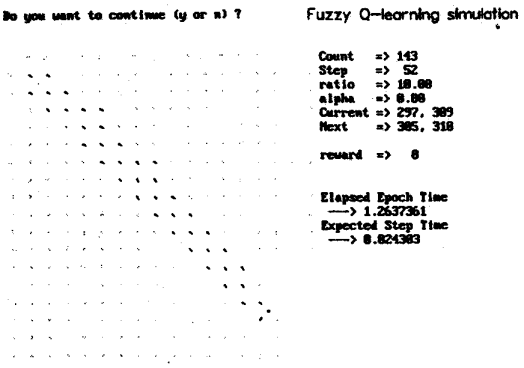


그림 12 각축의 resolution이 18인 경우
Fig. 12 In case of 18 axis resolution

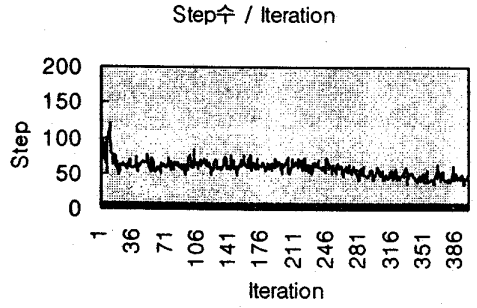


그림 13 Iteration 별 평균 step수(Q-learning)
Fig. 13 The step numbers for every iteration (Q-learning)

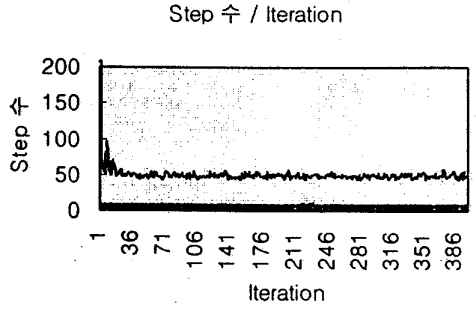


그림 14 Iteration별 평균 step수(FQ-learning)
Fig. 14 The step numbers for every iteration(FQ-learning)

초기에는 Go to goal reward, 식 (18)에 의해 목적지에 대한 reward 가 큰 작용을 하여, Box Canyon의 골짜기에 빠져 이리 저리 탐색하게 되지만(그림 15), 일단 상태들이 visit된 수가 높아지면 reward의 두번째 항이 증가하고, 따라서, 전체 reward는 상대적으로 낮아지게 된다. 궁극적으로는 그림 16에서 처럼 원하던 목적지에 도달하게 된다. 그림 17, 18은 각각 반복횟수가 4회, 8회 지난 후의 학습 상태를 나타내며, 8회 iteration후에 학습이 수렴됨을 나타낸다. 여기서 Move step size는 일정한 것으로 간주하였지만 실질적인 실험에서는

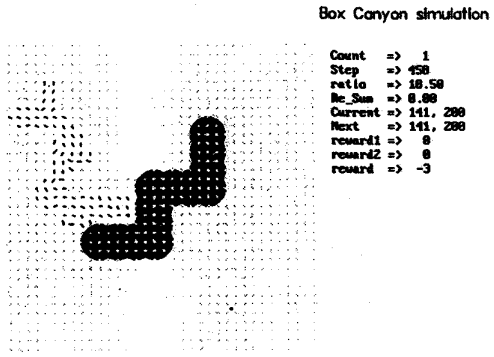


그림 15 1번째 Iteration의 전반부
Fig. 15 1'st iteration initial status

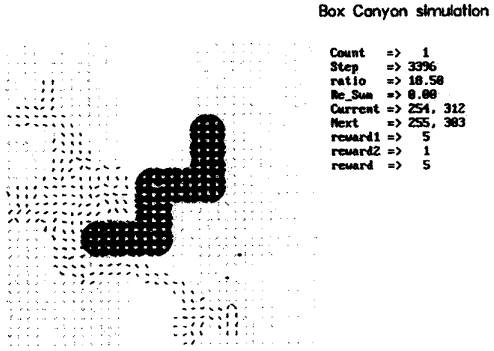


그림 16 1번째 Iteration의 후반부
Fig. 16 1'st iteration

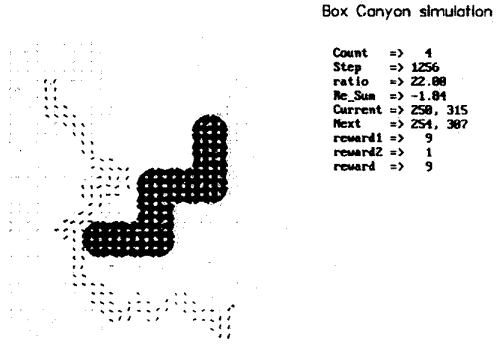


그림 17 4번째 Iteration
Fig. 17 4th iteration

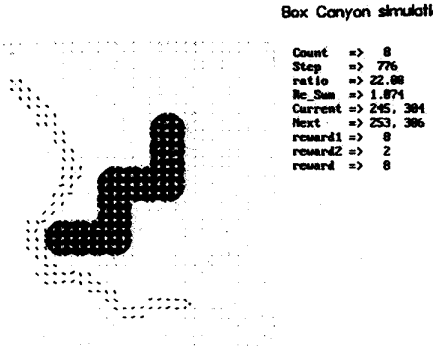


그림 18 8번째 Iteration
Fig. 18 8th iteration

Fuzzy Q-learning은 방향벡터를 생성하는 것과 더불어, 속도벡터의 크기는 일정한 sampling time내 가야할 거리로 정의하고 이를 또 다른 행위축으로 할당하여 행위벡터를 만든 후 Fuzzy Q-learning을 실시하여야 한다. 또한, 현재 위치로부터 목표물까지 혹은 장애물까지의 거리를 측정할 수 있는 초음파 sensor data 혹은 vision sensor data에 의해 구하고 이를 상태공간에 적용하여야 빠른 수렴을 보장할 수 있을 것이다

5 결론 및 향후 과제

본 논문에서는 연속 상태 공간에서 각 상태에서의 적절한 연속된 행동 양식을 학습하기 위해서 Fuzzy-Q learning 알고리즘을 제안하였다. 또한 이의 효율성을 증명하기 위해 기존의 Q-learning 알고리즘과의 비교 모의실험을 수행하였다. 본 연구는 앞으로 Reactive system 개발 등 많은 환경에 대한 정보가 부족한 상황에서 연속 공간에 대한 연속된 행위를 수행하여야 하는 응용 분야에 효과적으로 적용될 수 있을 것으로 판단된다. 향후에는 좀 더 적절한 파라메터 조합을 이루기 위해서는 learning 상태에 알맞은 파라메터들을 자동으로 결정하는 알고리즘에 대한 연구가 진행되어야 할 것이다.

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, IEEE Conf. On R&A, Vol. 1, pp. 1436-1441, 1993.

[2] M. Asada, E. Uchibe, S. Tawaratsumida, and K. Hosoda Coordination of Multiple Behaviors Acquired By A Vision-Based Reinforcement Learning, IEEE Conf. On R&A, Vol. 1, pp. 917-924, 1993.

[3] A. Barto, R. Sutton, C. Anderson Neuronlike elements can solve difficult learning control problems, IEEE Trans. On SMC, Vol. 13, Sept.83.

[4] A. Barto, R. Sutton, C. Watkins Sequential decision problems and neural networks, in , Advances in Neural Information Processing Systems 2, D. Touretzky, Ed. Morgan Kaufmann, San Mateo, CA, 1990.

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

[6] H. Berenji, Reinforcement learning and recruitment mechanism for adaptive distributed control,TR. IR/IRIDIA/92-4, Universite Libre de Bruxelles, 1992.

[7] R. Bellman. Dynamic Programming. Princeton University Press, Priceton, NJ, 1957.

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

[9] C. Watkins, P. Dayan, Q-learning, Technical Note, Machine Learning, Vol. 8, pp. 279-292, 1992.

[10] C. Watkins, Learning from delayed rewards, PhD Thesis, University of Cambridge, England, 1989.

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

[12] P. Y. Glorennec, Fuzzy Q-learning and Dynamical Fuzzy Q-Learning, IEEE Conf. On R&A, Vol. 1, pp. 474-479, 1994.

[13] H. R. Berenji, Fuzzy Q-learning : A new Approach for Fuzzy Dynamic Programming, IEEE Conf. On R&A, Vol. 1, pp. 486-491, 1994.

[14] T. Horiuchi A Fujino O. Katai T. Sawaragi, Fuzzy Interpolation-Based Q-learning with Continuous States and Actions, IEEE Conf. On Fuzzy Systems, Vol. 1, pp. 594-600, 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

서 일 홍 (徐 一 弘)

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