2003년도 대한전기학회 하계학술대회 논문집 2003.7.21-23

Acrobot제어를 위한 강화학습에서의 연속적인 행위 선택 알고리즘의 개발

서승환, 장시영, 서일홍
한양대학교 전자전기제어계측공학과
(+82-31-408-5802; FAX: 82-31-408-5803; E-mail:ihsuh@hanyang.ac.kr)

Development of reinforcement learning algorithm with countinuous action selection for acrobot

Sung Hwan Seo, Si Young Jang, Il Hong Suh
School of Electrical Engineering and Computer Science, Hanyang Univ.
(+82-31-408-5802; FAX: 82-31-408-5803; E-mail:ihsuh@hanyang.ac.kr)

Abstract - Acrobot은 대표적인 비선형, underactuated 시스템이며, acrobot의 제어목적에는 swing-up 제어와 balancing 제어가 있다. 이 두 가지 제어목적을 달성하기 위해 기존에 많은 연구가 진행되었다. 그러나 이 방법들은 두 개의 독립적인 제어기를 acrobot의 상태에 따라 전환하여 사용하는 방법으로서 전환 시점의 선정기준에 대한 어려움과 두 가지 제어목적의 달성을 위한 전체 학습 시간지연의 문제점이 있다. 이를 개선하기 위하여 우리는 acrobot의 두 가지 제어목적을 동시에 해결할 수 있도록 기존에 연구하였던 연속적인 상태공간의 근사화가 가능한 영역기반 Q-학습(Region based Q-Learning)[11]을 기반으로 한 하나의 제어기로 구현하는 방법을 연구하였다. 제안한 방법을 제작한 acrobot에 적용한 실험을 통하여 그 유용성을 검증하였다.

1. 서 론

Acrobot은 두 링크중 팔꿈치 부분에 단지 하나의 actuator를 갖는, 비선형 underactuated 시스템이다. 기존의 acrobot은 강화학습의 Sparse Coarse Coding[3], partial feedback linearization[1], energy based algorithm[2], fuzzy and adaptive fuzzy control[6] 등의 방법들로 제어해왔다. 이들 제어 방식들은 acrobot의 swing-up 제어와 balancing 제어를 각각 서로 다른 과제로 정의하여 서로 다른 제어기를 사용하여 제어하였다.

우리는 기존의 이런 방식들과 달리 두 가지 과제를 하나의 과제로 새로 새 정의하여 acrobot을 제어하려한다. 우리는 여기서 Sutton[3]의 강화학습을 적용하여 acrobot을 제어 하였고 여기에 연속적인 행위 선택을 위해 영역기반 Q 학습[11]의 μ의 개념을 도입하였다.

강화학습은 비선형 환경의 과제를 학습이라는 개념을 도입해 반복 수행을 통한 시행착오를 거쳐 우리가 원하는 목표에 수렴할 수 있도록 보상을 줌으로써 이런 비선형 시스템을 제어하는 방법들 중 하나이며 이 방법은 Watkin에 의해 수렴성이 증명되어졌다.

그러나 기존의 강화학습으로는 연속적인 행위의 선택이 가능하지 않기 때문에 서로 다른 행위의 크기를 가지는 두 가지 과제를 동시에 만족시키는 제어기를 구현할 수가 없었다. 그래서 우리는 이런 문제점을 해결하기 위해 기존에 제시되어진 영역기반 Q 학습[11]을 적용하여 연속적인 행위결정을 생성하려 한다.

영역기반 Q 학습은 기존의 불연속적인 공간과 행위에 대해 유클리드의 거리개념을 도입함으로써 미리 정의 되어진 상태들이 실제 상태에 얼마만큼 영향을 미치느냐에 따라 실제 상태의 행위를 결정하고 보상 역시 주변의 미리 정의 되어진 상태에 영향을 미친 만큼 보상을 줌으로써 좀더 연속적인 공간에 적용될 수 있는 방식이다. 이런 방식은 연속적인 공간에 대해 적은 기억공간으로 불연속적인 공간에 좀더 근사화 시킬 수 있는 방식으로 대표적인 방법으로는 Sparse Coarse Coding[3]에서의 타일링이 있다.

우리는 acrobot의 두가지 과제를 제어하기 위한 서로 다른 제어기를 갖는 시스템을 강화학습을 적용한 연속적인 공간에서 연속적인 행위의 표현이 가능한 재정의 된 하나의 과제를 제어하기 위한 단일 제어기를 갖는 시스템으로 통합하였다. 이 제어기의 알고리즘은 영역기반 Q 학습과 Sparse Coarse Coding방법을 혼합한 형태이며 이 알고리즘의 유용성을 acrobot 모의실험을 통하여 검증하고, 실제 제작한 acrobot에 적용하였다.

2. 본 론

2.1 Sparse Coarse Coding을 이용한 강화학습

우리는 기존에 사용되었던 강화학습 방법 중 Sparse Coarse Coding[3]방식을 적용하여 모의실험 하였고 연속적인 행위 생성을 위하여 영역기반 Q 학습의 방법을 행위 생성에 적용해 보았다. 여기서 각각의 알고리즘은 다음과 같다.

2.1.1 Q-학습

Q 학습은 강화학습에서의 대표적인 off policy TD 컨트롤 방식이다. Q 학습에서 행위의 설정은 식(2.1.1)과 같이 현재상태에서 의 최적행위와 다음상태에서의 예측되는 최적행위에 의해 결정되며 이에 따라 Q값은 갱신되고 새로운 정책이 수렴되게 된다.

$$Q(s_t,a) \leftarrow Q(s_t,a) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a)] \quad (2.1.1)$$

Q 학습을 간단히 설명하면 다음과 같다.

[Q 학습 알고리즘]

1. Q 테이블 및 각 파라미터들을 알맞게 초기화 한다.
2. 다음 내용을 반복한다.
   1) Q 테이블로부터 행위를 선택한다. 이때 임의의 비율로 랜덤 행위를 선택한다.
   2) 행위에 대해 다음 상태와 보답을 얻는다.
   3) 현재 상태의 행위 값을 갱신한다.
   4) 행위정책을 갱신한다.

2.1.2 타일 코딩

타일 코딩방법은 그림(2.1)과 같이 실제 연속 상태공간을 불연속 상태공간으로 근사화 하기 위해서 격자무늬의 타일들을 하나의 격자 단위 내에서 확률적으로 공평하게 엇갈려 배치시키는 방법이다.

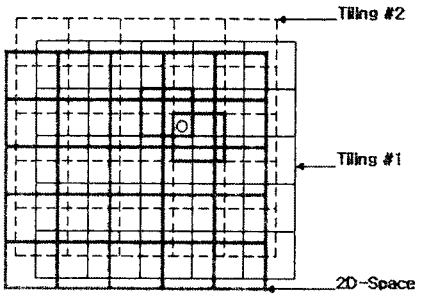

그림 2.1 타일 코딩

2.1.3 Eligibility Traces

로봇이 현재 위치한 상태에 대한 보상이 시간이 흘러감에 따라 적절한 비율로 감쇠보상 할 수 있도록 함으로써 λ만큼의 trace보답을 할 수 있도록 한다. trace에는 그림(2.2)과 같이 accumulating trace와 replacing trace 두 종류가 있다.

accumulating trace

$$e_t(s) = \begin{cases} r\lambda e_{t-1}(s) & \text{if } s \neq s_t \\ r\lambda e_{t-1}(s) + 1 & \text{if } s = s_t \end{cases} \quad (2.1.2)$$

replacing trace

$$e_t(s) = \begin{cases} r\lambda e_{t-1}(s) & \text{if } s \neq s_t \\ 1 & \text{if } s = s_t \end{cases} \quad (2.1.3)$$

그림 2.2 Eligibility Traces

2.1.4 Sparse Coarse Coding을 이용한 강화학습 알고리즘

위에 소개한 타일 코딩방식과 eligibility trace방식을 Q-학습에 적용하여 다음과 같은 Sparse Coarse Coding 알고리즘을 형성하게 된다.

[Sparse Coarse Coding 알고리즘]

1. 각각의 타일링들과 trace들 및 각 파라미터들을 알맞게 초기화 한다.
2. 다음 내용을 반복한다.
   1) 각각의 타일링들에서 현재상태에 위치한 타일들의 값을 통해 행위를 선택한다. 이때 임의의 비율로 랜덤 행위를 선택한다.
   2) 현재상태에 따라 trace들의 값들도 변화시킨다.
   3) 선택된 행위에 따라 현재상태를 갱신한다.
   4) 행위에 대한 다음 상태와 trace들에 의한 값을 통해 각각의 타일들에 보답을 얻는다.

2.2. 영역기반 Q-학습

타일 코딩과 마찬가지로 불연속 상태공간을 실제의 연속 상태공간과 일치시키기 위한 예측 기법의 하나이다. 여기서는 미리 설정되어 있는 Q-table에 대해 그림(2.3)과 같이 실제 상태와의 거리 값을 결정하는 μ값에 따라 각각의 주변의 Q값들이 미치는 영향을 계산하여 행위를 결정하고 보상 또한 거리 값에 따라 일정하게 행함으로써 좀더 연속적인 행위공간에 근사화 하였다. μ는 식(2.2.1)과 같이 이 정의되어 진다.

$$\lim_{d(s,s_i)\to 0} \mu_i \to 1 \qquad \mu_i = \frac{totaldis - d_i}{\sum_j (totaldis - d_i)}, \; totaldis = \sum_j d_i \quad (2.2.1)$$

기존의 $Q_i(Q_{i+1})$에 대한 적용과 보답은 식(2.2.2)과 같이 주변의 $Q_{i,j}(Q_{i+1,j})$값들을 실제 상태와의 거리 μ의 크기만큼의 합들로 형성된 $Q_i(Q_{i+1})$로 대치된다.

$$Q_i = \mu_{i,1} Q_{i,1} + \mu_{i,2} Q_{i,2} + \mu_{i,3} Q_{i,3} + \mu_{i,4} Q_{i,4} \quad (2.2.2)$$

그림(2.4)과 같이 각각의 행위에 대해 생성된 Q값에 따라 가장 적절한 행위를 선택하게 된다.
각각의 Q값에 대한 보상역시 식(2.2.3)과 같이 μ에 따라 일정한 비율로 보답한다.

$$Q'_{i,j} = \mu_{i,j} Q'_i \quad (2.2.3)$$

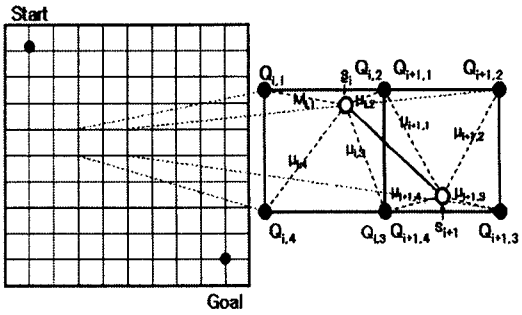

그림 2.3 영역기반 Q-학습

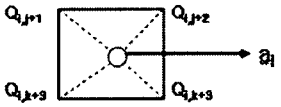

그림 2.4 행위 선택

2.3 영역기반 Q-학습을 이용한 연속적인 행위생성

우리는 지금까지 설명하였던 Sparse Coarse Coding방식과 영역기반 Q-학습방식을 이용하여 연속적인 행위생성을 위해 다음과 같은 알고리즘을 제안하였다.

영역기반 Q-학습방식을 Sparse Coarse Coding방식의 각각의 타일링에 적용하여 실제 정의된 행위들을 벡터 합을 통해 근사화 시킴으로써 연속적인 행위를 생성한다.

그림(2.5)과 같이 기존의 Sparse Coarse Coding에 적용한 타일링에서 현재 상태에 속해 있는 모든 타일들 중 몇 가지를 랜덤하게 추출하여 각각의 타일들에 영역기반 Q-학습을 적용하여 행위를 결정한다.

결정된 각각의 행위들은 벡터 합하여 그림(2.6)과 같이 적절한 행위를 얻는다.

각각의 보상역시 기존의 영역기반 Q-학습과 같은 방법으로 기여도에 따라 보상한다.

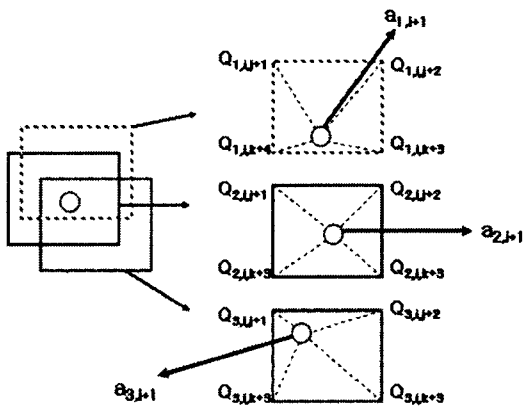

그림 2.5 행위선택

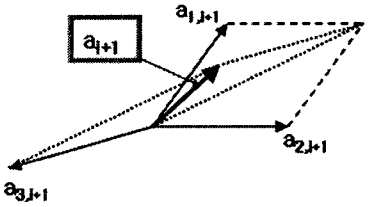

그림 2.6 각타일의 행위들에 의해 생성된 행위

2.3 실험 및 고찰

위 알고리즘을 시뮬레이션에 적용해보았다. 상태공간의 크기는 10×10 로 설정하고 이 공간을 1격자 크기 차이로 모두 덮기 위해 타일링의 크기는 11×11로 설정하였다. 타일링의 수는 10개로 설정하였다. trace는 replacing trace를 사용하였다. 행위의 종류는 기존에 -1, 0, +1에서 -0.33, -0.17, 0, 0.17, 0.33으로 5종류로 두어 -1~1까지의 연속적인 행위를 결정할 수 있도록 두었다. 각각의 파라메타들은 아래와 같이 설정하였다.

$$\lambda = 0.9 \qquad \varepsilon = 0.1 \qquad \alpha = 0.05\left(\frac{0.1}{m}\right) \; m- > tile\ num$$

도달 범위는 다음과 같이 balancing control이 가능한 영역까지 swing-up 할 수 있도록 제안하였다. 속도는 다음 영역에 들어섰을 경우 0이 되도록 하였다. 목표점의 파라메타는 다음과 같다.(각 축마다 10° 씩의 여유를 주었다)

$$\frac{\pi}{2} - 0.087 < \theta_1 < \frac{\pi}{2} + 0.087, \quad -0.087 < \theta_2 < 0.087,$$
$$\dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0$$

Balancing 제어는 도달 범위를 벗어난 경우 다시 수렴하도록 함으로써 지속적으로 보정하도록 하였다.

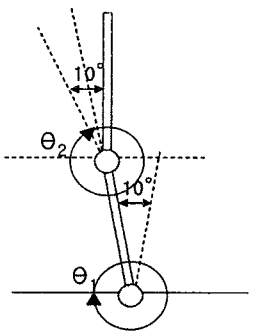

그림 2.7 도달 범위

2.3.1 모의실험 결과

수렴속도는 안정적 이였으나 하나의 에피소드를 위해서 상당히 많은 시행착오가 요구되는 것을 보았다. 어느 정도 수렴되어 가는 도중에 다소 긴 시행착오 후에 수렴하는 것은 이전에 학습하지 못했던 상태공간에 대해 학습하는 것을 나타낸다.

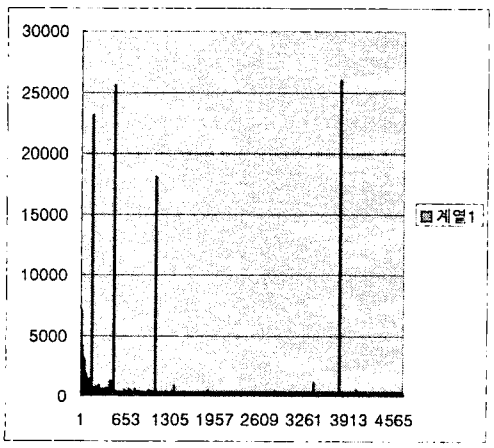

그림 2.8 모의실험

2.3.2 기구부의 구성

기구부는 사진(2.9)와 (2.10)과 같이 구성하였다.

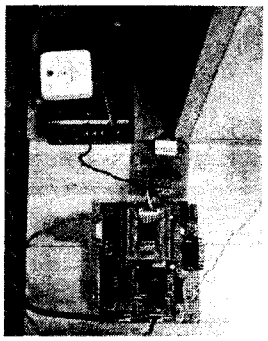

그림 2.9 제어부

그림 2.10 Acrobot

3. 결 론

기존의 acrobot은 swing-up과 balancing을 제어하기 위해 각각의 제어기를 두어 제어목적에 따라 전환하여 사용하였다. 이것은 실제 두 과제의 행위의 크기가 서로 달랐기 때문이다. 우리는 이런 문제를 영역기반 Q-학습의 연속적인 상태 공간의 근사화 기법을 행위 공간에 적용하여 연속적인 행위에 대한 근사화를 구현함으로써 swing-up시의 최대 입력과 balancing에서의 제어 입력이라는 상황에 따른 근사적인 행위의 크기를 벡터 합의 형태로 구현함으로 좀더 주변 환경에 대해 능동적인 행위선택이 가능하게 할 수 있었다.

기존의 강화학습보다 비교적 긴 스텝 후에 수렴하는 것은 계산시간뿐만 아니라 연속적인 행위 생성을 위해 기존에 3종류의 행위에서 적어도 5종류이상의 고정된 행위를 설정해 줘야 하기 때문이다. 많은 반복이 필요하기 때문에 실험을 행할 수 없는 로봇에는 적용하기 힘든 어려운 점이 있다.

추후 이 방법을 acrobot이 아닌 다른 모바일 로봇이나 어떤 환경에 따라 적절한 행위 생성이 필요한 시스템에 적용해보고자 한다.

[참 고 문 헌]

[1] Mark W. Spong, "The Swing up Control Problem for the Acrobot", IEEE Control systems, 1995.

[2] Mark W. Spong, "Swing Up Control of the Acrobot", IEEE, 1994.

[3] R. S. Sutton, "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding", IEEE, 1996.

[4] Gary Boone, "Efficient Reinforcement Learning : Model-based Acrobot control", IEEE Robotics and Automation, April 1997.

[5] Gary Boone, "Minimum-Time Control of the Acrobot", 1997.

[6] SCOTTC. BROWN and KEVIN M. PASSINO, "Intelligent Control for an Acrobot", Intelligent and Rbotic Systems, 18, 1997.

[7] J. Yoshimoto, M. Sato, "Application of reinforcement learning to balancing acrobot", IEEE, 1999.

[8] X.Lai, Z. Cai, "Fuzzy control strategy for acrobots combing model-free and model-based control", 1999.

[9] Xin Xin, masahiro KANEDA, "A new solution to the swing up control problem for the acrobot", Nagoya, July 2001.

[10] Xin XIN, masahiro KANEDA, "A robust control approach to the swing up control problem for the Acrobot", IEEE Intelligent Robotics and Systems, 2001.

[11] 김재현, 서일홍, "지능형 로봇 시스템을 위한 영역기반 Q-Learning", 제어자동화시스템공학 논문지, 제3권 제4호, 1997.8