Acrobot Swing Up Control을 위한 Credit-Assigned-CMAC-based 강화학습

論 文
53D-7-7

Credit-Assigned-CMAC-based Reinforcement Learning with Application to the Acrobot Swing Up Control Problem

張 時 榮* · 申 娟 溶** · 徐 承 煥*** · 徐 一 弘§
(Si Young Jang · Yeon Yong Shin · Seung Hwan Seo · Il Hong Suh)

Abstract - For real world applications of reinforcement learning techiniques, function approximation or generalization will be required to avoid curse of dimensionality. For this, an improved function approximation-based reinforcement learning method is proposed to speed up convergence by using CA-CMAC(Credit-Assigned Cerebellar Model Articulation Controller). To show that our proposed CACRL(CA-CMAC-based Reinforcement Learning) performs better than the CRL(CMAC- based Reinforcement Learnning), computer simulation and experiment results are illustrated, where a swing-up control problem of an acrobot is considered.

Key Words : CMAC, Credit-Assigned, Function Approximation, Reinforcement Learning, Acrobot

1. 서 론

Acrobot은 비선형성을 가지면서 말단장치(end-effector)의 위치제어를 위해 필요한 모터의 최소 개수(자유도)보다 모터의 개수가 부족한 underactuated 시스템이다. 그림 1.1에 Acrobot의 구조를 나타내었으며, 모터는 두 번째 조인트에만 연결되어 있다. Acrobot이라는 이름과 연구는 1991년 Murray 와 Hauser의 underactuated mechaniscal system 연구[1]로부터 시작하였으며, 그 후에도 Acrobot 제어를 위한 많은 연구가 진행되었다.

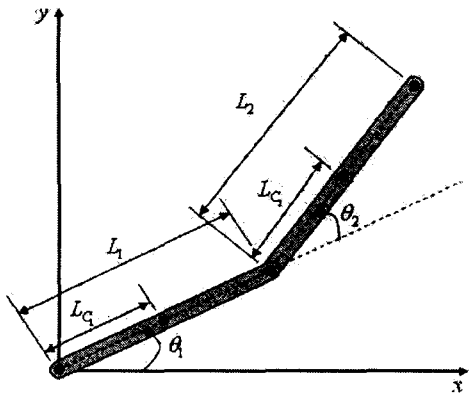

그림 1.1 Acrobot의 구조
Fig. 1.1 Structure of the Acrobot

* 正 會 員 : 漢陽大 電子電氣制御計測工學 博士過程
** 學生會員 : 漢陽大 電子電氣制御計測工學 碩師過程
*** 學生會員 : 漢陽大 메카트로닉스工學 碩師
§ 正 會 員 : 漢陽大 情報通信大學院 教授
接受日字 : 2004年 3月 1日
最終完了 : 2004年 5月 18日

Acrobot의 제어목적에는 말단장치를 일정 위치 이상으로 올리는 swing-up 제어와 inverted pendulum의 제어와 유사하게 두개의 링크를 수직으로 세워서 균형을 잡는 balancing 제어가 있다. Acrobot 제어를 위한 대표적인 연구로는 M. W. Spong의 partial feedback linearization[3]을 이용한 swing-up 제어 방법이 있으며, 그 후 이를 개선한 A. D. Luca, G. Oriolo의 Iterative State Steering 기법을 이용한 robust feedback control 방법[8]이 있다. swing-up 제어를 위한 또 다른 방법으로는 G. Boone의 Acrobot의 운동방정식과 energy 방정식을 이용한 energy based 제어방법[5], S. C. Brown 과 K. M. Passino의 adaptive fuzzy control 방법을 이용한 balancing 제어 방법[9] 등이 있다.

Acrobot에 대한 동역학 모델을 계산하고, 이 모델에 기반하여 제어하는 Model-based 방법과는 달리 Acrobot에 대한 모델을 계산할 수 없다는 가정 하에서 이를 학습을 통하여 배워나가며 제어하는 Model-free 방법으로는 R. S. Sutton의 Sparse Coarse Coding 기법을 이용한 강화학습 방법[6]이 있다.

강화학습(Reinforcement Learning)이란 로봇이 미지의 환경에서 행동과 보답을 주고 받으며, 임의의 상태에서 가장 적합한 행위를 학습하는 방법이다. 강화학습을 이용하여 주어진 문제를 해결하기 위해서는 1)상태공간과 행위공간의 정의, 2)임의의 상태에서 적절한 행위를 선택하기 위한 행위책략의 정의, 그리고, 3)환경에서 받은 보답으로부터 행위함수를 학습시키는 방법의 정의가 필요하다.

강화학습을 실제환경에 적용 시 문제점으로는 연속 상태공간을 불연속 상태공간으로 표현하면서 생기는 많은 용량의 메모리(curse of dimensionality)와 이에 따른 긴 학습시간이 필요하다는 것이다. 이를 해결하기 위한 방법으로 함수 근사화(Function Approximation) 또는 일반화(Generalization)가 있으며, 이 방법은 연속 상태공간 상의 임의의 상

태를 불연속 상태공간 상의 상태들의 조합으로 근사화하는 방법이다. 함수 근사화에 대한 연구로는 불연속 상태공간 상의 상태들의 행위함수의 선형 벡터 조합으로 실제 상태를 표현하는 Region-based Q-Learning 방법[10], 실제 상태의 행위함수를 불연속 상태들의 행위함수들의 Local Average 기법을 이용하여 계산하는 Kernel-based Reinforcement Learning 방법[11], 그리고, Sutton의 Sparse Coarse Coding을 이용한 강화학습방법 등이 있다. Sutton은 J. A. Boyan과 A. W. Moore가 강화학습에는 함수 근사화를 적용하기 어렵다고 주장한 논문[4]을 반박하면서 CMAC(cerebellar model articulation controllers)과 같은 구조를 이용한다면 강화학습의 상태공간을 함수 근사화하는게 가능하다고 주장하고, CRL(CMAC-based 강화학습)방법을 제안하였다. 함수 근사화를 이용한 강화학습에 관한 많은 연구들이 진행되어 왔음에도 불구하고, 실제 로봇에 강화학습을 적용하기에는 긴 수렴시간이 걸리며, 이를 줄이기 위한 연구는 강화학습이 실제 로봇 속에서 자리매김하는데 필수적이라 할 수 있다.

본 논문에서는 Sutton의 CRL 방법의 수렴속도를 개선하기 위한 연구로 S. Su, T. Tad, 그리고, T. Hung이 제안한 CA-CMAC(Credit-Assigned CMAC) 기법[7]을 이용한 **CACRL**(Credit-Assigned CMAC-based 강화학습) 방법을 제안하고, 그 수렴속도 개선 성능을 CRL 방법과 비교하였으며, 마지막으로 그 유용성을 Acrobot을 적용한 실험을 통하여 보였다.

논문의 구성은 2장에서 Convetional CMAC 기법, Credit-Assigned CMAC 기법, 그리고, 제안하는 CACRL에 대해서 설명하고, Sutton의 CRL 방법 대한 제안하는 CACRL 방법의 수렴속도 개선 성능을 보이기 위하여 3장에서는 모의실험 결과를 4장에서는 실험 결과를 설명하였다. 마지막으로 5장에서 논문의 결론을 맺었다.

2. Convetional CMAC, CA-CMAC, and CACRL

2.1 Conventional CMAC

CMAC(the cerebellar model articulation controller)는 인간의 소뇌에서 연상 작용을 모델링한 것으로 1975년 J. S. Albus에 의해 제안되었다. CMAC의 출력방법과 학습방법을 설명하기 위하여 그림 2.1에서 이차원 상에서의 CMAC의 구조를 도시하였다.

그림 2.1은 두개의 입력을 받아서 한개의 출력을 내는 이차원 CMAC이다. 입력은 각각 7개의 값을 가질 수 있으며, 따라서 표현 가능한 실제 상태의 총 개수 49이다. CMAC의 구조는 상태공간을 덮는 층(floor)와 그 층를 겹치지 않게 나눈 타일(tile)들로 구성되어 있으며, 이때 타일의 개수는 각각의 입력 축을 몇 개의 조각으로 나눌 것인가에 따라 결정된다. 그림 2.1의 예는 3개의 층을 갖고, 각 입력 축을 3개의 조각으로 구분하여 한 층을 덮는 타일의 개수를 9개로 한 경우이다. CMAC의 출력은 state(3,3)을 예로 들 경우 첫 번째 층에서는 Hh 타일, 두 번째 층에서는 Ee 타일, 그리고 세 번째 층에서는 Bb과 연관되어 계산된다. CMAC의 장점은 많은 상태를 적은 상태공간만으로 표현할 수 있다는데 있다(실제 상태의 수가 증가할 수록 장점 부각).

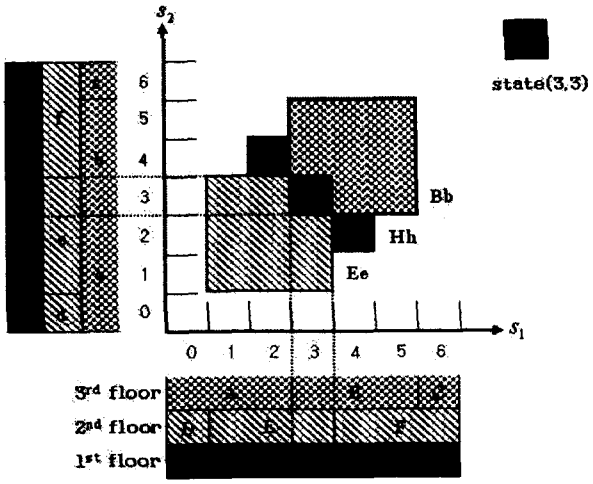

그림 2.1 이차원 CMAC의 구조
Fig. 2.1 Structure of 2-Dimensional CMAC

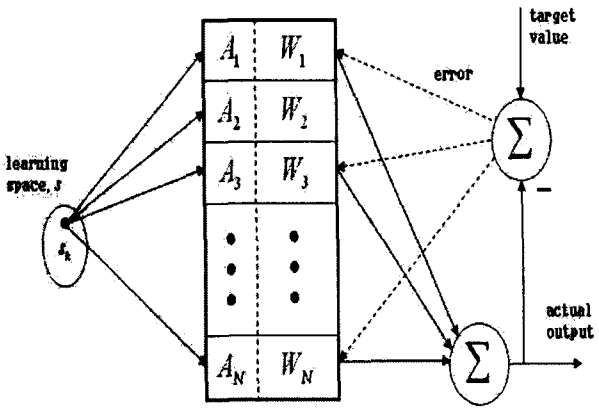

그림 2.2 CMAC의 동작
Fig. 2.2 Operation of CMAC

CMAC의 동작과정은 그림 2.2와 같으며 출력은 다음 수식에의해 결정된다.

$$y_{s_k} = \sum_{j=1}^{N} c_{s_k,j} w_j \qquad (2.1)$$

수식 (2.1)에서 N은 총 타일의 개수이며, w_j는 j-번째 타일에 저장된 데이터이며, $c_{s_k,j}$는 j-번째 타일이 현재 상태 s_k와 연관되어 있으면 1, 아니면 0을 나타내는 인덱스 이다. 임의의 상태에 대하여 연관된 타일은 각 층에 하나씩만 존재하므로 CMAC의 출력 값은 현재 상태와 연관된 층의 개수 만큼의 연관된 타일들의 저장된 데이터합으로 계산된다.

반면, CMAC의 학습(타일 내에 저장된 데이터의 갱신)은 다음 수식과 같다.

$$w_j^{(i)} = w_j^{(i-1)} + \frac{\alpha}{m} c_{s_k,j} \left(\bar{y}_{s_k} - \sum_{j=1}^{N} c_{s_k,j} w_j^{(i-1)} \right) \quad (2.2)$$

수식 (2.2)에서 w_j의 윗첨자 (i)는 반복(iteration) 수를 의미하며, m은 층의 개수, α는 학습율, $\bar{y}_{s_k}$는 임의의 상태 s_k에서 목표 값(target value)를 나타낸다. 따라서 (i)번 반복에서 w_j값의 갱신은 (i-1)번 반복에서의 w_j값 더하기, 목표 값과 출력 값의 에러를 타일의 개수로 나누어 학습율을 곱한 값에 의해 결정된다.

2.2 CA-CMAC[7]

CA-CMAC 방법은 S. Su, T. Tao, 그리고 T. Hung에 의해 제안되었다. S. Su 등은 CMAC의 수렴속도가 지연되는 이유 중의 하나를 수식 (2.2)에서와 같이 임의의 상태에 연관된 타일들의 데이터 갱신 값을 동등하게 두는데 있다고 생각하였다. 연관된 타일 중 많은 학습을 통하여 신뢰도가 높은 타일도 있을 수 있고, 반대로 신뢰도가 낮은 타일도 있을 수 있다. 현재의 출력 오차로부터의 각 타일의 갱신 비율은 신뢰도에 따라 차등을 두어야함에도 불구하고, 신뢰도가 낮은 타일의 영향으로 인한 현재의 출력오차를 동등한 비율로 신뢰도가 높은 타일의 갱신에 적용함으로써 CMAC의 비효율적인 학습으로 인한 수렴지연현상이 일어난다. 이를 해결하기 위하여 S. Su 등은 다음과 같은 가정 하에 CA-CMAC 방법을 제안하였다.

> **가정:**
> 많은 학습을 한 타일 내의 데이터 일수록 더욱 정확한 데이터를 가지고 있다.

CA-CMAC의 타일 내의 저장된 데이터의 갱신 수식은 다음과 같다.

$$w_j^{(i)} = w_j^{(i-1)} + \alpha c_{s_k,j} \left\{ \frac{\left(f(j)+1\right)^{-1}}{\sum_{l}^{m} \left(f(l)+1\right)^{-1}} \right\} \left(\bar{y}_{s_k} - \sum_{j=1}^{N} c_{s_k,j} w_j^{(i-1)} \right) \quad (2.3)$$

수식 (2.3)에서 $f(j)$는 j-번째 타일의 학습된 횟수이다. 수식 (2.2)와 수식 (2.3)의 다른 점은 갱신값의 1/m 부분이 $\left(f(j)+1\right)^{-1} / \sum^{m} \left(f(l)+1\right)^{-1}$으로 대치된 것이다. 임의의 상태와 연관된 m개 타일의 데이터들의 학습된 횟수가 모두 같거나 학습이 한번도 이루어지지 않았다면 수식 (2.2)와 수식 (2.3)은 같은 값을 갖으며, 학습된 횟수가 다를 경우에는 학습된 횟수가 많은 타일의 갱신 비율은 1/m 보다 작게 반대로 학습된 횟수가 적은 타일은 갱신 비율이 1/m보다 크게 된다.

2.3 CACRL

CRL은 연속 상태 공간에서의 Acrobot swing-up 제어를 위해서 CMAC 구조를 적용한 강화학습방법이다. Sutton은 CRL의 학습속도 개선을 위하여 Temporal Difference 학습과 Monte Carlo 방법의 장점을 취한 Sarsa(λ)와 Watkins의 Q(λ) 방법을 이용하였고, 이 방법들은 eligibility trace라는 변수를 도입하여 현재 상태로부터 앞으로 몇 step 후의 상태들의 기대치를 현재의 행위 함수 갱신에 이용한다.

Sutton은 미래의 기대치(Return)를 얻어서 함수를 갱신하는 것과 eligibility trace를 이용하여 과거의 값으로부터 함수를 갱신하는 것이 off-line backup일 경우 등가임을 증명하였다.

그러나, eligibility trace 방법이 미래의 상태들에 대한 누적된 가중치를 이용하여 credit을 assign하기는 하지만, 연속 상태에서 CMAC을 이용할 경우, 행위함수를 계산하는데 참조되는 타일들의 갱신에 credit을 assign하는 개념과는 구별된다. 기존의 CRL 방법의 타일의 갱신수식은 다음과 같다.

$$\vec{\theta} \leftarrow \vec{\theta} + \frac{\beta}{m} \delta \vec{e} \quad (2.4)$$

$\vec{\theta}$는 CMAC 내의 모든 타일의 벡터이고, β는 step 크기 비율, m은 타일의 개수, δ는 현재 행위함수의 예측되는 오차, 그리고, $\vec{e}$는 모든 eligibility trace를 나타내는 벡터이다. 수식 (2.4)에서 $\vec{e}$는 행위함수간의 credit assignment개념을 갖는 백터로서 현재의 상태와 연관된 타일들의 합으로 행위함수를 계산하고, 이 행위함수로 인하여 발생하는 예측오차를 다시 연관된 타일들의 가중치 학습에 분배하는 비율과는 구분된다. CRL방법에서 이용하는 타일들의 갱신 가중치는 1/m로 일정하게 분포된다.

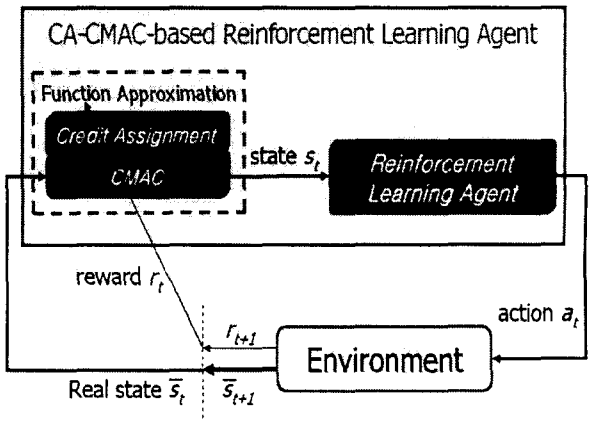

그림 2.3 제안하는 CACRL의 개념도
Fig. 2.3 Conceptual diagram of CACRL

본 논문에서는 행위함수의 예측오차를 다시 연관된 타일들에 학습시킬 때 eligibility trace 뿐만 아니라 CA-CMAC 방법을 이용하여 갱신 가중치를 두어 학습하는 방법으로 다음과 같은 갱신 수식을 제안한다.

$$\theta_i(s,a) \leftarrow \theta_i(s,a) + \beta \left\{ \frac{\left(L_i(s,a)+1\right)^{-1}}{\sum_{k=0}^{m}\left(L_k(s,a)+1\right)^{-1}} \right\} \delta e_i(s,a) \tag{2.5}$$

수식(2.5)에서 s는 연속상태공간 내의 현재의 상태이고, $\theta_i(s,a)$는 s상태에서의 행위 a에 대한 행위함수를 계산하는데 참조되는 i번째 타일의 값이며, $e_i(s,a)$와 $L_i(s,a)$는 각각 그 타일에 대한 eibilility trace와 갱신횟수이다. CMAC내의 모든 타일의 갱신을 위하여 s는 현재의 실제 상태에 대하여 타일의 크기 간격으로 확장한 가상의 불연속상태 공간의 집합에 속하는 상태로 정의 하였다.

2.4 Remarks

CACRL의 Credit Assign 기법은 한 episode 동안의 타일들의 가중치를 효율적으로 분배하여 학습 초기에 무작위로 학습되어 있지 않은 타일 내의 데이터들의 학습 속도를 향상시켜주는 역할을 한다. Credit Assign은 한 epdisode동안에 국한된 국부적인 가중치 분배 기법이다. 따라서, 일정 episode 후의 Credit Assign적용은 오히려 수렴속도를 감쇠시키며, 수렴단계에서는 더욱이 수렴을 지연시키는 효과를 나았다.

2.5 CACRL 알고리즘

Initialize $\vec{\theta}$ arbitrarily and $\vec{e} = \vec{0}$
Set n_p
Repeat (for each episode):
  $s \leftarrow$ initial state of episode
  For all $a \in A(s)$:
    $F_a \leftarrow$ set of features present in s,a
    $Q_a \leftarrow \sum_{i \in F_a} \theta(i)$
  Repeat (for each step of episode):
    With probability 1 - ε :
      $a \leftarrow \arg\max_a Q_a$
      $\vec{e} \leftarrow \gamma\lambda \vec{e}$
    else
      $a \leftarrow$ a random action $\in A(s)$
      $\vec{e} \leftarrow 0$
    For all $i \in F_a$: $e(i) \leftarrow e(i) + 1$, Update $L_i(\cdot)$
    Take action a, observe reward r, and next state s'
    $\delta \leftarrow r - Q_a$
    For all $a \in A(s')$:
      $F_a \leftarrow$ set of features present in s',a
    $Q_a \leftarrow \sum_{i \in F_a} \theta(i)$
    $a' \leftarrow \arg\max_a Q_a$
    $\delta \leftarrow \delta + \gamma Q_{a'}$
    if episode $\leq n_p$

$$\theta_i(\cdot) \leftarrow \theta_i(\cdot) + \beta \left\{ \frac{\left(L_i(\cdot)+1\right)^{-1}}{\sum_{k=0}^{m}\left(L_k(\cdot)+1\right)^{-1}} \right\} \delta e_i(\cdot)$$

    else
    $\vec{\theta} \leftarrow \vec{\theta} + \beta \frac{1}{m} \delta \vec{e}$
  until s' is terminal

3. Acrobot 모의 실험

3.1 모의 실험 환경

Acrobot의 swing up 제어를 위한 학습 파라미터로써 eligibility trace 감쇠율(λ)은 0.9, step 크기(β)는 0.05, 랜덤 비율(ε)은 0.0, 감쇠율(ϒ)은 1.0으로 각각 설정하였다. Acrobot의 상태공간은 조인트 1의 각도와 각속도, 조인트 2의 각도와 각속도로 표현되는 4차원의 연속상태공간이며 이를 함수 근사화하기 위하여 10 층의 9×9×9×9 타일을 갖는 구조의 CMAC을 적용하였다.

$$\begin{aligned}\begin{bmatrix}0\\ \tau_2\end{bmatrix} &= \begin{bmatrix} m_2L_2^2 + 2m_2L_1L_2c_2 + (m_1+m_2)L_1^2 & m_2L_2^2 + m_2L_1L_2c_2 \\ m_2L_1L_2c_2 + m_2L_2^2 & m_2L_2^2 \end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} \\ &+ \begin{bmatrix} -m_2L_1L_2s_2\dot{\theta}_2^2 - 2m_2L_1L_2s_2\dot{\theta}_1\dot{\theta}_2 + m_2L_2gc_{12} + (m_1+m_2)gL_1c_1 \\ m_2L_1L_2s_2\dot{\theta}_1^2 + m_2L_2gc_{12} \end{bmatrix} \\ &= M(\underline{\theta})\ddot{\underline{\theta}} + V(\underline{\theta},\dot{\underline{\theta}}) + G(\underline{\theta})\end{aligned} \tag{3.1}$$

모의 실험에서 사용한 acrobot의 운동방정식은 수식(3.1)과 같으며 조인트 2에 가해지는 토크를 강화학습의 행위로 설정하였다. 현재의 행위에 대하여 운동방정식(3.1)을 풀어서 조인트1, 조인트2의 각가속도에 해당하는 해를 구한 후 이를 Runge Kutta방법으로 적분하여 각각의 조인트의 각속도와 각도를 구하였고, 이렇게 구한 값들을 강화학습의 4차원 상태로 매핑시켰다.

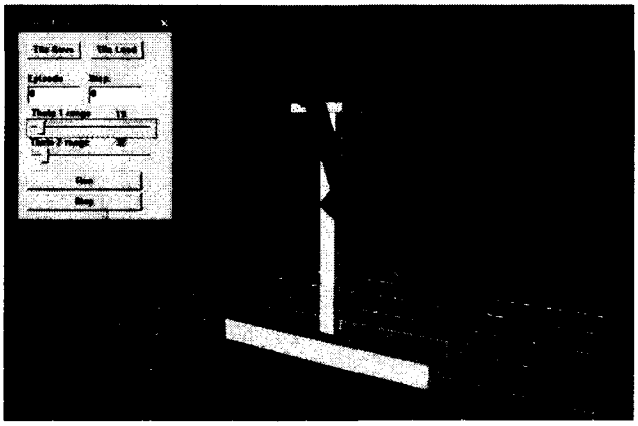

그림 3.1 모의 실험 환경
Fig. 3.1 Simulation Environment

그림 3.1은 모의 실험 환경이며 이때 acrobot의 swing-up 제어를 위한 목표 위치의 범위는 그림1.1에서 조인트 1의 각도는 5도 이상, 조인트 2의 각도는 30도 이상의 범위로 설정하였다.

3.2. 모의 실험 결과

CRL과 CACRL의 비교를 위한 모의 실험을 수행하였으며 그림 3.2의 그래프는 100 episode 동안의 acrobot이 목표범위에 도달하는데 걸리는 step 수를 나타낸다.

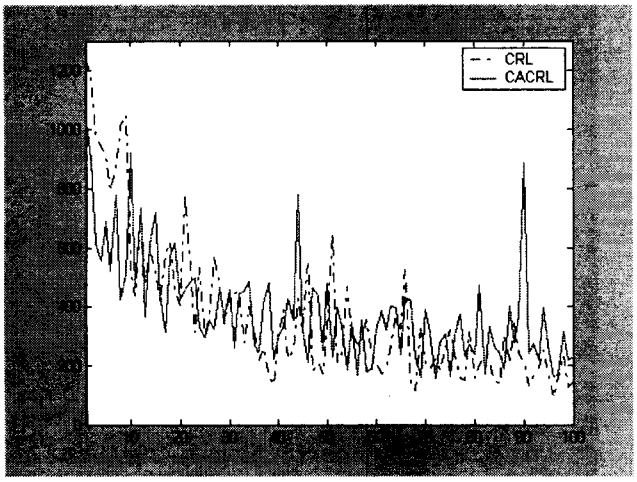

그림 3.2 CRL과 CACRL의 비교
Fig. 3.2 Comparison of CRL and CACRL

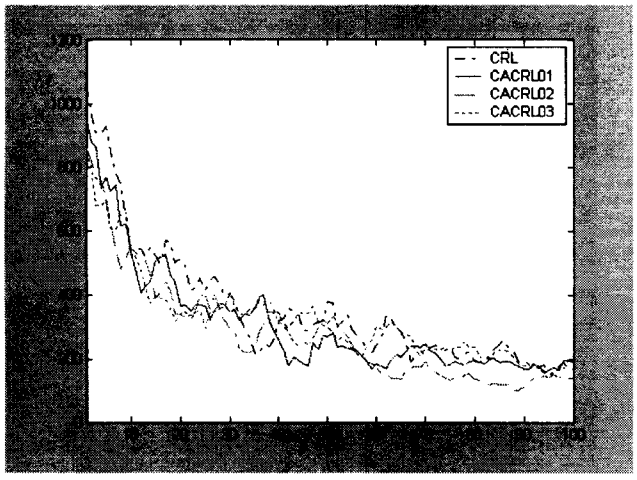

그림 3.3 CRL과 CACRL의 비교 (5-step moving average)
Fig. 3.3 Comparison of CRL and CACRL (5-step moving average)

그림 3.2에서와 같이 CACRL은 일정 episode까지는 CRL에 비하여 빠른 수렴속도를 보이나, 일정 episode 후에는 간헐적으로 수렴 범위를 벗어나거나, CRL과 비슷한 학습 속도를 보였다. 따라서 본 논문에서는 경험적으로 CACRL을 일정 episode까지만 적용하고, 일정 epdisode 후에는 CRL만을 적용하는 방법을 통하여 강화학습의 계산시간과 저장용량을 효율적으로 줄였다. 그림 3.3과 그림 3.4에서 CACRL01은 처음 한 번의 episode, CACRL02는 처음 두 번의 episode, 그리고, CACRL03은 처음 세 번의 episode 동안에 CACRL을 적용하고, 나머지 episode에는 CRL을 적용한 결과를 나타낸다.

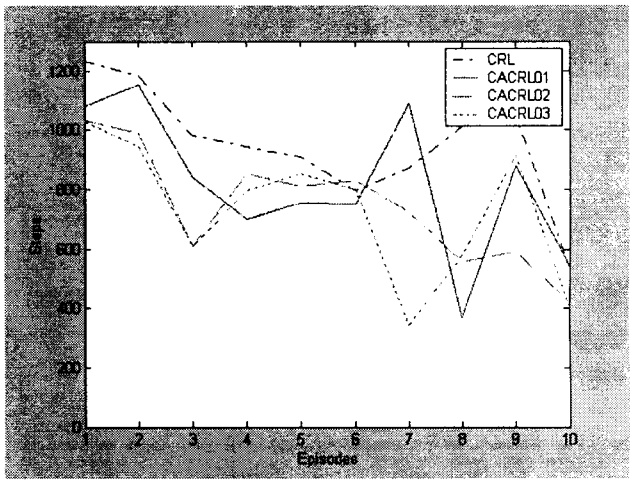

그림 3.4 CRL과 CACRL의 비교 2 (5-step moving average)
Fig. 3.4 2nd. Comparison of CRL and CACRL (5-step moving average)

그림 3.4는 그림 3.3의 그래프 중 10 episode동안의 결과를 나타낸 그래프이다. 비교 결과 CACRL은 CRL 보다 학습 초기에 빠른 수렴속도를 보였다. 모의 실험에서 전체적으로 CACRL와 CRL의 성능이 뚜렷이 구분되지 않을지라도 scratch 후 초기 탐색시간은 수렴 후의 탐색시간과 엄청난 차이가 있기 때문에 강화학습 분야에서 초기 탐색 시간을 줄이는 것은 중요한 과제로 다루어져 왔다. (다음 장의 Acrobot 실험에서 부연 설명함.)

4. Acrobot 실험

4.1 실험 환경

실험 환경과 제작한 Acrobot의 구성도 및 각 부의 설명을 그림 4.1에, 실험 환경을 표 4.1에 각각 나타내었다.

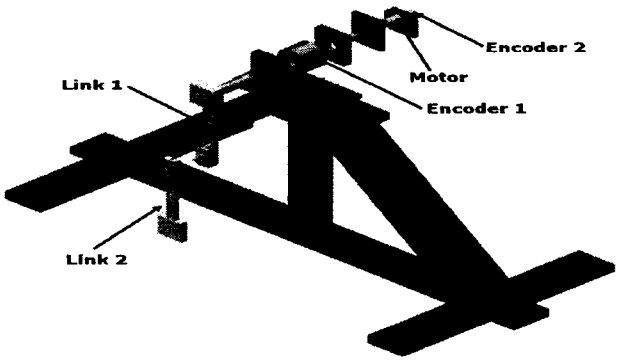

그림 4.1 Acrobot의 구성도
Fig. 4.1 Structural diagram of Acrobot

표 4.1 Acrobot 실험 환경
Table 4.1 Experimental Environment of Acrobot

항목	사양 및 모델명
Computer (CPU)	Intel Pentium III 800Mhz
OS	Windows 2000
Real-Time OS	VenturCom v5.5
Program Language	Visual C++ 6.0
Sampling Time	200ms
Motor Torque	96.3mNm
Motor	Maxon DC Motor 90Watt [RE 35]
Motor Driver	Maxon Motor Control [ADS 50/5]
Encoder 1	Autonics [E100H]
Encoder 2	Maxon Digital Encoder [HEDS5510]
Power Supply	MEAN WELL [S-350-27]

Acrobot 기구의 설계 사양은 표 4.2와 같고, 그림 4.2.는 실제 제작한 Acrobot의 사진이다. 제작한 Acrobot의 기본 구조는 Spong의 논문을 참조하였으며[3], 링크 2와 링크 1 무게의 균형을 맞추기 위하여 조인트 2에 부착 되어야 할 구동 모터를 밸트를 통하여 동력을 전달하도록 하고, 기저에 옮겨 부착하였다.

모터는 기어가 없는 DC 모터를 사용하여 조인트에 기어에의한 역부하가 걸리지 않으면서 부드럽게 swing up 될 수 있도록 하였다.

표 4.2 Acrobot 기구설계 사양
Table 4.2 Design specification of Acrobot

항목	사양
link 1의 길이	330mm
link 1의 mass	542.54g
link 1의 inertia	723724.48 gcm²
link 1의 동작범위	360°
link 2의 길이	330mm
link 2의 mass	493.46g
link 2의 inertia	566887.92 gcm²
link 2의 동작범위	360°

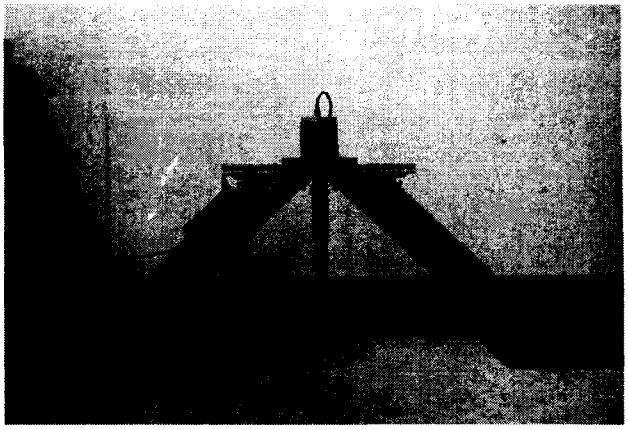

그림 4.2 Acrobot의 사진
Fig. 4.2 Picture of Acrobot

그림 4.3은 Acrobot의 swing up 동작사진이며, 그림 4.4은 목표범위에 도달한 Acrobot의 사진이다.

그림 4.3 Acrobot의 Swing up 동작 사진들
Fig. 4.3 Snapshots of Acrobot Swing up motion

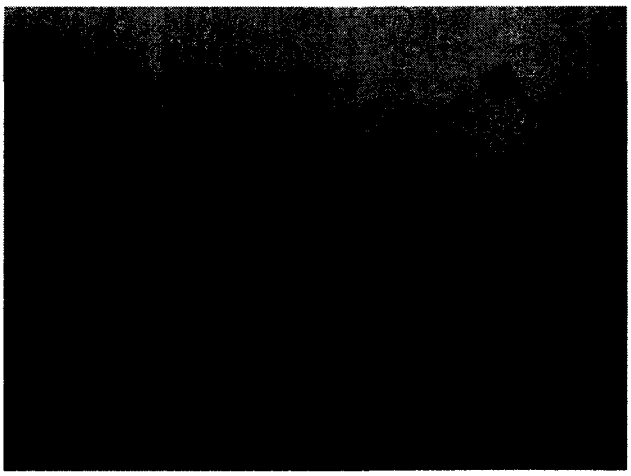

그림 4.4 목표범위에 도달한 Acrobot의 사진
Fig. 4.4 Snapshot of Acrobot reached target boundary

4.2 실험 결과

그림 4.5에 10번의 episode동안의 비교 실험 결과를 나타내었다. 그림에서와 같이 학습 초기에 CACRL을 적용하였을 경우가 CRL 만을 적용하였을 경우보다 적은 step으로 수렴해 가는 것을 볼 수 있다. 그림에서 CACRL10의 결과가 가장 좋지만 그림 4.6과 같이 100 episode를 관찰한 결과 CACRL03이 가장 효과적임을 경험적으로 알 수 있었다.

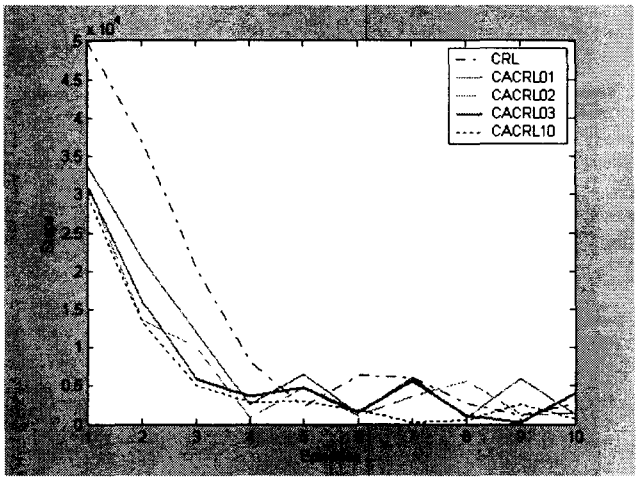

그림 4.5 10 Episode 동안의 결과
Fig. 4.5 Result during 10 Episodes

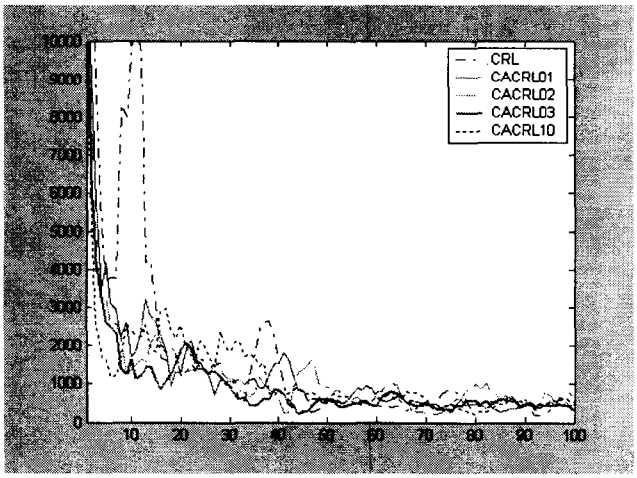

그림 4.6 100 Episode 동안의 결과(5-step moving average)
Fig. 4.6 Result during 100 Episodes(5-step moving average)

- **CACRL의 실험적 기여**: CACRL이 CRL에 비하여 우수한 점은 학습 초기 credit assign을 적용하여 빠른 수렴속도를 보인다는데 있다. 실제로 그림 4.5에서와 같이 초기 scratch상태의 타일들로부터 acrobot이 목표범위 도달하는데 걸리는 시간은 CRL의 경우 약 3시간, CACRL의 경우 약 1.7시간이 걸렸으며, 10번째 episode동안에 걸리는 시간의 합은 CRL의 경우 7.6시간, CACRL03의 경우 4.1시간으로 3.5시간 정도의 차이를 보였다.

5. 결 론

강화학습이 실제 환경에 적용되기 위해서 해결해야 할 중요한 문제 중의 하나는 연속 상태공간을 불연속 상태공간으로 표현함으로써 생기는 많은 용량의 메모리 문제(curse of dimensionality)이다. 이를 해결하기 위한 방법으로 강화학습 연구 분야에서는 함수 근사화 방법이 연구되어왔다. 본 논문에서는 함수 근사화를 적용한 Sutton의 CRL(CMAC-based 강화학습)방법의 수렴속도를 개선하기 위한 방법으로 S. Su 등이 제안한 CA-CMAC 기법을 적용한 CACRL (Credit-Assigned-CMAC-based 강화학습)방법을 제안하였고, CRL과 제안한 CACRL 방법을 Acrobot의 swing-up 제어에 적용한 실험을 통하여 비교하여 수렴속도의 우수성을 보였다

추후 연구로는 CACRL과 CRL의 전환시기의 자동화 방법과 Acrobot의 balancing제어를 위한 불연속 상태공간 및 불연속 행위공간에서 연속 행위의 생성 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] R. M. Murray and J. Hauser, "A Case Study in Approximate Linerization: The Acrobot Example," Electronics Research Lab. College of Engineering, University of California, Berkely, Apirl 1991.

[2] R. S. Sutton and A. G. Barto, "Reinforcement Learning, An Introduction," Cambridge, MA: MIT Press, 1998.

[3] M. W. Spong, "The swing up control problem for the acrobot," IEEE Control Systems Magazine, vol. 15, pp. 49-55, feb. 1995.

[4] J. A. Boyan and A. W. Moore, "Generalization in Reinforcement Learning: Safely Approximation the Value Function," NIPS-7. San Mateo, CA: Morgan Kaufmann, 1995.

[5] G. Boone, "Minimum-time control of the acrobot," International Conference on Robotics and Automation, pp. 3281-3287, 1997.

[6] R. S. Sutton, "Generalization in reinforcement learning: successful examples using sparse coarse coding," in Neural Information Proceeding Systems 8, pp. 1038-1044, MIT Press, 1996.

[7] S. Su, T. Tao, and T. Hung, "Credit Assigned CMAC and Its Application to Online Learning Robust Controllers," IEEE Transaction on Systems, Man, and Cybernetics-Part B:Cybernetics, vol. 33, no. 2, April 2003.

[8] A. D. Luca and G. Oriolo, "Stabilization of the Acrobot via Iterative State Steering," Proceeding of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 1998.

[9] S. C. Brown and K. M. Passino, "Intelligent Control of the Acrobot," Journal of Intelligent and Robotics

Systems 18: 209-248, 1997.
[10] I.H.Suh, J.H.Kim, J.S.Ryoo, Y.J.Cho, and Y.K.Chung, "Region-based Q-Learning using Convex Clustering Approach," in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp601-607, 1997.
[11] D. Ormoneit and P. Glynn, "Kernel-Based Reinforcement Learning in Average-Cost Problems," IEEE Transaction on Automatic Control, Vol. 47. No. 10, October, 2002.

저 자 소 개

장 시 영 (張 時 榮)

2000년 한양대학교 공학대학 전자컴퓨터전기제어공학부 졸업. 2002년 한양대학교 공과대학 전자통신전파공학과(공학석사).
2002년~현재 한양대학교 전자전기제어계측공학과 박사과정 재학중.
관심분야 : 지능제어, 로봇제어, 인공지능, 강화학습

신 연 용 (申 娟 溶)

2003년 한양대학교 공학대학 전자컴퓨터공학부 졸업.
2003년~현재 한양대학교 전자전기제어계측공학과 석사과정 재학중.
관심분야 : 로봇설계, 로봇제어, 기계학습

서 승 환 (徐 承 煥)

2002년 서울산업대학교 공학대학 전자공학과 졸업. 2004년 한양대학교 메카트로닉스공학과(공학석사).
2004~현재 (주)시텍 재직.
관심분야 : 강화학습, 기계학습, 마이크로컨트롤러

서 일 홍 (徐 一 弘)

1977년 서울대학교 졸업. 1982년 한국과학기술원 졸업(공학박사).
1982년~1985년 대우 중공업 기술연구소 근무. 1987~1988년 미국 미시간대 객원 연구원. 1985년~현재 한양대학교 교수.