

## 협조행동을 위한 자율이동로봇의 강화학습에서의 먹이와 포식자 문제

김서광\*, 김민수, 윤용석, 공성곤  
숭실대학교 전기공학과

### Prey-predator Problem in the Reinforcement Learning of Autonomous Mobile Robots for Cooperative Behavior

Kim Seo-Kwang\*, Kim Min-Soo, Yoon Yong-Seock, Kong Seong-Gon  
Intelligent Signal Processing Lab, Electrical Eng. Soongsil Univ.

**Abstract** - 협조행동이 요구되는 다수의 자율이동로봇 시스템에서 각 개체는 주변환경의 인식뿐만 아니라 지속적인 환경변화에 적응할 수 있는 고도의 추론능력을 요구하고 있다. 이에 본 논문에서는 강화학습을 이용하여 동적으로 변화하는 환경에서 스스로 학습하여 대처할 수 있는 협조행동 방법을 제시하였다. 강화학습은 동물의 학습방법 연구에서 비롯되었으며, 주어진 목표를 수행하는 과정에서 개체의 행동이 목표를 성취하도록 하였을 때는 그 행동에 보상을 주어 환경의 상태에 따른 최적의 행동방법을 찾아내도록 학습하는 방법이다. 따라서 본 논문에서는 포식자들이 협조행동을 통하여 능동적으로 움직이는 먹이를 잡는 까다로운 문제에 제안한 방법을 적용하여 그 성능을 검증하였다.

## 1. 서 론

자연계에 존재하는 사회적 동물의 대표적인 특징은 협조행동(Cooperative Behavior) 또는 군행동(Group Behavior)을 들 수 있다. 협조행동은 절대적인 능력을 가진 절대자나 중앙관리자에 의해 조작된 것이 아니라 각 개체간의 의사소통, 상황인지, 그리고 학습 등의 고등양식으로부터 나타나는 자연스러운 행동이다. 따라서 최근에는 이러한 협조행동이나 군행동을 인공적으로 구현하기 위해 다수의 자율이동로봇으로 구성된 다개체 시스템에 대한 많은 연구가 이루어지고 있다.[4][5]

본 논문에서는 두 마리의 포식자가 협조하여 한 마리의 먹이를 추적하는 먹이-포식자 문제에 강화학습을 이용하여 협조행동을 구현하였다. 각 개체들은 변하는 상황에 따라 협조를 통해 가장 효과적으로 먹이를 추적하도록 학습하였음을 시뮬레이션을 통해 검증하였다.

## 2. 강화학습

### 2.1 강화학습이란?

강화학습은 실험 심리학에서 동물의 학습방법 연구에서 비롯되었으나, 최근에는 공학에서 여러 다른 인공지능 분야들과 결합하여 학습 알고리즘을 향상시키는데 많이 이용되고 있다. 가장 간단한 강화학습 방법은 개체가 어떤 행동을 했을 때 수행하고 있는 작업의 방향으로 향상된 결과를 가져오면 그 행동을 유발하기 위하여 그 행동에 강화를 준다는 일반적 상식에 기초한다. 강화학습의 목적은 동적으로 변화하는 환경 하에서 재어기 또는 개체의 행동에 대한 보상을 최대화하는 상태-행동(state-action) 규칙이나 행동발생 전략을 찾는 것이다. 지금까지의 많은 학습론에서는 환경에 대한 정확한 모델링을 통해 교사학습(Supervised Learning)이 많았으나, 현실 적용에서는 모델에 대한 정확한 정보를 얻기가 어렵고 비용이 많이 들기 때문에 최근에는 비교교사학습(Unsupervised Learning)에 대한 연구가 활발히 이루어지고 있다. 그 대표적인 예가 바로 강화학습으로 개체는 “어떠한 행동을 취하라”라는 명령을 직접 받지 않고 행동결과에 대한 결과로서 보상/처벌을 받기 때문에 어떠한 행동을 표출했을 때 가장 높은 보상을 받는지

학습을 통해 발견해야만 한다.[1][2]

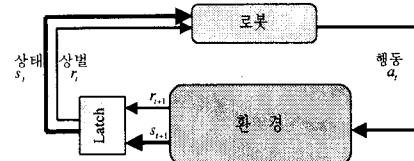


그림1: 강화학습의 기본구조

그림 1은 강화학습의 기본구조를 나타낸다. 학습시스템에서 행동의 결정은 전체 처리과정에 영향을 주게 되며 행동결과로서 상벌이 주어지는데, 상벌은 각 시간 단계마다 행동이 현재의 처리과정에서 어느 정도 공정적으로 작용했는지를 평가하는 강화신호로서 학습시스템에 보내지게 된다. 이러한 강화신호를 이용하여 학습시스템은 행동 발생 규칙을 개선시키고 개선된 규칙에 따라 다른 행동을 발생시키는 과정을 반복하게 된다.

### 2.2 학습 알고리즘

적용된 알고리즘은 Temporal Difference(TD) 방법으로 상태-행동으로 주어지는 함수를 학습시킴으로써 전 단계의 평가에 기초하여 새로운 평가를 내리는 방법이다. 개체의 행동은 상태-행동 쌍으로 이루어진 Q함수를 통해 행동결정 및 함수의 개선이 이루어진다. 즉, 모든 상태에서 가능한 모든 행동에 Q값을 부여하고 학습알고리즘(SARSA 알고리즘)을 이용하여 Q값을 개선시켜나간다. 그리고 Q값의 함수로부터 행동을 선택하게 되는데, 행동선택은 확률적인 방법을 적용한  $\epsilon$ -greedy 방법을 이용한다.  $\epsilon$ -greedy 방법은 다음과 같다.

“ $1 - \epsilon$ 의 확률로  $\max Q(s, a)$ 인 행동을 선택하고, 그 외의 경우 무작위로 행동한다.”

SARSA 알고리즘의 전체적인 순서는 아래와 같다.

- 단계1: 임의의 값으로  $Q(s, a)$  함수값을 초기화.
- 단계2: 새로운 상태  $s$ 의 선택
- 단계3:  $\epsilon$ -greedy 정책에 따라 행동  $a$  선택.
- 단계4: 행동  $a$ 의 실행, 보상  $r$ 의 관측.  
다음상태  $s'$  결정, 다음 행동  $a'$  선택.
- 단계5:  $Q(s, a)$  함수의 개선.
- 단계6: 다음상태를 현재 상태로 변경( $s = s'$ ).
- 단계7: 상태  $s$ 가 목표치에 도달하거나 일정 조건에 도달할 때까지 단계3부터 반복
- 단계8: 정해진 episode만큼 2-7사이를 반복

단계5에서의  $Q(s, a)$ 의 개선은 다음과 같이 주어진다.

$$Q(s, a) = Q(s, a) + \alpha TD_{err} \quad (1)$$

여기에서  $\alpha$ 는 학습상수이며,  $TDerr$ 는 temporal difference error로서 식(2)와 같다.

$$TDerr = r + \gamma Q(s', a') - Q(s, a) \quad (2)$$

여기에서  $\gamma$ 는 보상에 대한 감쇄인자이다.

모든 시간 간격마다 SARSA 알고리즘은  $(s, a, r, s', a')$  인자들을 이용하여 상태-행동 함수  $Q(s, a)$ 를 개선하는데, 이는 이미 잘 알려진 Q-learning 알고리즘에 정책을 추가한 것으로써 학습된 상태-행동 함수  $Q$ 를 이용하여 직접 최적의  $Q^*(s, a)$ 를 찾도록 만든다.

### 3. 먹이와 포식자 문제

먹이와 포식자 문제(Prey-predator Problem)는 한 마리의 먹이를 두 마리의 포식자가 사냥하는 문제를 모델링한 것으로서 포식자로 주어지는 두 개체가 서로 협조행동을 통해 능동적으로 움직이는 먹이 개체를 추적하는 문제이다. [1][2]

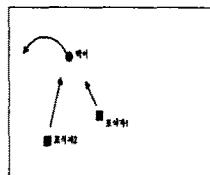


그림 2: 먹이와 포식자

#### 3.1 먹이의 행동

먹이는 포식자를 피하는 행동을 기본으로 하며 포식자 이외의 장애물 발견 시에는 자극-반응(Stimulus-Reaction) 개념을 통해 센서마다 서로 다른 가중치를 두고 센서의 값에 따라 모터의 회전방향 및 속도가 식(3) 및 식(4)와 같이 결정된다.

$$v_L = \mathbf{u}^T \mathbf{s} + v_F \quad (3)$$

$$v_R = \mathbf{w}^T \mathbf{s} + v_F \quad (4)$$

여기서  $\mathbf{u}$ 와  $\mathbf{w}$ 는 각 센서에 곱해지는 가중치 벡터이며  $\mathbf{s}$ 는 센서의 입력 벡터이다. 그리고  $v_F$ 는 정회전 바이어스로 센서에 장애물이 발견되지 않았을 때 로봇을 전진하게 하는 역할을 하는 전진속도이다.

$$\mathbf{s} = [s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7] \quad (5)$$

$$\mathbf{u} = [4, 4, 6, -18, -15, -5, 5, 3] \quad (6)$$

$$\mathbf{w} = [-5, -15, -18, 6, 4, 4, 3, 5] \quad (7)$$

$\mathbf{s}$ 는 8개로 주어지는 센서의 입력을 나타내며,  $\mathbf{u}$ 는  $reverse(\mathbf{w})$ 와 같은 값으로써 장애물이 로봇의 정면에 있을 때는 장애물을 피하지 못하는 경우가 발생하므로 몇 가지 값을 조절하여 사용한다.

#### 3.2 포식자의 학습을 위한 상태와 행동의 정의

포식자들이 먹이를 추적하는 모습을 그림3에 나타내었다. 먹이의 진행방향  $d_t$ (Direction of Target)이며 포식자1의 진행방향은  $d_{p1}$ (Direction of Predator1)이고 포식자2의 진행방향은  $d_{p2}$ 이다. 그리고  $d_{p1}$ (Direc-

tion of Predator to Target)은 포식자가 본 먹이의 방향이고  $d_{p1}$ 은 먹이가 본 포식자의 방향을 나타낸다.

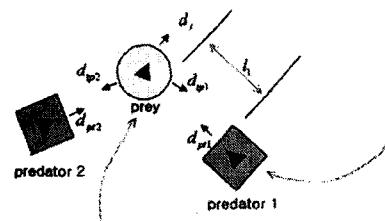


그림3: 포식자들이 먹이를 추적하는 모습

강화학습은 모델링하기 어려운 문제를 개념적인 용어들을 이용하여 쉽게 설계 할 수 있다는 장점이 있지만 상태와 행동을 정의하고 정책을 설정하기가 매우 힘들다. 학습 시스템의 구조는 상태를 어떻게 설정하는지에 따라 크게 달라지며 정책을 설정하는 방법에 따라 임무의 완수 여부와 학습 속도가 크게 달라진다.

그림3은  $d_{p1}=3$ 이며  $d_{p2}=0$ 인 경우를 보여주고 있다. 따라서 포식자1은  $l_1$ 만큼의 거리에서 먹이의 오른쪽 방향으로  $d_{p2}$ 로 먹이를 추적 중인 것을 알 수 있다. 마찬가지로는 먹이에서 바라본 포식자2의 방향이며  $d_{p2}$ 는 포식자2가 바라본 먹이의 방향이다. 협조행동을 할 때, 포식자1은 포식자2가 먹이의 어느 쪽에서 어떤 행동을 하고 있는지를 파악하고 상황에 맞는 행동을 취하게 된다. 먹이가 본 포식자의 방향  $d_{p1}$ 과 포식자가 본 먹이의 방향  $d_{p2}$ 에 대한 상태는 표1과 같이 정의한다.

표1: 방향에 대한 상태 값의 정의

direction	degree	value
↑ (앞)	$-\pi/4 \sim \pi/4$	0
← (왼쪽)	$\pi/4 \sim \pi 3/4$	1
↓ (뒤)	$-\pi 3/4 \sim \pi 3/4$	2
→ (오른쪽)	$-\pi 3/4 \sim -\pi/4$	3

먹이와 포식자간의 거리는 표2와 같이 4단계로 나눈다. 단계를 높일수록 보다 정교한 행동들이 만들어지지만 기하급수적으로 늘어나는 상황과 학습속도를 고려할 때 상태 값을 최소 필요량으로 줄일 필요가 있다.

표2: 거리에 대한 상태값의 정의

distance	value
0 ~ 20cm	0
20cm ~ 50cm	1
50cm ~ 1m	2
1m 이상	3

행동(action)은 표3과 같이 4가지 방법으로 각각 포식자가 먹이에게 접근하는 방향을 정의한다. 정의된 행동은 단위 벡터장을 이용하여 각각 모터의 회전속도로 바꿔게 된다.

표3: 행동에 대한 정의

action	value
앞으로 접근	0
왼쪽으로 접근	1
오른쪽으로 접근	2
뒤로 접근	3

각각의 상태를 표4와 같이 binary coding하면 가능한 상태의 개수는 255가지가 된다.

표4: 포식자의 상태 코딩

거리 ( $l_1$ )	다른포식자의 상태 ( $d_{h2}$ )	먹이가 보는 방향 ( $d_{m1}$ )	먹이를 보는 방향 ( $d_{m1}$ )				
$d_7$	$d_6$	$d_5$	$d_4$	$d_3$	$d_2$	$d_1$	$d_0$

### 3.3 단위벡터장을 이용한 이동로봇의 경로

단위벡터장  $N$ 은 식(8)과 같으며 식(8)에서 F는 로봇의 작업공간, I는 일의방향으로의 단위벡터들의 집합이다. 로봇을 제어하는데 있어 이 단위벡터는 로봇의 제어해야 할 방향을 나타낸다. 정규화된 벡터를 사용함으로 단위벡터장  $N$ 은 방향으로 표시된다. [3]

$$N: F \rightarrow I \quad (8)$$

$$\theta_N: F \rightarrow [-\pi, \pi] \quad (9)$$

그림 4는  $g$ 점에서의 동작을 위한 벡터장이다. 그림4에서 각 점에 연결된 직선은 그 점에서의 단위 벡터장을 나타낸 것이며, 이와같은 단위 벡터장의 각도는 다음과 같은 식으로 나타내어진다.

$$Q_N(p) = \angle pg - n\phi \quad (10)$$

$$\phi = \angle pr - \angle pg \quad (11)$$

$n$ 은 조정가능한 양의 상수이며 단위 벡터장의 모양이나 그에 따른 로봇의 움직임은 상수  $n$ 값이나  $g$ 점과  $r$  점 사이의 거리에 따라 변화한다. 위 식에서 점  $g$ 는 먹이로봇의 위치이며 점  $r$ 은 행동방향을 결정하는 점이된다. 위와 같은 단위 벡터장은 경험적인 방법에 의해 구해진 것이며, 실제 로봇축구에서 사용되고 있다

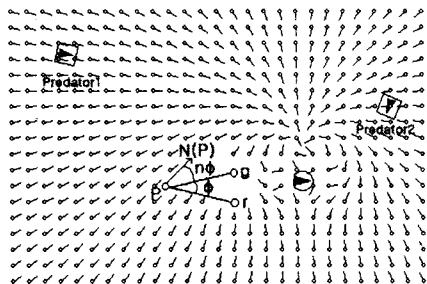


그림4: 포식자1의 행동1에 대해 형성된 단위 벡터장

### 4. 시뮬레이션

먹이-포식자 문제에서 각 개체들의 속도는 동일하게 설정하여 협조행동을 통해 포식자가 먹이를 추적할 수 있도록 하였으며, 강화학습에서 상수  $\alpha=0.1$ ,  $\gamma=0.9$ , 그리고  $\epsilon=0.01$ 로 설정하였다. 또한, 학습과정에서 포식자가 먹이추적에 성공한 경우에는 보상이 0이고, 실패한 경우 -1을 주었다. 그림5는 강화학습 과정에서 포식자가 각 상태에 따른 행동을 통해 얻은 보상값들의 합이며, 초기에 -200에서 1000회의 학습 후에는 -10 까지 증가되었음을 알 수 있다. 즉, 포식자는 협조행동을 통해 10회의 상태변화만을 통해 포획에 성공할 수 있음을 알 수 있다.

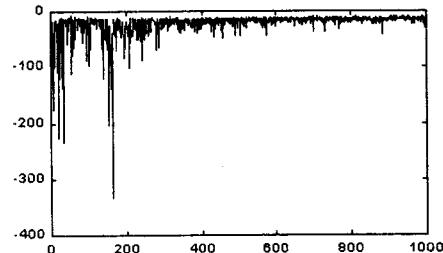
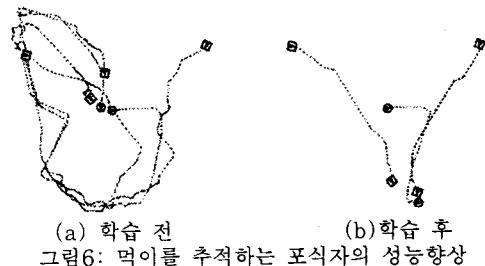


그림5: 강화학습에서 보상값의 변화

그림 6은 학습 전의 Q값과 학습 후의 Q값을 이용하여 실제 먹이추적을 시뮬레이션 해 본 결과이다.



(a) 학습 전 (b) 학습 후  
그림6: 먹이를 추적하는 포식자의 성능향상

학습 전의 포식자는 먹이를 추적하기 위해 오직 한 쪽 방향으로만 접근함으로써 먹이를 포획하지 못하거나 어느정도 시간이 경과한 후에는 운에 의해 포획하지만, 학습 후에는 두 마리의 포식자가 서로 협조하여 먹이를 포위함으로써 능동적으로 이동하는 먹이를 포획하게 됨을 알 수 있다.

### 5. 결 론

협조행동을 하는 다수의 자율이동로봇 시스템에서 각각의 개체는 주변환경의 인식뿐만 아니라 환경변화에 적응할 수 있는 고도의 추론능력을 요구하고 있다. 따라서 본 논문에서는 강화학습을 이용하여 동적으로 변화하는 환경에서 스스로 학습하고 대처할 수 있는 협조행동 문제를 해결하기 위한 협조행동 방법을 제안하였으며, 제안한 방법을 먹이-포식자 문제에 적용하여, 두 개체로 구성된 포식자는 강화학습을 통해 학습된 협조행동에 기초하여 먹이를 추적함을 시뮬레이션을 통해 검증하였다.

### (참 고 문 헌)

- [1] Andrea Bonarini, "Reinforcement Learning of Hierarchical Fuzzy Behaviors for Autonomous Agents", Proceedings of IPMU96, pp. 1223-1228, 1996
- [2] Andrea Bonarini, "Anytime learning and adaptation of structured fuzzy behaviors", publication on the Special Issue of the Adaptive Behavior Journal about "Complete agent learning in complex environments", n.5, 1997.
- [3] K.C. Kim, "Evolutionary Programming and Q-Learning based Controller Design for Soccer Robot", Proceedings of the 1st Workshop on Soccer Robotics, pp. 59-71
- [4] M. Sipper, "An Introduction to Artificial Life Explorations in Artificial Life", pp. 4-8, 1995.
- [5] J.S. Bay, "Design of the 'army-ant' cooperative lifting robot", IEEE Robot. Automat. Mag., vol 2, no.1m Mar.1995