

# 행동기반 시를 이용한 슈팅게임 캐릭터의 적응형 행동생성

## Creating Adaptive Behaviors for Shooting Game Characters

### Behavior-based Artificial Intelligence

구자민, 홍진혁, 조성배  
연세대학교 컴퓨터과학과

Ja-Min Koo, Jin-Hyuk Hong, Sung-Bae Cho  
Dept. of Computer Science, Yonsei University  
E-mail : {icicle, hjinh, sbcho}@sclab.yonsei.ac.kr

#### 요 약

로보코드는 사용자가 직접 제작할 수 있는 슈팅게임 환경으로서, 이를 이용한 경진대회가 개최되고 있다. 매우 다양한 작전을 구사하는 로봇들이 인터넷을 통해 공개되지만, 대부분의 전략은 사람이 직접 설계하여 행동이 단순하고, 변화하는 환경에 따라 행동을 구사하는데에 어려움을 가지고 있다. 이로 인해 아무리 훌륭한 전략을 가지고 있더라도 환경적 요소에 따라 예상치 못한 이벤트가 발생했을 경우 적절한 행동을 선택하여 행하기가 어렵다. 본 논문에서는 동적인 환경에서 적절한 행동을 선택하는 행동선택 네트워크를 이용하여 상대 전략에 따라 적절한 행동을 선택하는 방법을 제안하고 로보코드에 적용하여 실험하였다. 실험결과, 상대 탱크의 전략에 따라 다양한 행동들을 자동으로 선택하였으며, 경기 결과로 그 전략의 우수성이 입증되었다.

#### 1. 서론

오래부터 다양한 목적에 따라 전쟁이 시작되었으며 게임의 기원은 이런 전쟁에서부터 비롯되었다 [1]. 또한 전쟁에서 승리하기 위해서는 전쟁의 목적에 따라 다양한 전략이 필요함과 마찬가지로 게임에서 역시 승리하기 위해서는 다양한 전략이 필요하며 이에 다양한 전략 게임들이 소개되었다[2,3]. 특히, IBM 알파워크에서 개발한 로보코드가 그 대표적인 예이며 이를 이용한 각종 대회들이 개최되고 있으며 이를 위해 다양한 전략을 가진 탱크들이 사용자들에 의해 개발 및 공개되고 있다. 하지만, 불확실한 상황에도 효과적으로 대응할 수 없으며, 구현된 행동들은 변화하는 환경에 적응력있게 잘 대처하지 못한다.

본 논문에서는 불확실한 환경에서도 적응력있게

유연한 행동을 만들 수 있는 행동선택 네트워크 기법을 로보코드에 적용하여 다양한 전략을 수립하였다. 그리고 로보코드에서 제공하는 우수한 샘플 로봇과의 대전을 통해 행동의 유연성을 보인다.

#### 2. 관련연구

##### 2.1 행동선택 네트워크

복잡한 문제를 해결하기 위해 큰 문제를 여러 개의 쉬운 부분문제로 분할하여 제어기를 개발한 후에 결합하는 방식이 사용된다[4,5].

행동 네트워크는 행동, 외부목표, 내부목표가 링크를 통해 연결되어있다. 행동은 선행조건, 추가조건, 삭제조건, 활성화도 및 실행코드로 이루어져 있다. 선행조건은 행동이 실행되기 위해서 참여해야 하는 조건들의 집합이다. 추가조건은 행동이 실행되었을 때 참여 되기 쉬운 조건들의 집

합이다. 삭제조건은 행동이 실행되었을 때 거적이 되기 쉬운 조건들의 집합이다. 활성화도는 행동이 활성화 된 정도를 나타낸다. 실행코드는 행동이 선택되었을 때 수행되는 내용을 담는다. 그림 1은 행동의 구성요소를 보여준다.

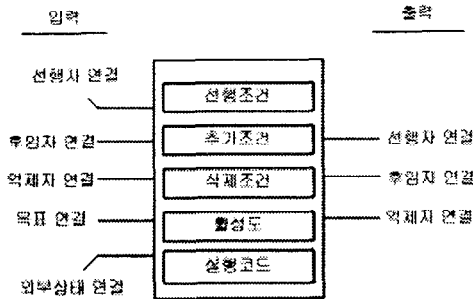


그림 1. 행동의 구성요소와 연결 관계

외부 상태가 행동의 선행조건일 때 외부상태연결이 설정된다. 목표에 도움이 되는 행동일 경우 목표와 행동사이에는 목표연결이 설정된다. 행동들 사이의 연결은 선행자연결, 후임자연결, 억제자연결로 이루어진다. 행동 B의 실행이 행동 A의 실행에 도움이 되는 경우 A에서 B로 선행자연결이 이루어진다. 행동 A의 실행이 행동 B의 실행에 도움이 되는 경우 A에서 B로 후임자연결이 이루어진다. 행동 B의 실행이 A의 실행에 방해가 될 경우에는 A에서 B로 억제자연결이 이루어진다.

여러 행동 중에서 하나의 적절한 행동을 선택하기 위한 과정은 다음과 같이 진행된다.

- ① 외부상태와 목표로부터 들어온 신호를 행동에 입력한다.
- ② 행동들 사이의 내부연결을 통해 활성화도를 교환한다. 이때 행동들은 상호 협조하기도 하고 억제하기도 한다.
- ③ 행동들의 활성화도가 무한히 커지는 것을 막기 위해 정규화를 수행하며, 행동들의 활성화도의 합이  $\pi$ 가 되도록 한다.
- ④ 행동들 중에서 선행조건이 모두 참이고, 활성화도가 임계치  $\theta$ 보다 클 경우 선택한다. 이때 선택된 행동이 한 개 이상일 경우 활성화도가 가장 높은 행동을 선택하고, 선택된 행동이 없을 경우 임계치  $\theta$ 를 10% 감소시킨 후 과정 ①로 돌아간다. 행동이 선택될 때까지 ①~④의 과정을 반복한다.

## 2.2 로보코드[6]

본 논문에서 제안하는 방법을 인공지능 시뮬레이터 중 하나인 로보코드에 적용하였다. 로보코드란 IBM 알파워크에서 개발한 자바기반의 시뮬레이터로 사용자에게 의해 개발된 탱크들이 규정된 경기장에서 상대 탱크들을 향해 총을 쏘면서 경

기를 한다. 로보코드는 실행환경을 비주얼하게 할 수 있으므로, 각 로봇들의 행동들을 직접 눈으로 관찰하기에 아주 용이하다.

로봇들은 한 로봇이 남을 때까지 경기가 진행되고, 최종 생존로봇이 그 경기에서의 승자가 된다. 로봇들은 자신과 적들의 다양한 정보를 수집하여 어떠한 행동을 행할지를 결정한다. 경기에서 이기기 위해서는 다양한 환경에서 적절한 행동을 하게끔 하는 전략이 필요하다. 하지만, 로봇들은 사람에 의해 프로그래밍 되므로 단순한 행동을 하는 경향이 있으며, 미처 고려하지 않은 일이 발생할 경우 적절한 행동을 잘 하지 못한다.

경기가 시작되면 로봇은 일정한 에너지를 가지고 움직이며, 포탄을 발사하거나 벽 혹은 상대 로봇과 부딪히게 되면 일정한 양의 에너지가 감소한다. 에너지가 0이 되면 로봇은 폭발하며 경기에서 패배한다. 반면, 자신이 쏜 포탄을 상대가 맞게 되면 상대가 잃은 만큼의 에너지를 자신이 획득한다.

로봇은 포, 레이더 및 본체로 이루어져있고 그림 2는 로봇의 구조를 보여준다. 포는 포탄의 발사, 레이더는 상대의 위치를 탐지 및 상대 정보

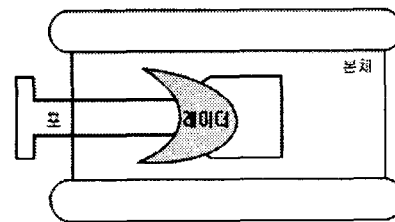


그림 2. 로봇의 구조도

의 수집, 본체는 이동을 담당하며 각 부분은 독립적으로 동작한다.

## 3. 제안하는 방법

### 3.1 기본행동

직진하기는 장애물이 없을 경우 빠르게 다른 장소로 이동하게 한다. 우측으로 회전은 상대 로봇이 근접했을 경우 살짝 비켜서게 하며, 좌측으로 회전은 벽에 근접하거나 상대 포탄이 근접했을 때 이들과의 충돌을 방지하는 역할을 한다. 포탄발사는 상대 로봇이 자신의 레이더에 포착되었을 경우 상대의 이동정보를 파악하여 포탄을 발사하는 역할을 수행한다.

### 3.2 행동 네트워크

행동 네트워크는 그림 3과 같이 정의되고, 중앙의 원은 로봇이 선택할 수 있는 기본행동을 나타낸다. 각각의 상태는 표 1과 같이 정의된다.

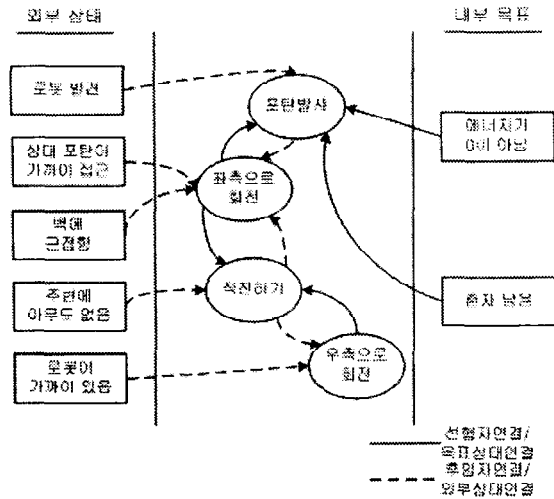


그림 3. 행동 네트워크

표 1. 행동 네트워크의 외부상태와 내부목표

종류	이름	참일 조건	값
외부상태	로봇 발견	상대로봇 레이더에 발견	이진
	상대포탄접근	레이더에 발견된 포탄과의 거리가 2 미만	이진
	벽에 근접함	벽과의 거리가 2 미만	이진
	주변에 아무도 없음	레이더에 발견된 것이 아무것도 없음	이진
	로봇이 가까이 있음	레이더에 발견된 로봇과의 거리가 2 미만	이진
내부목표	에너지가 0이 아님	$\frac{m-n}{m}$	실수
	경기장에 혼자남음	num_robot = 1	이진

상대 로봇이 레이더에 탐지되었을 때 상대 로봇의 정보를 분석하여 로봇은 상대의 이동 경로를 예측하여 포탄을 발사한다. 상대 포탄과 벽 및 로봇과의 거리는 유클리디안 거리를 의미한다. 에너지에서  $m$ 은 로봇의 초기 에너지 수준을 의미하고,  $n$ 은 로봇의 현재 에너지 수준을 의미한다. 로보코드에서  $m$ 은 100으로 항상 일정한 초기값을 부여한다. num\_robot은 처음 경기에 참여했던 로봇의 수를 가지고 있다가 로봇이 파괴될 때마다 감소되어 1이 되었을 때를 의미하며 그 때 자신이 살아있음을 의미한다. 하지만 로보코드에서는 경기 시작 전에 로봇이 1대일 경우의 예외 상황을 검사하지 않으므로, 설계 할 때 이런 예외 경우를 감안해야 한다.

4. 실험 환경 및 결과

4.1 실험환경

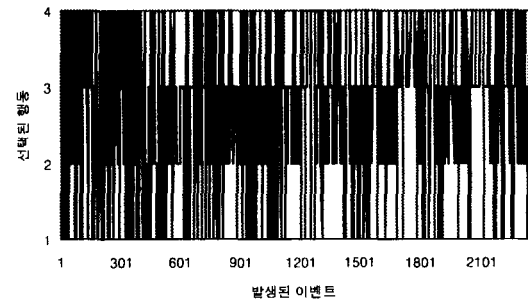
본 논문에서 설계한 행동선택 네트워크를 이용한 로봇의 우수성을 입증하기 위해 로보코드에서 제공하는 wallRobot, cornerRobot, 및 ramfire Robot과 실험하였다.

wall-Robot은 벽을 따라다니면서 포를 쏘는

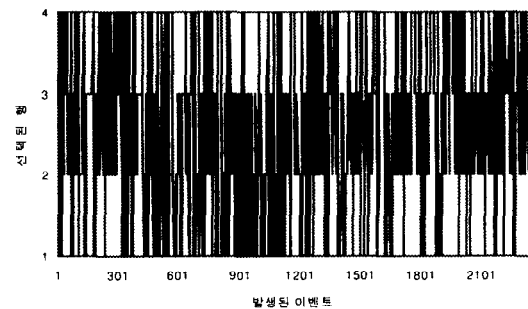
단순하면서도 안정된 형태로 경기를 진행하며 로보코드에서 샘플로 제공하는 가장 우수한 로봇이며, corner-Robot은 경기 시작과 동시에 경기장 가장 구석부분에 자리를 두고 상대가 감지되면 포탄을 발사한다. 끝으로 ramfire-Robot은 상대를 들이받을 경우 2점을 득점하고 들이받아 죽었을 경우 자신이 상대에게 가한 모든 점수의 30%를 추가적으로 획득하는 전략을 가지고 있다.

4.2 실험결과

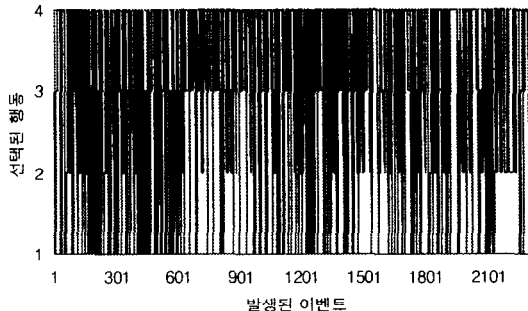
그림 4는 로보코드에서 제공하는 샘플 로봇과의 경기에서 행동 네트워크를 이용하여 동작하는 로봇의 선택되는 행동들의 변화를 보여주고 있으며, 이는 각 로봇과의 경기 중 처음부터 2200번의 행동선택 과정이다.



(a) wall-Robot과의 경기결과



(b) corner-Robot과의 경기결과



(c) ramfire-Robot과의 경기결과

그림 4. 각 로봇과의 경기 중 선택된 행동순서 (1: 포탄발사, 2: 좌측회전, 3: 직진 4: 우측회전)

그림 4에서 보는 바와 같이 상이한 전략을 가진 로봇들과의 경기 때마다 항상 다른 행동 선택과정을 거치게 된다. wall-Robot과의 경기의 경우, 초반에는 상대의 움직임을 파악하기 위해 회전 및 이동이 많았으나, 그 후 규칙적인 포탄 발사의 행동을 한다. Corner-Robot과의 경기에서는 상대가 경기장 구석에 자리 잡았을 때부터 집중적으로 발포하는 행동을 하는 것을 알 수 있다. Ramfire-Robot과의 경기는 이 로봇의 작전상품으로 부딪혀서 상대를 파괴시키려는 전략의 특성상 상대에게 포를 발사하기 보다는 경기 전체적으로 상대를 피하기 위한 행동이 선택되었음을 알 수 있다. 이러한 행동 선택과정이 적합한지를 확인하기 위해 본 논문에서는 실제 위 실험에서 사용한 상대 로봇과의 1:1 경기를 10차례에 걸쳐 실험하였으며 표 2는 그 경기 결과를 보여준

표 2. 각 로봇과의 경기 종합 결과

	총점	생존	포탄으로 입은 피해
JMRobot	495	500	747
Wall	279	0	275
JMRobot	475	500	979
Corner	156	0	156
JMRobot	173	500	965
Ramfire	281	0	281

다. 표 2에서 보는 바와 같이 행동 네트워크를 이용해 구현된 로봇의 경우 큰 점수차이로 경기에서 승리를 하게 되며 상대포탄에 의한 피해 역시 적게 받음을 알 수 있다. 하지만, 설계된 행동 네트워크의 수가 적으므로 두드러지게 다양한 행동들을 선택하기는 어렵다.

### 5. 결론 및 향후 연구

다양한 전략 게임들이 소개되고 있으며, 특히 로보코드를 이용한 각종 행사들이 개최됨에 따라 다양한 전략을 구사하여 상대를 제압하는 로봇들이 공개되고 있다. 하지만 모든 예외 상황 및 환경적 요인의 변화에 따른 상황에서 적절한 행동을 선택하여 행하기란 어렵다. 본 논문에서는 인공생명 분야에서 많이 이용되는 행동선택 네트워크를 로보코드에 이용하여 상대에 따라 다양한 전략으로 상대를 이기는 로봇을 제안하였다. 설계된 행동선택 네트워크가 환경 변화에 따라 다양하고 적절한 행동을 잘 선택하는지를 알아보기 위해 이를 이용한 로봇과 다른 전략을 가진 3분류의 로봇들과의 실험을 하였다. 실험결과 상대 로봇의 전략에 따라 모두 다른 행동 선택과정을 보임을 알 수 있었다. 또한 선택된 행동이 적합한지를 보이기 위해 각 로봇과의 10번의 경기를 가졌으며, 그 결과 높은 점수차이로 승리함을 보였다.

하지만, 현재 설계된 행동 네트워크의 종류가 단조롭기 때문에 아주 다양한 행동을 선택하여

행하기는 어렵다. 따라서 향후 좀 더 구체적인 행동들을 추가하여 좀 더 다양한 행동이 선택되도록 하는 연구가 필요하다.

### 참고문헌

- [1] A. Hausrath, "Venture simulation in war," *Business and Politics*, McGraw-Hill, 1971.
- [2] N. Baba, *THE COMMONS GAME*, Introduction Booklet, 1980.
- [3] J. H. Ausbel, *The greenhouse effect: An educational board game*, Introduction Booklet, IIASA, 1981.
- [4] R. A. Brooks, "A Robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, March 1986.
- [5] P. Maes and R. Brooks, "Learning to coordinate behaviors," *The American Conference on Artificial Intelligence*, pp. 796-802, August 1990.
- [6] S. Li, Rock'em, sock'em robocode!, 2002.  
<http://www-106.ibm.com/developerworks/java/library/j-robocode>