

# 고차원 행동구현을 위한 행동 네트워크와 규칙기반 결합방식의 비교

김경중, 조성배

연세대학교 컴퓨터과학과

e-mail:uribyul@candy.yonsei.ac.kr, sbcho@csai.yonsei.ac.kr

## A Comparison of Behavior Network and Rule-based Integration Method for Higher-level Behavior Implementation

Kyung-Joong Kim, Sung-Bae Cho

Dept. of Computer Science, Yonsei University

### 요약

분산 인공지능 분야에서는 여러 개의 기본 객체들이 상호작용을 통해 원하는 작업을 수행하는데, 그 한 예로 행동기반 이동로봇 제어 시스템을 들 수 있다. 이것은 여러 개의 기본 행동 모듈을 개발한 후, 적절한 조정방법을 사용하여 어려운 문제를 해결하며, 행동 네트워크는 행동 모듈들간의 협력과 억제를 모델링 하여 주어진 작업을 달성하도록 행동들의 실행순서를 조정하는 방법중의 하나이다. 정적으로 행동을 선택하는 규칙기반 결합 방법과는 달리 목표에 기반 하여 행동 네트워크는 다양한 행동들의 실행순서를 동적으로 선택한다는 장점이 있다. 본 논문에서는 진화 방식으로 설계된 기본 행동 모듈을 행동 네트워크와 규칙기반 방법으로 선택하는 실험을 수행하며, 그 결과 행동 네트워크가 변화하는 환경에서도 좋은 성능을 보인다는 사실을 확인할 수 있었다.

### 1. 서 론

인공지능의 한 분야로 분산인공지능이 있는데, 하나의 작업을 수행하기 위해 여러 개의 기본 객체를 이용하여, 각각의 협력과 억제를 조정하는 방법이 필요하다. 분산인공지능 분야의 하나인 행동기반 이동로봇 제어 시스템은 독립적으로 설계된 기본 행동모듈들을 조정방법으로 통합하여 어려운 작업을 수행하도록 한다.

기본 행동모듈들의 실행순서를 적절히 조정하여 보다 높은 수준의 행동을 수행하도록 하는 것이 행동기반 이동로봇 제어의 중요한 문제중의 하나로 연구되고 있다. 행동 네트워크는 기본 행동모듈들간의 협력과 억제를 바탕으로 목표에 적합한 행동을 선택하는 모델이다. 이의 장점은 높은 수준의 행동을 표현하는데 적합하며, 학습도 가능하다는 점이다[1].

행동 네트워크의 또 다른 장점으로 동적인 행동선택을 들 수 있다. 이는 정해진 규칙에 의존해서 행동의 실행순서를 결정하지 않고, 주어진 목표에 맞게 실행순서를 조정함을 의미한다. 정적으로 행동을 선택한다는 것은 목표와는 관계없이 동일한 환경이 설정되면, 동일한 순서로 행동을 선택한다는 것을 의미한다. 고차원의 행동표현은 저차원의 행동들을 선택하는 순서에 의해 결정되기 때문에 정적으로 행동을 선택하는 것과 동적으로 행동을 선택하는

것은 큰 차이를 야기시킨다.

본 논문에서는 하나의 행동 모듈만으로는 해결할 수 없는 문제에 대하여, 행동 네트워크와 규칙기반 결합방법을 이용하여 기본 행동모듈들의 실행순서를 조정한다. 행동 네트워크 방법으로 표현된 고차원 행동과 정적인 몇 개의 규칙으로 정의된 고차원 행동의 차이를 비교하기 위해 환경을 변화시켜 실험을 수행해 본다. 기본 행동 모듈은 이동로봇을 제어하는 역할을 수행한다.

### 2. 관련 연구

Mataric은 행동 네트워크를 이용하여 로봇의 상호작용과 학습에 관한 연구를 수행하였다[1]. Mataric의 행동 네트워크의 노드는 기본 행동 모듈로서 활성화도와 행동수행을 담당하며, 예지는 영구적 연결, 가능성 연결, 제약조건 연결로 이루어진다.

Maes는 기존 행동 선택 방법들의 장점을 결합한 행동 네트워크 모델을 제안하였다[2]. Maes의 행동 네트워크는 외부환경의 상태뿐만 아니라 로봇의 내부 목표도 반영하는 모델이다. 로봇은 행동 네트워크를 통해 한번에 하나의 행동만을 선택하며 기본 행동들은 상호 경쟁한다.

Dorer는 Maes의 행동 네트워크의 단점을 수정하여 확장된 행동 네트워크를 제안하였다[3]. Maes의 행동 네트워크 모델은 연속적인 영역에서는 잘 작동하지만, 연속적

인 상태에 의해 제공되는 추가적인 정보를 이용하지는 못한다. Dorer는 실수값 명제들을 소개하여 이산적인 영역에서 연속적인 영역에까지 문제를 해결하였다.

### 3. 행동 네트워크

행동 네트워크는 노드, 외부환경, 목표를 통해 구성된다. 노드는 기본 행동 모듈을 나타내고, 외부환경은 주어진 환경을 표현하기 위한 기본환경 변수들을 나타내고, 목표는 개체가 내부적으로 달성하여야 할 기본 상태를 나타낸다. 개체가 선택하는 행동은 외부환경과 목표에 영향을 받으며, 내부적으로는 기본 행동 모듈들간의 활성화도 교환에 영향을 받는다. 노드들간의 연결관계는 선행자연결, 후임자연결, 억제자연결로 나누어지며, 노드가 지니고 있는 조건들간의 관계를 통해 설정된다.

- 선행자 연결: 명제  $p$ 가 행동 A의 선행조건이고 상태가 거짓이며, 행동 B의 추가조건이면, A에서 B로 선행자 연결이 설정된다.
- 후임자 연결: 명제  $p$ 가 행동 B의 선행조건이고 상태가 거짓이며, 행동 A의 추가조건이다. 만약 행동 A가 실행 가능하다면, A에서 B로 후임자 연결이 설정된다.
- 억제자 연결: 명제  $p$ 가 행동 A의 선행조건이고 상태가 참이며, 행동 B의 삭제조건이라면, A에서 B로 억제자 연결이 설정된다.

노드는 기본 행동모듈의 수행이외에 선행조건, 추가조건, 삭제조건, 활성화도를 추가적으로 가지고 있다. 선행조건은 행동 모듈이 실행되기 위해서 반드시 참이어야하는 조건들을 나타낸다. 추가조건은 행동 모듈이 실행될 때 참이 되기 쉬운 조건들을 나타내고, 삭제조건은 행동 모듈이 실행될 때 거짓이 되기 쉬운 조건들을 나타낸다. 활성화도는 행동 모듈을 선택하기 위한 기준으로 사용되고 서로 다른 종류의 연결을 통해 전파된다.

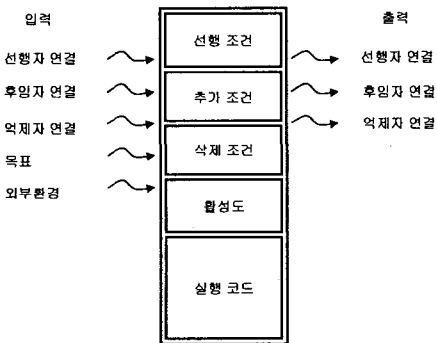


그림 1. 행동 네트워크의 노드.

행동 네트워크를 이용하여 행동을 선택하는 과정은 그림 2에 설명되어져 있다. 처음 단계에서 외부환경과 내부목표로부터 입력신호가 행동으로 전달된다. 두 번째 단계에서

내부연결의 종류에 따라 활성화도 전파를 수행한다. 활성화도가 무한히 커지는 것을 막기 위해 평균이  $\pi$ 가 되도록 정규화 한다. 세 번째 단계에서는 모든 선행조건이 참이면서 활성화도가 임계치보다 큰 행동들을 선택한다. 선택된 행동이 없다면 임계치를 10% 줄여서 처음단계로 돌아간다. 선택된 행동이 존재할 때까지 위의 과정을 반복한다. 선택된 행동들 중에서 가장 높은 활성화도를 가지는 행동이 로트의 실행권한을 가진다.

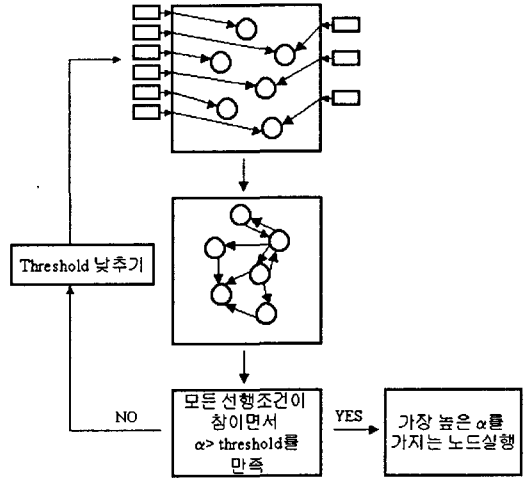


그림 2. 행동 네트워크의 행동선택 과정 ( $\alpha$  = 활성화도값).

행동 선택과정에서 활성화도의 전달은 파라미터에 의해 조정된다. 외부환경으로부터 행동으로 들어오는 입력신호의 값은  $\phi$ 에 의해 곱해진다. 내부목표로부터 들어오는 신호의 값은  $\gamma$ 에 의해 곱해진다. 내부목표로부터 들어오는 억제신호의 값은  $\delta$ 에 의해 곱해진다. 선행자 연결, 후임자 연결, 억제자 연결은 각각  $1, \frac{\phi}{\gamma}, \frac{\delta}{\gamma}$ 에 의해 곱해진다.

명제  $p$ 는 기본적으로 외부환경의 상태와 내부목표를 나타낸다. 여러 개의 명제가 모여서 추상화된 형태의 조건을 설정하기도 한다. 외부환경 상태를 설정하는 명제는 0과 1의 이진값으로 이루어지며, 내부목표를 설정하는 명제는 실수값으로 설정된다. 외부환경과 내부목표의 설정은 전체 행동 네트워크의 성격을 결정하는 중요한 요소들이다.

행동이 실행되기 위해서는 선행조건이 모두 참이어야만 하기 때문에, 선행조건의 설정은 행동의 실행에 중요한 영향을 행사한다. 내부 목표는 반드시 목표의 달성에 도움이 되는 행동의 활성화에만 영향을 준다. 즉, 행동이 수행되었을 때 목표의 활성화도가 올라갈 수 있는 경우에만 목표와 행동 사이의 연결이 존재한다.

파라미터의 설정은 행동 네트워크의 성능을 결정하는 요소이지만, 결정적이지는 않다. 행동 네트워크의 성격은 외부환경 상태 변수와 내부 목표의 설정 그리고 행동들간의 내부관계에 가장 크게 영향을 받는다. 물론 내부목표나 외부상태 변수로부터 들어오는 입력의 영향을 조정하기 위해서 파라미터는 유용하게 사용된다.

규칙기반 결합방법은 고차원 행동을 구현하기 위해 사용하는 방법중의 하나이며, 하나 이상의 IF-THEN-ELSE

구문을 통해 행동의 실행을 결정한다[4]. IF-THEN-ELSE 규칙은 반드시 한번에 한 개의 행동만을 수행하도록 하며, 부과적인 계산이 필요하지 않기 때문에 행동 네트워크에 비해 빠른 수행시간을 가진다.

#### 4. 실험 및 결과

로봇의 행동제어를 위하여 이 논문에서는 다음과 같이 4개의 기본행동을 정의하였다.

- 장애물 피하기: 주변에 장애물이 있을 경우 충돌 없이 피해간다.
- 빛 따라가기: 주변에 빛이 강한 곳으로 이동해 나가며 배터리 충전영역으로 이동하도록 하는 중요한 역할을 한다.
- 직진하기: 주변에 장애물이 없을 경우, 직진한다.
- 배터리 충전하기: 배터리 충전영역안에 로봇이 있을 경우, 로봇의 배터리를 초기값으로 설정한다.

기본 행동 모듈중 “장애물 피하기,” “빛 따라가기”는 셀룰라 오토마타 기반 신경망을 이용하여 진화되었고, “배터리 충전하기,” “직진하기”는 단순하기 때문에 프로그래밍 되었다[5].

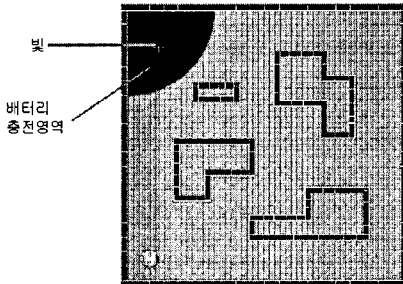


그림 3. 시뮬레이션 환경.

그림 3은 시뮬레이션 환경을 보여준다. 로봇이 수행해야 하는 작업은 장애물과의 충돌을 최소화하면서, 배터리 수준이 낮은 경우 배터리 충전을 수행하여 지속적인 작업이 가능하도록 하는 것이다. 시뮬레이션 환경에서 로봇이 한번 행동을 수행할 때마다, 배터리 수준은 1 감소한다. 초기 배터리 값은 7000이며, 로봇이 배터리 충전영역에서 “배터리 충전하기” 행동을 수행하면 다시 원래값으로 돌아온다. 배터리 충전영역은 검은색으로 표시되어 있으며, 로봇이 찾아올 수 있도록 빛이 설정되어져 있다.

행동 네트워크는 4개의 기본 행동 모듈, 5개의 외부환경 센서, 2개의 내부목표로 구성되어져 있다. 그림 4는 고차원 행동 구현을 위한 행동 네트워크를 보여준다. 그림 4에서 실선은 목표와 행동의 연결 또는 선행자 연결을 나타내고, 점선은 외부환경 센서와 행동의 연결 또는 후임자 연결을 나타낸다. 로봇의 목표가 배터리 수준을 유지하면서 오래 생존하는 것이기 때문에 “배터리 충분한 상태”와 “배터리가 재료가 아닌 상태”를 목표로 설정하였다.

소모된 배터리의 상태를 회복하기 위해서는 “배터리 충전하기” 행동이 수행되어야 하기 때문에 두 개의 목표는 이 행동으로 입력 신호를 전달한다. 4개의 기본 행동모듈은 하나 이상의 외부환경 센서를 선행조건으로 가지며 행

동이 실행되기 위해서는 반드시 참이어야 한다. 행동 네트워크의 파라미터는 수작업을 통해  $\phi=1.2, \gamma=0.8$ 로 설정하였다.

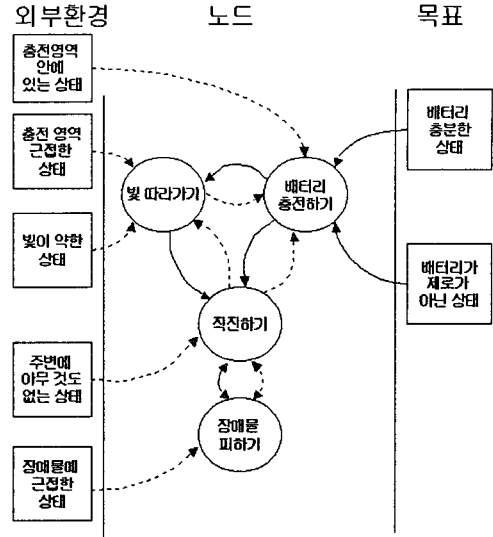
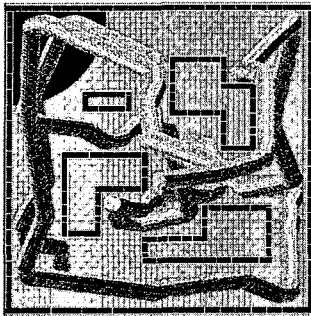


그림 4. 고차원 행동구현을 위한 행동 네트워크.

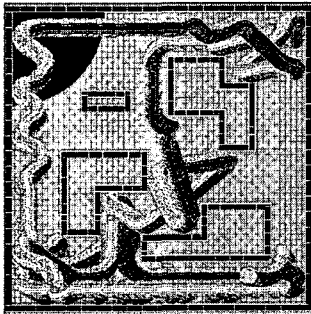
IF (배터리 충전 영역) 배터리 충전하기 모듈 실행  
 ELSE IF (배터리 <  $\alpha$ ) AND (빛 감지 센서의 최솟값  $\leq \gamma$ )  
 IF (거리 감지 센서의 최댓값  $\leq \beta_1$ ) 빛 따라가기 모듈 실행  
 ELSE 장애물 피하기 모듈 실행  
 ELSE IF (거리 감지 센서의 최댓값  $\leq \beta_2$ ) 직진하기 모듈 실행  
 ELSE 장애물 피하기 모듈 실행

위의 IF-THEN-ELSE 규칙은 4개의 기본 행동들을 주어진 작업에 맞게 규칙기반 결합한 것이다. 로봇은 배터리 충전영역 안에 있게되면 “배터리 충전하기” 행동을 수행한다. 배터리가 낮고 빛이 가까이 있으며 장애물이 없다면 “빛 따라가기” 행동을 수행한다. 배터리가 높거나 빛이 일정 거리 이상 떨어져 있으면서 장애물이 없다면 “직진하기” 행동을 수행한다. 이외의 경우에 로봇은 “장애물 피하기” 행동을 수행한다.  $\alpha, \beta_1, \beta_2, \gamma$ 는 환경에 맞게 설정 해주어야 하는 파라미터들이다.  $\alpha$ 는 최대 배터리 양의 2/3,  $\beta_1$ 은 200,  $\beta_2$ 는 250으로,  $\gamma$ 는 450으로 정하였다.

실험은 두 가지 서로 다른 환경에서 수행되었다. 첫 번째 환경은 그림 3과 같은 환경이고, 두 번째 환경은 그림 3에서 배터리 충전영역으로 가는 길목에 장애물을 추가한 것이다. 그림 6과 7은 두 가지 서로 다른 환경아래에서 로봇의 이동경로를 보여준다. 처음 환경에서 로봇은 행동 네트워크를 이용하거나 규칙기반 결합을 한 경우 모두 성공적인 작업을 수행하였다. 그러나 변화한 환경아래에서 규칙기반 결합의 경우 성공적으로 작업을 마치지 못했고 행동 네트워크의 경우 성공적으로 작업을 마쳤다. 그림 8은 처음환경에서 두 가지 결합모델의 행동 선택 순서를 보여준다. 그림 8에서 규칙기반 결합방법은 일정한 행동선택 패턴을 지니고 있고 행동 네트워크는 비교적 다양한 행동선택 패턴을 지니고 있다.

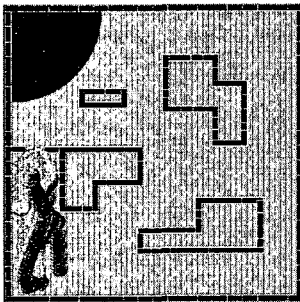


(a) 규칙기반 선택

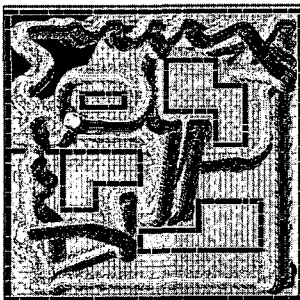


(b) 행동 네트워크

그림 5. 본래 환경에서 로봇의 이동경로.



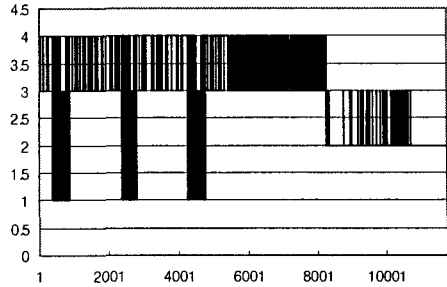
(a) 규칙기반 선택



(b) 행동 네트워크

그림 6. 변화한 환경에서 로봇의 이동경로.

본래 환경에서 규칙 기반 행동 선택



본래 환경에서 행동 네트워크

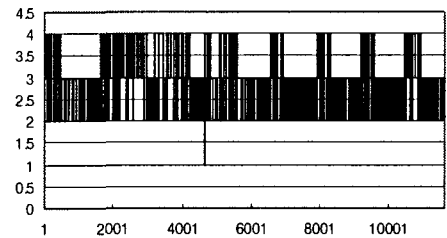


그림 7. 행동선택 순서 비교 (1 = 배터리 충전하기, 2 = 빛 따라가기, 3 = 장애물 피하기, 4 = 직진하기).

### 5. 결론 및 향후연구

고차원의 작업을 수행하기 위해 행동 네트워크 모델은 외부상태, 내부목표, 내부 연결관계를 정의한다. 규칙기반 결합방법은 IF-THEN-ELSE 규칙을 바탕으로 상황에 적합한 행동을 선택하도록 한다. 본 실험에서는 환경을 변화하여 두 가지 서로 다른 결합모델의 적응성을 비교하였다. 실험결과 행동 네트워크 모델이 변화하는 환경에서 적절히 대처하는 것을 확인할 수 있었다.

#### 참고문헌

- [1] M. Niculescu and M. J. Mataric, "Learning and interacting in human-robot domains", to appear in *Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 2001.
- [2] P. Maes, "How to do the right thing," *Connection Science Journal*, Vol. 1, No. 3, pp. 291-323, 1989.
- [3] Klaus Dorer, "Behavior networks for continuous domains using situation-dependent motivations," *International Joint Conference on Artificial Intelligence (IJCAI'99)*, pp. 1233-1238, 1999.
- [4] G.-B. Song and S.-B. Cho, "Rule-based integration of multiple neural networks evolved based on cellular automata," *FUZZ-IEEE'99*, pp. 791-796, August 1999.
- [5] S.-B. Cho and G.-B. Song, "Evolving CAM-Brain to control a mobile robot," *International Journal of Applied Mathematics and Computation*, Vol. 111, pp. 147-162, May 2000.