

모바일 로봇의 상자 밀어내기 작업을 위한 리액티브 플랜의 자동 생성

Automatic Construction of Reactive Plans for A Box-Pushing Task of A Mobile Robot

차 병근 · 서 일 홍 · 이 상 훈
Byoung Keun Cha · Il Hong Suh · Sanghoon Lee

Abstract - Reactive plans for a box-pushing task of a mobile robot are automatically built up, where sequential action plans are found in a configuration space by A* algorithm for various initial configurations. Then, conjunction of conditions to associate with a same behavior are found by a back tracking algorithm. And corresponding reactive plans are generated. Finally, a clustering technique is applied to identify which reactive plan should be applied for a given perceptual condition. Several simulation results are shown to justify our proposed approach.

Key Words : Reactive Plan, Configuration Space, Action Selection

1. 서론

지능 로봇은 다양하고 태스크(task)를 수행할 수 있어야 한다. 그러나 대부분 지능 로봇은 제한되거나 고정된 태스크만을 수행하고 태스크가 달라지면, 새로운 태스크 프로그램을 입력받아야 한다. 태스크 성취가 매우 중요한 미래 지향 지능 로봇(서비스 로봇 또는 가사 지원 로봇)은 다양하고 복잡한 태스크 수행을 요구 받는다.

로봇이 수행해야 하는 태스크는 일반적으로 일련의 순서를 갖는 행동들로 표현되며, 태스크 프로그램은 그러한 순서 행동들로 작성된다. 그리고 실제 로봇은 복잡한 환경에서 태스크를 수행하기 때문에 대부분 복잡하고 다양한 순서 행동 집합들을 가진다. 이와 같이 주어진 태스크를 수행하고 문제를 해결하기 위하여 행동들의 순서를 정하는 일을 계획 작성(planning)이라고 한다[1].

로봇이 계획 작성을 얼마나 잘 하는가가 로봇이 지능적인 행동을 할 수 있는가의 척도가 된다. 그러나 실제 환경은 불확실성이 많고 복잡하며 또한 변화가 다양하기 때문에 모든 상황을 고려하여 계획을 작성하는 것은 매우 어렵다. 이에 따라 환경의 불확실성과 변화를 고려한 conditional planning, replanning, 계획 작성과 행동 수행의 실시간성을 보장하기 위한 continuous planning, 그리고 태스크와 환경의 복잡함과 상태의 변화를 고려한 partial-order planning, hierarchical planning 등의 방법들이 연구되고 있다[1][2]. 그런데 이러한 계획 작성 방법들은 개발자가 모든 상황을 고려하여 계획 작

성에 필요한 행동들을 작성해야 한다. 그러나 개발자가 모든 순서 행동 집합들을 사전에 작성하는 것은 불가능 하며, 따라서 로봇이 스스로 순서 행동 집합을 작성할 수 있는 능력을 필수적으로 가져야 한다..

본 연구에서는 태스크 수행을 위해 구축한 순서 행동 집합을 리액티브 플랜(reactive plan)으로 정의하고, 로봇이 리액티브 플랜을 스스로 구축 하는 방법을 제안한다.

2. 리액티브 플랜(reactive plan)

2.1 리액티브 플랜의 정의

리액티브 플랜[3][7][8]이란 태스크의 시작과 목표를 통해 전체 태스크를 진행하는데 필요한 행동들의 집합, 행동들의 연관 관계, 그리고 순서들을 찾아내고 찾아진 순서 행동 집합들을 통해 태스크를 수행할 수 있는 구조로 작성해 놓은 것이 리액티브 플랜이다.

그림 1은 각각의 행동 집합들을 연관 관계에 따라 구조적으로 나타낸 리액티브 플랜의 예를 보여준다.

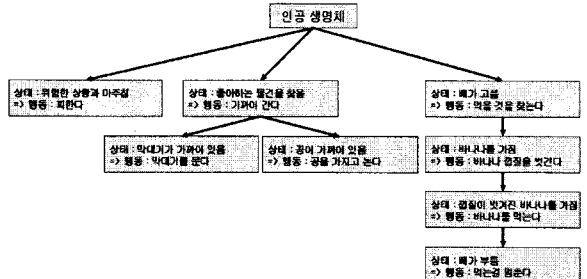


그림 1. 리액티브 플랜(reactive plan)

2.2 리액티브 플랜의 특성

저자 소개

- * 차병근 : 漢陽大學 情報通信工學科 碩士課程
- ** 이상훈 : 漢陽大學 電子電氣制御計測工學科 博士課程
- ***서일홍 : 漢陽大學 情報通信工學科 助教授 · 工博

1) 목표 지향적(goal oriented) 특성

리액티브 플랜에서 일련의 순서 행동들은 동일한 목적을 가지는 일련의 행동끼리 묶여 있고, 이는 목표 지향적 특성을 가진다. 그림 1.에서 보듯 배가 고른 상황에서 배를 채우기 위한 일련의 행동들은 배를 채우고자 하는 목표를 향해가도록 이루어져 있다.

2) 상황 의존적인 특성

순서 행동들은 상황에 의존적인 특성을 가지고 있다. 즉, 리액티브 플랜을 구성하는 일련의 행동들은 상황에 따라 유동적으로 선택되어 진다. 따라서 리액티브 플랜은 상황 인지와 판단에 매우 효율적인 특성을 가진다. 그림 1.과 같이 인공 생명체가 바나나를 얻으면 껍질을 벗기지만, 껍질이 벗겨진 바나나를 얻는다면 굳이 바나나를 가져야 하는 상위 행동을 하지 않는다.

3) 계층적이고 상호 배타적인 구조적 특성

리액티브 플랜은 다양한 태스크 수행 시에 필요한 여러 행동들(다른 목적을 위해 공유되는 행동, 동일한 목적을 위해 상황에 따라 서로 다르게 나타나는 행동, 또는 서로 동시에 일어날 수 없거나 순서적으로 먼저 일어나는 행동, 등)로 표현되어진다. 이때 순서 행동 집합들은 그 전체로서 리액티브 플랜을 의미하지만, 각각의 순서 행동 집합으로도 하위 리액티브 플랜의 의미를 가진다. 그림 1.에서 보듯이 각각의 하위 리액티브 플랜은 그 자체로서 하나의 하위 태스크를 수행할 수 있다.

2.3 리액티브 플랜의 장점

리액티브 플랜은 위의 3가지 특성을 통해 다른 계획 작성(planning)의 연구에서 추구하는 목표를 만족할 수 있다. 리액티브 플랜 자체가 순차성을 가지는 순서 행동 집합이기 때문에 계획 전체를 하나하나의 행동들로 일일이 작성할 필요가 없다.

또한, 태스크에 필요한 모든 행동집합들이 리액티브 플랜으로 작성되어 있으므로 상황에 따른 계획의 부분적인 재수정이 용이하고 빠르다. 즉, 불확실성에 대한 대처가 빠르고, 계획 수정 및 로봇의 반응에 실시간성을 보장 한다. 그리고 목적에 맞는 행동을 상황에 따라 계층적으로 선택할 수 있어 임무 수행의 강인함도 보장해준다.

2.4 동물 행동학 기반의 행위선택 아키텍처

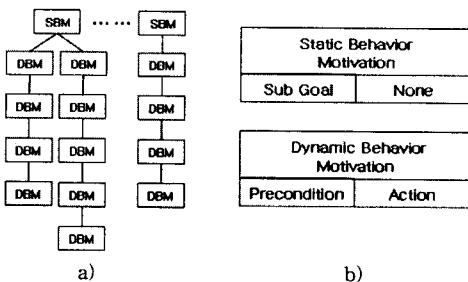


그림 2. a)BM(behavior motivation) tree, b)SBM과 DBM

본 논문에서 제안하는 리액티브 플랜의 구조는 선행 연구한 동물 행동학 기반의 행위선택 아키텍처[3][5][6]에 행동 선

택을 위한 BM(behavior motivation: 행위 동기) 모듈(module)로 들어가 있다. 본 연구에서 수행한 B.P 태스크 시뮬레이션에서도 행위선택 아키텍처를 가진 agent를 이용하였다. 이 모듈에서 리액티브 플랜은 SBM(static behavior motivation)과 DBM(dynamic behavior motivation)으로 이루어진 BM tree구조로 표현되어진다. SBM은 태스크 또는 동기를 나타낸다. DBM은 SBM을 수행하기 위한 순차적인 행동들로 구성되어 있다. DBM은 내부적으로 상황을 인지하는 자극인식 필터(perception filter)와 그 상황(자극)에 적합한 행동으로 이루어져 있다.

3. 리액티브 플랜 자동 구축 방법

로봇이 주어진 태스크를 수행하기 위해선 태스크와 연관된 주위 환경과 상호 작용을 해야 한다. 이때 나타나는 상호 작용은 로봇과 로봇 주위 환경과의 상대적인 관계를 의미한다. 태스크 진행시 발생하는 상호 작용 중에는 임무 수행에 있어 필수적인 요소들이 존재한다. 그리고 이러한 요소들은 태스크 종류, 목적, 그리고 수행에 따라 다양하게 나타난다. 본 연구에서는 로봇과 박스, 그리고 목표지점까지의 상대적인 위치 관계를 로봇과 상호 작용하는 환경의 요소로 보았다.

3.1 모양 공간(configuration space)의 정의와 형성

태스크 수행에 필요한 다양한 환경을 표현하기 위하여 태스크를 모양 공간(Configuration Space)으로 맵핑을 하였다. 즉, 모양 공간은 로봇의 태스크에 따라서 그 차원이 달라진다. 모양 공간에서의 각 차원은 태스크의 임무 수행 시 환경과 상호 작용하는 요소로 나타낼 수 있다. 박스를 골까지 미는 태스크의 경우 로봇이 주목해야 하는 환경은 박스와 골이다. 그리고 그 환경에서 상호 작용을 일으킬 수 있는 요소는 박스와 골의 2차원 위치 관계이다. 본 연구에서는 로봇과 박스와의 위치 관계(거리, 진행 방향에 따른 상대적인 각도), 로봇과 골과의 위치관계(거리, 진행 방향에 따른 상대적인 각도), 그리고 로봇에서 본 박스와 골과의 위치 관계(박스와 골과의 거리, 로봇에서 본 박스와 골의 상대적인 각도)를 이용하여 6차원의 위치 관계를 나타내는 모양 공간을 정의하였다.

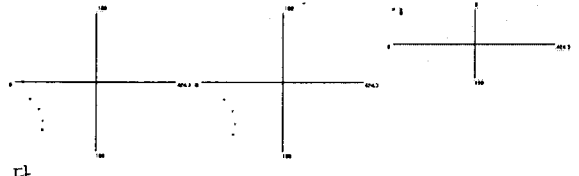


그림 3. 모양 공간 a)로봇과 박스의 거리, 상대적인 각도 관계의 모양 공간, b)로봇과 골의 거리, 상대적인 각도 관계의 모양 공간, c)박스와 골과의 거리, 로봇에서 본 박스와 골의 상대적인 각도

3.2 모양공간(configuration space)에서의 목표점 탐색

6차원으로 생성된 모양 공간에서는 로봇이 태스크를 수행하게 되는 시작 위치의 상태와 태스크를 마치게 되는 목표 상태를 표현 할 수 있다. 여기서부터 A* 알고리즘을 이용하여 통해 현 상태에서부터 골 상태로의 최적의 값을 갖는 이동

경로를 찾았다. 이동 경로를 찾을 때는 아래 식 1)을 이용하여 이동 경로에 대한 비용을 계산하였다.

$$\text{Path Cost in Robot \& Box C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2}$$

$$\text{Path Cost in Robot \& Goal C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2}$$

$$\text{Path Cost in Box \& Goal C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2} \quad 1)$$

$$\text{Heuristic Cost in Robot \& Box C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2}$$

$$\text{Heuristic Cost in Robot \& Goal C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2}$$

$$\text{Heuristic Cost in Box \& Goal C-Space} = \sqrt{(s - s')^2 + (\text{ang} - \text{ang}')^2}$$

Total Cost = Path Cost + Heuristic Condition
 = Path Cost in Robot & Box C-Space + Path Cost in Robot & Goal C-Space
 + Path Cost in Box & Goal C-Space + Heuristic Cost in Robot & Box C-Space
 + Heuristic Cost in Robot & Goal C-Space + Heuristic Cost in Box & Goal C-Space

그리고 최적의 경로를 이동하면서 로봇의 행동에 변화가 생기는 시점을 로봇이 주위 환경과 상호 작용을 일으키는 시점으로 보고 그 때의 상황 데이터를 모양 공간으로부터 추출하였다.

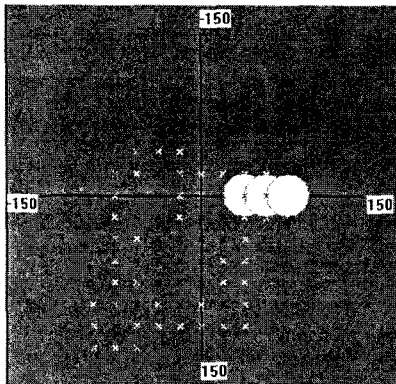


그림 4. 탐색을 통한 태스크의 수행

3.3 역추적(back tracking)을 이용한 클러스터링(clustering)

탐색을 이용하여 수집한 데이터를 그래프 상에서 분석하여 보면 수집된 상태들이 경로를 형성함을 볼 수가 있었다. 그리고 그 경로들 중 일정 부분이 중복해서 나타남을 알 수 있었다. 예를 들어 로봇이 어떠한 위치에서 시작하는 박스와 골의 일직선상으로 이동하게 될 것이고, 또한 로봇이 박스를 밀기위하여 진행방향을 조정하고 그 후 박스를 끝까지 밀게 된다. 즉, 태스크를 수행하기 위해서는 어떤 특정한 상황 또는 행위에 앞서서 먼저 나타나야 하는 행위 또는 상황들이 있고 그 상황과 행위가 해결, 수행되어야 태스크가 진행이 될 수 있다는 것을 의미한다. 이와 같은 상황 및 행동들을 분류해 내기 위해서는 그림 5.의 알고리즘에 따라 태스크가 끝나는 목표 상태에서부터 역 추적을 해나간다. 그리고 동일한 목표 상태를 가지는 그 전 상태들을 하나의 집합으로 묶어 나

```
function Same-Goal-Check returns SBM set
inputs : nodes (current_state-action-next_state), goal_state
variables : Next_State, queue, SBM

queue ← all nodes
Next_State ← goal_state

while( queue is not empty ) do
    for all nodes do
        if Next_State is same to node's next_state
            then clustering nodes to SBMi
            and erase nodei from queue
            and increase i
        increase k
    return SBM set
```

그림 5. 역추적 알고리즘

모양 공간에서의 목표점 탐색 실험에서는 80 * 120 픽셀의 지도를 20 * 20픽셀 크기의 셀(35개)로 나눈 실험 환경을 이용하였다. 역추적에서 사용한 상황 데이터들은 위의 실험 환경에서 로봇과 박스 그리고 골의 위치를 이동시키고, 로봇의 초기 진행 방향을 45도 각도씩 변화시키며 수집하였다. 그림 6.은 본 연구에서 역추적을 이용한 클러스터링 방법으로 구한 리액티브 플랜을 보여준다. 생성된 리액티브 플랜의 내부에 있는 SBM과 DBM의 개수는 다음의 표 1.과 같이 나타난다.

	SBM 1	SBM 2	SBM 3	SBM 4	SBM 5	SBM 6	SBM 7	SBM 8
DBM	3	7	5	8	10	10	5	2

표 1.

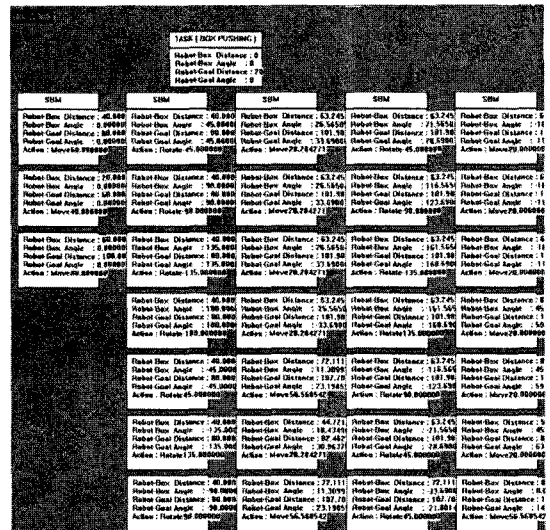


그림 6. 역추적을 통해 클러스터링된 SBM과 DBM

3.4 유사 상황 기반의 리액티브 플랜 수행

모양 공간(configuration space)에서 찾은 리액티브 플랜(reactive plan)의 DBM안에는 각각의 자극 인식 필터가 들어 있다. 처음 만들어진 리액티브 플랜의 경우는 이 자극인식

필터에 들어있는 상태정보는 단 하나의 상태 정보만을 가지고 있다. 실험 환경에서의 모든 상황 정보를 이용하여 리액티브 플랜을 만든다면 무수히 많은 상태 정보로 인해 모양 공간에서의 탐색이 매우 어려워지며 또한 리액티브 플랜의 DBM도 무수히 많아지게 된다. 하지만 태스크를 수행 시에 나타나는 많은 상태들 중에는 로봇이 행동하기 위해 사용하는 파라미터값(모터의 회전수, 로봇의 회전 각도, 또는 로봇의 이동 거리, 등)에 상관없이 행동(좌회전, 우회전, 또는 전진, 등)자체에 따라 유사한 상태로 볼 수 있는 많은 상태들이 존재한다. 예를 들어 로봇의 정면에서 오른쪽으로 박스가 있다면 그 거리와 각도가 얼마가 되었든 로봇은 우회전과 전진이라는 행동들 통해 박스로 이동하게 된다. 이때 로봇의 위치는 박스와 위치관계가 정면에서 오른쪽을 만족하는 상황이라면 모두 유사한 상황이라고 볼 수 있다. 즉, 로봇은 상황 인지를 통해 마주친 상황과 유사한 상황을 DBM의 자극인식 필터를 통해 찾고 그에 해당하는 행동을 취함으로써 태스크를 완수할 수 있다[3][4][8].

본 연구에서는 태스크 수행 시 발생하는 임의의 상태와 perception filter와의 유사성을 찾기 위하여 6차원 모양 공간에서의 거리함수를 이용하였다. 구체적으로 상태 정보는 6차원의 모양 공간에서의 하나의 점을 나타내며, 현재 상태를 나타내는 점과 자극인식 필터에 들어있는 각각의 상태를 나타내는 점들과의 거리를 아래 있는 식 2)를 통해 구한다.

$$\text{similarity_value} = \sqrt{((\text{current_state}) - (\text{perception_filter_state}))^2} \quad 2)$$

similarity_value: 현재 상태와 자극인식 필터의 상태와의 유사도
i : index of 자극인식필터

그리고 거리의 최소값을 갖는 자극인식 필터의 DBM을 선택하게 된다. 그리고 현재 상태는 선택된 자극인식 필터에 새로운 상태 정보로 추가가 되어 진다. 로봇의 행동의 경우 수행 중이던 행동은 멈추게 되고 선택된 DBM의 행동을 수행하게 된다.

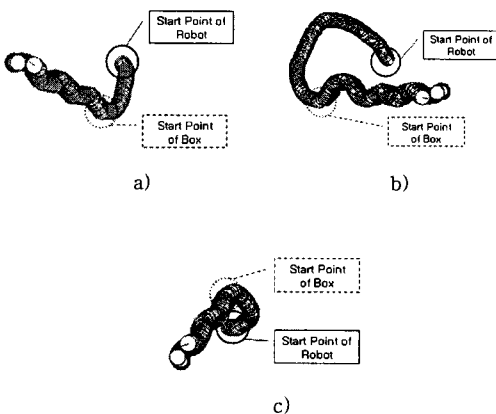


그림 7. 리액티브 플랜이 구축시 사용하지 않았던 태스크 환경을 통한 리액티브 플랜의 검증

그림 7.은 로봇이 역추적을 통해 생성되어진 리액티브 플랜을 이용하여 유사 상황을 클러스터링 하면서 태스크를 수

행하는 예를 보여준다. 그림 7.에서 보여주고 있는 3개의 태스크 환경은 모양 공간에서의 목표점 탐색 실험에서 사용하지 않았던 임의의 태스크 환경이다. 또한 그림 7. a)와 b)에서 보듯이 로봇이 탐색 실험 시에 제한하였던 지도의 범위를 넘어가는 초기 위치에서 시작을 하여도 태스크를 수행함을 볼 수 있다.

4. 결론

본 논문에서는 로봇에게 태스크를 주어지면 스스로 태스크 자체를 직접 수행 할 수 있는 리액티브 플랜으로 구축하는 방법을 제안하였다. 그를 위해 태스크를 모양 공간(configuration space)으로 맵핑하여 표현하였다. 그리고 모양 공간 안에서 태스크의 시작점으로 부터 목표점에서의 경로를 탐색하고 경로에 따라 태스크 수행 시 나타나는 상태 데이터를 추출하였다. 그 후 역추적을 통해 태스크를 구성하는 행동들의 집합을 찾아내고 행동들의 연관 관계에 따라 순서행동 집합인 리액티브 플랜을 구축하였다. 그리고 구축된 리액티브 플랜을 동일 태스크의 임의의 상황에 대해서 작동함을 검증하였다.

참 고 문 헌

- [1] S. Russell, Peter Norving "Artificial Intelligence : A Modern Approach", pp. 375-459, 2003
- [2] R. C. Arkin, "Behavior Based Robotics", MIT Press, Cambridge, MA, 1998
- [3] 권우영, 민현석, 장국현, 이상훈, 서일홍, "지능형 소프트웨어 로봇을 위한 행동학습구조," 대한전기학회 하계학술대회 논문집, 2002. 7.10~12
- [4] 김민조, 권우영, 이상훈, 서일홍, "인공생명체를 위한 행동선택 구조," 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집, 2002.11.30
- [5] B. M. Blumberg "Old Ticks, New Dogs : Ethology and Interactive Creatures", MIT, Media laboratory, Learning and Common Sense Section, 1996
- [6] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, RA-2:14-23
- [7] J. J. Bryson "The Behavior Oriented Design of Modular Agent Intelligent", 2004
- [8] I. H. Suh, S. H. Lee, W. Y. Kwon, and Y. J. Cho, "Learning of Action Patterns and Reactive Behavior Plans via a Novel Two-Layered Ethology-Based Action Selection Mechanism", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1232-1238, August 2-6, 2005, Edmonton, Canada