

자율로봇을 위한 계획기반의 제어 구조

김하빈^o 김인철
경기대학교
{talkable^o, kic}@kyonggi.ac.kr

A Plan-Based Control Structure for Autonomous Robots

Habin Kim^o, Inchul Kim
Kyonggi University

요 약

본 논문에서는 주문형 작업계획 생성기와 실행기의 연동을 통해 Sony사에서 개발된 AIBO로봇의 행동을 제어하는 계획기반의 제어구조를 개발하였다. 그리고 이 제어구조의 효과를 알아보기 위해, 미로 속에서 경로를 계획하고 이동하는 작업에 이 제어 구조를 적용하고 실험하였다.

1. 서론

지능 로봇에 대한 제어구조는 복잡도를 고려하여 몇 개의 서로 다른 계층들로 나누어 구현될 수 있으며, 각 계층은 각기 다른 레벨의 제어문제를 다룬다. 일반적으로 제어구조의 최상위 계층은 작업(task) 단위의 제어를 다루며, 따라서 작업 관리자(task manager)로도 불린다. 이 계층에서는 로봇이 처한 현재 상황과 작업 목표를 고려하여 향후 수행할 작업 계획을 수립하고 계획된 바대로 작업이 성공적으로 실행되고 있는지 모니터링하며, 필요한 경우 새로운 상황에 맞도록 재계획(replan)하는 기능들을 수행한다. 일반적으로 로봇 작업 계획(plan)이란 시간과 자원에 관한 제약(constraint)을 만족하면서 주어진 작업 목표들을 달성할 수 있는 일련의 동작(action)들을 말한다. 반복되는 일상적인 작업들이나 실시간성이 요구되는 작업들을 수행하기 위해서는 매번 작업계획을 새로 생성하는 것 보다 한번 실행에 성공한 적이 있는 계획을 기억하거나 수행할 계획을 미리 정의해주는 방법이 더 효과적이다. 그러나 저장하고 있는 계획들을 실행하는 도중에 적용 가능한 모든 계획들이 실패로 끝나거나 더 이상 적용할 새로운 계획이 없는 상황에 도달하면, 주문형 계획 생성기(on-demand planner)의 도움을 받아 이러한 상황에 맞는 새로운 계획을 생성할 필요가 있다. 본 논문에서는 이러한 개념을 바탕으로 Sony사에서 개발된 AIBO로봇을 위한 계획기반의 제어구조를 개발하였다.

이 AIBO 로봇플랫폼은 64비트 RISC 프로세서와 64MB 메모리를 가지고 있으며, 각각 3개의 관절(joint)을 포함한 4개의 다리를 가지고 있다. 또 이 로봇플랫폼은 시각인식을 위한 CMOS 컬러 이미지 센서를 비롯하여, 장애물과의 거리측정이 가능한 적외선 센서 등 다수의 센서들을 몸에 부착하고 있으며, 스피커와 마이크로폰을 통해 소리를 발생시키거나 들을 수 있다. 또한 얼굴에는 감정표현이 가능한 LED패널을 가지고 있고, 몸통부에는 무선 랜(wireless LAN) 통신기능을 내장하고 있다. AIBO ERS-7M3의 모든 서비스 프로그램들은 실시간 운영체제인 Aperios 위에서 실행

되며, 개발자를 위한 프로그래밍 환경으로 C++ 기반의 Open-R과 Remote Framework 등을 제공하고 있다.

본 연구에서 이러한 AIBO로봇을 이용하여 이루고자 하는 작업들은 대부분 미로(maze) 환경에서 로봇의 초기 위치에서 출발해 주어진 목적지까지 장애물을 피해 최단경로로 찾아가는 일들이다. 로봇이 이러한 임무를 성공적으로 수행하기 위해서는 자신이 처음 놓여지는 위치를 파악하는 일, 이동 가능한 인접 경로점(way point)들을 찾아내는 일, 밀어서거나 몸을 돌려 이동을 준비하는 일, 선택된 경로점까지 벽이나 다른 장애물들을 피해 걸어가는 일 등 다양한 지능행위들이 요구된다.

2. 계획기반의 제어구조

AIBO 로봇을 위한 전체 제어구조는 [그림 1]과 같이 하나의 계층구조를 이룬다. AIBO 로봇의 세부적인 모션(motion)과 센서 데이터(sensor data)를 다루는 제어부의 최하위 계층은 Sony사에서 AIBO 프로그래밍을 위해 제공하는 Remote Framework 층이다. Remote Framework층은 다시 공룡의 Virtual AIBO Server와 이에 대한 서비스를 요청하는 Remote Framework API층으로 구성된다. 이 계층 위에 본 연구에서 제시하는 계획기반의 제어부가 위치한다. 먼저 맨 아래에 로봇의 행위모듈 층이 존재하는데, 이 계층은 Remote Framework API를 이용하여 C++로 구현된 로봇의 다양한 행위모듈들로 구성된다. AIBO 로봇을 위한 유용한 행위들은 수없이 많이 존재할 수 있으나 본 연구에서는 앞서 설명한 데모용 작업환경에 필요한 행위들만 한정하여 구현하였다. 본 연구의 AIBO로봇 작업환경은 주로 목적지까지 경로를 계획하고 이동하는 작업들이 대부분이며, 이를 위해 로봇은 특별한 패턴으로 표시된 경로점(way point)들을 찾아 그래프 탐색(graph search)을 하여야 한다. 각 경로점에 해당하는 패턴은 AIBO로봇의 패턴 학습 기능을 통해 미리 학습시킴으로서 실험 중 이동 가능한 인접 노드를 실시간적으로 인식할 수 있도록 하였다.

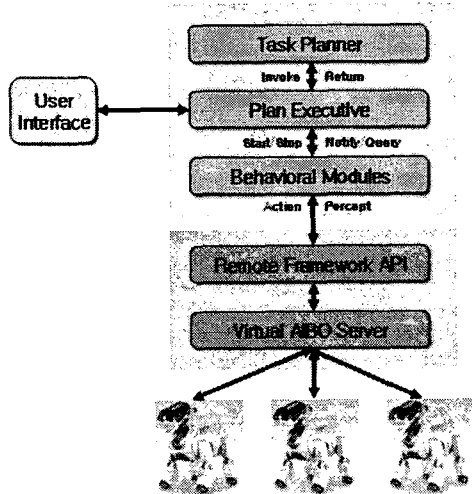


그림 1. 계획기반의 제어구조

행위모듈 계층 위에는 행위 모듈 각각의 실행을 제어하는 계획 실행기 층이 위치한다. 계획 실행기는 각 행위모듈의 실행과 중지 에 대응하는 기본 작업계획들뿐만 아니라 이들을 결합하여 보다 복잡한 상위의 작업 목표를 달성해주는 다양한 복합 작업계획들을 미리 내장하고 있으며, 상황에 맞게 이들을 선택하여 실행하는 역할을 수행한다. 계획기반 제어구조의 최상위 계층은 주문형 작업계획 생성기 층으로서, 실행 중 계획 실행기의 요청이 있을 때 에 필요한 새로운 작업계획을 즉시 생성해주는 역할을 수행한다. 끝으로 사용자의 모니터링과 직접 제어를 지원하기 위해 제공되는 그래픽 기반의 사용자 인터페이스가 제어시스템의 한 부분임 이된다.

3. 계획 생성과 실행

본 연구에서는 Michigan대학교에서 개발된 UM-PRS[3]를 AIBO 로봇 제어를 위한 계획 실행기로 채용하였다. UM-PRS는 높은 목표-지향성(goal-orientation)과 환경변화에 빠른 반응성 (reactivity)을 보이는 대표적인 계획기반의 에이전트 구조이다. UM-PRS의 인터프리터(interpreter)는 미리 정의된 일반화된 계획 들(plans)과 달성하고자 하는 목표(goal), 그리고 환경과 로봇의 상태정보(world model)를 내부에 유지하면서, 매 순간 가장 적합 한 계획을 결정하고(intention) 이것을 실행한다. 현재 실행중인 계획이라도 환경의 상태나 목표 등의 변화가 있으면 언제든지 실행을 중단하고, 다른 계획을 찾아본다. UM-PRS에서 하나의 계획 은 하나의 프로시저어(procedure)로 기술되는 본체(body)와 계획 이름(name), 목표(purpose), 적용조건(context), 효과(effect), 실패 처리부(failure), 우선순위(priority) 등으로 표현된다. 또 하나의 UM-PRS계획은 본체에 다시 여러 개의 부속 목표(subgoal)들을 포함할 수 있도록 허용하고 있다. 따라서 UM-PRS의 계획실행 과정은 최상위 작업목표를 하위의 작은 세부 작업목표들로 나누어 가면서 각 작업목표에 적합한 구체적인 계획들을 결정하고 실행 하는 것으로 요약할 수 있다.

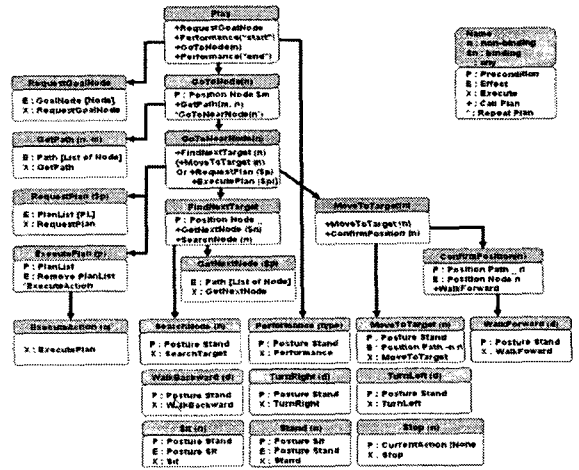


그림 2. 작업계획들의 계층구조

작업계획 실행기인 UM-PRS에 내장되는 작업계획들은 크게 2 가지 종류로 나눌 수 있다. 첫 번째 유형은 하위 계층에 속한 각 행위모듈의 실행을 시작하거나 멈추는 역할을 하는 기본 작업계획(primitive plan)들이다. 두 번째 유형은 이와 같은 기본 계획들을 기초로 보다 상위의 작업목표를 달성하기 위한 복합 계획 (composite plan) 혹은 추상 계획(abstract plan)들이다. 따라서 AIBO 로봇 제어를 위한 작업계획들 간의 관계는 [그림 2]와 같이 계층구조를 이룬다. 주문형 계획 생성기를 호출하여 새로운 작업 계획을 효과적으로 생성하기 위해서 계획 생성기 호출을 위한 특별한 작업계획들을 미리 정의해준다. 이 계획들의 본체는 초기 상태와 목표조건으로 이루어진 하나의 계획문제를 작성하여 주문형 계획 생성기를 호출하고, 하나의 행동 시퀀스를 결과계획으로 받아와 이들을 차례대로 실행하는 절차로 정의된다.

한편, 본 연구에서는 Macedonia 대학교에서 개발된 GRT[5]를 AIBO 로봇을 위한 작업계획 생성기로 이용하였다. GRT는 FF 나 HSP 등과 같이 상태공간(state-space) 상의 탐색을 통해 작업계획을 생성하는 대표적인 휴리스틱 계획 생성기(heuristic planner)이다. 주문형 계획 생성기 GRT 를 이용하여 효율적으로 작업계획을 생성하기 위해서는 수행 가능한 로봇 동작(action)들을 미리 표준 언어인 PDDL(Planning Domain Description Language)로 기술하여 내장하고 있어야 한다. 이를 위해 본 연구에서는 앞서 소개한 각각의 행위모듈을 하나의 PDDL 동작으로 정의 해주었다. 각 PDDL 동작은 STRIPS 의 전통을 따라 전-조건 (precondition)과 효과(effect)로 이루어진 하나의 쌍으로 표현된다. [그림 3]은 로봇의 이동 동작인 move 의 PDDL 정의를 보여 주고 있다. 효율적인 계획 생성을 위해 로봇 동작들에 대한 정의 는 미리 내장하고 있는 반면에, 실제 계획문제(planning problem) 들은 계획 실행기의 요청에 의해 필요할 때마다 즉각적으로 만들어진다. 하나의 계획문제는 일반적으로 현재 상태(current state) 묘사와 목표 조건(goal condition)들의 쌍으로 정의할 수 있다. 하나의 계획문제를 구성하는 이와 같은 정보들은 계획 실행기에 의 해 계획 생성 요청시에 함께 전달된다.

```

(:action move
:parameters (?aibo ?source ?destination)
:precondition
  (and
    (node ?source)
    (node ?destination)
    (link ?source ?destination)
    (link ?destination ?source)
    (on ?aibo ?source)
  )
:effect
  (and
    (not (on ?aibo ?source))
    (on ?aibo ?destination)
  )
)
    
```

그림 3. PDDL로 표현된 동작의 예

4. 실험

로봇을 이용한 실험은 로봇이 이동할 표면의 상태, 기계적 오차, 이동 중 충돌 등으로 인한 이동에 관련된 오작동과 조명의 변화나 시야의 한계 등으로 인한 인식에 관련된 오류가 존재하며, 이러한 오차들간의 연쇄적인 인과 관계로 발생하는 잘못된 움직임으로 인해 작업 계획의 수가 늘어날수록 그 수행 성공률은 급격히 감소할 수 있는 특징이 있다. 본 실험은 물리적 오류를 보일 수 있는 환경에서 BDI구조를 이용한 로봇 제어 방법이 복잡한 작업 계획에서도 적용 될 수 있음을 보이려 한다. 실험환경은 2m² 크기의 정사각형 판 위에 8개의 바닥 표식이 존재하고 근접한 표식 사이의 통로를 제외한 부분은 벽으로 이루어져 있다. 사용자는 로봇의 초기 위치와 최종 목표를 로봇에게 전달하며 더 복잡한 작업 계획을 위해 중간 경유 노드가 추가로 주어질 수 있다. 로봇은 초기 위치에서 중간 경유 노드를 거쳐 최종 위치까지 최단 거리로 이동하게 되는데 이때 가능한 경로의 수는 1에서 6까지이다. 작업 계획을 수행하는 동안 물리적인 충돌로 인해 더 이상 작업 수행이 불가능할 경우와 사용 가능한 작업 계획으로 더 이상 진행되지 못할 경우, 그리고 자신의 위치를 잃어버린 이후 다시, 위치를 찾지 못하거나 잘못된 위치로 인식할 경우, 작업 수행 실패로 간주 하였다.

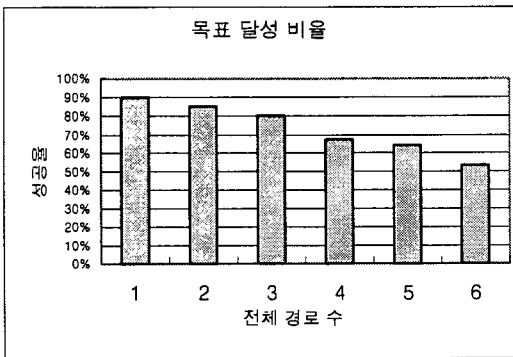


그림 4. 작업 계획 수행 달성 비율

[그림 4]에서 보이는 실험 결과는 전체 작업 수행 경로 중 작업 수행 실패까지의 수행 경로 수의 비율을 보이고 있다. 실험 중 발생할 수 있는 잘못된 인식과 움직임을 제어하지 않았기 때문에 이

동 하여야 할 경로 수의 증가에 따라 그 달성률이 점차적으로 감소하였고, 가장 긴 작업 계획을 가지는 6개 경로 이동 계획 수행의 경우 53%의 달성률을 보였다.

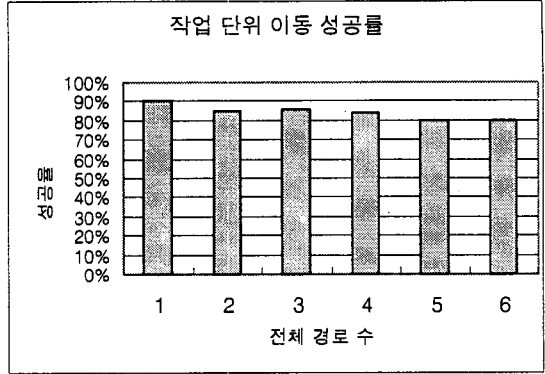


그림 5. 단위 작업 성공률

[그림 5]는 단위 경로 이동을 시도 할 때 마다의 성공 비율을 보이고 있다. 작업 실패로 인한 이후 작업의 시도를 제외한 전체 경로 이동의 시도 횟수는 170회였으며 이 중 단일 노드 이동 성공 횟수는 141회였고 83%의 성공률을 보인다. 이 가운데 비교적 짧은 작업 계획인 1~3의 경로 이동을 가지는 실험의 경우 86%가량의 성공률을 보였으며 4~6의 긴 경로 이동을 가지는 실험의 경우 81%가량의 성공률을 보이고 있다. 작업 길이에 따른 성공률 변화가 적음을 알 수 있다.

5. 결론

본 논문에서는 주문형 작업계획 생성기와 실행기의 연동을 통해 로봇의 행동을 제어하는 계획기반의 제어구조를 개발하였다. 그리고, 미로 속에서 경로를 계획하고 이동하는 작업에 이 제어구조를 적용하였고, 실험을 통해 이 제어구조의 견고성과 효율성을 알아 보았다.

참고문헌

- [1] V. Verma, et al., "Survey of Command Execution Systems for NASA Spacecraft and Robots," Proc. of the Workshop on Plan Execution, Int. Conf. Automated Planning and Scheduling, 2005.
- [2] A. Rao and M. Georgeff, "BDI-agents: from Theory to Practice," Proc. of the 1st Int. Conf. Multiagent Systems, 1995.
- [3] J. Lee, M. Huber, E. Durfee, and P. Kenny, "UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications," Proc. of the Conf. Intelligent Robotics in Field, Factory, Service, and Space, pp. 842-849, 1994.
- [4] M. Huber, "JAM: A BDI-Theoretic Mobile Agent Architecture", Proc. of the 3rd Int. Conf. Autonomous Agents, 1999.
- [5] I. Refanidis and I. Vlahavas, "The GRT Planner: Backward Heuristic Construction in Forward State-Space Planning", Journal of Artificial Intelligence Research, Vol.15, pp.115-161, 2001.