

원격 조작되는 자율주행 이동로봇을 위한 계층별 데이터 추상화 소프트웨어 구조

이상문⁰ 박준호 강순주
경북대학교 전자공학과
{moony_zec}@paigong.knu.ac.kr sjkang@ee.knu.ac.kr

A Layered Data Abstraction Software Architecture for Remote-Controlled Autonomous Mobile Robots

Sang-Moon Lee⁰ Jun-Ho Park Soon-Ju Kang
School of Electronic and Electrical Engineering, Kyungpook National University

요 약

본 논문에서는 자율 주행 로봇을 위한 계층화된 소프트웨어 구조 제안한다. 제안된 소프트웨어 구조에서는 데이터 종류를 추상화 단계에 따라 수치형 데이터, 명제형 데이터, 사실형 데이터로 분류했다. 그리고, 사용하는 데이터의 종류에 따라 계층을 분류해서, 실행 계층, 제어 계층, 추론 계층을 구성하고 각 계층의 기능을 정의했다. 또한 각 계층별 데이터 특성에 따른 고유의 데이터 처리 방법을 적용하였으며, 처리 결과에 대한 계층간 연동 구조에 대해서도 제안 한다. 이러한 계층의 명확한 구분을 통하여 실시간 문제이면서도 복잡한 자료 처리 구조를 가지는 자율 주행 로봇의 소프트웨어 구조를 체계화 하였고 각 계층별 소프트웨어를 콤포넨트화 하여 재 사용성을 높이게 되었다.

1. 서론

로봇의 자율 주행은 목표 지점까지 사용자의 제어 없이 자율적인 판단으로 이동하는 것을 말한다. 이동 로봇에 자율성을 부여하기 위해 여러 가지 소프트웨어 구조가 제안되었다. 크게 3 가지로 구분되며, 계층화 구조[1][3], 동작 기반 구조[4], 앞의 두 구조의 특징을 혼합한 하이브리드형 구조[2]가 그것이다. 기존의 연구들은 모두 계층화 구조에서 기인한 것이며, 계층 간의 구분과 기능 정의가 중요하다. 각 계층을 명확하게 구분하기 위해서는 그 기준이 필요하며, 이를 통해 구분된 계층은 독립성이 보장되어야 한다. 기존의 방법론에서는 주로 기능을 위주로 계층을 구분하였다. 그런데, 자율 주행 로봇에서 계층마다 사용되는 데이터는 형태가 다르며 하위 레벨에서 상위 레벨로 갈수록 추상화되고 함축적인 성질을 지닌다. 그러므로, 데이터의 추상화 정도를 기준으로 계층을 구분할 수 있으며, 이를 통해서, 각 계층 안에서의 역할과 계층 간의 인터페이스를 좀 더 명확하게 정의할 수 있다. 또한, 원격으로 로봇과 사용자간의 데이터 교환이 이뤄질 때, 데이터량이 많아지게 되면,

데이터 전달에 지연이 발생할 수도 있고, 망의 트래픽을 가중시키는 결과를 가져와서, 실시간적인 응답을 보장하지 못하게 된다. 그러므로, 사용자와 로봇 사이에는 추상화된 데이터만을 교환함으로써 적은 양의 데이터로 효율적인 제어를 할 수 있는 구조를 고려해야 한다. 이 논문에서는 데이터 추상화 기법[4]을 이용한 자율 주행 이동 로봇의 소프트웨어의 구조를 제안한다.

2. 데이터 추상화

자율 주행 로봇에서 사용되는 데이터는 명령의 처리 단계나 상태의 표현 방식에 따라 구분될 수 있다. 본 논문에서는 이러한 데이터의 종류를 수치형 데이터, 명제형 데이터, 사실형 데이터로 나누었다. 수치형 데이터는 주로 하드웨어와 직접적으로 연관이 되는 태스크에서 지속적으로 변화하는 시스템의 상태를 확인하기 위해 받아들이는 센서 값이나 실제 장치에 부여하는 명령 코드 등의 수치적인 값을 말한다. 명제형 데이터는 센서의 상태 판단이나 동작의 시작 또는 중지를 위해 사용되는 참/거짓 형태의 값을 의미한다. 사실형 데이터는 시스템의 상태나 동작의 목표를 나타낸다. 이러한 구분은 추상화 개념에 의해 설명된다.

상태 데이터에 대해, 초음파 센서로부터 직접적으로 받게 되는 수치적인 값은 수치형 데이터를 의미한다. 그렇지만, 일반적으로는 이 모든 수치가 의미를 가진 않는다. 초음파 센서의 상태를 판별하기 위해서는 기준치를 정하고 그것을 넘어서느냐 그렇지 않느냐, 즉 ON 인지, OFF 인지가 중요하다. 이것은 좀 더 추상화된 의미를 지니며 바로 명제형 데이터이다. 그러나, ON/OFF 가 모든 것을 의미하지 않는다. 장애물 발견 상태는 여러 개의 초음파 센서 중 하나라도 검출되었을 경우를 의미하게 된다. 또한 초음파 센서의 검출된 위치에 따라서 장애물의 위치 상태도 판별할 수 있다. 이러한 의미에서 사용되는 것이 사실형 데이터이다. 명령의 경우에도 마찬가지로이다. "A 지점으로 이동"이라는 명령은 추상화된 명령이다. 이를 구체적으로 분석하기 위해서 현재 위치에서 A 지점으로 이동하기 위한 위치를 계산하고 "θ 각도 회전, R 거리 이동"이라는 명령으로 구체화 될 수 있다. 이 자체의 명령도 추상적이며, 이것은 다시 "+ 방향으로 n 번 턴, v 속도로 s 초 전진" 과 같은 더 구체화된 명령을 내릴 수 있게 된다.

3. 시스템의 구조 및 설계

본 논문에서는 데이터 추상화 기준에 따라 자율 주행 로봇의 소프트웨어 구조를 세 개의 계층으로 나누었다. 세 계층은 가장 하위 단계로부터 실행 계층, 제어 계층, 추론 계층으로 구성되며, 그림 1은 제한하는 시스템의 계층 구조를 나타낸다. 실행 계층은 수치형 데이터를, 제어 계층은 명제형 데이터를, 추론 계층은 사실형 데이터를 처리하게 된다. 각 계층은 내부의 처리 데이터 형태에 따라 데이터를 처리하고, 상위 단계로 데이터를 전달할 때는 추상화된 데이터로 변환하고, 하위 단계로 데이터를 전달할 때는 구체화된 동작 명령으로 전달한다. 그러므로, 각 계층은 다시, 하위 단계로부터 추상화된 상태 데이터를 받아들이는 부분과 상위 단계로 구체화된 동작 명령으로 전달하는 부분으로 나뉜다. 이를 통해서 각 계층 간의 독립성이 보장된다. 계층 별로 사용되는 데이터 종류를 통해 개발 분야의 분리가 되고 독립적인 개발 진행이 가능하다. 또한, 데이터의 추상화에 따른 데이터의 사용 정도를 고려할 수 있다. 추상화 정도가 낮을수록 데이터를 사용하는 빈도가 잦아진다. 예를 들어, 실행 계층에서 센서값의 처리는 현재 상태를 알아내기 위해 계속적으로 이뤄져야 한다. 이에 대해, 처리하는 태스크의 주기가 빨라진다. 주기가 빠른 태스크에 대해 높은 우선순위를 부여해서, 실시간성을 향상시킬 수 있다. 실행 계층의 경우, 실시간성이 중요하기 때문에, 상위의 계층보다 높은 우선 순위 태스크들로 구성된다. 설계 과정에서 각 계층에 대해 DFD를 작성했으며, 이를 통해, 각 계층 내부의 데이터의 흐름을 지정할 수 있다.

3.1 실행 계층

실행 계층은 Receiver와 Executor로 나뉜다.

Receiver는 로봇으로부터 센서값을 주기적으로 받아, 명제형 데이터로 값을 변환해서 제어 계층으로 전달한다. 또한, Executor에서의 현재 동작 완료를 확인해서 제어 계층으로 넘겨준다. 응급 사항 발생 시, 예를 들어, 초음파 센서 값으로 장애물이 발견되었을 때, 즉시 Executor의 동작을 멈추게 해서 즉각적인 응급 조치를 할 수 있게 된다. Executor에서는 제어 계층으로부터 명령을 받아 로봇의 회전, 전진과 같은 기본적인 동작만을 통해 실제로 로봇을 구동시킨다. Receiver로부터의 센서값을 실시간적으로 확인함으로써 정확한 동작을 수행하며, 동작의 완료를 판단할 수 있게 되어서 Receiver로 그 결과를 넘겨준다. 그림 2는 실행 계층에 대해 DFD를 작성한 것이다. 정보패킷으로부터 각 센서값을 분류하며, 각도나 위치 정보는 계층 내에서의 실제 동작을 위해 처리되고, 초음파 센서값을 통해 장애물 감지가 되면, 동작을 일시 중지하도록 설계되었다. 전원 상태와 장애물 발견 정보는 명제형 데이터로 변환되어서 제어 계층으로 전달된다.

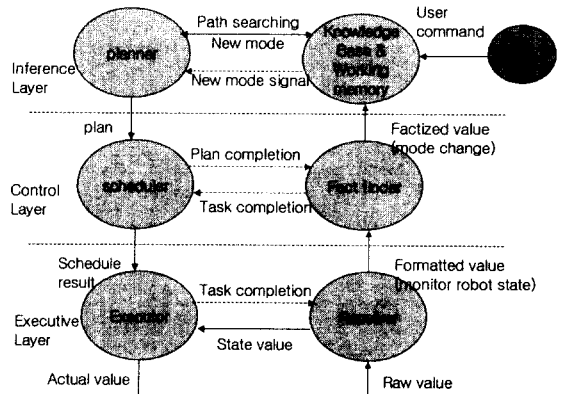


그림 1. 데이터 추상화 기법을 이용한 계층 구조

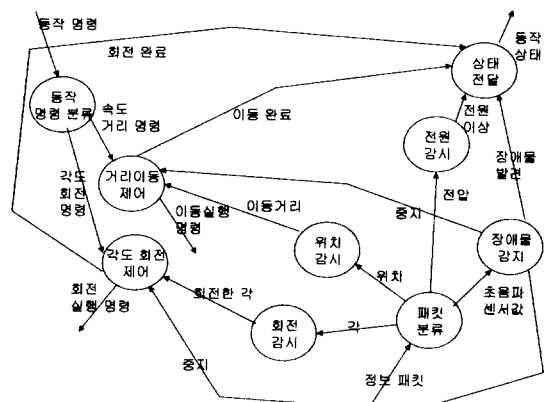


그림 2. 실행 계층의 DFD

3.2 제어 계층

제어 계층은 Fact Finder와 Scheduler로 나뉜다. Fact Finder는 실행 계층으로부터 명제형 데이터를 받아 현재의 동작 모드(주행, 회피 등) 전환이 필요한지 판단하며,

사실형 데이터로 변환해서 추론 계층으로 넘긴다. 또한, 실행 계층에서의 동작 완료 상태를 scheduler에게 넘겨준다. Scheduler는 추론 계층에서 목표에 대한 계획을 받아 그것에 대한 동작을 정렬하고 선택해서 실행 계층으로 명령을 내린다. 다음 동작의 선택은 Fact Finder로부터 전달되는 동작 완료 신호를 통해 이뤄진다.

그림 3은 제어 계층의 DFD를 작성한 것이다. 데이터 흐름을 확인하는 예를 하나 들면, 동작 상태 분류를 통해 장애물 감지 상태 정보를 모드 판단에서 확인하게 되고, 감지가 되었을 경우, 회피 모드 전환을 요구하게 된다. 추론 계층으로부터 회피 모드에 대한 계획이 전달되면, 계획 확인을 통해 동작을 선택해서 실행 계층으로 전달한다.

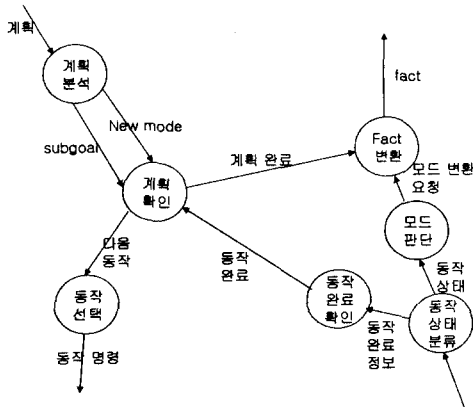


그림 3. 제어 계층의 DFD

3.3 추론 계층

추론 계층은 Working memory와 Planner로 구성된다. Working memory는 제어 계층으로부터의 모드 전환 요청이나 계획 완료 등의 사실형 데이터를 저장한다. 또, 시스템이 동작하는 환경에 대한 목표 지점 정보를 가지고 있어서 경로 결정 시에 사용된다. Planner는 Working Memory로부터 동작 모드 정보를 가져와서 그것에 따른 계획을 지시한다. 이것의 주기능은 설정된 목표 지점에 대해서 그 사이에 거쳐야 할 부 목표 지점을 설정하는 것이다. 모드 전환 요구가 들어올 때마다 재계획을 세워서 제어 단 계로 전달한다. 그림 4는 추론 단계에 대한 DFD이다. 사용자로부터 명령을 받아 목표 지점이 설정되면, 경로 정보를 이용해 부 목표 지점들이 선택되며, 추론 엔진에서는 그 지점을 순차적으로 선택한다. 제어 계층으로부터 부 목표 지점까지의 계획이 완료되었다는 정보를 받으면, 다음 부 목표 지점에 대한 계획을 제어 계층으로 전달한다.

4. 결론 및 향후 과제

본 논문에서 이동 로봇의 자율 주행을 위해 데이터 추상화 정도에 따라 계층화된 소프트웨어 구조를 제안하였다. 데이터 처리 단계에 따라, 실행, 제어 및 추론 계층으로 분류하였으며, 각 계층은 독립적인 데

이터 처리와 제어를 수행한다. 명확한 계층 구분을 통해 개발 과정의 독립성을 보장하였으며, 실행 계층에서의 응급 명령 수행을 통해 자율 주행 로봇에서의 필수적인 실시간 처리를 보장해 주었다. 제시한 구조와 설계를 이용해서 구현이 진행중이다. VxWorks[6]를 기반으로 실행 계층에는 C언어를 이용해서 개발하고, 제어 계층은 JAVA를 이용하며 두 계층 사이의 인터페이스를 위해서 JNI를 구축한다. 추론 계층은 JESS(Java Expert Shell)[7]를 이용해서 이미 CLIPS[8]에서 입증된 추론 엔진의 구성을 JAVA로 접근한다.

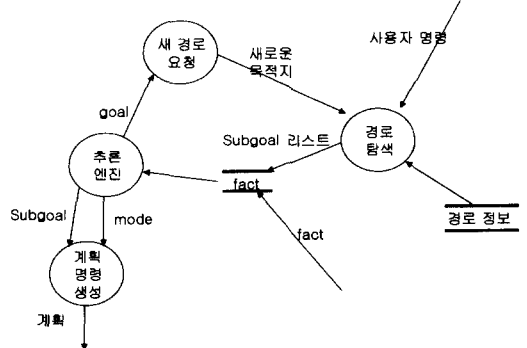


그림 4. 추론 계층의 DFD

5. 참고 문헌

- [1] M. Morin, S. Nadjm-Tehrani, P. Osterling, and E. Sandewall, "Real-Time Hierarchical Control," *IEEE Software*, Volume: 9 Issue: 5, Sept. 1992, pp51-57.
- [2] J. M. Park, I. Song, Y. J. Cho, and S. R. Oh, "A Hybrid Control Architecture Using a Reactive Sequencing Strategy for Mobile Robot Navigation," *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on Volume: 3*, 1999.
- [3] Michael L. Nelson, "A Design Pattern for Autonomous Vehicle Software Control Architectures," *Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twenty-Third Annual International*, 1999, pp172-177.
- [4] 이명진, "계층적 데이터 추상화 기법을 이용한 웹 기반 실시간 원격 제어 시스템 구조", 석사학위논문, 경북대학교 전자공학과, 1998.
- [5] R. J. Firby, "An Investigation into Reactive Planning in Complex Domains," *Proc. Of Sixth National Conf. On Artificial Intelligence(AAI)*, July, 1987.
- [6] WindRiver, <http://www.windriver.com>
- [7] the Java Expert System Shell, <http://herzberg.ca.sandia.gov/jess/>
- [8] CLIPS, <http://www.ghg.net/clips/CLIPS.html>