

자율로봇을 위한 reactive/deliberative hybrid 제어 구조 개발을 위한 3차원 시뮬레이터

윤도영**, 김영철**, 오상록*, 박귀태*, 조웅렬*, 서일홍*, 김광배*
 *고려대학교 전기공학과, † 한국과학기술연구원(KIST) 지능제어연구센터,
 **연세대학교 컴퓨터과학과, ‡ 한양대학교 전자공학과

3-D simulator for development of reactive/deliberative hybrid autonomous robot architecture

Do-Young Yoon**, Young-Chul Kim**, Sang-Rok Oh†, Gwi-Tae Park*,
 Woong-Ryul Cho†, Il-Hong Suh‡, Kwang-Bae Kim†

*Dept. of Electrical Engineering, Korea Univ., † Intelligent System Control Research Center, Korea Institute of Science and Technology(KIST), **Dept. of Computer Science, Yonsei Univ.,

‡ Dept. of Electronic Engineering, Hanyang Univ.

Abstract - In the case of designing autonomous robot architecture using deliberative and reactive control methods, we can use mixed hybrid form as well as purely reactive scheme or purely deliberative scheme respectively according to its own goal and environment within the robot operates. It needs time and endeavors to design robot control architecture in either case above. In our research, we implemented a 3-dimensional robot simulator in order to help designing reactive/deliberative autonomous robot control architecture by offering methods which is capable of selecting design parameters and confirming its performances. It can be used, of course, to design purely reactive or purely deliberative architecture. The architecture and performance of simulator is shown and a sample hybrid robot architecture designed with the simulator is introduced in this article.

으로 설계하여 자유롭게 탐색할 수 있도록 하였고, 쉽게 수행결과를 볼 수 있는 그래픽 환경도 제공하였다.

2. 본 론

2.1 시뮬레이터 Platform

3-D 시뮬레이터의 platform은 아래의 그림 1과 같이 크게 deliberative 모듈과 여기서 사용하는 모델링 정보 제공 모듈, reactive 모듈과 여기서 사용하는 센서 정보 제공 모듈, deliberative 모듈과 reactive 모듈사이의 interface 모듈의 세 부분으로 구성하고 전체적인 시스템의 동작과 그 때의 각 모듈의 동작 모습을 볼 수 있는 그래픽 display 모듈을 첨가하였다. 다음 절에 각각의 모듈에 대하여 기술하였다.

1. 서 론

Reactive control의 지역적인 시야에서 비롯되는 문제점들과 전체적인 입장에서 임무 수행 측면에서의 약점과 전통적인 AI에서의 deliberative planning방법의 느린 반응속도와 외부환경의 모델에 의존하는 약점을 상호 보완하는 구조로서 reactive/deliberative hybrid 구조를 설계할 필요성이 자주 대두된다^[1]. Reactive (behavior-based) algorithm에서는 sensing과 plan, action의 기본 단위로 sensory-motor process를 구성하고, 이 sensory-motor process는 분산되어 있으면서 sensing 정보에 즉각적인 각각의 반응을 나타낸다. 반면 deliberative 방식에서는 sense-plan-act의 단계에 따라 순차적인 process를 통해 action을 수행한다. 두 방식이 사용하는 정보 측면에서 보면 reactive 방법에서는 외부의 환경을 need-to-know(action oriented)^[2] 방식으로 받아들이며 사용하는 반면, deliberative 방식에서는 sensor로부터의 입력을 모델링을 통해 일단 제어기에서 사용할 수 있는 구조로 만들어 이를 모든 planning의 판단 근거로 이용한다. 두 방식은 processing 과정과, 사용하는 정보의 취득과 처리 형태가 다르기 때문에 이들 각각에 맞는 정보 처리와 저장 구조와 control 방식이 필요하고, 아울러 이들을 상호 연결할 수 있는 인터페이스 구조가 필요하다. 본 연구에서 구현한 시뮬레이터에서는 위의 각각의 방식을 모두 포함하고, 두 방식 사이의 통신을 통한 인터페이스 구조도 구현하여 쉽게 hybrid 구조를 구현할 수 있게 하였다. 각각의 설계 요소는 모듈 방식

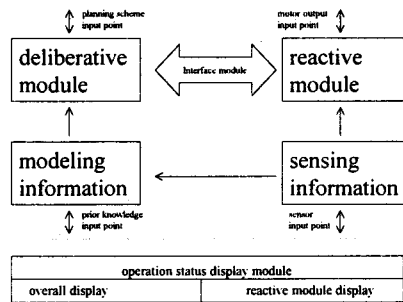


그림 1 Simulator Platform

2.1.1 Reactive 모듈

여러 개의 필요한 reactive 모듈을 구성하는 부분이다. Motor schema-based 구조의 예를 든다면 각 behavior를 담당하는 schema들이 개별적으로 구성되어 나열되어 있는 곳이다. 각 schema들은 고유의 need-to-know 전략에 따른 센서의 입력 부분과 이에 따른 실제의 각각의 시스템 고유의 motor interface로 이루어져 있다. 매 action의 단계마다 각 schema들 중의 일부가 개체화되어 활성화된다. Deliberative 모듈과의 interface 모듈에 의해 동시에 deliberative 모듈과도 연결되어 있어 필요 시에는 deliberative 모듈의 통제를 받거나 참조하여 장기적이고, global한 임무를 수행할 수도 있다. 필요한 reactive 모듈(motor schema)을 탐색하는 구조로 되어 있다.

2.1.2 Deliberative 모듈

장기적이고 global한 계획을 세우고 Reactive 모듈이 문제점에 봉착하는 경우에 해결책을 제시하는 역할을 하는 곳이다. 주로 이용하는 정보는 사용자로부터 사전에 입력되는 prior knowledge와 action을 수행하면서 얻는 sensor 입력이 information provider 모듈에 의해 가공된 추상적인 형태의 정보이다. Motor schema-based navigation^[3]의 예를 든다면 deliberative 모듈의 중요한 임무의 하나는 motor schema part의 문제점을 풀어주는 것이다. 예를 들어 motor schema part가 local minima에 빠져 있을 때 이를 판단하고, 새로운 path를 제시하는 등의 역할을 수행한다. 또한 최상위 수준의 임무를 부여받아 이를 해석하여 global path를 형성하고, 이를 reactive 모듈에 전달하여 실제의 path를 구현한다.

2.1.3 Sensing Information 모듈

외부 환경으로부터의 sensing을 담당하는 부분으로 필요한 각 센서정보와 reactive 모듈, modeling 정보 모듈과의 interface를 침착하는 부분이다.

2.1.4 Modeling Information 모듈

사용자로부터의 사전정보와 동작중의 sensor정보로부터 들어온 정보를 가지고 deliberative 모듈이 필요로 하는 형태로 모델링을 하여 그 결과를 deliberative 모듈에 제공하는 부분이다. 이 모듈의 역할은 정보의 저장을 필요로 한다. Long-term memory와 short-term memory의 두 가지를 모두 운용한다. Motor schema-based navigation의 예에서는 주행에 필요한 지형정보를 가지는 map을 작성하는 역할을 수행하는데 Long-term memory는 미리 입력되는 전역적(global)이고, 주로 정적인 data를 기반으로 운용되고, short-term memory는 시스템이 동작하면서 획득하는 지역적이고(local), 주로 동적인 data를 기반으로 운용된다.

2.1.5 Interface 모듈

Interface 모듈의 주임무는 reactive 모듈과 deliberative 모듈의 연결이다. 이 연결은 reactive 모듈에 대한 deliberative 모듈의 역할의 성격을 기준으로 크게 네 가지로 구분할 수 있다. 첫 번째 interface는 deliberation을 reactive 요소의 파라미터 등을 재구성하는 역할을 하도록 하는 것이다. 두 번째 interface 방식은 deliberation을 reactive 모듈에 대한 adviser의 역할을 하게 한다^[4]. 이 방식에서는 action의 최종선택은 deliberation의 결과를 참조하여 reactive 모듈에 의해서 이루어진다. 세 번째 방식은 deliberation이 적용성을 향상시키는 역할을 하는 것이다^[5]. 이 방식에서는 planning은 주로 reactive 모듈의 문제점을 해결하여 주는 역할을 한다. 마지막으로 네 번째 방식은 deliberation이 reactive action의 수행을 조건이 성숙될 때까지 기다리게 하여 수행하므로써 전체적인 임무에 보다 적합한 행위를 할 수 있도록 하는 것이다. 시뮬레이터에서는 경우에 따라 자기 다른 방식의 interface를 단독으로, 또는 혼합하여 구성하면 된다.

2.2 Control Architecture 설계

위에서 제시한 control architecture의 platform에 따라 reactive control에 중점을 두고 deliberation의 기능을 보완한 3차원에서 수행하는 자율 로봇의 제어 구조를 설계하였다. 그림 2에 전체 control architecture의 구조를 나타내었다. Deliberation 모듈에는 일반적인 임무를 수행하는 Mission Planner와 특별히 많은 부분의 임무가 주어지는 자율주행의 임무를 위하여 Navigation Planner를 두었다. Deliberation은 Information Provider에서 모델링에 의하여 얻어진 정보를 바탕으로 Planning을 수행한다. Information Provider의 주된 역할은 주행에 필요한 map을 초기에 작성하고, 동작 중에 update하는 것이

다.

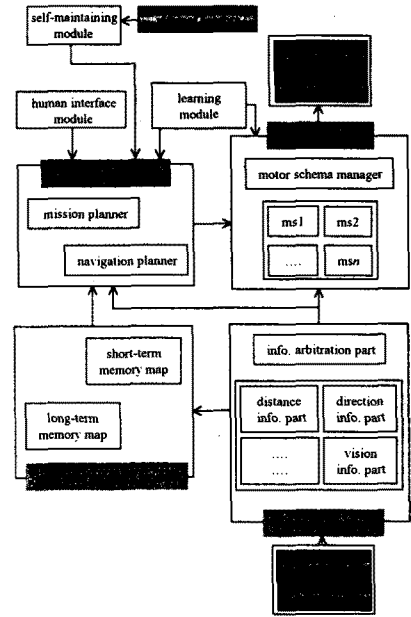


그림 2. 설계된 제어 구조의 전체 구성

Reactive 모듈 부분은 schema-based reactive control 방식을 선택하여 3차원 자율주행에 필요한 여러 가지 schema를 구현하여 구성하였다. 주요 schema를 다음과 같이 구현하였다.

· Move-ahead schema :

$$V_{ahead} = G_{ahead} \frac{\begin{matrix} \rightarrow & \rightarrow \\ \text{Goal} & \text{Start} \end{matrix}}{\begin{matrix} \rightarrow & \rightarrow \\ \text{Goal} & \text{Start} \end{matrix}} \quad (1)$$

· Move-to-goal schema:

$$V_{goal} = G_{goal} \frac{\begin{matrix} \rightarrow & \rightarrow \\ \text{Goal} & \text{Position} \end{matrix}}{\begin{matrix} \rightarrow & \rightarrow \\ \text{Goal} & \text{Position} \end{matrix}} \quad (2)$$

· Avoid-obstacle schema :

$$|V| = \begin{cases} 0 & \text{for } d > S \\ \frac{S-d}{D-R} * G_{obs} & \text{for } R < d \leq S \\ \infty & \text{for } d \leq S \end{cases} \quad (3)$$

$V_{direction}$ = radially along a line from robot to center of obstacle, directed away from the obstacle.

· Stay-on-path schema

$$|V| = \begin{cases} P & \text{for } d > (W/2) \\ \left(\frac{d}{(W/2)} \right) G_{path} & \text{for } d \leq (W/2) \end{cases} \quad (4)$$

$V_{direction}$ = normal to center of the channel heading toward centerline

• Noise schema :

$$|V| = G_{noise} \quad (5)$$

$$V_{direction} = random\ direction$$

• Move-Robot schema :

$$\sum Schema \quad (6)$$

이들 각각의 schema는 Move-Robot schema에 의해서 cooperative 방법에 따라 벡터 합으로 coordination되어 최종적인 motor 출력을 내게 된다. 이 관계를 아래 그림 3에 나타내었다.

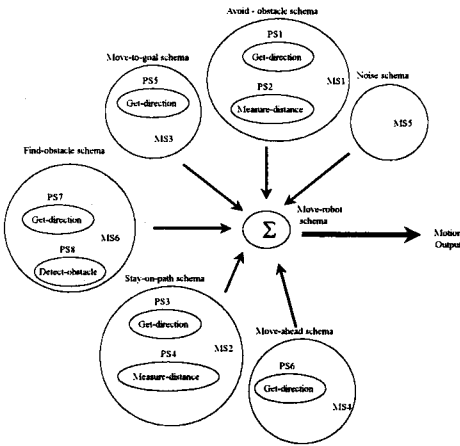


그림 3 Schema Coordination

Reactive 모듈에 정보를 제공하는 perception 모듈은 sensing을 담당하는 부분으로 vision system을 중심으로 ultrasonic sensor, encoder, gyro 등으로 구현하였다.

위와 같이 설계한 hybrid architecture는 시스템의 고유 dynamics와 완전히 독립적인 것이다. 즉, Architecture를 실제의 시스템에 적용하고자 할 때는 reactive 모듈의 motor interface만을 시스템별로 그 dynamics에 맞추어 구현하여 첨가하게 하였다. 본 architecture는 모든 종류의 임무의 수행이 가능하게 한 architecture이다. 다만 navigation의 임무가 로봇시스템에 있어 중요한 부분을 차지하므로 특별히 고려하여 설계하였다. 또한 모든 종류의 임무를 각각의 고유한 motor schema로 구성하여 첨가할 수 있게 하여, 단순한 한 가지 임무만을 수행할 수 있는 상태에서 시작하여 필요할 때 마다 원하는 기능을 첨가할 수 있는 형태로 하여 설계 및 유지 보수가 용이하다. 시뮬레이션 결과 deliberation 모듈이 navigation planning을 통하여 reactive 모듈이 볼 수 없는 전역적인 path를 설정하고 이를 reactive 모듈이 모델링 없이 반사적인 action으로 복잡한 도시 형태의 장애물을 피하여 3차원 주행하는 것을 확인하였다. 이를 통하여 본 연구에서 설계한 reactive control 기반의 hybrid robot control architecture의 navigation 임무에서의 유효성을 확인하였다. 또한 Reactive system의 각 motor schema들의 목표에 따른 역할을 확인하고, 각 schema의 이득 값의 조절과 메모리의 사용을 통하여 최적의 경로를 따라 주행하는 algorithm의 구현 가능성을 확인하였다. Deliberative 모듈과의 협력을 통하여 global mission을 완수할 수 있음을 보았다. 그림 4에 시뮬레이션의 모습을 보이고 있다.

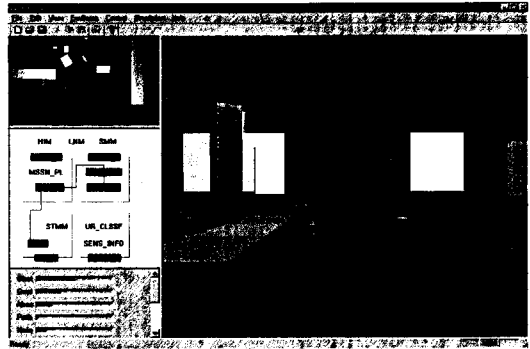


그림 4 시뮬레이션

3. 결론

본 연구에서 구현한 시뮬레이터는 reactive control과 전통적인 AI의 deliberation 방식을 혼합한 hybrid autonomous robot control architecture를 설계하는데 도움이 되는 platform의 환경이다. 이를 이용하여 제어 구조의 개발 시에 각각의 모듈을 독립적으로 설계하여 쉽게 병합하고, 결과를 확인할 수 있게 하였다.

또한 구현한 시뮬레이터 platform을 이용하여 hybrid architecture를 설계하여 시뮬레이션을 통하여 그 유효성을 검증하였다. 앞으로 시뮬레이션의 sensing 모듈의 현실적인 파라미터의 반영을 통하여 현실에 충실한 시뮬레이터를 구현할 계획이며, 친근한 user interface와 설계 변경의 유연성(flexibility)을 보장하여 보다 유용한 시뮬레이터가 되도록 할 것이다.

(참고 문헌)

- [1] Paolo Pifjanian, "An Overview of System Architectures for Action Selection in Mobile Robotics", Laboratory of Image Analysis, Aalborg Univ. 1997.
- [2] Arkin R.C. Behavior Based Robotics, MIT Press, 1998, pp.237-303
- [3] Arkin, R.C. "Behavior-Based Robot Navigation for Extended Domains," Adaptive Behavior, Vol. 1, No. 2, pp.201-225.
- [4] Gat, E. "Reliable Goal Directed Reactive Control of Autonomous Mobile Robots," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg.
- [5] Lyons, D., and Hendriks, A. "Planning for Reactive Robot Behavior," Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, pp2675-80.