

이동로봇 자율주행을 위한 행위모듈의 실행순서 조정기법

Sequencing Strategy for Autonomous Mobile Robots in Real Environments

송인섭, 박정민, 오상록, 조영조, 박귀태
(Insup Song, Jung-min Park, Sang-rok Oh, Young-jo Cho, and Gwi-tae Park)

Abstract : Autonomous mobile robots are required to achieve multiple goals while responding quickly to the dynamic environments. An appropriate robot control architecture, which clearly and systematically defines the relationship among the inputs, the processing functions and the outputs, thus needs to be embedded in the robot controller. This paper proposes a kind of hybrid control architecture which combines the key features of the two well-known robot control architectures: hierarchical and behavioral-based. The overall control architecture consists of three layers, i.e. the highest planner, the middle plan executor, and the lowest monitor and behavior-based controller. In the planned situation, only one behavior module is chosen by the logical coordinator in the plan executor according to the way point bin. In the exceptional situation, the central controller in the plan executor issues an additional control command to reach the planned way point. Several simulations and experiments with autonomous mobile robot show that the proposed architecture enables the robot controller to achieve the multiple sequential goals even in dynamic and uncertain environments.

Keywords : mobile robots, control architecture, behavior, hybrid control

I. 서론

기술의 발달로 인해 로봇은 보다 정밀한 동작을 수행할 수 있게 되었고 성능이 향상됨에 따라 인간이 생활하는 실제 환경에서 인간의 생활을 돕는 로봇을 원하게 되었다. 고정된 위치에서 정해진 작업만을 수행할 수 있는 기존의 산업용 로봇과는 달리 자유롭게 움직일 수 있는 장점이 있는 자율이동로봇에 대한 여러 가지 연구가 행해지고 있다.

인간이 생활하는 환경에서 다중 작업을 수행하는 로봇은 동적이며 불확실한 환경에 효과적으로 대응하면서 빠른 응답 특성을 가져야 하기 때문에 기존의 산업용 로봇에 사용되던 방법과는 다른 로봇의 제어구조에 대한 연구가 필요하게 된다. 로봇의 입력정보처리, 계획, 출력의 상호관계를 명확히 정의하고, 체계적으로 관리하기 위한 설계방법, 즉 제어구조에 대한 연구가 필요하다.

제어구조에 관한 초기 연구는 미래에 실행될 계획의 생성에 중점을 두었던 계층적인 구조를 사용하였으나 이 방법은 동적인 환경에 빠르게 응답하지 못하였다. 이를 보완하기 위하여 제안된 행위기반제어구조[1]-[6]는 동적인 환경에서 빠른 응답 특성을 보이나 다중 작업 수행에 문제점이 있으며 로봇이 교착 상태(deadlock)에 빠지게 되면 환경이 변하지 않는 한 로봇은 교착 상태에서 빠져나올 수 없다.

본 연구의 목적은 동적이며 불확실한 환경에서 다중 작업 목표를 수행할 수 있는 자율이동로봇의 명령을 생

성하는 로봇의 제어구조를 설계하는 것이다. 불확실한 환경에서 동작하는 자율 이동로봇의 제어 구조는 신뢰성과 강성이 있어야 하며 변화하는 환경에 빠르게 대응해야 하고 다중 작업 목표를 수행할 수 있어야 한다.

이와 같은 요구사항을 만족시키는 제어구조의 설계 시 필수적으로 고려해야 하는 사항은 변경이 용이하게 하기 위한 융통성, 작은 작업단위로 분할이 가능한 모듈성, 새로운 작업 요소의 추가가 가능한 확장성, 제한된 시간 내에 제어출력을 만들 수 있도록 하는 실시간성 등이 있다[5][7].

본 논문에서는 이러한 네 가지의 요구사항을 만족시키는 동시에 동적이며 불확실한 환경에서 다중 작업목표를 수행할 수 있는 자율 이동로봇의 제어구조를 제안하고, 이 제어구조를 사용하여 모의실험과 건물의 복도에서 실제로 실험함으로써 타당성을 보이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 로봇의 제어구조에 관한 기존의 연구들에 대해 살펴보고 각 제어구조의 특징 및 장단점을 서술한다. 3장에서는 제안하는 제어구조에 대한 특징과 본 논문에서 개선된 사항을 예와 함께 설명한다. 4장에서는 본 논문의 실험에서 사용되는 자율이동로봇에 대하여 설명하고 모의실험 및 실험 결과를 서술한다. 5장에서는 본 연구의 결론 및 앞으로의 연구과제에 대해 언급한다.

II. 자율 이동 로봇의 제어구조

1. 계층적 제어구조(hierarchical control architecture)

계층적인 제어구조는 정보를 처리하는 기능에 따라서 각 모듈들을 나누며 센서로 입력된 정보는 각 모듈들에서 순차적으로 처리되어 구동부로 출력된다. 크게 입력, 처리, 출력의 세 부분으로 나뉜다(그림 1참조). 이 제어

접수일자 : 1998. 6. 19., 수정완료 : 1998. 12. 22.

송인섭, 박정민, 오상록, 조영조 : 한국과학기술연구원 지능제어 연구센터

박귀태 : 고려대학교 전기공학과

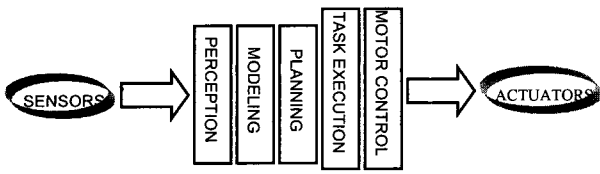


그림 1. 자율이동로봇의 계층적 제어구조의 기능별 모듈 분류.
Fig. 1. Functional module decomposition of hierarchical control architecture.

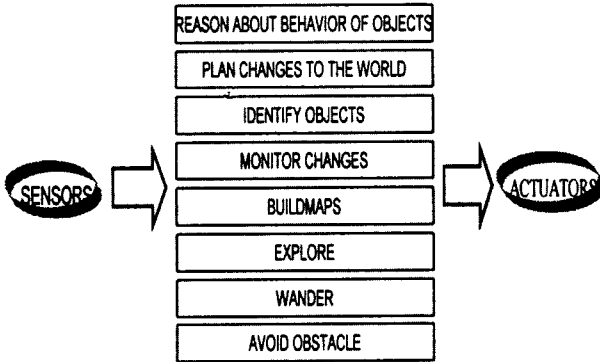


그림 2. 자율이동로봇의 행위 기반 제어구조.
Fig. 2. Behavior-based control architecture.

구조에서는 정보의 추상화된 수준에 따라서 기능별로 모듈을 나눈다. 분할된 각 부분은 서로 다른 정보 내용을 처리한다. 이러한 구조는 중앙에 전체 모듈을 통제하는 제어부가 있다.

계층적인 제어구조는 정보를 순차적으로 처리하기 때문에 자율이동 로봇이 실시간으로 빠르게 대응할 수 없다. 센서 정보 입력 및 구동기 출력 등은 빠르게 동작하지만 추론, 계획 등으로 구성된 처리부는 시간이 많이 걸리기 때문에 입력에서 출력사이에 피할 수 없는 시간 지연이 생긴다. 따라서 실시간 환경에서 예측하지 못한 상황에 대해 신속하게 대응하지 못한다. 정보가 순차적으로 처리됨으로 인한 또 하나의 단점으로는 한 모듈에서 생긴 오류가 정보의 흐름에 따라서 점점 더 커지게 됨으로 인해 잘못된 로봇의 출력을 만들어 내게 된다. 이러한 단점들을 해결하고자 행위기반 제어구조가 연구되었다.

2. 행위기반 제어구조(behavior-based control architecture)
행위기반 제어구조는 “특정 작업 행위”(task specific behavior)를 수행하는 행위 모듈로 구성되며 기본 단위인 각 행위 모듈은 작업 수행에 필요한 입력만을 받아서 입력에 대해 계층적 제어구조에서와 같이 일반적인 추론과 계획을 하지 않고 작업 수행에 필요한 간단한 처리를 한 후 구동부로 보낸다. 즉, 각 행위 모듈들은 입력, 처리, 출력을 수행함으로써 독립적으로 동작하므로 빠른 응답 특성을 갖는다. 계층적 제어구조와는 달리 중앙에서 통제하는 제어부가 없다.

그림 2와 같이 하위의 단순한 행위들로 시작해서 상위로 갈수록 차츰 복잡한 행위 모듈들을 구성한다. 상위

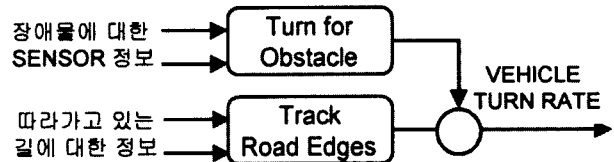
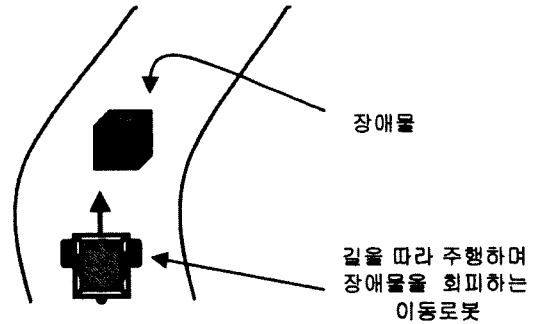


그림 3. 행위기반 제어구조의 예[8].
Fig. 3. The example of behavior based control architecture[8].

의 행위 모듈들은 하위의 행위 모듈들을 기초로 하지만 모든 모듈들은 다른 모듈들과 상관없이 독립적으로 정보를 입력받고 처리하여 출력한다. 모든 행위 모듈들은 수행해야 할 작업에 알맞은 센서 정보만을 이용하여 처리하기 때문에 여러 센서 정보를 통합하는 것은 특정한 모듈에서만 일어나게 되고 그 외의 행위 모듈들은 이러한 센서정보를 통합하지 않아도 된다.

모든 행위 모듈들이 센서 정보를 동시에 입력받아 처리하고 출력하므로 각 모듈들은 서로 상충되는 제어 출력을 만들어 낼 수 있으므로 각 모듈들의 출력을 중재(arbitration)해야만 한다. 출력 중재는 센서 통합, 명령 통합이나 명령 중재를 대신하는 것이다. 출력 중재의 한 예인 그림 3과 같은 포함 구조(subsumption architecture) [1]는 언제나 하위 행위 모듈의 출력에 상위 행위모듈의 출력이 포함(subsume)되는 구조이다.

행위기반 제어구조를 그림 3과 같은 예를 들어 설명하면 다음과 같다. 길을 따라 주행하는 동안 장애물을 안전하게 회피하는 것을 목표로 하는 자율이동로봇에서, 길을 따라 주행하는 행위 모듈과 장애물을 피하는 행위 모듈인 두 개의 모듈이 있고 각 모듈은 동시에 모듈이 담당할 작업만을 처리한다. 장애물이 나타났을 때 장애물 회피 행위 모듈의 출력이 길을 따라가는 행위모듈의 출력을 누르고 로봇 구동부로 출력되어 로봇은 장애물을 피한다. 이 중재방법은 하위 행위모듈의 의미있는 정보가 상위 행위모듈에 의해서 없어질 가능성이 있다. 이러한 단점을 보완하기 위해서 스칼라 출력이 아닌 벡터로 출력을 하는 등의 연구가 있다[5][8].

행위기반 제어구조는 확장이 용이한 장점이 있으나 설계의 융통성이나 모듈성을 제공하지 못한다. 주어진 작업을 적절한 모듈로 분할하는 구체적인 방법이 없기 때문에 좀 더 일반적이고 융통성 있는 행위간의 관계와 분할 방법 등이 필요하다. 또한 행위기반 제어구조는 전체 시스템을 관리 제어하는 모듈이 존재하지 않으므로

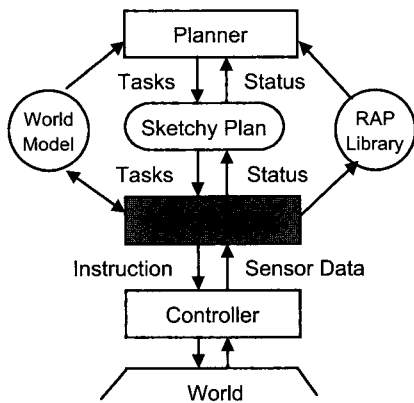


그림 4. RAP 구조의 블록도.
Fig. 4. Block diagram of RAP.

예측하기 어려운 동적인 상황에 빠르게 대처할 수 있으나 현재의 전체적인 상황을 파악하는데는 어려움이 있다. 또한 순차적으로 여러 단계를 거쳐야 하는 작업 목표를 수행함에 있어서 적절히 표현하거나 적합한 명령을 생성하지 못한다. 행위기반 제어구조의 복잡한 다중 작업 목표를 수행하기 어려운 단점으로 인하여 계층적 제어구조를 기반으로 하여 행위기반 제어구조를 결합하거나 행위기반 제어구조에 중앙 제어부를 추가하는 등 복합 제어구조에 대한 연구가 수행되고 있다.

3. 복합 제어구조(hybrid control architecture)

실시간 응답성이 우수하면서도 다중 작업 목표를 수행하기 위하여 복합 제어구조[9]-[13]가 제안되었다. 본 절에서는 복합제어구조에 대한 대표적인 예인 RAP(reactive action package)을 살펴봄에 복합 제어구조의 특징과 연구 동향에 대하여 간략히 설명한다.

RAP 제어구조는 계획기(planner), 실행기(executor), 제어기(controller)인 세 개의 층으로 구성된 제어구조로 계획기와 제어기가 존재한다는 가정 하에서 RAP 실행기를 정의하고 구현하였다. RAP 실행기는 그림 4에서 보는 바와 같이 계획기와 환경에 접하여 동작하는 제어기 사이에서 다리 역할을 담당한다. 작업을 구성함으로써 로봇의 프로그램을 기술하는 역할도 한다. RAP안의 각 방법은 간단한 행위기반 제어구조의 모듈이며 각 RAP에는 수행하고자 하는 목표와 목표 수행을 위한 방법들, 이 방법들이 수행할 상황들이 기술되어 있다. 실행기에서 제어권이 제어기로 옮겨진 후 실행기는 제어기에서 동작하는 방법들이 성공하거나 실패할 때까지 기다린다. RAP은 현재 상황에 적절한 행위모듈을 선택할 수 있는 강력한 경험적 방법(heuristic)을 가지고 있다. 즉 행위기반 제어구조처럼 여러 개의 행위 모듈이 동작하는 것이 아니라 환경 상황에 적절한 몇 개의 행위모듈만 활성화되어 자율이동로봇을 동작시킨다.

RAP을 기반으로 한 제어구조는 ATLANTIS[11], 3T[9]등이 제안되었다. 상위의 계획기, 중앙의 RAP 실행기, 하위의 제어기인 삼단계로 제어구조를 정의하고 구현하였다. RAP 실행기는 각각의 상황에 대응하여 동작하는 제어기 내부의 제어모듈들을 환경의 상황에 알맞고 작업

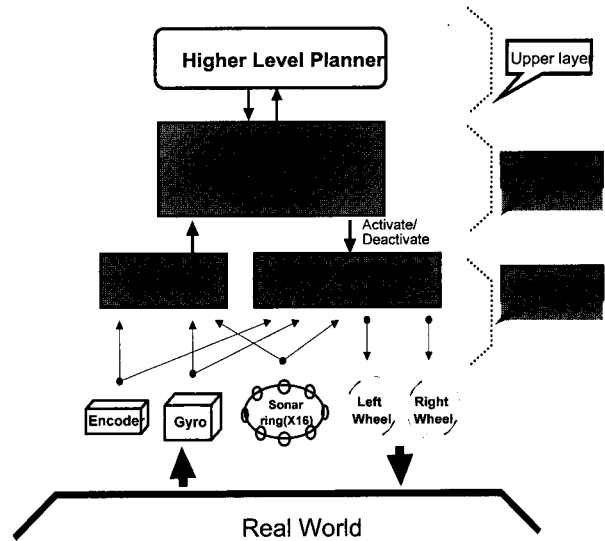


그림 5. 실행순서 조정기법을 이용한 로봇제어구조의 블록도.
Fig. 5. Block diagram of suggested robot architecture.

목표에 적합하게 선택하여 활성화시킨다. 전체 자율 이동로봇 시스템의 제어는 가장 상위의 로봇 동작을 계획하는 계획기가 담당하며 이는 기존 인공지능 분야에서 연구되었던 것들을 사용하고 있다.

III. 실행순서 조정기법을 이용한 제어구조 설계

2장에서 살펴본 것과 같이 계층적 제어구조는 중앙통제 모듈이 있으므로 전체적인 작업 목표를 관리하지만 환경변화에 빠르게 응답하지 못하는 단점이 있었다. 행위기반 제어구조는 불확실한 환경 상태와 예측하지 못한 동적인 환경 변화에 빠르게 대응할 수 있지만 전체적으로 작업목표를 관리하는 능력이 없으며 계획된 순차적 작업을 수행할 수 없다. 본 논문에서는 이런 두 제어구조 각각의 장점인 빠른 응답특성과 전체적인 작업 목표 관리 능력을 이동로봇에 적용하기 위해 일종의 복합 제어구조 형태를 가진 실행순서 조정기법(sequencing strategy)을 이용한 제어구조를 제안한다.

1. 실행순서 조정기법을 이용한 제어구조의 개요

본 제어구조는 그림 5와 같이 크게 세 개의 층으로 구성되어 있다. 하위 층(lower layer)은 환경으로부터 들어온 실시간 입력에 빠르게 대응할 수 있는 기능을 수행하는 행위기반 제어기(behavior-based controller)와 입력된 센서 정보에서 필요한 정보만을 추출하는 감시기(monitor)로 구성된다. 중간 층(middle layer)인 계획 실행기(plan executor)는 환경에서 감지된 입력 정보를 행위기반 제어기 내의 각 행위 모듈들을 보다 높은 수준으로 처리한다. 상위 층(upper layer)인 고 수준의 계획기(higher level planner)는 추상적인 형태의 정보를 미리 생성하거나 실행 중 재생성한다.

본 논문에서는 그림 5의 색이 칠해진 두개의 층 즉, 하위 층과 중간 층만을 고려하였으며, 상위 층은 향후 연구에서 확장 가능하다. 행위 모듈들은 각 모듈의 동작

에 필요한 최소한의 환경정보만을 사용하므로 환경의 특징을 추출하기에는 부족하기 때문에 환경정보 추출을 목적으로 하는 감시기는 한 가지 이상의 센서 정보와 필요한 연산을 통해 환경의 특징을 찾아낸다. 중간 층은 상황에 알맞게 하위 층의 행위 모듈의 실행순서를 조정하는 역할 및 중앙 통제 역할 등을 실행하는 계획 실행기로 구성되며 계획 실행기는 중앙 제어기(central controller), 논리 중재기(logical coordinator), 경유 점 저장소(way point bin)등으로 구성된다. 하위 층의 행위기반 제어기 내부에는 여러 개의 행위 모듈들이 존재하며, 이 행위 모듈들 중 현재의 상황과 환경에 가장 적합한 행위 모듈 하나만이 감시기에 의해서 선택된다.

즉, 상위 층은 전체적인 작업 목표 설정 및 재생성 등을 담당하며, 중간 층은 추상적인 작업 목표들과 구체적인 행위모듈 간의 연결을 담당하며, 하위 층에서는 환경정보 획득 및 불확실하고 동적인 환경에 적응할 수 있는 출력을 만든다.

2. 행위기반 제어기와 감시기

행위기반 제어기의 내부에는 여러 종류의 행위 모듈들이 존재하며 이 행위 모듈들은 특정 시점에서 단 하나의 모듈만이 각각 독립적으로 동작하며, 다른 행위 모듈의 영향을 받거나 영향을 주지 않는다. 행위기반 제어기 내부의 여러 가지 행위 모듈 중 계획 실행기에 의해서 활성화된 하나의 행위 모듈만이 환경에 직접 대응하여 구동기의 출력을 생성하여 로봇이 동작한다.

각 행위 모듈들은 행위기반 제어구조[1]-[3][5][6][14]에서 주장하는 바와 같이 특정한 작업목표만을 수행하기 위하여 필요한 최소의 센서 정보를 입력받고, 탐색이나 추론과 같은 복잡한 연산이 아니라 센서 정보를 간단히 처리하고 계산하여 구동기로 출력한다. Lee 등은[15] 동작하는 환경에서 상황에 따라서 로봇을 가장 잘 제어해주는 행위 모듈이 있다는 것을 정량적인 실험 데이터를 통하여 보여주며, 환경에 따라서 로봇을 제어하는 행위 모듈의 선택이 중요하다고 주장한다. 로봇이 주어진 환경에서 작업목표를 수행하기 위해서는 하나의 일관된 행위 모듈이 아닌 특정한 상황에 잘 적응하는 몇 가지의 행위 모듈을 구성하여, 이러한 행위 모듈들의 실행순서를 조정함으로써 특정한 상황뿐만 아니라 전체 환경에서도 잘 동작할 수 있다. 따라서 본 연구에서 제안하는 로봇의 제어구조는 상황에 적절히 대응할 수 있는 행위모듈을 행위기반 제어구조를 기초로 하여 구현해 놓은 후, 이 행위모듈들의 실행 순서를 조정하여 주어진 작업목표를 수행하도록 한다.

실행순서를 조정하기 위해서는 반드시 현재 로봇의 동작 환경 상황에 대한 정확한 인식 및 정보획득이 필요하다. 그러나 행위 모듈은 자신에게 필요한 최소량의 정보만을 획득 처리하며 내부상태 정보(internal state information)가 적기 때문에 환경의 전체적인 정보를 획득하기에는 부족하다. 그러므로 감시기를 두어서 환경에 대한 전체적인 이해와 인식을 위한 정보 획득을 담당하도록 한다. 일반적인 사무실 환경에서 얻을 수 있는 자

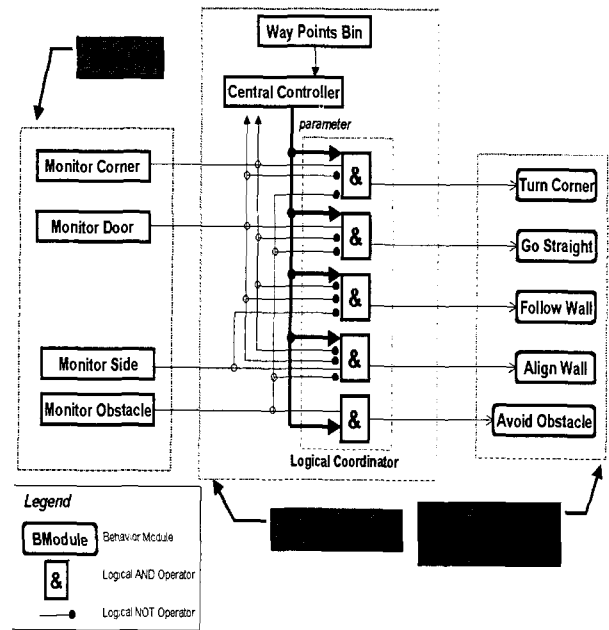


그림 6. 실행순서 조정기법의 상세도.
Fig 6. Diagram of sequencing strategy.

연 랜드마크 정보로는 모퉁이나 문 등이 있으며 사무실 환경에서 동작하는 로봇은 이런 자연 랜드마크만을 이용하여 주어진 목표를 수행할 수 있다, 감시기는 일정한 주기로 항상 동작하며 환경정보를 획득하고 환경 정보들을 계획 실행기로 보내는 역할을 한다. 감시기의 내부는 다수의 감시 모듈들이 독립적으로 동작하며 환경을 감시한다. 예를 들면, 복도를 주행하는 로봇의 감시기에는 문을 감시하는 '문 감시 모듈', 모퉁이가 있는지를 확인하는 '모퉁이 감시 모듈', 로봇의 주행 궤적과 시간을 기록하고 감시하는 모듈, 교차 상태에 빠져서 같은 동작을 반복하고 있는지를 감시하는 모듈 등이 있으며, 이런 감시 모듈은 일정한 주기 즉, 센서 정보의 갱신 주기보다 빠르게 현재 환경 상태 정보를 획득하고 수정하여 행위기반 제어기 내부의 행위 모듈을 선택하는데 기초가 되는 정보를 제공한다.

3. 실행순서 조정 기법

계획 실행기의 내부에 있는 논리 중재기(logical coordinator)는 감시기로 부터의 환경 입력을 바탕으로 현재 실행될 행위모듈을 결정한다. 계획 실행기는 논리 중재기와 중앙 제어기를 기반으로 행위 모듈의 실행순서를 결정한다.

그림 6에서 보는 바와 같이 논리 중재기는 감시기로 부터의 입력을 추론이나 복잡한 연산에 아닌 논리회로와 같은 연결방법을 통하여 행위 모듈을 선택한다. 이 논리 중재기는 간단한 논리 조합을 사용하여 단순히 행위모듈을 선택하는 것뿐만 아니라, 현재 수행 중인 동작이나 수행해야 할 경유 점 정보에 기초하여 중앙제어기가 내보내는 매개변수(parameter)를 행위 모듈에 전달한다.

예를 들어 설명하면, 그림 6에서 문이 아니고 장애물이 없으며 모퉁이가 아니면서 벽과 일정한 거리에 있다

면, 이 조건에 의해 벽을 따라 주행하는 행위모듈(follow wall)이 선택된다. 각 상황에 적절한 행위모듈이 수행되어 실시간을 보장하지만 순서가 정해진 계획을 수행하려면 논리 중재기만으로는 부족하다. 따라서 중앙제어기와 경유 점 저장소를 추가하여 계획된 작업 목표를 수행할 수 있도록 한다. 일반적인 사무실 환경의 복도를 주행하는 자율 이동로봇에 대하여 설명하면, 경유점 저장소에 주행 목표 지점까지의 경유점의 정보들이 들어가며 실시간으로 변화하는 각 상황은 논리 중재기가 담당하여 처리하고 중앙 제어기에서는 경유점과 주행 목표지점까지의 도달을 확인하고 교착 상태를 인식하여 새로운 경로를 생성하고 수행을 명령하거나, 작업이 수행 불가능한 상태임을 판단한다. 즉, 중앙 제어기는 행위 모듈의 시작과 끝의 상태를 파악하여 다음 명령을 수행할 수 있도록 하며, 행위 모듈로는 처리할 수 없는 예외 상황에 대처하도록 한다. 이 밖의 상황은 논리 중재기가 행위모듈을 선택한다.

자율 이동 로봇의 정확한 환경좌표(world coordinate)는 모터의 엔코더 값을 기초로 추측할 수 있다. 하지만 바퀴의 미끄러짐 등으로 인해서 생기는 위치 오차는 로봇이 주행할수록 커지게 된다. 여러 가지 요인에 의해서 로봇의 정확한 위치를 얻기는 어려우므로 본 연구에서는 로봇의 절대적인 위치 좌표를 사용하지 않고 환경의 특징인 자연 랜드마크(natural landmark)를 인식하여 주행한다. 일반적인 사무실 환경에서 얻을 수 있는 자연 랜드마크 정보인 모퉁이 및 문에 대한 정보는 초음파 센서의 패턴을 사용하여 추출한다. 경유점 저장소에는 자연 랜드마크에 대한 정보가 들어 있다.

4. 실행 순서 조정기법을 이용한 제어구조의 특징 및 장점

본 제어구조의 특징은 미리 실험을 통하여 구현해 놓은 단위 행위모듈을 이용하여 주어진 작업을 기술하며, 상황에 따라 하나의 행위모듈만이 동작하도록 하기 때문에 기존의 행위기반 제어구조만을 이용한 로봇과는 달리 행위모듈 간의 중재가 명확히 기술됨으로 자율 이동로봇의 움직임을 정확히 예측할 수 있다. 계획 실행기 내부에는 논리 중재기, 중앙 제어기, 경유 점 저장소가 있으며, 새로운 목표지점의 시작, 현재 목표지점의 종료, 교착 상태 등 행위모듈로는 처리할 수 없는 예외 상황이 발생하면 중앙제어기가 동작하여 새로운 명령을 생성하는 등의 조치를 취한다.

행위 모듈의 실행 순서 조정기법을 이용한 제어구조는 필요에 따라서 탐색(search), 추론, 지식 베이스 연산(knowledge-based system) 등 좀 더 복잡한 연산을 수행하는 모듈을 계획 실행기의 중앙 제어기 내부에 추가하기에 용이한 구조이다. 새로운 작업을 추가하려면 감시기에 해당하는 모듈을 추가하거나, 행위 모듈을 추가하고 논리 중재기에 추가하는 방법으로 간단히 수정하면 된다. 본 제어구조는 모듈성이 뛰어나고 확장성이 용이한 장점이 있으며 일반적인 상황과 예외적인 상황을 나누어 다른 방법으로 대처함으로써 동적인 환경과 불확실한 환경에서도 실시간성을 보장하도록 한다.

IV. 실험

1. 자율 이동로봇의 시스템 구조

본 연구의 실험은 우리 실험실에서 제작한 HULK 로봇(HUMAN Like Robot)을[5] 사용하여 수행하였는데 이 로봇은 VME rack을 실장하고 있으며, Force CPU40보드[16]를 사용하였다. 서보 모터의 위치제어는 DSP TMS320C30을 이용한 동작 제어기를 사용하였다. 16개의 초음파센서로 구성된 Normadic의 Sensus 200을 사용하여 실시간 주행용 환경정보를 얻는다. 본 로봇의 제어 구조 소프트웨어는 실시간 운영체제(real-time OS)인 VxWorks를 사용하였고, 개발에 이용한 언어는 C++이며 객체지향적 구조로 설계되었다.

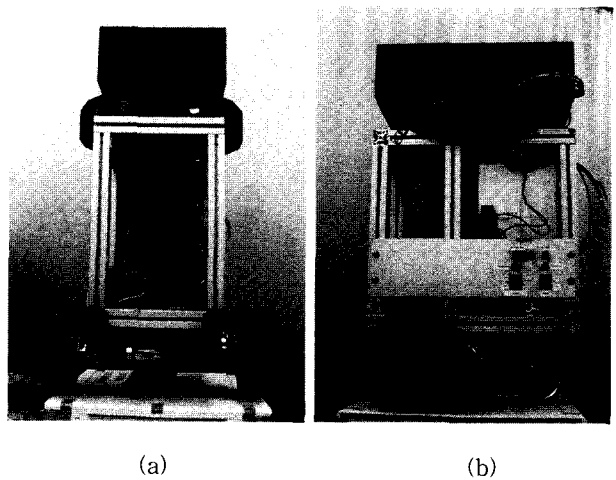


그림 7. 로봇의 사진. (a) 로봇 전면, (b) 로봇 측면
Fig. 7. The Robot.

2. 실험 구성 및 구현 방법

모의 실험 및 실제 실험은 일반적인 사무실 환경의 복도를 주행하면서 장애물을 회피하고 각 방의 문과 모퉁이를 인식하고 미리 주어진 경유점(way point)과 비교하여 확인한 후 다음 경유 점을 향하여 방향을 전환하여 주행을 계속한다. 실제 실험을 수행한 복도에는 여러 가지 실험 장비들이 놓여있으며 모퉁이는 완전 대칭형이거나 구조적이지 않으며 주위에 여러 가지 장비들이 놓여 있다.

모의 실험과 실제 실험에서 구현된 제어 구조는 그림 6과 같으며 경유점 저장소를 구성하는 경유점들의 정보는 그림 8과 같이 기술된다. 복도의 입의 지점에서 출발한 자율 이동로봇에게 초음파 센서의 정보를 처리하여 장애물이 없다는 것을 전방 감시기(monitor front)가 알려주고 현재의 위치가 모퉁이인지의 여부를 감시한다. 현재의 위치가 벽도 모퉁이도 아니며 벽과 일정한 범위의 거리에 있다면 논리 중재기는 감시기의 신호들을 이용하여 벽을 따라가는 행위 모듈(follow wall)을 선택한다. 이 행위모듈이 선택된 동안에는 계속해서 필요한 센서 정보만을 이용하여 구동기를 구동하여 벽을 따라 주행하도록 한다.

선택된 하나의 행위 모듈을 수행하는 동안 감시 모듈

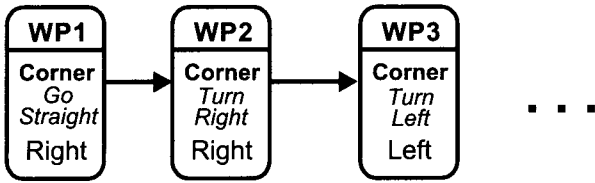


그림 8. 저장되는 경유 점들의 예.
Fig. 8. Example of way points.

은 계속해서 행위 모듈과 병렬로 센서 정보를 사용하며 환경을 감시한다. 전방 감시 모듈에서 장애물을 감지하면 이동 로봇의 제어는 장애물 회피 행위 모듈(avoid obstacle)이 담당하여 진행하려는 방향의 장애물을 회피한다. 장애물을 회피한 후 전방 감시기가 더 이상 장애물을 감지하지 못하면 장애물 회피 행위모듈은 선택되지 않고 다른 행위 모듈이 선택된다.

로봇이 주행할 때 장애물을 회피하거나 벽을 따라가는 행위 모듈이 정확히 동작하지 않는 등의 이유로 인하여 로봇이 벽과 일정 범위를 벗어나면 측면 감시 모듈(monitor side)에 의해서 논리 중재기는 벽과 정렬시키는 행위 모듈(align wall)을 선택하여 벽을 향해 방향을 전환한 후 벽을 향해서 진행한다. 이와 같이 벽을 따라가는 모듈, 벽과 정렬시키는 모듈과 장애물 회피 모듈 등을 이용하여 벽과의 거리를 일정한 범위 이내로 유지하면서 벽과 평행하게 주행할 수 있도록 한다.

벽을 따라 주행하던 이동 로봇은 문 감시 모듈과 모퉁이 감시 모듈에 의해서 문과 모퉁이를 감시하며 전방의 장애물과 벽과의 거리는 주행 중 계속해서 감시한다. 문과 모퉁이의 인식은 이동 로봇에 장착된 16개의 초음파 센서를 이용하여 측정된 환경의 거리 정보를 이용한다. 그림 9는 16개의 초음파 센서의 거리 정보값을 이용하여 모퉁이를 인식할 때 초음파 센서의 거리 정보값을 보인 것이다. 모퉁이는 모퉁이 쪽의 거리 정보의 특징만을 이용하여 인식할 수 있다. 인식 과정에서 로봇이 항상 벽과 일정한 거리를 유지하며 주행할 수 있어야 하기 때문에 벽을 따라가는 행위 모듈과 벽과의 거리가 멀 때 거리를 가깝게 해주기 위한 벽과 정렬하는 행위 모듈이 필요하다. 또한 거리가 가깝더라도 각도가 벽과 평행하지 않으면 센서 정보가 옆쪽으로 이동된 값이 되므로 이 경우는 자이로 센서를 이용하여 로봇의 절대 각도를 측정하여 이동된 값을 보상한다. 본 논문에서 실험을 통해 인식할 수 있는 자연 랜드마크의 종류는 그림 10과 같다.

실제 주행 실험에서는 장애물 회피와 문, 모퉁이 등 자연 랜드마크 인식 및 해당하는 행위 모듈이 실행되는 것을 보인다. 모의 실험을 통해서 긴 거리를 주행하며 경유 점들을 주행하는 것을 보였으며 복합제어구조의 대표적인 예인 Firby의 RAP 시스템과 비교 실험을 수행하여 총 수행 스텝 수를 비교하였다. 실제 주행 실험에서 방의 문, 모퉁이 등 자연 랜드마크가 있는 지점에는 장애물이 없다고 가정하였다. 초음파 센서의 입력 주기가 2Hz이며 실험에서 이동 로봇의 속도가 0.14m/sec임을 고려하여 0.5m/sec 정도 이상의 장애물은 없다고 가

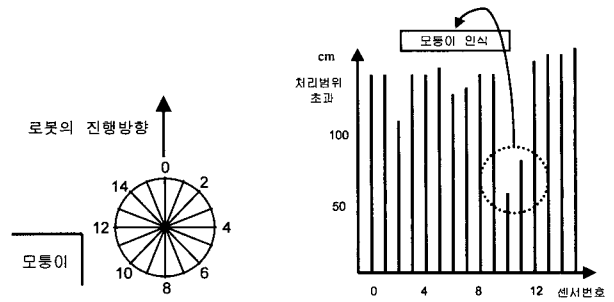


그림 9. 초음파 센서를 이용한 모퉁이의 인식
Fig. 9. Corner recognition using Ultrasonic sensor.

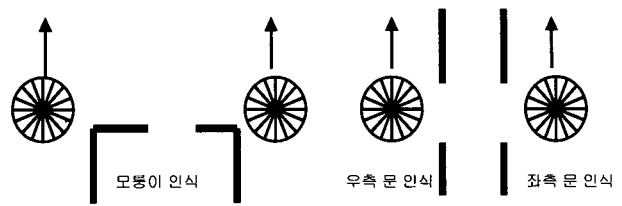


그림 10. 인식가능한 자연 랜드마크.
Fig. 10. Natural Landmarks.

정하였다.

3. 모의실험

실제 실험을 하기 전에 모의 실험을 통하여 실행순서 조정 기법의 타당성을 검증하였다. 모의 실험 환경으로는 Georgia Tech의 JavaBots[17]을 사용하였다. 실제 실험의 프로그램 환경과 동일한 구조를 유지하기 위해서 Java 언어의 쓰레드(thread)를 사용하여 실제 이동로봇의 실시간 운영 체제에서 TASK(task)를 생성하는(spawn) 기능을 모방(emulate)하였다. 본 모의 실험에서 사용된 감시기와 행위 모듈들은 실제 로봇의 하드웨어에 관련된 부분을 제외하고는 같은 구조로 프로그램 되어 로봇을 구동하는데 사용하였다. 비교실험을 하기 위해서 복합제어구조의 대표적인 예인 Firby의 RAP[10]에 대해서 동일한 감시기와 행위 모듈을 이용하였다.

그림 11과 같은 복도 환경에서 이동 로봇은 좌측 하부에서 시작하여 벽을 따라서 주행을 하면서 첫 번째 모퉁이가 인식되면 경유점 저장소(way point bin)에 기술되어 있는 것과 같이 통과하고, 주행을 계속한다. 첫 번째 문을 인식하면 이 문을 지나가며 두 번째 모퉁이에서는 오른쪽으로 회전하여 진행한다. 주행 사이에 존재하는 장애물들은 로봇의 장애물 회피 행위 모듈이 처리한다. 세 번째 모퉁이에서는 우측으로 회전을 하고 최종 목표점까지 계속 진행한 후 주행을 끝마친다. 로봇은 로봇의 오른쪽 벽을 따라가며 주행하고 이 정보도 경유점에 포함되어 있다.

모의실험에서 실행순서 조정기법을 이용한 주행 결과와 Firby의 RAP을 이용한 로봇의 주行的 결과를 비교하면, 주행 경로에는 미세한 차이가 있다. 로봇이 주어진 목표를 수행하기 위해서 필요한 스텝의 수와 각각의 행위 모듈들의 변화하는 수의 차이는 표 1과 같다. 모의 실험한 결과는 표 1에서 보는 바와 같이 제안된 제어구

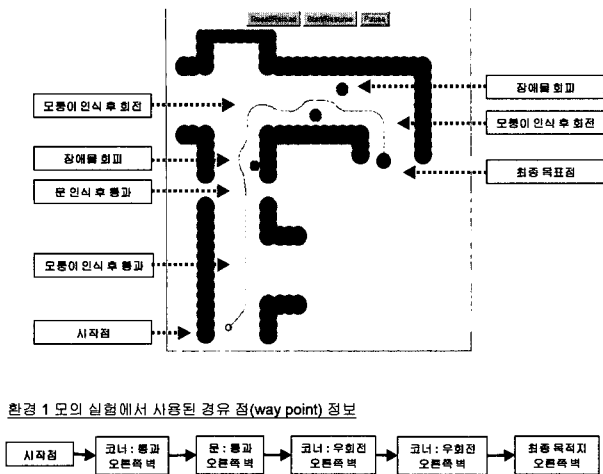


그림 11. 모의실험 결과.
Fig. 11. Result of simulation.

표 1. 제안하는 제어구조와 Firby의 RAP과 주행 결과 비교.

Table 1. Comparison of simulation result of suggested architecture and RAP.

	로봇 총 주행 스텝	행위 모듈 변화	행위 모듈 변화 비율
본 제어구조	363	51	14 %
RAP제어구조	516	119	23 %

조에 비해서 RAP을 이용한 경우, 행위 모듈의 변화하는 비율이 1.64배가 많다. 이는 제안된 제어구조가 감시 모듈에 의해 효율적으로 행위 모듈의 실행 순서를 조정함을 보이는 것이다.

RAP 제어구조에서 행위 모듈간의 변화가 많은 것은 환경 정보를 인식하는 효율이 떨어지기 때문이다. 실행 순서 조정기법에서는 병렬로 동작하는 감시 모듈이 있기 때문에, 행위 모듈이 실패하면 다음으로 수행될 행위 모듈을 선택하기 위해 전체적인 환경 정보를 인식하는 RAP보다 전체적인 환경 정보를 효율적으로 얻을 수 있다. 병렬로 동작하는 감시 모듈로 인하여 로봇의 실제 수행시간이 영향을 받지 않는다. 초음파센서나 시각 센서 등 이동로봇에 사용되는 센서들은 구동기와 로봇의 컴퓨터에 비해 절대적으로 아주 느리게 동작하므로 새로운 센서 정보가 입력되는 일정한 주기 내에 감시기가 동작하는 것을 보장하면 된다. 실시간 운영체제를 이용하여 감시 모듈을 타스크를 생성하는 방법으로 구현하여 전체 로봇의 수행 시간에 영향을 끼치지 않도록 하였다.

4. 주행 실험

주행 실험은 연구실 복도의 특정 지점에서 시작하여 왼쪽 벽을 따라가며 문을 지나고 장애물을 회피하면서 왼쪽 모퉁이를 만나면 왼쪽으로 회전하여 진행하는 것을 작업 목표로 수행하도록 하였다. 주행 실험은 복도의 벽을 따라가는 행위 모듈(follow wall), 문을 통과하는 행위 모듈(pass door), 장애물을 회피하는 행위 모듈(avoid

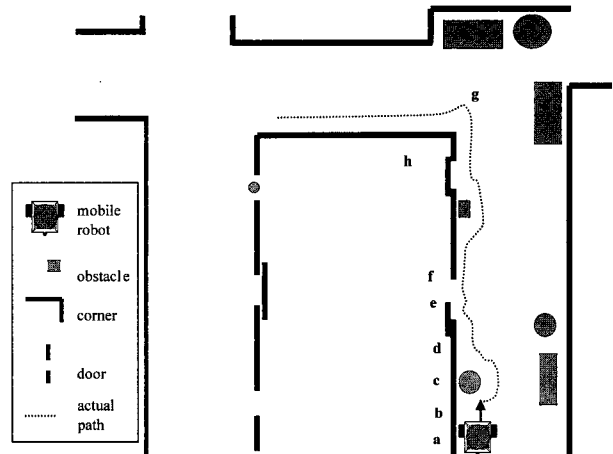


그림 12. 복도 주행 실험 환경 예.
Fig. 12. Example of real environment.

obstacle)들을 각 환경에 최적으로 대응할 수 있도록 미리 구현해 놓은 후 행위 모듈들을 순차적으로 수행시킨다. 주행 실험을 수행한 환경을 그림으로 도시하면 그림 12와 같으며 이동 로봇이 실제로 주행한 경로를 그림 12의 환경에 대응시켜 표현하면 그림 12의 실제 경로(actual path)와 같이 나타난다.

그림 13은 그림 12에 표기된 각 위치에서 실제 주행 실험을 수행한 이동로봇의 사진으로 본 연구자가 속해 있는 실험실 복도의 실제 모습이다. 그림 12의 우측 하단 (a)에서 출발하여 장애물을 회피하면서 벽을 따라 주행하며, 실험실의 문을 인식하면 문을 지나가고, 다시 벽을 따라서 주행한다. 최종적으로 모퉁이를 인식하면 왼쪽으로 회전을 하고 주행을 계속하여 주어진 작업 목표의 수행을 종료하였다. 즉, 모의 실험과 같이 복도 중간에 놓여있는 장애물들을 회피하면서 목표를 수행하였다. 앞의 IV의 2절에서 설명한 바와 같이 본 논문에서 구현한 주행실험은 초음파 센서의 입력 주기가 2 Hz 이며 이동 로봇의 주행 속도가 최대 0.14m/sec였다.

5. 실험결과 고찰

본 연구에서 제안한 제어구조에 대하여 모의 실험과 실제 실험실 환경에서 이동로봇을 이용하여 실험하였다. 모의 실험은 두 가지의 환경에 대하여 실험하였으며 기존의 복합 제어구조인 RAP과 비교 실험을 통하여 행위 모듈에 대한 변화비율에 차이가 있음을 보였다. 따라서 병렬로 동작하며 환경을 인식하는 감시기로 실행 순서를 조정하는 본 제어구조의 우수성을 살펴보았다. 주행 실험은 실제 실험실 복도에서 직접 실험함으로써 주어진 목표를 수행함을 보였다. 실제 실험에 사용하는 프로그램은 모의 실험에서 사용한 것과 동일한 구조로 구성하였다.

실험을 통하여 제안한 제어구조는 감시기 모듈을 이용하여 행위모듈의 실행순서를 효과적으로 조정할 수 있으며, 순차적으로 기술된 작업목표를 수행할 수 있음을 검증하였다.

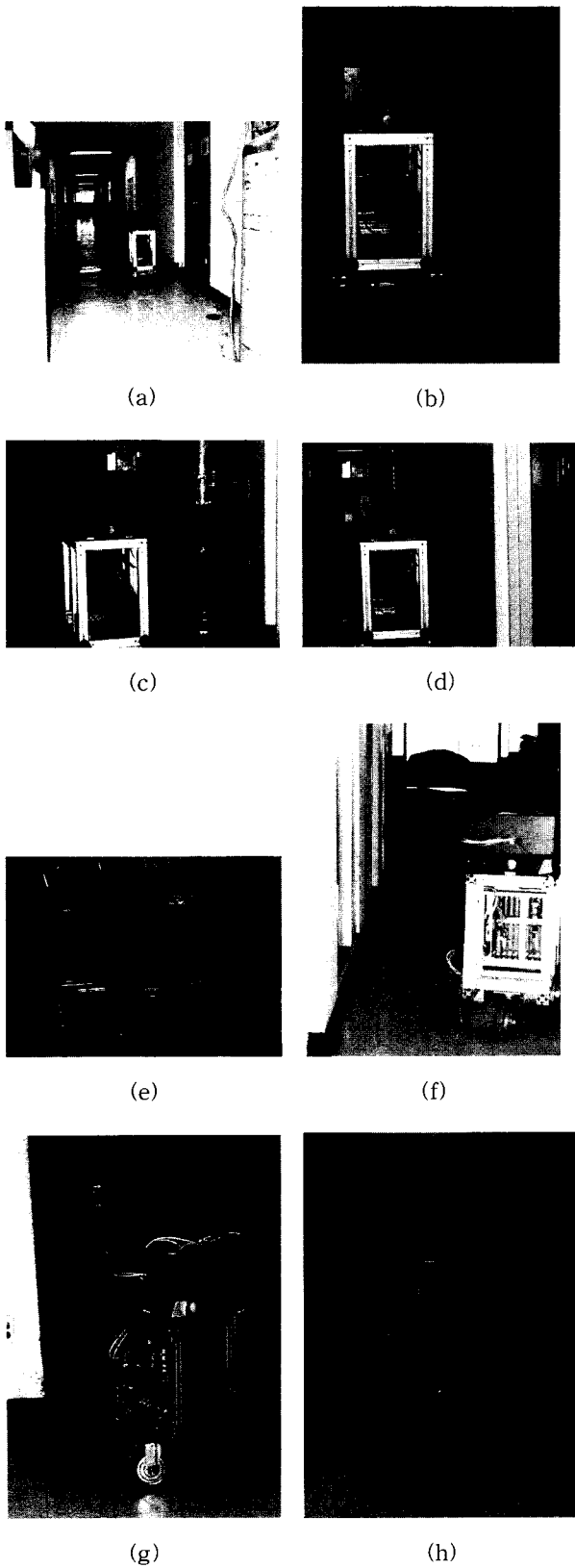


그림 13. 실제 복도의 주행 예. (a) 이동로봇의 주행, (b) 장애물 회피, (c) 장애물 회피, (d) 벽 따라가기, (e) 문 인식 후 문통과중, (f) 문 통과 후 벽 따라가기, (g)모퉁이 인식 후 회전 중, (h) 회전 후 벽 따라가기.

Fig. 13. Experiment example at real environment.

VI. 결론 및 향후 연구과제

실행순서 조정기법을 이용한 제어구조를 모의 실험과 실제 이동로봇에 적용하여 인공적이지 않은 실제 복도 환경에서 실험함으로써 빠른 반응을 보이면서 주어진 작업 목표를 수행하였다.

본 논문에서는 계층적 제어구조에서 로봇의 출력을 전체적으로 통제하는 중앙 제어를 도입하고, 행위기반 제어구조에서 동적 환경에 대한 빠른 반응 특성을 이용하여 구성된 복합 제어구조를 제안하였다. 제안한 실행순서 조정기법은 모의 실험을 통하여 여러 환경에서 센서 정보가 주기 내에 처리됨을 보임으로써 환경 변화를 빠르게 인식하여 효율적으로 제어 모듈의 변환이 가능함을 확인하였다. 하위 층은 행위 모듈로 구성되므로 모듈성이 우수하며 작업 수행의 변경이 용이해짐을 보였다. 또한 실제 복도 주행 실험을 통하여 기존의 행위 기반 제어 구조의 문제점인 상충되는 목표를 수행함에 있어서 출력 중재의 문제가 해결됨을 보였다. 더불어 행위 기반 제어 구조에서는 수행할 수 없었던 순서를 가지는 작업 목표도 수행할 수 있음을 확인하였다. 본 제어구조는 행위기반 제어구조의 특징인 빠른 응답성을 가지는 동시에 계층적 제어구조의 장점인 중앙 제어 기능이 있다. 본 제어구조의 특징은 행위 기반 제어구조의 문제점이었던 각 행위 모듈의 출력 중재에 대해서 감시기와 논리 중재기를 이용하여 해결하였으며, 중앙 제어가 로봇의 전체적인 작업 목표를 관리한다.

현재 경유점은 설계자가 직접 입력하도록 되어 있으나, 향후 연구에서 경유점을 자동으로 생성하고 재계획하는 계획기를 현재의 제어구조에 통합한다면, 목표점만이 주어졌을 때 자동으로 경유점을 생성하고 통로가 막혔을 경우 재계획할 수 있으므로 자율 능력을 좀 더 향상시킬 수 있을 것이라고 생각한다.

참고문헌

- [1] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. of Robotics and Automation*, vol. RA-2, no. 1, pp. 14-23, April, 1986.
- [2] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, pp. 139-159, Jan., 1991.
- [3] J. H. Connell, "A behavior-based arm controller," *IEEE Trans. Robotics and Automation*, vol. 5, no. 6, pp. 784-791, Dec., 1989.
- [4] E. Gat, "On the role of stored internal state in the control of autonomous mobile robots," *AI Magazine*, 1993.
- [5] Y. Jeon, J. M. Park, I. Song, Y.-J. Cho and S.-R. Oh, "An object-oriented implementation of behavior-based Control Architecture", *Proc. of IEEE Int. Conf. on Robotics and Automation, ICRA*, pp. 22-28, Minnesota, Apr., 1996.
- [6] M. J. Matric, "Integration of representation into goal-driven behavior-based robots," *IEEE Trans.*

Robotics and Automation, vol. 8, no. 3, pp. 304-312, June, 1992.

[7] R. E. Fayek, R. Liscano and G. M. Karam, "A system architecture for a mobile robot based on activities and a blackboard control unit," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 267-274, 1993.

[8] D. W. Payton, J. K. Rosenblatt and D. Keirseay, "Plan guided reaction," *IEEE Trans. System, Man, and Cybernetics*, vol. 20 no. 6, pp. 1370-1382, Nov., 1990.

[9] R. P. Bonasso, R. Firby, E. Gat, D. Kortenkamp, D. Miller and M. Slack, "Experiences with an architecture for intelligent, reactive agents," *AAAI Spring Symposium: Lessons Learned from Implemented Software Architecture for Physical Agents*, March, 1995.

[10] R. J. Firby, "An Investigation into reactive planning in complex domains," *Proc. of Sixth National Conf. on Artificial Intelligence(AAAI)*, July, 1987.

[11] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots," *Proc. of Eleventh National Conf. on Artificial Intelligence* (AAAI), pp. 809-815, 1992.

[12] M. K. Sahota, "Reactive deliberation : an architecture for real-time intelligent control in dynamic environments," *Proc. of Thirteenth National Conf. on Artificial Intelligence AAAI*, pp. 1303-1308, 1994.

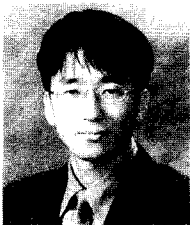
[13] R. G. Simmons, "Structured control for autonomous robots," *IEEE Trans. Robotics and Automation*, vol. 10, no. 1, pp. 34-43, Feb., 1994.

[14] E. Gat, R. Desai, R. Ivlev, J. Loch and D. P. Miller, "Behavior control for robotic exploration of planetary surfaces," *IEEE Trans. Robotics and Automation*, vol. 10, no. 4, pp. 490-503, Aug., 1994.

[15] D. Lee and M. Recce, "Quantitative evaluation of the exploration strategies of a mobile robot," *The Int. J. of Robotics Research*, vol. 16, no. 4, pp. 413-447, Aug., 1997.

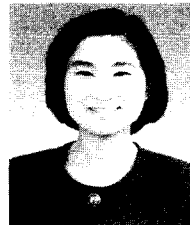
[16] *CPU-40 User's Manual*, Force Computers Inc, Nov., 1993.

[17] T. Balch, J. C. Santamar, D. Huggins, "JavaBots Ver. 1.4," *Georgia Tech's Intelligent Systems and Robotics Group*, (<http://www.gc.gatech.edu/~tucker>).



송 인 섭

1995년 고려대 제어계측학과 졸업. 동대학원 전기공학과 석사(1998). 1998년-현재 한국국제협력단(KOICA)의 국제협력봉사요원. 관심분야는 이동로봇자율 주행.



박 정 민

1989년 이화여대 전자계산과 졸업. 동대학원 석사(1991). 1991년-현재 한국과학기술연구원 지능제어연구센터 연구원. 관심분야는 소프트웨어구조, 네트워크.



오 상 록

1980년 서울대 전자공학과 졸업. 1982년 KAIST 전기 및 전자공학과 졸업(공학석사), 1987년 동대학원 박사(공학박사), 1988년-현재 KIST 지능제어연구센터 책임연구원. 관심분야는 로보틱스 및 제어공학.



조 영 조

1983년 서울대 제어계측공학과 졸업. 1985년 한국과학기술원 전기 및 전자공학과 석사, 동대학원 박사(1989). 1990년 - 1998년 한국과학기술연구원 선임연구원. 1998년-현재 한국과학기술연구원 지능제어연구센터 책임연구원. 관심분야는 네트워크기반 분산형 제어시스템, 지능형 로봇제어구조.

박 귀 태

제어·자동화·시스템공학회 논문지 제4권, 제3호, 참조.