

## 실내용 서비스 로봇을 위한 거리 센서 기반의 통합 자율 주행 시스템 개발

### Development of Range Sensor Based Integrated Navigation System for Indoor Service Robots

김 건 희, 정 우 진\*, 김 문 상  
(Gunhee Kim, Woojin Chung, and Munsang Kim)

**Abstract :** This paper introduces the development of a range sensor based integrated navigation system for a multi-functional indoor service robot, called PSR (Public Service Robot System). The proposed navigation system includes hardware integration for sensors and actuators, the development of crucial navigation algorithms like mapping, localization, and path planning, and planning scheme such as error/fault handling. Major advantages of the proposed system are as follows : 1) A range sensor based generalized navigation system. 2) No need for the modification of environments. 3) Intelligent navigation-related components. 4) Framework supporting the selection of multiple behaviors and error/fault handling schemes. Experimental results are presented in order to show the feasibility of the proposed navigation system. The result of this research has been successfully applied to our three service robots in a variety of task domains including a delivery, a patrol, a guide, and a floor cleaning task.

**Keywords :** navigation system, autonomous map building, localization, path planning, discrete event control, Petri nets, service robots

#### I. 서론

근래에 들어 로봇 기술은 외과 수술, 재활, 바닥 청소, 순찰 등 비산업 분야로의 적용이 활발해지고 있다. 특히 실내용 서비스 로봇 분야는 그 잠재적인 기술적, 경제적, 사회적 파급 효과로 인해 많은 관심을 받고 있다. 또한 이 분야는 인간 로봇 상호작용(Human Robot Interaction), 동적 환경에서의 주행(Navigation) 및 조작(Manipulation) 작업 등, 현재 로봇 학계에서 가장 중점적으로 연구되고 있는 주제와도 밀접하게 연관되어 있다.

한국과학기술연구원(KIST)에서는 실내 서비스 작업을 위하여 다기능 로봇 시스템인 PSR(Public Service Robot)을 개발해 왔다. 현재 두 가지 종류의 로봇들이 개발되었는데, 하나는 PSR-1, PSR-2와 같은 오피스 환경에서의 서비스 로봇이고, 다른 하나는 안내 로봇 지니(Jinny) 시스템이다. 전자의 경우에는 주행과 조작의 협조 기술이 중요하고, 후자의 경우에는 주행과 인간 로봇 상호작용 기술이 강조되는 특성이 있다. 현재 이 로봇들을 이용해 물건 이송, 순찰, 안내, 바닥 청소 등 4가지 목표 작업이 성공적으로 구현하였다.

자율 주행은 서비스 로봇을 개발하기 위한 가장 기본적인 기술 중 하나로, 그 중요성으로 인해 지금까지 많은 연구가 있어 왔다. 기존 연구 중, 서비스 로봇에 완성된 형태로 구현된 주행 시스템들이 대표적인 예는 다음과 같다.

Minerva[1]는 학계에서 큰 주목을 받은 안내 로봇의 하나로, 1998년 미국 Smithsonian 미국사 국립 박물관에 2주간

설치되었다. 이 기간 동안 어떠한 가공도 가하지 않은 매우 불비는 환경에서, 개발된 자율 주행 기술이 성공적으로 실행됨을 보였다. Minerva의 주행 시스템은 마코프 위치 추정 기술(Markov localization)과 천장 모자이크 기술, 그리고 모델 기반 Dynamic window 기술 등으로 구성되어 있다.

Xavier[2]는 오피스 환경에서 자율 주행하면서, 촬영한 사진들을 웹상에 실시간으로 전송하는 기능을 가진 로봇이다. 이 로봇의 주행 시스템에는 위상 지도(Topological map) 기반의 경로 계획기와 장애물 회피 알고리즘, 그리고 POMD (Partially Observable Markov Decision Process) 위치 추정기 등이 적용되었다. 또한, 사용자는 World Wide Web 인터페이스를 통해 간단한 주행 작업 명령을 내릴 수 있고, 현재 로봇의 상태 및 주변 공간의 위치 정보들을 모니터링 할 수 있다.

Care-O-Bot-II[3]은 노인과 장애인을 위한 가정용 로봇으로 개발되었다. 이 로봇이 경로 계획 알고리즘은 Latombe이 제시한 Wavefront algorithm[4]과 Khatib의 elastic band 방법[5]을 혼용하여 사용하였고, 위치 추정은 엔코더와 자이로, 레이저 거리 센서를 이용한 Kalman filter 기반의 map-matching 방법을 적용하였다.

RoboX[6]는 최근에 개발된 안내 로봇 시스템 중 하나로, 2002년에 스위스 국립 전시회에서 10대가 5달 동안 설치 운용되었다. 이 시스템에는 카메라를 이용한 얼굴 추종 기술, 레이저 거리 센서를 사용한 인간 감지 기술, 음성 인식 및 합성 기술 등 여러 종류의 관람객 상호작용 기술이 적용되었다. RoboX의 주행 시스템은 확률론적인 Feature matching 위치 추정기와 Navigation function, Elastic band, Dynamic window 방법들을 혼합한 경로 계획기로 구성된다[7].

본 연구에서는 개발에 앞서 현장 조사를 통해 실제 서비스 작업을 수행하기 위해 필요한 주행 시스템의 목표 사양을 도출하였다. 또한, 이를 도달하기 위한 세부 기술들을 분류

\* 책임저자(Corresponding Author)

논문접수 : 2004. 6. 6., 채택확정 : 2004. 7. 18.

김건희, 정우진, 김문상 : KIST 지능로봇연구센터

(knir38@kist.re.kr/wjchung@kist.re.kr/munsang@kist.re.kr)

하였고, 주행 시스템의 요구 사항을 다음 네 가지로 정리하였다.

1) 로봇은 별도의 인위적 표식 없이 사람이 많이 존재하는 환경에서 주행 가능해야 한다. 이와 같은 동적인 특성은 위치 추정 문제나 안전 문제 등을 야기할 수 있다.

2) 개발될 주행 시스템은 환경의 불확실성으로 인한 여러 가지 예외 상황에 대하여 충분히 대처 가능해야 한다. 예를 들어, 만약 로봇이 사람에 의해 둘러싸이게 된다면, 로봇은 위치 추정에 실패하거나 혹은 지정된 위치로의 이동이 불가능할 수 있다. 따라서 개발될 시스템은 실생활에서 자주 발생하는 이와 같은 상황들을 효과적으로 처리할 수 있는 프레임워크를 갖추어야 한다.

3) 개발될 주행 시스템은 대역적(Global), 지역적(Local) 접근 방법을 둘 다 가지는 것이 유리하다. 병원이나 오피스 건물과 같은 대형 환경에서는, 자세한 정보를 모두 한꺼번에 처리하는 것이 상당히 비효율적이다. 대신 대역, 지역 이 두 가지 방식을 적절히 혼용하는 것이 더 나은 성능을 가져올 수 있다.

4) 본 시스템에 적용될 개별 컴포넌트의 알고리즘들은 일반성(Generality)을 가져야 한다. 즉, 하나의 요소 기능을 구현하기 위해서는, 서로 다른 여러 알고리즘을 융합하는 것보다는 단일 알고리즘을 사용하는 것이 더 바람직하다. 전자의 경우에는, 각각의 알고리즘을 구현하고, 매개변수 설정 등의 복잡한 통합과정을 필요로 하므로, 일반성 있는 알고리즘만을 사용하는 경우보다 복잡도나 실패율이 높다. 예를 들어, 본 시스템에서는 경로 계획을 구현하기 위해, 다른 많은 시스템에서 사용하는 것처럼 고정된 목표 궤적을 만드는 알고리즘과 동적으로 장애물 회피하는 알고리즘을 융합하여 사용하는 것을 지양한다. 또한, 복도 주행, 문 통과 등등 여러 상황을 분류해서 각각 서로 다른 알고리즘을 적용하는 방법도 피한다. 이 방법은 투입 환경에 따라 주행 알고리즘을 바꿔야 하는 단점이 있기 때문이다.

앞서 조사한 기존 주행 시스템 중에는 위의 네 가지 요구 사항을 완전히 만족시키는 시스템이 없었다. 본 논문의 범위는 다양한 주행 요소 기술들을 구현한 후 실험적으로 그 유효성을 입증하고, 실제 환경에 적용 가능하도록 개발된 기술들을 효과적으로 통합하는 데에 있다. 제안된 방법론은 모든 요소를 갖춘 완성된 해결책으로써, 센서 및 액추에이터를 포함한 하드웨어 통합, 지도 작성, 경로 계획, 위치 추정을 포함한 핵심 주행 기술 개발, 작업 계획 및 고장 진단, 복구와 같은 계획 체계 등을 모두 포함하고 있다.

본 논문에서 제안된 주행 시스템의 주요 특징은 다음과 같다.

1) 환경 정비가 필요 없는 거리 센서 기반의 일반적인 주행 방법론 : 지도 작성, 경로 계획, 위치 추정 등 구현된 주행 컴포넌트들은 거리 센서를 기반으로 한다. 제안된 주행 방법론은 적외선 센서, 초음파 센서, 레이저 센서 등 어느 종류의 거리 센서를 사용해도 무방하다. 본 시스템의 여러 가지 역할을 수행하는 일반화된 지도는 위치 추정 및 최적 경로 생성을 위한 기본적인 데이터 베이스 역할을 한다.

또한, 개발된 주행 알고리즘을 어떠한 인위적 표식도 필요로 하지 않는다.

2) 지능을 갖춘 주행 컴포넌트 : 제안된 주행 시스템의 경우, 각 컴포넌트들은 자신만의 고유한 지능을 가지고 있다. 즉, 각 컴포넌트들은 자신에게 주어진 가장 기본적인 기능 수행 뿐만 아니라 로봇 내부 및 환경 정보의 상태를 분석할 수 있는 기능도 아울러 가지고 있다. 예를 들어, 경로 계획기는 본연의 임무인 목표 경로 생성 뿐만 아니라 장애물에 의한 목표점 점유와 같은 중요한 상태 정보들을 이벤트의 형태로 제공한다. 이와 같은 이벤트들은 로봇의 거동을 결정하는 데에 직접적인 영향을 준다.

3) 다중 행위(Behavior)의 선택적 적용 및 오류 검지/복구를 지원하는 제어 프레임워크 구축 : 앞서 말한 지능을 갖춘 컴포넌트가 증가함에 따라, 이벤트 처리는 점점 어려워진다. 단순한 If-then 로직만으로는 이와 같은 복잡한 상황 분석은 어렵게 된다. 따라서, 본 시스템에서는 페트리넷(Petri net) 기반의 이산 이벤트 프레임워크가 제시되어 있다. 이를 이용하여 이벤트 처리, 작업 계획, 오류 검지/복구 등 여러 중요한 문제들을 동시에 해결할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 PSR 시스템의 하드웨어 및 소프트웨어 아키텍처를 소개한다. 3장에서는 제안된 주행 시스템의 개요, 지도 구성, 지도 작성 알고리즘, 위치 추정, 경로 계획, 추종 행위, 통합 제어 방법론 등을 순서대로 설명한다. 4장에서는 네 목표 작업의 구현을 통해 개발된 주행 시스템의 평가 결과를 보여주고, 마지막 5장에서는 결론을 제시한다.

## II. PSR 시스템 아키텍처

### 1. 하드웨어 구조

그림 1은 개발된 세가지 PSR 시스템의 모습을 보여 준다. PSR-2는 PSR-1의 차기 버전으로 서로 유사한 하드웨어 구성을 가지고 있다. 지니의 경우, 상업화를 목표로 만들어져 이전 버전과는 다른 구성을 보인다. 예를 들어, PSR-1의 경우에는 16축 동시 제어 가능한 상용 제어기인 Delta Tau사의 PMAC 보드[8]를 장착한 반면에, 지니는 상대적으로싼 서보 제어기를 가지고 있다. 이와 같은 제어기의 차이로 하위 제어의 샘플링 타임은 PSR-1과 PSR-2의 경우 700 ns정도이며, 지니의 경우에는 30ms이다. 또한, PSR-2는 4시간 정도 동작이 가능한 배터리를 포함하고 있지만, 지니는 8시간 연속 운전이 가능하다.

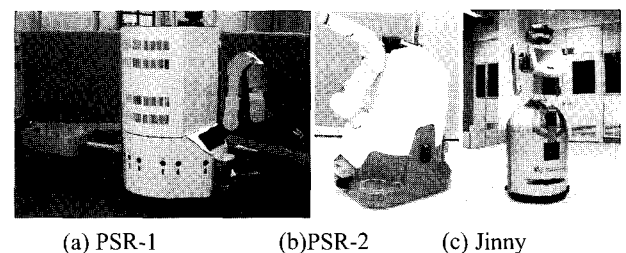


그림 1. PSR 시스템.  
Fig. 1. PSR Systems.

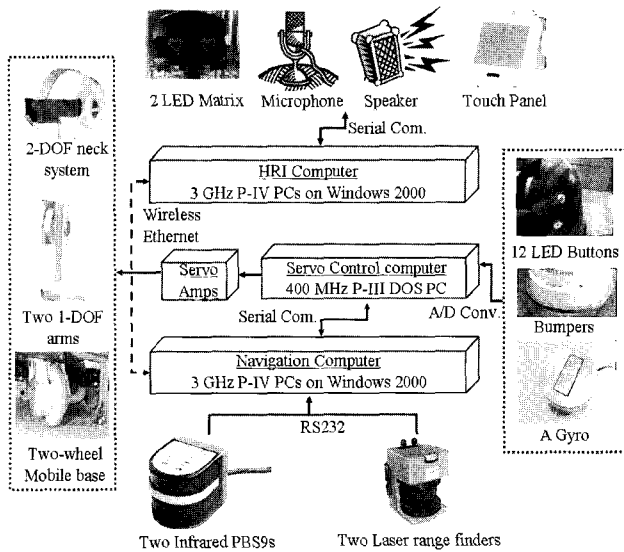


그림 2. 지니의 하드웨어 아키텍처.  
Fig. 2. Hardware Architecture of the Guide Robot Jinny.

그림 2는 지니의 하드웨어 아키텍처에 대한 설명이다. 지니에는, 세 대의 PC가 사용되는데 Windows 2000 OS가 설치된 두 대의 3 GHz Pentium-IV PC는 각각 주행 제어와 인간 로봇 상호작용(HRI)을 담당한다. 이 둘 사이의 통신은 10Mbit/s 무선 Ethernet을 사용하였다. 나머지 한 대의 400MHz Pentium-III DOS PC는 1 자유도의 두 팔, 2 자유도의 목, 이동부의 하위 제어를 담당하고, 다른 PC와 RS-232로 연결된다.

PSR 시스템은 자율 주행을 위해 여러 종류의 센서가 장착되어 있다. PSR-1은 한 대의 레이저 거리 센서와 적외선 및 초음파 센서 18세트가 로봇 들레에 따라 배치되어 있다. PSR-2의 경우, 두 대의 레이저 거리 센서가 앞뒤로 달려 있으며, 좌우에는 적외선 스캐너가 각각 부착되어 있다. 지니는 앞 뒤로 두 대의 적외선 스캐너가 레이저 거리 센서와 다른 높이로 장착된다. 지니는 두바퀴 차륜형 구조이므로, PSR-1이나 PSR-2와 같이 반드시 전방향의 시야를 확보할 필요가 없다. 단, 안전을 위해 범퍼 센서가 최하단부에 위치한다. 자이로 센서는 정확한 각도 측정을 위해 모든 PSR 플랫폼에 설치되어 있다.

지니는 두 바퀴로 구동 되는 차륜형 주행부를 가지는 반면, PSR-1과 PSR-2는 홀로노믹(Holonomic) 전방향 동작이 가능하다. 이와 같이 구성된 이유는, PSR-1과 PSR-2의 경우에는 조작부와의 협조 작업, 트레일러 제어 등 부가적인 작업을 수행하기 때문이다. 개발된 이중오프셋 능동회전(Double-offset active orientable) 구동부는 그림 3에 소개되어 있다. 이 구조를 통해 제어를 쉽게 할 수 있으며, 신뢰도와 내구성을 높일 수 있다. 개발된 주행부는 각 바퀴마다 회전(Steering)과 추진(Driving)을 담당하는 모터 2개씩이 설치된다. 바퀴 축은 휠 모듈의 회전축(i.e., Pivot point)에 대해 세로와 가로 방향으로 각각 오프셋이 존재한다. 이 구동부의 자세한 설계나 운동학적 분석, 만들어진 프로토타입과 간단한 성능 시험 등은 참고문헌[9]에 자세히

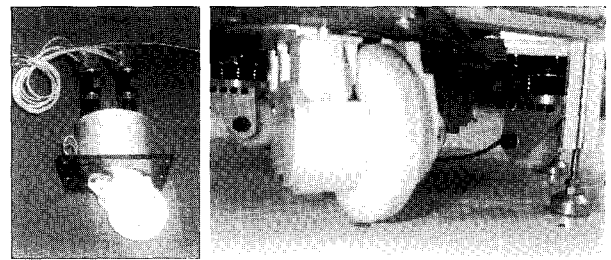


그림 3. 이중 오프셋 능동 캐스터 바퀴.  
Fig. 3. A Double-offset Active Caster Wheel.

소개되어 있다. 이 구동부는 0.5 m/s<sup>2</sup>의 최대 가속도와 1.0 m/s의 최고 속도로 움직일 수 있다.

2. 소프트웨어 구조

잘 정의된 제어 구조는 PSR과 같은 대형 로봇 시스템 개발에 필수적이다. 시스템 통합을 원활하게 하기 위해 제어 구조는 다음과 같은 주요 설계 요구 사항을 가진다.

1) 다양한 하드웨어 모듈의 결합 : PSR은 앞 절에서 설명한 바와 같이, 서로 다른 물리적, 기능적 특징을 지닌 다수의 하드웨어 모듈이 매우 복잡한 형태로 연결된 시스템이다.

2) 소프트웨어 컴포넌트의 통합 : 목표 작업과 로봇 플랫폼의 수가 늘어남에 따라, 필요한 소프트웨어 컴포넌트의 수가 폭발적으로 증가하게 되고 이들의 성능은 아키텍처의 효율성에 크게 의존하게 된다. 아울러 이 문제는 정보 연결 관계 정의나 정보 처리의 스케줄링과 같은 부가적인 문제들을 야기할 수 있다.

3) 컴포넌트의 재구성(Reconfiguration) : 목표 작업이 많아짐에 따라, 컴포넌트 공유나 재구성이 빈번하게 일어나게 된다. 또한, 이 과정은 동작 시점에 실시간으로 이루어질 수 있어야 한다.

4) 모듈화(Modularity) 및 재사용성(Reusability) : 여러 대의 로봇 플랫폼이 존재하지만 이들은 하나의 소프트웨어 프레임워크에 의해 동작되는 것이 자연스럽다. 본 아키텍처를 통해 개발된 소프트웨어들은 특별한 수정 없이 다른 플랫폼에서 재사용이 가능해야 한다.

5) 병렬적 개발 프로세스 : 여러 개발자 혹은 새롭게 참여하는 개발자들은 서로 독립적으로 자신의 컴포넌트를 추가, 수정을 할 수 있어야 한다. 따라서, 전체 구조를 쉽게 이해할 수 있는, 공통된 시스템 표현 도식을 가지는 것이 유리하다.

우리는 선행 연구[10][11]를 통해 아키텍처 및 통합 문제에 대한 해결책으로, Tripodal schematic control architecture를 제안하였다. 제안된 방법의 핵심 개념은 Layered functionality diagram, Class diagram, Configuration diagram이라는 세 가지 종류의 도식을 사용하여 로봇 통합을 진행한 것이다.

기존 연구에 비해, 제안된 제어 구조의 주요한 두 가지 장점은 다음과 같다. 첫째, 개발된 제어 구조는 시스템 통합의 지침으로써 세 가지 종류의 이해하기 쉽고, 직관적인 도식을 제안하였다. 이와 같은 도식 기반의 방법은 다수의 개발자가 참여하는 복잡한 시스템의 개발에 매우 유용하다.

컴포넌트의 구조, 정보의 흐름, 상호 의존도 같은 구조적 정보들을 한눈에 쉽게 이해할 수가 있다. 또한, 통합 과정이 제시된 지침에 단순히 따라가기만 하면 되므로 매우 직관적이다. 둘째, 개발된 제어 구조를 통해 작업 계획이나 오류 진단/복구 등의 처리 등을 페트리넷(Petri net) 기반의 정규화된(Formal) 방법으로 해결할 수 있다. 이를 통해 컴포넌트의 재조합이나, 실시간에서의 로봇 행위의 선택 적용, 오류 진단/복구 등이 시각적이고 수학적으로 표현될 수 있다.

### III. 통합 주행 시스템

#### 1. 주행 시스템 개요

그림 4는 본 연구에서 개발된 통합 주행 시스템의 개요를 보여주고 있다. 시스템은 크게 하위 계획부(Low-level planning part), 정보 생성부(Information part), 반응 제어부(Reactive part) 이렇게 세 부분으로 나뉜다. 이는 앞서 개발된 제어 구조의 일부로써, 여러 제어 구조적 문제를 고려하여 설계되었다.

위 그림에서의 주행 모듈(Navigation modules)은 많은 계산 시간을 요하는 정교한 알고리즘을 사용하여, 원 센서 데이터로부터 고차원적인 정보를 생성해낸다. 대표적인 모듈로써는 지도(Maps), 위치 추정기(Localizer), 경로 계획기(Path planner) 등이 있다.

반응 제어부는 실시간으로 간단한 계산을 반복 수행하며 로봇의 동작을 제어한다. 리소스(Resources)는 여러 종류의 센서 정보들을 관리하는 프레임워크이다. 행위(Behavior)는 가장 최소단위의 행동 모듈로써, 센서, 액추에이터와 밀접하게 연결되어 있다. 행위 조정기(Behavior coordinator)는 다수의 행위들로부터 온 제어 명령을 혼합하여 하나의 제어 명령을 만들고, 제어기(Controller)는 드라이버를 통해 액추에이터를 구동한다.

하위 계획부는 프로세스 관리자(Process supervisor)와 하위 컨피규레이션(Low-level configuration)으로 이루어져 있다. 프로세스 관리자는 하위 컨피규레이션을 참조하여 본 아키텍처에서 하위작업(Subtask) 개념으로 사용되는 프로세스들을 수행한다. 하위 컨피규레이션은 필요한 행위 목록, 매개 변수, 오류 복구 로직 등 프로세스를 수행하기 위해 필요한 모든 정보를 캡슐화하여 관리한다. 이와 같은 하위 컨피규레이션 내의 정보를 기술하기 위해 앞서 밝혔듯이 페트리넷이 사용된다. 본 논문의 III-7절에서는 기본적인 주행 프로세스인 PrAutoMove에 대한 페트리넷 모델을 설명하고 있다.

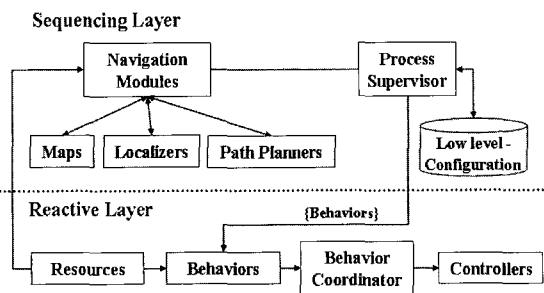


그림 4. 주행 시스템 개요

Fig. 4. Overview of Integrated Navigation System.

전체 주행 시스템의 동작에 대한 이해를 돕기 위해 간단한 물건 이송 작업을 예로 들면 다음과 같다. 로봇이 멀리 떨어져 있는 방으로 물건을 옮기도록 명령을 받게 되면, 우선 대역 경로 계획기가 위상 지도 상의 노드 경로 집합을 생성한다. 여기서의 각 노드 점은 주행 프로세스인 PrAuto Move의 목표점이 된다. 따라서, 이 프로세스를 반복함으로써, 로봇은 원하는 위치까지 이동이 가능하다. 만일 로봇이 처음 계획된 노드 경로 상에 막힌 곳을 발견하게 된다면, 위상 지도는 갱신되고, 대역 경로 계획이 다시 수행 된다.

앞으로 이어지는 절에서는 본 주행 시스템의 핵심 컴포넌트인 지도 구성, 자율 지도 작성 알고리즘, 위치 추정 알고리즘, 경로 계획 알고리즘, 추종 행위, 컨피규레이션 설계에 대해 차례로 설명한다.

#### 2. 지도 구성

지도는 사용자와 로봇 모두에게 매우 중요한 정보를 제공하는 컴포넌트이다. 본 시스템의 경우, 네 가지 종류의 지도가 존재하는데, 사용자를 위해서는 정보 지도(Information map)가 사용되고, 로봇을 위해서는 위상 지도(Topological maps), 격자 지도(Local grid maps), 활성화 지도(Active maps)가 있다.

정보 지도는 사용자 인터랙션 표시장치에서 사용자 친화적인 정보를 제시한다. 예를 들어, 사용자는 정보 지도를 통해 화장실이나 가고자 하는 방이 어디에 있는지 쉽게 알 수 있다. 이는 작업 명령을 내리거나 로봇 상태를 모니터링 하는 데에 유용하게 쓰인다.

위상 지도는 대역 경로 계획에 사용되는데, 전체 환경을 노드와 아크로 간결하게 기술한다. 각 노드는 실제로 의미 있는 지점을 나타낸다. 예를 들어, 관람물의 위치나, 우편함의 위치 등 로봇이 특정 작업을 해야 하는 곳은 노드점으로 지정될 수 있다. 따라서, 위상 지도의 노드점들은 사용자가 직접 지정하게 된다. 또한, 위상 지도는 지역 격자 지도들 사이의 연결 관계를 기술하는 역할도 가진다. 본 시스템에서는 메모리의 한계와 계산 부담으로 인해 전체 작업 영역을 여러 개의 지역 격자 지도의 조각들로 저장하고 있다. 로봇은 현재 자신의 위치에 따라 적절한 지역 격자 지도를 불러와서 사용한다. 이와 같이 위상 지도와 격자 지도를 혼합해서 사용함으로써, 각각의 장점을 극대화할 수 있다.

지역 격자 지도는 로봇 주위에 대한 정확한 정보를 가지고 있다. 이 지도는 위치추정 알고리즘에 사용된다. 사용되는 위치추정 알고리즘은 Map-matching 기법을 사용하므로, 계산 시간이 지도의 크기에 밀접하게 연관되어 있다. 따라서, 지도의 크기는 너무 크지 않도록 적절히 지정되어야 한다. 이 지도는 전체 환경을 10cm의 등간격 격자로 표현한다. 격자 값은 장애물의 유무나 바닥 상태 등 그 지점의 환경 특성에 맞게 지정되어 있다. 격자 지도의 경우, III-4절에 제시된 방법으로 자동 작성이 가능하다.

활성화 지도는 지역(local) 경로 계획을 위해 정의되었다. 이 지도는 위에 설명한 지역 격자 지도의 일부를 불러와서 생성하므로 지역 격자 지도와 유사한 구조를 가진다. 생성된 이후에는 실시간으로 여러 종류의 센서가 독립적으로 환경 정보를 갱신한다. 갱신 방법은 Histogramic In-Motion Mapping

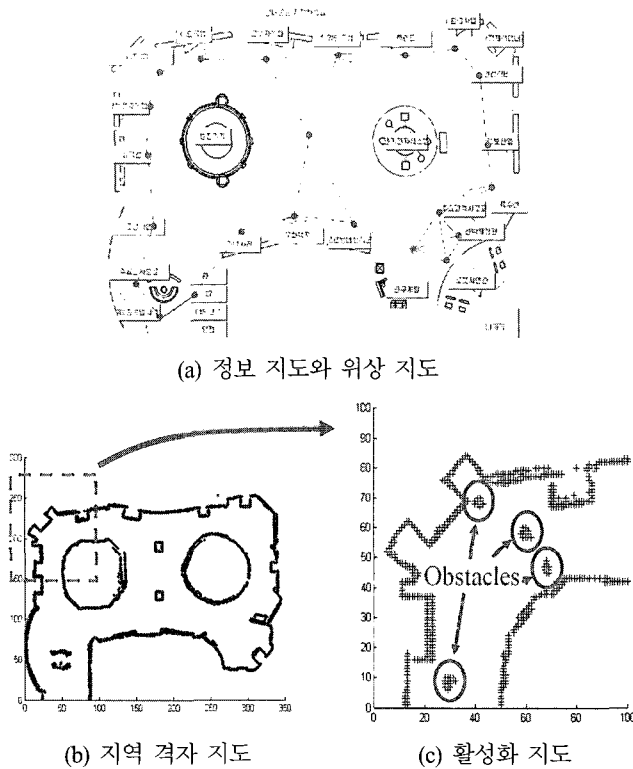


그림 5. 지도 구성.  
Fig. 5. Map Structure.

(HIMM)[12]를 사용한다. 지도 갱신 속도는 각 센서의 샘플링 타입에 따라 다른데, 레이저 거리 센서의 경우에는 100 ms. 마다 새로운 정보로 갱신한다. 활성화 지도는 프로세스 PrAutoMove가 시작될 때 생성되고, 종료될 때 삭제된다. 이 지도의 크기는 10m × 10m로 고정되어 있으므로, 이 영역 안에 반드시 시작 노드점과 목표 노드점이 존재해야 한다.

그림 5는 지금 설명한 네 가지 지도의 예제를 보여준다.

3. 위치추정(Localization) 알고리즘

위치추정은 성공적인 자율 주행에 있어 가장 중요한 문제 중 하나이다. 본 절에서는 실제 환경에서 신뢰성 있는 위치 추정이 가능한 통합 위치추정 알고리즘을 소개한다. 본 절의 내용은 우리의 선행 연구[13][14][15]의 일부이다.

본 위치 추정 알고리즘의 기여점은 다음 두 가지이다. 첫째, RISF(Range Image Similarity Function, grid map matching)와 ASF(Angular Similarity Function, feature map matching)라는 새로운 두 가지 종류의 Map-matching 함수가 제시되었다. 샘플링 기반의 확률론적 위치추정 기법인 RISF는 다각형 구조나 비다각형 구조의 환경에 상관없이 일반적으로 적용 가능하고, ASF는 다각형 구조의 환경에서 더 빠르고 신뢰도 높은 추정이 가능하게 한다. 둘째, 개발된 알고리즘은 이산이벤트 시스템 개념을 도입하여 체계적으로 구조화 되었다. 따라서, 위치추정 뿐만 아니라, 다양한 로봇 내부 및 환경 상황들을 분석하여 로봇 거동에 영향을 줄 수 있는 의미 있는 정보를 제공할 수 있다.

제안된 위치추정 방법은 실제 환경에서의 여러 차례 실험

을 통해 그 유효성이 입증되었다.

3.1. 샘플링 기반의 확률론적 Map-matching 위치추정

본 위치 추정 방법은 인공 표식 없이 거리 센서를 이용하는 map-matching 방식이다. 이 방법은 몬테카를로 위치추정 방법(MCL, Monte Carlo localization)을 기반으로 개발되었다[16].

RISF에서는 실제 센서로부터 얻은 거리 값과 격자 지도로부터 예측된 거리 값의 차이를 전체 센싱 영역에 대해 비교한다. 따라서, RISF는 이동하는 장애물로 인해 부분적으로 센서 값이 오염되어도 신뢰성 있는 결과를 얻을 수 있다는 장점이 있다.

$$\text{for } i = 1 \rightarrow N \text{ (number of samples)} \\ \text{for } j = 1 \rightarrow 361 \text{ (number of scan elements)} \quad (1)$$

$$p(o_{i,t} | s_{i,t}, m) \leftarrow \frac{1}{|\text{ScanDist}_j - \text{ReferDist}_{i,j}|}$$

ScanDist<sub>j</sub>는 j번째 측정값이고, ReferDist<sub>i,j</sub>는 i번째 샘플의 j번째 예측 값이다. 로봇이 특정 점에 위치할 확률 p(o<sub>i,t</sub> | s<sub>i,t</sub>, m)은 (1)에 의해 구해진다. (1)에서 분모가 0이 되는 경우는 실제 측정값과 예측 값이 정확히 일치하는 경우인데, 실제로 이와 같은 상황은 거의 발생되지 않는다.

ASF는 센서로부터 얻은 기하학적 특징들과 미리 등록된 특징 지도 사이의 비교를 통해, 다각형의 형태를 띤 환경에서 위치추정 정확도를 높인다. 실제 서비스 로봇이 투입될 많은 실내 환경은 다각형의 모습이며, 벽, 복도, 기둥과 같은 여러 이용 가능한 기하학적 특징 패턴이 존재한다.

이와 같은 기하학적 특징 패턴들은 Hough 변환을 통해 추출된다. 본 방법에서는 선 형태의 패턴만을 고려한다. 측정된 센서 정보와 지도로부터 예측된 정보는 Hough 매개변수인 (ρ, θ)의 차이로써 유사도 측정에 사용된다. Hough 변환의 결과는 Hough 영역에서의 득표수(Voting number)로 표현된다. 득표수를 통해, 로봇은 기하학적 특징을 이용할지, 얼마나 많은 특징 선들을 뽑아낼지 등을 자동적으로 결정할 수 있다.

$$\text{for } i = 1 \rightarrow N \text{ (number of samples)} \\ \text{for } j = 1 \rightarrow NHT \text{ (number of geometric features)}$$

$$\text{sum} = \text{voting}_j((\text{ReferHT}\theta_{i,j} - \text{ScanHT}\theta_j)^2 + (\text{ReferHT}\rho_{i,j} - \text{ScanHT}\rho_j)^2) \quad (2)$$

$$p(o_{2,t} | s_{i,t}, m) \leftarrow \frac{1}{\text{sum}}$$

ScanHT<sub>j</sub>는 j번째 측정 Hough 선이고, ReferHT<sub>i,j</sub>는 i번째 샘플의 j번째 예측 Hough 선을 나타낸다. 수식 (2)에서 보듯이, 이 둘 사이의 차이를 통해 로봇의 위치 확률을 계산한다. 측정된 선은 현재 로봇의 위치에 따라 다른 득표수를 가지게 되므로, 득표수는 유사도 함수의 가중치로써 사용된다. ASF는 기하학적으로 잘 정돈된 환경에서 더 빠르고 더 정확한 샘플의 수렴을 돕는다. 특히 로봇의 각도 위치 추정에 중요한 역할을 한다.

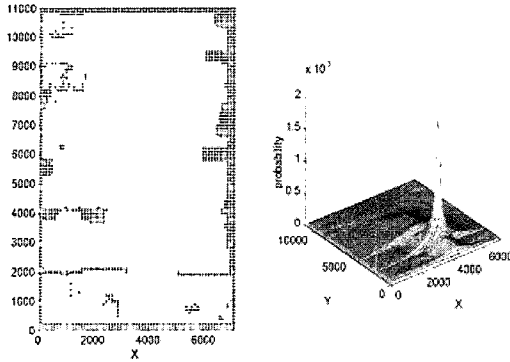


그림 6. 격자 지도의 RISF 결과.  
Fig. 6. The Grid Map and RISF Feasibility.

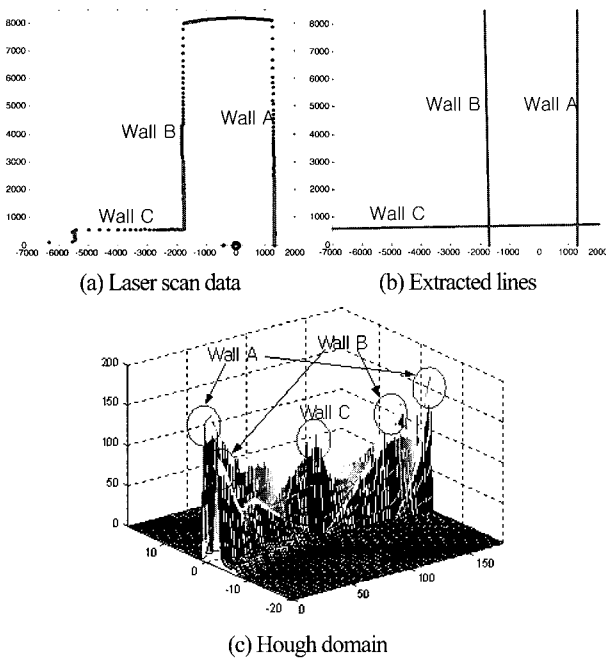


그림 7. Hough Transform 결과.  
Fig. 7. The Results of Hough Transform.

그림 6은 RISF의 실험 결과를 보여준다. 왼쪽 그림은 실험 환경의 격자 지도이다. 실험에서 사용한 데이터는 지도 상의  $[x, y, \theta] = [4000, 2000, 0]$  점에서 측정된 레이저 값이다. 우측 그림은 지도 상의 각 점에서 로봇이 위치할 확률을 z축에 나타내었다. 그림에서 보듯이 실제 위치인  $[4000, 2000, 0]$ 에서 가장 높은 확률 값  $p(o_i | s_i, m)$ 을 가진다. 즉, 샘플들이 RISF를 통해 실제 위치에 제대로 수렴되었음을 알 수가 있다.

그림 7은 Hough 변환의 예를 보여준다. 그림 7.(a)는 로봇 정면의 180도 영역에 대해 레이저로 측정된 값이고, 그림 7.(b)는 그로부터 얻어진 Hough 선이다. 그림 7.(c)는 이 측정값에 대한 Hough 영역 상의 득표수가 나타나 있다. 해상도는  $\theta$ 에 대해서는  $1^\circ$ 이고,  $\rho$ 에 대해서는 10cm이다. 그림상의 세 봉우리는 추출된 기하학적 특징 선을 나타내는데,  $[\rho, \theta]$  형태로 나타낸다면, 첫번째 특징 선은  $[0, 3]$

표 1. 위치추정알고리즘의 상태 집합.

Table 1. Exclusive sets of the Localization statuses.

States	Decision parameters
Static vs. Dynamic	Corruption ratio of measured sensor data.
Polygonal vs. Non-polygonal	The result of Hough transform (voting number of candidate feature).
Global vs. Local	Sample distribution and multiple guess possibility.
Reliable vs. Not Reliable	Matching ratio of measured sensor data whose errors are less than 10%.

(득표수=154), 두 번째 선은  $[1, -17]$ (득표수=105), 세 번째 선은  $[91, 7]$ (득표수=38) 임을 알 수 있다.

전체 알고리즘의 설명과 더 자세한 실험 결과는 [13]에 소개되어 있다.

3.2. 통합 위치 추정 방법론

실제 환경에서 위치 추정 알고리즘이 알아낸 추가적인 상태 정보들은 로봇 행동 결정에 매우 유용하게 사용될 수 있다. 본 알고리즘에서는 이산 이벤트 개념을 적용하여 다양한 상황을 모델링하고 있다. 표 1은 본 알고리즘에서 사용되는 상태 집합을 보여준다.

본 위치 추정 시스템은 로봇 상태 분석을 통해 Map matching 방법을 선택하거나 로봇 행동을 조인하는 역할을 체계적으로 수행할 수 있도록, 페트리넷 기반의 프레임워크를 제공하고 있다.

본 논문에서는 지면상 제한으로 인해 “Global vs. Local problem”에 대한 상태 집합에 대해서만 설명한다. 표 1에 제시된 모든 상태 집합에 대한 설명과 페트리넷 모델은 참고문헌[14][15]에 설명되어 있다.

그림 8은 대역 위치추정(Global positioning)에서 지역 위치추정(Local tracking)으로 전이되는 것을 보여주는 실험 결과이다. 그 중 대표적인 네 순간의 샘플 분포와 측정 센서 값을 보여준다.

그림 8.(a)에서는 로봇의 초기 위치가 주어지지 않은 상태이므로, 1000개의 샘플이 전체 영역에 고르게 분포되어 있다. 실제 로봇 위치는  $[x, y, \theta] = [3.9, 14.5, -1^\circ]$ 이고, 이 상황에서 최대 확률을 갖는 위치는  $[4.44, 12.92, 11.25^\circ]$ . 따라서, 우측 그림에서 보듯이, 실제 센서 값은 지도와 전혀 일치하지 않고 있으며, 샘플의 분산  $v(s)$ 는 91494439이고, 신뢰도는 12%이다. 즉, 이 상태에서 로봇은 현재 위치를 모르고 있으며, 따라서 환경 정보를 더 얻기 위해 무작위 운동을 한다.

그림 8.(b)는 어느 정도 주행을 한 이후에 데이터이다. 최대 확률 값을 가지는 위치는  $[4.01, 16.45, -3.5^\circ]$ 이고, 실제 위치인  $[4.02, 16.51, -3.1^\circ]$ 에 많이 근접한 상태이다. 따라서 우측 그림의 레이저 측정값도 지도와 상당히 일치하고 있고, 신뢰도는 89%이다. 비록 위치 추정은 성공적이라고 할 수 있으나, 샘플의 분포는 아직도 수렴되지 않은 상태이다. 분산  $v(s)$ 는 40701845이고, 수렴 성공을 나타내는 기준 값 30000000에는 아직 미치지 못한다. 대역 위치 추정 중에는 유사하게 생긴 장소가 여러 곳 존재 가능하므로, 이 상태에

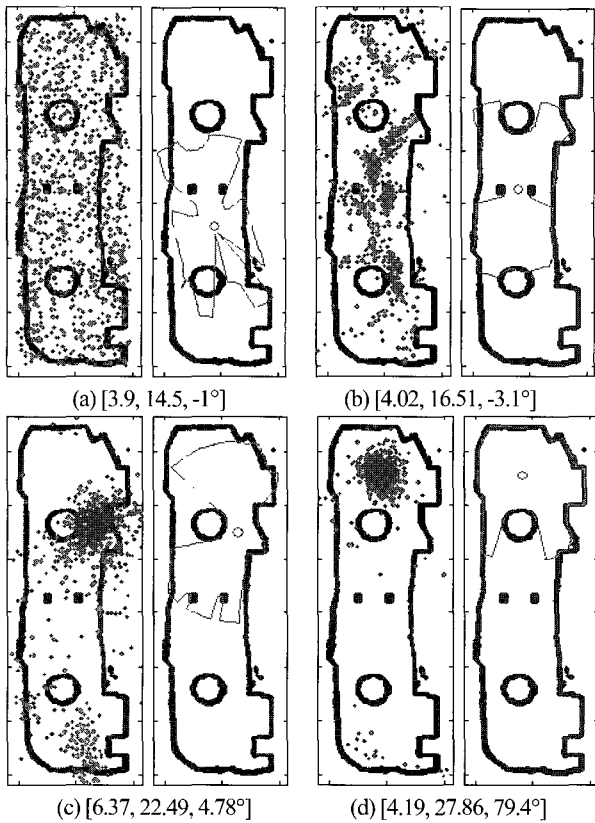


그림 8. 대역 위치추정 결과.  
Fig. 8. Samples and Sensor Data During Global Positioning.

서는 아직 로봇 위치를 갱신하지는 않는다. 그러므로, 아직 로봇은 자신의 위치를 모르는 상태로 인식하여, 계속 무작위 운동을 한다. 이 상태는 (샘플 분포, 신뢰도, 추정된 위치값 갱신 여부, 로봇 운동)로 나타낼 때, (global positioning, reliable localization result, robot position is unknown, random motion)로이다. 이 상태 분포는 페트리넷을 통해 일목요연하게 표현할 수 있다[14][15].

그림 8.(c)는 로봇이 [6.37, 22.55, 4.92°]에 위치해 있고, 샘플도 어느 정도 수집된 상태이다. 예측 위치는 [6.37, 22.49, 4.78°]이고,  $v(s)$ 는 29988312, 신뢰도는 74%이다. 이 상태에서 로봇은 현재 자신의 위치를 알고 있다고 가정한다.

그림 8.(d)에서는 로봇의 실제 위치는 [4.19, 27.86, 79.4°]이고, 거리계(Odometry)기반의 예상 위치는 [4.93, 27.77, 67.28°]이다. 이 둘 사이의 차이는 거리계의 누적 오차 때문이다. 위치 추정 알고리즘이 정확히 현재 위치를 가리키고 있으므로, 이 값으로 갱신한다면 데드레컨(Dead reckoning)에 의한 위치 오차는 해소될 수 있다. 이 때의  $v(s)$ 는 1438415, 신뢰도는 99%이다. 이 상태는 (local tracking, reliable localization result, robot position is known form localizer, planned path tracking)이다.

4. 자율 지도작성 알고리즘

본 시스템의 자율 지도작성 알고리즘은 다음 세 단계로 이루어진다.

1) 환경 정보 수집 : 로봇이 전혀 모르는 새로운 환경에 투입되면, 전체 환경을 이동하면서 정보를 수집하게 된다.

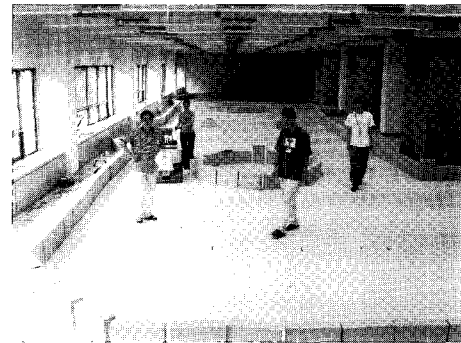


그림 9. 자율 지도 작성 실험을 수행한 동적인 환경.  
Fig. 9. A Dynamic Experimental Environment.

만일, 거리계만으로도 정확한 위치 추정이 가능하다면, 데드레컨 값과 저장된 센서 값만으로도 정확한 지도 작성이 가능할 것이다. 하지만, 데드레컨으로는 오차 누적을 피할 수 없으므로, 위치 보정 작업이 수행되어야 한다.

2) 레이저 거리 센서 값 등록 : 환경을 돌아다니면서 얻은 센서 측정 값들은 앞 절에 소개된 RISF와 ASF를 사용한 위치 추정 방법을 통해 연결된다. 기존의 연구에서는 이웃하는 순간의 두 측정값만을 사용하는데(즉, 현재 측정값( $s_t$ )과 그 이전 측정값( $s_{t-1}$ )), 이 방법의 경우, 한 번 잘못된 위치가 등록되면 다음 측정에도 영향을 끼치게 되므로 정확도가 떨어지게 된다. 반면, 제시된 방법에서는, 하나의 측정값( $s_t$ )은 그 이전의 모든 측정값( $s_1, \dots, s_{t-1}$ )에 대해 비교를 하므로 더 높은 정확도를 얻을 수 있다.

3) 격자 지도 및 특징 지도 작성 : 등록 단계가 끝나고 나면 Histogramic In-Motion Mapping (HIMM) method[12]를 통해 격자 지도가 만들어진다. HIMM은 어떤 종류의 거리 센서에 대해서도 사용 가능하고, 움직이는 물체로 인한 오염된 센서 값을 제거할 수 있다는 장점이 있다.

HIMM 방법에서는 각 격자에 대한 확실도를 0에서 15의 값으로 제한을 두었다. 그런데, 이와 같이 한다면, 최근의 데이터만 고려되고 이전의 값들은 무시되는 경향을 보이게 된다. 따라서 본 지도 작성 방법에서는 이에 대한 제한을 두지 않고, 모두 다 누적한 이후에 기준값을 적용하는 방식으로 격자 지도를 얻는다.

본 지도 작성은 실시간으로 진행되지 않는다. 그 이유는 빠르게 지도를 만드는 것 보다 더 정확한 지도를 얻는 것이 더 중요하기 때문이다.

그림 9와 10은 제안된 자율 지도 작성 방법의 실험 결과를 보여준다. 실험 공간의 크기는 10m x 35m이다. 결과에서 보듯이, 제안된 방법은 동적이고 임의의 모양을 가진 환경에 대해서도 적용이 가능하다. 그림 9에서 보듯이, 다수의 사람이 공존하는 환경에서 로봇은 지도 작성 작업을 수행하였다.

그림 10.(a)는 위치 수정을 하지 않은 상태에서 레이저 측정값을 연결한 결과이다. 거리계에 의한 오차와 임의로 움직이는 사람들로 인해 정확한 지도 작성이 불가능하다. 그림 10.(b)에서 보듯이 등록 알고리즘을 거친 이후에는 Odometry에 의한 오차가 많이 사라졌다. 그 이후에 HIMM

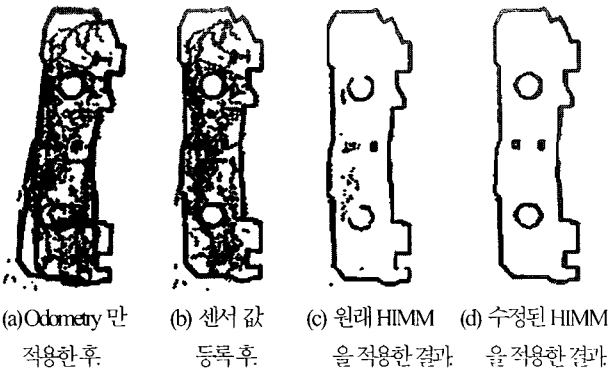


그림 10. 자율 지도 작성 결과.  
Fig. 10. Autonomous Map Building.

방법을 사용함으로써 움직이는 물체에 의한 오차를 제거할 수 있다. 하지만, 그림 10.(c)와 10.(d)에서 보듯이, 수정된 HIMM 방법을 사용하는 것이 더 나은 성능을 얻을 수 있다.

5. 경로 계획

본 주행 시스템에서는 대역 경로 계획과 지역 경로 계획이 독립적으로 구현되었다. 앞서 설명했다시피, 대역 경로 계획은 위상 지도 상에서 최적 노드 집합을 얻는 것이고, 지역 계획은 노드와 노드 사이의 이동을 장애물 회피를 보장하며 최적 경로로 주행하는 것이다. 본 경로 계획 알고리즘은 Konolige의 그레디언트 방법(gradient method)[17]를 수정하여 적용 되었다.

대역 경로 계획은 지역 경로 계획의 간략화된 형태이므로, 본 절에서는 지역 경로 계획에 대해서 주로 설명한다. 그림 11은 구현된 경로 계획 알고리즘에 대해 보여준다. 지역 경로 계획기는 로봇이 목표 지점에 도달할 때까지 이 알고리즘을 반복 수행한다.

첫 번째 단계는 경로 계획에 필요한 모든 정보를 수집하는 것이다. 이는 현재 로봇의 위치, 목표점 집합, 장애물 집합 등을 포함한다. 본 알고리즘은 목표점의 개수에는 제한 없이 적용가능하지만, 대다수의 경우에는 목표 집합은 한 개의 원소로 이루어진다. 장애물 집합은 활성화 지도로부터 얻을 수 있다. 본 알고리즘에서는 로봇을 하나의 점으로 가정하므로, 장애물 집합은 로봇의 반지름만큼 확대된다.

두 번째 단계는 주행 함수(Navigation Function)를 계산하는 것이다. 주행 함수(Navigation Function)은 활성화 지도상의 각 점에 최단 경로 비용(Path cost)을 지정한 함수이다. 국소 최소점들(Local Minima)이 존재하지 않는 최적 경로는 이 주행 함수(Navigation Function)의 최대 감소 방향으로 이동함으로써 얻을 수 있다. 주행 함수(Navigation Function)에 대한 자세한 설명은 참고문헌[4][17]에 자세히 제시되어 있다.

세 번째 단계는 오류 상태에 빠지지 않았는지 확인하는 작업이다. 제시된 알고리즘에서는 장애물에 의한 목표점 점유, 경로 폐쇄, 이 두 가지의 예외 상황을 조사한다. 이 두 상황은 실제 사람과 공존하는 환경에서 빈번히 발생하는 오류이다.

경로 폐쇄는 두 인접한 노드 점 사이에 경로가 존재하지 않는 것을 의미한다. 만일, 로봇이 이와 같은 상황에 빠지게

**Step 1: Gathering all necessary input data**  
 (1) Obtaining the current position of a robot  
 (2) Setting target node point as a goal set.  
 (3) Generating an obstacle set from the active map.

**Step 2: Building the navigation function.**  
 (1) Calculating intrinsic cost.  
 (2) Calculating adjacency cost.

**Step 3: Checking error states**  
 (1) Checking whether the original goal is restored.  
     (only if the original goal has been changed.)  
 (2) Checking whether the goal is occupied.  
 (3) Checking whether no path to the goal exists.

**Step 4: Generating path set (only if one or more of path updating rules are satisfied.)**  
 (1) Constructing a path set by steepest descent method.  
 (2) Smoothing the path using Bezier interpolation.  
 (3) Performing velocity planning.

그림 11. 지역 경로계획 알고리즘.  
Fig. 11. Local Path Planning Algorithm.

되면, 이동하는 것을 멈추고 새로운 경로가 생성될 때까지 기다린다. 장시간동안 이 상태가 지속되면, 경로 계획기는 목표점의 도달을 포기한다. 이를 통해 위상 지도는 갱신이 되고 다시 대역 경로 계획을 수행하게 된다.

목표점 점유는 목표점의 Navigation function 상의 장애물에 의한 비용이 0보다 큰 값을 가질 때 발생한다. 이런 상황이 발생하면 임시 목표점을 지정하고, 로봇은 이 점으로 이동한다. 이때, 경로 계획기는 계속적으로 원래 목표점의 장애물이 사라졌는지를 검사한다. 만약, 임시 목표점에 도착하고 오랜 시간 동안 원래 목표점이 계속 점유된 상태라면, 경로 계획기는 임무 실패를 알린다. 이 후 대역 경로 계획기에서는 다음 목표점을 바로 갈 수 있으면 다음 프로세스를 수행하고, 그렇지 않으면 새로운 목표점을 지정한다.

마지막 단계는 Navigation function 상에서 최적 경로를 생성하는 작업이다. 계획된 경로는 활성화 지도 상에 연속된 점들의 집합으로 표시된다. 경로의 갱신을 자주 하는 것은 경로 추종 제어 성능의 저하를 가져오므로, 바람직하지 않다. 따라서 경로의 갱신은 다음과 같은 갱신 법칙을 만족할 때만 이루어진다.

- 1) 최초 경로 계산 : 현재 수행하는 이동 작업에 대한 경로가 이전에 한 번도 생성된 적이 없는 경우를 말한다.
- 2) 목표 지점의 변경 : 이 경우는 임시 목표점이 지정됐을 때, 혹은 원래 목표점이 복구되었을 때, 이 두 가지로 나뉜다.
- 3) 기존 경로 위에 장애물 위치 : 기존 경로가 부분적으로 장애물에 점유됨을 의미한다.
- 4) 새로 계산한 경로 집합의 원소 수가 기존 것보다 많이 줄었을 경우 : 이는 목표 지점까지의 더 짧은 경로가 발견되었음을 의미한다.



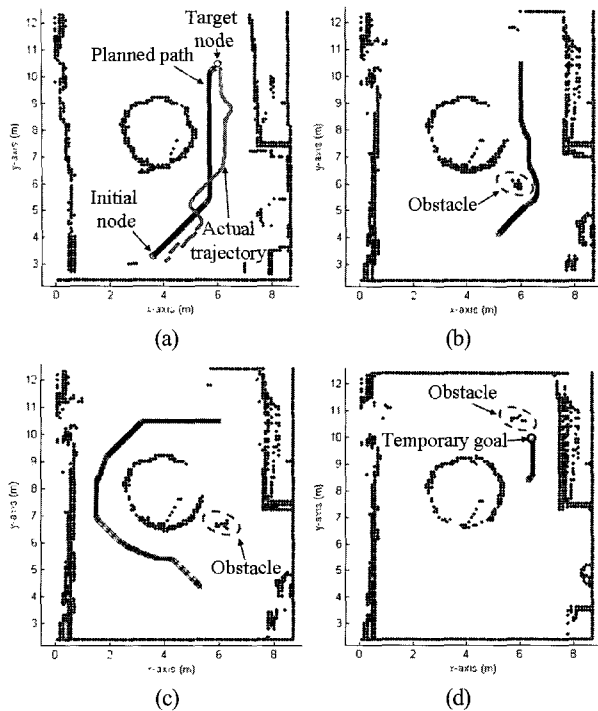


그림 12. 지역 경로계획 결과.  
Fig. 12. The Results of Path Planning.

최적 경로는 Navigation function에 Steepest descent method를 적용함으로써 얻을 수 있다. 이후 Bezier interpolation을 이용하여 경로를 부드럽게 하는 작업이 수행된다. 이 방법은 상대적으로 간단하면서도 경로의 연속성을 잃지 않는 장점을 가진다. 최종 경로 집합은 로봇의 이동 속도에 대한 정보도 포함하고 있다. 경로 집합 상의 연속된 두 원소 사이의 거리는 목표 속도와 비례해서 주어진다.

그림 11에 제시된 전체 경로 계획 알고리즘을 수행하는 데에 PSR-1의 제어 컴퓨터(1.2 GHz P-III PC)에서 평균 약 0.6초 정도 걸린다. 이는 PSR이 실제 서비스 작업을 수행할 때 측정된 결과이다. 즉, 위치 추정과 같은 복잡한 알고리즘이 모두 활성화된 상태에서 얻은 결과이므로, 구현된 경로 계획 알고리즘이 실제 로봇에서도 적용이 가능함을 보일 수 있었다.

그림 12는 구현된 지역 경로 계획 알고리즘의 실험 결과를 보여준다. 이 환경은 앞 절에서 위치 추정 실험을 한 장소와 동일한 곳이다. 주행의 시작점은 (3.6 m, 3.3 m)이고, 목표점은 (6.0 m, 10.5 m)이다. 각 그림에는 활성화 지도와 계획된 경로가 차례로 보이고, 그림 12.(a)의 경우에는 실제 로봇의 이동 경로가 표시되어 있다. 그림에서 보듯이 장애물의 위치에 따라 최적 경로가 변화됨을 알 수 있다. 그림 12.(c)와 같이 이동하는 장애물이 좁은 복도를 막게 되면, 로봇은 멀리 돌아가는 경로를 계획한다. 그림 12.(d)에서는 이동하는 장애물에 의해 원래 목표점이 점유되고, 임시 목표점이 설정되는 것을 보여준다. 그림 12.(a)의 로봇 실제 이동 경로를 보면, 여러 번의 불연속점이 존재하는데, 이는 위치 추정기에 의한 로봇 위치 갱신에 의한 것이다. 이런 불연속

점에 불구하고, 로봇은 부드럽게 이동이 가능하도록 구현되었다.

6. 추종 행위(Behavior)

이 행위 모듈은 지역 경로계획기로부터 온 경로 집합을 부드럽게 추종하는 운동을 만들어낸다. 즉, 한 샘플링타임에 경로 집합의 한 점씩을 차례로 목표점으로 하여 이동한다.

구현된 행위 모듈에는 실제 적용에 필요한 여러 추가적인 로직이 포함되어 있다. 예를 들어, 로봇은 원 센서값을 이용해서 위급 상황시 급히 정지가 가능하다. 이는 빠르게 움직이는 장애물로부터 안전성을 확보하기 위함이다.

7. 프로세스 PrAutoMove에 대한 Configuration 설계

본 절에서 다룰 컨피규레이션이란, 여러 모듈들을 어떻게 효과적으로 통합하여 주행 프로세스를 성공적으로 수행하는지에 관련된 모든 사항을 정의한다. 본 시스템의 가장 기본적인 주행 전략은 프로세스 PrAutoMove의 형태로 구현되었다. PrAutoMove가 주어지면, 프로세스 관리자는 지역 격자 지도, 위치 추정기, 경로 계획기, 추종 행위 모듈 등을 활성화시킨다. 이들은 지능을 갖춘 컴포넌트들로써, 자신에게 맡겨진 고유의 기능 뿐만이 아니라, 내부 혹은 외부 환경 상태 분석이 가능하다. 이미 살펴본 바와 같이, 경로 계획기의 경우에는 경로 생성 뿐만이 아니라 목표점 점유, 경로 폐쇄 등의 환경 정보를 이벤트의 형태로 제공한다. 이와 같은 이벤트들은 로봇 행동 결정에 중요한 정보가 된다.

그러나, 이벤트를 제공하는 컴포넌트가 늘어남에 따라, 이에 대한 처리를 효과적으로 할 필요성이 대두되는데, 이는 단순히 If-then 로직만으로는 해결이 힘들다. 따라서, 본 연구에서는 페트리넷 기반의 프레임워크를 개발하였다. 제안된 방법을 통해, 작업 분석, 오류 감지/복구, 여러 모듈로부터 오는 비동기적 이벤트 처리 등의 여러 주요 문제들을 한번에 해결할 수 있다.

그림 13과 표 2는 PrAutoMove에 대한 페트리넷 모델을 보여준다. 표 2의 트랜지션들은 위치 추정기, 경로 계획기, 행위 모듈 등에서 오는 여러 이벤트를 표현하고 있다. 예를 들어,  $t_1, t_2, t_3, t_4, t_8$ 은 경로계획기로부터,  $t_{10}$ 은 행위 모듈로부터 온다. PrAutoMove가 시작하면,  $t_1$ 이 발생하고, 이로 인해 토큰들이  $P_1, P_2, P_{10}$ 에 분포된다. 이 세 토큰들은 각각 위치 추정기, 경로 계획기, 행위 모듈의 내부 상태를 명시적으로 표시한다. 만약, 이 컴포넌트에서 이벤트가 발생하면, 토큰의 분포는 바뀌게 된다. 이는 로봇의 내부 상태가 변화했음을 의미하고, 로봇 거동에 영향을 줄 수 있다. 만약, 로봇이 오류 상태에 빠지면, 복구 과정이 진행된다. 복구 과정에도 불구하고, 이 오류가 해결되지 않으면, 토큰은 실패를 나타내는 플레이스로 이동하고, 프로세스 관리자는 주어진 프로세스의 실행을 종료하고 실패했음을 보고한다. 이는 중앙 작업 계획기에서 이미 만들어진 작업 계획 프로세스 목록을 다시 생성하도록 한다. 예를 들어, 지역 경로 생성기가 경로 폐쇄 오류가 발생하면 ( $t_3$  발생), 로봇은 움직임을 멈추고 새 경로가 생성되기를 기다린다(토큰이  $P_4$ 로 이동). 만약, 경로 계획기가 새로운 경로를 발견하면( $t_4$  발생), 로봇은 정상 동작을 수행 하고,

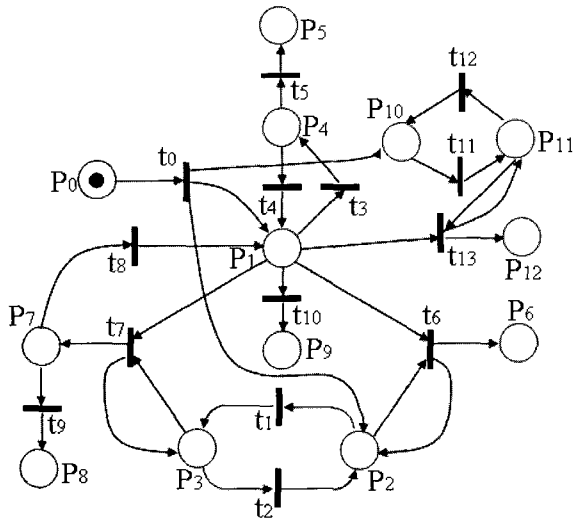


그림 13. 프로세스 PrAutoMove에 대한 페트리넷 모델.  
Fig. 13. Petri net Model of the Process PrAutomove.

그렇지 않으면, 경로 폐쇄로 인한 프로세스 실패를 선언한다. 제안된 주행 시스템에서는, 페트리넷이 컨피규레이션의 설계의 기본적인 형식론(Formalism)으로 사용된다. 컨피규레이션 상의 거동은 여러 이산적인 상태들로 나눌 수 있고 이벤트에 의한 동작으로 표현될 수 있으므로, 페트리넷 사용은 타당하다. 또한, 페트리넷은 여러 단계의 추상화를 표현하는데 유용하다. 페트리넷은 시각적인, 그리고 수학적인 관점에서 큰 장점을 가진다. 시각적인 관점에서, 페트리넷은 제어 로직을 시각화 할 수 있고 시스템의 상태를 보다 표현하기 쉽게 한다. 더욱이 토큰을 사용함으로써 시스템의 동적인 거동에 대한 기술이 가능하다. 수학적 관점에서는, 설계된 로직에 대한 수학적 모델을 세울 수 있고, 이에 대한 분석이 가능하다는 장점을 가진다. 그 외에, 다양한 소프트웨어 도구가 존재한다는 것도 큰 가치를 지닌다. 현재 시각적 편집이 가능하고, 시뮬레이션 및 분석이 가능한 다양한 무료/상용 도구를 쉽게 구할 수 있다. 페트리넷에 대한 자세한 소개는 [18][19]와 같은 여러 문헌을 통해 얻을 수 있다.

IV. 네 가지 목표작업을 통한 평가

물건 이송, 바닥 청소, 안내, 순찰 등 네 가지 목표작업이 PSR 플랫폼을 통해 성공적으로 구현되었다. 이 목표를 수행하는데 있어, 개발된 주행 시스템은 핵심적인 역할을 했다. 본 장에서는 개발된 주행 시스템이 구체적으로 어떤 형태로 서비스 작업에 사용되었는지 소개하도록 한다.

1. 물건 이송 작업

물건 이송 작업은 주행 및 조작 기술의 신뢰도를 보여 줄 수 있는 대표적 작업 중 하나이다. 그림 14는 구현된 이송 작업의 시나리오를 보여준다. 그림 15는 실제 실험이 수행된 환경과 전체 격자 지도를 나타낸다. 80m x 35m 크기의 일상적인 오피스 환경이다. 목표 작업은 문서함을 1번 방에서 3번 방으로 옮기는 것이다. 여러 대의 트레일러를 끌고 있는 PSR-1은 초기에 자신의 위치를 알지 못하므로 대

표 2. 그림 13의 페트리넷 모델 설명.

Table 2. Description of Places and Transitions.

PL/TR	Description
P <sub>0</sub>	Standby.
P <sub>1</sub> (P <sub>6</sub> )	Executing (Completing) the behavior BhAutoMove.
P <sub>3</sub> (P <sub>2</sub> )	State: The original goal has (NOT) been changed.
P <sub>4</sub>	Error: The path to the goal does not exist. (The robot stops moving and waits until a new path is detected.)
P <sub>5</sub>	Fault: The path to the goal does not exist.
P <sub>7</sub>	Error: Reach the temporary goal and stop moving until the original goal is restored.
P <sub>8</sub>	Fault: Reaching the original goal is failed since it is occupied by an obstacle.
P <sub>9</sub>	Error: The behavior BhAutoMove CANNOT extract the desired point from path set. (The behavior reaches the last point in the path set.)
P <sub>10</sub>	State: Normal localization. (Localizer updates a robot position periodically.)
P <sub>11</sub>	State: Abnormal localization. (Localizer doesn't update a robot position, and robot navigates only by using odometry.)
P <sub>12</sub>	Fault: A robot fails to find out its own position.
t <sub>0</sub> (t <sub>6</sub> )	Process supervisor (PS) starts (completes) the behavior BhAutoMove.
t <sub>1</sub>	Local Path planner (LPP) changes the original goal due to the goal occupation.
t <sub>2</sub>	LPP restores the original goal position since the obstacle is removed.
t <sub>3</sub>	LPP detects the error "no path to the goal."
t <sub>4</sub>	LPP finds out a path to the goal.
t <sub>5</sub> (t <sub>13</sub> )	PS terminates the behavior BhAutoMove due to "No path to goal exists." (due to "The failure of localization.")
t <sub>7</sub>	PS reaches the changed goal and stop moving.
t <sub>8</sub>	LPP detects a new goal which can be the original goal or another temporary goal.
t <sub>9</sub>	PS terminates the behavior BhAutoMove due to the failure of reaching the original goal.
t <sub>10</sub>	The behavior BhAutoMove CANNOT extract the desired point from the path set.
t <sub>11</sub> (t <sub>12</sub> )	Localizer finds out the estimated position is (NOT) accord with the actual robot position.

역 위치추정 프로세스를 수행한다. 성공적으로 위치 추정을 수행하면, PSR-1은 1번 방으로 이동하게 된다. 앞서 설명하였듯이, 1번 노드부터 10번 노드까지는 PrAutomove를 반복함으로써 이동한다. 10번 노드에 도착하면, PSR-1은 트레일러를 풀고, 혼자 방 안으로 들어가 문서함이 놓여있는 12번 노드로 이동한다. 이 지점에서, 탁자에 놓인 문서함을 집는 프로세스를 수행하고, 완료 후 다시 트레일러 놓은 곳으로 가서 도킹을 하고, 목표 방 앞까지 간다. 마찬가지로, 이 방 앞에서 다시 트레일러를 풀고, 21번 노드로 이동하여 문서함을 놓고, 다시 나와 트레일러와 도킹한 후 대기 상태로 복귀하게 된다.

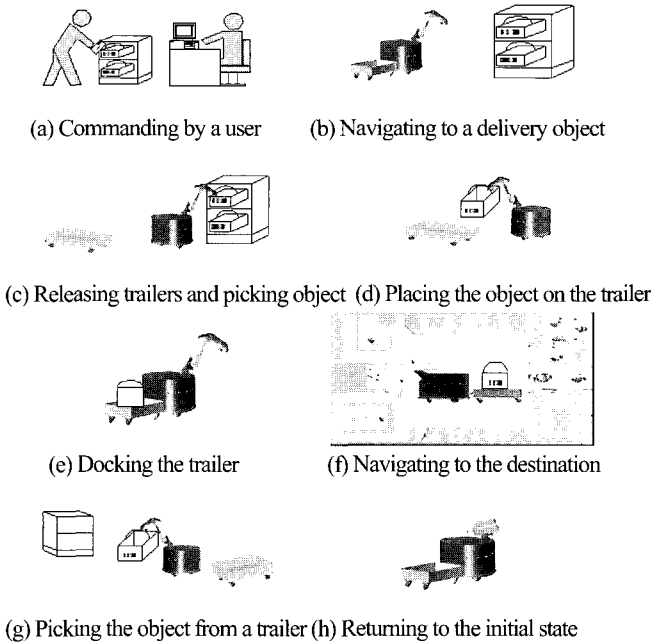


그림 14. 물건 이송 작업의 시나리오.

Fig. 14. Scenario of a Delivery Task.

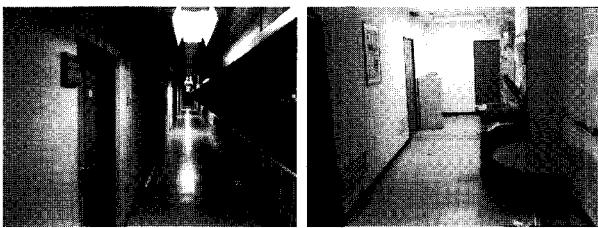
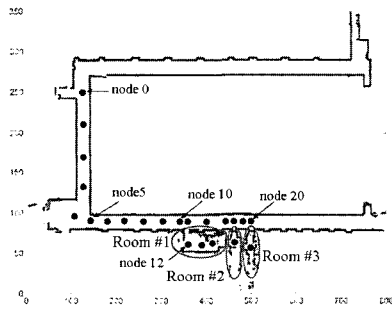


그림 15. 물건 이송 작업 환경.

Fig. 15. Experimental Environment for a Delivery Task.

수행된 물건 이송 작업은 상당히 복잡하고 여러 예외적인 상황이 발생하는 작업이지만, 주어진 시나리오에 따라 성공적으로 수행하였다. 특히, 주행에 대해서는 실제 환경의 불확실성에 대해서도 문제없이 동작함을 보였다. 사용자가 고의로 PSR의 동작을 방해하는 행동을 여러 차례 했음에도 PSR은 충돌 없이 주행하였다. 최대 주행 속도는 0.4 m/s이며, 정상 주행 속도는 0.3m/s이었다. 이는 이후에 설명될 나머지 작업에 대해서도 동일하게 적용되었다.

2. 바닥 청소 작업

청소 작업은 다음의 순서로 진행된다. 초기에, 청소 해야

할 작업 영역에 대한 격자 지도가 주어진다. 이후, 전체 영역은 몇 개의 세부 구역으로 나누어진다. 전체를 한 번에 청소하는 것보다 여러 구역으로 나누어서 하는 것이 위치 추정 정확도에 덜 민감하기 때문에 더욱 효율적이다. 이 때, 구역을 나누는 기준은 고립된 장애물의 유무로써, 고립된 장애물이란 벽과 연결되어 있지 않은 장애물을 말한다. 이 장애물이 위치하는 곳에는 가상의 벽을 설정하여 별도의 구역으로 나눈다. 각 구역은 Full-coverage 알고리즘을 통해 청소가 되고, 한 구역을 완료하면 아직 청소하지 않은 다른 구역으로 이동하게 된다. 이 두 프로세스를 차례로 반복하여 전체 영역에 대한 청소를 마치게 된다.

개발된 Full-coverage 알고리즘은 Wall-following 알고리즘을 응용하여 개발되었다. 즉, 주어진 영역의 벽을 따라 이동하면서 한 바퀴 다 돌아 처음 위치로 돌아오면 벽과의 거리를 증가시킨다. 이 과정은 주어진 영역을 완전히 뒤덮을 때까지 계속된다. 개발된 알고리즘은 회전 수와 전체 이동 거리를 최소화할 수 있도록 설계되었다. 로봇이 너무 자주 회전을 하면 그만큼 가감속 구간이 길어져 청소 시간이 증가하게 되고, 이동 거리는 에너지 효율 관점에서 최소화되어야 한다. 시뮬레이션을 통해, 개발된 알고리즘이 기존의 Distance transform이나 Plowing method[20] 보다 효율적임이 증명되었다. 또 다른 장점으로, 원 센서 정보를 직접 이용하므로, 위치 추정의 정확도에 덜 민감하다는 것이 있다.

한쪽 벽면을 따라 벽따라가기(Wall-following)를 하는 도중, 반대 편에 장애물이나 새로운 벽이 감지되는 문제를 극복하기 위해 다음과 같은 방법을 사용한다. 본 알고리즘에 대해서는 왼쪽 벽따라가기(Wall-following)를 할지 오른쪽 벽따라가기(Wall-following)를 할지 사용자가 초기에 선택할 수 있다. 일단, 한쪽에 대해서 결정되면 그 반대편의 일정 각도에 대해서는 레이저 값을 아예 사용하지 않는다. 예를 들어, 오른쪽 벽따라가기(Wall-following)의 경우 로봇 앞 180도 영역 중에 왼쪽 60도 정도 각도의 레이저 값은 무시된다.

청소 작업은 PSR-2를 통해 구현되고 시험되었다. 그림 16에서 보듯이, 전체 작업 환경은 가상의 벽에 의해 네 부분으로 나뉜다. 따라서, 네 번의 청소 프로세스와 중간에 적절히 PrAutoMove를 사용함으로써 주어진 작업을 완수할 수 있다. C구역에 대한 청소 결과가 아울러 제시되어 있다. 그림 17은 PSR-2가 청소 작업을 수행하는 모습을 보여준다.

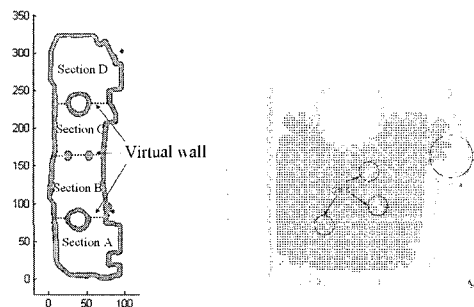


그림 16. 청소 작업 실험 환경 및 결과.

Fig. 16. The Map of a Workspace and the Result of Coverage of Section C.

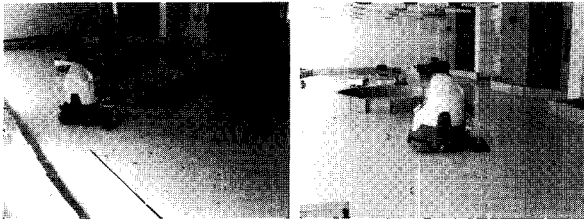


그림 17. 청소 작업을 수행하는 PSR-2.  
Fig. 17. PSR-2 in a Cleaning Experiment.

청소 작업의 특성상, 사람이 많이 이동하는 낮 시간에 수행하는 것보다는 밤시간에 하는 것이 유리하다. 따라서 본 알고리즘에서는 환경이 정적인 것으로 가정하였다. 또한, Full-coverage를 위해서는 피하는 동작보다 멈추고 기다리는 것이 훨씬 효율이 높으므로, 청소 작업 도중 사람을 만나는 경우에는 그 사람이 비켜설 때까지 제자리에서 기다린다.

### 3. 안내 작업

안내 작업은 작업의 특성상, 사람에게 친화적인 외관이 중요하므로, 지니 플랫폼을 통해 구현되었다. 지니의 경우, 다양한 HRI 기능이 존재하지만, 본 논문에서는 지니의 주행 능력에 초점을 맞추어 기술한다. 지니의 경우, 항상 사람이 분비는 공간에서 원하는 전시물로 이동하는 동작을 해야 하므로, 신뢰도를 높이기 위해 적응적 주행 능력이 필요하다. 즉, 로봇은 상황에 맞게 자신의 주행 알고리즘을 선택적으로 사용한다.

현재 지니에는 네 가지 동작 패턴이 구현되어 있다. 이와 같은 선택적 주행 능력 구현을 위해, 페트리넷 기반의 정규 프레임워크의 도움이 크게 작용하였다.

1) PrAutoMove: PrAutoMove의 장점은 일반성과 최적성에 있다. 이는 어떤 상황에서든 원하는 목적지까지의 최단 거리 이동이 가능하다. 이 방법은 주로 지니가 넓은 공간을 주행해야 할 때 사용된다.

2) 경로 갱신 없는 PrAutoMove : 일반적으로, PrAutoMove의 장애물을 회피하는 동작은 관람객에게 거부감을 줄 수 있다. 사용자가 로봇에 다가가는 이유는 로봇과의 인터랙션을 위한 경우가 많으므로, 피하는 동작보다 멈춰 서서 기다리는 동작이 더욱 인간 친화적이다. 따라서, 이 주행 방법은 시작 시점에 한 번만 경로 계획을 하고, 경로 갱신 없이 주행하다 장애물을 만나면 멈추고 비켜달라는 말을 한다. 이후에, 장애물이 비켜 서면 다시 이동하는 동작 패턴을 가진다. 이 동작이 적용되는 전형적인 예는 다수의 관람객과 함께 이동하는 경우이다.

3) Wall following : 이 운동 알고리즘은 청소 작업을 위해 개발된 Full-coverage 알고리즘을 기반으로 개발되었다. 이 패턴의 장점은 순수하게 센서 정보만을 이용한 운동이므로, 위치 추정 정확도에 덜 민감하다는 것이다. 따라서, 전시물 사이의 좁은 통로를 이동해야 하는 경우 등에 적용된다.

4) 원격 제어 운동 : 장시간 관람객에 의해 둘러싸여 있거나 악의적인 관람객에 의해 주행이 곤란한 경우에 사용한다. 즉, 안내 데스크의 원격 컴퓨터의 조이스틱이나 키보드를 이용하여 원격 제어를 한다.



그림 18. 현대 중공업 홍보관의 안내로봇 지니.  
Fig. 18. The Guide Robot Jinny in the Exhibit Hall of Hyundai Heavy Industries.

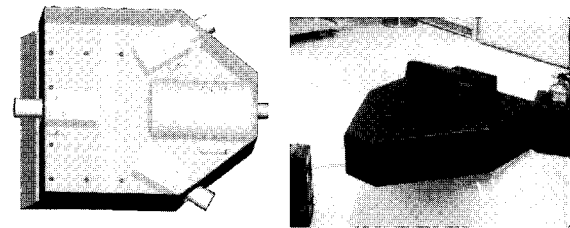


그림 19. 순찰 작업을 위한 카메라 시스템.  
Fig. 19. A Camera System for a Patrol Task.

지니는 그 동안 여러 박람회에 참가하거나 TV 프로그램에 출연하는 등 다양한 환경에서 적용 시험되었다. 그림 18은 현대 중공업 홍보관에서 시험 운용된 지니의 모습을 보여준다. 지니는 많은 사람이 존재하는 홍보관을 주행하며, 25개의 전시물에 대한 안내를 수행하였다. 사용자는 터치 스크린을 통해 관람 시나리오를 선택할 수 있다. 또한, 안내 도중 간단한 게임을 하거나 음악에 맞춰 춤을 추기도 하며, 추가 정보나 날씨 정보 등을 웹으로부터 얻어와 제공하기도 한다.

### 4. 순찰 작업

순찰 작업은 앞의 서비스 작업을 통해 개발된 기술을 그대로 이용함으로써, 별 어려움 없이 개발이 가능하다. 이 작업을 위해 사용된 주행 알고리즘은 안내 작업에서 설명한 PrAutoMove, Wall following, 원격 제어 운동 등이다. 그림 19에서 보듯이, 전방향 감시를 위해 네 대의 웹 카메라와 한 대의 팬/틸트 카메라가 장착되었다. 로봇은 이동하면서 찍은 영상을 원격지의 컴퓨터로 전송할 수 있다.

구현된 순찰 작업은 시나리오상 야간에 수행한다. 그러나, 구현된 주행 기술은 사람과 공존하는 환경에서 충분히 동작 가능하므로, 이에 대한 기술적 제한은 없다.

## VI. 결론

지금까지 본 논문은 다기능 서비스 로봇인 PSR 시스템에 적용된 주행 시스템에 대하여 다루었다. 개발된 시스템은 세대의 플랫폼에 물건 이송, 바닥 청소, 안내, 순찰 등 네 가지 목표 작업에 성공적으로 적용되어, 그 신뢰도와 적용 가능성이 증명되었다.

마지막으로, 본 주행 시스템의 장점은 다음과 같이 요약할 수 있다.

1) 환경 수정이 필요 없는 거리 센서 기반의 일반적인 주행 방법 개발 : 모든 종류의 거리 센서를 적용할 수 있는 일반화된 방법이다. 또한, 주행을 위해 어떠한 인위적 표식은 필요로 하지 않는다.

2) 지능을 갖춘 주행 요소 컴포넌트 : 각 요소 컴포넌트들의 지능을 통해 주행 시스템의 신뢰도를 높였다.

3) 다중 행위의 선택적 적용 및 오류 검지/복구를 지원하는 제어 프레임워크 구축 : 지능을 갖춘 컴포넌트들의 효과적 통합을 지원하는 페트리넷 기반의 일반화된 체계가 제시되어 있다. 이를 이용하여 이벤트 처리, 작업 세분화, 오류 검지/복구 등 여러 중요한 문제들을 동시에 해결할 수 있다.

**참고문헌**

[1] S. Thrun, M. Benezit, W. Burgard, A. B. Cremers, F. Dellaert, and D. Fox, "MINERVA : A Second-Generation Museum Tour-Guide Robot," *Proc. of IEEE Int. Conf. Robotics and Automation*, pp. 1999-2005, 1999.

[2] R. Simmons, R. Goodwin, S. Koenig, J. O'Sullivan, and G. Armstrong, "Xavier: An Autonomous Mobile Robot on the Web," in *Beyond Webcams: An Introduction to Online Robots*, edited by K. Goldberg and R. Siegwart, MIT Press, MA Cambridge, 2002.

[3] B. Graf, M. Hans, and R.D. Schraft, "Care-O-bot II - Development of a Next Generation Robotics Home Assistant," *Autonomous Robots*, vol. 16, pp. 203-222, 2003.

[4] J. C Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, UK, 1996.

[5] M. Khatib, H. Jaouni, R. Chatila, J.P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," *Proc. of IEEE Int. Conf. Robotics and Automation*, pp. 2920-2925, 1997.

[6] R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G.. Terrien, and N. Tomatis, "Robox at Expo.02 : A large-scale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42, no.3-4, pp. 203-222, 2003.

[7] K. Arras, R. Philippsen, N. Tomatis, M. Battista, M. Schilt, and R. Siegwart, "A Navigation Framework for Multiple Mobile Robots and its Application at the Expo.02 Exhibition," *Proc. of the IEEE Int. Conf.*

*Robotics and Automation*, pp. 1992-1999, 2003.

[8] <http://www.deltatau-com/>

[9] W. Chung, K. Kim, J. Lee, T. Lee, B. Kang, M. Kim, and C. Lee, "Design and Control of the Indoor Public Service Robot," *Proc. of the IFAC Workshop on Mobile Robot Technology*, pp. 303-308, 2001.

[10] G.. Kim, W. Chung, M. Kim, and C. Lee, "Tripodal Schematic Design of the Control Architecture for the Service Robot PSR," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2792-2797, 2003.

[11] G.. Kim, W. Chung, M. Kim, and C. Lee, "Implementation of Multi-Functional Service Robots Using Tripodal Schematic Control Architecture," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4005-4010, 2004.

[12] J. Borenstein, and Y. Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Trans. Robotics and Automation*, vol. 7, no.4, pp. 535-539, 1991.

[13] D. Lee, W. Chung, and M. Kim, "A Reliable Position Estimation Method of the Service Robot by Map Matching," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2830-2835, 2003.

[14] D. Lee, W. Chung, and M. Kim, "Integrated Localization of the Service Robot PSR," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4017-4022, 2004.

[15] D. Lee, W. Chung, and M. Kim, "Autonomous map building and smart localization of the service robot PSR," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and systems*, pp. 454-459, 2003.

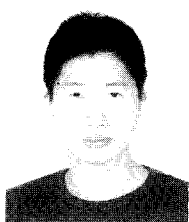
[16] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1322-1328, 1999.

[17] K. Konolige, "A Gradient Method for Realtime Robot Control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and systems*, pp. 639-646, 2000.

[18] T. Murata, "Petri nets: properties, analysis, and application," *Proc. of IEEE*, vol.77, no.4, pp. 542-580, 1989.

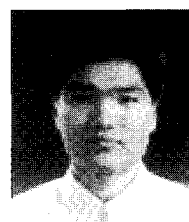
[19] R. Zurawski and M. Zhou, "Petri nets and industrial applications: A Tutorial," *IEEE Trans. Industrial Electronics*, vol. 41, no.6, pp. 567-583, 1994.

[20] J. Oh, J. Park, Y. Choi, "Complete coverage navigation of clean robot based on triangular cell map," *Proc. of IEEE Int. Symp. Industrial Electronics*, pp. 2089-2093, 2001.



**김 건 희**

1978년 2월 6일생. 1999년 한국과학기술원(KAIST) 기계공학과 졸업. 동대학원 석사(2001). 2001년~현재 KIST 연구원. 관심분야는 이동로봇 주행 시스템, 로봇 시스템의 제어 구조, 이산이벤트.



**정 우 진**

1970년 11월 20일생. 1993년 서울대 기계설계학과 졸업. 동경대 기계정보공학과 석사(1995). 동대학원 박사(1998). 1998~현재 KIST 선임연구원. 관심분야는 이동로봇 위치추정 및 주행제어, 논홀로노믹 시스템 제어, 다지 로봇손.

**김 문 상**

1957년 10월 13일생. 1980년 서울대 기계설계학과 졸업. 동대학원 석사(1982). 독일 Berlin 공대 기계공학과 박사(1987). 2003년~현재 과학기술부 21C 프론티어 사업 인간기능 생활지원 지능로봇 기술 개발사업 단장. 관심 분야는 지능 로봇, 햅틱인터페이스.