

격자위상혼합지도방식과 적응제어 알고리즘을 이용한 SLAM 성능 향상

논문

58-8-22

Increasing the SLAM performance by integrating the grid-topology based hybrid map and the adaptive control method

김수현* · 양태규*

(Soo-Hyun Kim · Tae-Kyu Yang)

Abstract - The technique of simultaneous localization and mapping is the most important research topic in mobile robotics. In the process of building a map in its available memory, the robot memorizes environmental information on the plane of grid or topology. Several approaches about this technique have been presented so far, but most of them use mapping technique as either grid-based map or topology-based map. In this paper we propose a frame of solving the SLAM problem of linking map covering, map building, localizing, path finding and obstacle avoiding in an automatic way. Some algorithms integrating grid and topology map are considered and this make the SLAM performance faster and more stable. The proposed scheme uses an occupancy grid map in representing the environment and then formulate topological information in path finding by A* algorithm. The mapping process is shown and the shortest path is decided on grid based map. Then topological information such as direction, distance is calculated on simulator program then transmitted to robot hardware devices. The localization process and the dynamic obstacle avoidance can be accomplished by topological information on grid map. While mapping and moving, pose of the robot is adjusted for correct localization by implementing additional pixel based image layer and tracking some features. A laser range finder and electronic compass systems are implemented on the mobile robot and DC geared motor wheels are individually controlled by the adaptive PD control method. Simulations and experimental results show its performance and efficiency of the proposed scheme are increased.

Key Words : SLAM, hybrid map, adaptive control, A* algorithm

1. 서론

자율주행로봇에 관한 연구는 주로 1980년대 이후부터 제어기술과 인공지능의 개발과 함께 발전해오고 있다. 두 개의 바퀴를 이동수단으로 하고 몸체 위에 컴퓨터와 하드웨어 보드 및 센서시스템을 장착한 주행로봇은 그 구조가 비교적 단순하고, 제어 메커니즘이 용이하여 생활에 필요한 기능을 수행하는 로봇의 개발을 용이하게 하는 등의 장점이 있기 때문에 로봇개발의 초창기부터 연구가 진행되어 왔으며, 현재는 기반기술이 응용되어 청소로봇, 서비스 로봇 등의 모델로 시중에 판매되고 있다.

주행로봇의 주요기술은 크게 지도작성, 자기위치인식, 장애물 감지 및 회피, 경로경유 및 복귀 등으로 나누어지며, 센서시스템에 의한 환경인식과 함께 향후 로봇의 기능 도약을 위한 기본 역할로서 그 중요성이 크다고 할 수 있다. 이중 센서시스템에 의해 환경정보를 받아들이고 이를 공간정보로 변환하여 인지하는 지도작성과정과 작성된 지도정보를 바탕으로 현재의 자기 위치를 인지할 수 있는 자기위치인식

과정은 상호간 연관되어 있는 문제로서 동시에 이를 추정하고 조정하는 과정을 통해 위치 정확성을 향상시키려는 연구분야를 동시적 자기위치인식 및 지도작성(Simultaneous Localization And Mapping)이라 한다[1].

SLAM 기술에서 환경의 복잡성에 강인하면서 광활한 영역에도 적응성이 있도록 하고 동시에 위치 및 경로이동의 정확성을 향상시키기 위해선 다수의 정밀한 센서와 수많은 영역정보를 저장하기 위한 컴퓨터상의 메모리 그리고 이를 주행 중에 신속하게 처리하기 위한 하드웨어 및 소프트웨어 프로그램의 속도가 빠를 것이 요구된다. 하지만, 현실적으로 많은 데이터를 처리하기 위한 컴퓨터의 용량 및 속도는 한계가 있으므로 주로 지도작성과정에서의 효율성 증진을 위해 격자지도(Grid based map)와 위상기하지도(topology map)의 방식이 연구되어 왔다. 한편 자기위치인식(localization)기술은 GPS 나 비콘-수신기(beacon-listener), 경계표식(landmark)의 적외선감지 시스템과 같이 외부적으로 고정된 지점에서 주행로봇의 위치정보를 송수신할 수 있는 방식의 경우 비교적 용이하지만, 외부적인 위치정보의 실시간 주입이 없이 독립적으로 공간상을 이동하는 경우에 그 중요성이 부각된다. 이는 영역검색방법(map coverage pattern), 위치추정(position estimation), 스캔정합(scan matching)을 비롯한 다양한 방법들이 주행추정(odometry)에 의한 오차를 극복하기 위해 이루어졌다.

거리정보를 인지할 수 있는 센서시스템, 즉 예를 들어 레

† 교신저자, 정회원 : 목원대 IT공학과 박사, 국립암센터 연구원
E-mail : dukesoo@mokwon.ac.kr

* 정 회원 : 목원대 지능로봇공학과 교수

접수일자 : 2008년 11월 19일

최종완료 : 2009년 7월 22일

이저, 적외선, 레이저, 스테레오카메라 등을 이용하여 지도를 작성하게 되는데, 일반적으로 측정 대상 영역의 크기, 센서 측정의 정밀도에 따라 구현방법 및 사양은 다양하다. 격자 지도방식은 Moravec, Elfes 와 Borenstein, Koren 에 의해 제안된 방법으로서 주변환경을 등간격의 격자로 표현함으로써 지도를 작성한다. 반면, 위상기지도는 Kuipers, Bynn 과 Mataric 에 의해 제안된 점, 선, 길이, 각도 등의 위상기하정보로 지도를 표현한다[2]. 이 경우 절대적인 기준좌표점이 설정되지 않고, 주변 환경과의 특징적 관계만 나타나고, 격자지도방식에 비하여 경로계획이 간단하고 간결한 반면 복잡한 환경에서 정보가 급격히 증가하고 자기위치인식 과정에서 오차가 커지고 보정이 어려워 정확성을 높이기 어려운 단점이 있다. Thrun 은 이 두 가지 방법을 병행한 혼합(Hybrid)지도 방식을 제시한 바 있다[3][4]. 자기위치인식은 국소 영역에서의 위치인식과 전체영역기반에서의 위치인식으로 구분이 가능한데, 전자의 경우는 대부분 스캔정합으로 후자의 경우에는 확률통계에기반한 미세 시간적 반복수행 방식에 의한다. 전자의 경우 Hough 변환방법, IDC(Iterate Dual Correspondence)방법, Cox 방법, 각도히스토그램(Angle Histogram), 강도히스토그램(Force Histogram) 등의 방법이 사용되었다. 후자의 경우는 대표적으로 Markov 상태추정방식과 Monte Carlo 상태추정방식 그리고 확장된 칼만필터방식(Extended Kalman filtering)이 센서의 노이즈와 오차정보 등을 확률적인 정규분포 데이터로 가정하고 반복적인 갱신과 최적화 추정 과정을 통하여 위치정보의 정확성을 기하는 방법이다[5][6].

본 연구에서는 효율적이고 자동적으로 수행되도록 하기 위한 SLAM 기술을 위해 시뮬레이터를 구성하고 주행로봇이 미지의 영역에 접하였을 때 사용자에게 의한 목표점 설정만으로 지도검색, 지도인식 및 자기위치인식과정을 통합적으로 수행하도록 하였다. 이 시뮬레이터의 특징은 SLAM 과정에서 정보처리성능의 향상과 위치정확성을 위하여 소프트웨어 프로그램 부분과 하드웨어 제어 모듈을 효과적으로 분리하여 분산처리방식을 도입한 점이다. 지도검색 과정에서 모든 영역을 빠짐없이 검색해나가기 위해 가지점(tree-node) 구조의 역추적(backtracking) 알고리즘이 응용되었고, 지도작성과정에서 지도의 데이터형태는 격자지도방식으로 저장하고, 경유점들을 지나기 위한 최단 검색 경로 및 경유점 간의 이동은 위상기하정보로 변환하여 이동하는 격자위상혼합 지도 방식을 사용하였다. 자기위치인식 및 보정은 주행 중에 인공적 경계표식이나 외부에서의 기준점 정보가 주어지지 않으므로 위상기하정보 상에서 방향이 전환되는 점 및 지정된 경유점들에서 각도히스토그램 정보를 이용하고, 차후 위치의보정은 이전단계에서의 오차정보를 반영한 적응 PD 제어(adaptive PD control) 알고리즘으로 위치의 정확성을 높여 SLAM의 성능을 높일 수 있다.

본 연구내용은 다음과 같이 구성되었다. 2장에서는 주행로봇의 운동학과 동역학에 관해 분석하고, 3장에서는 SLAM 기술, 4장에서는 실험 결과와 분석을 수행한 후, 5장에서 마무리 및 향후 내용에 관한 고찰을 언급하였다.

2. 주행로봇 운동학과 동역학 분석

주행로봇은 일반적으로 두 개의 바퀴에 의해 조향 및 이동이 가능하도록 단순하게 설계되어 있다. 전면의 중앙부에는 균형유지 및 조향이 원활하게 수행되도록 볼캐스터 바퀴가 설치되어 있다. 지면과 접촉한 상태에서 이동 및 조향이 되는 주행로봇의 운동과 이동 간의 관계를 아는 것은 효과적인 제어를 위해서 필수적이다. 두 바퀴의 회전각 또는 회전속도, 가속도 등과 이로 인한 로봇의 주행거리, 속도, 가속도 등의 관계를 분석하는 것은 운동학 분석에 해당되며, 이를 토대로 원하는 바퀴의 회전각을 센서정보로부터의 피드백 관계로부터 결정하는 과정은 동역학 분석 및 제어과정이라 할 수 있다.

2.1 운동학 분석

자율주행로봇의 무게중심이 C 이고, 두 바퀴의 중심점이 A 일 때 주행로봇좌표계 {x, y} 를 이차원 평면의 절대좌표계 {X, Y} 상에서 나타내면 다음 그림과 같이 표현될 수 있다.

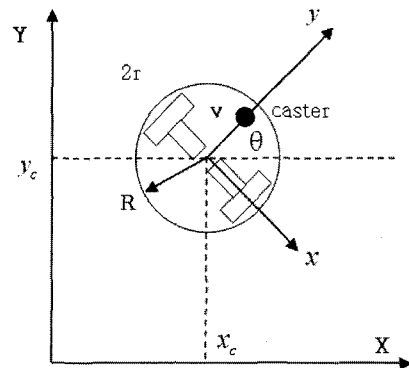


그림 1 자율주행로봇의 평면좌표계 표현

Fig. 1 Cartesian coordinate representation of mobile robot

주행로봇의 주행속도 v 와 각속도 w 를 절대좌표계의 변수로 표현하면 다음 행렬식과 같다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & -d \sin \phi \\ \sin \phi & d \cos \phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (1)$$

여기서 d 는 A 점과 C 점간의 직선거리를 의미한다. 각 바퀴의 각속도와 주행로봇 좌표계의 속도성분간의 관계는 다음과 같다.

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \quad (2)$$

위의 두 식으로부터 다음 식이 유도될 수 있다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \phi - \frac{r}{L} \cdot d \cos \phi & \frac{r}{2} \cos \phi + \frac{r}{L} \cdot d \sin \phi \\ \frac{r}{2} \sin \phi + \frac{r}{L} \cdot d \cos \phi & \frac{r}{2} \sin \phi - \frac{r}{L} \cdot d \cos \phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \quad (4)$$

즉, 다음과 같은 관계를 만족한다.

$$\dot{x}_A = \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L) \cos \theta \quad (5)$$

$$\dot{y}_A = \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L) \sin \theta \quad (6)$$

$$\dot{\theta} = \frac{r(\dot{\theta}_R - \dot{\theta}_L)}{2R} \quad (7)$$

이 때 식 (4)는 식 (3)으로부터 $r \cdot d/L$ 값이 극히 작다는 가정 하에 소거되었다. 식 (5), (6), (7)로부터 두 바퀴의 속도는 주행로봇의 주행속도 및 방향전환과 관련되므로 두 바퀴의 속도를 제어함으로써 주행로봇의 주행을 가능하게 한다.

2.2 동역학 분석 및 제어

주행로봇의 전체 운동에너지는 다음과 같은 세 항의 합으로 표현된다.

$$T = T_t + T_r + T_{kr} \quad (8)$$

여기서 T_t 은 주행로봇의 순수한 병진운동에너지성분을 의미하고 T_r 은 XOY 평면상에서의 회전에너지성분, T_{kr} 은 바퀴의 회전운동에너지를 나타낸다. 주행로봇의 질량이 M , A점에서의 관성모멘트가 I_A , 바퀴의 회전관성 모멘트가 I_o 이고 주행로봇 중심에서의 선속도를 v_c 라고 하면 각각의 에너지 항들은 다음 식들과 같이 풀어서 쓸 수 있다.

$$T_t = \frac{1}{2} M v_c^2 = \frac{1}{2} M (\dot{x}_c^2 + \dot{y}_c^2) \quad (9)$$

$$T_r = \frac{1}{2} I_A \dot{\theta}^2 \quad (10)$$

$$T_{kr} = \frac{1}{2} I_o \dot{\theta}_R^2 + \frac{1}{2} I_o \dot{\theta}_L^2 \quad (11)$$

위 식들을 라그랑지(Lagrange) 에너지 방정식에 대입하여 정리하면,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_R} \right) - \frac{\partial L}{\partial \theta_R} = \tau_R - K \dot{\theta}_R \quad (12)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_L} \right) - \frac{\partial L}{\partial \theta_L} = \tau_L - K \dot{\theta}_L \quad (13)$$

여기서 첨자 R 과 L 은 각각 오른쪽, 왼쪽 바퀴임을 의미하고 K 는 모터와 바퀴 간의 점성마찰계수이다. 위 두 식을 풀면 최종적으로 다음과 같은 동역학 방정식을 얻을 수 있다.

$$A \ddot{\theta}_R + B \dot{\theta}_L = \tau_R - K \dot{\theta}_R \quad (14)$$

$$B \ddot{\theta}_R + B \dot{\theta}_L = \tau_L - K \dot{\theta}_L \quad (15)$$

여기서

$$A = \left(\frac{Mr^2}{4} + \frac{(I_A + Md^2)r^2}{4R^2} + I_o \right) \quad (16)$$

$$B = \left(\frac{Mr^2}{4} - \frac{(I_A + Md^2)r^2}{4R^2} \right) \quad (17)$$

주행로봇이 특정한 목적을 수행하기 위해 목표지점을 설정하거나, 목표지점에 도달하는 과정에서 경유점들이 설정되어 있는 경우에 최단거리, 안전한 거리 또는 에너지소모가 가장 작은 거리로 정확한 위치와 방향을 유지하면서 이동하는 것은 경로설정 및 제어의 중요한 목표이다. 위와 같은 동역학 분석을 통하여 로봇의 하중과 이로 인한 지면과의 마찰력, 바퀴의 필요 토크량, 회전 시 적절한 제어력 등을 산정할 수 있고, 컴퓨터 시뮬레이션 등의 선행 실험 후에 적절한 주행로봇 구조를 설계할 수 있다. 제작된 주행로봇의 제원은 무게 8 Kg, 지름 28 cm, 바퀴를 제외한 몸체 높이 54 cm, 바퀴지름 8 cm, 엔코더 DC 모터의 정격토크 3 Kg-cm 이다. 본 연구에서는 A* 알고리즘을 적용하여 최단거리검색을 수행하였고, 이를 두 바퀴의 PD 제어방식과 본 연구에서 개발한 퍼지논리기반 적응PD제어(Fuzzy based adaptive PD Control)방식을 사용하여 제어속도 및 제어각을 조절하였다.

3. 동시적 자기위치인식 및 지도작성 (SLAM)

SLAM은 지도작성, 자기위치 정확화, 이동제어의 모든 과정이 상호 통합적으로 연계되어야만 주행로봇의 목적하는

바인 주행 성능을 확보할 수 있고, 주행 성능은 원위치 복귀 정확도(loop closing)로 평가할 수 있다. 본 연구에서는 지도 작성 과정과 자기위치인식 및 정확화 과정을 구분하여 최종적으로 원위치 복귀 정확도를 향상하기 위한 방안을 논의한다. 본 연구에서는 거주지 또는 사무실 등의 실내 환경에서 특정 목적을 수행하면서 이동 정확성을 갖출 수 있도록 하기 위해서 시뮬레이터에 의한 실시간 지도 정보의 생성과 이와 연동하는 로봇의 주행 제어 성능 향상에 관하여 다룬다. 시뮬레이터에 의한 지도 작성 과정은 실제 로봇이 초기의 검색 과정을 거치면서 지도를 작성한 후에 서비스 기능을 수행할 수 있기도 하고, 초기에 실내 공간에 대한 지도의 CAD 도면 데이터를 로봇에 사용자가 직접 입력함으로써, 지도를 작성해 줄 수도 있도록 한다.

3.1 지도작성과정

일정 영역을 지도화하는 과정에서 모든 영역을 빠짐없이 정보로 저장하기 위해서는 작은 공간 또는 격자 형태로 나누는 것이 필요하다. 환경정보를 세밀하게 표현하기 위해서는 격자의 크기는 작을수록 좋으나, 센서의 성능 및 개수와 로봇의 이동정확성 등의 한계에 의해 격자크기는 제한되게 된다. 격자지도는 표현정보가 격자의 점유와 비점유의 두 형태로 표현되므로 간단하지만, 검색영역이 넓어질수록 정보가 대량화되어 저장 및 처리속도저하 문제가 생기는 등의 단점이 있다.

주행로봇은 레이저센서 또는 레이저센서 등의 거리인식센서를 사용하여 이동과정 중에 격자지도를 구성해가며 지도의 영역을 확대해가게 된다. 격자지도와 달리 위상기하지도는 환경정보의 직접적인 측정을 피하고 로봇의 상대적 위치 관계정보를 이용한다. 위상기하지도방식은 격자지도방식에 비하여 이를 적용한 로봇의 이동속도가 비교적 빠르고, 인간의 인식과 유사한 방식으로 이동정보가 간단명료하게 표현되는 장점이 있다. 또한 주변지형정보에 대한 정확한 인식 정보가 필요하지 않으므로 격자지도에서 발생한 주행로봇의 미끄러짐 또는 거리격차 등의 오차현상에 비교적 안정적이다. 또한 센서잡음과 미소한 환경의 변화에 강한 특성을 보이며 저장되는 정보가 상대적으로 적으므로 넓은 이동영역까지의 지도 영역 확장이 가능하다. 하지만 이는 환경이 복잡할수록 더욱 많은 정보량이 필요해지고, 이미 지나간 지점으로의 복귀 정확성이 낮은 단점이 있다.

지도작성과정은 센서의 특성과 센서로부터의 정보에 크게 의존하는데, 센서정보의 분석, 시간에 대한 분리과정, 로봇 자세 추정, 일부영역 지도 생성 그리고 이동의 순서로 진행된다. 본 연구에서 사용한 레이저센서는 가격이 저렴하고, 비교적 넓은 영역(30cm ~ 3m) 에 이르는 거리정보를 제공하지만, 레이저의 진행형태가 거리에 따라 원추형으로 전개되므로 이로 인한 인식 방향정보가 부정확할 수 있다. 따라서 스캔이 가능하도록 일정 시간과 각도간격(2°)으로 회전이 가능한 레이저센서와 고정된 전방과 양측면의 초음파센서를 추가 설치하고, 이 정보를 함께 해석함으로써 방향성과 거리정보를 보다 세밀하게 결정하는 과정을 거치도록 하였다. 실제 지형의 가상의 격자 간격은 10cm 로, 로봇 크기의 1/3 크기가 되도록 하며, 레이저 센서는 거리가 멀어질수록 검출

해상도가 낮아지게 되므로 약 1.5 m 내의 영역 정보만을 이용하고, 검색을 위한 이동은 벽면을 따라서 일정 거리 간격을 유지하면서 이동하거나, 직선 주행만을 하게 되므로 양 측면에 위치하는 초음파 센서에 의하여 측정되면서 주행 경로가 제어되도록 하게 된다.

일정 영역공간을 모두 검색하기 위해서 검색방법에 대한 고려가 필요하다. 이를 검색패턴(Coverage pattern)이라 하며 검색패턴알고리즘에 관한 연구가 많이 수행되었으나 환경의 특수성에 따라 각각의 장단점이 존재한다[5]. 본 연구에서는 모든 영역을 빠짐없이 검색하는데 우수한 성능을 갖는 가지구조방식의 역추적 알고리즘(Backtracking Algorithm)을 사용하였다. 역추적은 일종의 반복적 회귀형태로 모든 가능한 선택방법 중에서 유일한 해를 체계적으로 찾아낸다. 모든 가능한 해는 일정값을 갖는 벡터형태로 표현되고 해를 찾을 때까지 깊이우선방식(depth first manner)과 이후의 횡단방식으로 반복적으로 수행하게 된다. 매 단계에서 벡터값을 채우게 되고, 더 이상 국소해를 찾을 수 없을 때까지 벡터차원연장과정은 계속되면서 대체값으로 벡터성분을 바꾸는 역추적 과정을 거치는데, 이 과정에서 기존의 경로성분은 삭제된다. 다음은 이의 의사코드(Pseudo Code)를 표현한 것이다.

ALGORITHM

```

try(v1,...,vi)
IF (v1,...,vi) is a solution THEN RETURN (v1,...,vi)
FOR each v DO
  IF (v1,...,vi,v) is acceptable vector THEN
    sol = try(v1,...,vi,v)
    IF sol != () THEN RETURN sol
  END
END
RETURN ()
    
```

만약 S_i 가 v_i 의 차원공간에 해당한다면 $S_1 \times \dots \times S_m$ 는 주어진 문제의 최종해가 된다. 검색대상 공간에서 검색할 부분을 결정하는 벡터공간을 검토하기 위한 유효성 기준이 사용되고, 해공간의 검색을 위한 횡단검색은 모든 가능한 해의 정보가 저장되는 가지구조에 대하여 깊이우선 탐색방식에 의해 표현되어 해를 찾을 때 말단가지부터 근본 가지까지 역으로 경로추적을 할 수 있다. 이 방식은 시간적으로 연산수행속도 면에서 손해를 보지만 정밀하고 지역 함정에 빠지는 오류 등이 거의 없는 장점이 있어 시뮬레이터에 의해 경로가 결정되고 이를 실시간 제어 시스템에 명령정보만으로 전송해주는 본 연구와 같은 분산제어방식에 적합하다. 이에 관한 간략한 도해적 표현은 다음과 같다[8].

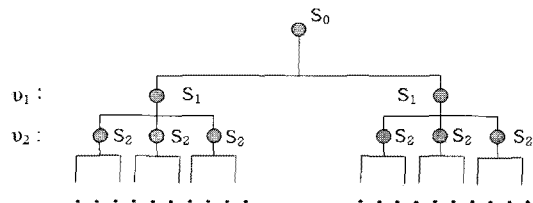


그림 2 역추적 알고리즘의 가지점구조
Fig. 2 Tree node structure of back tracking algorithm

역추적 알고리즘의 검색경로설정의 결과는 시뮬레이터 윈도우 상의 격자지도 상에서 실시간으로 표현되며, 로봇의 실제이동에 따른 환경인식과 연계되어 이루어진다. 초기 지도의 모든 격자 내부 정보는 미지영역으로 회색으로 표시되고, 검색 과정 중 장애물 영역은 점유 상태(검정색)로 빈 공간은 비점유 상태(흰색)로 표시된다. 시뮬레이터 상의 지도 격자는 2 cm 간격으로 구성되고, 실제 지형의 20%로 축소되어 표현되며 로봇의 이동 과정에 따라 지도 작성 과정이 실시간으로 이루어지게 된다. 역추적 알고리즘이 수행될 때, 일부 복잡한 환경영역에서 로봇이 영역검색과정 중 막다른 상황에 도달할 때가 있는데, 이 때는 다음 검색대상의 초기 지점까지 A* 알고리즘에 의한 최단 경로로 이동한 후 지도 검색을 계속 수행하게 된다. 이동 중에는 환경정보에 대한 센서인식이 필요 없으므로 이동시간을 절약할 수 있다. 다음 그림은 지도작성과정의 HOKUYO(주) 레이저센서의 빔 진행 패턴과 스캔방향을 표현한 것이다.

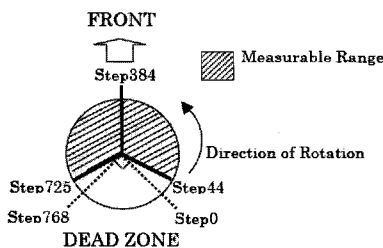


그림 3 지도작성 과정의 레이저빔 진행 패턴
Fig. 3 Propagation pattern of laser beam during mapping

레이저 센서에 의한 거리 정보는 2mm 해상도 이상으로 정확하므로 본 연구에서 목표로 하는 10cm 이내의 최종 원위치 복귀 오차 성능을 만족하기 위한 조건으로 충분하며, 생성되는 지도 정보의 격자 간격은 2cm 로 레이저 센서에 의해 격자의 10% 이상의 검출 영역이 존재 시에 점유영역으로 지정된다. 위와 같은 지도작성과정을 통해서 작성된 지도는 실제 실내의 지형의 1/5의 크기로 축소된 형태이며 정확하다고 전제하고, 이 정보를 이용하여 주행 중 위치인식과정을 수행하게 된다. 지도 정보를 이용하여 주행하는 중에 로봇이 예상치 못한 장애물에 접하였을 때에는 초음파 센서 정보를 이용하여 전방의 장애물을 기준으로 좌우측 중에 열린 공간이 더 많은 쪽, 즉 전방의 세 개의 센서 측정값 중 큰 값이 나오는 방향을 선정하고, 이 방향으로 장애물과 일정한 거리인 10cm 를 유지하면서 장애물 주위로 돌게 된다. 장애물을 우회하는 도중에 지도에서 결정된 경로를 만나게 되면, 그 지점부터 다시 경로를 따라 진행하게 된다. 이러한 주행 과정은 각 영역 내부에 확실성 지수를 부여하여 갱신 과정에 반영하게 되는데, 초기 선정된 지도의 지수값을 0 으로 지정하고, 주행 과정 중에서 인식되는 가변 지형물을 발생과 소멸시에 각각 0.1, -0.1 의 지수 변동률을 줌으로써 지도 정보를 일정 조건에서 갱신하게 된다, 즉, 10 번 주행 시에 반복적으로 고정위치에서 발견되었을 때, 즉 1 의 값이 되었을 때 점유로 변경하고, 그 격자영역 지도 확실도를 0으로 변경하게 된다.

3.2 자기위치 인식과정(localization)

외부에서 기준 위치정보가 주어지지 않으므로 주행로봇은 이동 경로 중의 방향전환점에 이르렀을 때, 주변 240°정보를 약 2°간격으로 레이저센서로 스캔하고 이에 관한 정보를 분석함으로써 국소영역에서의 자기위치인식을 수행하게 된다. 초기 Weiß가 제안하고 Rüfer[9] 가 개선한 각도히스토그램(angle histogram) 방법과 Martinelli[10] 등의 상대지도지형 필터(The Relative Map Geometric Filter, RMGF)로 보완하여 코너 및 모서리의 인식의 정확도를 향상시켰다. 레이저 센서는 이러한 각도 히스토그램과 상대지도지형을 검출하기에 적당한 성능을 구현할 수 있으며, 관련 프로그램의 처리 속도도 주행 속도에 비하여 빠른 장점이 있다.

자기위치인식과정은 회전각도정보와 직교좌표계상 좌표정보의 인지 과정으로 나뉜다. 회전각도의 정보를 표현하기 위한 각도히스토그램은 다음과 같은 과정을 통해 구해진다. 우선 스캔과정에서 얻어진 거리정보들의 중첩적 분석을 통해 장애물의 외형을 의미하는 점들을 구하고, 각 점에서 반시계방향으로 인접한 점들과 연결한 후 이 연결선의 기울기, 즉 각도를 산출한다. 전체 영역을 스캔한 후 각각의 연결선으로부터의 각도정보를 분포도형식으로 표현하게 된다[9].

주행로봇이 초기 시작점을 출발하여 몇 개의 경유점들을 거친 후 최종 목표점으로 진행하기 위해 최단거리검색 알고리즘이 수행되고, 이 결과가 시뮬레이터 상의 지도에 표시된다. 일정 경유점이나 경로의 진행방향이 꺾인 점들은 위상기하적으로 방향이 달라지는 특징이 있으므로, 주행로봇에게 경유점 간의 벡터정보가 명령된다. 다음 이동 경유점까지의 거리와 방향각 정보가 주어지며, 각 경유점에 도달한 이후에는 180° 방위를 2° 간격으로 스캔하고 이 정보들을 정합하는 과정을 통해 주변상황을 인식하게 된다. 이 과정은 후에 다시 이 지점을 지나갈 때 처음 인식한 환경과의 비교를 통해 일치 정확도를 높이기 위한 것이다.

시뮬레이터 상의 위상기하정보 추출과정에는 격자지도 상에 표현된 최단경로점들의 연결상태를 다음과 같은 조건에 의해 판단하게 된다. 즉 $i-1$ 번째 좌표 (x_{i-1}, y_{i-1}) , i 번째 좌표 (x_i, y_i) 와 $i+1$ 번째 좌표 (x_{i+1}, y_{i+1}) 간의 기울기 정보가 다를 때, 즉,

$$\frac{y_{i+1} - y_i + \epsilon}{x_{i+1} - x_i + \epsilon} \neq \frac{y_i - y_{i-1} + \epsilon}{x_i - x_{i-1} + \epsilon} \quad (18)$$

일 때 새로운 이동명령정보가 생성된다.

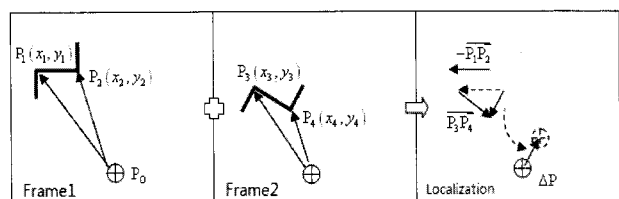


그림 4 레이저 스캔 각도히스토그램 방법에 의한 위치인식
Fig. 4 Localization by angle histogram of laser scan

위 그림과 같은 위치인식과정은 새로운 이동 명령이 생성되는 경로 변경점에서 주위의 특징점들을 기준하여 이루어진다. 시뮬레이터 상에서 구성된 지도상의 좌표와 특징점 좌표의 위치가 기준 정보가 되고, 측정되는 특징점 좌표는 비교 정보가 된다. 로봇은 비교 정보와 기준 정보를 비교하여 Fig. 4 와 같은 기하학적 역연산 과정을 통하여 현재의 자기 위치를 파악하고 기준 위치와 어긋난 정도를 알 수 있게 된다. 이와 같은 위치 오차 정보는 다음 경유점을 향하여 주행할 때 두 바퀴의 가해지는 제어력에 반영되어 위치 정확도를 점차 높게 된다.

이동 중의 지면과의 마찰과 미끄러짐 현상, 모터기어의 백래시, 센서잡음 등에 의해 제어결과는 항상 에러를 포함하게 되며, 주행로봇의 위치오차는 이동거리가 길어질수록 커지는 경향이 있다. 따라서 본 연구에서는 각 경로점에서 자기위치인식을 통해 환경정보를 저장하고, 다시 이 지점에 도달했을 때 주행로봇의 현재 위치와 각도정보를 지도상의 위치 및 각도정보와 비교하여 오차를 분석하고 다음 경로이동시에 이를 보상하는 퍼지논리기반 적응PD 제어를 수행하였다. 이는 부분적으로 확장칼만필터의 특성인 과거의 에러정보를 현재의 제어기설계에 반영하는 성질과 퍼지논리제어기와 같은 퍼지집합적 입력 및 언어규칙에 의한 제어기출력 기능을 응용한 것이라 할 수 있다. Begum 등[12] 은 자기위치인식과정에서 퍼지논리제어기를 활용하였고, 본 연구에서는 이 방법을 비교하여 보다 효과적인 제어성능을 가질 수 있도록 개선하고자 하였다.

Begum 의 문헌에서 위치 오차는 퍼지 모델에 의해 결정되는데, 위치 오차에 영향을 주는 변수를 고려하여 다음과 같은 식을 구성하였다.

$$X_t = \Phi(u_{t-1}, v_t, w_t, d_t, F_t, K) \quad (19)$$

여기서 v_t, w_t, d_t 와 F_t 는 (t-1,t] 시간 동안의 각각 로봇의 선속도, 각속도, 이동거리와 지면마찰상태를 의미한다. 또한 변수 K 는 엔코더 신호, 거리 인식 센서 및 구동기 해상도 오차 등과 기구 자체에서 기인하는 구조적 오차의 영향을 포함하기 위해 정의되었다. 위와 같은 위치 오차 원인이 되는 변수들은 모두 비선형 거동을 하며, 회전각도가 급격히 커지거나 바다면 상태가 변경되는 때에는 상호 영향을 주고받으므로 일반적으로 선형 모델로 단순화하기 어렵다. 퍼지논리 제어기는 비선형 거동을 하는 변수들이 많은 경우에 유용한 제어 방식으로서 인간의 인지 방식과 유사하고 각 변수의 한계값 설정에 용이하다는 장점이 있다.

퍼지 시스템은 주행로봇의 위치오차에 대한 전문적 지식에 기반하여 구축되며 언어적 표현으로 각 변수와의 논리식이 구성된다. 문헌에서는 선속도, 각속도, 바다면 상태, 이동거리가 입력변수가 되고, 미끄러짐, 위치오차, 회전각 오차, 위치오차율과 회전가공차율이 출력변수가 된다. 입력변수와 출력변수 사이에 퍼지 언어 변수에 의해 표현되는 IF-THEN 조건문 관계가 다음과 같이 구성되었다.

IF F_t is C_k AND d_t is D_l AND v_t is A_i THEN 미끄러짐($\delta\theta_d$) is E_w ,

IF F_t is C_k AND v_t is A_i AND w_t is B_j THEN 회전각오차($\delta\theta_{re}$) is G_z ,

IF F_t is C_k AND d_t is D_l AND v_t is A_i THEN 위치오차(δS_{te}) is F_v ,

IF d_t is D_l THEN 회전각오차율($\delta\theta_K$) is H_{1q} ,

IF d_t is D_l THEN 위치오차율(δS_K) is H_{2q} .

여기서 $A_i, B_j, C_k, D_l, E_w, F_v, G_z, H_{1q}$ 와 H_{2q} 는 각각 v_t, w_t, F_t, d_t , 미끄러짐, 위치오차, 회전각오차, 회전각오차율과 위치오차율의 퍼지 부분 집합을 의미한다. 마찬가지로 $\delta S_{te}, \delta S_K, \delta\theta_d, \delta\theta_{re}$ 와 $\delta\theta_K$ 는 위치오차, 위치오차율, 미끄러짐, 회전오차와 회전오차율의 비퍼지화 값들을 의미한다. 각각의 퍼지 부분집합의 언어적 명칭은 다음과 같다.

$A_i : \{LOW, MEDIUM, HIGH\}, i=\{1,2,3\}$.

$B_j : \{LOW, MEDIUM, HIGH\}, j=\{1,2,3\}$.

$C_k : \{SLIPPERY, ROUGH\}, k=\{1,2\}$.

$D_l, E_w, F_v, G_z : \{VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH\}, l, w, v, z = \{1,2,3,4,5\}$.

H_{1q} 와 $H_{2q} : \{LOW, MEDIUM, HIGH\}, q=\{1,2,3\}$.

퍼지 추론 방법은 Mamdani 형태의 퍼지 함수와 최대최소 무게중심법(Max-Min Gravity)에 의해 이루어지고, 이러한 퍼지 시스템의 결과는 로봇에 이동 명령에 의해 바퀴가 일정 각도만큼 회전한 후의 시점에서 현재 위치에 대한 위치와 방향에 대한 오차값 $r_s(t), r_\theta(t)$ 가 된다. 오차값은 일반적으로 매 시간마다 증가하게 되므로 이전 시간의 오차값이 다음과 같이 반영된다.

$$r_s(t) = r_s + r_{s(t-1)} \quad (20)$$

$$r_\theta(t) = r_\theta + r_{\theta(t-1)} \quad (21)$$

위와 같은 일반적인 퍼지논리제어기는 비퍼지화하는 과정에서 연산량이 많아지고, 하드웨어적으로 32 비트 이상의 부동소수점 연산과정이 필요하다. 본 연구에서의 주행로봇에 사용되는 바퀴축의 모터는 PWM 방식에 의해 회전속도가 결정되는 12V 엔코더 DC 모터로서 감속비 1/62 이고, 최대속도는 초당 약 1.8 회전이며, 모터를 구동시키기 위한 제어 회로는 ATmega128 과 시뮬레이터의 MATLAB 프로그램에 의해 시리얼 통신을 통해 이루어지게 되어 있다. 여기서 ATmega128 은 8 bit 명령어 처리방식으로 연산을 수행하므로 위에서 언급한 퍼지논리제어기 방식으로는 적용하기 어려운 점이 있다. 따라서, 본 연구에서는 퍼지논리제어기의 퍼지논리 추론 시스템과 같이 전문가 시스템(expert system)에 기초하면서 제어 변수를 보다 효율적으로 구성하고, 비퍼지화 단계와 같은 부동소수점 연산과정이 필요없도록 적응 PD 제어 알고리즘을 구성하였다. 구성된 제어 알고리즘의 변수들은 센서에 의한 정보를 이용하여 경로점에서 갱신되도록 하는 적응성을 갖추면서 PD 제어기의 계인값들을 조건에 맞도록 적용시킨 제어 시스템이라고 할 수 있다.

처음 이동 중 저장되었던 위치를 원점 O 라 하고 다시 이 경로점에 도달했을 때의 위치를 P 라 하면 진행방향을 Y 축이라 할 때 XY 평면 상에서 사분면 영역과 오차 반경으로 오차정보를 구분할 수 있다.

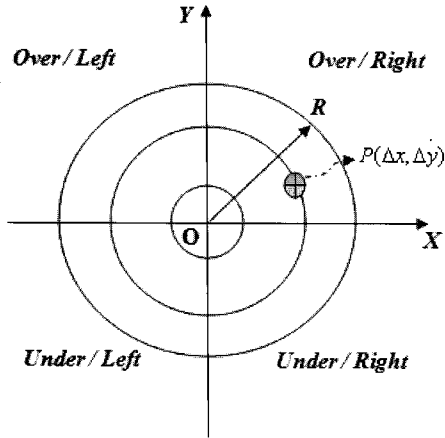


그림 5 로봇위치목표점 기준의 오차영역 및 방향
Fig. 5 Error position and direction based on target point of a robot

다음 이동단계에서 오차를 보상하기 위해 주행로봇의 두 바퀴의 제어각을 각각 다음 식과 같이 표현한다.

$$\begin{aligned} \theta_L &= \theta_L^F + \Delta\theta_L \\ \theta_R &= \theta_R^F + \Delta\theta_R \end{aligned} \quad (22)$$

여기서 θ_L^F 와 θ_R^F 는 현재 위치에서 다음 경유점으로 이동할 좌우바퀴의 회전제어각도이며, $\Delta\theta_L$ 과 $\Delta\theta_R$ 는 이전 단계의 이동변수정보와 현재위치의 오차정보를 반영한 좌우바퀴의 보상제어 각도이다. 보상제어 각도는 다음과 같은 요소로 구성되며,

$$\Delta\theta_{L,R} = \Delta\theta_{L,R}^p + \Delta\theta_{L,R}^e \quad (23)$$

$$\Delta\theta_{L,R}^p = k_1\theta_{err} + k_2l_{err} \quad (24)$$

$\theta_{err} = (\theta_m - \theta^p)$, $l_{err} = sign(\Delta x) \cdot (l - 4b)$ 를 만족한다. 여기서 l 은 다음 진행경로의 직선거리, b 는 로봇의 전후 길이이며, 실험경험 상 이동거리가 로봇길이의 4 배를 기준으로 좌우방향의 오차정보가 변경되는 성향을 반영한 것이다.

$\Delta\theta_{L,R}^e$ 현재위치정보를 이용하여 다음과 같은 방식으로 결정되는 보상각도이다. $\Delta\theta_{L,R}^e$ 는 그림 5의 영역구분정보와 기준점에서부터 오차거리 정보에 의한 제어각도 조정각을 의미하며 다음과 같이 표현된다.

$$\Delta\theta_{L,R}^e = a \cdot Z + b \cdot \frac{|\Delta x|}{\sqrt{\Delta x^2 + \Delta y^2}} \quad (25)$$

$$Z = \eta(x^2 + y^2) + 1 \quad (26)$$

여기서 η 는 매 시간단계에서 다음과 같은 업데이트과정을 거친다.

$$\eta_i = \left(1 + \frac{\sqrt{(\Delta P_i)^2 + (\Delta \theta_i)^2} - \sqrt{(\Delta P_{i-1})^2 + (\Delta \theta_{i-1})^2}}{\sqrt{(\Delta P_{i-1})^2 + (\Delta \theta_{i-1})^2}} \right) \eta_{i-1} \quad (27)$$

두 바퀴의 전후 및 좌우 조향 정보를 의미하는 계수 a, b 는 다음 표와 같이 영역위치에 따라 결정된다.

표 1 오차영역에 대한 보정계수
Table 1 Compensation coeff. of error position

	O/R		O/L		U/L		U/R	
	a	b	a	b	a	b	a	b
L	-1	0	-1	1	1	1	1	0
R	-1	1	-1	0	1	0	1	1

다음 그림은 오차보상함수의 곡면윤곽형상을 3차원그래프로 도시한 것이다.

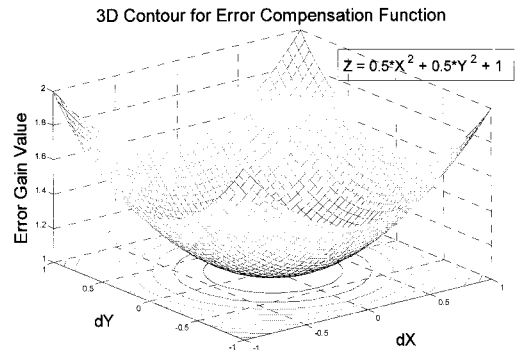


그림 6 오차보상함수의 곡면윤곽형상($\eta=0.5$)
Fig. 6 Contour surface shape of error compensation func.

즉, 자기위치추정 오차의 영역을 분석할 때, 중심점에서의 멀어진 정도에 비례한 제어력이 가해지고, 현재 진행방향을 기준으로 치우쳐진 방향정보를 다음 이동정보에 반영하는 것이다. 이는 언어적 방향정보의 분석처리와 노면 등 환경 및 운동학 오차를 보상하기 위한 제어력 분포함수의 형태를 관련 변수를 매번 업데이트 가능하게 함으로써 적응성을 높인 것이라 할 수 있다.

4. 실험결과 및 분석

주행로봇의 크기는 폭 30cm, 길이 20cm, 높이는 10cm 이다. 시뮬레이터 프로그램은 MATLAB 7.5 에 의해 제작되었고, 하드웨어 및 제어기와와의 인터페이스는 ATmega128 제어 회로를 통해서 DC기어모터와 센서시스템에 연결되었다. 주변환경을 인식하기 위한 센서는 양쪽의 고정된 초음파센서와 앞쪽 상단 부위에 스캔이 가능한 레이저 센서로 구성되

며, 40° 간격으로 설치된 초음파 센서의 정보는 로봇의 이동 중에 양 측면의 지형 정보를 실시간으로 받아들이고 이 정보로부터 벽면의 방향 및 거리 정보를 인지하게 된다. 또한 각도정보를 알기 위해 전자나침반을 장착하여 자기방향인식 및 주행방향조절 과정에 이용되었다. 그림 7은 주행로봇의 주행 중 사진과 시스템 개략도를 표현한 것이다.

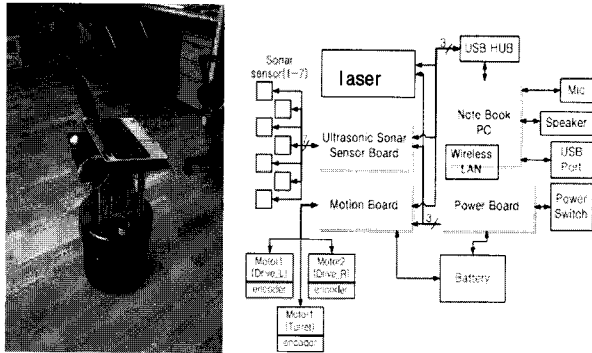


그림 7 주행 중인 로봇(좌)과 시스템 개략도(우)
 Fig. 7 The navigating robot(left) and system diagram(right)

실험은 회의용 탁상, 파티션, 책상, 의자 등의 사무용가구로 구성된 연구실에서 수행되었고, 이동 속도는 평균 15 cm/s 가 되도록 하였다. 그림 8은 지도작성과정이 시뮬레이터 상에서 표현되는 상황을 나타낸 것이며, 그림 9는 사용자가 설정한 초기점, 경유점들 및 목표점들로부터 자동적으로 위상기하정보로 변환된 이동명령정보가 산출되고 있음을 표시된 빨간 타원 선 영역으로 나타낸 것이다.

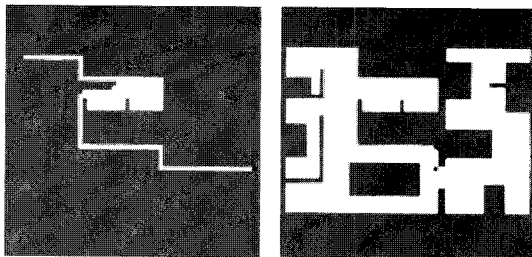


그림 8 시뮬레이터 상의 지도작성 과정
 Fig. 8 The mapping process on the simulator

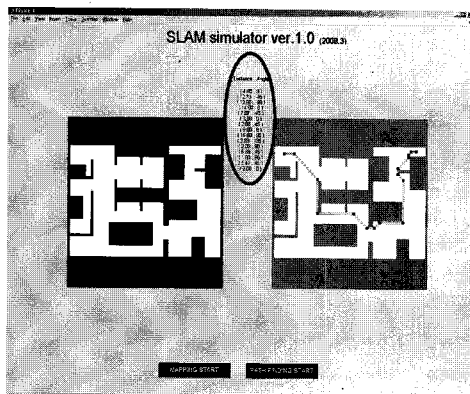


그림 9 격자위상혼합지도의 경로작성
 Fig. 9 Trajectory construction by hybrid map

위의 이동경로를 반복하여 수행할 때의 위치정확성 및 제어성능을 비교하기 위해 다음과 같은 실험을 수행하였다. 주행 시험을 총 10회 실시하고 최종 위치점을 포함한 10개의 경로 변경 점에서 지도상의 좌표점과 측정되는 위치의 거리 차이를 비교하여 주행로봇의 이동 성능을 평가하고자 하였다. 다음 식은 이동 정확성을 평가하기 위한 지표로서 평균오차항(Average Errors)을 나타낸 것이고,

$$AE = \sum_{i=1}^n \frac{\sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}}{N} \quad (i=1,2,3,\dots,n) \quad (28)$$

여기서 N 은 총시험 횟수이며 i 는 경로점의 개수를 의미한다. 다음 그림은 실험에 의해 작성된 지도와 시뮬레이터 상에서 두 실험 결과를 나타낸 것이다.

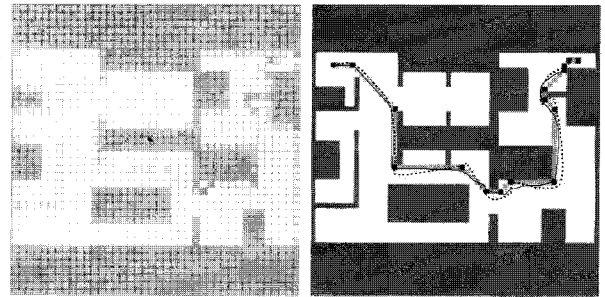


그림 10 작성된 지도와 이동정확성 실험 과정
 Fig. 10 Built map and experimetal process of localization

그림 10의 우측 그림에서 점선은 비교되는 대상 알고리즘을 적용한 주행 로봇의 이동 과정 결과를 나타낸 것이며 실선은 본 연구에서 제안한 제어 알고리즘을 적용한 결과이다. 위와 같은 실험을 10번 수행하여 최종 목적지점에서 경로점을 포함하여 10개 지점의 위치 정확도를 평가한 결과는 다음 그림과 같다.

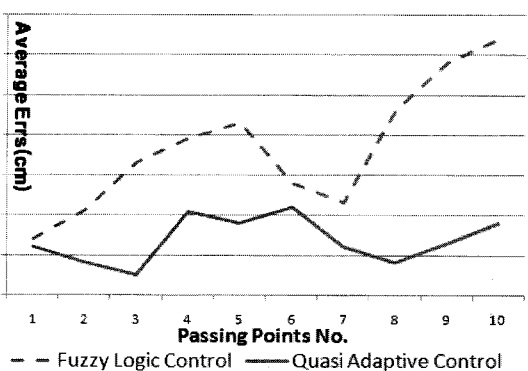


그림 11 제어성능 비교 그래프
 Fig. 11 Comparison graph of controlled performance

그림 11은 9개의 경유점을 지나 최종목표점에 도달할 때까지의 평균 오차값을 cm 단위로 비교하여 표시한 그림이며, 본 연구에서 제안한 퍼지논리기반 적응PD 제어 알고리즘을 적용한 결과가 일반적인 퍼지논리 제어기에 의한 결과

에 비하여 평균오차가 약 2.5 배 감소하는 성능개선효과가 있음을 보여주었고, 최종 위치에 도달하는 위치정확도 또한 3 cm 이내에서 유지되었다.

5. 결론 및 향후 과제

본 연구에서는 SLAM 과정에서 정보처리성능을 향상시키면서 위치정확도를 유지하기 위한 통합 시뮬레이터와 이와 연계된 제어시스템을 개발하였다. 지도 작성과정의 정확성을 전제로 하고 실제 지형을 축소한 형태로 작성된 지도 정보를 시뮬레이터 상에서 기하학적으로 처리하고 이 정보를 직접 이용하는 통합 시뮬레이터 소프트웨어와 주행로봇의 이동정확성을 높이기 위한 하드웨어 제어 알고리즘을 제안하였다. 지도 작성과정은 실내 영역을 효율적이고 지형 변화 과정에서 지도 정보를 갱신하는 데 유리한 격자지도방식으로 검색을 수행하였고, 국소영역에 고립되지 않으면서 안정적으로 전 영역을 검색할 수 있는 역검색 알고리즘을 수행하였다.

지도작성과정에서의 주행로봇의 SLAM 정확성 향상 및 자동 수행 능력을 갖추기 위하여 다음과 같은 개선된 방식이 적용되었다. 우선 지도작성과정에서 시뮬레이터 상의 축소된 지도 영역에서 격자지도방식으로 폐쇄공간의 모든 영역을 검색하고 정보를 저장한 후, 목적수행을 위한 경로를 시뮬레이터 상에서 최단거리검색알고리즘에 의해 구하였다. 시뮬레이터는 주행로봇의 이동지령을 기하위치정보로 주게 되며, 이는 격자지도상에서의 경로점 좌표와 직선정보로부터 얻어진다. 시뮬레이터에 의해 다량의 데이터로 구성되는 지도정보를 그래픽영역에서 처리하고, 자동적으로 경로가 구성됨으로써, 이 과정에서 적은 정보를 효율적으로 사용함으로써, 정보처리속도를 약 3배 이상 향상시킬 수 있도록, 제어시스템과 분리되는 분산처리기술을 적용함으로써 이동시 정보처리 시간을 줄이고 이동속도를 유지할 수 있었다. 또한 동력기계 및 제어시스템에는 최소한의 이동명령정보가 전송되고, 레이저와 초음파센서 정보를 복합적으로 이용하여, 주행 속도에 비하여 정보처리 속도를 높이면서 이동과정에서의 자기위치 정확성을 높일 수 있었다. 이는 기존의 확률기반 정보처리 방식들, 예를 들어 칼만필터방식, Rao-Blackwellized 입자필터방식(Rao-Blackwellized particle filter) 등에 비하여 적은 정보 처리양과 빠른 속도를 유지함으로써, CPU 처리와 서비스를 위한 제어 성능 구현의 안정성을 높일 수 있다.

레이저 센서 시스템은 지도 작성 과정에서 전방의 장애물 정보를 파악하면서 특정 위치에서는 지도 정보에 기반한 경로 변경 명령을 받으면서 동시에 위치 인식 수정 과정을 수행하였다. 레이저 센서시스템은 주행로봇의 몸체와는 별도의 자유도를 가진 구조로서 몸체에 고정되어 있는 7개의 초음파 센서와는 다른 역할을 하며, 지도작성과정 중 시뮬레이터 상에서 결정된 경로점 및 방향전환점에서 주변 환경정보를 검색하고 저장하며, 이후 주행로봇이 동일한 지점에 도달했을 때, 자기위치인식과정과 이동 중의 오차보상과정을 통해 위치의 일관성 및 정확성을 유지할 수 있었다.

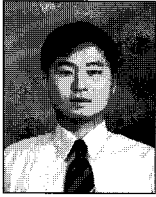
다만, 플랫폼 기구적 한계로 인해 실내나 지면이 평탄하고 비교적 지면마찰이 양호한 환경에서 성능을 유지할 수

있는 제약이 있으므로 본 시스템을 폐쇄공간이 아닌 실외환경에서도 적용이 가능하고, 평평한 지면뿐 아니라 경사면 및 높이가 불균형이 있는 지면에서도 기능수행이 가능하도록 입체 시각 시스템의 도입에 의한 지도구성 과정에의 정보 연동 및 환경정보 처리 성능 향상 및 공간지형 상의 SLAM 연구로 계속 진행할 예정이다.

참 고 문 헌

- [1] A. Elfes "Using Occupancy Grid" IEEE Journal of Computer, Vol. 22, No. 6 pp 46-57, 1989.
- [2] B.Kuipers, Y.T.Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," Journal of Robotics and Autonomous Systems, 1991.
- [3] S.Thrun, "Learning maps for indoor mobile robot navigation," Artificial Intelligence, 1999.
- [4] Tae-Bum Kwon, Jae-Bok Song, "Thinning-based topological map building for local and global environments," Journal of Control, Automation and Systems Engineering, Vol. 12, No.7, July 2006.
- [5] G. Weiss, E.Puttkamer, "A map based on laser scans without geometric interpretation," Intelligent Autonomous Systems, pp. 403-407, IOS Press, 1995.
- [6] M.Skubic, G.Chronis, P.Matsakis, J.M.Keller, "Generating linguistic spatial descriptions from sonar reading using the histogram of forces," Proc. 2001 IEEE Int. Conf. on Robotics and Automations, Seoul, Korea, 2001.
- [7] J. J. Leonard and H. F. Durrant-Whyte, "Directed sonar sensing for Mobile Robot Navigation," Kluwer Academic Publishers, Boston, MA 1992.
- [8] W.D.D.and T.hmidt. "Estimating the absolute position of a mobile robot using position probability grids," Proc.of the National Conference on Artificial Intelligence (AAAI), 1996.
- [9] Thomas Rüfer, "Using histogram correlation to create consistent laser scan maps," Proc. of the IEEE International Conference on Robotics Systems(IROS), EPFL, Lausanne, Switzerland, pp. 625-630, 2002.
- [10] Agostino Martinelli et al, "A relative map approach to SLAM based on shift and rotation invariants," Robotics and Autonomous Systems, Vol.55, pp.50-61, 2007.
- [11] C. C. Wang and C. Thorpe. "Simultaneous localization and mapping with detection and tracking of moving objects," Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2002.
- [12] M.Begum, G.K.I. Mann, R.G.Gosine, "Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots," Applied Soft Computing, Vol.8, pp. 150-165, 2008.

저 자 소 개



김수현 (金秀賢)

1975년 12월 18일생, 고려대학교 기계공학과 학사, 2003년 고려대학교 대학원 기계공학과 석사, 2005년 KAIST 기계공학과 박사과정수료, 2009년 2월 목원대학교 IT공학과 대학원 박사, 휴머노이드 로봇 전공. 현재 국립암센터 연구원.



양태규 (梁兌奎)

1958년 2월 7일생, 1982년 광운대학교 전자공학과 학사, 1984년 광운대학교 대학원 전자공학과 석사, 1989년 광운대학교 대학원 전자공학과 박사, 1991년~2006년 목원대학교 전자공학과 교수, 2007년~현재 목원대학교 지능로봇공학과 교수, 관심분야는 지능제어, 지능로봇 등.