자코비안과 퍼지 추론 시스템을 이용한 이동로봇의 주행문제에 관한 연구

Study on Mobile Robot's Navigation Problem Using Jacobian and Fuzzy Inference System

최 규 종*, 안 두 성
(Gyu Jong Choi and Doo Sung Ahn)

Abstract : In this paper, we propose the topological map building method about unknown environment using the ultrasonic sensors. An ultrasonic sensor inherently has the range error due to the specular reflection. To decrease this error, we estimate the obstacle states(position and velocity) using the local minimum sensor values and Jacobian. Estimated states are used to avoid the obstacles and build the topological map similar to the type that human being memorizes an environment. When a mobile robot is faced with three problems(corner way, cross way and dead end), it senses the movable directions using FIS(Fuzzy Inference System). Among these directions, it can select the target direction using binary decision tree(Turn Side Selector). Proposed algorithm has been verified with three simulations and three implementations.

Keywords : navigation, jacobian, fuzzy inference system, mobile robot

I. 서론

산업 자동화에서 핵심적인 생산수단으로 사용되던 매니퓰레이터(manipulator) 개념의 로봇으로부터 이제는 인간 삶의 질 향상에 따른 인간 친화적인 로봇의 중요성이 확대되고 있다. 인간 친화적인 로봇은 미지의 환경(unknown environment)에 대한 정보를 스스로 탐색(exploration)하여 학습(learning)하고 학습된 정보를 통해 판단(decision)하고 행동(behavior)[1]할 수 있어야 한다. 이러한 능력들을 구현하기 위해 첫 번째로 로봇은 미지의 환경에 대한 신뢰성 있는 정보를 획득하여 모델링할 수 있어야 하며, 두 번째로 획득된 정보를 바탕으로 하여 자신의 위치와 자세를 인식할 수 있어야 한다. 전자를 지도작성(map building), 후자를 자기위치인식(localization)이라고 한다.

지도작성(map building)과 자기위치인식(localization)을 수행하기 위해 우선적으로 신뢰성 있는 장애물 정보를 획득하여 로봇 주위의 상태를 인식해야 한다. 이동로봇에 사용되는 일반적인 센서들은 환경에 대한 영상정보를 받아들이는 시각센서(vision sensor)와 장애물까지의 거리정보를 알아내는 거리센서(ultrasonic sensor, infra radiation sensor)[2]가 사용된다. 특히 초음파를 이용해 거리를 측정하는 초음파 센서는 시각 센서를 사용하기에 제약이 많은 환경에서 사용함으로써 환경의 상태를 정확히 감지할 수 있다. 초음파 센서는 크기가 작고 저가이기 때문에 이동로봇에 널리 사용되고 있으며 측정범위가 넓고 다양한 물체와의 거리 측정이 가능하다는 장점이 있다. 그러나 초음파 센서는 그림 1과 같이 초음파를 발사한 후, 대상 물체에서 반사되어 되돌아오는 시간을 이용하여 거리를 계산하기 때문에 specular reflection[3-5]에 의한 거리값의 오차가 발생한다. 즉, 센서, r_6와 같이 장애물의 표면과 초음파의 반사각(θ_s)이 클수록 실제값 보다 큰 거리값이 읽혀진다.

그리고 지도를 작성하는 방법으로 크게 두 가지로 나눌 수 있다. 첫 번째는 환경을 그리드(grid)[6,7]로 분할하여 지도를 작성하는 방법과 두 번째는 토폴로지컬(topological)[8-10]한 형태의 지도를 작성하는 것이다. 전자는 간단하면서 정확한 지도를 작성할 수 있지만 큰 메모리 용량과 과도한 계산시간을 요구하여 비효율적이다. 후자는 효율적이고 적은 메모리 용량으로 지도를 작성할 수 있지만 정보부족으로 인한 자기위치인식에 있어서 문제가 발생할 수 있다.

따라서 본 논문에서는 이러한 문제들을 해결하기 위해 신호강도가 크고 상대적으로 장애물과의 반사각이 적으며

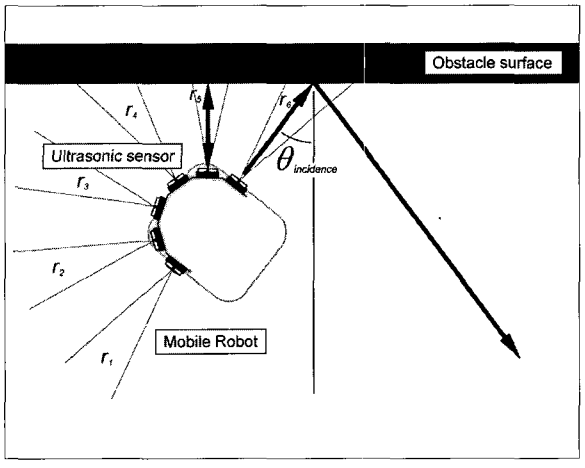

그림 1. 초음파 센서의 specular reflection.
Fig. 1. Specular reflection of ultrasonic sensor.

* 책임저자(Corresponding Author)
논문접수 : 2005. 12. 9., 채택확정 : 2006. 2. 9.
최규종 : 부경대학교 기계공학과(websignr@pknu.ac.kr)
안두성 : 부경대학교 기계공학부(dsahn@pknu.ac.kr)

로봇 주위에 장애물이 존재할 때 최소한 하나 이상 존재하는 국부최소(local minimum)에 해당하는 센서(r_5)정보와 매니퓰레이터의 자코비안을 이용하여 장애물의 상태정보를 실시간으로 추정한다. 추정된 상태정보를 이용하여 로봇은 장애물을 회피하고, 그래프 기법[11]과 방향결정기(turn side selector)를 이용하여 미지의 환경에 대한 지도를 간단(simple)하고 토폴로지컬(topological)한 형태로 작성한다.

II. 시스템 구성

이동로봇은 초음파 신호를 거리값으로 변환하여 로봇으로 전송하는 센서시스템(sensor system)과 로봇의 속도값을 이용하여 현재 위치를 추정하는 추측항법시스템(dead reckoning system)을 갖추고 있다. 그리고 감지된 정보를 이용하여 장애물 회피와 지도구축 및 자기위치인식을 위해 두 개의 레벨(level)로 나눈다. 장애물 회피는 하위레벨(low level)에 포함되며, 지도작성을 위해 필요한 결정, 탐색, 학습과 같은 지능적인 부분은 상위레벨(high level)에서 이루어진다. 이와 같은 구조의 세부적인 구성도는 그림 2와 같다.

1. 이동로봇 시스템

저가의 SRF04 초음파 센서(r_1~r_6) 6개로 이루어진 센서시스템과 현재 위치를 추정할 수 있는 추측항법시스템(dead reckoning system)을 갖추고 있다. 센서 시스템은 최소 및 최대 감지거리를 각각 10cm와 210cm로 제한하였으며, 로봇의 전방에 센서를 배치하여 진행방향에 대한 장애물을 보다 확실하게 인식할 수 있도록 하였다. 이와 같은 시스템을 시뮬레이션에서도 동일하게 적용하기 위해 그림 1과 같이 센서의 빔폭 내에서 로봇까지의 거리가 최소값에 해당하는 정보를 센서 데이터로 사용하였다. 추측항법시스템은 로봇에 부착된 양쪽 바퀴에 대한 회전속도를 이용하여 현재 로봇의 위치와 자세를 추정하였다.

2. 장애물의 자세추정

로봇 주위에 분포한 장애물들의 자세를 추정하기 위해 필요한 정보는 로봇의 현재 위치(X_r), 자세(d_r), 그리고 장애물까지의 거리정보(r_i)가 필요하다. 이러한 정보를 이용하여 절대 좌표계 상에서 장애물의 위치 정보를 계산한 후 로봇의 메모리에 실시간으로 저장한다. 저장된 전체 센서 데이터에서 현재 국부 최소값($r_{\min}$)을 가지는 센서에 해당하는 데이터 n개를 추출하여 (1)과 같은 선형최소자승법(linear least squares approximation)[12]을 이용하여 절대 좌표계를 기준으로 장애물의 자세를 추정한다.

$$\begin{aligned} E &= [f(x_1)-y_1]^2-[f(x_2)-y_2]^2+\cdots+[f(x_n)-y_n]^2 \\ &= [ax_1+b-y_1]^2+[ax_2+b-y_2]^2+\cdots+[ax_n+b-y_n]^2 \end{aligned} \quad (1)$$

$$\begin{aligned} S_{xx} &= \sum_{i=1}^{n} x_i^2, \quad S_x = \sum_{i=1}^{n} x_i \\ S_{xy} &= \sum_{i=1}^{n} x_i y_i, \quad S_y = \sum_{i=1}^{n} y_i \end{aligned} \quad (2)$$

여기서, (x_i, y_i)는 절대 좌표계상에서 장애물의 위치이며, $\frac{\partial E}{\partial a}=0$ 일 때의 a가 장애물의 자세이며, (2)의 4개 항을 이용하여 (3)과 (4)에서 구할 수 있다.

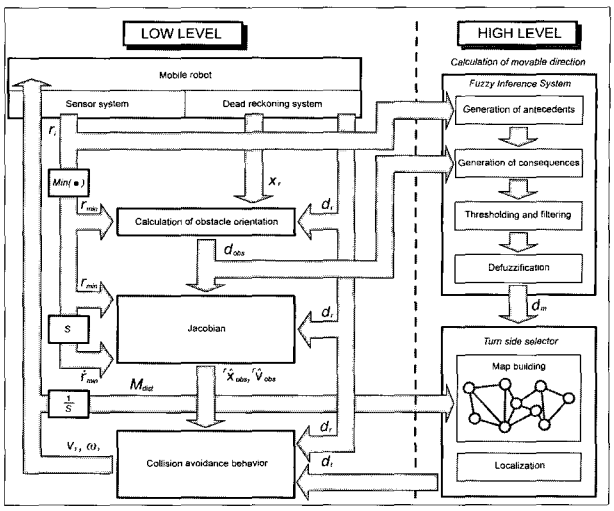

그림 2. 전체 시스템 구성도.

Fig. 2. Overall system configuration.

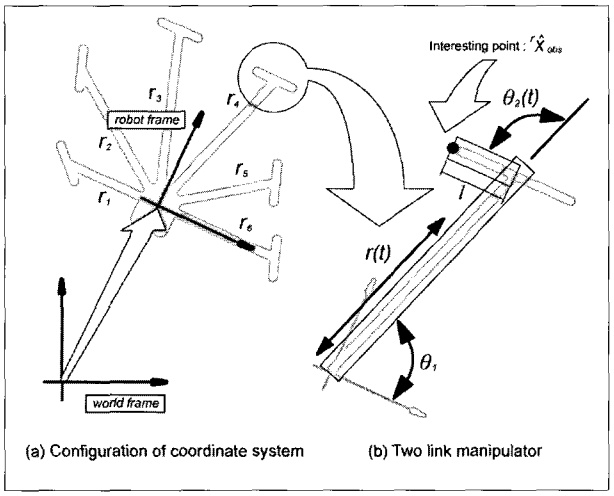

그림 3. 자코비안을 이용한 장애물 회피.

Fig. 3. Collision avoidance using Jacobian.

$$\begin{aligned} aS_{xx}+bS_x &= S_{xy} \\ aS_x+b(n) &= S_x \end{aligned} \quad (3)$$

$$a = \frac{nS_{xx}-S_xS_y}{nS_{xx}-S_xS_x} \quad (4)$$

3. 자코비안을 이용한 장애물 회피

본 논문에서는 로봇 주위의 환경정보를 획득하여 장애물을 효율적으로 회피하기 위해 자코비안을 이용한다. 먼저, 그림 3(a)와 같이 좌표계를 절대좌표계와 로봇좌표계로 구분하며, 이동 로봇에 부착된 6개의 센서들을 각각 하나의 2링크 매니퓰레이터로 가정한다. 그리고 (b)에서 명명된 변수에 대한 자코비안을 유도하여 관심점(interesting point)에 대한 위치와 속도 정보를 추정한다. 그림 3(b)에서 $r(t)$와 l은 각각 매니퓰레이터의 링크들로 가정한다. 이때 첫 번째 링크의 길이는 센서에 의해서 감지되는 거리이며, 두 번째 링크의 길이는 이동로봇의 속도에 따라 가변적으로 설정한

다. 즉, 로봇이 빠른 속도로 움직일 경우엔 l 을 크게 하고 느린 속도로 움직이는 경우엔 정밀한 움직임을 위해 링크 길이를 작게 설정하여 효율적으로 로봇의 움직임을 제어하고자 한다. 또한, 관절각 θ_1은 로봇에 부착된 센서의 배치 각도를 이용하여 구해지는데 이것은 로봇에 항상 고정되어 있기 때문에 상수가 된다. 그리고 $\theta_2(t)$는 절대좌표계를 기준으로 하여 표현된 장애물의 자세를 로봇좌표계로 변환하여 자코비안을 구함으로써 관심점의 위치정보($\widehat{{}^r x_{ds}}$)와 접근 속도($\widehat{{}^r v_{ds}}$)를 로봇을 기준으로 표현한다. 즉, 위치 정보는 현재 속도로 로봇이 진행할 때 상대적으로 "장애물이 로봇을 기준으로 얼마나 멀리 위치할 것이냐?"하는 장애물까지의 추정된 거리정보이며, 속도 정보는 상대적으로 "장애물이 얼마나 빠르게 접근할 것이냐?"하는 로봇을 기준으로 장애물의 추정된 접근 속도이다.

다음은 장애물의 접근속도정보를 추정하기 위한 자코비안, J 의 유도과정을 (5)~(9)까지 나타내었다.

$$\begin{aligned} x &= r(t)\cos(\theta_1) + a\cos(\theta_1 + \theta_2(t)) \\ y &= r(t)\sin(\theta_1) + a\sin(\theta_1 + \theta_2(t)) \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{x} &= \dot{r}(t)\cos(\theta_1) - a\,\dot{\theta}_2(t)\sin(\theta_1 + \theta_2(t)) \\ \dot{y} &= \dot{r}(t)\sin(\theta_1) + a\,\dot{\theta}_2(t)\cos(\theta_1 + \theta_2(t)) \end{aligned} \tag{6}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -a\sin(\theta_1 + \theta_2(t)) \\ \sin(\theta_1) & a\cos(\theta_1 + \theta_2(t)) \end{bmatrix} \begin{bmatrix} \dot{r}(t) \\ \dot{\theta}(t) \end{bmatrix} \tag{7}$$

여기서,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \dot{X}$$
$$\begin{bmatrix} \cos(\theta_1) & -a\sin(\theta_1 + \theta_2(t)) \\ \sin(\theta_1) & a\cos(\theta_1 + \theta_2(t)) \end{bmatrix} = J \tag{8}$$
$$\begin{bmatrix} \dot{r}(t) \\ \dot{\theta}(t) \end{bmatrix} = \dot{q}$$

라고 하면 다음과 같이 쓸 수 있다.

$$\dot{X} = J\dot{q} \tag{9}$$

$$\omega_r = K_1 d_e + K_2 r_e + K_3 \widehat{{}^r v_{obs}} \tag{10}$$

(10)에서 장애물을 회피하기 위해 3가지 정보를 이용하여 로봇의 각속도(ω_r)를 계산한다. 첫 번째 항은 로봇의 현재 진행 방향(d_r)과 목표 방향(d_t)의 오차, d_e 이며 이것은 목표방향으로 나아갈 수 있게 한다. 두 번째 항은 국부 최소값들의 상대적인 거리오차, r_e 이며, 세 번째 항은 장애물이 로봇에 접근하는 속도($\widehat{{}^r v_{ds}}$)를 나타낸다. 두 번째와 세 번째 항은 로봇 주위에 분포하는 장애물을 회피할 수 있게 한다. 그리고 거리오차(r_e)를 환경 속에 존재하는 통로의 폭에 따라 최대값을 적절히 제한할 필요가 있다. (11)은 로봇의 각속도에 따라 선속도를 계산하는 방법을 나타낸다.

$$v_r = v_{\max} \pm \frac{v_{\max}}{\omega_{\max}} \omega_r, \begin{cases} + & : \omega_r \le 0 \\ - & : \omega_r > 0 \end{cases} \tag{11}$$

4. 퍼지 시스템을 이용한 이동가능방향 추정

주행 시 발생할 수 있는 세 가지 문제(corner way, cross way and dead end)를 접했을 때 이동할 수 있는 방향들을

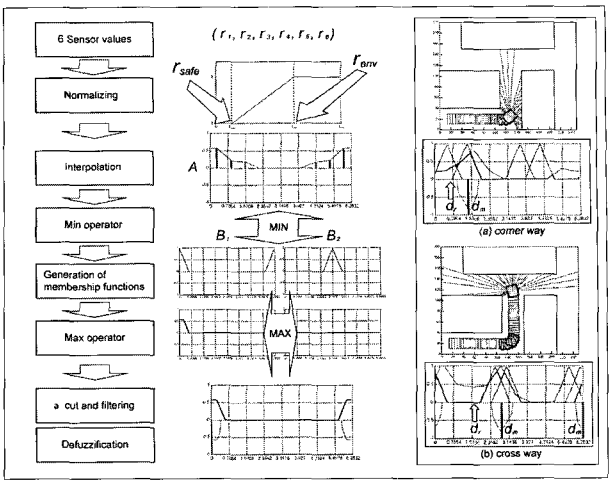

그림 4. 퍼지 추론 시스템을 이용한 이동가능방향 추정.

Fig. 4. Movable direction estimation using fuzzy inference system.

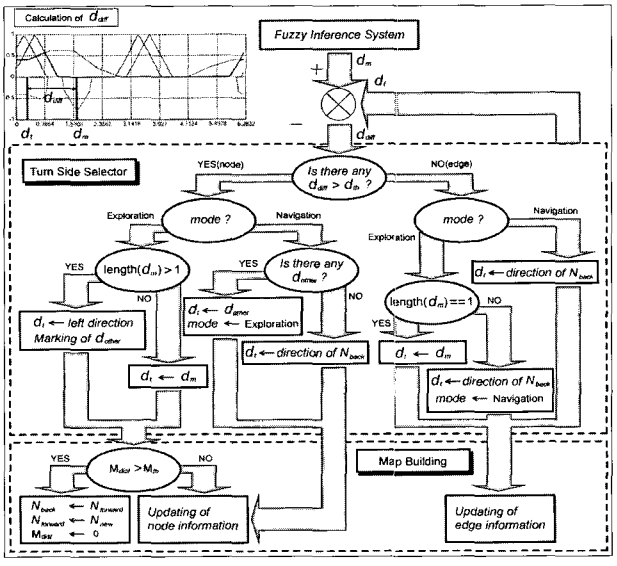

그림 5. 이진결정트리 구조를 이용한 진행방향결정기.

Fig. 5. Turn side selector using binary decision tree.

감지하고 원하는 방향으로 이동방향을 설정할 수 있어야 한다. 본 논문에서는 이러한 세 가지 사건들을 그림 4와 같은 퍼지 추론 시스템[13]을 이용한다. 퍼지의 입력값은 센서에 의해서 감지되는 거리값(r_1~r_6)을 이용하며 출력값은 로봇이 이동 가능한 방향(d_m)들이다. 그리고 r_{safe}와 r_{em}는 각각 환경 속에서 장애물의 분포정도와 실제 감지되는 거리의 최대값에 따라 제한하였다. 이와 같이 거리값의 정규화를 통해 로봇 주위의 상태를 나타내는 상태값(s)들로 변환시키고 변환된 정보는 보간(interpolation)을 통해 센서들 사이의 값들을 보정한다. 이것을 퍼지의 전제부 소속함수(A)로 사용한다. 결론부 소속함수(B)는 삼각형 소속함수(triangular MF)를 사용하고 그 수는 감지된 장애물들의 자세의 수(n)에 따라 가변적이다. 최종적으로 mamdani의 max-min composition방법을 사용하여 출력값을 결정하며 비

퍼지화(defuzzification)방법은 무게중심법(center of area)을 적용하였다. 그림 4(a)는 모퉁이에서 전제부 소속함수와 결론부 소속함수를 나타냈으며 출력은 움직일 수 있는 방향(d_m)을 나타내며 그림 4(b)는 교차로에서 그것을 나타낸다.

5. 진행 방향 결정기

진행 방향 결정기(turn side selector)는 두 가지 역할을 수행한다. 첫 번째는 현재의 위치가 노드(node) 혹은 에지(edge)인지를 판단하는 것이다. 즉, 퍼지의 출력 결과인 이동할 수 있는 방향(d_m)과 목표방향(d_t)의 차이값(d_{dif})을 이용하여 그 값이 d_{th} 이상일 경우 노드(node), 이하이면 에지(edge)로 판단한다. 두 번째는 교차로에서 진행방향을 결정하는 것이다. 현재 로봇의 이동 모드가 탐색모드(exploration mode)인지 주행모드(navigation mode)인지에 따라 로봇을 이동시킬 진행방향을 결정한다. 즉, 탐색모드일 경우 로봇은 왼쪽부터 시계방향으로 탐색방향을 결정하며, 주행모드일 경우 진행방향은 탐색이 되지 않은 방향을 결정하여 이동한다. 이와 같은 과정을 본 논문에서는 실시간처리를 위해 그림 5와 같이 이진결정트리(binary decision tree) 구조로 구성하였으며 총 7가지의 경우로 나누어 생각할 수 있다.

III. 지도작성과 자기위치인식

지도작성은 노드와 에지로 구성된 그래프 기법을 이용하며, 그림 5와 같은 이진결정트리 구조를 이용하여 진행방향을 결정한다. 또한, 로봇이 이동시 필요한 정보들을 획득하고, 획득된 정보를 분석하여 토폴로지컬한 지도를 작성한다. 이때 작성되는 지도의 노드에 저장되는 정보는 이웃노드들과 그 노드들까지의 방향과 거리 정보이며, 에지는 노드들 간의 상호 연결성을 나타낸다. 이러한 정보들의 상호 상관관계를 조사함으로써 현재 로봇의 위치에 해당하는 노드 혹은 에지를 알아낼 수 있다.

IV. 시뮬레이션 및 실험결과 분석

시뮬레이터(simulator)는 매트랩(matlab)[14]을 사용하여 전체 환경을 가로 450, 세로 200cm로 각각 설정된 환경을 적용하였으며 실험은 그림 6과 같은 실험실에 자체 제작한 이동로봇(WEBO)을 사용하였으며, 제시된 알고리즘(algorithm)

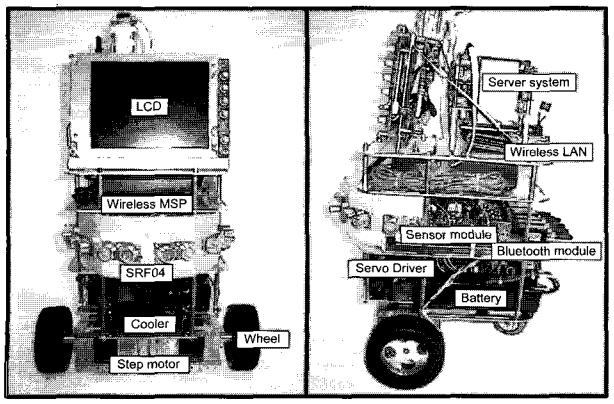

그림 6. 실험용 이동로봇(WEBO).
Fig. 6. Experiment mobile robot (WEBO).

의 검증을 위해 3가지 시뮬레이션과 실험을 수행하였다. 먼저 이동가능방향을 추정하고 그것에 따라 지도를 작성하였으며, 마지막으로 작성된 지도를 이용하여 시작 위치와 목표 위치가 주어졌을 때 목표위치까지 주행을 수행하였다.

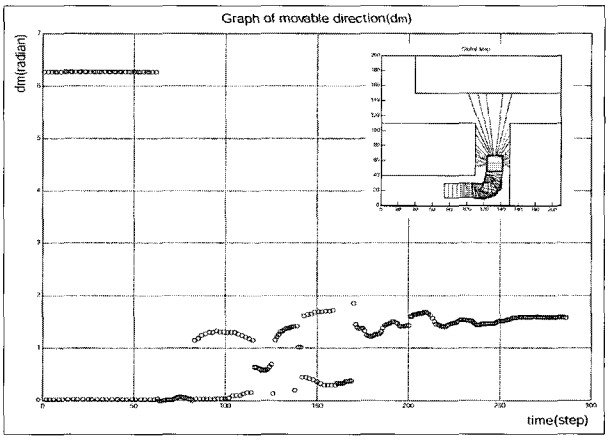

그림 7. 모퉁이에서의 이동가능 방향(시뮬레이션).
Fig. 7. Movable direction at corner way(simulation).

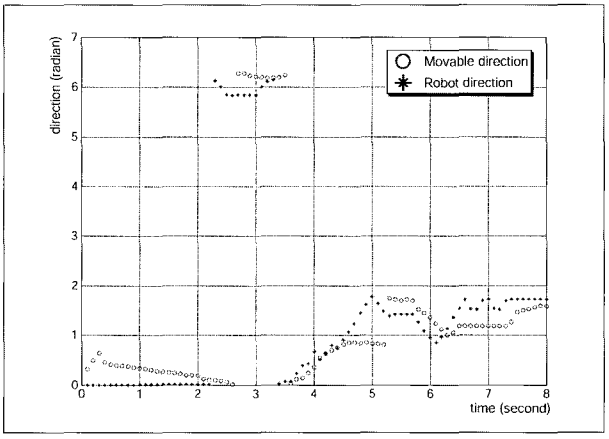

그림 8. 모퉁이에서 이동가능 방향(실험).
Fig. 8. Movable direction at corner way(implementation).

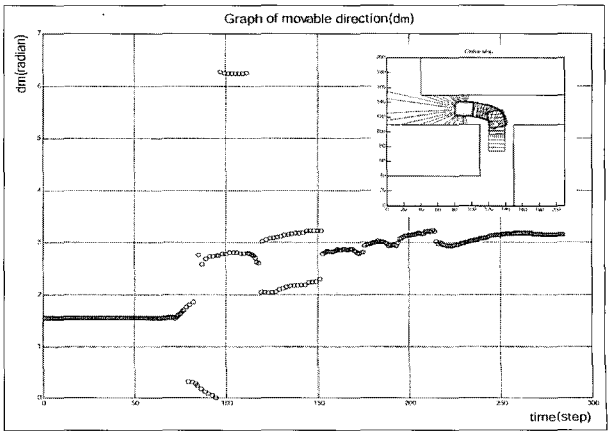

그림 9. 교차로에서의 이동가능 방향(시뮬레이션).
Fig. 9. Movable direction at cross way(simulation).

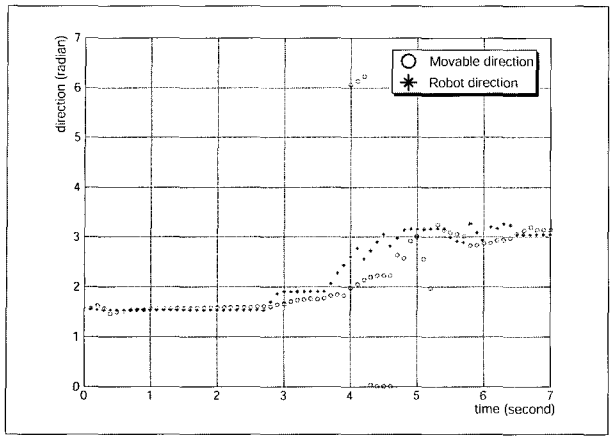

그림 10. 교차로에서 이동가능 방향(실험).

Fig. 10. Movable direction at corner way(implementation).

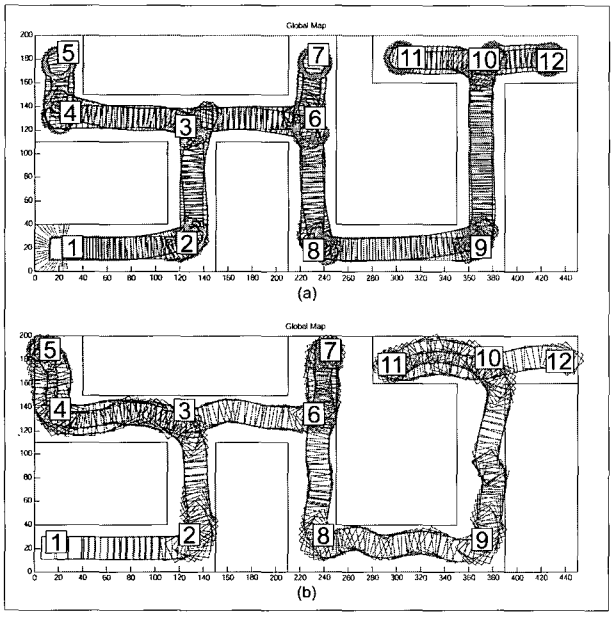

그림 11. 지도작성 시뮬레이션 (a)와 실험결과(b).

Fig. 11. Map building results simulation(a) and implementation (b).

1. 이동가능방향 추정

두 가지 경우에 대해 이동가능방향(d_m)을 추정하였다. 그림 7과 8은 0의 방향에서 $\pi/2$ 방향으로 이루어진 모퉁이(corner)를 이동하는 경우 각각 시뮬레이션과 실험결과를 나타내었으며, 그림 9와 10은 $\pi/2$의 방향에서 0과 π 방향으로 이루어진 교차로(cross way)에서 그것을 나타내었다.

그래프의 결과에서 볼 수 있듯이 그림 7과 8은 과도기적인 상태에서 0~$\pi/2$ 범위 내에서 비교적 통로와 유사한 방향을 나타내고 있다는 것을 알 수 있다. 또한, 그림 9와 10은 교차로를 통과할 때 다른 범위의 각도값이 존재하는 것을 볼 수 있는데, 이것은 교차로 이동 시 진행방향과 다른 또 하나의 통로가 존재한다는 것을 알 수 있으며 로봇이 막다른 길에 도달했을 때 되돌아와서 다시 탐색 작업을 수행할 수 있는 정보가 된다.

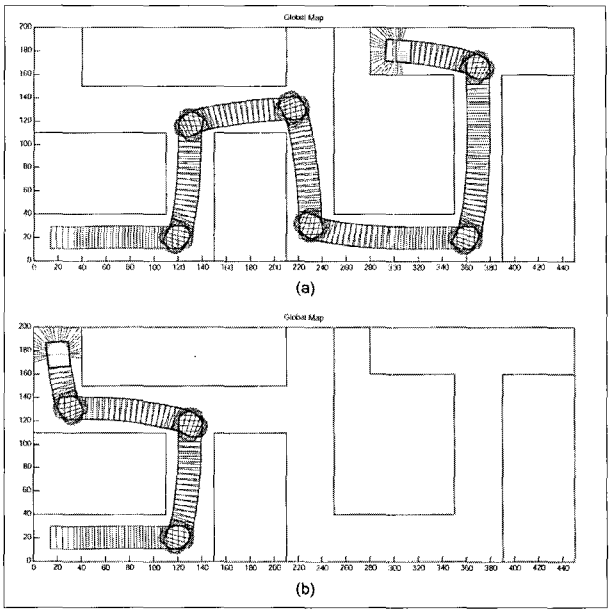

그림 12. 작성된 지도 정보를 이용한 주행(시뮬레이션).

Fig. 12. Navigation using the constructed map(simulation).

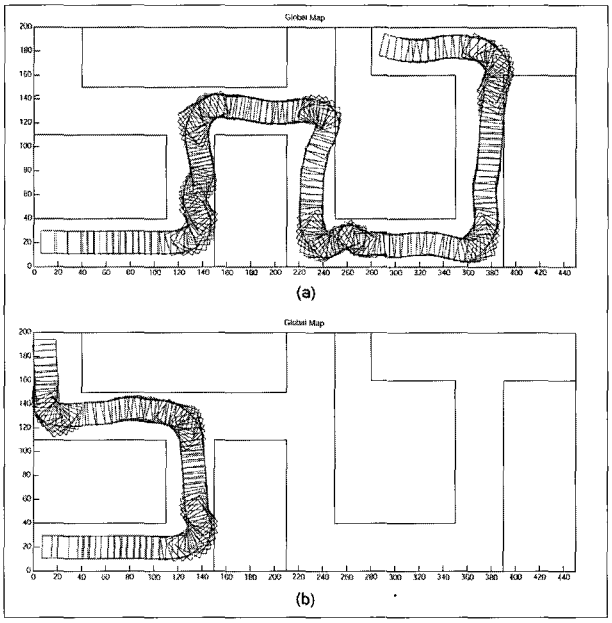

그림 13. 작성된 지도 정보를 이용한 주행(실험).

Fig. 13. Navigation using the constructed map(implementation).

2. 지도작성

시뮬레이션과 실험은 결과분석을 위하여 동일한 환경의 크기와 장애물을 적용하였다. 4.1절에서 검증된 방법을 적용하여 지도를 작성한 결과를 그림 11에서 나타내었다. 그림 11에 나타낸 내부의 숫자는 지도작성 시 생성된 순서를 나타내는 노드의 번호이다. 초기에 탐색모드로 동작하여 이동하며, 5-4-3과 같이 탐색이 완료된 통로는 주행모드로 빠르게 움직인다. 이동로봇이 지도작성을 시작하는 위치는 (20, 20)이며 이것을 1번 노드로 설정하여 탐색을 12번 노드까지 수행한다. 탐색 시 방문하는 노드들을 스택(stack)에

저장함으로써 탐색이 완료된 후 혹은 막다른 길에 도달한 경우 스텍에 저장된 정보를 이용하여 다시 시작 위치 혹은 탐색하지 않은 경로로 돌아올 수 있다. 탐색이 완료된 후 미지의 환경은 노드와 에지로 이루어지진 그래프의 형태를 이루며, 그래프에 포함된 각 노드는 이웃한 노드, 방향 그리고 거리에 대한 정보를 각각 포함한다.

3. 주행

그림 12와 13(a)는 시작노드를 1번 목표노드를 11번으로 지정하였을 때 작성된 지도상에서 경로를 설정하여 주행한 결과를 보여주고 있다. 환경에 대해 작성된 지도를 가지고 있는 상황에서는 탐색과정에서 코너 혹은 교차로에서 볼 수 있었던 복잡한 움직임 보다는 노드에 포함된 정보를 이용함으로써 빠른 판단에 의해 부드러운 움직임을 나타내고 있다. 그림 12와 13(b)는 목표노드를 5번 노드로 지정하여 얻은 결과를 나타내었다.

4. 결과 분석

본 논문에서 제시한 알고리즘을 검증하기 위해 사용된 장애물은 복잡한 형태가 아니라 통로의 벽과 같이 단순한 형태를 사용하였다. 또한, 실험에서 사용된 환경은 시뮬레이션 결과와 비교하기 위해 동일하게 장애물들을 배치하였다. 첫 번째 시뮬레이션과 실험에서 추정한 이동가능방향은 장애물의 상태를 나타낸다. 즉, 이것은 현재 로봇 주위에 장애물이 어떤 자세로 놓여 있는가를 나타내며, 미지의 환경에 대한 지도를 작성하는데 있어서 중요한 정보가 된다.

시뮬레이션과 실험결과를 비교했을 때 로봇의 하드웨어적인 특성과 센서의 노이즈 영향으로 다소 복잡한 움직임을 나타내고 있으며, 그로 인해 실험결과에서의 이동 거리는 시뮬레이션 결과에 비해 크게 나타났다. 또한 로봇에 부착된 바퀴의 슬립(slip)으로 인해 실제 진행방향과 추정된 진행방향은 약간의 차이를 보이고 있으며, 이것은 이동거리가 길면 길수록 누적되어 나타난다는 것을 실험결과로부터 알 수 있다. 그러나 이러한 오차는 로봇이 전체 환경을 잘못 인식할 정도로 큰 값은 아니다. 왜냐하면 표 1과 같은 토폴로지컬한 형태의 지도 정보를 가지고 있기 때문에 대략적인 방향과 이동거리를 이용해서 충분히 목표 위치에 도달할 수 있으며, 이동시 발생할 수 있는 여러 가지 오차는 재방문을 통한 학습으로 줄일 수 있다.

표 1. 지도작성 후 노드 정보.
Table 1. Node information after map building.

N	N_{nbr}	d	M_{dist}
1	2	0.0049	102.4521
2	1	3.1465	102.4521
	3	1.5995	115.5316
3	2	4.7411	115.5316
	4	3.1707	115.0013
	6	0.0155	104.1010
4	3	0.0291	115.0013
	5	1.6512	65.0167
5	4	4.7928	65.0167
6	3	3.1571	104.1010
	7	1.5101	56.1824
	8	4.6331	114.0951
7	6	4.6517	56.1824
8	6	1.4915	114.0951
	9	6.2301	144.5112
9	8	3.0885	144.5112
	10	1.4120	157.8571
10	9	4.5536	157.8571
	11	3.0813	73.0701
	12	6.2714	55.8902
11	10	6.2229	73.0701
12	10	3.1298	55.8902

V. 결론

이동로봇이 미지의 환경에서 지능적이고 자율적인 주행을 수행하기 위해 신뢰성 있는 데이터가 필수적이다. 그러나 본 논문에서 사용한 초음파 센서는 본질적으로 specular reflection에 의해 거리값의 오차가 발생한다. 이러한 오차를 줄이기 위해 국부최소(local minimum)에 해당하는 센서정보들과 자코비안을 이용하여 장애물의 상태정보를 추정하였으며, 또한 추정된 상태정보는 장애물을 회피하는데 효율적이라는 것을 시뮬레이션과 실험을 통해 검증하였다. 또한 이것을 이용하여 교차로에서 로봇이 이동할 수 있는 방향들을 퍼지추론 시스템으로 계산하였으며, 이동해야할 목표방향을 이진결정트리 구조를 가지는 방향결정기를 통해 결정하였고 동시에 이러한 방향정보들을 이용하여 토폴로지컬한 지도를 작성하였다. 비록 작성된 지도는 실제 시스템에서 발생하는 오차로 인해 시뮬레이션과 실험결과에서 차이를 보이지만 이것은 알고리즘상의 문제라기보다는 하드웨어적인 문제로 인해 발생한 것으로 분석할 수 있으며, 실제 주행 시 목표 위치에 도달하지 못하게 하는 원인은 되지 않는다는 것도 실험을 통해 검증하였다.

참고문헌

[1] T. fukuda and N. Kubota, "An intelligent robotic system based on a fuzzy approach," *Proceedings of the IEEE*, vol. 87, pp. 1448-1470, Sept. 1999.

[2] S. H. Park and B. H. Lee, "Practical environment modeling based on a heuristic sensor fusion method," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pp. 200-205, April, 2004.

[3] A. C. Victorino, P. Rives, and J.-J. Borrelly, "Mobile robot navigation using a sensor-based control strategy," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, vol. 3, pp. 2753-2758, May, 2001.

[4] F.-J. Zhao, H.-J. Guo, and K. Abe, "A mobile robot localization using ultrasonic sensors in indoor environment," *IEEE International Workshop on Robot and Human Communication*, pp. 52-57, 1997.

[5] Z. Yi, H. Y. Khing, Chua Chin Seng, and Z. X. Wei, "Multi-ultrasonic sonic fusion for mobile robots," *Proceedings of the IEEE Intelligent Vehicles Symposium*

2000, pp. 387-391, October 3-5, 2000.

[6] D. H. Lee, W. J. Chung, and M. S. Kim, "Autonomous map building and smart localization of the service robot PSR," *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.* pp, 454-459, vol.1 Oct. 27-31 2003.

[7] C. Jennings, D. Murray, and J. J. Little, "Cooperative robot localization with vision-based mapping," *Robotics and Automation*, Vol. 4, pp. 2659-2665, 10-15 May 1999.

[8] C. Howie, K. Ilhan, and B. Joel, "Mobile robot navigation: issue in implementating the generalized vornoi graph in the plane," *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 241-248, Dec. 1996.

[9] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Simulataneous localization and map building: A global topological model with local metric maps," *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 421-426, Oct. 29-Nov. 03, 2001.

[10] C.-H. Choi, J.-B. Song, W. J. Chung, and M. S. Kim, "Topological map building based on thining and Its application to localization," *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 552-557, October, 2002.

[11] E. Horowitz, S. Sahni, and A. Freed, "Fundamentals of data structures in C," *Computer science press*, 1992.

[12] L. V. Fausett "Applied numerical analysis using Matlab," *Prentice hall*, 1999.

[13] K. R. S. Kodagoda, W. S. Wijesoma, and E. K. Teoh, "Fuzzy speed and steering control of an AGV," *IEEE Transaction on Control Systems Technology*, vol. 10, pp. 112-120, Jan. 2002.

[14] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft-computing," *Prentice hall.*, 1997.

최 규 종

1997년 부경대학교 공과대학 기계공학과(공학사, 기계공학). 1999년 부경대학교 공과대학 기계공학과(공학석사, 생산자동화). 2005년 부경대학교 공과대학 기계공학과(공학박사 재학, 지능제어). 관심분야는 인공지능, 이동로봇.

안 두 성

1978년 서울대학교 공과대학 기계공학부(공학사, 기계공학). 1980년 KAIST 생산공학부(공학석사, 생산공학). 1992년 KAIST 정밀공학부(공학박사, 정밀공학). 관심분야는 생산자동화, 인공지능, 로봇제어.