

論文96-33B-12-1

# 거리 형태 히스토그램을 이용한 이동로봇의 장애물 회피 주행 (Obstacle Avoidance Navigation Using Distance Profile Histogram)

金賢泰\*, 盧興植\*, 曹泳完\*, 朴玟用\*

(Hyun Tae Kim, Heung Sik Noh, Young Wan Cho, and Mignon Park)

## 요 약

초음파 센서를 이용하여 미지의 장애물이 분포하고 있는 환경에서 실시간으로 장애물을 회피하며 목적지까지 주행을 제어하는 DPH(Distance Profile Histogram)방법을 제안한다. 본 논문에서 제안된 방법은 초음파 센서로 관측한 정보를 이용하여 환경지도를 작성하고 존재 가능성이 높은 장애물들의 위치관계를 나타내는 DPH를 이용하여 이들을 회피하며 주행할 수 있는 경로를 생성한다. DPH방법은 2차원의 카테시안 격자(Cartesian Grid)를 환경 모델(World Model)로 사용하고 있다. 제안된 제어기는 상위의 DPH를 이용한 지역 경로 계획 알고리즘, 중간계층의 위치 이동제어기(Point-to-Point Translation Controller), 그리고 최하위의 속도 서보로 구성되는 계층적인 구조로 구현된다. 제안된 알고리즘의 타당성과 성능을 컴퓨터 시뮬레이션과 실제 주행 실험을 통하여 검증한다.

## Abstract

A new local path planning algorithm using DPH(Distance profile Histogram) is suggested in this paper. The proposed method makes a Grid type world map using distance values from multiple ultrasonic sensors and generates local points through which the mobile robot can avoid obstacles safely. The DPH(Distance Profile Histogram) represents geometrical arrangement of obstacles around the robot in the local polar coordinate system which is assumed to be attached to the robot. To control robot's navigation, a three-layered control structure is adopted. The proposed local path planning algorithm is placed on the top level. And a point-to-point translation controller takes the middle level. The bottom level consists of a velocity servo and sonar driver modules which take charge of driving physical hardwares. The validity of the proposed method is demonstrated through several experiments.

## 1. 서 론

자율 이동 로봇은 센서를 이용하여 주위의 환경을 파악하고 스스로 결정을 내리면서 주어진 일을 자율적으로 수행하는 시스템이다. 이러한 자율 이동 로봇은 센서정보를 종합하여 현재 상황에 가장 적합한 기계적인 조작량을 결정하는 의사결정 구조를 가지고 있어야 한다. 본 논문에서는 실시간 처리성이 만족되는 초음파

센서를 사용하여 환경을 파악한다. 초음파 센서의 거울 현상(Mirror Effect)이 만들어 내는 문제점을 최소화하기 위해 9개의 초음파 센서 어레이를 로봇의 정면에 원형으로 설치하였다. 우리의 이전 연구에서는 초음파 센서 정보를 장애물에 접하는 접선 선분(Tangential Line Segment)으로 모델링하는 방법<sup>[1]</sup>과 장애물면의 존재범위와 방향을 표현하는 타원으로 모델링하는 방법<sup>[2]</sup>을 사용하였다. 이 방법들은 일단 환경 모델이 구해지면 복잡한 장애물을 회피하는 경로를 쉽게 구할 수 있다는 장점이 있다. 그러나 모델을 구성하는데 비교적 많은 시간이 요구되기 때문에 실시간 제어를 위

\* 正會員, 延世大學校 電子工學科

(Dept. of ElectronicS Engineering, Yonsei Univ.)

接受日字:1996年3月14日, 수정완료일:1996年11月30日

해서 3-4초 동안 누적된 센서 정보밖에 활용하지 못하는 문제가 있었다. 또한 시간에 따라 감소하는 가중치를 사용하기 때문에 현재 발생한 에러에 민감해 질 수 있는 잠재적 위험성을 내포하고 있다. 이러한 문제점들을 보완하기 위해서 본 논문에서는 격자형태의 환경 모델(World Model)<sup>13, 10)</sup>을 사용한다.

격자형 환경 모델을 이용하는 방법으로 장애물의 존재 확률을 이용하여 환경 지도를 작성하는 확실성 격자 지도(Certainty Grid Map) 방법<sup>13, 7)</sup>과 확률값의 결정에 베이시안 규칙(Basian Rule)을 사용한 방법<sup>8, 9)</sup>이 제안되었다. 또한 확실성 지도를 이용하여 작성된 환경 지도를 이용하여 전역 경로를 설계하는 방법<sup>17)</sup>도 제안되었다. 그러나 위의 방법들은 실시간으로 장애물을 회피하는 방법을 다루지는 못했다. 격자형 환경 모델을 이용한 실시간 장애물 회피를 위하여 Borenstein 과 Koren은 VFF(Virtual Force Field) 방법을 원격 조정 로봇의 장애물 회피 주행에 적용하였다.<sup>14)</sup> 또한 위치 힘장(Potential Field) 방법의 문제점을 보완하는 VFH(Vector Field Histogram)방법<sup>15, 7)</sup>으로 자율 이동 로봇의 실시간 장애물 회피 주행을 성공적으로 실현하였다.

본 논문에서는 장애물의 기하학적 배치 정보를 활용하기 위해 거리 형태 히스토그램(Distance Profile Histogram) 방법을 제안한다. 이 방법은 확실성 지도 방법을 단순화시킨 CCG(Cumulative Certainty Grid)를 기반으로 장애물이 점유하고 있는 거리값을 히스토그램의 형태로 표현하여 장애물의 존재 위치 및 평균 거리 정보와 같은 기하학적 정보를 이용해서 장애물을 회피하며 주행할 수 있는 지역 경로를 생성하게 된다. 이 알고리즘은 주행 제어기의 최상위를 차지하며, 중간 계층의 위치 이동 제어기 및 최하위의 속도 서보와 협력하여 이동 로봇의 주행을 제어하게 된다. 먼저 2장에서는 주행제어 시스템의 기본이 되는 안정한 위치 이동 제어기의 설계 방법을 설명하고 3장에서 DPH를 이용한 경로 계획 알고리즘을 설명한다. 4장에서는 실제로 구현된 제어기의 구조와 실험 결과를 보이고 고찰한 뒤 5장에서 결론을 맺는다.

## II. 위치 이동 제어기의 설계

본 장에서는 안정한 위치 이동 제어기의 설계 방법을 설명하고 위치 이동 실험의 결과를 보인다.

### 1. 이동 제어기의 설계

2차원 평면으로 구성된 전역 좌표계에서 이동로봇의 상태를 다음과 같이 위치와 방향을 갖는 벡터로 정의한다.

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

일반적으로 이동로봇의 운동은 두 개의 제어 입력인 선형속도  $u$ 와 각속도  $\omega$ 에 의해 제어되는 시스템으로 모델링된다.<sup>11), 13), 115)</sup> 이때 이동로봇의 동역학 방정식은 자코비안 행렬  $J$ 에 의해 다음과 같이 표현할 수 있다.

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = Jq = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} q \quad (2)$$

$$q = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

식 (2)와 같은 시스템은 논 홀로노믹 제약을 가지며 Brockett<sup>11)</sup>의 정리에 따라  $[0, 0, 0]^T$ 을 점근적으로 안정시키는 미분가능(smooth)하고 시불변(time invariant)인 피드백 제어가 존재하지 않게 된다. 이러한 논 홀로노믹 시스템을 제어하기 위한 여러 가지 방법들이 제안되어 왔다.<sup>17), 18)</sup> 특히 Aicardi<sup>18)</sup>는 상태 공간의 변환을 사용하여 안정성을 보장해주는 비선형 제어기를 설계할 수 있음을 보여 주었다.

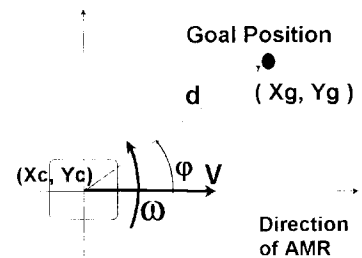


그림 1. 오차의 정의

Fig. 1. Definition of Error.

본 논문에서는 현재 위치로부터 목표지점까지 이동하는 제어기를 설계하기 위해 그림 1과 같이 정의된 오차 공간상에서 상태 방정식을 유도하고 Aicardi의 방법을 적용하여 이동로봇의 중심점  $(x, y)$ 에 대한 안정한 이동 제어기를 설계한다. 그림 1로부터 거리 오차

d 와 방향 오차  $\varphi$  에 대해 오차의 동역학 방정식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} \dot{d} &= -v \cos \varphi \\ \dot{\varphi} &= -\omega + v \frac{\sin \varphi}{d} \end{aligned} \quad (4)$$

위와 같은 운동방정식으로 표현되는 이동로봇트가 현재 위치  $(X_c, Y_c)$ 로부터 목표점  $(X_g, Y_g)$ 까지 이동하기 위해서는  $[d, \varphi]^T$  를  $[0, 0]^T$ 로 수렴시키는 제어 입력  $[v, \omega]^T$ 를 입력해야 한다. 아주 짧은 시간내에서는  $d = 0$  이 될 수 없다는 가정하에서 Aicardi의 방법에 따라 제어기를 설계하기 위한 Lyapunov 후보 함수를 다음과 같이 정의한다.

$$V = V_1 + V_2 = \frac{1}{2} d^2 + \frac{1}{2} \varphi^2 \quad (5)$$

식 (5)를 사용하여 V의 시간에 따른 도함수를 구하면 다음과 같다.

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = (-v \cos \varphi) + (-\varphi \omega + \frac{v \varphi \sin \varphi}{d}) \quad (6)$$

이제 피드백 제어 입력 v와  $\omega$ 를 다음과 같이 설계한다.

$$\begin{aligned} v &= k_1 d \cos \varphi & ; k_1 > 0 \\ \omega &= k_2 \varphi + k_1 \cos \varphi \sin \varphi & ; k_2 > 0 \end{aligned} \quad (7)$$

식 7을 6에 대입하면  $\dot{V}_1$ 과  $\dot{V}_2$ 는 다음과 같이 항상 0보다 작은 값을 갖게 된다.

$$\dot{V}_1 = -k_1 d^2 \cos^2 \varphi \leq 0, \quad \dot{V}_2 = -k_2 \varphi^2 \leq 0 \quad (8)$$

식 (8)로부터 V가 하한값을 가지며,  $[d, \varphi]^T$  가  $[0, 0]$ 인 경우를 제외하고는 항상  $\dot{V} < 0$  를 만족하므로 식 (7)에 의한 제어 입력  $[v, \omega]^T$ 에 의해 이동로봇트는 목표지점까지 안정적으로 접근해 가는 것을 보장해 준다. 이제 상위의 경로 계획기로부터 목표점이 주어지면 그 점으로 수렴하기 위한  $[\dot{v}, \dot{\omega}]^T$ 를 계산한 뒤 하위의 모터 속도 서보에 전달되는 좌,우 모터의 속도 명령을 다음의 식으로 계산한다.

$$\begin{bmatrix} V_{ref\_r} \\ V_{ref\_l} \end{bmatrix} = \begin{bmatrix} 1 & \frac{d}{2} \\ 1 & -\frac{d}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (9)$$

여기서 d는 이동로봇트의 두 바퀴 사이의 거리이다.

위와 같이 설계된 이동제어기를 실제 로봇에 구현하여 실험하였다. 이 제어기는 10ms로 동작하는 PID 속도 서보(servo)보다 10배 느리게 동작하며 추종해야 할 기준 속도 명령을 속도 서보에 발생한다. 아래의 그림 2는 서로 다른 8개의 점에서 출발하여 원점으로 이동할 때의 이동 궤적과 초기 위치(2, 2, 90°)에서 최고 직진 평균 속도(Maximum Linear Average Speed)를 0.3m/sec로 제한했을 때, 모터 서보에 발생한 속도 명령 데이터를 보이고 있다. 실험 결과 데이터는 이동로봇트의 측정된 속도값으로부터 변화된 위치를 추정하는 추측항법<sup>[16]</sup>에 의해 계산된 좌표값이며 실제 이동로봇트가 이동할 때 발생하는 슬립이 고려되지 않은 계산상의 결과이다. 슬립에 의해 발생하는 오차는 반복 실험 결과 0.1%(0.1m/10m)미만 이었다. 사용된 제어기의 계수  $k_1$  과  $k_2$  는 제어기의 샘플주기 100ms에 대하여 실험적으로 설정하였으며, 각각 0.5, 6.0의 값을 사용하였다. 이 값은 수렴 속도보다 수렴의 안정성과 궤적의 부드러움에 초점을 맞추어 설정된 값이다.

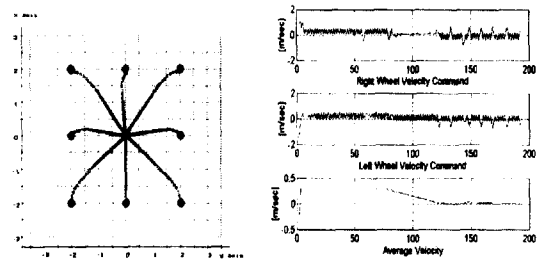


그림 2. 이동제어기에 의한 위치 이동 실험  
Fig. 2. Potin to Point Navigation Result.

### III. DPH를 이용한 지역 경로 계획 알고리즘

이 장에서는 환경모델 및 DPH(Distance Profile Histogram)를 만드는 방법과 지역경로를 설계하는 알고리즘을 설명한다.

#### 1. 격자형 환경 모델의 구성

장애물의 존재 가능성을 표현하는 확실성 격자(Certainty Grid) 방법은 Elfes, et. al.<sup>[3]</sup>에 의해 제안되었다. 이 방법은 개개의 격자 셀(Grid Cell)에 장애물이 존재하는 확률값을 할당하는 것으로서 확률을 계산하는 변형된 방법들<sup>[13] [16] [17]</sup>이 제안되어 왔다. 확률값을 할당하는 방법에 따라 작성된 환경 지도의

정확도와 특성이 달라진다. 격자형 환경 지도를 이용하여 실시간 장애물 회피 주행을 하고자 하는 경우에는 장애물의 정확한 모양보다는 장애물의 존재 유무와 거리를 정확히 아는 것이 더욱 중요하다. 특히 획득된 센서 정보를 경로 계획에 활용할 수 있도록 빠른 시간내에 환경 모델을 구성하는 것이 필수적이다.

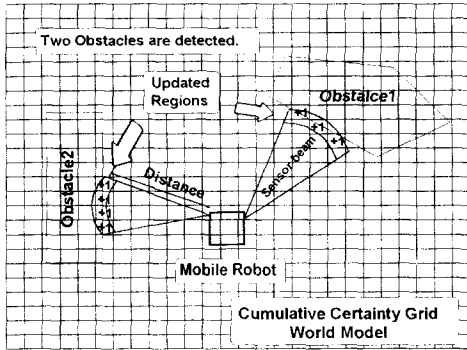


그림 3. 센서 정보로 갱신되는 환경영역  
Fig. 3. Updated Cells by Sensor Information.

이것은 실시간으로 주행하는 경우 자세한 부분까지 정확한 지도를 얻기가 어렵기 때문이며 대략적인 환경 정보밖에 활용하지 못하는 경우에도 안전한 경로를 계획할 수 있어야 하기 때문이다. 본 논문에서는 빠른 시간내에 확실성 지도를 얻기 위하여 단순화된 CCG (Cumulative Certainty Grid) 방법을 사용한다. 이와 유사한 방법을 Borenstein 과 Koren<sup>[5-7][14]</sup>이 채택한 바 있다. 본 논문에서 사용한 방법은 조건 확률의 개념을 응용한 방법으로서, 그림 3에서와 같이 관측된 거리에 대하여 콘(corn) 모양의 초음파 센서 빔 영역 안에서 장애물이 존재할 확률이 높은 영역과 겹치는 모든 격자 셀의 값을 1씩 증가시키는 방법이다. 이 방법을 사용하면 장애물의 모서리 부근에서 실제 장애물을 약간 벗어나는 격자 셀까지 값을 할당하게 되므로 장애물 회피시 모서리에 충돌하지 않고 주행할 수 있는 경로를 생성할 수 있다. 이 방법의 잠재적인 단점은 거리가 먼곳에 있는 작은 장애물이나 장애물의 에지(edge)가 검출된 경우 실제 필요한 영역보다 더 넓은 영역을 장애물이 점유하는 영역으로 갱신해 버릴 수 있다는 점인데, 4장에서의 실험 결과에서처럼, 실제 환경에서 이동하면서 측정을 하는 경우 거리가 가까워질수록 더 좁은 영역의 격자를 갱신하게 되므로 적절한 임계치를 적용할 경우 장애물 회피 주행에 큰 영향을 미치지 않게 된다.

## 2. 거리 형태 히스토그램 (Distance Profile Histogram)

DPH는 다음에 설명하는 몇 개의 단계를 거쳐 만들어진다. 먼저 그림 4와 같이 전체 환경 지도 위에서 로봇의 현재 위치를 중심으로 반경 R의 Active Circle을 설정한다. Active Circle을 만들어 이 원안의 데이터만을 처리하는 것은 계산 시간을 감소시켜 실시간 처리를 유리하게 한다. 이 Circle은 1° 간격으로 분할되는 360개의 섹터로 나누어진다. 이 섹터들의 간격은 처리 시간의 향상을 위해 더 넓게 설정되어도 무방하며, 실제 센서의 배치를 고려하여 360° 보다 작은 영역으로 설정될 수도 있다. 본 논문의 실험에서는 270개의 섹터를 사용하고 있다. R은 설계 변수이며 본 논문에서는 1.5m를 사용하고 있다. 이제 Active Circle안의 각 섹터의 COD(Cumulative Obstacle Density)와 DP (Distance Profile)를 다음의 식으로 구한다.

$$COD(i) = \sum_{k=1}^n C_i(k)/n \quad (10)$$

$$DP(i) = \min \| C_R - C_i^k \| \quad (11)$$

, for all k in i th sector

여기서  $C_i(k)$ 는 i번째 섹터와 겹치는 0이 아닌 값을 갖는 k번째 CCG의 값이고  $C_R$ 는 현재 이동로봇의 중심점의 좌표,  $C_i^k$ 는 i번째 섹터와 겹치는 k번째 CCG의 중심점의 좌표이다.

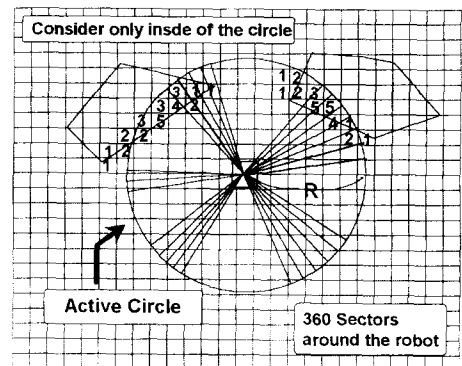


그림 4. Active Circle의 설정과 섹터 분할  
Fig. 4. Active Circle Setting and Sector Division.

위에서 정의된 COD는 섹터별로 장애물이 존재하고 있을 가능성의 크기를 나타내며, DP는 가장 가까운 장애물까지의 거리정보로서 섹터의 위험 정도를 나타낸다. 위에서 구한 COD는 이산적인 특성을 가지므로 다

음의 이동 평균법(moving average)에 의해 평활화(smoothing)한다.

$$COD(i) = \frac{\sum_{k=-n}^n COD(i+k)}{2n+1} \quad (12)$$

여기서 n은 적당한 정수이며 본 논문에서는 10을 사용하였다.

DP는 평활화하지 않는데 이것은 평활화에 의해 장애물이 점유한 영역의 좌, 우끝에서 실제 거리 정보가 변하게 되면 장애물의 기하학적 위치 정보를 이용하는 지역경로 설계시 실제값과 다른 거리정보를 이용하게 되는 문제가 발생할 수 있기 때문이다. 평활화된 COD는 아래의 임계함수(threshold function)를 이용하여 이치화된다.

$$COD(i) = f_{th}[COD(i)] \quad (13)$$

$$\text{where } f_{th}(k) = \begin{cases} 0 & k < \text{threshold} \\ 1 & k \geq \text{threshold} \end{cases}$$

k는 설계 변수이며 본 논문에서는 4를 사용하였다. COD(i)가 1이면 i번째 섹터는 장애물이 점유한 섹터이며 0이면 점유되지 않은 안전한 섹터이다. 이제 COD와 DP를 곱하여 최종적인 DPH를 식 5와 같이 얻는다.

$$DPH(i) = COD(i) * DP(i) \quad \text{where } i = 1 \sim 360 \quad (14)$$

아래의 그림 5에 위에서 설명한 과정의 개념도를 보이고 있다.

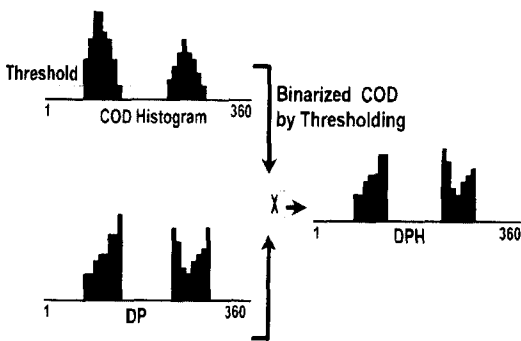


그림 5. COD와 DP를 이용한 DPH의 생성  
Fig. 5. Generation of DPH using COD and DP.

Borenstein 과 Koren의 VFF 히스토그램 방법<sup>15-7</sup>은 장애물이 점유한 그리드 셀의 거리가 인공힘장의 크기를 나타내는 값으로 변환되어 셀의 누적값과 곱해

진 다음 섹터별로 이값을 모두 합산한 값이 히스토그램의 값이 되지만, 본 논문에서 제안한 방법에서는 외부 장애물의 위치에 따른 기하학적 특징을 이용하기 위하여 장애물의 점유 가능성을 나타내는 COD(Cumulative Obstacle Density)와 장애물의 점유 위치를 나타내는 DP(Distance Profile)를 결합하여 DPH(Distance Profile Histogram)를 만든다, 결과적으로 DPH는 장애물이 점유하고 있는 섹터와 점유하고 있는 거리정보를 동시에 표현하는 히스토그램이 된다.

### 3. 지역 경로 계획 알고리즘

#### 1) 중간 경유점(subgoal) 설정

이동로봇의 주행경로상에 복수개의 장애물이 검출되었을 때 이 장애물들의 사이로 주행하며 장애물을 회피하는 것이 필요하다. 본 논문에서 제안한 DPH를 이용한 방법은 장애물의 위치정보를 계산하여 장애물들의 사이로 주행할 수 있는 지역경로를 생성한다. 그림 6에서와 같이 DPH에는 몇 개의 비어있는 영역과 점유된 영역이 존재하게 된다. 회피 경로를 결정하기 위해 먼저 장애물이 존재하지 않는 비어 있는 영역을 찾는다. 이때 영역의 크기가 너무 작은 것은 이동시 충돌의 위험이 있으므로 버린다. 다음 비어있는 영역 중 목표지점에 가까운 방향의 영역을 목표영역으로 선택한다. 선택된 영역은 장애물이 존재하지 않는 섹터들의 집합이다. 이제 장애물의 기하학적 특성을 이용하기 위해 그림 6에서와 같이 목표영역의 양쪽 끝 섹터의 각  $\theta_R, \theta_L$ 을 구한다. 이각은 이동로봇에 부착된 지역 극좌표 시스템(local polar coordinate system)안에서의 각이다. 이각과 장애물이 점유한 영역이 시작되는 섹터의 DPH값을 이용하여 두 장애물의 양 끝점과 두 점의 중점 ( $Mid_x, Mid_y$ )를 다음의 식으로 구한다.

$$Mid_x = \frac{P_R(x) + P_L(x)}{2} \quad (15)$$

$$P_R(x) = \cos(\theta_R) * DPH(\theta_R), \quad P_L(x) = \cos(\theta_L) * DPH(\theta_L)$$

$$Mid_y = \frac{P_R(y) + P_L(y)}{2} \quad (16)$$

$$P_R(y) = \sin(\theta_R) * DPH(\theta_R), \quad P_L(y) = \sin(\theta_L) * DPH(\theta_L)$$

만약 목표영역이 그림 7의 경우에서와 같이 탐색 최대 영역의 크기  $\theta_{max}$ 보다 큰 경우에, 오른쪽 영역일 때는  $DPH(\theta_{max}) = DPH(\theta_R)$ 를, 왼쪽 영역일 때는  $DPH(\theta_{max}) = DPH(\theta_L)$ 를 수행한다. 이것은 장애

물이 존재하지 않는 영역이 매우 클 때 이동로봇이 지나치게 많이 진행 방향을 바꾸는 것을 방지해준다. 이제 (Mid<sub>x</sub>, Mid<sub>y</sub>)를 좌우의 장애물이 점유한 영역의 평균 거리값에 따라서 좀더 안전한 방향으로 다음식에 의해 구해지는 각도만큼 회전이동 시킨다.

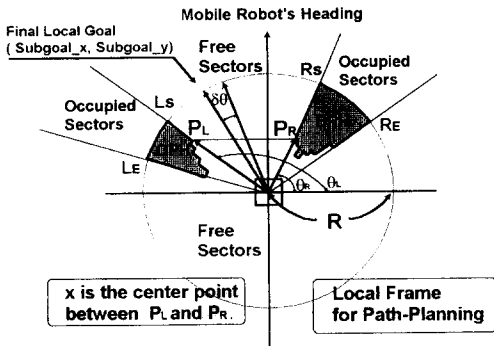


그림 6. DPH를 이용한 기하학적 지역 경로 계획  
Fig. 6. Geometrical Local Path Planning using DPH.

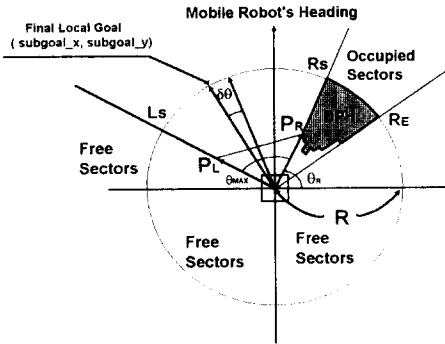


그림 7. 한쪽영역만 장애물이 점유하고 있는 경우  
Fig. 7. Case of One Side Occupation.

$$\delta\theta = K_d \left( \frac{\sum_{i=R_S}^{R_E} DPH(i)}{R_S - R_E} - \frac{\sum_{i=L_S}^{L_E} DPH(i)}{L_E - L_S} \right) \quad (17)$$

R<sub>S</sub>와 R<sub>E</sub>는 각각 목표영역의 오른쪽 점유영역의 시작 섹터와 마지막 섹터를 의미하며, L<sub>S</sub>와 L<sub>E</sub>는 각각 목표영역의 왼쪽에 있는 점유영역의 시작 섹터와 마지막 섹터를 의미한다. 만약 오른쪽 영역에 장애물이 존재하지 않으면 괄호안의 첫번째 항목은 0이된다. 반대로 왼쪽 영역에 장애물이 존재하지 않으면 괄호안의 두 번째 항목이 0이된다. 여기서 K<sub>d</sub>는 설계 변수이며 본 논문에서는 10.0을 사용하였다.

최종적인 점유점 (subgoal<sub>x</sub>, subgoal<sub>y</sub>)는 다음 식에 의

하여 결정된다.

$$\begin{aligned} \text{subgoal}_x &= R \cos(\delta\theta + \theta_{Mid}) \\ \text{subgoal}_y &= R \sin(\delta\theta + \theta_{Mid}) \end{aligned} \quad (18)$$

where  $\theta_{Mid} = \tan^{-1}(\text{Mid}_y / \text{Mid}_x)$

그러나 위에서 설명한 지역 경로 계획 알고리즘을 반드시 실행해야만 하는 것은 아니다. 즉 그림 8에서처럼 목표점으로 곧장 진행하여도 충돌의 위험이 없는 경우에는 지역 점유점을 설정하지 않아도 된다.

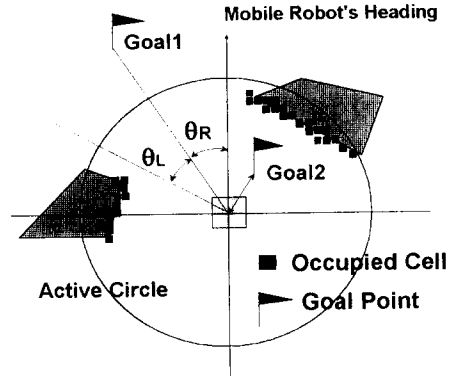


그림 8. 목표점으로 직진할 수 있는 두 가지 경우  
Fig. 8. Two cases when local path planning is not necessary.

따라서 다음과 같은 조건이 만족되면 지역 점유점의 설정없이 곧장 목표점을 향해 진행한다.

- Case 1 :
- if ( $\theta_{L_i} - \theta_{goal} \geq \theta_i$  and  $\theta_{goal} - \theta_{R_i} \geq \theta_i$ )
  - then subgoal<sub>x</sub> = goal<sub>x</sub> and subgoal<sub>y</sub> = goal<sub>y</sub>
- Case 2 :
- if (distance(goal, robot) < DPH(i) for all DPH(i) ≠ 0)
  - then subgoal<sub>x</sub> = goal<sub>x</sub> and subgoal<sub>y</sub> = goal<sub>y</sub>
- (19)

즉 목표점 방향의 좌우로 각각  $\theta_i$  이상의 점유되지 않은 섹터가 존재하는 경우는 중간 점유점을 거치지 않고 직접 목표점으로 주행하게 된다.  $\theta_i$ 는 설계 변수이며 본 논문에서는 60°를 사용한다. 또한 Case 2와 같이 모든 장애물이 목표지점보다 멀리 있는 경우에도 목표점으로 직진하게 된다.

지금까지 설명한 알고리즘은 CCG(Cumulative Certainty Grid)로 파악된 환경모델을 기반으로 가능한 안전한 지점을 경유하여 목표지점까지 주행할 수 있는 지역경로를 생성하게 된다. 장애물 사이의 기하학

적 중점을 중간 경유점으로 선택하는 것은 장애물이 점유하지 않은 공간의 중간 방향을 진행 방향으로 설정하는 것<sup>[5-7]</sup>, 즉 방향적인 중간을 선택하는 것보다 장애물 사이에 있는 공간의 위치적인 중점을 지나게 함으로써 충돌의 위험을 감소시켜 보다 안전한 주행을 보장할 수 있게 한다. 또한 장애물이 점유한 부분의 평균거리에 따라 경유점의 방향을 조정하는 것은 역시 가능한 가까운 장애물로부터 먼 쪽으로 이동로봇이 움직이도록 하여 충돌의 위험을 감소시키기 위한 것이다.

2) 속도 제어

장애물이 밀집한 지역에서 빠른 속도로 주행하는 것은 센서 정보를 충분히 확보하지 못할 뿐만 아니라 충돌의 위험도 크게 증가시킨다. 따라서 이동로봇 주변의 장애물이 위치한 상황에 따라서 주행 속도를 변화시켜야 한다. 이동로봇의 주행속도 변화는 DPH의 특정 구간을 점유하고 있는 장애물의 거리값을 이용하여 다음과 같이 결정된다.

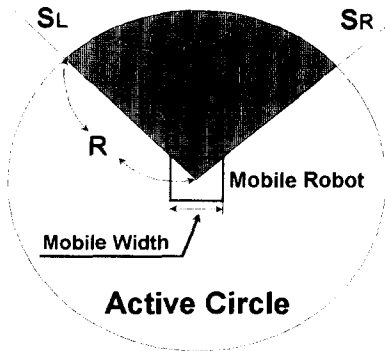


그림 9. 속도제어 영역의 결정  
Fig. 9. Velocity Control Region.

$$V = V_{min} + (V_{max} - V_{min}) * \frac{K_v}{R} \quad \text{여기서, } R = 1.5$$

$$K_v = \frac{\sum_{i=S_L}^{S_R} f(DPH(i))}{S_L - S_R} \quad \text{여기서, } f(j) = \begin{cases} R & j=0 \\ j & j \neq 0 \end{cases} \quad (20)$$

여기서  $S_L$ 과  $S_R$ 은 속도 제어를 위해서 고려되는 충돌의 위험이 있는 구간의 범위이며, 그 크기는 그림 9에서와 같이 이동로봇의 크기를 고려하여 결정된다. 본 논문에서 사용한 이동로봇은 50cm x 50cm의 크기를 가지므로  $S_L - S_R = 90^\circ$ 가 된다. 이 구간내에 장애물이 존재하지 않으면  $K=R$ 이 되고, 이때  $V$ 는  $V_{max}$

가 되어 최고속도가 된다. 반대로, 이 구간의 DPH가 작은 값을 갖는 경우,  $K$ 는  $R$ 보다 작아지게 되고,  $V_{min}$ 에 가까운 값이 되어 속도는 느려진다.

지금까지 설명한 알고리즘에 의해 결정되는 중간 경유점 (subgoal<sub>x</sub>, subgoal<sub>y</sub>)와 속도  $V$ 는 2장에서 설명한 이동제어기의 최고 속도 명령 (reference point 와 maximum velocity)으로 사용된다. 즉, 속도  $V$ 는 실제 구현시 이동제어기에서 발생하는 제어 명령의 직진 평균 속도(linear average speed)의 상한 값으로 작용하게 된다. 단 (subgoal<sub>x</sub>, subgoal<sub>y</sub>)는 이동로봇을 중심으로한 지역 좌표계안에서의 점이므로 실제 명령을 생성하기 전에 전역 좌표계로의 변환이 필요하다.

IV. 실험과 고찰

1. 시스템 구성

본 논문에서 사용한 이동로봇은 2개의 구동륜을 갖는 로봇로서 주제어 컴퓨터인 KVME68040보드를 제외한 모든 하드웨어가 자체 설계 제작되었다. 이동로봇을 제어하는 주행제어 시스템은 그림 10과 같이 구성되어 있으며 제어 프로그램의 구현에는 실시간 O/S인 VxWorks를 사용하고 있다. 제어 시스템의 최하위계층은 두 개의 구동 모터를 직접 제어하는 PID 모터 속도 서보와 초음파 센서를 구동하고 검출된 정보를 처리하는 센서 정보 처리부로 구성되어 있다. PID 속도 서보는 10ms 주기로 동작하며 추측항법 (dead reckoning)에 의해 변화된 이동로봇의 좌표 값을 매 샘플 주기마다 계산한다. 초음파 센서는 3개씩 하나의 그룹을 형성하여 50ms마다 한 그룹의 센서를 구동하고 검출된 거리를 계산한다. 이동로봇을 9개의 센서를 사용하도록 설계되어 있으므로 전체 센서를 구동하고 장애물을 검출하는데 모두 150ms가 소요된다. 중간 계층은 이동 제어기와 충돌 방지로 구성된다. 충돌 방지는 이동로봇의 정면에 10cm이내의 장애물이 검출되었을 때, 이동 제어기에 긴급 정지 명령을 발생시켜 이동로봇을 정지 시키는 기능을 담당한다. 이동 제어기는 상위의 경로 계획 알고리즘에 의해 주어지는 이동 좌표점(subgoal point)을 전달받아 그 점으로의 안정된 이동을 제어하는 속도 명령을 발생시킨다. 이동 제어기의 동작 주기는 속도 서보보다 10배 느린 100ms이며 2.1절에서 설명한 비선형 제어기를 사용하여 구성된다. 최상위 계층의 경로 계획기는 지역

경로 계획 알고리즘으로 구성된다. 이 알고리즘은 200ms 주기로 동작하며, 초음파 센서 정보로 작성된 환경의 누적 확실성 지도(cumulative certainty grid map)로 거리 형태 히스토그램(Distance Profile Histogram)을 만들고, 이를 이용하여 장애물을 회피하며 목적지까지 안전하게 주행할 수 있는 중간 경유점을 설계한다. 이 경유점은 하위의 이동제어기로 전달된다.

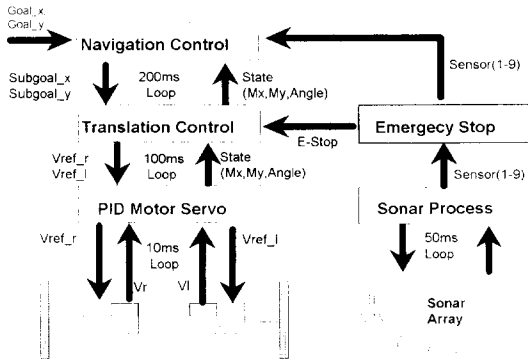


그림 10. 주행 제어 시스템의 구성  
Fig. 10. Configuration of Navigation Control System.

표 1. 실험에 사용된 변수  
Table 1. Experimental Parameters.

Parameter	Value
Maximum Speed(Vmax)	0.7 m/s
Maximum Acceleration	0.25 m/s <sup>2</sup>
Minimum Sampling Time	0.01 sec
Mobile Robot's Size	0.5m x 0.5m x 0.7m ( W x D x H )
Maximum Sonar Range	2.0 m
Number of Sonar Module	9
Angle between Modules	22.5°
World Size	10m x 5m
Grid Size	0.1m x 0.1m
Radius of Active Circle	1.5 m
Threshold of COD	4
DPH Scan Range	-135° ~ +135°
Max Scan Angle	120 °
$\theta_f$	60°
Speed Control Range	-45° ~ +45°
Minimum Speed(Vmin)	0.05m/s
Kd	10.0
$\theta_{max}$	120°

2. 시뮬레이션 및 실험

본 절에서는 시뮬레이션 실험 결과와 실제 실험결과를 보이고 고찰한다. 먼저 시뮬레이션 및 실제 실험에 사용된 설계 변수들과 이동로봇의 물리적인 변수들의 값을 표 1에 보이고 있다.

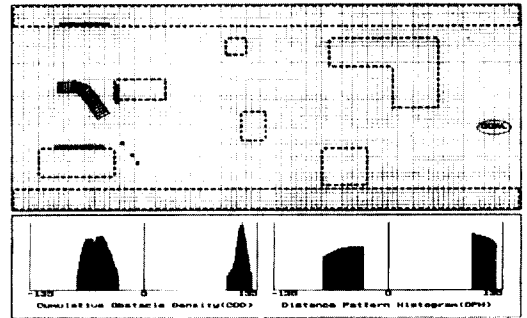
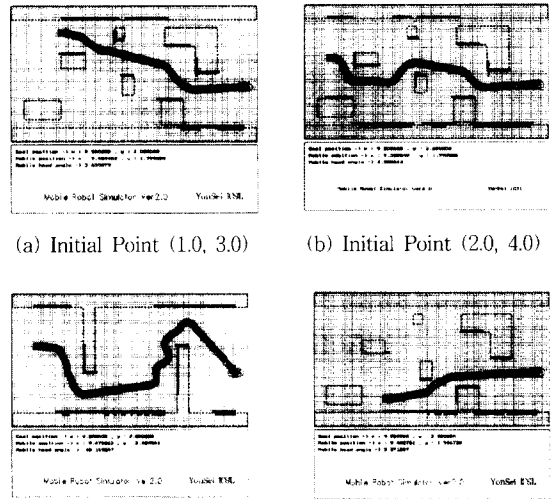


그림 11. 전형적인 COD와 DPH  
Fig. 11. Typical Shape of COD and DPH.



(a) Initial Point (1.0, 3.0) (b) Initial Point (2.0, 4.0)  
(c) Initial Point (3.0, 1.0) (d) Initial Point (1.0, 3.0)  
그림 12. 다양한 초기 위치와 환경에서의 주행 시뮬레이션 결과  
Fig. 12. Simulation Result Starting from Various Initial Point

그림 11는 정면과 오른쪽에 위치한 두 장애물의 사이로 회피 주행을 하고 있는 중간에 획득된 COD 와 DPH, 그리고 DPH를 이용하여 결정된 중간 경유점들을 보이고 있다. 그림의 왼쪽하단의 히스토그램이 COD이고 오른쪽 하단에 있는 히스토그램이 DPH이다. 그리고 두 개의 장애물 사이로 검은색으로 표시된



점이 지역 경로기에 의해 결정된 중간 경유점이 된다. 중간 경유점은 매 200ms 마다 새로이 변화된 장애물의 점유 상태에 따라서 변경되며 장애물을 회피할 수 있는 경로를 생성하게 된다. 센서를 이동로봇의 전면에만 설치하였기 때문에 DPH를 생성하는 영역은 이동로봇의 진행 방향을 0° 로 하여 좌우 135° 로 설정하였다.

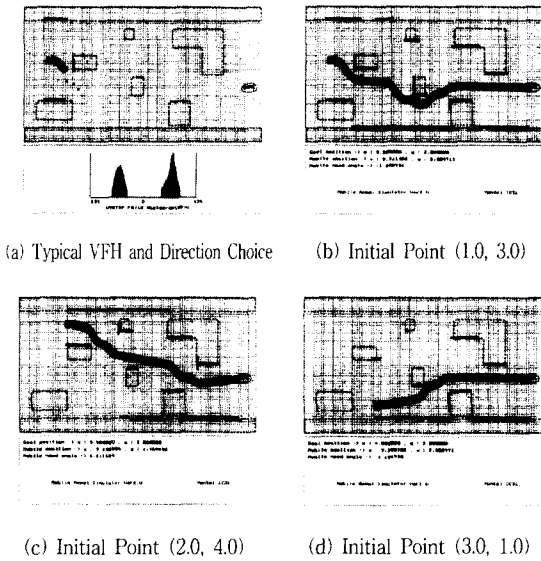


그림 13. Borenstein과 Koren의 VFH방법 [5-7]을 적용한 시뮬레이션 결과  
Fig. 13. Simulation Result of VFH Method.

그림 11은 다양한 초기 위치와 다른 환경에서 주행한 결과를 보이고 있다. 그림에서 원점은 좌측 최하단 꼭지점이며 주행 목표점은 타원으로 표시되어 있다. (a), (b), (c)는 미지의 장애물이 여러개 분포한 복도 환경에서 서로 다른 초기 위치에서 출발한 주행 결과이다. DPH(Distance Profile Histogram)를 이용한 경로 계획 알고리즘은 누적 확실성 지도를 기반으로 장애물들의 기하학적 분포로부터 장애물들의 중간 영역을 통해 주행하도록 설계된 알고리즘이다. 그림 (a), (b), (c)에서 보이는 주행 궤적을 살펴보면, 이동로봇이 설계한 의도대로 장애물들이 점유한 공간의 사이를 통해 장애물을 회피하며 주행하였음을 알 수 있다. 주행의 후반부에서는 (a), (b), (c) 모두 목표점에 근접한 위치에서 좌측에 장애물이 존재하지만 식 20의 직진 주행 조건에 의해 목표점으로 곧장 주행함을 확인할 수 있다. 그림에서 점선으로 표시된 것은 실제 장애물

과 벽이며 검은 색으로 칠해진 격자는 초음파 센서에 의해 검출된 장애물과 벽면들 CCG(Cumulative Certainty Grid)가 0이 아닌 격자를 표시한 것이다. 그림 (d)는 서로 반대 방향이 열려 있는 통로를 통해 주행 목표점까지 주행한 궤적이다. 특히 2번째 벽의 근처에서 회피 주행을 할 때 벽에 매우 근접한 위치에서 주행의 진행 방향이 크게 변화하였음을 알 수 있다, 이것은 오른쪽에 장애물이 매우 가깝게 위치함에 따라 식 188에 의해  $\delta\theta$ 가 큰 값을 갖게 되었기 때문이다.

그림 13는 본 논문에서 제안한 DPH방법과 같은 환경에 VFH(Vector Field Histogram)<sup>[5-7]</sup> 방법을 적용한 결과를 보이고 있다. 그림 13의 (a)와 그림 11를 비교해 보면 두 방법의 경로 결정시의 차이점을 명확히 알 수 있다. 그림 13의 (a)에서는 VFH로부터 점유되지 않은 영역의 중간 방향을 진행 방향으로 택하고 있지만 이 방향은 왼쪽의 가까운 장애물 쪽으로 약간 치우친 경로가 되기 때문에 주행이 진행됨에 따라 충돌의 위험이 증가할 수 있는 위험성을 내포한다. 그림 13의 (b),(c),(d)를 그림 12의 (a),(b),(c)와 비교해 보면 VFH를 이용한 방법도 대체적으로 잘 적용되고 있지만 본 논문에서 제안한 방법이 더욱 안전한 경로를 따라 주행하고 있음을 확인할 수 있다.



그림 14. 이동로봇과 주행 환경 사진  
Fig. 14. Photograph of Experimental Setup with Mobile Robot.

1) 그림 4.4의 결과는 VFH 방법의 방향 결정 알고리즘 [5-7]과 본 논문에서 사용하고 있는 식(3-9)에 의한 중간 경유점 생성 및 2장에서 설명한 비선형 위치 이동 제어를 결합하여 실험한 결과이다. 따라서 방향각만을 제어 기준으로 사용하는 원 저자의 실험결과와 차이가 있을 수 있음을 밝혀둔다

그림 14는 실제 주행 실험을 하기 위해 설치한 장애물들과 이동로봇의 사진을 보이고 있다. 이동로봇은 그림에서와 같이 전면에 9개의 센서를 장착하고 있다. 이러한 복도 환경은 일반적으로 흔히 접할 수 있는 건물의 내부환경이다. 그림 15은 본 논문에서 제안한 DPH를 이용한 지역 경로 계획 알고리즘을 이용하여 실제 주행 실험을 행한 결과를 보이고 있다.

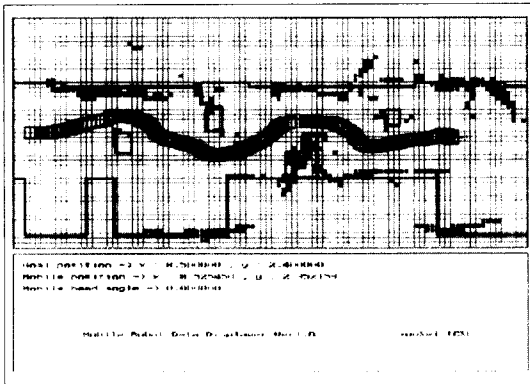


그림 15. 미지의 장애물이 존재하는 복도 환경에서 DPH방법을 사용한 실제 주행  
Fig. 15. Result of Real-Time Navigation using DPH in a cluttered corridor.

이 실험은 VxWorks와 C 언어를 이용하여 구현되었으며 실험에 사용된 이동로봇과 실험환경의 사진을 부록에 보이고 있다. 결과에서 알 수 있듯이 제안된 DPH를 이용한 지역 경로 계획에 의한 장애물 회피 알고리즘이 실제 환경에서도 안전하게 장애물을 회피하며 이동로봇의 주행을 제어함을 알 수 있다. 실제 실험에서 얻어진 주행 궤적에서도 시뮬레이션 실험과 마찬가지로 장애물들이 점유하는 기하학적인 공간의 중간을 경유하며 안전하게 목표지점까지 주행을 성공적으로 제어하고 있다. 실제 실험 환경은 시뮬레이션 환경과 달리 센서 정보에 랜덤한 오차가 발생하고 다중 반사 현상 등이 빈번히 발생한다. 즉 초음파 센서에 의해 관측되는 환경에 많은 오차가 포함된다. 이와 같은 특성은 위의 실험 결과에서 쉽게 관찰할 수 있다. 또한 주행이 진행됨에 따라 새로이 획득되는 환경정보와 이동된 이동로봇의 위치변화에 따라 새로운 경로를 생성해 주어야 한다. 본 논문에서 사용한 CCG(Cumulative Certainty Grid)방법은 환경의 정확한 묘사력은 다소 떨어지지만 장애물이 점유하고 있을 조건 확률 정보를 실시간으로 생성하는 장점이 있음을 확인

할 수 있었고, 섹터(sector)별로 평균된 COD(Cumulative Obstacle Density)에 적절한 임계치를 적용함으로써 오차에 의한 영향을 최소화시키면서 장애물의 회피에 충분한 정보를 제공하는 DPH(Distance Profile Histogram)을 얻을 수 있음을 확인하였다.

## V. 결 론

초음파 센서는 이동로봇이 환경을 인식하는데 가장 널리 사용되고 있는 센서이다. 이 센서는 정보처리가 빠르고 쉬운 반면 거울 현상에 의해 각도 분해능이 떨어지는 단점이 있다. 본 논문에서는 이러한 초음파 센서의 단점을 보완하는 격자형 환경 모델과 CCG(Cumulative Certainty Grid)를 이용하여 COD(Cumulative Obstacle Density)와 DPH(Distance Profile Histogram)를 생성하는 방법과 DPH를 이용해 장애물들의 기하학적 배치에 따라 지역 경로를 계획하는 알고리즘을 제안하였다. 특히 실제 이동로봇의 제어 시스템을 구현하기 위해, 제안된 알고리즘을 최상위에 위치시키고 중간 계층에 안정성이 보장된 이동제어기, 그리고 최하위에 모터 서보 및 초음파 센서 처리부를 두어 계층적인 제어 구조를 형성하고, 역할에 따라 서로 다른 주기로 동작하는 제어 시스템을 구현하였다. 컴퓨터 모의 실험을 통해 제안된 DPH를 이용한 장애물 회피 주행 알고리즘이 장애물들이 밀집해 분포하는 환경에서는 장애물이 점유하지 않은 공간의 가운데를 경유하여 목표지점까지 주행함으로써 알고리즘의 설계 의도가 장애물의 회피 주행에 잘 동작함을 보여 주었고, 진행 방향각만을 제어명령으로 사용하는 V로(Vector Field Method)보다 안전한 경로를 생성하며 목표지점까지 주행함을 확인하였다. 또한 미지의 장애물이 다수 분포하는 실제 복도 환경에서의 실험을 통하여, 제안된 알고리즘이 랜덤하게 발생하는 센서의 오차를 잘 극복하고 안전한 경로를 통하여 이동로봇의 주행을 제어함을 보임으로서 제안된 알고리즘의 성능과 타당성, 그리고 실시간 제어성을 입증하였다. 추후 과제로 막힘상황(dead end)과 같이 충분한 센서 정보를 확보하기 전에는 파악하기가 어려운 환경에서 연속적인 센서 정보의 상관 관계를 이용하여 보다 빨리 장애물의 기하학적 배치 상황을 파악할 수 있는 알고리즘과 이동하는 장애물의 검출과 회피방법에 대한 연구를 진행하고 있다.

## 참 고 문 헌

- [1] H. T. Kim, Mignon Park, "Tangential Segment Based Half Step Path Planning for Sonar Based Navigation", The 1st Asian Control Conference, Japan, vol. 2, pp. 945-949, 1994.
- [2] J.J.Han, H.T.Kim, Mignon Park, "Mobile Robot Navigation using Ellipsodal World Modeling Method", Proc. of JTC-CSCC '95, Japan, pp. 153-156, 1995.
- [3] H.P.Moravec and A.Elfes, "High Resolution Maps from Wide Angle Sonar", IEEE International Conference on Robotics and Automation, St.Louis, pp. 116-121, 1985.
- [4] Roman Kuc, "Physically Based Simulation Model for Acoustic Sensor Robot Navigation", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 6, pp. 766-778, 1987.
- [5] Koren, Y., Borenstein, J., "Realtime Obstacle avoidance for Fast Mobile Robot in cluttered Environment", IEEE International Conference on Robotics and Automation, vol 1, pp 572-577, 1990.
- [6] Koren, Y., Borenstein, J., "Real-Time Obstacle Avoidance for Fast Mobile Robots", IEEE Transactions of Systems, Man, and Cybernetics, vol. 19, no. 5 pp 1179-1186, 1989.
- [7] Koren, Y., Borenstein, J., "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", IEEE Trans. Robotics and Automation, vol. 7, no. 3, pp 278-288, 1991.
- [8] D.W.Cho, "Certainty Grid Representation for Robot Navigation by a Bayesian Method", Robotica, vol 8, pp. 159-165, 1990.
- [9] Jong Hwan Lim and Dong Woo Cho, "Physically Based Sensor Modeling for a Sonar Map in a Specular Environment", Proceedings of the 1992 IEEE International Conference on Robotics and Automation, pp. 1714-1719, 1992.
- [10] H. Y. Chung, J. G. Lee, "Path Planning for a Mobile Robot with Grid Type World Model", IEEE/RSJ International Conference of IROS, USA, vol 1, pp. 439-443, 1992.
- [11] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, "A Stable Tracking Control Method for a Non-Holonomic Mobile Robot," IEEE/RSJ Int. Workshop on Intelligent Robot and Systems IROS '91, pp. 1236-1241, 1991.
- [12] O. J. Sordalen, C. Canudas de Wit, "Exponential Control Law for a Mobile Robot: Extension to Path Following," IEEE Trans. Robotics and Automation, vol. 9, no. 6, pp. 837-842, 1993.
- [13] M.Aicardi, G.Casalino, A.Bicchi, A.Balestrino, "Closed Loop Steering of Unicycle-like Vehicles via Lyapunov Techniques", IEEE Robotic & Automation Magazine, vol 2, no. 1, March, 1995.
- [14] Koren, Y. Borenstein, J., "Teleautonomous Guidance for Mobile Robots", IEEE Trans. on Systems, Man and Cybernetic, vol. 20, no. 6, 1990.
- [15] Wilco Oelen, Harry Berghuis, Henk Nijmeijer, Carlos Canudas de Wit, "Hybrid Stabilizing Control on a Real Mobile Robot", IEEE Robotic & Automation Magazine, vol 2, no. 2, June, 1995.
- [16] Hyuntae Kim, Mooseung Shin, Mignon Park, "A Tracking Control of Mobile Robot using Variable Reference Step Algorithm", 제 13 회 일본 로보트학회 학술대회 논문집, no 1, pp 179-180, November, 1995
- [17] R. W. Brockett, R. S. Millman, and H. J. Sussman, "Asymptotic stability and feedback stabilization" in Differential Geometric Control Theory, MA: Birkhauser, 1983, pp. 181-208.

## — 저 자 소 개 —



金賢泰(正會員)

1968년 10월 3일생. 1991년 2월 연세대학교 전자공학과 졸업(학사). 1993년 2월 연세대학교 대학원 전자공학과 졸업(석사). 1993년 3월 ~ 현재 연세대학교 대학원 전자공학과 박사과정. 주관심분야는 자율이동로봇, 로

봇제어, 로봇 비전, 지능제어, 제어이론, 퍼지이론 등

曹泳完(正會員) 第33卷 B編 第11號 參照

현재 연세대학교 대학원 전자공학과 박사과정



盧興植(正會員)

1968년 5월 15일생. 1991년 2월 연세대학교 전자공학과 졸업(학사). 1993년 2월 연세대학교 대학원 전자공학과 졸업(석사). 1993년 3월 ~ 현재 연세대학교 대학원 전자공학과 박사과정. 주관심분야는 제어이론, 로봇공

학, 퍼지이론 등임

朴玟用(正會員) 第33卷 B編 第11號 參照

현재 연세대학교 대학원 전자공학과 교수