

Lane-Curvature Method: A New Method for Local Obstacle Avoidance

高樂裕* 李相基**
(Nak Yong Ko · Sang Kee Lee)

Abstract - The Lane-Curvature Method(LCM) presented in this paper is a new local obstacle avoidance method for indoor mobile robots. The method combines Curvature-Velocity Method(CVM) with a new directional method called the Lane Method. The Lane Method divides the environment into lanes taking the information on obstacles and desired heading of the robot into account; then it chooses the best lane to follow to optimize travel along a desired heading. A local heading is then calculated for entering and following the best lane, and CVM uses this heading to determine the optimal translational and rotational velocities, considering the heading direction, physical limitations, and environmental constraints. By combining both the directional and velocity space methods, LCM yields safe collision-free motion as well as smooth motion taking the dynamics of the robot into account. Experiments using the mobile robot Xavier, show the efficiency of the proposed method.

Key Words : Local obstacle avoidance, Lane method, Curvature-velocity method, Collision-free local heading, Heading command, Translational velocity, Rotational velocity

1. 서 론

본 논문에서는 실내에서 이동 로봇이 주변의 검출된 장애물을 피하며 주어진 시작 위치에서 목표 위치까지 이동하기 위한 지역 장애물 회피(Local obstacle avoidance) 방법을 제안한다. 로봇이 장애물을 피하며 가능한 한 빠른 시간 내에 목표 위치까지 이동하기 위해서는 장애물 회피 알고리즘이 실시간 실행이 가능하도록 효율적이어야 하고, 로봇의 동역학적인 제한 조건들을 만족시킬 수 있어야 한다.

로봇을 제어하기 위한 명령의 형태를 고려하여 지금까지의 연구 결과들을 크게 두 가지로 구분해보면, 로봇의 무충돌 이동 방향을 결정하여 로봇에 명령을 내리는 "방향 명령 방식"과 장애물 회피에 적합한 병진 속도와 회전 속도를 구하는 "속도 공간 명령 방식"으로 크게 구분할 수 있다. 방향 명령 방식은 효율적으로 충돌 회피를 위한 이동 방향을 구해내지만 로봇의 동역학적인 제한 조건을 고려하지 않은 것들이 대부분이었고, 동역학적인 제한 조건을 고려하기에는 부적합한 방법들이었다. 따라서 실제 주행시 로봇이 좁은 통로를 빠져나가지 못한다든지, 이동 궤적에 진동이 발생하기도 한다. 이러한 방식들의 예로는 V-그래프 탐색 방법[1], 인공 전위계 방법[2,3,4], 벡터장 히스토그램(Vector Field

Histogram) 방법[4]등이 있다. 벡터장 히스토그램 방법[4]은 인공 전위계 방법을 발전시킨 것으로서 좁은 통로에서도 비교적 부드럽고 성공적으로 로봇을 이동하게 하지만 로봇의 동역학적 제한 조건을 고려하지 못하여 많은 장애물이 흩어져 있는 경우 로봇의 이동에 문제를 발생시킨다.

속도 공간 명령 방식은 장애물에 관한 정보와 로봇의 동역학적 제한 조건을 고려하여 로봇의 병진 속도 명령과 회전 속도 명령을 구해낸다. 이 방식에서는 로봇의 이동 궤적이 원호들로 이루어져있다고 가정하였다[5,6,7,8,9,10]. 이들 중 CVM(Curvature-Velocity Method)[5]은 병진 속도-회전 속도 공간에서 제한 조건들을 만족시키는 동시에 설정한 목표 함수를 최대로 만드는 점을 구하였다. 여기에서는 로봇의 병진 속도와 가속도, 그리고 회전 속도와 가속도를 최대값 이하로 유지하여, 로봇의 동역학적인 제한 조건을 고려하였다. 이 방법은 실내 환경에서 부드럽고 빠르게 그리고 안정되게 로봇이 이동할 수 있게 한다. 그러나 때때로 좀더 효율적인 충돌 회피 경로가 있는 경우에도 이를 찾지 못하는 단점이 있다. 예를 들어 로봇의 진행 방향과 직각 방향으로 충돌 회피 통로가 있고 그 통로가 좁은 경우에 이를 찾지 못하기도 한다. 또한 로봇의 진행 방향에 장애물이 있고 그 장애물의 주변에 충분한 충돌 회피 경로가 있는 경우에도 이를 피하기 위한 회피 동작을 장애물에 가까이 다가가서야 시작하여 급격한 우회 동작을 하게 한다. 이러한 문제점은 로봇의 이동 경로를 원호로 가정하여, 이 원호 경로들 중 장애물까지의 원호 거리가 긴 경로를 선택하여 로봇이 이를 따라가도록 제어하기 때문에 발생한다.

원호 궤적을 이용한 방법들 중 조종각 필드 방법

*正 會 員 : 朝鮮大 電氣制御計測工學部 副教授 · 工博

**正 會 員 : 朝鮮大 航空造船工學部 副教授 · 工博

接受日字 : 1998年 11月 16日

最終完了 : 1999年 1月 28日

(Steering Angle Field Method)[7]에서는 장애물과 접하는 원호 궤적을 로봇 조종각에 대한 제한 조건으로 하여 로봇의 조종각 명령을 구한다. 이 방법은 곡률-속도 방법과 달리 병진 속도와 회전 속도를 동시에 구하지 못하는 단점이 있다. Rhino 로봇의 주행에 관한 연구[6]에서는 로봇의 동력학적 제한 조건을 고려하여 원호 궤적들을 이산(discrete) 집합으로 나누고, 목표 방향에 가장 근접한 원호 궤적을 선택한다. 이 방법은 곡률과 속도를 2 단계 과정을 통하여 구한다. 동적 윈도우(dynamic window) 방법[8]에서는 이러한 2 단계 방법을 통합하여 한번의 과정에서 곡률과 속도를 구한다. 이러한 방법들은 대부분 실내 주행 로봇의 이동에 적용하기 위한 방법들인데, 비슷한 방식으로 실외 주행 로봇의 장애물 회피에 적용하기 위한 방법[9]도 개발되었다. 이상의 방법들은 원호들을 이산 집합으로 구분함으로써 실제의 최적 원호 궤적을 찾지 못하는 경우도 발생하는 단점이 있다.

본 연구에서는 방향 명령 방식과 속도 공간 명령 방식의 단점들을 보완하고 장점들을 살리기 위해서 두 방식을 결합한 차선-곡률 방법(Lane-Curvature Method: LCM)이라는 새로운 방법을 제안한다. 차선-곡률-방법은 두 단계의 과정을 거쳐서 로봇의 병진 속도와 회전 속도 명령을 구해낸다. 첫 단계에서는 차선 방법이라는 새로운 방향 명령 방식을 통하여 로봇의 동작 영역을 차선들로 구분한 후, 장애물 회피, 운행의 효율성, 그리고 목표 방향으로의 지향성(Goal directedness)을 고려하여 최적의 차선을 선택한다. 그리고 선택된 차선으로 로봇을 진입시키고 차선을 따라가게 하기 위한 로봇의 무충돌 이동 방향(Collision-free local heading)을 결정한다. 따라서 속도 공간 명령 방식에서의 문제점인 효율적인 충돌 회피 경로를 지나쳐버리는 현상을 제거할 수 있다.

차선 방법은 방향 명령 방식이므로 충돌을 피하기 위한 효율적인 이동 방향을 결정할 수 있지만, 로봇의 동력학적 제한 조건을 고려하지는 못한다. 따라서 차선-곡률 방법에서는 차선 방법에 의해서 구해진 무충돌 이동 방향을 곡률-속도 방법에 방향 명령(Heading command)으로 입력하여, 로봇의 동력학적인 제한 조건을 고려한 병진 속도 명령과 회전 속도 명령을 구한다.

벡터장 히스토그램 방법[4]은 장애물이 없는 무충돌 공간들 중, 로봇에서 계산한 무충돌 각도 영역이 넓은 공간 쪽으로 이동 방향을 선택한다. 그런데 장애물들이 로봇에 가까이 있는 경우에는, 실제 무충돌 공간이 좁아도 무충돌 각도 영역은 넓어질 수 있다. 이 경우 벡터장 히스토그램 방법은 실제 무충돌 공간이 더 넓은 영역이 있더라도 찾지 못하고 좁은 공간 쪽으로 이동 방향을 설정하게된다. 반면, 본 논문에서 제안하는 차선 방법은 무충돌 공간의 폭(width)과 장애물까지의 거리를 기준으로 하여 무충돌 이동 방향을 결정하므로, 벡터장 히스토그램 방법에 비해서 더 안전한 공간으로 로봇을 유도한다.

본 논문은 2 장에서 곡률-속도 방법을 소개하고, 3 장에서는 차선 방법을 설명한다. 4 장에서는 본 연구에서 제안한 방법을 이동 로봇의 주행에 이용하여 실험한 결과를 비교하고 분석한다. 5 장에서는 본 연구의 결과를 검토하여 결론을 내린다.

2. 곡률-속도 방법

곡률-속도 방법[5]은 이동 로봇의 지역 장애물 회피 문제를 속도 공간에서의 제한 최적화(Constrained optimization) 문제로 변환한다. 이 방법은 속도 공간에서 로봇의 병진 운동과 회전 운동상의 한계 내에서 병진 속도 명령과 회전 속도 명령을 구하므로, 로봇의 동력학적인 제한 조건을 고려할 수 있다. 곡률-속도 방법은 다음의 목적 함수 $f(tv, rv)$ 를 최대화 하는 병진 속도 tv 와 회전 속도 rv 를 구한다.

$$\begin{aligned} f(tv, rv) &= \alpha_1 \cdot dist(tv, rv) + \alpha_2 \cdot head(rv) + \alpha_3 \cdot speed(tv) \\ dist(tv, rv) &= d(tv, rv, OBS)/L \\ head(rv) &= 1 - |\theta_c - rv \cdot T_c| / \pi \\ speed(tv) &= tv / tv_{max} \end{aligned} \quad (1)$$

T_c : 로봇에 이동 명령을 입력시키는 시간 주기

여기에서 $d(tv, rv, OBS)$ 는 로봇이 장애물 집합 OBS 와 충돌하지 않으며 곡률이 $c = rv/tv$ 인 원호 경로상에서 이동할 수 있는 최대 원호 길이이다. $d(tv, rv, OBS)$ 는 일정한 제한 최대 거리 L (본 연구에서는 3 m)에 의해서 $dist(tv, rv)$ 로 정규화 된다. 이것은 무충돌 원호 길이가 L 이상 되는 경우에는 장애물 회피에 문제가 없으므로, 거리의 차이에 따라 목적 함수 값이 달라지는 것을 막기 위해서이다. L 을 너무 크게 하면 실제 장애물 회피에 문제가 없음에도 목적 함수가 작은 값을 갖게되고, L 을 너무 작게 하면 무충돌 원호 길이의 차이를 장애물 회피에 충분히 고려하지 못하게 된다. $head(rv)$ 는 로봇이 T_c 시간 동안 회전 속도 rv 로 회전했을 때의 방향각인 $rv \cdot T_c$ 와 로봇의 이동 목표각 θ_c (θ_c 는 로봇에 부착된 로봇 좌표계에서의 값)와의 차이를 최대값 π 에 의해서 정규화한 값으로서, 로봇의 목표 방향으로부터의 방향 편차를 나타낸다. 그리고 $speed(tv)$ 는 로봇의 병진 속도를 최대값 tv_{max} 에 의해서 정규화한 값이다. $f(tv, rv)$ 를 최대화함으로써 로봇은 가능한 가장 빠른 병진 속도로, 최단 시간 내에 목표각 방향으로 회전하는 동시에 장애물과의 거리를 멀리 유지하는 원호 경로를 따라 이동하게된다. $\alpha_1, \alpha_2, \alpha_3$ 값들은 각각 무충돌 원호 거리, 목표 방향으로의 방향성, 로봇의 속도를 어느 정도의 우선 순위에 의해서 고려할 것인지를 결정하는 가중치이다.

곡률-속도 방법에서는 로봇의 동력학적인 면을 고려하기 위해서, 로봇의 병진 운동과 회전 운동상의 물리적인 한계를 다음 식 (2)와 같이 표현하여 제한 조건으로 한다.

$$\begin{aligned} 0 &\leq tv \leq tv_{max}, & -rv_{max} &\leq rv \leq rv_{max} \\ rv &\geq rv_{cur} - (ra_{max} \cdot T_{accel}) \\ rv &\leq rv_{cur} + (ra_{max} \cdot T_{accel}) \\ tv &\leq tv_{cur} + (ta_{max} \cdot T_{accel}) \end{aligned} \quad (2)$$

여기에서 tv_{cur} 와 rv_{cur} 은 각각 로봇의 현재 병진 속도와 현재 회전 속도를 나타내며, T_{accel} 은 로봇의 가속 시간

주기로서 식 (1)의 T_c 와 같은 값으로 설정하였다. 이 제한 조건은 로봇의 병진 속도, 회전 속도, 병진 가속도, 그리고 회전 가속도를 각각 최대값 tv_{max} , rv_{max} , ta_{max} , 그리고 ra_{max} 이하로 제한한다. 제한 조건들 중 $0 \leq tv$ 는 로봇이 후진하지 않도록 한다는 것을 의미한다. 로봇의 병진 속도를 항상 T_{accel} 시간 내에 0 으로 만들어, 문제가 발생하는 경우 로봇을 되도록 빠르게 정지시키기 위해서 제한 조건에 $tv \geq tv_{cur} - (ta_{max} \cdot T_{accel})$ 를 포함하지 않는다. 전체적으로 곡률-속도 방법은 식 (2)의 동역학적 제한 조건을 만족시키며 식 (1)의 $f(tv, rv)$ 를 최대로 하는 병진 속도 명령 tv 와 회전 속도 명령 rv 를 구함으로서, 로봇이 물리적인 구동 능력의 한계 내에서 장애물을 피하면서 목표 방향으로 빠르고 부드럽게 이동하게 한다.

3. 차선 방법에 의한 무충돌 이동 방향 결정

차선-곡률 방법에서는 먼저 차선 방법에 의해서 무충돌 이동 방향을 구하고, 구해진 무충돌 이동 방향(무충돌 이동 방향은 곡률-속도 방법에 입력되어 목표 방향으로 사용되므로, 이하 "방향 명령"이라고 함)을 목표 방향 θ_c 로 설정하여 곡률-속도 방법에서 최적의 병진 속도와 회전 속도를 구한다. 작업 공간을 차선으로 구분하는 방법은 다음과 같다.

3.1 차선 구분

로봇의 작업 공간은 장애물의 위치 및 크기, 로봇의 이동 목표 방향을 고려하여 몇 개의 차선으로 구분된다. 차선의 방향은 로봇의 이동 목표 방향(로봇의 현재 위치에서 목표점까지의 방향)으로 한다. 각각의 차선들은 로봇이 차선 방향으로 진행했을 때 장애물과 만나기까지의 무충돌 직선 거리가 비슷한 공간들을 서로 합쳐서 구성한다.

본 연구에서는 차선의 결정을 용이하게 하고 곡률-속도 방법에서의 장애물 처리 방법과의 일치를 위해서 장애물들의 형상을 원으로 근사시킨다. 그리고 각 장애물의 반지름을 로봇 크기의 반지름(30cm) 만큼 증가시켜서 장애물을 형상 공간(Configuration space) 상의 장애물로 변환하여, 로봇을 점으로 취급한다. 본 연구에서 사용한 이동 로봇은 24개의 초음파 센서들을 원형으로 배열하여 사용하므로, 인접한 초음파 센서들 사이의 초음파 지향각은 15° 의 차이를 가진다. 따라서 어떤 초음파 센서에 의해서 로봇에 대한 장애물의 방향과 거리가 검출되면, 장애물을 초음파 센서의 검출 방향을 중심으로 15° 영역의 크기를 가지는 원으로 근사시킨다. 만일 n 개의 초음파 센서에서 장애물이 검출되면 원으로 근사된 n 개의 장애물 정보를 가지고 차선 방법을 적용한다. 그리고 벽은 여러 개의 연속된 원형 장애물들의 배열로 나타난다.

각 차선을 기술하기 위한 변수들은 다음과 같다.

- $w(k)$: k 번째 차선의 폭(Width)
- $d(k)$: k 번째 차선의 무충돌 거리
- $va(k)$: 로봇으로부터 k 번째 차선으로의 무충돌 방향각

(Viewing angle)

이들 변수들과 차선들의 결정 방법을 그림 1에서 설명한다. 여기에서 차선의 숫자 N_L 은 $N_L=6$ 이다. 로봇의 작업 공간 중에서 센서의 장애물 검출 영역(Sonar detecting range)이내의 공간에서 차선을 결정한다. 본 연구에서는 장애물 검출 영역을 로봇으로부터 4 m 이내의 거리로 한다. 그리고 로봇은 후진 이동을 하지 않으며 급격한 방향 전환을 하지 않으므로, 차선을 결정할 때 목표 방향 gd 에서 일정한 각도 이상 벗어나 있는 장애물들은 고려하지 않는다. 이 각을 경계각(Boundary angle)이라 부르고, ba 로 표시한다. ba 는 목표 방향 gd 의 왼쪽과 오른쪽에 대하여 각각 결정하며 다음과 같이 설정한다.

$$|ba| = \begin{cases} 90^\circ & \text{starting line에 장애물이 없을 경우} \\ 55^\circ & \text{starting line에 장애물이 있을 경우} \end{cases} \quad (3)$$

$|ba|$ 를 너무 크게 하면 불필요하게 넓은 영역의 장애물들을 고려하여 차선을 결정하므로 불필요한 계산 시간이 소요되고, $|ba|$ 가 너무 작으면 로봇의 방향 전환 영역을 너무 제한하여 충돌 회피 경로를 찾지 못하게 된다.

k 번째 차선의 무충돌 거리 $d(k)$ 는 로봇이 출발선(Starting line)으로부터 차선을 따라 이동할 때 장애물과 충돌하기까지의 최대 직선 거리이다. k 번째 차선으로의 무충돌 방향각 $va(k)$ 는 로봇이 현재 위치에서 k 번째 차선으로 충돌 없이 이동하기 위한 최소 조향각으로서 목표 방향 gd 로부터 측정한다. 방향각 $va(k)$ 를 결정할 때, k 번째 차선 전체가 출발선에서부터 $d(k)$ 의 거리 이후에는 막혀있는 것으로 가정한다. 그리고 $k < l < n_r$ (n_r 은 로봇이 현재 위치해있는 차선의 번호) 이고 $va(k) \leq va(l)$ 인 경우, $va(k) \leftarrow va(l)$ 로 한다. 즉, 차선 k 의 장애물이 차선 l 의 장애물에 가려지는 경우 $va(k)$ 는 $va(l)$ 로 설정한다.

차선을 지나치게 세분화하면, 어떤 차선은 주변 차선과 합쳐져서 안전한 통로를 만들 수는 있지만 자신의 폭은 좁은 경우가 발생할 수 있다. 그런데 충돌 회피를 위해 차선을 선택할 때, 좁은 차선은 선택하지 않으므로 이러한 차선은 선택되지 않게 된다. 또한 지나치게 차선을 세분화하여 차선의 갯수가 많아지면 처리하는데 많은 시간이 소요된다. 따라서 불필요하게 차선을 세분화하는 것 보다, 차선의 폭이 매우 좁은 경우나 두개의 인접한 차선의 무충돌 거리가 비슷한 경우, 두개의 차선을 통합하여 하나의 차선으로 만드는 것이 바람직하다. 다음 (규칙 1)에 의하여 폭이 매우 좁은 차선을 주위의 다른 차선과 통합한다.

(규칙 1) $1 \leq k \leq N_L - 2$ 인 어떤 k 에 대하여, $w(k) \leq w_{min}$ 이고 $d(k) > \text{Min}\{d(k-1), d(k+1)\}$ 이면, k 번째 차선을 $k-1$ 번째 차선과 $k+1$ 번째 차선 중 무충돌 거리가 짧은 차선으로 통합한다.

w_{min} 은 본 연구에서 2.0 cm로 한다. w_{min} 을 너무 크게 하면 충돌 회피가 가능한 좁은 통로를 찾지 못하게 되고, w_{min} 을 너무 작게 하면 좁은 차선이 많이 형성되어 위에 제시한 문제점이 발생한다.

두 개의 인접한 차선이 무충돌 거리가 비슷한 경우에는 다음의 (규칙 2)에 의하여 차선을 통합한다.

(규칙 2) $0 \leq k \leq N_L - 2$ 인 k 에 대하여, $|d(k) - d(k+1)| \leq \Delta d_{\min}$ 이면, 무충돌 거리가 긴 차선을 무충돌 거리가 짧은 차선으로 통합한다.

본 연구에서 Δd_{\min} 은 2.5 cm로 하였다. Δd_{\min} 을 크게 하면 무충돌 거리가 긴 차선이 무충돌 거리가 짧은 차선과 통합되므로 좀더 안전한 무충돌 공간을 찾지 못하게 되고, Δd_{\min} 을 작게 하면 차선의 통합 효과가 저하된다.

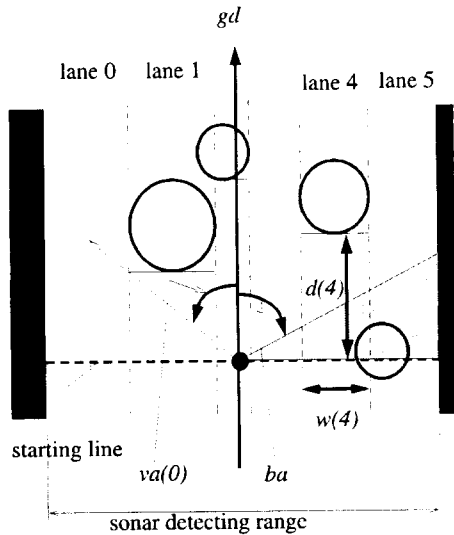


그림 1. 차선 및 차선을 기술하는 변수들
Figure 1. Lanes and their describing parameters

3.2 최적의 차선 선택

차선이 구분되면 이들 중 로봇이 장애물을 피하면서 효율적으로 목표 방향으로 이동하기에 알맞은 차선을 선택한다. 충돌 회피를 위해서는 무충돌 거리가 길고, 폭이 넓은 차선을 선택하여야 한다. 그리고 효율적인 로봇의 방향 조정을 위해서는 방향 명령의 변화량이 적을수록 좋다. 잡음의 영향을 많이 받는 초음파 센서에 의해서 실제 존재하지 않는 장애물이 검출되어 갑자기 방향 명령이 크게 변화하는 것을 막기 위해서도 방향 명령의 변화량을 되도록 작게 하는 것이 바람직하다. 또한 로봇이 빠르게 이동하기 위해서는 로봇의 방향 명령이 로봇의 현재 방향과 가까운 것이 바람직하다. 이러한 사실들로부터 차선 선택 함수(Lane selection function) $f_s(k)$ 를 다음과 같이 설정한다.

$$f_s(k) = \beta_1 \cdot \bar{d}(k) + \beta_2 \cdot \bar{w}(k) - \beta_3 \cdot \overline{ad}_{va,c}(k) - \beta_4 \cdot \overline{ad}_{va,o}(k) \quad (4)$$

$$\begin{aligned} \bar{d}(k) &= \text{Min}\{d(k), D_{\lim}\} / D_{\lim} \\ \bar{w}(k) &= \text{Min}\{w(k), W_{\lim}\} / W_{\lim} \\ \overline{ad}_{va,c}(k) &= \text{Min}\{|va(k) - c_p|, C_{\lim}\} / C_{\lim} \\ \overline{ad}_{va,o}(k) &= \text{Min}\{|va(k) - o_r|, O_{\lim}\} / O_{\lim} \end{aligned} \quad (5)$$

여기에서, c_p : 이전 방향 명령, o_r : 로봇의 현재 방향.

위의 식에서는 방향각 $va(k)$ 를 로봇을 k 번째 차선으로 이동시키기 위한 방향으로 간주하여 사용하였다. 이것은 방향각 $va(k)$ 가 로봇을 현재 위치에서 k 번째 차선으로 충돌 없이 이동하게 하는 최소 조향각이기 때문이다. 식 (4)에서 각각의 항들은 식 (5)에서와 같이 각각 최대값 $D_{\lim}, W_{\lim}, C_{\lim}, O_{\lim}$ 에 의해서 한정되어지고, 정규화(Normalization)되어졌다. $\overline{ad}_{va,c}(k)$ 는 방향 명령의 변화량을 작게 하기 위한 항이고, $\overline{ad}_{va,o}(k)$ 는 방향 명령을 로봇의 현재 방향에 가깝게 하기 위한 항이다. β 값들은 각각의 항들에 대한 가중치(Weighting)들이다. 만일 β_1 과 β_2 를 다른 값들보다 상대적으로 크게 하면 로봇 이동 방향의 변화가 커지며 좀더 안전한 공간을 찾아갈 수 있게되지만, 이동 방향의 변화가 심하여 이동 궤적에 진동이 발생하고 센서의 잡음에 지나치게 민감하게 반응하여 충돌 회피 공간을 찾지 못하게 된다. 반면, β_3 와 β_4 가 다른 값에 비하여 지나치게 크면 로봇 이동 궤적은 부드러워지지만, 무충돌 공간을 찾지 못하게 된다. 본 연구에서는 여러 번의 실험 결과들을 종합하여 이 값들의 비율 $\beta_1:\beta_2:\beta_3:\beta_4=6:1:6:1$ 로 조정하였다. 식 (4)를 최대로 하는 k 를 구하면, 로봇이 충돌을 피하면서 빠르고 효율적으로 목표 방향으로 이동하기 위한 차선이 구해진다.

3.3 방향 명령[무충돌 이동 방향]의 결정 방법

식 (4)를 최대로 하는 차선을 n_s 번째 차선이라 하고, 로봇이 속해있는 차선의 번호를 n_r 이라 하자. 만일 로봇이 이미 선택된 차선 내에 있는 경우, 즉 $n_r = n_s$ 이면 로봇은 목표 방향으로 계속 나아가면 되므로 로봇의 방향 명령 hc 는 로봇의 목표 방향 gd 과 일치하게 된다.

만일 로봇이 선택된 차선 내에 있지 않는 경우에는 로봇을 선택된 차선에 보내기 위한 방향 명령이 결정되어야 한다. n_s 번째 차선으로의 방향각 $va(n_s)$ 는 로봇이 현재 위치에서 n_s 번째 차선으로 진입하기 위한 최소 조향각이므로, 방향 명령의 절대값은 $|va(n_s)|$ 보다 커야 한다. 그리고 차선을 구분할 때 목표 방향으로부터 경계각 ba 이내의 장애물들만을 고려하였으므로, 방향 명령의 절대값은 $|ba|$ 보다 작아야 한다. 따라서 방향 명령 hc 는 다음과 같이 주어진다.

$$hc = \begin{cases} gd, & \text{if } n_r = n_s \\ va(n_s) + \delta \cdot (ba - va(n_s)), & \text{if } n_r \neq n_s \end{cases} \quad (6)$$

여기에서 δ 는 $0 \leq \delta \leq 1.0$ 이며, 방향 명령을 방향각 $va(n_s)$ 와 경계각 ba 사이에서 어떤 값을 가지게 할 것인지를 결정한다. 만일 $\delta=0$ 이면 방향 명령은 방향각 $va(n_s)$ 가 되어, 로봇이 장애물과 여유 공간이 없이 n_s 번째 차선으로 진입한다. 만일 $\delta=1.0$ 이면 방향 명령은 경계각 ba 가 되

어 로봇이 항상 왼쪽 또는 오른쪽의 극단적인 방향으로 이동한다. 본 연구에서는 $\delta=0.5$ 로 하여 방향 명령이 n_s 번째 차선으로의 방향각과 경계각 사이의 중간값을 가지게 하였다. 그림 2 에 방향 명령, 방향각, 그리고 경계각 사이의 관계를 나타내었다.

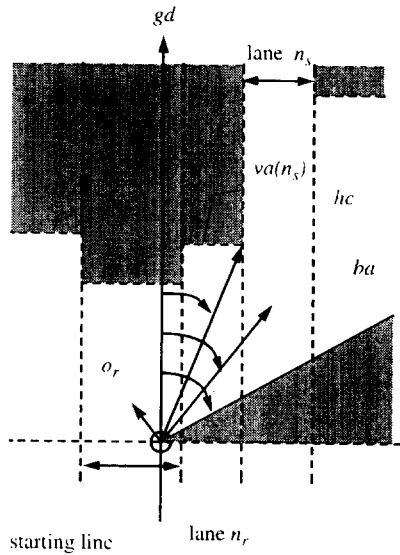


그림 2. 방향 명령과 방향각, 경계각 사이의 관계
Figure 2. The relationship between heading command, viewing angle, and boundary angle

4. 실험 결과와 검토

본 연구에서는 Xavier(그림 3)라는 이동 로봇을 이용한 실험을 통하여 차선-곡률 방법의 성능을 확인하였다.

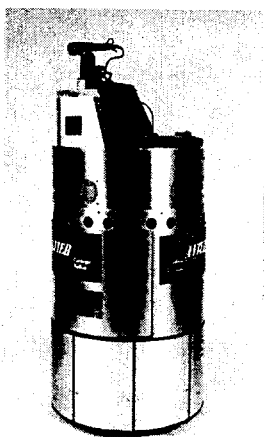


그림 3. 이동 로봇 Xavier
Figure 3. Xavier mobile robot

Xavier는 지름이 24"인 원통형 동체를 가진 RWI의 B24 이동 로봇을 기반으로 개발되었다. 병진 속도 명령과 회전 속도 명령에 의하여 4 개의 바퀴가 동기 구동(Synchro-drive) 방식으로 구동된다[11]. 최대 병진 속도는

90cm/sec 이고 최대 회전 속도는 167° /sec 이다. 본 연구에서는 속도 여유와 안전을 고려하여 최대 병진 속도와 회전 속도를 $tv_{max}=50cm/sec$ 와 $rv_{max}=60^\circ/sec$ 로 설정하였다. 24 개의 초음파 센서가 2 Hz의 주기로 장애물을 검출한다. 차선-곡률 방법은 Xavier상의 200MHz Pentium-Pro computer에서 처리되어 8Hz의 주기로 병진 속도 명령과 회전 속도 명령을 Xavier에 전달한다.

본 실험에서는 동일한 환경 하에서 차선-곡률 방법과 곡률-속도 방법에 의해 로봇이 주행하게 하여, 두 방법의 차이점을 비교하고, 차선-곡률 방법의 효율성을 보였다. 식 (1)의 목적 함수 $f(tv,rv)$ 에서 α 값들은 많은 실험과 시행착오를 통하여, 경험적으로 로봇을 충돌 없이 부드럽고 빠르게 이동시키는 값으로 결정되었다. α 값들은 다음과 같이 차선-곡률 방법을 사용하는 경우와 곡률-속도 방법을 사용하는 경우에서 서로 다르게 설정되었다.

$$\begin{aligned} \alpha_1=0.1, \alpha_2=0.6, \alpha_3=0.3 & : \text{차선-곡률 방법} \\ \alpha_1=0.6, \alpha_2=0.1, \alpha_3=0.3 & : \text{곡률-속도 방법 단독} \end{aligned} \quad (7)$$

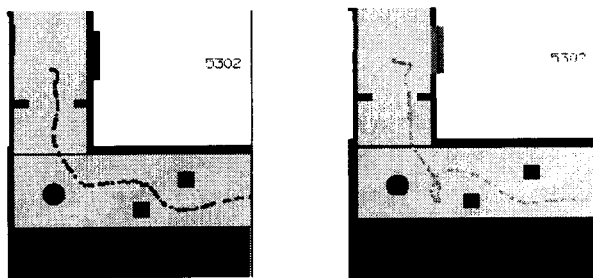
α_1 은 목적 함수 $f(tv,rv)$ 에서 무충돌 원호 경로의 길이의 대한 가중치로서, 곡률-속도 방법에서 장애물과의 충돌을 피하게 해주는 기능을 한다. 차선-곡률 방법에서는 일차적으로 차선 방법에서 장애물 회피에 대한 고려를 하기 때문에 이 값이 작은 반면, 곡률-속도 방법에서는 이 값에 의하여 장애물 회피를 모두 고려해주기 때문에 이 값이 커져야 장애물 회피 성능이 좋아진다. α_2 는 목적 함수 $f(tv,rv)$ 에서 목표 방향으로의 회전에 대한 가중치로서, 로봇이 좀더 짧은 시간 내에 목표 방향을 향하도록 해준다. 차선-곡률 방법에서는 α_1 을 작게할 수 있으므로, α_2 를 크게 하여 로봇이 좀더 빠르게 목표 방향을 향하게 한다. 곡률-속도 방법만을 사용하는 경우에는 충돌 회피를 위해서 α_1 이 커져야하므로 α_2 는 작아져서 목표 방향으로의 지향성이 떨어진다.

로봇의 실내 주행시 자주 만나게되는 다음 네 가지 경우에 대하여 차선-곡률 방법에 의한 주행 결과와 곡률-속도 방법만에 의한 주행 결과를 구하였다: (1) 3 개의 장애물이 있는 좌회전 길을 주행하는 경우, (2) 하나의 장애물이 있는 복도를 직선 주행하는 경우, (3) 넓은 복도에서 좁은 복도로 들어가는 경우, (4) 복도 우측의 좁은 입구에 들어가는 경우. 모든 경우에서 실내 지도와 장애물에 대한 정보는 미리 주어지지 않는다. 로봇은 주행을 시작할 때, 목표 방향 gd 만을 입력받고, 장애물에 대한 정보는 주행중 초음파 센서를 통하여 얻는다.

그림 4 는 첫 번째 경우에 대한 결과이다. 로봇은 위쪽에서 아래쪽을 향한 방향에서 출발하며 $gd=-90^\circ$ 로 주어져서, 모서리 부분에서 왼쪽으로 회전하도록 명령받는다. 모서리 부분에서 로봇은 두 가지의 충돌 회피 경로중 하나를 선택할 수 있다. 하나는 두 번째 장애물의 위쪽으로의 경로이고 다른 하나는 두 번째 장애물의 아래쪽으로의 경로이다. 아래쪽의 경로는 위쪽의 경로보다 좋다. 차선-곡률 방법

에서는 위쪽에서의 경로를 쉽게 찾는다. 그러나 곡률-속도 방법에서는 먼저 아래쪽으로의 경로를 선택하여 이동하다가 너무 좁아 통과하지 못하고, 마침내 위쪽에서의 무충돌 경로를 찾아낸다. 결과적으로, 곡률-속도 방법은 첫 시도에서는 보다 넓은 충돌 회피 경로를 찾지 못하고 지나치게 된다.

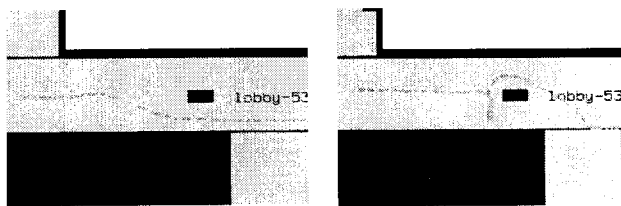
그림 5는 두 번째 경우에 대한 결과이다. 로봇은 오른쪽을 향한 상태에서 출발하며, 왼쪽에서 오른쪽 방향으로 이동하도록 $gd=0^\circ$ 이다. 차선-곡률 방법에서는 로봇이 이동 초기에서부터 장애물을 피하기 위해 아래쪽 방향으로 경로를 바꾼다. 그런데 곡률-속도 방법에서는 로봇이 장애물에 가까이 다가간 후 회피 동작을 시작하여 급격하게 회전한다. 이러한 결과의 원인은 곡률-속도 방법이 무충돌 원호 거리를 기준으로 경로를 선택하기 때문이다. 반면, 차선-곡률 방법은 무충돌 직선 거리와 무충돌 공간의 폭을 기준으로 경로를 선택하므로 곡률-속도 방법에 비해 초기 단계부터 무충돌 공간을 찾아 충돌 회피 동작을 시작한다.



(a) 차선-곡률 방법 (b) 곡률-속도 방법

그림 4. 첫 번째 경우에서의 Xavier의 무충돌 이동

Figure 4. Collision-free movement of Xavier for the first case.



(a) 차선-곡률 방법 (b) 곡률-속도 방법

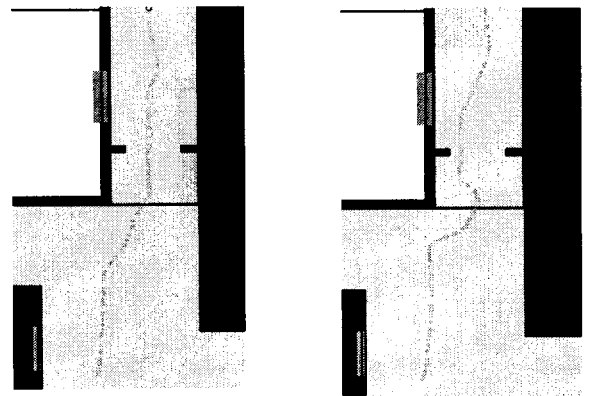
그림 5. 두 번째 경우에서의 Xavier의 무충돌 이동

Figure 5. Collision-free movement of Xavier for the second case.

그림 6은 세 번째 경우에 대한 결과이다. 로봇은 초기에 위쪽 방향을 향하고 있으며, $gd=0^\circ$ 에서 아래쪽의 넓은 복도 공간에서 위쪽의 좁은 복도로 진입하여야 한다. 곡률-속도 방법에서는 로봇이 거의 직선 이동하여 위쪽의 벽에 가까이 다가간 후에야 이 벽을 피한 후, 좁은 복도에 진입하기 위해 오른쪽으로 급히 방향을 바꾼다. 차선-곡률 방법은 비교적 초기 단계부터 진입할 좁은 복도를 찾아서 로봇을 이동시킨다.

그림 7은 네 번째 경우에 대한 결과이다. 로봇은 좌측

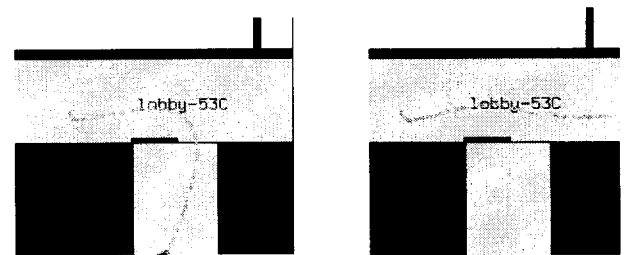
위쪽에서 이동을 시작하며 초기에 오른쪽을 향하고 있다. 그리고 아래쪽의 좁은 입구를 통과하여 이동하기 위해 $gd=90^\circ$ 이다. 차선-곡률 방법에서 로봇은 목표 방향의 입구를 찾아서 부드럽게 통과한다. 그런데 곡률-속도 방법에서는 로봇이 직각 방향의 입구를 찾지 못하고 계속 직진한다(그림상에서 오른쪽으로 이동하다, 그림에는 나타나있지 않은 다음의 복도에서 우회전한다). 이것은 첫 번째 경우에 대한 실험에서 곡률-속도 방법이 무충돌 경로를 지나친 것과 비슷한 결과이다. 여기에서 입구의 폭은 120cm 인데, 만일 140cm로 하면 곡률-속도 방법도 아래 방향의 입구를 찾아서 로봇을 목표 방향으로 이동하게 할 수 있다.



(a) 차선-곡률 방법 (b) 곡률-속도 방법

그림 6. 세 번째 경우에서의 Xavier의 무충돌 이동

Figure 6. Collision-free movement of Xavier for the third case.



(a) 차선-곡률 방법 (b) 곡률-속도 방법

그림 7. 네 번째 경우에서의 Xavier의 무충돌 이동

Figure 7. Collision-free movement of Xavier for the fourth case.

이상의 네 가지 경우에서와 같이 곡률-속도 방법에 의해 로봇이 목표 방향의 경로를 찾지 못하거나 비효율적으로 이동하는 원인을 그림 8에 의해 설명할 수 있다. 곡률-속도 방법은 무충돌 직선 거리가 아닌 원호 거리를 계산하여, 원호 거리가 긴 경로를 선택한다. 그리고 곡률-속도 방법만을 사용하면 식 (1)의 목적 함수에서 α_1 이 크므로, 그림 8에서 원호 \widehat{aa} 나 \widehat{bb} 를 따라가는 경로보다 \widehat{ac} 를 따라가는 경로가 선택된다. 그러나 원호 \widehat{oc} 를 따라가는 경로는 충돌

회피의 효율성 면에서 원호 \widehat{oa} 나 \widehat{ob} 를 따라가는 경로보다 우수하지 않다. 만일 곡률-속도 방법만을 사용하면 α_1 을 작게 하고 α_2 를 크게 하면, 그림 8 과 같은 경우에는 성능을 향상시킬 수 있으나, 그 이외의 경우에는 오히려 장애물 회피 능력이 떨어져서, 전반적인 충돌 회피 성능이 떨어진다. 반면 차선-곡률 방법에서는 차선 방법에 의해 차선의 폭과 무충돌 직선 거리 정보로부터 보다 효율적인 방향 명령 hc 를 구한다.

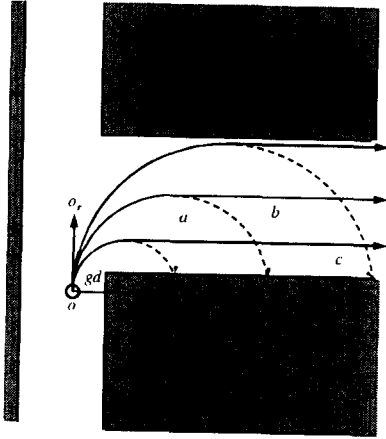


그림 8 곡률-속도 방법에서의 경로 선택
Figure 8. Path selection of Curvature-Velocity Method

이상의 결과들에서 보인바와 같이 차선-곡률 방법은 곡률-속도 방법만을 사용했을 때 나타나는 문제점들을 해결할 수 있다. 그런데 α 값들의 선택은 차선-곡률 방법의 성능을 크게 좌우한다. 만일 α_1 을 0.1 이상으로 증가시키고, α_2 는 0.6 이하로 감소시키면, 위의 실험들에서 보인 차선-곡률 방법의 장점들이 점점 감소한다. 그리고, 곡률-속도 방법만을 사용하는 경우와 같이 $\alpha_1 = 0.6$, $\alpha_2 = 0.1$ 로 하면 차선-곡률 방법에 의한 결과와 곡률-속도 방법에 의한 결과가 비슷해져서, 차선-곡률 방법의 장점을 살리지 못하게 된다. 위의 네 가지 결과는 차선-곡률 방법과 곡률-속도 방법의 결과가 큰 차이를 보이는 대표적인 경우들에 대한 실험 결과들이다. 그러나 이상의 경우들에 비해 충돌 회피가 비교적 용이한 경우들에서는 차선-곡률 방법과 곡률-속도 방법의 결과들이 큰 차이를 보이지 않는다. 그 이유는 충돌 회피가 용이한 경우에는 곡률-속도 방법이 구한 로봇의 최적 이동 방향이 차선-곡률 방법이 구한 최적 이동 방향과 비슷해지기 때문이다. 하지만, 차선-곡률 방법은 곡률-속도 방법을 포함하기 때문에 일반적으로 곡률-속도 방법보다 나쁜 결과를 보이지는 않는다.

차선-곡률 방법도 항상 해를 갖지는 못한다. 통로의 폭이 매우 좁은 경우에는 무충돌 공간을 찾지 못하기도 하며, 로봇이 구석 부분에 빠지게 되면 다시 무충돌 공간으로 빠져

나오지 못하기도 한다.

차선 방법과 곡률-속도 방법은 각기 다른 측면에서 충돌 회피를 고려한다. 곡률-속도 방법은 무충돌 원호 길이가 긴 경로를 선택하는 반면, 차선 방법은 무충돌 직선 길이가 길고 무충돌 영역의 폭이 넓은 공간으로의 경로를 선택한다. 차선-곡률 방법은 두 단계를 거침으로서 곡률-속도 방법만을 사용하거나 차선 방법만을 사용하는 경우에 비해서 다양한 측면에서 충돌 회피를 고려한다. 또한 차선-곡률 방법을 사용할 때의 α 값들은 곡률-속도 방법만을 사용하는 경우의 α 값들과 다르게 설정되므로, 차선-곡률 방법 속에서의 곡률-속도 방법은 곡률-속도 방법만을 사용하는 경우에 비해서 장애물 회피를 덜 고려하게 된다.

차선-곡률 방법은 두 번째 단계에서 곡률-속도 방법을 사용하여 병진 운동과 회전 운동의 속도와 가속도 제한을 고려한다. 따라서 곡률-속도 방법이 인공 전위계 방법[2,3,4]과 같은 방향 명령 방식에 비해서 가지는 우수성들을 그대로 유지하게 되어, 로봇의 이동 궤적이 부드러워지고 충돌 회피 능력이 향상된다. 이에 비해 인공 전위계 방법은 로봇의 속도와 가속도 제한을 적절히 고려하지 못하여 곡률-속도 방법에 비해서 덜 부드러운 충돌 회피 경로를 생성한다[5].

로봇의 동력학적인 면을 고려하는 몇가지 기존의 장애물 회피 방법들[12,13]은 실시간 적용을 목표로하지 않아서 많은 계산을 필요로한다. 반면, 차선-곡률 방법은 8Hz의 주기로 속도 명령을 구할 수 있어서, 실시간 적용에 적합하다. 계산 시간 면에서는 곡률-속도 방법이 차선-곡률 방법보다 우수하나, 실험에서와 같이 8 Hz의 주기로 속도 명령을 구할 수 있을 정도의 계산량이라면, 계산 시간은 차선-곡률 방법의 성능에 큰 영향을 미치지 못한다.

5. 결론

본 연구에서는 실내 이동 로봇의 장애물 회피를 위한 새로운 방법을 제시하였다. 이 방법은 속도 공간 명령 방식(곡률-속도 방법)과 방향 명령 방식(차선 방법)을 융합하여 두 방식의 단점들을 서로 보완하게 하였다. 차선 방법에서는 각 차선의 장애물까지의 직선 거리와 차선의 폭을 기준으로 최적의 차선을 선택하므로, 곡률-속도 방법만을 사용하는 경우에 발생할 수 있는 "로봇이 목표 방향의 좁은 복도나 통로를 지나쳐버리는 문제점"을 해결할 수 있다. 그리고 차선 방법과 함께 곡률-속도 방법을 사용함으로써, 로봇의 이동 목표 방향은 물론 동력학적인 제한 조건을 고려하여 병진 속도 명령과 이동 속도 명령을 구한다.

제안된 방법을 적용한 실험 결과, 많은 경우 차선-곡률 방법이 곡률-속도 방법에 비해서 보다 안전하고 부드럽게 로봇을 장애물이 없는 공간으로 유도함을 볼 수 있었다. 특히 차선-곡률 방법은 경우에 따라서 곡률-속도 방법으로는 찾지 못하는 무충돌 경로를 찾기도 하였다. 이러한 결과들로부터, 차선-곡률 방법은 방향 명령 방식과 속도 공간 명령 방식을 결합함으로써, 로봇을 좀더 안전한 경로로 부드럽게 이동시킴을 알 수 있다.

본 연구에서 제안한 차선-곡률 방법을 더욱 발전시키기 위하여, 차선 선택 함수에서 β 값들과 곡률-속도 방법의 목

적 함수에서 α 값들을 최적의 값들로 조정할 수 있는 조직적인 방법이 더 연구되어야 할 것으로 생각된다. 그리고 로봇이 이동 중에 주변 환경(장애물의 배치, 크기 등)의 변화에 따라 이 값들을 상황에 맞는 최적의 값들로 적응시키는 방법도 고려될 수 있다고 판단된다.

감사의 글

이 연구는 한국과학재단의 "1996년도 하반기 해외 Post-Doc. 연수 지원 프로그램"에 의한 지원을 받아 수행되었습니다.

참 고 문 헌

[1] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," The Int. Journal of Robotics Research, Vol. 5. No.3, pp.72-89, Fall 1986.

[2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The Int. J. Robotics Research, Vol.5. No.1, pp.90-98, Spring 1986.

[3] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," IEEE Trans. on Robotics and Automation, Vol. 8, No.1, pp.23-32, Feb. 1992.

[4] J. Borenstein and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," IEEE Trans. Robotics Automat., pp.278-288, Vol.7, No.3, June 1991.

[5] R. G. Simmons, "The Curvature-Velocity Method for Local Obstacle Avoidance," International Conference on Robotics and Automation, Minneapolis MN, pp.3375-3382, April 1996.

[6] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Stricou, and S. Thurn, "The mobile robot Rhino," AI Magazine, 16:2, pp.278-288, Summer 1995.

[7] W. Feiten, R. Bauer, and G. Lawitzky, "Robust obstacle avoidance in unknown and cramped environment," In Proc. IEEE Int. Conf. Robotics and Automation, pp. 2412-2417, San Diego, CA, May 1994.

[8] D. Fox, W. Burgard, and S. Thurn, "The dynamic window approach to collision avoidance," Tech Report IAI-TR-95-13, CS department, University of Bonn, 1995.

[9] A. Kelly, "An intelligent predictive control approach to the high speed cross country autonomous navigation problem," Tech Report CMU-CS-TR-95-33, School of Computer Science, Carnegie Mellon Univ.

[10] Nak Yong Ko and Reid G. Simmons, "The Lane-Curvature Method for Local Obstacle Avoidance," in Proc. 98 IEEE/RSJ Int. Conf.

Intelligent Robots and Systems, Victoria, Canada, pp.1615-1621, October 1998.

[11] R. Simmons, R. Goodwin, K. Zita Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in Proc. Autonomous Agents '97, pp. 245-252, Marina del Rey, CA, February 1997.

[12] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in Proc. IEEE Int. Conf. on Robotics and Automation, Scottsdale AZ, pp. 2-7, May 1989.

[13] J. C. Latombe, "Robot motion planning," Kluwer Academic Publishers, 1991.

저 자 소 개



고 낙 용(高樂溶)

1961년 3월 13일 생. 1985년 서울대 공대 제어계측공학과 졸업. 1987년 동 대학원 제어계측공학과 졸업(석사). 1993년 동 대학원 제어계측공학과 졸업(공학박). 1997년~1998년 Carnegie Mellon Univ. 방문 연구, 1992년~현재 조선대학교 전기제어계측공학부 부교수



이 상 기(李相基)

1960년 8월 14일 생. 1981년 서울대학교 항공공학과 졸업. 1984년 동 대학원 항공공학과 졸업(석사). 1996년 동 대학원 항공우주공학과 졸업(공학박). 1984년 1월~1989년 4월 (주)대한항공 한국항공기술연구소 연구원. 1989년 5월~현재 조선대학교 항공조선공학부 부교수