

# Gyro/Vision Sensor Module 을 이용한 주행 로봇의 미끄러짐 및 넘어짐 회피 제어 기법

## Control Method of Mobile Robots for Avoiding Slip and Turnover on Sloped Terrain Using a Gyro/Vision Sensor Module

이 정 희\*, 박 재 병, 이 범 희  
(Jeong-Hee Lee, Jae-Byung Park, and Beom-Hee Lee)

**Abstract :** This article describes the control method of mobile robots for avoiding slip and turnover on sloped terrain. An inexpensive gyro/vision sensor module is suggested for obtaining the information of terrain at present and future. Using the terrain information and the robot state, the maximum limit velocity of the forward velocity of the robot is defined for avoiding slip and turnover of the robot. Simultaneously the maximum value of the robot velocity is reflected to an operator in the form of reflective force on a force feedback joystick. Consequently the operator can recognize the maximum velocity of the robot determined by the terrain information and the robot state. In this point of view, the inconsistency of the robot movement and the user's command caused by the limit velocity of the robot can be compensated by the reflective force. The experimental results show the effectiveness of the suggested method.

**Keywords :** laser line generator, gyro sensor, limit velocity, artificial reflective force, sloped terrain, slip and turnover avoidance

### I. 서론

일반적인 원격 주행 로봇 제어 분야에서는 장애물을 감지하고 그 정보를 이용하여 기하학적 경로(geometric path)를 구하기 위한 전체 경로 계획(global path planning)과 실시간 장애물 회피를 위한 국부 경로 계획(local path planning)을 수립하는 연구가 매우 활발하게 진행 되어왔다[1-3]. 그러나 경사가 있는 지형에서 주행하는 로봇의 경우 장애물뿐만 아니라 로봇이 주행하는 지형조건까지 고려해야 한다. 그렇기 때문에 경사가 있는 지형에서 주행하는 로봇을 제어할 경우 지형 정보를 고려한 새로운 제어 기법이 필요하다.

본 논문에서는 로봇의 안정적인 주행 제어를 위해 로봇의 주행 속도와 전방 지형에 대한 정보를 이용해 로봇의 미끄러짐과 넘어짐 회피를 위한 주행 속도의 제한 조건을 분석할 것이다. 이때 로봇의 주행 정보는 로봇 내부 센서를 이용해 쉽게 얻을 수 있으나 전방 지형에 대한 정보를 획득하기 위해서는 외부 센서가 필요하다. 주행 로봇에 일반적으로 사용되는 초음파 센서는 방향 부정확성(poor directionality), 외란 민감성(frequent misreadings), 반사각 민감성(specular reflections)과 같은 단점을 가지고 있어 전방 지형 정보 획득에는 부적합하다[4-6]. 따라서 본 논문에서는 전방 지형 정보 획득을 위하여 gyro sensor와 laser line generator, 그리고 camera를 이용한 sensor module을 제작하였다. 전방 지형 정보 획득을 위한 sensor module은 로봇 전방 지형에 line 형태의 laser beam을 투사하여 그것을 camera로 감지하고, 기하학적 분석(geometric analysis)를 통해 로봇 전방 지형에 대한 정보를 획득하며 gyro sensor에 의해 얻어진 정보와 융합하는 과정을 통해 전

역 좌표계에 대한 로봇 전방 지형 정보를 구할 수 있다.

이렇게 획득된 로봇의 전방 지형 정보를 이용해 로봇의 주행 속도에 따른 미끄러짐 및 전복 회피 조건을 구하기 위해 주행 로봇의 운동 방정식을 사용한다[7]. 본 논문에서는 운동 방정식을 단순화 시키기 위하여 몇 가지 가정을 제시하며 앞에서 구한 로봇 전방 지형의 정보에 의해 정의된 지형에 대한 Parameter들을 이용하여 로봇의 주행 속도에 대한 제한 조건을 구할 것이다[8].

로봇의 미끄러짐 및 전복 회피를 위한 로봇의 주행 속도의 제한 조건을 이용하여 작업자(operator)가 제어할 수 있는 로봇의 최대 주행 속도를 제한하고, haptic device의 일종인 힘 되먹임 조이스틱(force feedback joystick)을 이용하여 작업자가 로봇의 주행 속도가 한계 속도에 의해 제한되었는지를 인지 할 수 있도록 인공 반사 힘(artificial reflective force)을 생성하는 알고리즘을 구현하였다[9].

본 논문의 구성은 다음과 같다. 2장에서는 전체 시스템의 구성에 대해 설명하고, 3장에서는 지형 정보를 획득하는 방법에 대해 설명한다. 4장에서는 주행 로봇의 주행 속도 제한 조건을 분석하고, 5장에서는 주행 속도 제한 조건에 따른 로봇의 주행 제어 방법에 대해 설명한다. 6장에서는 실험을 통해 제안된 알고리즘의 효과를 검증하고, 7장에서 본 논문의 결론을 맺는다.

### II. 전체 시스템 구성

전체 시스템은 그림 1과 같이 원격 모니터링 프로그램과 로봇 제어 프로그램 두 부분으로 구성되며 서로 무선 랜(wireless LAN)을 통해 통신을 한다.

로봇 제어 프로그램은 주행 로봇에 내장된 프로그램으로서 로봇의 기구학 및 동력학을 고려하여 로봇을 구동 명령을 생성 할 뿐만 아니라 로봇의 상태 정보와 지형 정보를 획득한다. 이때 전방 지형 정보를 획득하기 위해 그림 2에 보이

\* 책임저자(Corresponding Author)

논문접수 : 2004. 1. 20., 채택확정 : 2005. 3. 17.

이정희, 박재병, 이범희 : 서울대학교 전기컴퓨터공학부

(meeckee2@snu.ac.kr/pjb0922@snu.ac.kr/bhlee@asri.snu.ac.kr)

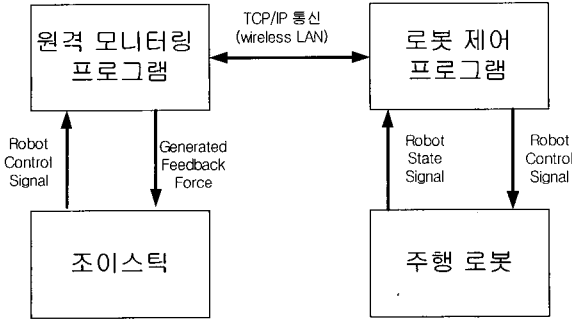


그림 1. 전체 시스템 구성도.  
Fig. 1. System block diagram.

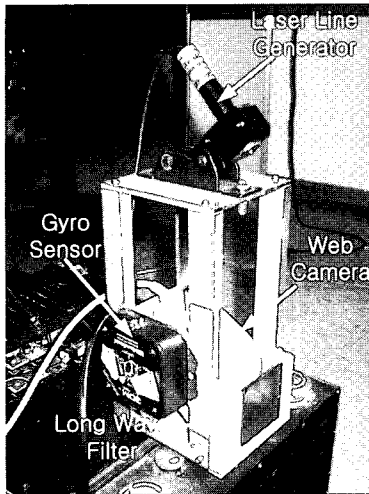


그림 2. Gyro/vision 센서 모듈.  
Fig. 2. Gyro/vision sensor module.

표 1. DirectX 8.0 SDK에서 제공하는 대표적인 인공 반사 힘.  
Table 1. Typical reflective forces by DirectX 8.0 SDK.

힘의 형태	설 명
Constant Force	한 방향으로 일정한 크기의 힘
Ramp Force	힘의 크기가 일정하게 증가하거나 감소하는 힘
Periodic Effect	정의된 파형에 따라 진동하는 힘
Condition Force	축에 따른 움직임이나 위치 변화에 대응하는 힘

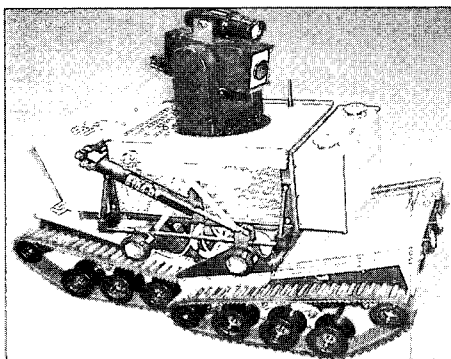


그림 3. KIST에서 개발한 이동 로봇 ROBHAZ-DT  
Fig. 3. Mobile robot, ROBHAZ-DT developed by KIST

는 gyro/vision sensor module을 제작하였다. gyro/vision sensor module은 저가의 컬러 web camera와 파장 대역이 635nm~639nm인 laser line generator, 그리고 gyro sensor로 구성되어 있다. 이때 web camera에 의해 습득된 image 정보의 noise를 제거하기 위해 630nm의 cutoff wavelength를 갖는 long wave pass filter를 사용하였다.

원격 모니터링 프로그램은 힘 되먹임 조이스틱(force feedback joystick)을 통해 작업자(operator)의 로봇 제어 명령을 받아들이고 이 명령을 로봇 제어 프로그램에 무선 통신을 통해 전달한다. 이와 동시에 로봇 제어 프로그램으로부터 전달된 로봇의 상태 및 지형 정보를 이용해 로봇의 미끄러짐 및 전복 회피를 위한 로봇 제어 명령을 생성하고 작업자가 로봇 제어 상태를 인지할 수 있도록 인공 반사 힘(artificial reflective force)을 생성하게 된다.

작업자의 제어 명령을 받아들이는 힘 되먹임 조이스틱(force feedback joystick)은 haptic device의 일종으로서 명령을 받아들일 수 있을 뿐만 아니라 인공 반사 힘(artificial reflective force)을 생성하여 작업자에게 전달 할 수 있다. 사용하는 조이스틱은 2DOF를 갖으며 조이스틱의 앞뒤 방향에 대응되는 Y축과 좌우 방향에 대응되는 X축으로 구성되며 각 축에 독립적인 인공 반사 힘을 생성할 수 있다. 이때 작업자의 로봇의 주행 속도 명령과 회전 속도 명령은 각각 조이스틱의 Y, X축에 대응된다. 힘 되먹임 조이스틱 인터페이스를 위해 DirectX 8.0 SDK(Software Development Kit)을 이용하였으며 제공하는 인공 반사 힘의 형태는 표 1과 같이 4가지 형태가 있다. 본 논문에서는 조이스틱의 위치 변화에 따라 생성되는 힘의 형태인 condition force를 이용하였다.

주행 로봇은 한국 과학 기술 연구원(KIST)에서 개발한 ROBHAZ-DT를 사용하였으며 그 형태는 그림 3과 같다. ROBHAZ-DT는 4개의 바퀴가 달린 일반적인 주행 로봇과는 달리 무한 레노 트랙을 이동부로 사용하기 때문에 험준한 지형에서도 주행이 가능하다.

### III. 지형 정보 획득

전방 지형 정보를 획득하기 위해서 그림 4와 같이 주행 로봇 전방 상단에 Gyro/vision sensor module을 설치하였다.

#### 1. Camera 좌표계의 기하학적 분석

Camera에 의해 얻어지는 정보는 2차원의 image정보이다. 하지만 전방 지형 정보를 정확히 알기 위해서는 3차원 좌표 정보가 필요하기 때문에 본 논문에서는 기하학적 분석(geometric analysis)을 통해 2차원의 image 정보에서 depth 정보를 계산하여 3차원의 좌표 정보를 구하였다.

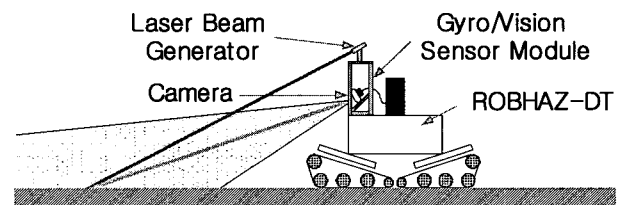


그림 4. 주행 로봇과 gyro/vision sensor module.  
Fig. 4. Mobile robot and gyro/vision sensor module.

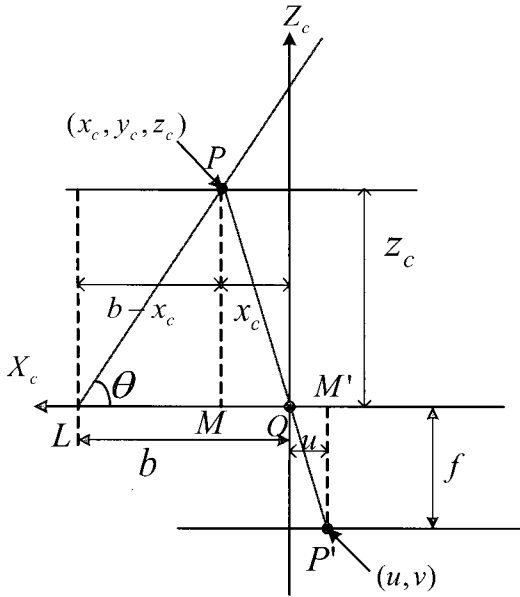


그림 5. Camera 좌표계의 기하학적 분석도.  
Fig. 5. Geometric analysis of camera coordinate.

그림 5는 camera 좌표계의 기하학적 분석도이다. 여기에서  $O$ 는 camera 좌표계의 원점,  $L$ 은 laser line generator의 위치,  $P'(u, v)$ 는 camera image 평면에서의 laser line의 좌표값,  $f$ 는 camera의 focal length,  $b$ 는 laser line generator와 camera간의 거리,  $\theta$ 는 camera와 laser beam간의 각도, 그리고  $P(x_c, y_c, z_c)$ 는 camera 좌표계에서의 좌표값을 나타낸다.

그림 5에서 보이는 것과 같이 기하학적 분석을 통해 2차원의 Image 정보로부터 3차원의 좌표 정보를 획득하는 과정은 다음과 같다.

그림 5에서

$$z_c = (b - x_c) \tan \theta \tag{1}$$

이다.

여기서  $\triangle OPM$  과  $\triangle OP'M'$ 은 닮은꼴이므로,

$$x_c : z_c = u : f \tag{2}$$

와 같은 관계가 성립하므로  $x_c$ 는 다음과 같이 구할 수 있다.

$$x_c = \frac{u}{f} \cdot z_c \tag{3}$$

다음으로 (1)을 (2)에 대입하면,

$$x_c = \frac{u}{f} \cdot (b - x_c) \tan \theta \tag{4}$$

이 되며, 다음과 같이 정리할 수 있다.

$$x_c = \frac{ub}{f \cot \theta + u} \tag{5}$$

같은 방법으로  $y_c$ 와  $z_c$ 도 다음과 같이 구할 수 있다.

$$y_c = \frac{vb}{f \cot \theta + u} \tag{6}$$

$$z_c = \frac{fb}{f \cot \theta + u} \tag{7}$$

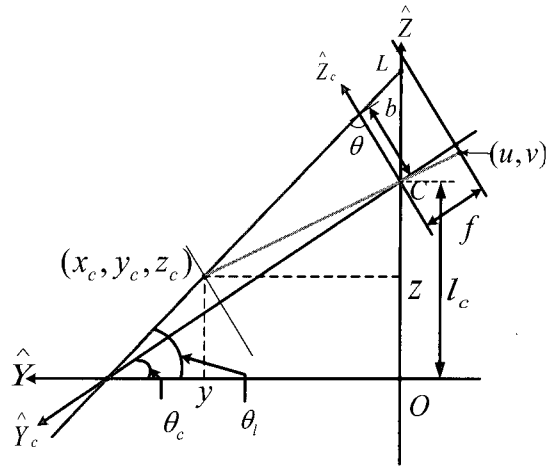


그림 6. Camera좌표계와 로봇 좌표계의 관계도.  
Fig. 6. Camera coordinate and robot coordinate.

결과적으로 (5), (6), (7)은 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} x_c & y_c & z_c \end{bmatrix} = \frac{b}{f \cot \theta + u} \begin{bmatrix} u & v & f \end{bmatrix} \tag{8}$$

따라서, (8)에 의해 camera에 의해 얻어진 2차원 image 정보에서 3차원 좌표 정보를 구할 수 있다.

### 2. 좌표 변환

앞에서 구한 3차원 좌표는 camera좌표계를 기준으로 구한 값이다. 따라서 로봇을 기준으로 한 지형 정보를 얻기 위해서는 camera 좌표계에 대한 정보를 로봇 좌표계에 대한 정보로 변환하여야 한다. 그림 6은 camera좌표계와 로봇 좌표계의 관계도이다.

여기에서  $\hat{Y}$  축은 지면과 평행하게 정의되며,  $\theta_c$ 는 지면을 기준으로 한 camera중심 축과의 각도,  $\theta_l$ 은 지면과 laser beam과의 각도,  $l_c$ 은 지면으로부터 camera의 높이, 그리고  $b$ 는  $\hat{Z}_c$  축 상에서의 camera와 laser line generator사이의 거리이다. 그림 6에서 보이는 것과 같이 camera좌표계와 로봇 좌표계의 관계를 통해 두 좌표계 사이의 변환식이 (9)와 같이 정의됨을 쉽게 알 수 있고 이때  $z$ 는 감지된 지형의 높이,  $y$ 는 감지된 지형과 로봇사이의 거리가 된다.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_c & \sin \theta_c & 0 \\ 0 & -\sin \theta_c & \cos \theta_c & l_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{9}$$

### 3. 전방 지형의 기울기 각 계산

앞에서 구한 변환식들을 이용하여 전방 지형 기울기 각에 따른 로봇의 roll, pitch 각을 예측 할 수 있다. 로봇의 roll, pitch 각은 그림 7과 같이 정의된다. 그림에서 볼 수 있듯이 roll각도는 로봇이 주행하는 축에 대한 값으로 좌측이 높을 경우를 (+) 방향으로 정의하였고, pitch각도는 로봇이 주행하는 축의 수직인 축에 대한 값으로 앞쪽이 높을 경우를 (+) 방향으로 정의하였다.

전방 지형에 따른 로봇의 roll각도는 camera에 의해 감지

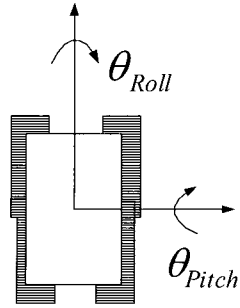


그림 7. 로봇의 roll 각도와 pitch 각도의 방향.  
Fig. 7. Directions of the roll and pitch angles of the robot.

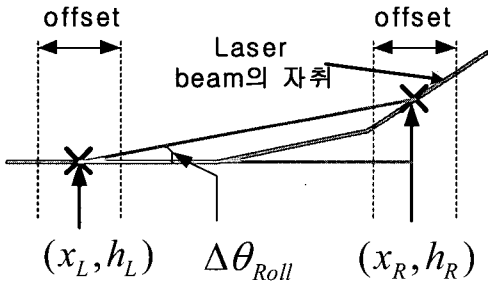


그림 8. 감지된 laser line과 roll 각도.  
Fig. 8. Detected laser line and roll angle.

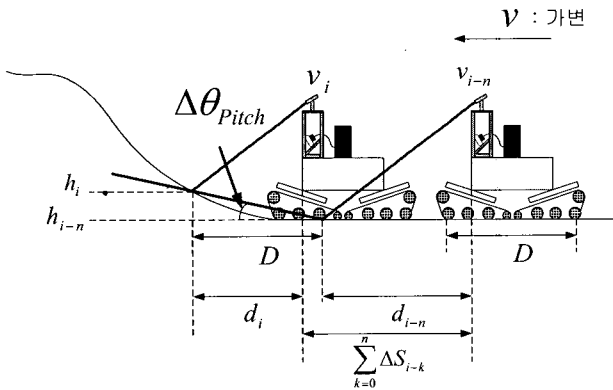


그림 9. ROBHAZ-DT의 자취.  
Fig. 9. Trace of ROBHAZ-DT.

된 laser line 정보를 이용하여 계산하는데, 감지된 laser line의 정보를 모두 처리하는 것은 계산량이 많기 때문에 로봇의 좌·우측 트랙 부분의 정보만을 추출하여 roll각도를 계산한다. 그림 8은 camera에 의해 감지된 laser line의 자취와 그것의 좌·우측 트랙 부분의 값을 나타낸 것이다. 여기서  $h_L$ 과  $h_R$ 은 전방 지형에 따른 좌측 트랙의 높이와 우측 트랙의 높이,  $x_L$ 과  $x_R$ 은 좌측 트랙의 X 좌표값과 우측 트랙의 X 좌표값을 각각 나타낸다. 이때  $h_L$ 과  $h_R$ 을 구할 때 noise의 영향을 최소화 하기 위해 offset이라는 구간을 두어 평균값을 취하였다.

결과적으로 그림 8에 보이는 전방 지형에 의한 로봇의 roll 각도는 (10)과 같이 구할 수 있다.

$$\Delta\theta_{Roll} = \tan^{-1} \left( \frac{h_L - h_R}{x_L - x_R} \right) \quad (10)$$

다음으로 전방 지형에 따른 로봇의 pitch각도를 알기 위해 각 sampling time동안의 로봇이 주행한 거리를 계산하여 연속적으로 저장한 후, 현재 트랙 앞쪽의 높이와 트랙 뒤쪽의 높이 차이와 트랙의 길이를 이용하여 pitch 각을 계산한다. 이때 pitch 각을 계산하기 위해 누적된 지형 image 정보를 이용한다.

그림 9에 보이는 것과 같이 로봇이 주행할 때,  $v_i$ 는  $i$ 번째 로봇의 속도,  $h_i$ 는  $i$ 번째 로봇의 높이,  $\Delta S_i$ 는  $i-1$ 번째에서  $i$ 번째로 주행할 때의 로봇의 주행 거리, 그리고  $D$ 는 로봇의 앞·뒤 트랙의 거리이다. 여기에서  $i-1$ 번째에서  $i$ 번째 까지 로봇이 주행한 거리를  $\sum_{k=0}^n \Delta S_{i-k}$ 와 같이 표현할 수 있는데, 이때 로봇이 주행한 거리가  $D$ 보다 커질 경우의  $n$  값을 구하여 pitch각도를 구할 수 있다. 즉,

$$\sum_{k=0}^n \Delta S_{i-k} \geq D \quad (11)$$

일 때, pitch각도는

$$\Delta\theta_{Pitch} = \tan^{-1} \left( \frac{h_i - h_{i-n}}{\sum_{k=0}^n \Delta S_{i-k}} \right) \quad (12)$$

가 된다.

이때 위에서 구한 roll각도  $\Delta\theta_{Roll}$ 과 pitch각도  $\Delta\theta_{Pitch}$ 는 로봇 좌표계를 기준으로 구해진 값이기 때문에 gyro sensor에 의해 구해진 값  $\theta_{3DM-Roll}$ ,  $\theta_{3DM-Pitch}$  값을 이용하여 중력을 기준으로 정의된 전역 좌표계에 대한 좌표 값으로 다음과 같이 구할 수 있다.

$$\theta_{Roll} = \Delta\theta_{Roll} + \theta_{3DM-Roll} \quad (13)$$

$$\theta_{Pitch} = \Delta\theta_{Pitch} + \theta_{3DM-Pitch} \quad (14)$$

#### IV. 주행 속도의 제한 조건

##### 1. 주행 로봇의 운동 방정식

로봇의 미끄러짐이나 넘어짐에 대한 제한 조건을 구하기 위해서는 주행 로봇의 운동 방정식을 고려하여야 한다. 본 논문에서는 다음과 같은 몇 가지 가정을 통해서 운동 방정식을 최대한 단순화 하였다.

가정 1: 로봇이 발생시키는 모든 힘은 네 개의 트랙에 똑같이 배분되어 지면으로 전달된다.

가정 2: 로봇의 무게는 네 개의 트랙에 똑같이 분배된다.

가정 3: 주행 로봇의 모든 질량은 무게 중심 한 점에 몰려 있다.

가정 1, 2에 의해서 네 개의 트랙에 전달되는 각각의 마찰력은 하나의 전체적인 마찰력으로 합쳐서 생각할 수 있고, 가정 3에 의해서 주행 로봇이 방향을 변화시킬 때 자체의 관성 모멘트에 의한 영향을 무시할 수 있다. 가정 1~3에 의해

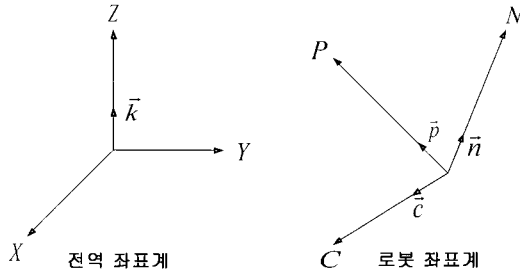


그림 10. 전역 좌표계와 로봇 좌표계.  
Fig. 10. Universal coordinate and robot coordinate.

운동 방정식은 하나로 합쳐진 마찰력과 중력, 수직항력과 관성력만을 가지고 구할 수 있다.

그림 10은 전역 좌표계와 로봇 좌표계를 나타낸다. 여기서  $\bar{p}$  는 로봇의 주행방향에 대한 단위벡터,  $\bar{n}$  는 지면에 대한 법선벡터, 그리고  $\bar{c}$  는  $\bar{p}$ ,  $\bar{n}$  에 수직인 단위벡터로 정의되며 로봇이 회전주행 할 때의 구심력 방향이 된다. 그리고 단위벡터  $\bar{k}$  를 전역 좌표계에 대해서  $\bar{k} = (0, 0, 1)$  으로 정의하였다.  $\bar{k}$  를 로봇 좌표계의 각 축에 대한 단위벡터와 내적하면,

$$k_p = \bar{k} \cdot \bar{p} \tag{15}$$

$$k_c = \bar{k} \cdot \bar{c} \tag{16}$$

$$k_n = \bar{k} \cdot \bar{n} \tag{17}$$

이 되며, 이때  $\bar{k}_p, \bar{k}_c, \bar{k}_n$  을 지형에 대한 Parameters로 정의하였다.

마찰력은 진행방향에 대해 접히는 방향과 수직하는 방향으로 나누어 생각 할 수 있다.

$$\bar{F} = f_p \bar{p} + f_c \bar{c} \tag{18}$$

그리고 수직 항력은 로봇의 수직방향에 대해 정의하였다.

$$\bar{R} = R \bar{n} \tag{19}$$

또한 중력은 전역 좌표계에서 정의된 단위벡터  $\bar{k}$  에 의해서 다음과 같이 나타낼 수 있다.

$$\bar{g} = -g\bar{k} = (0, 0, -g) \tag{20}$$

그러면 주행 로봇의 운동 방정식은 다음과 같이 선속도  $v$  와 선가속력  $a$  에 의해서 나타낼 수 있다.

$$f_p \bar{p} + f_c \bar{c} + R \bar{n} - mg\bar{k} = \frac{mv}{r} \bar{c} + ma\bar{p} \tag{21}$$

그리고 (21)은 로봇 좌표계의 각 축에 대한 단위벡터와 내적하여 다음과 같이 세 개의 스칼라 식으로 분리하여 나타낼 수 있다.

$$f_p = mgk_p + ma \tag{22}$$

$$f_c = mgk_c + \frac{mv}{r} \tag{23}$$

$$R = mgk_n \tag{24}$$

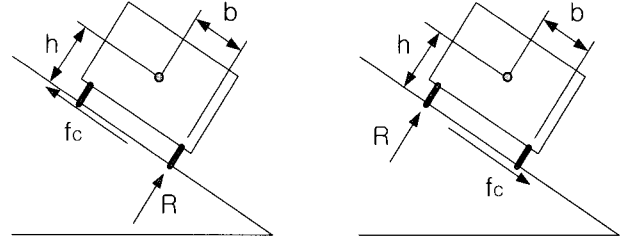


그림 11. 넘어지는 순간의 힘의 분포.  
Fig. 11. Force distribution at tumover.

(22)-(24)는 다음에 나오는 로봇 주행 속도의 제한 조건을 구할 때 사용될 것이다.

2. 미끄러짐에 대한 제한 조건

로봇의 트랙과 지면 사이에 작용하는 마찰력의 크기가 최대 정지 마찰력보다 커질 경우 로봇은 지면에서 미끄러져 경로를 이탈하게 될 것이다. 따라서 다음과 같은 제한 조건식이 성립한다.

$$|F|^2 = f_p^2 + f_c^2 \leq \mu^2 R^2 \tag{25}$$

여기서  $\mu$  는 정지마찰계수이다.

(25)에 (20)-(24)를 대입하여 정리하면 다음과 같이 가속도에 대한 제한 조건을 구할 수 있다.

$$a_{lower} \leq a \leq a_{upper} \tag{26}$$

$$a_{upper} = -gk_p - \sqrt{-\frac{1}{r^2}v^4 - \frac{2gk_c}{r}v^2 - g^2(k_c^2 - \mu^2k_n^2)} \tag{27}$$

$$a_{lower} = -gk_p + \sqrt{-\frac{1}{r^2}v^4 - \frac{2gk_c}{r}v^2 - g^2(k_c^2 - \mu^2k_n^2)} \tag{28}$$

여기에서  $r$ 은 로봇의 회전 반경을 나타낸다. 이때 가속도는 실수값을 갖기 때문에 근호 안의 식은 양수가 되어야 한다. 따라서 다음과 같은 속도에 대한 제한 조건을 구할 수 있다.

$$-\frac{1}{r^2}v^4 - \frac{2gk_c}{r}v^2 - g^2(k_c^2 - \mu^2k_n^2) \geq 0 \tag{29}$$

(29)를 만족시키는 속도  $v$  의 범위를 벗어나면 로봇은 지면에서 미끄러지게된다. 따라서 (29)는 미끄러짐에 대한 제한 조건이 된다.

3. 넘어짐에 대한 제한 조건

로봇의 속도와 가속도, 그리고 지형의 형태에 따라서 로봇은 넘어질 수 있다. 여기서는 미끄러짐에 대한 속도의 제한 조건과 독립적으로 로봇의 넘어짐에 대한 속도의 제한 조건을 구한다.

그림 11은 로봇이 넘어지는 순간의 힘의 분포를 나타낸 것이다.

그림 11에서 로봇의 무게 중심의 높이  $h$ 와 무게 중심과 트랙간의 거리  $b$ , 수직항력  $R$ 과 구심력 방향의 힘  $f_c$ 에 대한 모멘트를 분석하여 넘어지지 않는 제한 조건을 구하면 다음과 같다.

$$-R \frac{b}{h} \leq f_c \leq R \frac{b}{h} \quad (30)$$

그리고 제곱을 이용하여 다시 쓰면,

$$f_c^2 \leq R^2 \frac{b^2}{h^2} \quad (31)$$

가 된다. (31)에 (23), (24)를 대입하여 정리하면 다음과 같은 속도의 제한 조건을 구할 수 있다.

$$\frac{1}{r^2} v^4 + \frac{2gk_c}{r^2} v^2 + g^2 k_c^2 - g^2 k_n^2 \frac{b^2}{h^2} \leq 0 \quad (32)$$

(32)를 만족시키는 속도  $v$ 의 범위를 벗어나면 로봇은 넘어지게 되고, 따라서 (32)는 넘어짐에 대한 제한 조건이 된다.

4. 한계 속도 설정

로봇이 안정적으로 주행하기 위해서는 미끄러짐이나 넘어짐이 발생하지 않는 속도로 주행을 해야 된다. 그렇기 때문에 미끄러짐에 대한 한계 속도와 넘어짐에 대한 한계 속도를 비교하여 낮은 값을 로봇의 한계 속도로 설정하였다.

즉, 미끄러짐에 대한 한계 속도를  $u_{slip}$ , 넘어짐에 대한 한계 속도를  $u_{turnover}$  라고 하였을 때, 로봇의 한계 속도  $u$ 는,

$$u = \min(u_{slip}, u_{turnover}) \quad (33)$$

라고 설정하였다.

V. 주행 제어

Gyro/vision sensor module에 의해 감지된 지형과 실제 로봇의 위치 사이에는 거리 오차가 발생한다. 그 결과 로봇의 주행 한계 속도를 로봇에 바로 적용할 경우에는 거리 오차에 의한 미끄러짐이나 넘어짐이 발생할 가능성이 존재한다. 본 논문에서는 로봇의 주행 한계 속도를 미끄러짐이나 넘어짐이 발생하는 위치에 도달하였을 때 적용하지 않고, 로봇이 그 위치에 도달하기 이전에 미리 로봇에 적용하도록 하여 로봇 주행의 안정성을 확보하였다.

1. 한계 속도 적용

주행 로봇이 그림 12에서 보이는 것과 같이 로봇이주행할 경우 한계 속도 적용 방법은 다음과 같다.

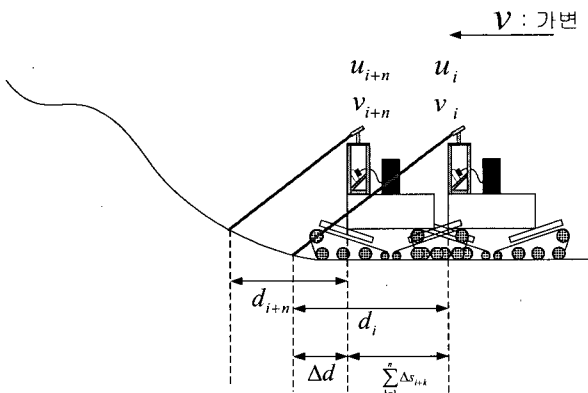


그림 12. ROBHAZ-DT의 자취.  
Fig. 12. Trace of ROBHAZ-DT.

먼저  $v_i$ 는 로봇의  $i$  번째 주행 속도,  $u_i$ 는  $i$  번째에서 구한 한계 속도,  $d_i$ 는  $i$  번째의 로봇과 감지된 지역간의 거리, 그리고,  $\Delta s_i$ 는  $i-1$  번째와  $i$  번째 사이에 로봇이 주행한 거리를 나타낸다.

로봇이  $i$  번째에서  $i+n$  번째까지 주행한 거리는  $\sum_{k=1}^n \Delta s_{i+k}$ 와 같이 표현할 수 있으며, 로봇이 감지한 지역과 로봇간의 거리  $d_i$ 와 로봇이 주행한 거리  $\sum_{k=1}^n \Delta s_{i+k}$ 와의 차이가 일정거리  $\Delta d$ 보다 짧아졌을 때 로봇의 주행 한계 속도를 적용한다. 즉,

$$d_i - \sum_{k=1}^n \Delta s_{i+k} \leq \Delta d \quad (34)$$

일 때,  $i$  번째에서 계산된 한계 속도  $u_i$ 를  $i+n$  번째의 위치에서 적용시킨다.

2. Velocity Cut-off

그림 13은 주행 한계 속도에 대해서 로봇에 전달되는 속도 명령을 제한할 때의 작업자의 속도 명령과 로봇의 주행 속도간의 관계를 나타낸 것이다. 작업자의 속도 명령  $v_{command}$ 가 한계 속도  $u$ 보다 작을 경우에는 로봇의 미끄러짐이나 넘어짐이 발생하지 않기 때문에 로봇의 속도 명령  $v_{robot}$ 은 작업자의 속도 명령  $v_{command}$ 과 같은 값으로 정의되지만,

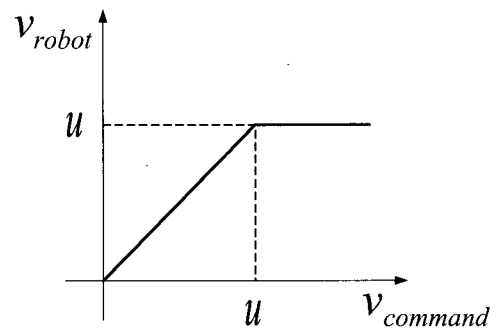


그림 13. 주행 한계 속도에 대한 속도 명령과 로봇 속도간의 관계.

Fig. 13. Velocity command and robot velocity for limit velocity.

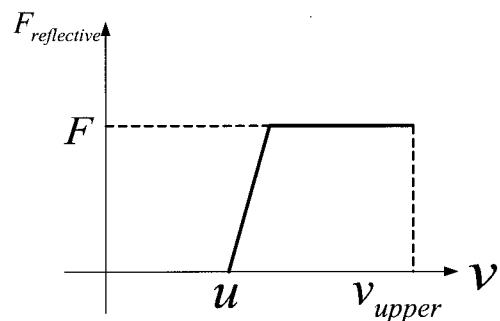


그림 14. 한계 속도에 대한 reflective force.

Fig. 14. Reflective force for limit velocity.

작업자의 속도 명령  $v_{command}$  가 주행 한계 속도  $u$  보다 크게 되면 미끄러짐이나 넘어짐이 발생할 가능성이 높기 때문에 로봇의 주행 속도  $v_{robot}$  은 주행 한계 속도  $u$  로 Cut-off 된다. 이 방법을 사용하면 작업자가 한계 속도 이상의 속도 명령을 내려도 주행 로봇이 한계 속도 이상의 속도로 주행하지 않기 때문에 미끄러짐이나 넘어짐을 회피할 수 있게 된다.

3. Reflective Force 생성

로봇의 주행 속도  $v_{robot}$  이 위와 같이 정의될 경우로봇의 주행 속도와 작업자의 제어 명령 사이에서는 불일치가 발생하게 된다. 따라서 위에서 구한 한계 속도를 로봇에 적용시킬 때, 작업자가 어느 시점에서 주행 한계 속도가 어떻게 로봇에 적용하였는지에 대한 정보를 알 수 있도록 힘 되먹임 조이스틱에(force feedback joystick)에 인공 반사 힘(artificial reflective force)을 생성하게 된다. 이와 같은 방법으로 주행 한계속도를 고려한 로봇의 주행속도와 작업자의 제어 명령 사이의 불일치를 보완할 수 있게 된다. 이때 reflective force는 DirectX 8.0 SDK에서 제공하는 condition force를 사용한다. 이 형태의 힘은 그림 14에서 보이는 것과 같이 조이스틱의 위치값에 따라 생성되는 힘이다. 이때  $F$ 는 인공 반사 힘의 크기를 나타내며  $u$  는 로봇의 주행 한계 속도로서 조이스틱의 Y축 좌표값에 대응되는 속도 명령이  $u$ 보다 클 경우  $u$ 를 넘지 못하도록 가상의 벽(virtual wall) 형태의 인공 반사 힘을 생성하게 된다. 따라서 조이스틱의 좌표값이 로봇의 주행 한계 속도  $u$  보다 작을 경우에는 작업자가 전혀 인공 반사 힘의 방해받지 않고 로봇의 주행 속도를 제어할 수 있게 된다. 이때 인공 반사 힘의 크기는 미끄러짐과 넘어짐 회피를 위한 주행 한계 속도를 나타내기 위한 목적에 따라 작업자에게 강하게 로봇의 상태를 인지 시키기 위해 조이스틱이 낼 수 있는 최대 값으로 설정하였다.

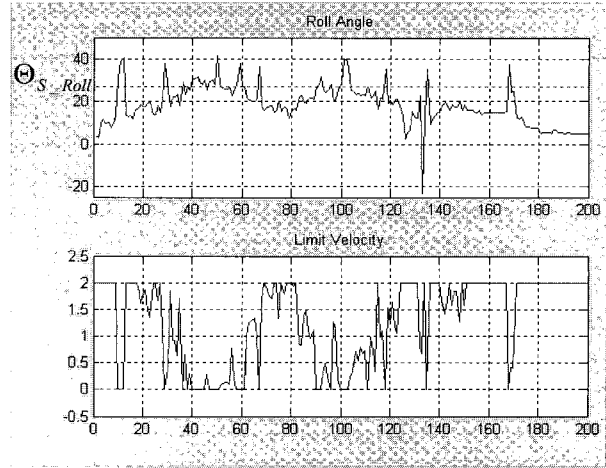
VI. 실험

미끄러짐에 대한 한계 속도는 로봇의 트랙의 재질과 지면의 종류에 따라 달라지는데 이는 둘 간의 정지 마찰 계수가 달라지기 때문이다. 그리고 넘어짐에 대한 한계 속도는 로봇의 무게 중심의 위치에 따라 달라지게 된다. 그래서 본 논문에서는 경사 지형으로 로봇이 접근할 때 정지 마찰 계수와 로봇의 무게 중심의 높이를 변화시키면서 지형 정보에 따른 로봇의 한계 속도를 계산한 후, 실제 로봇의 주행 속도를 측정하였다. 그리고 로봇의 기구학적 특성상 로봇이 앞뒤 방향보다는 좌우 방향으로 미끄러짐이나 넘어짐이 발생하기 때문에 본 실험에서는 로봇의 roll 각도 방향에 대해 분석하였다. 이때 로봇의 최대 속도는 2m/s로 정의하였다.

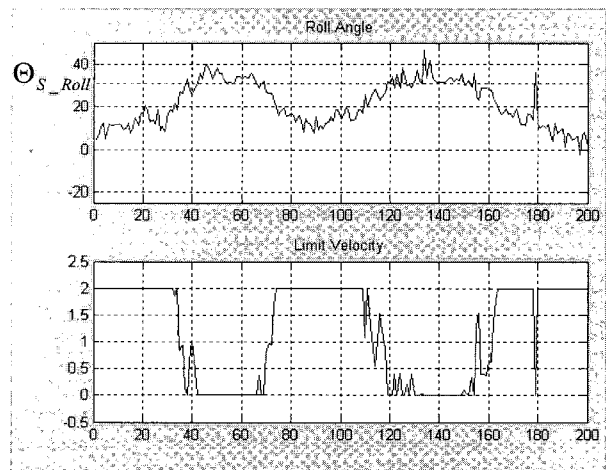
그림 15는 로봇과 지면의 정지 마찰 계수를 변화시키면서 측정한 roll각도와 로봇의 주행 속도(한계 속도)를 나타낸 그래프이다. 이때 무게 중심의 높이는 30cm, 무게 중심과 트랙 간의 거리는 22cm로 하였다.

그림 15에서 상단의 그래프는 로봇의 1m 전방 지형에서의 roll각도이고 하단의 그래프는 같은 시간의 로봇의 주행 한계 속도이다. 그리고 roll각도에서 점선은 미끄러짐이 발생하는 roll각도  $\Theta_{S\_Roll}$  이다. 로봇이 미끄러짐이 발생할 가능성이

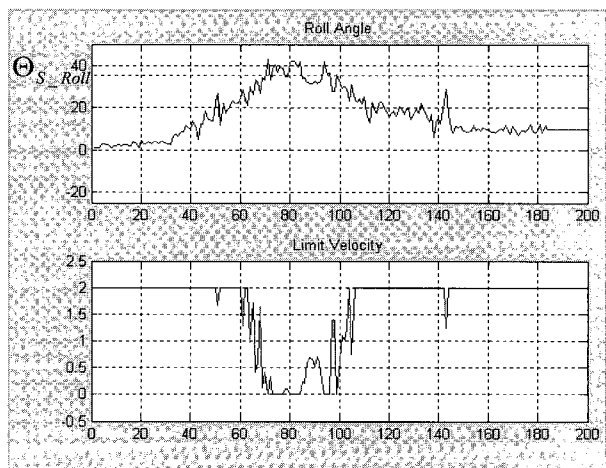
있는 지형에서는 한계 속도를 0으로 설정하여 로봇에 전달하는데, 그래프에서 roll각도가 점선에 가까워질수록 로봇의 주행 속도가 감소하며 점선을 넘을 경우에는 주행 속도가 0으로 제한되어 위험 지형으로 진행하는 것을 막는다.



(a) Coefficient of friction: 0.5

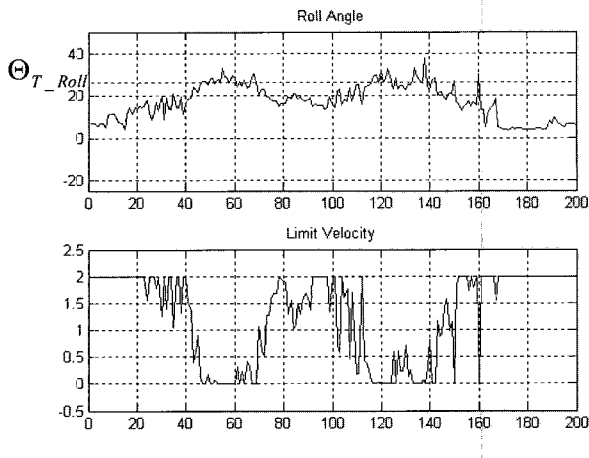


(b) Coefficient of friction: 0.6

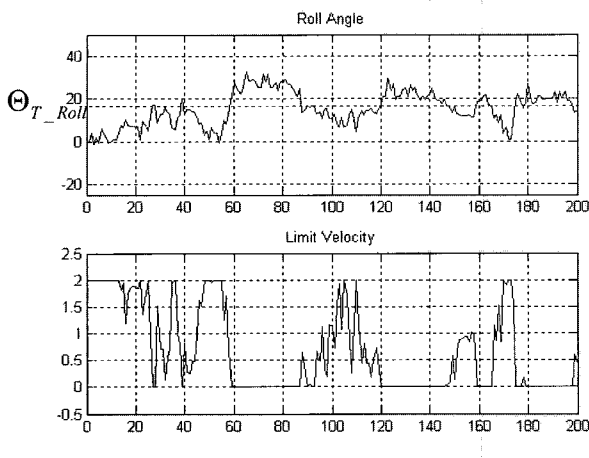


(c) Coefficient of friction: 0.7

그림 15. 정지 마찰 계수 변화에 따른 한계 속도.  
Fig. 15. Limit velocity for changing coefficients of friction.



(a) Height of center of mass: 45cm



(b) Height of center of mass: 75cm

그림 16. 무게 중심의 높이 변화에 따른 한계 속도.

Fig. 16. Limit velocity for changing height of center of mass.

그림 16은 정지 마찰 계수는 0.6으로 일정하게 두고 로봇의 무게 중심의 높이를 변화시키면서 측정된 roll각도와 로봇의 주행 속도를 나타낸 것이다. 여기서 점선은 전복이 발생하는 Roll 각도  $\Theta_{T\_Roll}$  이다. 무게 중심의 높이의 변화에 따른 한계 속도와 정지 마찰 계수의 변화에 따른 한계 속도가 유사한 결과를 낸다는 것을 알 수 있다. 결과적으로 적용되는 한계속도는 (33)과 같이 미끄러짐과 넘어짐에 의해 정의된 두 가지 한계속도 중 작은 값으로 결정 된다.

위 실험을 통해 주행 로봇이 험준한 지형을 주행할 경우 로봇의 전방 지형 기술기에 따른 roll 값을 예측하여 로봇이 미끄러짐이나 넘어짐이 발생하지 않게 하는 한계 속도를 구할 수 있었고, 이 값에 따라 로봇의 주행 속도를 제한함으로써 로봇이 미끄러짐이나 넘어짐이 발생하지 않고 안정적으로 주행할 수 있도록 제어 할 수 있었다.

## VI. 결론

본 논문에서는 주행 로봇의 전방 지형에 대한 정보와 로봇의 주행 상태 정보를 이용하여 경사 주행에 따른 로봇의 미끄러짐 및 넘어짐 회피 제어 기법을 제안하였다. 이때 로봇

의 전방 지형에 대한 정보 획득을 위하여 저가형 web camera 와 laser line generator 및 gyro sensor로 구성된 sensor module을 제안하였다. 제안된 sensor module을 이용하여 지형 정보에 따른 로봇의 roll, pitch 각을 중력을 기준으로 한 전역 좌표계에서 나타낼 수 있었다. 이와 같이 구해진 roll, pitch 각과 로봇의 주행 속도에 따른 로봇의 미끄러짐 및 전복 조건을 로봇의 운동 방정식에 의해 분석하고 그에 따른 로봇의 주행 한계 속도를 정의하였다. 이때 로봇의 운동 방정식은 몇 가지 가정을 통해 단순화 되었다. 이렇게 정의된 로봇의 주행 한계 속도는 로봇의 최대 주행 속도를 제한하고 작업자의 주행 제어 명령과 로봇의 주행 속도 사이의 불일치를 보완하기 위해 인공 반사 힘(artificial reflective force)를 생성하여 작업자에게 반영해 주는 방법을 제안하였다.

제안된 제어 기법 실험에서는 로봇의 미끄러짐에 대한 조건을 변화시키기 위해 정지 마찰 계수를 변화시키며 실험을 수행 하였고, 로봇의 넘어짐 조건을 변화시키기 위해 무게 중심의 높이를 변화시키면서 실험을 수행 하였다. 이때 기울기가 같은 지형에 대해 정지 마찰 계수가 낮을수록 로봇의 주행 한계 속도가 낮은 것을 확인할 수 있었고, 무게 중심의 위치가 높을수록 주행 한계 속도가 낮아지는 것을 그래프를 통해 확인할 수 있었다.

결과적으로 본 논문에서 제안된 제어 기법을 이용해 작업자가 경사가 있는 지형에서 원격으로 주행 로봇을 제어할 때 미끄러짐 및 넘어짐을 회피하여 안정적으로 로봇을 제어 할 수 있었다. 제안된 제어 기법의 적용 예로서 manipulator가 부착된 이동로봇의 경우, manipulator의 동작 상태에 따른 로봇의 무게 중심의 변화에 의한 미끄러짐 및 전복을 회피하여 주행 할 수 있게 된다.

## 참고문헌

- [1] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, September/October, 1989.
- [2] M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarin, P. Carinena, and S. Barro, "Fuzzy temporal rules for mobile robot guidance in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 31, no. 3, August 2001.
- [3] T. Fukuda, N. Kubota, "An intelligent robotic system based on a fuzzy approach," *Proceedings of the IEEE*, vol. 87, no. 9, September 1999.
- [4] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, June 1987.
- [5] J. H. Lim, D. W. Cho, "Physically based sensor modeling for a sonar map in a specular environment," *IEEE International Conference on Robotics and Automation*, May 1992.
- [6] S. Lacroix, G. Dudek, "On the identification of sonar features," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 586-592, Grenoble, France, 1997.
- [7] T. J. Greattinger and B. H. Krogh, "Evaluation and time scaling of trajectories for wheeled mobile robots," *ASME J. Dynamic Syst., Measurement, Control*, vol. 111, pp. 222-231, June 1989.
- [8] Z. Shiller, Y.-R. Gwo, "Dynamic motion planning of



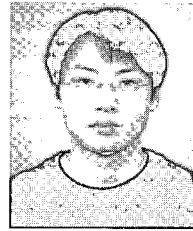
autonomous vehicles," *IEEE Transaction on Robotics and Automation*, vol. 7, no. 2, April 1991.

[9] 이정희, "Laser Beam을 이용한 주행 로봇의 안정적 주행 기법에 관한 연구," 서울대학교 공학 석사 학위 논문, 2004.



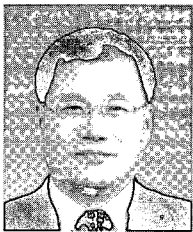
**이정희**

1976년 6월 19일생. 2002년 부산대학교 전자공학과(공학사). 2004년 서울대학교 전기·컴퓨터 공학부(공학석사). 2004년~현재 서울대학교 전기·컴퓨터 공학부 박사과정 재학중. 관심분야는 주행 로봇 제어.



**박재병**

1977년 9월 22일생. 2000년 성균관대학교 전기·전자 및 컴퓨터 공학부(공학사). 2002년 서울대학교 전기·컴퓨터 공학부(공학석사). 2002년~현재 서울대학교 전기·컴퓨터 공학부 박사과정 재학중. 관심분야는 주행 로봇 제어.



**이범희**

1955년 5월 29일생. 1978년 서울대학교 전자공학과 졸업. 1980년 서울대학교 대학원 전자공학과(공학석사). 1985년 Univ. of Michigan. Computer, Information & Control Eng(공학박사). 1985년~1987년 퍼듀대학교 교수. 1987년~1995년 서울대

학교 제어계측공학과 교수. 1996년~현재 서울대학교 전기공학부 교수. 관심분야는 로봇 제어.