

# 이동 및 센싱 불확실성을 고려한 로봇 시뮬레이터

## Robot Simulator Considering Uncertainties in Motion and Sensing

고낙용<sup>1</sup> · 서동진<sup>2</sup> · 김태균<sup>3</sup> · 배영철<sup>4</sup>

Nak Yong Ko, Dong Jin Seo, Tae Gyun Kim and Young-Chul Bae

<sup>1</sup>조선대학교 제어계측로봇공학과

E-mail: nyko@chosun.ac.kr

<sup>2</sup>(주)레드윈테크놀로지

E-mail: commidi@gmail.com

<sup>3</sup>조선대학교 대학원 제어계측공학과

E-mail: ktg9114@naver.com

<sup>4</sup>전남대학교 공과대학 전기·전자통신·컴퓨터공학부

E-mail: ycbae@chonnam.ac.kr

### 요 약

본 논문은 이동 로봇의 이동 및 거리 센싱의 불확실성을 고려한 시뮬레이터 개발에 대해 소개한다. 이동 로봇은 구동기, 바닥의 불안정성, 바퀴 및 구동 기구의 불확실성, 그리고 기타 구조적으로 어려운 다양한 원인으로 동작 명령과 차이가 있게 이동한다. 또한 이동 로봇에 장착된 각종 센서는 센서 자체의 불안정성, 주변 환경의 불안정성 등에 의하여 정확한 측정값을 출력하지 못한다. 이러한 이동 및 센서의 불안정성은 로봇의 자율 주행 알고리즘의 구현에 가장 큰 장애물이 되고 있다. 예측하기 어려운 불안정성을 고려하지 않는 알고리즘은 실제 환경에서 필연적으로 동작에 실패하여 크고 작은 사고를 일으킨다. 따라서 알고리즘의 검증에 시뮬레이터가 각종 불확실성을 포함하여 로봇 동작이 실제에 유사하도록 하여야 한다. 본 연구에서는 이동 로봇의 이동과 센싱에 불확실성을 포함하도록 시뮬레이터를 개발하였다. 다양한 센서들 중 이동 로봇의 위치 추정, 장애물 인식, 지도 작성 등에 가장 기본적으로 사용되는 영역 센서를 대상으로 불확실성을 구현하였다. 개발된 시뮬레이터를 사용하여 알고리즘을 검증하는 경우와 불확실성을 고려하지 않은 시뮬레이터를 사용하여 알고리즘을 검증하는 경우를 비교하여, 제안된 시뮬레이터의 성능을 검증하였다.

키워드 : 로봇 시뮬레이터(Robot Simulator), 이동(Motion), 센싱(Sensing), 불확실성(Uncertainty), 영역센서(Range Sensor)

## 1. 서 론

로봇시장은 산업용메니플레이터 중심에서 서비스로봇과 같은 지능로봇을 중심으로 급속히 확산되고 있다. 군사용 로봇<sup>[1]</sup>, 서비스로봇<sup>[2]</sup>, 산업용 지능로봇, 재활보조로봇<sup>[3]</sup>, 시설물 감시로봇<sup>[4]</sup> 등의 영역에 많은 로봇 연구자들이 포진되어 관련연구를 수행하고 있으며 일본 업체인 소니, 혼다, 도요다 등의 회사는 미래에 있을 로봇판매 시장을 위해 사람의 형태를 띄고 있는 휴머노이드 로봇을 속속 개발해 나가고 있다.

로봇을 개발하는데 있어서 가장 중요한 부분 중의 하나는 로봇의 오류 없는 정확한 동작을 만들어내는 것이다. 만일 로봇이 어떤 작업을 수행하고 있을 때 연산상의 오버플로우나 다른 문제점으로 인해 동작에 문제가 발생하게 된다면, 로봇이 설치되어 있는 작업환경과 로봇은 심각한 파손 등 비용 및 시간 등에 큰 손실이 생길 수 있다<sup>[8]</sup>. 이런 문제를 사전에 예방하기 위한 최선의 방법으로 로봇이 작업환경에 설치되기 전에 모든 동작에 대한 검증작업을 거쳐야 한다. 실제 로봇 하드웨어만을 가지고 이러한 동작검증 작업을 진행하기에는 많은 시간과 비용이 들게 되며, 이런 문제를 해결하기 위해서 실제 로봇과 똑같이 동작하는 시뮬레이션이 반드시 필요하다고 할 수 있다. 이런 필요성에 의해 현재까지 많은 종류의 로봇 시뮬레이션 프로그램이 개발되고 있으며, 각 프로그램들은 공간의 표현이나, 사용가능한 운영체제, 로

봇의 제어방식 등에 나름대로의 특징을 가지고 있다. 그러나 대부분의 로봇 시뮬레이터들이 모터의 동작이나, 센서의 부적절한 동작 등의 불확실성에 대해서 고려를 하고 있지 못하며 이점은 실제 로봇과 시뮬레이션에서의 실험에 많은 차이를 보여준다. 이런 차이점을 극복하고 더 강건한 알고리즘을 개발하기 위해서는, 시뮬레이션 상에서 로봇의 동작의 불확실성을 적용해야 하며 현실과 더 가까운 시뮬레이션 동작을 구현 할 수 있다.

본 논문에서는 로봇의 동작 불확실성과 거리영역센서의 불확실성이 적용된 사실감 있는 3차원 로봇 시뮬레이션의 제작에 그 목적을 두고 있으며, 가상로봇의 손쉬운 제어를 위해 윈도우, 리눅스, 유닉스, 맥킨토시등의 다중 플랫폼에서 사용가능한 TCP-IP기반 IPC라이브러리를 사용하였다. 각각의 로봇들은 설계시 확장이 용이하도록 모두 클래스로 제작되어 있으며, 모든 프로그램들은 서버-클라이언트구조를 가진다. 또한 로봇을 연구하는 사용자들이 시뮬레이션 프로그램을 사용에 용이하도록 클라이언트와 호스트 사이에 정보를 교환하는 부분을 라이브러리로 다시 재작성 하였다.

## 2. 시뮬레이션 프로그램의 형태

제작된 시뮬레이션 프로그램은 TCP IP기반의 서버 클라이언트 구조를 가지고 있으며, 다음의 여섯 가지 부분의 구현에 그 목표를 두고 있다.

- 1) 로봇에 장착된 영역센서의 불확실성 포함
- 2) 로봇 동작의 불확실성 포함
- 3) 유연한 제어를 위한 네트워크 리소스사용
- 4) 쉽게 확장 가능한 로봇 제어 구조
- 5) 새로운 로봇의 쉬운 추가 및 표현형식
- 6) 실제 로봇과 가상 로봇의 동일한 제어

1)을 제외한 모든 부분은 1차 버전이 이미 제작되어 연구에 사용되고 있으며<sup>[6]</sup> 관련된 부분에 대해서는 이해에 필요한 간단한 내용만을 담는다.

시뮬레이션 프로그램의 전체적인 구조는 그림 1과 같으며 전체적으로 다음 4가지 유닛으로 구성되어 있다.

- 1) 로봇제어유닛 - 모션에러, 센서에러 포함
- 2) 기구디자인유닛
- 3) 통신유닛
- 4) 실제 하드웨어 제어를 위한 드라이버 유닛

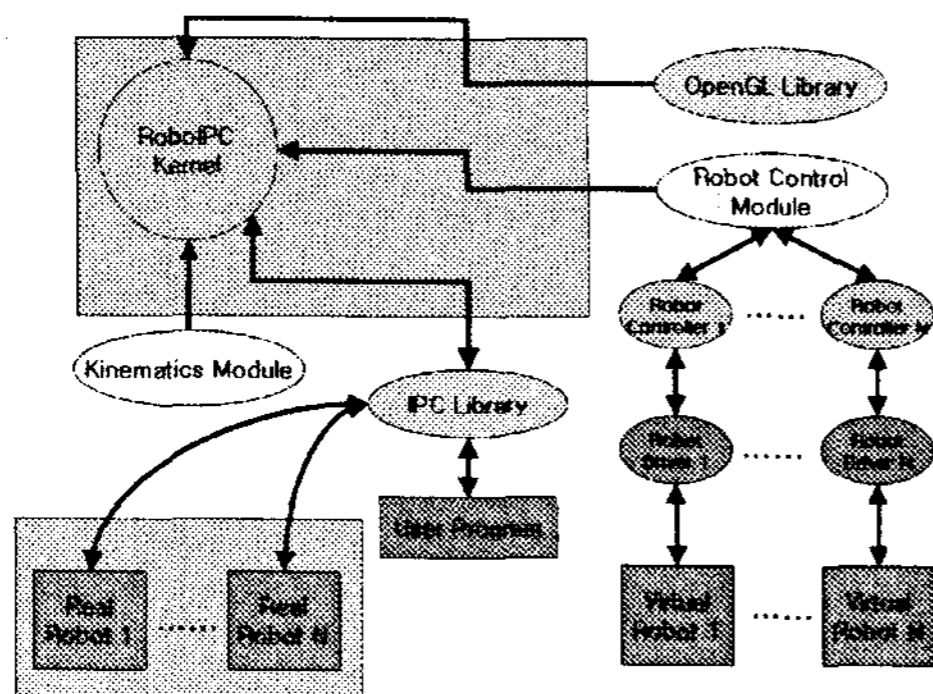


그림 1. 시뮬레이션 프로그램의 전체구조  
Fig 1. Block Diagram of Simulation Program

### 3. 시뮬레이션 프로그램의 구조

시뮬레이터는 크게 로봇제어유닛, 기구디자인유닛, 통신유닛, 실제로봇운동을 위한 드라이버 유닛으로 구성된다.

#### 3.1 로봇 제어 유닛

로봇제어유닛은 모터의 모션제어와, 센서, 맵관리, 로봇과 환경의 디스플레이 모듈로 이루어져 있으며 시뮬레이터 프로그램의 핵심을 이루는 부분이다.

로봇제어유닛내부의 모션제어모듈은 로봇의 동작불확실성이 구현되어 있다. 센서유닛에는 현재 초음파 영역센서, 레이저레인지파인더, 스테레오비전이 구현되어 있으며, 초음파영역센서와 레이저레인지파인더는 센서의 측정에 대한 불확실성이 포함되어 있다. 맵관리 모듈은 로봇의 작업환경의 정보를 담고있으며, 두 개의 포인트를 갖는 구조체 배열을 이용하여 직선으로 지도를 표현한다. 디스플레이 모듈은 사용자에게 가상의 작업공간과 로봇을 보여주는 모듈이며 OpenGL 라이브러리를 사용하여 로봇과 작업환경을 보다 현실감 있게 보여준다.

##### 1) 모터제어모듈

모터제어모듈은 로봇에 사용되는 기구부의 움직임을 위한 모듈로서 한 개의 로봇에 여러 개의 모터 모듈이 장착될 수 있다. 로봇에 하나의 모터가 추가될 때 모터의 동작을 위한 독립적인 쓰레드가 생성되게 된다. 각

객체의 시간에 따른 정확한 움직임을 위한 쓰레드 내에서의 실행시간을 측정하기 위해 MS윈도우즈 OS에서 시간측정을 위해 지원하는 QueryPerformanceFrequency 함수와 QueryPerformanceCounter 함수를 이용하여 타이머 클래스를 제작하였다. 모터제어모듈은 로봇의 불확실성이 포함되어 있으며[참고문헌] 식(1)에서와 같이 구해질 수 있다.

$$\begin{aligned} x_{i+1} &= x_i - \frac{\hat{t}v}{\hat{r}v} \sin\theta_i + \frac{\hat{t}v}{\hat{r}v} \sin(\theta_i + \hat{r}v \cdot \Delta t) \\ y_{i+1} &= y_i + \frac{\hat{t}v}{\hat{r}v} \cos\theta_i - \frac{\hat{t}v}{\hat{r}v} \cos(\theta_i + \hat{r}v \cdot \Delta t) \\ \theta_{i+1} &= \theta_i + \hat{r}v \cdot \Delta t + \hat{\gamma} \cdot \Delta t \end{aligned} \quad (1)$$

식(1)에서  $\hat{t}v, \hat{r}v, \hat{\gamma}$  는 식 (2)로부터 유도된다.

$$\begin{aligned} \hat{t}v &= tv + \epsilon_{\alpha_1} tv^2 + \epsilon_{\alpha_2} rv^2 \\ \hat{r}v &= rv + \epsilon_{\alpha_3} tv^2 + \epsilon_{\alpha_4} rv^2 \\ \hat{\gamma} &= \epsilon_{\alpha_5} tv^2 + \epsilon_{\alpha_6} rv^2 \end{aligned} \quad (2)$$

식(2)에서  $\epsilon_{b^2}$ 은 평균값이 0이고 분산이  $b^2$ 인 에러를 나타낸다. 파라미터  $a_1, \dots, a_6$ 은 불확실성에 대한 확률분포를 나타내며, 로봇 모션에 대한 정밀도를 모델링한다. 더 적은 해상도의 로봇모션을 모델링하기 위해서는 파라미터 값들에 더 큰 값들을 할당해 주면 된다. 그림 2와 3은 시뮬레이션 상에서 동작불확실성이 어떻게 로봇 동작에 영향을 미치고 있는 지를 보여준다.

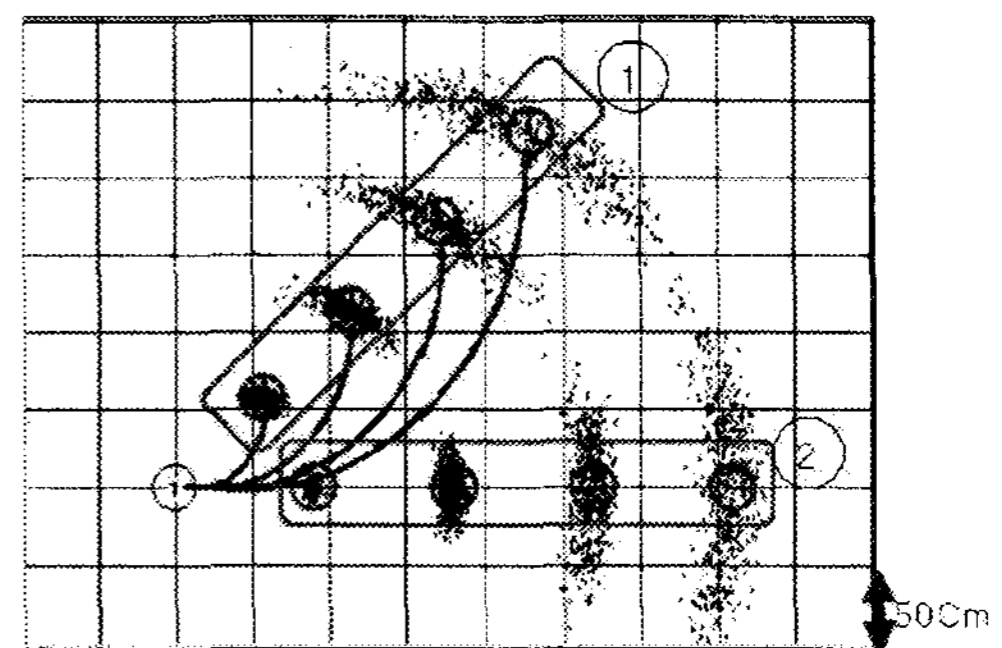


그림2.  $a_1=a_2=0.1, a_3=a_4=1, a_5=a_6=0.01$ 인 경우 불확실성  
Fig 2. Motion Uncertainty in case of  $a_1=a_2=0.1, a_3=a_4=1, a_5=a_6=0.01$

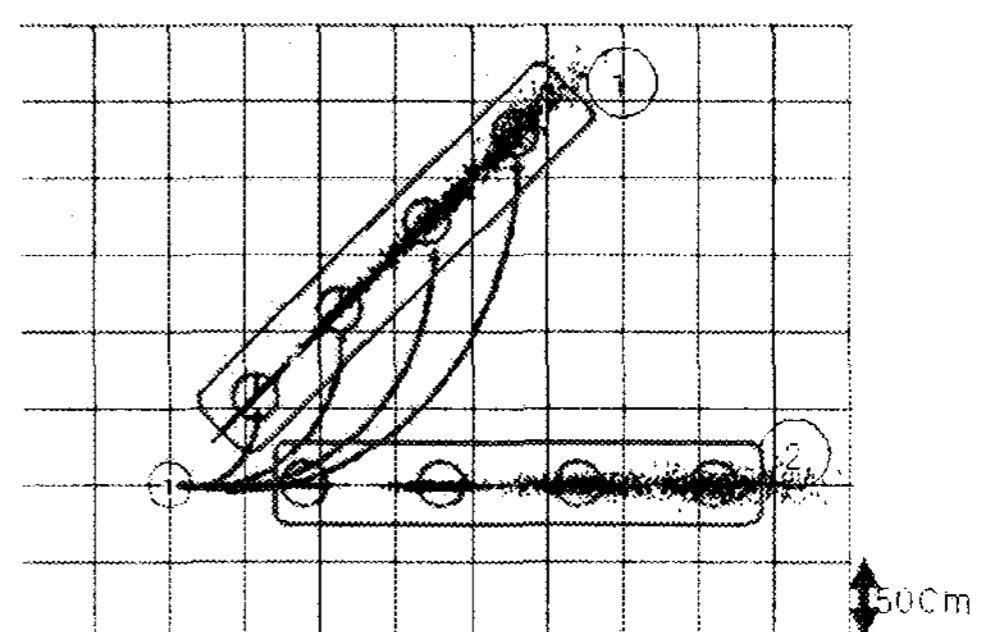


그림3.  $a_1=a_2=1, a_3=a_4=0.01, a_5=a_6=0.01$ 인 경우 불확실성  
Fig 3. Motion Uncertainty in case of  $a_1=a_2=1, a_3=a_4=0.01, a_5=a_6=0.01$

그림 2와 3에서 각 그림의 왼쪽 하단의 '1'로 표기가 되어 있는 원은 로봇의 출발위치이며, 로봇의 방향은 우

측을 보고 있다. 각각의 그림에서 각각의 점들은 로봇의 시작점에서 출발하여 3초 후에 로봇이 위치할 수 있는 가능성들을 보여주는 위치이다. 그림에서 네모로 둘러싸여 있는 2개의 그룹을 볼 수 있는데 대각선으로 놓여진 1번 그룹은 로봇에 회전속도  $rv$  를  $0.5326rad/sec$ 로 하고 직진속도  $tv$ 를 각각  $0.3m/sec$ ,  $0.6m/sec$ ,  $0.9m/sec$ ,  $1.2m/sec$  로 했을 때 로봇의 동작불확실성을 보여준다. 그룹2의 경우는 로봇이 회전속도 명령은 주지 않고 직진속도 명령만 그룹 1과 동일하게 주었을 때 로봇의 동작의 불확실성을 보여준다.

2) 센서모듈

센서모듈은 초음파, 레이저, 영상센서의 3가지가 구현되어 있으며 초음파와 레이저영역센서는 센서 측정의 불확실성이 구현되어 있는 상태이다. 센서측정의 불확실성을 구현하기 위해서 스텐포드대학의 세바스찬 Thurn 교수가 파티클 필터<sup>[7]</sup>를 위해 사용한 기법을 역으로 모델링하여 사용하였다.

이상적인 경우의 장애물과의 거리 값을  $Z_t^{k*}$  라고 정의한다. 이 값은 프로그램 상에서 "Ray casting" 기법에 의해 결정될 수 있으며 센서방향의 가장 가까운 물체를 검색해서 결정될 수 있다. 센서 시스템 모듈은 이렇게 획득된(Ray casting) 이상적인 값에 센서에러모델을 추가하여 불확실성이 포함된 값을 사용자에게 보내게 된다. 센서에러모델은 측정노이즈, 예측되지 않은 물체, 측정실패, 랜덤한 측정에러의 4가지의 경우로 나누어 생각될 수 있다. 이 중에서 예측되지 않은 물체에 대한 에러 모델은 FASTSLAM을 구현하는데 있어서 맵을 알고 있을 때 인간과 같은 장애물이 나타날 경우를 모델링한 것으로 실제 센서를 모델링하는데 있어서는 필요한 경우가 아니므로 사용하지 않았다.

측정노이즈는  $Z_t^{k*}$ 를 평균,  $\sigma_{hit}$ 의 표준편차를 갖는 식 (3)과 같이 가우시안 분포로 모델링 될 수 있다.  $\eta$ 는 정규분포  $N(z_t^k; z_t^{k*}, \sigma_t^k)$ 의 합이 항상 1이 되도록 해주는 변수이다.

$$p_{hit}(z_t^k|x_t, m) = \begin{cases} \eta N(z_t^k; z_t^{k*}, \sigma_t^k) & \text{if } 0 \leq z_t^k \leq z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

만일 센서의 신호가 어떤 알 수 없는 이유에 의해 측정되지 않았다면 센서의 값은 그 최대값으로 나타나게 된다. 예를 들자면, 초음파센서의 경우에는 물체에 반사되는 각도에 따라서 이런 경우가 많이 발생할 수 있으며, 레이저 센서의 경우에는 유리로 되어있는 벽을 감지할 경우 레이저가 투과하고 돌아오지 못하는 경우가 생길 수 있다. 랜덤한 에러는 센서의 거리 값의 모든 영역에 존재하는 것으로 가정한다.(식 4, 5)

$$p_{max}(z_t^k|x_t, m) = P(z = z_{max}) = \begin{cases} 1 & \text{if } z = z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$p_{rand}(z_t^k|x_t, m) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \leq z_t^k < z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

센서의 불확실성은 위의 모든 경우가 고려되어야 하며 각 3가지의 불확실성을 모두 합하여  $p(z_t^k|x_t, m)$ 를 구한다. 모든 확률의 합을 1로 만들어주기 위해  $z_{hit}$ ,  $z_{max}$ ,  $z_{rand}$ 를 사용한다.  $p(z_t^k|x_t, m)$ 의 합을 1로 만들어주기 위해서  $z_{hit}$ ,  $z_{max}$ ,  $z_{rand}$ 의 합이 1이 되어야 한다.(식6)

$$p(z_t^k|x_t, m) = \begin{pmatrix} z_{hit} \\ z_{max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_t^k|x_t, m) \\ p_{max}(z_t^k|x_t, m) \\ p_{rand}(z_t^k|x_t, m) \end{pmatrix} \quad (6)$$

구해진 확률변수  $p(z_t^k|x_t, m)$ 를 이용하여 에러가 포함된 센서 값을 얻기 위해, 거부법(Rejection Method)을 사용하여 구현한다.

3) 맵 관리 모듈

로봇의 작업환경을 담고있는 모듈로, 맵의 정보를 저장하기 위하여 두 개의 포인트를 갖는 구조체 배열을 이용하여 직선으로 지도를 표현한다. 센서모듈은 거리값을 출력하기 위하여 지도모듈에서 구성한 구조체 배열을 매개변수로 받아들이게 되며 이시기에 센서의 거리정보가 계산되게 된다. 이 작업은 시뮬레이션 환경 내에서 자동적으로 이루어지기 때문에 사용자 프로그램에서는 실제의 센서를 사용하는 것과 같은 효과를 느낄 수 있다.

4) 디스플레이 모듈

디스플레이 모듈은 컴퓨터 화면상에 실제 작업용 로봇과 기구학적인 구조가 같은 가상의 로봇을 사용자에게 보여주는 모듈이며, 각각의 센서모듈, 모터제어모듈, 맵 관리 모듈로 부터의 정보를 처리하여 프로그램상의 화면에 출력해주는 기능을 담당한다. 사용자에게 보여주기 위한 화면을 구성하기 위해서 3차원 공간상에서의 로봇의 세밀한 표현이 가능한 OpenGL 라이브러리를 사용하고 있다.

3.2 기구디자인 유닛

기구 디자인 유닛은 사용자가 로봇구조물의 D-H파라미터 정보를 담고있는 설정파일을 이용하여 암과 같은 관절형 로봇의 형태를 만들어 내는 모듈이며, 디스플레이 모듈과 함께 동작한다.

3.3 통신유닛

제작된 시뮬레이션 프로그램은 카네기 멜론 대학의 Reid Simmons가 제작한 IPC 라이브러리를 사용한다. IPC라이브러리는 복잡한 공학적인 시스템에서 이종 제어 프로세스간의 쉬운 통신을 위해 제작되었으며, 실시간 자율 시스템에서 필요한 충분한 기능과 유연성 그리고 강건성과 신뢰성을 제공하는 것에 디자인 규칙을 두고 있다.

4. 시뮬레이터의 적용

본 장에서는 시뮬레이터를 이용하여 새로이 제작된 불확실성이 포함된 센서데이터를 불확실성이 포함되어 있지 않은 경우와 비교하여 그 차이를 비교하고 고찰한다.

실험을 위해 그림 4와 같이 로봇을 P(0,0) 상태에 위치시키고 벽을 'L'형태로 배치시킨 후 레이저센서에서 얻어지는 값을 획득하였다. 경우 1은 레이저 센서값에 어떠한 불확실성도 포함시키지 않고 데이터를 획득하였으며 경우 2는 불확실성을 포함시킨 상태에서 데이터를 획득하여 경우 1과 2를 비교하였다. 불확실성을 포함시킨 상태에서의 데이터는 일정한 값이 나오지 않고 그 확률에 따라 항상 다르게 출력되기 때문에 한 개의 샘플만

을 가지고 비교하였다.

### 5. 결론

본 논문에서는 로봇의 동작 및 센싱에 있어서 불확실성을 적용한 로봇 시뮬레이션 프로그램의 개발에 대한 내용을 다루고 있으며 그 특징은 다음과 같다.

- (1) 실제 시스템을 구축하지 않고 미리 동작에 관한 실험을 한 뒤 알고리즘을 보완할 수 있으므로 실제 시스템 사용 시 발생할 수 있는 위험에 대비할 수 있으며, 로봇의 개발에 있어서 많은 시간과 비용을 단축 할 수 있다.
- (2) 로봇의 움직임에서 발생할 수 있는 동작과 센싱의 불확실성을 적용하여 실제동작 형태와 더 유사한 시뮬레이션의 실행이 가능하다.
- (3) TCP-IP 기반의 응용 어플리케이션인 IPC를 사용한 클라이언트, 서버기반의 시뮬레이션은 IT기술에 있어서 이슈중의 하나인 유비쿼터스 기술에 적절하게 대응할 수 있다.
- (4) 가상현실장치 개발에 사용되는 OpenGL 라이브러리를 사용하여 제작되어 2차원 시뮬레이터보다 더 현실감 있는 시뮬레이션이 가능하다.

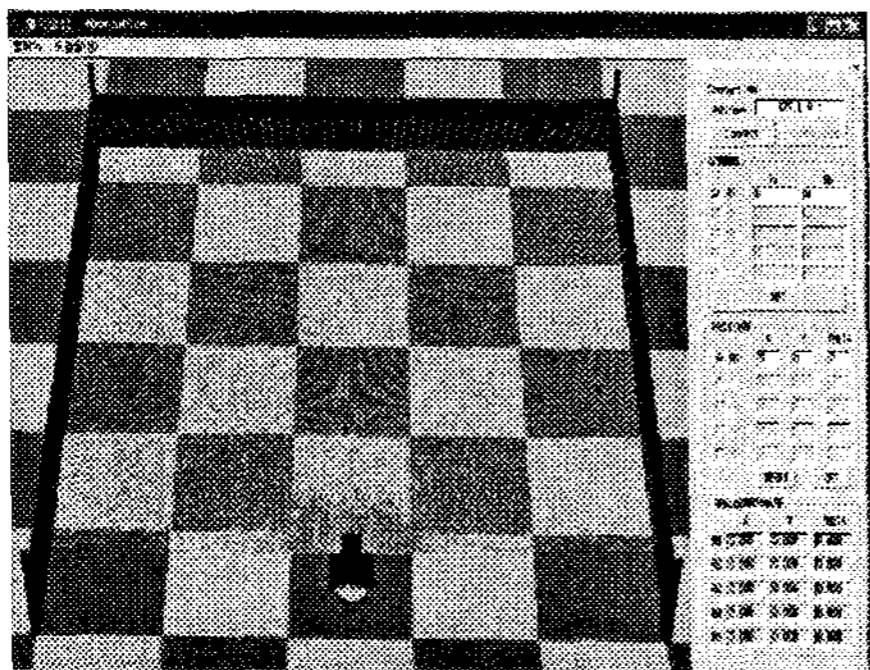


그림 4. 작성된 시뮬레이션 환경

Fig 4. Simulation Environment for Case 1, 2

그림 5는 불확실성이 포함되어 있지 않은 경우의 레이저 영역센서의 출력형태를 보여주며 그림 6은 불확실성이 포함되어 있는 경우의 출력형태를 보여준다. 그림 5에서는 어떠한 불확실성도 포함되어 있지 않으므로 지도의 데이터와 완전히 일치하는 'ㄷ'자를 그리고 있으며 그림 6에서는 경우 1과는 다르게 'ㄷ'형태의 패턴을 유지하고 있지만 그 불확실성 때문에 데이터가 약간씩 흩뿌려지는 형태를 볼 수 있으며 간간히 이상적인 값과 약간 동떨어진 값들이 출력되고 있음을 볼 수 있다.

실제로 하드웨어를 직접 제작하여 이와 같은 실험을 수행하는 경우 반사되는 장애물의 재질과 센서의 회로상태 등의 다른 여러 이유 때문에 경우 1과 같이 이상적인 값이 출력되는 경우는 없다고 볼 수 있으며, 경우 2와 같이 센서의 값이 흩뿌려지거나 초음파센서에서 예지나 코너 영역에서의 특징처럼 센서의 값이 예상과는 전혀 다른 형태로 나오는 경우가 많다. 이런 이유로 실제로 로봇 알고리즘을 개발하는데 있어서 경우 2와 같은 형태로 데이터가 출력되는 형태가 더 강건한 알고리즘을 만드는 데 더욱 효율적으로 사용될 수 있음을 예상할 수 있다.

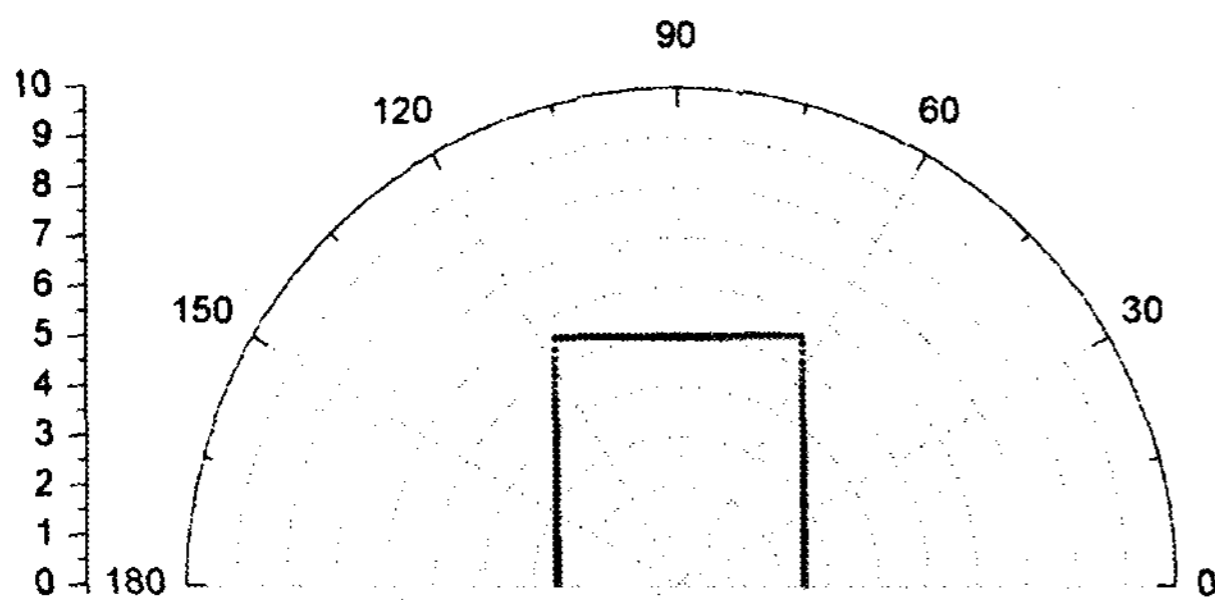


그림 5. 경우 1에서의 레이저영역센서 데이터  
Fig 5. Laser Range Sensor Data of Case 1

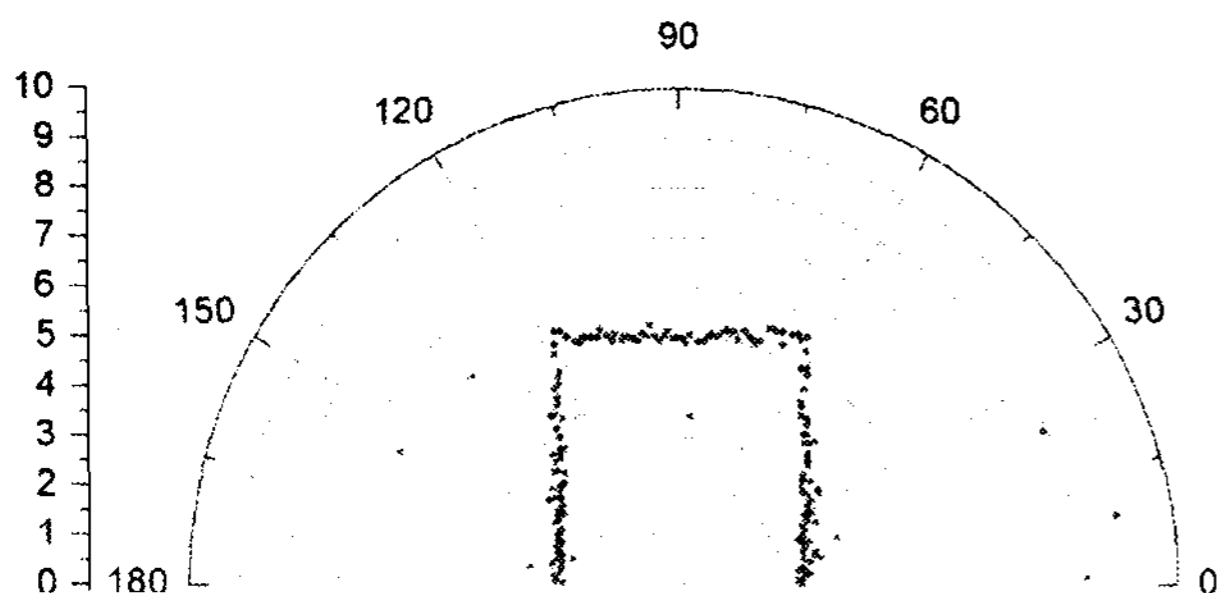


그림 6. 경우 2에서의 레이저영역센서 데이터  
Fig 6. Laser Range Sensor Data of Case 2

### 참 고 문 헌

- [1] Long, M. Gage, A. Murphy, R. Valavanis, K., "Application of the Distributed Field Robot Architecture to a Simulated Demining Task," *Robotics and Automation, ICRA 2005 Proc. of the 2005 IEEE International Conference on*, pp. 3193-3200, 2005.
- [2] Weckesser, P. and Dillmann, R., "Navigating a Mobile Service-Robot in a Natural Environment Using Sensor-Fusion Techniques," *Proc. IROS 02297 Grenoble*: 1423-1428, Sept. 1997.
- [3] Vorwerg, C. et. al., "PLATBOT: A Visually-Guided Robot for Physically Disabled Children," *Proceeding of Projective relations for 3D space: Computational model, application, and psychological evaluation*. 1997.
- [4] Stoeter, S. A. et. al., "A robot team for exploration and surveillance: Design and architecture," *The Sixth International Conference on Intelligent Autonomous Systems*: 767-774, Venice, Italy, July 2000.
- [5] Mckey, E. S., Drake, K. C., and Inigo, R. M., "Range Measurements by a Mobile Robot Using a navigation Line," *IEEE Transaction on PAMI* 8: 105-109, 1986.
- [6] Nak Yong Ko, Dong Jin Seo et al, "3-Dimensional Simulator for Multiple Robot System using TCP-IP," *The IASTED Conference on Software Engineering*, pp130-135, 2007
- [7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, 2005.