

# 이동로봇의 GPS위치 정보 보정을 위한 파티클 필터 방법

노성우\* · 김태균\*\* · 고낙용\*\*\* · 배영철\*\*\*\*

Particle filter for Correction of GPS location data of a mobile robot

Sung-Woo Noh\* · Tae-Gyun Kim\*\* · Nak-Yong Ko\*\*\* · Young-Chul Bae\*\*\*\*

## 요 약

본 논문은 실외환경에서 이동하는 자율주행로봇의 위치추정 문제를 다룬다. 위성 GPS정보와 IMU센서 정보를 보정하여 로봇의 위치를 확률적으로 추정하는 MCL방법을 제안한다. MCL 방법은 로봇의 위치 예측 과정과 센서 정보에 의해 예측된 위치를 보정하는 과정으로 구성된다. 위치 예측을 위해 필요한 모션모델은 이동로봇이 구동시의 직진 오차와 회전 오차를 포함한다. 보정은 신뢰도 값에 기반한 리샘플링에 의해 이루어진다. 신뢰도 값은 사용된 GPS와 IMU의 센서 모델에 의해 구해진다. 센서 모델을 구하기 위하여 GPS의 오차 범위를 반복 실험을 통해 구하였다. GPS는 로봇의 위치 추정을 위해 사용되며 IMU는 로봇의 이동 방향을 추정하기 위해 사용된다. 본 논문에서 제안한 방법을 실외환경에서의 이동로봇 위치 추정에 적용하였고, 실험결과를 분석하여 제안한 방법을 유효성을 보였다.

## ABSTRACT

This paper proposes a method which corrects location data of GPS for navigation of outdoor mobile robot. The method uses a Bayesian filter approach called the particle filter(PF). The method iterates two procedures: prediction and correction. The prediction procedure calculates robot location based on translational and rotational velocity data given by the robot command. It incorporates uncertainty into the predicted robot location by adding uncertainty to translational and rotational velocity command. Using the sensor characteristics of the GPS, the belief that a particle assumes true location of the robot is calculated. The resampling from the particles based on the belief constitutes the correction procedure. Since usual GPS data includes abrupt and random noise, the robot motion command based on the GPS data suffers from sudden and unexpected change, resulting in jerky robot motion. The PF reduces corruption on the GPS data and prevents unexpected location error. The proposed method is used for navigation of a mobile robot in the 2011 Robot Outdoor Navigation Competition, which was held at Gwangju on the 16-th August 2011. The method restricted the robot location error below 0.5m along the navigation of 300m length.

## 키워드

autonomous navigation, localization, IMU, GPS, MCL  
자율주행, 위치추정, 관성항법장치, GPS, MCL

\* 조선대학교 정보통신공학과(nswking0212@naver.com)

\*\* 조선대학교 제어계측공학과(ktg9114@naver.com)

\*\*\* 교신저자 : 조선대학교 제어계측로봇공학과(nyko@chosun.ac.kr)

\*\*\*\* 전남대학교 전기·전자통신·컴퓨터 공학부(ycbae@jnu.ac.kr)

접수일자 : 2012. 01. 16

심사(수정)일자 : 2012. 03. 19

게재 확정일자 : 2012. 04. 07

## I. 서론

무인 자율 주행을 실현하기 위해 개최된 그랜드 챌린지(Grand Challenge)대회는 세계최초의 무인자동차 장거리 주행 대회이다. 로봇은 스스로의 판단에 의하여 주어진 목적지까지 주행하여야한다[1]. 이러한 임무를 완수하기 위해 로봇은 자율주행이 이루어져야 한다. 로봇 자율 주행을 위해서는 다양한 요소기술들이 필요하다. 물체에 충돌하지 않고 이동할 수 있는 장애물 회피기술(obstacle avoidance), 원하는 목표점까지의 경로를 찾아내는 경로계획기술(path planning), 주변의 환경정보를 인식하여 지도를 작성하는 맵 빌딩(map building) 기술, 그리고 각종 센서 정보를 융합하여 지도상에서 자신의 위치를 알아내는 위치추정(localization)기술이 있다[2]. 특히 위치추정 기술은 자율 주행의 핵심기술로써 로봇이 특정장소에서 로봇자신의 위치(Pose)를 파악하는 것이다[3][4].

이동로봇이 실외환경에서 위치를 추정하기위해 일반적으로 이용되는 방법 중 하나는 위성 항법 장치인 GPS(Global Positioning System)정보를 이용하는 것이다[5]. 하지만 GPS는 로봇의 위치추정을 하는데 많은 문제들이 있다. 상용의 GPS정보는 기본적으로 수m의 오차를 가지고 있으며 순간적으로 신호를 수신하지 못하는 현상으로 인해 신호의 불확실성을 지니고 있다[6]. 이에 대한 대안으로 DGPS(Differential GPS)를 사용할 수 있다. DGPS는 위치 오차범위가 줄어드는 효과가 있다. 하지만 DGPS는 가격이 고가이고 DGPS 이용 중에는 데이터 비용이 발생하게 된다. 뿐만 아니라 정보수신시 GPS, DGPS 수신기 주변에 나무, 건물 등과 같은 장애물에 의해서 위성의 가려짐, 전파 반사 그리고 간섭 등의 문제로 인해 GPS정보를 제대로 수신할 수 없는 현상이 발생할 수 있다[9]. 그리고 GPS 신호의 갱신 주기가 0.5~1Hz이기 때문에 위성신호가 갱신되기전까지는 로봇의 위치를 알 수 없다.

이러한 GPS 사용의 단점들을 보완하기 위해 본 논문은 GPS 정보와 로봇 모션 정보를 확률적으로 융합하여 로봇의 위치를 추정할 수 있는 MCL(Monte Carlo Localization)[7][8] 방법을 제안한다. 제안된 방법의 실험 결과 GPS만을 이용하였을 때보다 더 높은 정확도로 로봇의 위치를 추정할 수 있음을 확인하였다. 그리고 이동로봇은 순간적으로 로봇의 위치가 변하지 않고 유연하게

이동하였다. 또한 순간적으로 GPS신호가 수신이 되지 못하는 경우나 GPS의 갱신이 이루어지지 않는 시점에서도 이동로봇은 MCL 알고리즘에 의해 위치를 추정할 수 있음을 보였다.

본 논문은 2장에서 GPS를 통해 얻어온 위도, 경도 좌표를 2차원상의 평면 좌표로 변환하는 방법에 대해서 설명한다. 3장에서는 GPS에서 추출한 위치정보와 관성 측정 장치인 IMU(Inertial Measurement Unit)에서 획득한 헤딩정보를 기반으로 MCL알고리즘과 융합하는 방법에 대해서 설명한다. 4장에서는 구현한 기술요소들을 통합하여 실제로로봇을 가지고 실험 및 고찰에 대해서 설명하고 끝으로 5장에서는 결과 및 향후 연구 진행 방향에 대해서 기술한다.

## II. GPS를 이용한 2차원 좌표

### 2.1 GPS처리 및 데이터

본 논문에서 사용한 GPS는 AKM-1M제품을 사용하였다. NMEA0183규격을 사용한 이 GPS는 그림 1과 같이 4가지 규격(\$GPGGA, \$GPGSV, \$GPRMC, \$GPGSA)을 1Hz마다 전송한다[10][11].

```
$GPGGA,2.15235,670,3735.0064,N,12701.6746,E,1.03,50.0,0.0,M,19.6,M,0.0,0.00004F
$GPGSA,A,2,10,08,02,,,,,,,,,50.0,50.0,20.0*0A
$GPGSV,3,1,10,26,78,314,00,29,71,017,00,24,48,156,00,10,44,079,46*79
$GPGSV,3,2,10,21,39,309,32,08,21,046,36,09,16,187,00,02,14,151,35*73
$GPGSV,3,3,10,06,14,236,00,15,06,319,00*7F
$GPRMC,2.15235,670,A,3735.0064,N,12701.6746,E,0.000000,060905,12
$GPGGA,2.15236,670,3735.0066,N,12701.6748,E,1.03,50.0,0.0,M,19.6,M,0.0,0.000040
$GPGSA,A,2,10,08,02,,,,,,,,,50.0,50.0,20.0*0A
```

그림 1. GPS NMEA코드  
Fig. 1 GPS NMEA code

\$GPGGA는 시간, 위치등의 정보데이터를 표현하고, \$GPGSV는 시야에 있는 위성의 개수, 위성 ID, SNR값 등의 정보 데이터를 표현하고, \$GPRMC는 시간 날짜, 위치, 경로, 속도 등의 정보 데이터를 표현하고 \$GPGSA는 GPS 수신모드, 사용되는 위성 표시, 고도 등의 정보 데이터를 표현한다. 4가지의 데이터형식 중에서 \$GPGGA코드만을 이용하여 시간, 위도, 경도를 이용하였다. 그림 2는 \$GPGGA규격의 샘플 데이터이다.

\$GPGGA,114455.532,3735.0073,N,12701.6446,E,1,03,7.9,48.8,M,19.6,M,0.0,0.0000+48

시간	위도	경도	계산종류	위상수	노이즈	고도	고도2	dppa	체크섬
시간	114455.532					그라나치 표준기 기준으로 11-7144를 95.512로			
위도	3735.0073,N				3735.0073, 북위				
경도	12701.6446,E				12701.6446, 동경				
	0				fix not available	위상신호를 잡지 못함			
계산	1				위상신호를 잡았으나	최소값 보고함			
종류	2				Differential GPS fix				
	3				RTCM-104에서 나온				
위상수	3				계산에 사용된 위상수				
노이즈	7.9				(DOP)Horizontal Dilution of precision 4.8, 수평 분산, 노이즈, 위상차지 정확성				
고도	48.8M				수평면기준으로 48.8M				
고도2	19.6M				가장해수면 기준으로 19.6M				
dppa	0.0,0.0000				dppa를 사용했을 경우 update한 시간과, dppa가 작을수록				
체크섬	+48				checksum				

그림 2. GPGGA 데이터 형식코드  
Fig. 2 GPGGA data format code

실제 로봇 네비게이션 용도로 적절한 데이터 형식 코드는 \$GPRMC이다. \$GPRMC는 추가적으로 진행 방향의 각도와 속도도 출력이 된다. 하지만 실제 측정을 해본 결과 로봇이 정지하거나 저속으로 움직일 경우 방향 오차가 높아서 이용하는데 많은 문제점이 있었다. 따라서 본 논문에서는 \$GPGGA를 이용하였다.

\$GPGGA에서 수집한 위도와 경도의 오차 정도를 알기위해 한 정점에서 5000개의 데이터를 수집하여 오차범위를 측정하였다. 그림 3은 5000개의 데이터를 순차적으로 받았을 때 나온 그림이다. 그림 3은 데이터가 1500개가 지난 시점에서 안정화되는 것을 알 수 있다. 따라서 1500데이터를 받고 나서 데이터를 측정하면 신뢰도가 높아진다.

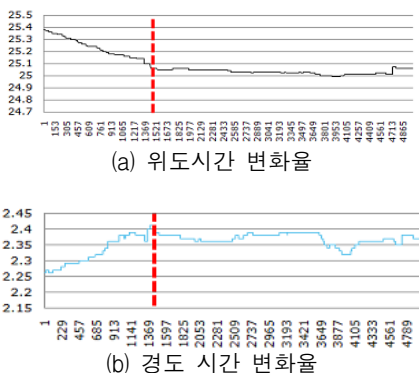
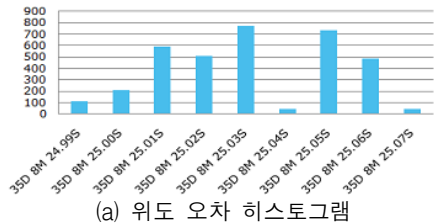
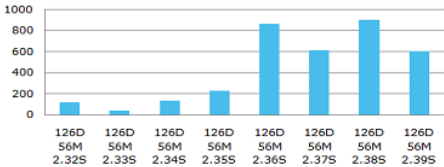


그림 3. 위도,경도 시간 변화율  
Fig. 3 Latitude, longitude time rate of change



(a) 위도 오차 히스토그램



(b)경도 오차 히스토그램

그림 4. 위도, 경도 거리 오차율  
Fig. 4 Latitude, longitude distance error rate

그림 4는 1500개의 데이터를 받은 시점에서부터 측정된 위도, 경도를 위치별로 누적한 그림을 나타내었다. 그림 4의 히스토그램의 간격은 0.01초이다. 거리로 환산하게 되면 약 30cm가 된다. 그림 4를 보면 위도는 2.4m, 경도는 2.1m의 오차율을 갖는다.

## 2.2 GPS와 로봇과의 매칭

GPS에서 획득한 위도, 경도는 구형체에서 출력된 좌표 값이다. 실제로봇에 적용시키기 위해 본 절에서는 2차원 범지구 측지좌표계(WGS84)방식[12]을 이용하여 2차원평면 위치 값으로 변환하였다. 본 논문에서는 좌표변환에 그림 5와 같이 UTM 좌표계를 사용하여 좌표를 변환하였다.

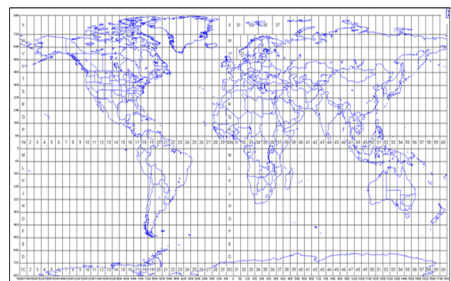


그림 5. UTM 좌표계  
Fig. 5 UTM coordinator

이 방법은 가우스 크뤼게 투영법을 이용한다 [13][14]. 가우스 크뤼게 투영법은 회전타원체에서 구체로 구체에서 평면으로 투영하는 가우스 상사이중투영법과는 달리 회전타원체에서 평면으로 직접 투영하는 투영법이다. 가우스 크뤼게 투영법에서  $x, y$  값은 다음 식(1),(2)와 같이 구해진다.

$$x = SF \left[ B + \frac{\lambda^2}{2} \cdot N \cdot \sin \phi \cos \phi + \frac{\lambda^4}{24} \cdot N \cdot \sin \phi \cos^3 \phi (5 - t^2 + 9\eta^2 + 4\eta^4) - \frac{\lambda^6}{720} \cdot N \cdot \sin \phi \cos^5 \phi (61 - 58t^2 + t^4 + 270\eta^2) \right] \quad (1)$$

$$y = SF \left[ \lambda \cdot N \cos \phi + \frac{\lambda^3 \cos^3 \phi}{6} \cdot N \cdot (1 - t^2 + \eta^2) + \frac{\lambda^5 \cdot \cos^5 \phi}{120} \cdot N \cdot (5 - 18t^2 + t^4 + 14\eta^2 - 58t^2 \eta^2) + \frac{\Delta \lambda^7 \cdot \cos^7 \phi}{5040} \cdot N \cdot (61 - 479t^2 + 179t^4 - t^6) \right] \quad (2)$$

여기에서,  $\lambda$ 는 중앙자오선과의 경도 차,  $\phi$ 는 적도와 위도 차, SF는 축척계수(0.9996),  $t$ 는  $\tan \phi$ ,  $a$ 는 지구타원체의 장반경, B는 적도로부터 원점까지의 자오선상 거리,  $\eta$ 는  $e' \cos \phi$ ,  $e$ 는 제1이심율(0.0816968312),  $e'$ 는 제2이심율 ( $\frac{e^2}{1-e^2}$ ),  $N$ 은 묘유선의 곡률반경 ( $\frac{a}{\sqrt{1-e^2}}$ )이다.

### III. GPS와 융합한 MCL 알고리즘

MCL 알고리즘은 기존 이동로봇의 위치추정 방법인 데드레크닝(Dead Reckoning), 삼각측량(Trialteration) 방법들의 문제점들을 확률론적 방법으로 해결 할 수 있는 장점을 지니고 있다. MCL 알고리즘은 이동로봇의 위치 정보를 파티클로 나타낸다. MCL 알고리즘은 전역 위치 추정이 가능하고, 확률이 높은 위치에 계산을 집중시킬 수 있는 장점을 지니고 있다.

MCL 알고리즘은 Sample Motion Model을 사용한 위치 예측과정, Measurement Model을 사용하는 신뢰도 계산과정, 그리고 예측된 위치를 보정하는 Resampling 3 단계로 구성된다[15]. 그림 6은 MCL 알고리즘을 나타낸

것이다. Line(3)의 Sample Motion Model은 이전 샘플링에서 생성된 각각의 파티클 위치로부터 기준으로 현재 속도명령인 직진속도와 회전속도를 사용하여 각각의 파티클 위치를 예측하는 과정이다.

Algorithm MCL( $X_{t-1}, u_t, z_t$ )

```
{
1:  $\bar{X}_t = X_t = \phi$ 
2: for  $m = 1$  to  $M$  do
3:    $x_t^m = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
4:    $\omega_t^{[m]} = \text{measurement\_code}(z_t, x_t^{[m]})$ 
5: endfor
6: for  $m = 1$  to  $M$  do
7:   draw  $i$  with probability  $\propto \omega_t^{[m]}$ 
8:   add  $x_t^{[i]}$  to  $X_t$ 
9: endfor
10: return  $X_t$ 
}
```

그림 6. MCL 알고리즘

Fig. 6 MCL algorithm

Line(4)에서 Measurement Model은 추정신뢰도를 계산하는 부분으로서 로봇의 현재 위치(pose)에서 수신된 GPS의  $x, y$  정보와 IMU의 방향정보를 바탕으로 파티클의 신뢰도를 계산한다. Line(6)~(9)은 파티클에 대한 신뢰도를 기반으로 추정 파티클들을 선별하는 과정이다. 이러한 과정을 매 샘플링 시각마다 수행하여 파티클들의 분포에 의해 로봇의 위치가 추정된다.

#### 3.1 MCL 알고리즘 모션 모델

모션 모델을 사용하여 이동로봇의 직진속도와 회전 속도 정보에 의해 파티클의 위치한다. MCL 알고리즘은 이동로봇이 위치할 확률을 파티클들을 통해 표현한다. 아래 그림 7에서 Sample Motion Model Velocity는 이전 파티클들이 현재 속도 정보에 대응하여 파티클들이 이동하는 모션모델 알고리즘으로 파라미터에 따라 파티클들에 불확실성을 추가하여 이동로봇에 대한 동작을 나타낸다. 즉, 모션 모델은 직진, 회전 속도 정보와 더불어  $\alpha_1 \sim \alpha_6$ 의 파라미터에 따라 파티클의 동작에 대해 수학적으로 표현하기 위한 모델이다. 그림 7의  $u_t$ 는 이동로봇의 속도정보,  $x_{t-1}$ 은 시각  $t-1$ 에서의 파티클의 위치( $x, y, \theta$ )를 나타낸

다. Sample Normal distribution 함수는 속도명령 파라미터인  $\alpha_1 \sim \alpha_6$ 에 대한 파티클 동작의 불확실성을 추가하는 알고리즘이다.

```

Sample_Motion_Model_Velocity( $u_t, x_{t-1}$ )
{
     $\hat{v} = v + \text{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$ 
     $\hat{\omega} = w + \text{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$ 
     $\hat{\gamma} = \text{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$ 
     $x' = x - \frac{\hat{v}}{\hat{\omega}} \sin \theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega} \Delta t)$ 
     $y' = y + \frac{\hat{v}}{\hat{\omega}} \cos \theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega} \Delta t)$ 
     $\theta' = \theta + \hat{\omega} \Delta t + \hat{\gamma} \Delta t$ 
    return  $x_t = (x', y', \theta')^T$ 
}

Sample_Normal_distribution( $b^2$ )
{
    return  $\frac{1}{2} \sum_{i=1}^{12} \text{random}(-b, b)$ 
}
    
```

그림 7. MCL 모션 모델  
Fig. 7 MCL motion model

### 3.2 MCL 알고리즘 센서 모델

GPS 센서모델은 Sample Motion Model에서 생성 하였던 각각의 샘플들에 대해 가중치를 부여한다. 본 연구에서는 2개의 분포도(Measurement Noise, Unexplainable Measurement)로 구성하였다.

Measurement Noise는 식(3)과 같이 가우시안 분포로 모델링 될 수 있다. 여기서  $Z_t^{kx*}$ 와  $Z_t^{ky*}$ 는 GPS센서로부터 수신된 위도, 경도 정보를  $X, Y$ 로 변환한 데이터이고  $Z_t^{kx}$ 와  $Z_t^{ky}$ 는 모션모델에 의해 예측된 파티클의  $X$ 좌표와  $Y$ 좌표이다.  $z_{x\_max}$ 와  $z_{y\_max}$ 는 GPS 최대  $X$ 좌표와  $Y$ 좌표를 의미한다.

$$\begin{aligned}
 p_{GPSX}(z_t^{kx} | x_t) &= \begin{cases} \eta N(z_t^{kx}; z_t^{kx*}, \sigma_x^2) & \text{if } 0 \leq z_t^{kx} \leq z_{x\_max} \\ 0 & \text{otherwise} \end{cases} \\
 N(z_t^{kx}; z_t^{kx*}, \sigma_x^2) &= \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(z_t^{kx} - z_t^{kx*})^2}{2\sigma_x^2}} \\
 p_{GPSY}(z_t^{ky} | y_t) &= \begin{cases} \eta N(z_t^{ky}; z_t^{ky*}, \sigma_y^2) & \text{if } 0 \leq z_t^{ky} \leq z_{y\_max} \\ 0 & \text{otherwise} \end{cases} \\
 N(z_t^{ky}; z_t^{ky*}, \sigma_y^2) &= \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(z_t^{ky} - z_t^{ky*})^2}{2\sigma_y^2}} \quad (3)
 \end{aligned}$$

Unexplainable Measurement의 경우는 GPS 센서의 신호가 어떤 알 수 없는 이유에 의해 측정되지 않거나 순간적으로 튀어 잘못된 값이 나올 경우의 확률분포를 나타낸 것으로 아래 식(4)와 같이 표현한다.

$$\begin{aligned}
 p_{GPSX}(z_t^{kx} | x_t) &= I(z_x = z_{x\_max}) \begin{cases} 1 & \text{if } z_x = z_{x\_max} \\ 0 & \text{otherwise} \end{cases} \\
 p_{GPSY}(z_t^{ky} | y_t) &= I(z_y = z_{y\_max}) \begin{cases} 1 & \text{if } z_y = z_{y\_max} \\ 0 & \text{otherwise} \end{cases} \quad (4)
 \end{aligned}$$

아래의 식(5)는 각각의 센서모델 알고리즘들을 확률분포도로 표현한 것으로 2개의 분포도를 나타낸다.

$$\begin{aligned}
 p(z_t^{kx} | x_t) &= \begin{pmatrix} z_{hitGPSX} \\ z_{randGPSX} \end{pmatrix}^T \begin{pmatrix} p_{hitGPSX}(z_t^{kx} | x_t) \\ p_{randGPSX}(z_t^{kx} | x_t) \end{pmatrix} \\
 p(z_t^{ky} | y_t) &= \begin{pmatrix} z_{hitGPSY} \\ z_{randGPSY} \end{pmatrix}^T \begin{pmatrix} p_{hitGPSY}(z_t^{ky} | y_t) \\ p_{randGPSY}(z_t^{ky} | y_t) \end{pmatrix} \quad (5)
 \end{aligned}$$

### 3.3 IMU 센서 모델링

IMU에서는 Unexpected Object, Sensing Failure모델링은 적용하지 않고 Measurement Noise, Unexplainable Measurement 만을 고려하였다.

식(6)은 Measurement Noise를 나타낸 것이다. 여기서  $Z_t^{k*}$ 는 IMU센서로부터 수신된 이동로봇의 heading 정보이다.  $Z_t^k$ 는 모션모델에 의해 예측된 파티클의 heading정보이고  $Z_\ominus$ 는  $0 \sim 2\pi$  라디안으로 나타낸다.

$$p_{IMU}(z_t^k | x_t) = \begin{cases} \eta N(z_t^k; z_t^{k*}, \sigma^2) & \text{if } 0 \leq z_t^k \leq z_\theta \\ 0 & \text{otherwise} \end{cases}$$

$$N(z_t^k; z_t^{k*}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2_{IMU}}} e^{-\frac{(z_t^k - z_t^{k*})^2}{2\sigma^2_{IMU}}} \quad (6)$$

Unexplainable Measurement의 경우는 IMU 센서의 신호가 순간적으로 튀어 잘못된 값이 나올 경우의 확률분포를 나타낸 것으로 아래 식(7)과 같이 표현한다.

$$p_{IMU}(z_t^k | x_t) = \begin{cases} \frac{1}{z_\theta} & \text{if } 0 \leq z_t^k \leq z_t^{k*} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

아래의 (8)의 식은 IMU센서모델을 하나의 확률분포도로 표현한 것이다.

$$p(z_t^k | x_t) = \begin{pmatrix} z_{hitIMU} \\ z_{randIMU} \end{pmatrix}^T \begin{pmatrix} P_{hitIMU}(z_t^k | x_t^k) \\ P_{randIMU}(z_t^k | x_t^k) \end{pmatrix} \quad (8)$$

그림 8은 GPS와 IMU센서 값을 확률변수로 하여 파티클의 신뢰도를 추출하는 알고리즘이다.

```

Belief_of_OneParticle_Acquisition_Model(z_t, x_t)
{
    q = 1
    conduct z_t^{GPS*} for measurement z_t^{GPS} using predicted particle pose X
    P_{GPS} = z_{hitGPS} * P_{hitGPS}(z_t^{kx} | x_t^k) + z_{randGPS} * P_{randGPS}(z_t^{kx} | x_t^k)
    conduct z_t^{GPSY*} for measurement z_t^{GPSY} using predicted particle pose Y
    P_{GPSY} = z_{hitGPSY} * P_{hitGPSY}(z_t^{ky} | y_t^k) + z_{randGPSY} * P_{randGPSY}(z_t^{ky} | y_t^k)
    conduct z_t^{IMU*} for measurement z_t^{IMU} using predicted particle pose Heading
    P_{IMU} = z_{hitIMU} * P_{hitIMU}(z_t^k | x_t^k) + z_{randIMU} * P_{randIMU}(z_t^k | x_t^k)
    q = P_{GPS} * P_{GPSY} * P_{IMU}
}

```

그림 8. MCL 센서 모델  
Fig. 8 MCL sensor model

## IV. 실험 및 고찰

### 4.1 시스템 구성

로봇 플랫폼으로는 레드윈 테크놀로지社의 4륜 구동방식인 NRLAB-04를 이용하였다[16]. 전방 장애물을 회피하기 위해 레이저 센서인 LRF(Laser Range Finder)를 장착하였으며, 위치 추정시 필요한 AKN-

1M모델 GPS수신기를 사용하였다. 자세 제어는 IMU, Electronic Compass를 이용하였다. 또한 사용자가 직접 집 로봇을 제어하기 위해 조이스틱을 이용하였다. 본 로봇의 시스템 구성은 그림 9와 같다.



그림 9. 로봇 시스템 구성  
Fig. 9 Robot system composition

각종 센서와 로봇에 구동명령을 송수신할 메인 컨트롤러는 Dell latitude 6300 PC를 사용하였다. 센서와 컨트롤러의 성능은 표 1과 같다.

표 1. 센서의 특성 표  
Table 1. Specification of sensor system

Sensor	Specification
NRLAB-04	Dimension : (L)600 * (W)438 * (h)310 mm Weight : 40kg Speed : (Max)1.0 m/s
NoteBook PC	Network Type : EtherCAT Transmission Rate : 100Mbps
LRF	Field of View : 180° Resolution : 1.0° Range : (Max) 80m
IMU	Angular Error : ±2.0deg Resolution : 0.01°/d/s
Electronic Compass	Accuracy : ±3.0deg Resolution : 0.1° d/s
GPS	Position Accuracy : 0~2.0m Regeneration interval : 1HZ

### 4.2 실험 환경

자율주행 실험은 광주 테크노파크 벤처지원 센터에서 실시하였다. 그림 10에 보면 출발점에서 도착점까지는 약250m이다.





그림 10. 자율주행 실험 장소

Fig. 10 Workspace for experiment of autonomous navigation

### 4.3 위치 추정 실험

이동로봇은 출발점 S에서 도착점 G까지를 테드레코닝, 단독 GPS, MCL방법을 적용하여 로봇의 위치 추정을 그림 11과 같이 나타내었다.

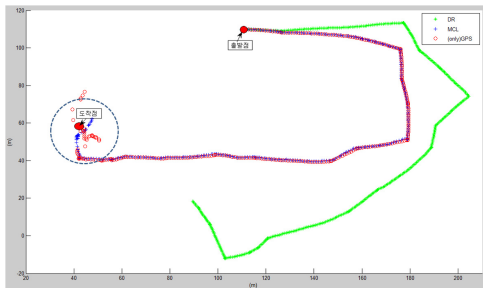


그림 11. 이동로봇 주행 경로

Fig. 11 Mobile robot navigation path

그림 11에서 보면 시작 지점에서는 테드레코닝, 단독 GPS, MCL방법 모두 같은 위치를 추정하면서 이동을 하였다. 하지만 테드레코닝 방법은 약20m 주행 때부터 실제로봇의 경로로부터 이탈하였다. 이는 테드레코닝의 단점으로 로봇의 드래프트현상과 한번 오차가 발생하면 오차의 보정 없이 누적되면서 이동하는 현상 때문이다. 단독 GPS, MCL을 적용시켰을 경우에는 서로 일치한 경로로 시작점에서 목표점까지 주행하는 그림처럼 보인다. 그림 11은 화면 뷰가 너무 작기 때문에 육안으로 확인하기가 쉽지 않다. 그림 12는 그림 11의 원호의 구역을 확대한 그림이다.

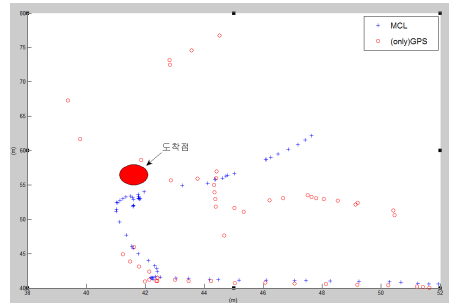


그림 12. 확대한 이동로봇 경로주행

Fig. 12 Zoomed in mobile robot navigation path

그림 12는 경로 상에서 단독 GPS만을 사용했을 경우와 MCL를 사용했을 경우를 비교한 그림이다. 위의 그림에서 확인한 결과 단독 GPS센서만을 가지고 로봇의 위치를 추정했을 때 보다 MCL방법을 적용하였을 때 순간적으로 위치 값이 변경되지 않고 오차의 범위가 줄어들고 안정적으로 위치추정이 되는 것을 알 수 있다.

## V. 결론

실외주행에서도 일반적인 GPS에 의한 위치추정은 정확성이 낮아서 로봇의 주행에 사용할 수 없다. 본 논문에서는 GPS와 IMU센서를 기반으로 MCL을 적용한 위치추정 방법을 제안하였다. 본 연구를 통해 도출된 결론은 다음과 같다.

- 1) MCL을 적용한 위치추정 방법은 단독 GPS만을 이용한 위치추정 방법보다 로봇 위치 정확도가 향상되었다.
- 2) 순간적으로 신호가 수신이 되지 않았을 때 GPS만을 사용하는 경우 위치를 인식할 수 없었다. 하지만 MCL을 방법을 이용하면 GPS가 신호를 수신하지 못하더라도 로봇의 위치를 추정할 수 있다.
- 3) 갱신이 되지 않는 시점에서 GPS는 이전 로봇의 위치를 추정 하고 있지만 MCL 적용 방법은 실시간으로 로봇의 위치를 추정하였다.
- 4) 불규칙한 잡음이 있는 GPS 출력 위치정보를 보정하여 실제 로봇의 경로에 가깝게 유연한 경로

가 추정되었다.

제안한 MCL방법을 사용하여 실외 자율주행대회에서 자율주행에 성공하였다. 향후 2개 이상 GPS를 추가하여 MCL의 센서모델링에서 위치의 신뢰도를 높여 위치추정의 성능을 향상시킬 예정이다.

### 감사의 글

본 연구는 교육과학기술부, 한국연구재단의 2010년 지역혁신인력양성사업(과제명 : 로봇의 자율주행 요소 기술 상용화 및 인력양성, 과제관리번호 : 2010-04-대-01-016)의 지원에 의해 이루어짐

### 참고 문헌

- [1] Defense Advanced Research Projects Agency (DARPA), DARPA GrandChallenge, Online source: <http://www.darpa.mil/grandchallenge>.
- [2] 윤강섭, "초음파위성 시스템을 위한 개선된 위치추정 알고리즘", 한국전자통신학회논문지, 6권, 5호, pp. 775-781, 2011.
- [3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, Probabilistic Robotics, The MIT Press, Cambridge, 2005.
- [4] S. Kim, C. Roh, S. Kang, and M. Park, "Outdoor navigation of a mobile robot using differential gps and curb detection," Proceedings of IEEE International Conference on Robotics and Automation, 2007.
- [5] 노성우, 김태균, 고낙용, "GPS센서와 MCL 알고리즘을 이용한 실외환경에서의 이동로봇 위치추정" 한국지능시스템학회 2011년도 추계학술대회 학술발표 논문집, 21권, 2호, pp. 49~51, 2011.
- [6] H. K. Lee, J. G. Lee, and G. I. Jee, "Channelwise multipath detection for general GPS receivers," Journal of Control, Automation, and Systems Engineering (in Korean), Vol. 8, No. 9, pp. 818-826, 2002.
- [7] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," Artificial Intelligence, Vol. 99, No. 1, pp. 21 - 71, 1998.
- [8] Borenstein, J., Everett, B., and Feng, L., "Where am I? Sensors and Methods for Mobile Robot Positioning," pp 130-131, 1996.

- [9] 국립지리원, "지형·지적정보의연계활용연구", 1998.
- [10] Braasch, M.S., Fink, AM, "Improved Modeling of GPS Selective Availability", Proceeding ION GPS National Technical Meeting, San Francisco, pp. 121-130, 1993.
- [11] Goodchild, M.F., Introduction to GIS, National Center for Geographic Information and Analysis, Univ. of California, 1991.
- [12] Liddle, D.A., "Orthometric height determination by GPS", Surveing and Mapping, Vol. 49, No. 1, pp. 5-16, 1989.
- [13] 김용일, 박경환, "우리나라 국가기본도의 편차도표에 관한 연구", 한국GIS학회, Vol. 3, No. 1, pp. 139-149, 1996.
- [14] V. Zavorotny, A. Voronovich, "Scattering of GPS signals from the ocean with wind remote sensing application, IEEE Trans Geosci. Remote Sens., Vol. 38, No. 2, pp. 951-964, March, 2000.
- [15] 김태균, 고낙용, 노성우, 이영필, "몬테 카를로 위치추정 알고리즘을 이용한 수중로봇의 위치추정", 한국전자통신학회논문지, 6권, 2호, pp. 288-295, 2011.
- [16] 문용선, 배영철, 노상현, 조광훈, 박용구, "이동로봇 모듈의 RTC 미들웨어 개발", 한국전자통신학회논문지, 5권, 2호, pp. 214-220, 2010.

### 저자 소개



#### 노성우(Sung-Woo Noh)

2007년 조선대학교 제어계측공학과 졸업(공학사)

2010년 조선대학교 대학원 제어계측공학과 졸업(공학석사)

2010년~현재 조선대학교 대학원 정보통신공학과 박사과정

※ 관심분야 : 이동로봇, 경로계획, Microprocessor





**김태균(Tae-Gyun Kim)**

2007년 조선대학교 제어계측공학과 졸업(공학사)

2009년 조선대학교 대학원 제어계측공학과 졸업(공학석사)

2009년~현재 조선대학교 대학원 제어계측학과 공학박사과정

※ 관심분야 : 이동로봇, 수중로봇, 위치추정



**고낙용(Nak-Yong Ko)**

1985년 서울대학교 제어계측공학과 졸업(공학사)

1987년 서울대학교 대학원 제어계측공학과 졸업(공학석사)

1993년 서울대학교 대학원 제어계측공학과 졸업(공학박사)

1997~1998, 2004~2005 미국 Carnegie Mellon Univ. Visiting research scientist

1992년~현재 조선대학교 제어계측로봇공학과 교수

※ 관심분야 : 지상로봇과 수중로봇의 자율주행



**배영철(Young-Chul Bae)**

1984년 2월 광운대학교 전기공학과(공학사)

1986년 2월 광운대학교대학원 전기공학과(공학석사)

1997년 2월 광운대학교대학원 전기공학과(공학박사)

1986년~1991년 한국전력공사

1991년~1997년 산업기술정보원 책임연구원

1997년~2006년 여수대학교전자통신전기공학부 부교수

2006년~현재 전남대학교 전기전자통신 컴퓨터공학부 교수

※ 관심분야 : Chaos Control and Chaos Robot, Robot control etc.