

# 몬테카를로 위치추정 알고리즘을 이용한 수중로봇의 위치추정

김태균\* · 고낙용\*\* · 노성우\*\*\* · 이영필\*\*\*\*

## Localization on an Underwater Robot Using Monte Carlo Localization Algorithm

Tae-Gyun Kim\* · Nak-Yong Ko\*\* · Sung-Woo Noh\*\*\* · Young-Pil Lee\*\*\*\*

### 요약

본 논문에서는 몬테 카를로 방법을 사용한 수중로봇의 위치추정 방법을 제안한다. 수중로봇의 위치추정은 자율 주행을 위한 기본 기능의 하나이다. 제안된 알고리즘에 의하면 추측항법(데드 레크닝 방법)의 약점인 위치 오차 누적 문제를 해결할 수 있다. 제안된 방법은 확률적인 방법으로 로봇 동작의 불확실성과 센서 정보의 불확실성을 처리한다. 특히 칼만 필터 방법과 달리, 로봇의 비선형 운동 특성과 센서의 비가우시안 출력 분포 특성을 모델링할 수 있다. 본 논문에서는 수중로봇 위치 추정에 몬테카를로 위치추정(Monte Carlo Localization : MCL, 이하 MCL로 표기함) 알고리즘을 적용하기 위하여 오일러각을 이용하여 모션모델을 구하였다. 또한 수중로봇에 모션모델과 센서모델을 적용하여 시뮬레이션을 구현하고, 이를 통해 수중로봇에 MCL 알고리즘의 적용 가능성을 보였다.

### ABSTRACT

The paper proposes a localization method of an underwater robot using Monte Carlo Localization(MCL) approach. Localization is one of the fundamental basics for autonomous navigation of an underwater robot. The proposed method resolves the problem of accumulation of position error which is fatal to dead reckoning method. It deals with uncertainty of the robot motion and uncertainty of sensor data in probabilistic approach. Especially, it can model the nonlinear motion transition and non Gaussian probabilistic sensor characteristics. In the paper, motion model is described using Euler angles to utilize the MCL algorithm for position estimation of an underwater robot. Motion model and sensor model are implemented and the performance of the proposed method is verified through simulation.

### 키워드

Underwater robot, localization, MCL algorithm, Euler angle, motion model, sensor model, autonomous navigation

## 1. 서론

수중로봇은 수중연구와 해양지도 작성, 기후 변화, 해양서식지 모니터링등과 같은 해양 생태계에 직면한

문제들을 접근할 수 있는 잠재성을 가지고 있다[1]. 수중로봇이 올바른 작업을 수행하기 위해서는 이동 중에 나타날 수 있는 장애물이나 기타 외부 환경에 대해 주어진 경로를 어느 정도의 위치에서, 어떤 방향

\* 조선대학교 제어계측공학과(ktg9114@naver.com)

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

\*\*\* 조선대학교 정보통신공학과(nswking0212@naver.com)

\*\*\*\* 레드원테크놀러지(주)(lyp@urc.kr)

접수일자 : 2011. 02. 09

심사(수정)일자 : 2011. 03. 10

게재확정일자 : 2011. 04. 12

으로, 얼마만큼 벗어난 상태인지 로봇 스스로가 판단하여, 보정할 수 있어야 한다[2]. 하지만 정확화 되지 않는 수중환경에서 실현가능한 위치추정과 내비게이션은 어려운 작업이라 할 수 있다[3].

본 논문에서 제안하는 MCL[4] 방법은 센서에 대한 불확실성, 오차 누적의 문제를 처리할 수 있다. 또한 로봇의 내부센서가 환경에 대한 상황을 감지하지 못하는 상태에서 로봇을 다른 장소로 이동시키는 로봇 유괴문제를 해결 할 수 있다. 이러한 장점을 지닌 MCL 알고리즘은 이동로봇 위치추정 분야에서 활발히 연구되어졌고 검증되어지고 있다. 본 논문은 이동로봇이 아닌 수중로봇 위치추정에 MCL 알고리즘을 적용하고, 시뮬레이션을 통해 위치추정 성능을 보인다.

## II. 수중로봇의 위치추정 알고리즘

본 논문에 이용된 위치추정 알고리즘은 확률에 기반한 MCL 알고리즘이다. MCL 알고리즘은 수중로봇이 위치할 확률을 파티클들로 나타내고, 확률이 높은 위치에 계산을 집중시킬 수 있으며, 구현이 간편하다는 장점을 가지고 있다.

그림 1은 MCL 알고리즘의 흐름을 나타낸 것으로, 수중로봇의 명령에 따른 파티클들의 위치를 예측하는 모션모델, 수신된 센서데이터를 기반으로 파티클들의 신뢰도를 획득하는 센서모델, 그리고 획득된 신뢰도에 의해 파티클을 선택하는 리샘플링 모델로 구성되어진다.

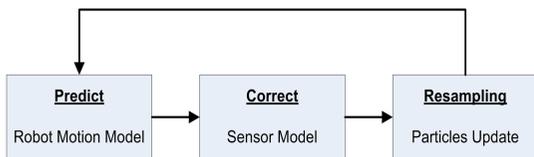


그림 1. 제안된 MCL 알고리즘의 구성  
Fig. 1 Composition of Proposed MCL Algorithm

### 2.1 MCL 알고리즘

본 논문에 제안된 MCL 알고리즘은 *Sample Motion Model*, *Measurement Model*, *Resampling* 3 단계로 구분한다. 표 1의 *Sample Motion Model*(3)은 이전 샘플링 타임에서 생성된 각각의 파티클 위치를

기준으로 현재 속도명령인 직진속도와 회전속도에 의해 각각의 파티클 위치를 예측하는 과정이다. *Measurement Model*(4)은 추정 신뢰도를 계산하는 부분으로 로봇의 현재 위치(*Pose*)에서 수신된 센서정보와 지도정보를 가지고 센서 정보의 불확실성에 대한 분포도를 만들어 실제 수신된 센서정보를 분포도에 대응시킨 후 하나의 파티클에 대한 신뢰도를 계산하게 된다. *Line*(6)~(9)은 파티클에 대한 신뢰도를 기반으로 추정 파티클들을 선별하는 과정이다. 이러한 과정을 통해 매 샘플링마다 파티클들의 분포에 의해 로봇의 위치가 추정된다.

표 1. MCL 알고리즘  
Table 1. MCL Algorithm

Algorithm UnderWater Robot MCL( $X_{t-1}, u, z_t, m$ )

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

## III. 수중로봇 모션모델

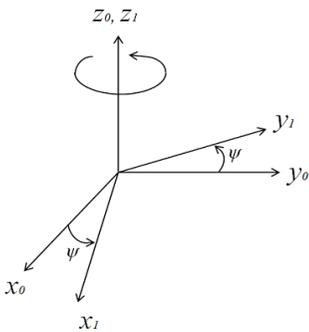
수중로봇에 대한 파티클들의 자세를 계산하기 위해, 본 논문에서는 관성항법 시스템에서 사용되는 오일러각(Euler angle)을 이용한다. 오일러각을 이용하는 방법은 로봇 좌표계에서 표현한 고정 좌표계와 로봇 좌표계 사이의 회전 각속도와 오일러각의 미분치 사이의 관계를 이용하여 좌표변환에 필요한 롤(roll,  $\phi$ ), 피치(pitch,  $\theta$ ) 그리고 요(yaw,  $\psi$ )를 계산하고, 이들 세 각을 이용하여 좌표변환 행렬을 구성한다. 오일러각은 물리적 의미가 확실하고 사용자가 직접적으로 자세를 바로 인식할 수 있다. 따라서 오일러각을 이용하여 계산을 수행하면 계산상의 의미를 쉽게 이해할

수 있고 다른 변환 과정 없이 출력을 직접적으로 얻을 수 있는 장점이 있다. 그러나 이 방식의 문제점은 피치 각이 90°인 경우에는 미분치를 구할 수 없는 문제(singularity problem)가 있다[5].

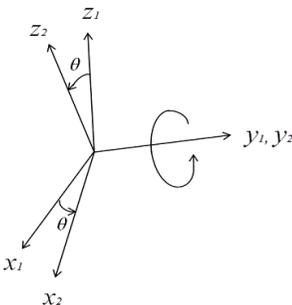
본 절에서는 오일러 각을 이용하여 수중로봇의 자세 및 위치와, 이를 이용한 MCL 알고리즘의 모션모델을 제안한다.

**3.1 오일러 각을 이용한 수중로봇의 자세와 위치**

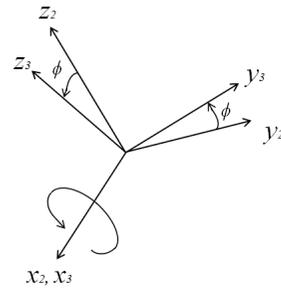
수중로봇의 좌표계를 지구의 고정좌표계로 표현하기 위해 사용되는 오일러각은 롤, 피치, 요로 구성된다. 먼저 기준 좌표계의 z축으로부터 요각을 그림 2(a)와 같이 회전하고, 회전된 y축에서의 피치각을 그림 2(b)와 같이 회전한다. 이 후 그림2(a)와 (b)에 의해 회전되어진 축을 기준으로 새로운 x축에서의 롤각을 그림 2(c)와 같이 회전시킨다.



(a) 요각에 대한 회전  
(a) Rotation on yaw angle



(b) 피치각에 대한 회전  
(b) Rotation on pitch angle



(c) 롤각에 대한 회전  
(c) Rotation on roll angle

그림 2. 오일러각의 정의  
Fig. 2 Definition of euler angle

그림 2와 같은 자세회전은 식 1과 같은 행렬식으로 정의할 수 있다.  $C_N^{F_1}(\psi)$ 는 요각에 대한 회전 행렬식,  $C_{F_1}^{F_2}(\theta)$ 는 피치각에 의한 회전 행렬식, 그리고  $C_{F_2}^B(\phi)$ 는 롤각에 대한 회전 행렬식이다.

여기서,  $N$ 은 고정 좌표계,  $B$ 는 로봇좌표계, 그리고  $F_n$ 은 행렬식에 의해 변환되어진 변환된 좌표계를 의미한다.

$$C_N^{F_1}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_{F_1}^{F_2}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{1}$$

$$C_{F_2}^B(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

이와같은 자세회전은 각각의 좌표 변환행렬의 곱을 통해 로봇좌표계에 대한 수중로봇의 자세를 계산할 수 있다. 따라서 고정좌표계에서 로봇좌표계로의 변환 행렬  $C_N^B$ 은 식 2와 같이 정의될 수 있다[6,7].

$$C_N^B = C_{F_2}^B(\phi)C_{F_1}^{F_2}(\theta)C_N^{F_1}(\psi) \tag{2}$$

고정좌표계에 대한 수중로봇의 좌표를 구하기 위하여, 식2에서의 로봇 좌표계에 대한  $(C_N^B)^T$ 로 변환하면 식 3과 같이 고정좌표계의 변환 행렬  $C_B^N$ 를 구할 수 있다. 여기서,  $c$ 와  $s$ 는  $\cos$ 과  $\sin$ 을 의미한다.

$$\begin{aligned}
 C_B^N &= (C_N^B)^T \\
 &= [C_{F_1}^B(\phi)C_{F_1}^{F_2}(\theta)C_N^{F_1}(\psi)]^T \\
 &= [(C_N^{F_1}(\psi))^T (C_{F_1}^{F_2}(\theta))^T (C_{F_1}^B(\phi))^T] \\
 &= \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}
 \end{aligned} \tag{3}$$

식 4는 수중로봇의 좌표계에서 발생하는 수중로봇의 속도정보를 이용하여, 고정좌표계에서의 수중로봇의 속도정보를 표현한다. 여기서,  $u, v, w$ 는 로봇 좌표계에서의 수중로봇의 속도이고,  $\dot{x}, \dot{y}, \dot{z}$ 은 고정좌표계에서의 속도정보를 의미한다.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = C_B^N \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4}$$

식 4의  $\dot{x}, \dot{y}, \dot{z}$  고정좌표계에 대한 속도정보로 다시 표현하면 식 5와 같다.

$$\begin{aligned}
 \dot{x} &= uc\theta c\psi + v(s\phi s\theta c\psi - c\phi s\psi) + w(c\phi s\theta c\psi + s\phi s\psi) \\
 \dot{y} &= uc\theta s\psi + v(s\phi s\theta s\psi + c\phi c\psi) + w(c\phi s\theta s\psi - s\phi c\psi) \\
 \dot{z} &= -us\theta + vs\phi c\theta + w c\phi c\theta
 \end{aligned} \tag{5}$$

식 6은 시간에 따른 각축의 각속도와 오일러각과의 관계를 나타낸다.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + C_{F_1}^B(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_{F_1}^B(\phi)C_{F_1}^{F_2}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{6}$$

식 6에 대하여 식 1을 적용하여 전개하면 식 7과 같이 표현할 수 있다.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

식 7은 오일러각의 변화량을 각속도로 표현한 식이다. 이를 롤, 피치 그리고 요각에 대한 식으로 나타내면 식 8이 된다.

$$\begin{aligned}
 \dot{\phi} &= p + q(\sin\phi \tan\theta) + r(\cos\phi \tan\theta) \\
 \dot{\theta} &= qc\cos\phi - rs\sin\phi \\
 \dot{\psi} &= q(\sin\phi \sec\theta) + r(\cos\phi \sec\theta)
 \end{aligned} \tag{8}$$

### 3.2 수중로봇의 모션 모델

본 논문에서 제안하는 수중로봇에 대한 모션 모델은 수중로봇의 속도 명령에 따른 파티클들의 위치를 예측하는 알고리즘으로 표 2와 같다.

표 2는 이전 파티클들이 현재 수중로봇의 속도 정보에 대응하여 파티클들이 이동하는 수중로봇의 모션 모델 알고리즘으로, 파라미터에 따라 파티클들에 불확실성을 추가하여 로봇에 대한 동작을 나타낸다. 즉, 모션모델은 수중로봇의 속도 정보( $u, v, w, p, q, r$ )와 더불어  $a_{uu} \sim a_{rr}$ 의 파라미터에 따라 파티클의 동작에 대해 수학적으로 표현하기 위한 모델이다. 표 2의  $u_t$ 는 수중로봇의 속도정보,  $x_{t-1}$ 은 시각  $t-1$ 에서의 파티클의 위치( $x, y, z, \phi, \theta, \psi$ )를 나타낸다. 여기서  $v=w=p=0$  으로 가정한다.

표 2. 수중로봇 모션모델  
Table 2. Underwater Robot Motion Model

SampleUWRobot Motion Model Velocity( $u_t, x_{t-1}$ )	
$\hat{u}$	$u + \text{sample}(\alpha_{uu} u  + \alpha_{uq} q  + \alpha_{ur} r )$
$\hat{q}$	$q + \text{sample}(\alpha_{qu} u  + \alpha_{qq} q  + \alpha_{qr} r )$
$\hat{r}$	$r + \text{sample}(\alpha_{ru} u  + \alpha_{rq} q  + \alpha_{rr} r )$
$\lambda_x$	$\text{sample}(\lambda_{xu} u  + \lambda_{xq} q  + \lambda_{xr} r )$
$\lambda_y$	$\text{sample}(\lambda_{yu} u  + \lambda_{yq} q  + \lambda_{yr} r )$
$\lambda_z$	$\text{sample}(\lambda_{zu} u  + \lambda_{zq} q  + \lambda_{zr} r )$
$\lambda_\phi$	$\text{sample}(\lambda_{\phi u} u  + \lambda_{\phi q} q  + \lambda_{\phi r} r )$
$\lambda_\theta$	$\text{sample}(\lambda_{\theta u} u  + \lambda_{\theta q} q  + \lambda_{\theta r} r )$
$\lambda_\psi$	$\text{sample}(\lambda_{\psi u} u  + \lambda_{\psi q} q  + \lambda_{\psi r} r )$
$x'$	$x + \dot{x}\Delta t + \lambda_x \Delta t$
$y'$	$y + \dot{y}\Delta t + \lambda_y \Delta t$
$z'$	$z + \dot{z}\Delta t + \lambda_z \Delta t$
$\phi'$	$\phi + \dot{\phi}\Delta t + \lambda_\phi \Delta t$
$\theta'$	$\theta + \dot{\theta}\Delta t + \lambda_\theta \Delta t$
$\psi'$	$\psi + \dot{\psi}\Delta t + \lambda_\psi \Delta t$
return $x_t$	$(x', y', z', \phi', \theta', \psi')^T$

서지(Surge)속도 파라미터인  $a_{uu}, a_{uq}, a_{ur}$ 는 서지방향에 대한 파티클의 분포를 결정한다.  $a_{uu}, a_{uq}, a_{ur}$ 의 파라미터 값이 크면 클수록 서지 방향으로 파티클들

이 넓게 분포되어지고, 피치 각속도 파라미터인  $\alpha_{ql}$ ,  $\alpha_{qr}$ ,  $\alpha_{qr}$  또한 파라미터 값이 클수록 피치 방향으로 파티클이 넓게 분포되어진다. 요 각속도 파라미터  $\alpha_{rl}$ ,  $\alpha_{rq}$ ,  $\alpha_{rr}$  역시 파라미터 값이 크면 클수록 요 방향으로 파티클들이 넓게 분포되어진다.

표 3의 *Sample Normal distribution* 함수는 속도 명령 파라미터인  $a_{lll} \sim a_{rr}$ ,  $\lambda_{lll} \sim \lambda_{rr}$  대한 파티클 동작의 불확실성을 추가하는 알고리즘이다.

표 3. 파티클 동작에 대한 불확실성 알고리즘  
Table 3. Uncertainty Algorithm on Moving Particle

*Sample Normal Distribution*( $b^2$ )

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

### V. 수중로봇 센서모델

센서모델은 각각의 파티클에 신뢰도를 부여하는 알고리즘이다. 모션모델에 의해 불확실성이 포함되어진 파티클들은 수신된 센서데이터 바탕으로 신뢰도를 부여 받게 되며, 4개의 분포도( *Measurement Noise*, *Unexpected Object*, *Sensing Failure*, *Unexplainable* )로 구성되어 있다.

*Measurement Noise*는 식 9와 같이 파티클과 센서 사이의 거리 및 각도를 중심으로 한 에러분포로 가우시안 확률분포로 표현된다.

$$p_{hit}(z_t^k | x_t, m) = \begin{cases} \eta N(z_t^k; z_t^{k*}, \sigma^2) & \text{if } 0 \leq z_t^k \leq z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$N(z_t^k; z_t^{k*}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma_{hit}^2}} e^{-\frac{(z_t^k - z_t^{k*})^2}{2\sigma_{hit}^2}}$$

*Unexpected Object*는 식 10과 같이 주어진 지도환경내에서 동적인 장애물이 존재할 수 있는 에러 분포도로 지수 확률분포로 표현할 수 있다.

$$p_{short}(z_t^k | x_t, m) = \begin{cases} \eta \lambda_{short} e^{-\lambda_{short} z_t^k} & \text{if } 0 \leq z_t^k \leq z_t^{k*} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

식 11은 *Sensing Failure*를 나타낸 것으로 센서데이터 감지 또는 동작하지 않을 때의 분포도 이다.

$$p_{max}(z_t^k | x_t, m) = I(z = z_{max}) \begin{cases} 1 & \text{if } z = z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

*Unexplainable Measurement*는 벽의 반사나 혼선으로 인해 센서의 값이 올바르게 못한 확률분포를 나타낸 것으로 식 12와 같다.

$$p_{rand}(z_t^k | x_t, m) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \leq z_t^k \leq z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

식 13과 같이 4개의 분포도를 가지고 하나의 확률 분포도로 센서모델의 분포도를 표현한다.

$$p(z_t^k | x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{pmatrix}^T \begin{pmatrix} p_{hit}(z_t^k | x_t, m) \\ p_{short}(z_t^k | x_t, m) \\ p_{max}(z_t^k | x_t, m) \\ p_{rand}(z_t^k | x_t, m) \end{pmatrix} \quad (13)$$

cf)  $z_{hit} + z_{short} + z_{max} + z_{rand} = 1$

표 4는 센서모델의 분포도들을 가지고 파티클에 대한 신뢰도를 추출하는 알고리즘을 표현한 것이다. 수중로봇에 수신되는 센서데이터를 확률 변수로 하여 파티클들과 지도정보를 바탕으로 파티클에 대한 신뢰도를 추출한다.

표 4. 각 파티클에 대한 신뢰도 획득 알고리즘  
Table 4. Belief Calculation Algorithm on respective Particle

```
=====
Weight_of_OneParticle_Calculation_Model( $z, x_t, m$ )
-----
q = 1
for k = 1 to K do
    complete  $z_t^k$  for the measurement  $z_t^{k*}$  using ray casting
     $p = z_{hit} \cdot p_{hit}(z_t^k | x_t, m)$ 
     $q = q \cdot p$ 
endfor
=====
```

### VI. 파티클 리샘플링 모델

파티클 리샘플링 모델은 센서신호에 근거해 이동로봇의 위치를 추정하는 과정으로, 파티클의 신뢰도에 따라 파티클을 복제하거나 제거하는 과정이다. 본 논문

문에 적용되어진 리샘플링 과정은 룰렛 방법[8]으로 표 5와 같이 표현된다.

표 5. 파티클 리샘플링 알고리즘  
Table 5. Particle Resampling Algorithm

---

*Roulet\_Method\_Particle\_Select(Pc,Pw)*

---

*Draw Roulet(Pw)*  
for 1 to Pc do  
    *Shoot an Arrow to the Roulet*  
    *Draw Selected Particle*  
endfor

---

### VII. 실험 및 고찰

본 논문에서 제안된 MCL알고리즘이 수중로봇에 적용 가능성을 확인하기 위하여, 시뮬레이션을 통해 검증하였다.

시뮬레이션에서 가정되어진 센서 데이터는 거리센서 2개, 깊이센서 그리고 요센서이다. 또한 시뮬레이션의 수중로봇에 주어진 명령은  $u = 0.5m/s$ ,  $q = r = 0.05rad/s$  이고  $v = w = p = 0$ 으로 제한하였다.

그림 3 ~ 5는 위와 같은 명령을 주었을 때 X, Y, Z축에 대한 위치추정 결과로서, Robot은 시뮬레이션 로봇을 나타내고, PT Avg는 파티클들 평균값을 의미한다.

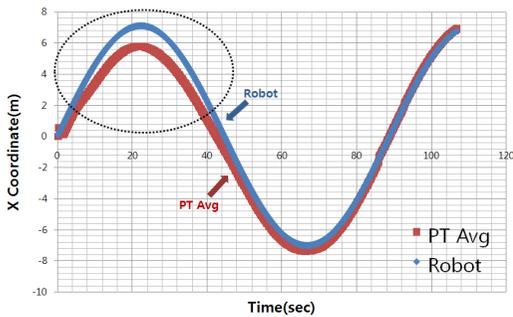


그림 3. X좌표에 대한 위치추정  
Fig. 3 Localization on X Coordination

위의 그림 3에서 점선으로 표시된 부분을 보면 초기에는 X좌표의 위치에 대하여 파티클들이 위치추정

을 정확히 못하지만 시간이 지남에 따라 정확히 위치추정을 하고 있다.

그림 4는 Y좌표에 대한 로봇의 위치추정 결과이다. 알고리즘 실행 초기에는 파티클들이 무작위로 퍼져 있는 결과로 평균을 취하기 때문에 점선으로 표시된 부분과 같이 보인다. 하지만 수신되는 센서데이터로 인하여 위치할 확률이 높은 파티클이 선택되어져 위치추정이 가능함을 알 수 있다.

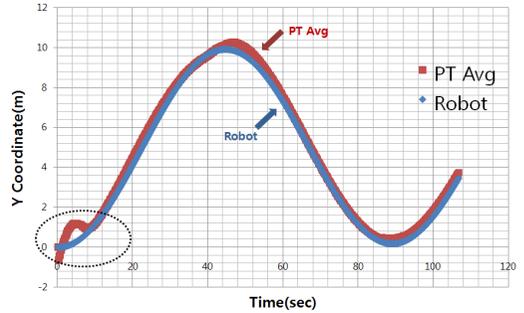


그림 4. Y좌표에 대한 위치추정  
Fig. 4 Localization on Y Coordination

그림 5는 Z좌표에 대한 수중로봇의 위치추정 결과이다. 점선으로 표시된 부분은 그림 14에서 설명되어진 것과 같이 알고리즘 실행 초기에 파티클들을 무작위로 퍼져 있는 결과를 가지고 평균을 취하기 때문이다.

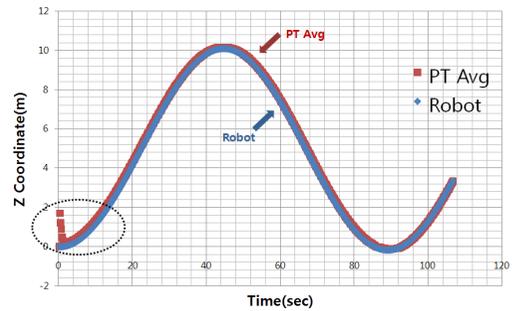


그림 5. Z좌표에 대한 위치추정  
Fig. 5 Localization on Z Coordination

### VIII. 결론

본 논문은 MCL알고리즘을 통하여 수중로봇의 위치추정 방법을 제안하였다. 본 논문에서는 수중로봇

좌표계를 지구고정 좌표계로 표현하는 방법을 보였다. 또한 MCL알고리즘을 수중로봇에 적용하기 위해 수중로봇 모션에러 모델을 제안하였고, 제안된 알고리즘을 시뮬레이션을 통해 검증했다.

제안된 위치추정 알고리즘은 수중로봇 뿐 아니라 다양한 로봇에 적용 가능하도록 알고리즘의 효율성과 신뢰성을 가질 필요가 있다. 이를 위해 다양한 센서들을 사용하여 알고리즘을 검증을 시도하고 있다. 앞으로 시뮬레이터로 검증된 위치추정 알고리즘을 실제 수중로봇에 적용할 계획이다.

**감사의 글**

본 연구는 교육과학기술부, 한국연구재단의 2010년 지역혁신인력양성사업(과제명: 로봇의 자율주행 요소 기술 상용화 및 인력양성, No. 100061)의 지원에 의해 이루어짐.

**참고 문헌**

[1] J. J. Leonard, A. A. Bennet, C. M. Smith, and H. J. S. Feder, "Autonomous underwater vehicle navigation", MIT Marine Robotics Laboratory, Technical Memo 98-1, 1998.

[2] Borenstein, J., Everett, B., and Feng, L., "Where am I? Sensors and Methods for Mobile Robot Positioning" pp.130-131, 1996.

[3] P. Corke, C. Detweiler, M. Hunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with Underwater Robot Localization and Tracking", In Proceedings of the IEEE 2007 Robotics and Automation, pp. 4556-4561, 2007.

[4] Sebastian Thrun, Wolfram Burgard and Dieter Fox, Probabilistic Robotics, MIT Press, 2005.

[5] 강철우, 유명민, 박찬국, "변형된 오일러각 기반의 칼만필터를 이용한 자세 추정 성능향상", 제어·로봇·시스템 학회, Vol. 14, No. 9, 2008.

[6] D. H. Titterton and J. L. Weston, Strapdown Inertial Navigation Technology Stevenage, U.K.: Peregrinus, 1997.

[7] H. Hong, J. G Lee, C. G. Park, and H. S. Han, "A leveling algorithm for an underwater vehicle using extended Kalman filter",

Proceedings of the IEEE 1998 Position Location and Navigation Symposium, Palm Springs, California, pp.280-285, 1998.

[8] 김태균, "유비쿼터스 센서 환경에서의 이동로봇 위치추정", 조선대학교 석사학위 논문, 2009.

**저자 소개**

**김태균(Tae-gyun Kim)**



2007년 조선대학교 제어계측공학과 졸업(공학사)

2009년 조선대학교 대학원 제어계측공학과 졸업(공학석사)

2009년~현재 조선대학교 대학원 제어계측공학과 공학 박사과정

※ 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

**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



**이영필(Young-pil Lee)**

2006년 2월 순천대학교 전자공학과  
(공학사)

2008년 2월 순천대학교 전자공학과  
(공학석사)

2008년~현재 : 레드윈테크놀로지(주) 연구원

※ 관심분야 : 로봇 제어, 모터 제어, 산업통신망