

Chirp Spread Spectrum 거리 측정을 이용한 이동 로봇의 위치 추정

Localization for Mobile Robot Based on Chirp Spread Spectrum Ranging

조 현 우, 이 영 훈, 김 상 우*
(Hyeonwoo Cho, Younghun Lee, and Sang Woo Kim)

Abstract: CSS (Chirp Spread Spectrum) specified in IEEE 802.15.4a can be used for ranging applications. In this paper, we apply the CSS to estimate the coordinates of a mobile robot. Four anchor nodes are installed at known positions and a tag node is attached to the target mobile robot. By CSS ranging, we measure the distances between each anchor and the tag node. Based on the measured distances, the coordinates of the mobile robot can be calculated by the method of trilateration. However the calculated coordinates are not accurate because of the errors of the measured distances. Therefore we propose an algorithm for reducing the effect of the errors. The proposed algorithm is executed with the extended Kalman filter. Through localization experiments, we show the performance of the proposed algorithm and the accuracy of the estimated position.

Keywords: chirp spread spectrum, localization, extended Kalman filter, trilateration, measurement error

I. 서론

유비쿼터스 환경을 구현함에 있어서 목표가 되는 대상의 위치 추정이 중요한 문제로 부각 되고 있다[1]. 특히 사용자에게 다양한 서비스를 제공하기위한 목적으로 사용되는 지능형 이동 로봇의 경우, 자율 주행을 가능케 하기 위하여 자신의 위치 정보를 획득하는 것이 매우 중요하다.

이러한 필요성에 따라 다양한 위치 추정 시스템 및 알고리즘들이 개발되어 왔다. 그 중 GPS (Global Positioning System) 시스템을 기반으로 한 연구가 있다[2,3]. 하지만 저가의 GPS 시스템의 경우 오차가 커 이동 로봇의 위치 추정에 적합하지 않으며, RTK (Real Time Kinematic)-GPS의 경우 정확한 위치 좌표를 얻을 수 있지만 고가의 장비이므로 비용에 민감한 이동 로봇에는 적용하는데 애로 사항이 있다[2]. 또 다른 문제점 중 하나로 GPS는 실내에서 활용할 수 없다는 점이다. 따라서 실내 환경에서도 운용이 가능한 위치추정 방법이 요구된다.

실내 이동 로봇의 위치 추정을 위한 시스템 중 초음파를 이용한 시스템이 있다[4,5]. 먼저 U-SAT 시스템은 GPS위성과 유사하게 실내에 설치된 초음파 위성이라는 개념을 이용하여, GPS위성이 RF신호를 사용하는 것과 달리 초음파 위성의 초음파 신호를 이용하고 이를 통해 각 위성과 대상 간의 거리를 측정한다. 이를 바탕으로 하여 이미 알려진 초음파 위성의 좌표와 함께 대상의 위치를 계산하는 방법을 이용 한다[4]. Cricket 시스템 또한 초음파를 이용하는 시스템으로써, 사전에 알려진 좌표 점들에 다수의 초음파 센서 노드를 배치하고, 이 노드들은 서로간의 신호 간섭을 피해

초음파를 방출한다. 이 초음파 신호를 수신한 대상은 센서 노드간의 거리를 측정하고, 이를 이용하여 자신의 위치를 계산하는 방식이다[5]. 이러한 초음파 기반의 위치 추정 시스템들은 위치 추정의 정확도가 높으며 실내 환경에서 사용 가능하다는 장점이 있다. 하지만 초음파 신호의 직진성에 의해 신호가 도달하지 못하는 음영지역이 존재하므로 초음파 센서를 배치하는데 환경적 요소를 고려한 고찰이 필요하다. 또한 초음파 신호는 전달 속도가 RF신호에 비해 매우 느리기 때문에 하나의 거리 값을 획득하는데 걸리는 시간, 즉 샘플링 주기가 길다. 이는 이동 로봇의 위치가 빠르게 변할 경우 위치 추정의 정확성을 떨어뜨리는 단점으로 작용할 수 있다. 따라서 거리 값 샘플링 주기가 빠르며 음영지역 발생 문제를 해결할 수 있는 위치 추정 기법이 요구된다.

본 논문에서는 이러한 문제점을 해결하기 위하여 CSS (Chirp Spread Spectrum)를 이용한 위치 추정 시스템을 적용 하였다. CSS는 IEEE 802.15.4a에 정의된 근거리 무선 통신 표준으로써, RF신호의 TOF (Time-of-Flight)를 이용하여 거리 측정(ranging)이 가능하다는 특징이 있다[6]. CSS는 RF신호를 이용하므로 전 방향 안테나를 사용할 경우 음영지역을 해소할 수 있다. 또한 초음파에 비해 RF신호의 속도는 매우 빠르므로 거리 측정값을 획득하는데 걸리는 시간이 초음파보다 짧다. 따라서 CSS는 초음파 시스템의 단점을 극복할 수 있을 것이라 기대된다.

CSS기반의 위치 추정 시스템을 이동 로봇에 적용하기에 앞서 CSS 시스템으로부터 측정된 거리의 특징을 분석한 후 거리 측정 오차를 감소시키는 방법을 제안한다. 또한 EKF (Extended Kalman Filter)를 기반으로 한 이동 로봇의 위치 추정 방법을 소개한다. 마지막으로 이동 로봇의 위치 추정 실험을 통해 얻은 결과를 바탕으로 CSS 위치 추정 시스템의 정확도 및 실제 적용 가능성에 대해 평가하고자 한다.

* 책임저자(Corresponding Author)

논문접수: 2009. 6. 18., 수정: 2009. 7. 2., 채택확정: 2009. 7. 20.

조현우, 이영훈, 김상우: 포항공과대학교 전자전기공학과

(lighto@postech.ac.kr/leeyounghun@postech.ac.kr/swkim@postech.ac.kr)

※ 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008-313-D00350).

II. CSS 거리 측정 및 삼각 측량

1. CSS기반 거리 측정의 개요

근거리 무선 통신 표준인 IEEE 802.15.4a의 물리 계층에는 IR-UWB (Impulse Radio Ultra Wide Band) 및 CSS가 정의되어 있다[7]. IR-UWB와 CSS는 모두 RF의 TOF를 기반으로 거리를 측정할 수 있다. 이 중 본 연구에서는 초점을 맞춘 CSS는 chirp 신호를 사용하기 때문에 비교적 다중경로에 강인한 특성을 가지고 있으며 그림 1에 나타난 바와 같이 SDS-TWR (Symmetric Double Sided Two Way Ranging)을 이용하여 거리측정을 수행한다[6,7].

그림 1에서 볼 수 있듯이 거리를 측정하고자 하는 두 CSS노드 A, B가 각각 신호를 보낸 후 응답확인신호(acknowledgment)를 받을 때 까지 걸리는 시간을 측정하는 TWR (Two Way Ranging)을 서로 대칭이 되도록 수행한다. 두 번의 TWR에 의해 얻어진 시간을 통해 최종적으로 결정되는 두 노드 간 신호전달시간 T_d 는 식 (1)과 같이 계산된다[8].

$$T_d = \frac{1}{4}(T_1 - T_{pA} + T_2 - T_{pB}) \quad (1)$$

여기서 T_1, T_2 는 각각 노드 A, B에서 측정된 신호 round time이며, T_{pA}, T_{pB} 는 각각 노드 A, B에서 응답확인신호를 보낼 때까지 소요된 처리 시간이다. 이러한 SDS-TWR 방식은 TWR을 기반으로 하므로 두 노드간의 시간 동기화 없이 거리를 측정하는 것이 가능하며, SDS에 의해 두 노드간의 클락 드리프트(clock drift)에 의한 오차를 줄일 수 있다[8]. 본 논문에서 다루는 이동 로봇의 위치 추정에는 이러한 CSS에 의한 거리 측정값을 기반으로 이루어졌다.

2. 삼각 측량 및 EKF를 통한 위치 계산

삼각 측량은 위치 좌표가 이미 알려진 기준 노드(anchor node)들과 위치를 계산하고자 하는 대상 노드(target node)간의 거리를 각각 측정하고, 이를 바탕으로 대상의 위치를 계산하는 방법이다[9]. CSS 시스템은 TOF를 바탕으로 거리를 측정할 수 있으므로, 삼각 측량은 CSS 기반의 위치 추정에 매우 적합하다. 그림 2는 평면상의 삼각 측량에 대한 기본

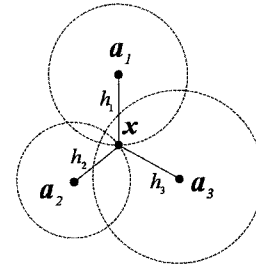


그림 2. 평면 좌표에서의 삼각 측량 개념도.

Fig. 2. Conceptual scheme of the trilateration on a plane.

적인 개념도이다. 여기서 k 번째 기준 노드의 2차원 좌표를 $a_k = [a_{k1} a_{k2}]^T$ 로, 대상 노드의 2차원 좌표를 $x = [x_1 x_2]^T$ 와 같이 벡터로 표현할 경우, 기준 노드 k 와 대상 노드간의 거리 h_k 는 다음과 같이 나타낼 수 있다.

$$h_k = \|x - a_k\|_2, \quad (2)$$

여기서 $\|\cdot\|_2$ 는 2-norm을 의미한다. 기준 노드 a_k 와 그에 대응되는 거리 h_k 의 쌍이 최소 3개 이상 존재할 경우 식 (2)를 연립하여 대상 노드의 2차원 좌표 x 를 구하는 것이 가능하다[10]. 그러나 실제의 경우 거리 h_k 는 실제 거리가 아닌 측정에 의해 얻어지므로 값에 오차가 포함되어 있다. 이 문제를 해결하기 위해서 연립방정식을 벡터-행렬 형태로 변환한 후 LS (Least Square) 방법을 사용할 수도 있다 [10]. 하지만 실시간 위치 추정을 위하여 대상의 좌표 x 를 상태 벡터(state vector)로 하는 비선형 상태 방정식을 기술한 후 이를 EKF에 적용하여 상태 벡터를 추정할 수 있다[11].

식 (3)과 (4)는 $N \times 1$ 시스템 잡음(system noise) 벡터 w_i 와 $M \times 1$ 측정 잡음(measurement noise) 벡터 v_i 을 포함하는 일반적인 형태의 비선형 상태 방정식(nonlinear state space equation)이다[12].

$$x_{i+1} = f(i, x_i) + w_i \quad \text{with initial state of } x_0 \quad (3)$$

$$y_i = h(i, x_i) + v_i, \quad (4)$$

여기서 $f(i, x_i)$ 와 $h(i, x_i)$ 는 각각 비선형 상태 전이 행렬 함수(nonlinear state transition matrix function)와 비선형 측정 행렬 함수(nonlinear measurement matrix function)이며, i 는 불연속 시각이다. $N \times 1$ 벡터 x_i 와 $M \times 1$ y_i 를 각각 시각 i 에서의 상태 벡터(state vector)와 측정 벡터(measurement vector)라고 한다. 또한 w_i 와 v_i 의 자기 상관 행렬(auto correlation matrix)을 각각 S_i 와 Q_i 로 나타낸다. N 은 상태(state)의 차원이며, M 은 측정값(measurement)의 차원이다.

식 (3)-(4)와 같은 비선형 상태 방정식으로 삼각 측량 및 로봇의 위치 좌표의 관계를 모델링하기 위하여, 먼저 시각 i 에서의 대상의 2차원 위치 좌표를 상태 벡터 $x_i = [x_1(i) x_2(i)]^T$ 로 정의한다. 이와 같이 정의된 상태 벡터에 대해, 일반적인 시스템 방정식 (3)은 대상 로봇의 위치 변화를 기술하기 위한 식 (5)로 표현된다.

$$x_{i+1} = x_i + \delta x_i + w_i \quad \text{with initial state of } x_0, \quad (5)$$

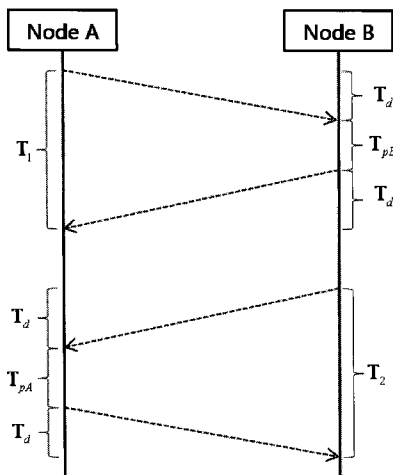


그림 1. Symmetric double sided 양방향 거리 측정.

Fig. 1. Symmetric double sided two way ranging.

여기서 $\delta \mathbf{x}_i$ 는 한 샘플링 주기 동안의 이동 로봇의 위치 변화를 의미한다. 즉, $\delta \mathbf{x}_i$ 는 일반적으로 로봇이 가진 엔코더와 같은 센서를 통해 측정된 이동 로봇의 속도 및 진행 방향(heading angle)의 변화를 통해 구해질 수 있으며, 그 값에 포함된 측정 잡음을 \mathbf{w}_i 로 표현한 관계식이 시스템 방정식 (5)이다. 다음으로 측정 방정식 (4)는 삼각 측량 관계식 (2)을 이용하여 다음과 같이 주어진다.

$$y_k(i) = h_k(i) + v_k(i) = \|\mathbf{x}_i - \mathbf{a}_k\|_2 + v_k(i), \quad (6)$$

여기서 $y_k(i)$ 는 측정 벡터 \mathbf{y}_i 의 k 번째 요소로서 시각 i 에서의 기준 노드 k 와 대상 간 측정된 거리이며, $v_k(i)$ 는 그 측정값에 더해지는 잡음으로서 벡터 \mathbf{v}_i 의 k 번째 요소이다. $h_k(i)$ 역시 벡터 함수 $\mathbf{h}(i, \mathbf{x}_i)$ 의 k 번째 요소이다. 즉, 식 (5)과 (6)은 이동 로봇의 위치를 상태 벡터로 하는 비선형 상태 방정식으로서 삼각 측량 및 로봇의 위치 좌표 변화를 모델링 한 것이다. 이러한 비선형 상태 방정식에 대한 EKF 알고리즘은 상태 벡터 \mathbf{x}_i 값을 추정하기 위해 사용되며, 사전에 주어진 추정된 상태 벡터 초기 값 $\hat{\mathbf{x}}_{1|0}$ 과 추정 오차의 자기 상관 행렬의 초기 값 $\hat{\mathbf{P}}_{1|0}$ 에 대해 다음과 같은 갱신식(update equations) (7)-(12)를 반복한다[12].

$$\mathbf{K}_i = \mathbf{P}_{i|i-1} \mathbf{H}_i^T [\mathbf{H}_i \mathbf{P}_{i|i-1} \mathbf{H}_i^T + \mathbf{Q}_i]^{-1} \quad (7)$$

$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}) \quad (8)$$

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i \mathbf{e}_i \quad (9)$$

$$\hat{\mathbf{x}}_{i+1|i} = \mathbf{f}(i, \hat{\mathbf{x}}_{i|i}) \quad (10)$$

$$\mathbf{P}_{i|i} = [\mathbf{I} - \mathbf{K}_i \mathbf{H}_i] \mathbf{P}_{i|i-1} \quad (11)$$

$$\mathbf{P}_{i+1|i} = \mathbf{F}_i \mathbf{P}_{i|i} \mathbf{F}_i^T + \mathbf{S}_i, \quad (12)$$

여기서 행렬 \mathbf{F}_i 와 \mathbf{H}_i 는 각각 식 (13)-(14)와 같이 정의된 Jacobian 행렬이다.

$$\mathbf{F}_i = \partial \mathbf{f}(i, \mathbf{x}) / \partial \mathbf{x}|_{\mathbf{x}=\hat{\mathbf{x}}_{i|i}} \quad (13)$$

$$\mathbf{H}_i = \partial \mathbf{h}(i, \mathbf{x}) / \partial \mathbf{x}|_{\mathbf{x}=\hat{\mathbf{x}}_{i|i-1}} \quad (14)$$

또한 행렬 \mathbf{I} 는 단위행렬(identity matrix)이며, $\hat{\mathbf{x}}_{i|i-1}$ 은 주어진 시스템 방정식을 기반으로 한 시간 갱신(time update)된 추정 값이며, $\hat{\mathbf{x}}_{i|i}$ 는 측정값을 기반으로 측정 갱신(measurement update)된 추정값이다. EKF은 새로운 측정값에서 현재까지의 추정치가 가지지 못한 정보만을 취하여 이를 측정 갱신에 반영 하므로, 추정값이 보다 정확한 값으로 수렴하도록 한다. 또한 EKF는 잡음으로 인한 측정값의 변동성이 있더라도 이를 처리할 수 있는 능력이 있으므로 위치 측정에 적용하기에 매우 적합한 방식이라고 할 수 있다.

III. CSS 측정값이 가지는 문제점 및 제안한 알고리즘

1. 측정 잡음이 가지는 문제점

앞서 언급한 일반적인 EKF는 잡음 \mathbf{w}_i , \mathbf{v}_i 의 평균이 0인 경우에 대해서 유효하다[12]. 식 (6)에 나타난 잡음 $v_k(i)$ 는 CSS 거리 측정값에 더해지는 잡음으로서 측정 잡음 벡터

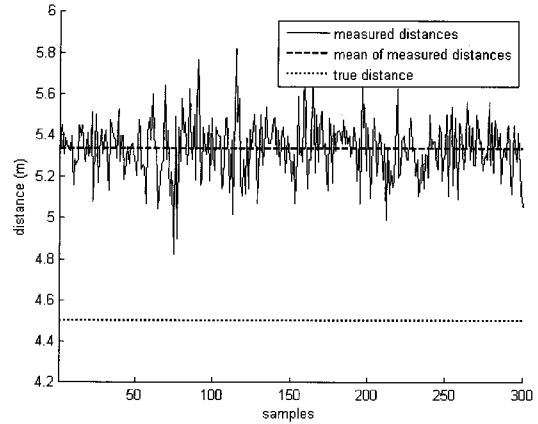


그림 3. CSS 기반의 측정거리와 평균값 및 실제거리.

Fig. 3. The measured distances and its average value with the true distance.

\mathbf{v}_i 를 구성하는데 EKF가 요구하는 잡음의 특성을 만족시키려면 CSS 거리 측정값이 가지는 잡음의 평균이 0, 즉 \mathbf{v}_i 의 평균이 영 벡터(zero vector)이어야 한다. 하지만 CSS 측정 거리는 매 샘플마다 변동하는 것은 물론 그 변동의 중심, 즉 매 샘플들의 평균이 실제 거리와 같지 않다. 이는 측정 거리 잡음의 평균이 0이 아님을 의미한다. 이러한 현상이 발생하는 이유는 가시 경로가 확보 되지 않거나 RF신호의 확산 등의 이유로 측정값이 실제 값과 다른 경우가 일반적이기 때문이다[11].

그림 3은 300회의 실제 CSS 거리 측정값들을 나타낸 것이다. 거리 측정은 4.5 m만큼 떨어진 두 CSS 노드 사이에서 이루어졌다. 측정 거리는 그림 3에 나타난 바와 같이 매 샘플마다 변동한다. 또한 그 샘플들의 평균은 실제 거리와 다른 5.33 m 정도임을 알 수 있다. 즉, 측정거리의 평균적인 오차는 약 1.33 m이다. 이러한 결과는 본 논문에서 제시한 실험 결과에만 국한 된 것이 아니며, 기존의 발표된 바 있는 다른 문헌상에서도 CSS 거리 측정값이 평균 오차를 가지고 있음을 알 수 있다[13].

이와 같이 CSS 측정값은 0이 아닌 평균과 분산을 가지는 잡음을 포함하는 것으로 모델링 할 수 있으며, 이러한 특성을 가지는 측정값을 EKF에 그대로 적용할 경우, EKF가 측정된 잡음 평균 0을 가정하여 유도된 알고리즘이므로, 그 결과가 올바르지 않을 수 있다. 물론 잡음의 평균값을 사전에 알고 있다면 그 평균값을 측정 거리에서 지속적으로 제거해줌으로써 잡음의 평균을 0으로 만드는 것이 가능하다. 하지만 거리 측정을 시도하는 두 노드간의 거리 및 주변 환경에 따라 잡음의 평균값이 다르기 때문에 실시간 알고리즘에는 적용될 수 없다. 따라서 측정된 거리가 가지는 잡음의 평균, 즉 측정값이 가지고 있는 평균 오차를 실시간으로 줄이는 알고리즘이 필요하다.

2. 제안한 알고리즘

앞서 설명한 CSS 측정거리가 가지는 평균 오차는 주변 환경의 특성에 따라 변동할 수 있다고 언급하였다. 이에 따라 본 논문에서 제안한 오차를 감소시키기 위한 알고리즘을 설명하기에 앞서 오차에 대한 논의가 필요하다. 먼저

CSS 노드 A, B, C, D가 존재한다고 가정하자. 또한 이때 노드 A, B와 노드 C, D는 각자 짝을 이루어 거리 측정을 수행하며 노드 A, B사이의 거리와 노드 C, D사이의 실제 거리는 동일하다고 가정한다. 만약 거리 측정이 이루어지는 환경이 매우 불규칙한 특성을 가진다면 노드 A, B간의 측정된 거리와 노드 C, D간의 측정된 거리에 포함된 평균 오차가 서로 다를 것이다. 이는 노드 A, B와 노드 C, D사이의 신호가 통과하는 경로의 성질이 서로 다르다는 것을 의미한다. 각 노드 쌍은 실제로 같은 거리에 존재하지만 측정된 거리의 평균이 각기 다른 평균오차에 의해 서로 다르게 측정되었다면 실제 거리는 알지 못하는 상황에서 다른 추가적인 장치 없이 그 오차 보정을 하는 것이 사실상 불가능하다. 본 논문은 CSS 외 추가적인 장비 없이 정확한 위치 추정이 가능하도록 하는 것을 목적으로 하므로, 이러한 상황은 배제하였다. 따라서 다음과 같은 환경을 가정하여 알고리즘을 개발하였다. 거리 측정을 시도하는 두 CSS 노드 간의 신호가 통과하는 경로는 잡음을 유발하여 평균 오차를 발생시키되, 그 오차의 정도는 두 노드가 어떤 위치에서 거리 측정을 하는가에 관계없이 두 노드의 거리가 동일하게 유지되고 동일 시간에 측정된다면 유사한 오차를 발생시킨다. 또한 측정 거리가 길어지거나 짧아질 경우, 해당 평균 오차 역시 유사한 비율로 커지거나 줄어든다고 가정한다. 이는 실험 환경이 오차를 발생시키지만 같은 순간 그 정도가 균일하다는 가정이다. 이러한 가정은 실제 CSS 거리측정을 수행하는 동안 관찰된 결과를 바탕으로 타당성이 있다고 평가된다.

이러한 가정 하에서 거리 측정값 벡터와 각 측정 거리에 포함되어 있는 잡음의 평균 오차의 관계를 다음과 같이 설명할 수 있다. 먼저 논문의 편의를 위해 2개의 거리 측정값 평균을 2차원 벡터 $\bar{\mathbf{y}}$ 로 표현한다. 즉, $\bar{\mathbf{y}} = [\bar{y}_1 \ \bar{y}_2]^T$ 이다. 여기서 \bar{y}_1, \bar{y}_2 는 각 고정 노드와 대상 간 측정된 거리를 의미한다. 또한 각 고정 노드와 대상 간 실제 거리 벡터를 $\mathbf{t} = [t_1 \ t_2]^T$ 라고 정의한다. 즉 t_1, t_2 는 측정 거리 y_1, y_2 에

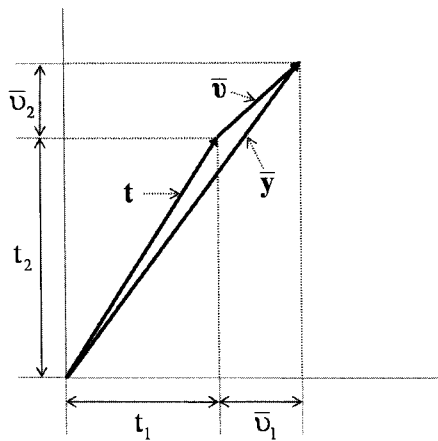


그림 4. 측정된 거리 벡터 및 실제 거리 벡터와 평균 오차 벡터의 관계.

Fig. 4. The relationship among the measured distance vector, the true distance vector and the average error vector.

대응되는 실제 거리이다. 그리고 실제 거리에 더해지는 잡음의 평균 오차 벡터를 $\bar{\mathbf{v}} = [\bar{v}_1 \ \bar{v}_2]^T$ 로 나타낸다. 즉 여기서 v_1, v_2 는 실제 거리 t_1, t_2 에 대응되는 평균 오차를 의미한다. 이러한 정의에 따라 벡터 $\bar{\mathbf{y}}$ 는 다음과 같이 벡터 \mathbf{t} 와 $\bar{\mathbf{v}}$ 의 합으로 나타낼 수 있다.

$$\bar{\mathbf{y}} = \mathbf{t} + \bar{\mathbf{v}}, \tag{15}$$

만약 실제 거리 벡터 \mathbf{t} 가 그림 4와 같이 주어진다면 앞서 언급한 가정에 따라 평균 오차 $\bar{\mathbf{v}}$ 는 실제 거리에 따라 유사한 비율로 나타날 것이므로 $\bar{\mathbf{y}}$ 는 그림 4와 같이 표현된다.

그림 4에 나타난 $\bar{\mathbf{y}}$ 를 통해 우리는 벡터 $\bar{\mathbf{y}}$ 의 방향은 실제 거리 벡터 \mathbf{t} 와 유사하지만 그 크기는 평균 오차 $\bar{\mathbf{v}}$ 가 더해짐에 따라 크게 차이가 남을 알 수 있다. 따라서 EKF에 사용될 측정 거리 벡터 \mathbf{y}_i 의 크기 정보를 제거하고 각도 정보만 취하는 방법을 이용하여 CSS 측정값이 가진 평균 오차, 즉 잡음의 평균을 제거하도록 할 것이다.

측정값 벡터 \mathbf{y}_i 가 가진 크기 정보를 배제하기 위해, 먼저 EKF의 갱신 식 중 \mathbf{e}_i 를 구하는 식 (8)에 주목해야 한다. 벡터 \mathbf{e}_i 는 실제 측정된 거리와, EKF에 의해 추정된 위치 $\hat{\mathbf{x}}_{i|i-1}$ 로부터 계산된 추정된 거리의 차이에 해당한다. 이는 \mathbf{y}_i 가 가진 정보 중, 현재까지의 추정치 $\hat{\mathbf{x}}_{i|i-1}$ 에 반영되지 못한 새로운 정보만을 추출하는 것과 같다. 따라서 \mathbf{e}_i 를 이노베이션(innovation)이라 부른다. 그러므로 평균 오차를 가진 \mathbf{y}_i 를 그대로 반영할 경우 오차에 해당하는 정보가 새로운 정보로 취급되는 것이며, 그 결과 추정결과가 잘못되게 된다. 앞서 언급 한 바와 같이 오차는 \mathbf{y}_i 의 크기 정보를 크게 변형 시키므로 우리는 먼저 \mathbf{y}_i 를 정규화(normalize) 하여 크기 정보를 제거한 후, 이를 이용한 새로운 이노베이션(innovation) \mathbf{e}'_i 을 다음과 같이 제안한다.

$$\mathbf{e}'_i = \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|_2} - \frac{\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})}{\|\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})\|_2}. \tag{16}$$

즉, 정규화한 \mathbf{y}_i 와 $\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})$ 의 차이를 구함으로써 크기 정보가 제거된 이노베이션을 얻고, 다시 $\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})$ 의 크기인 $\|\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})\|_2$ 을 그 값에 곱함으로써 추정된 거리 벡터의 크기 정보로 제거된 크기 정보를 복원하는 방식이다. 벡터 \mathbf{y}_i 의 크기와 마찬가지로, $\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})$ 의 크기 역시 실제 거리 벡터와는 다르다. 하지만 $\mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1})$ 는 측정값이 아닌 모델에 의한 계산에 의해 산출된 값이며, 특히 EKF에 의해 추정된 위치인 $\hat{\mathbf{x}}_{i|i-1}$ 가 실제 위치에 접근함에 따라 실제 거리 벡터로 수렴하게 되므로, 이러한 방식은 적합성을 가진다. 또한 식 (16)을 다음과 같이 다시 쓸 수 있다.

$$\mathbf{e}'_i = \mathbf{y}'_i - \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}), \tag{17}$$

여기서 위에서 \mathbf{y}'_i 는 식 (18)과 같이 주어진다.

$$\mathbf{y}_i = \frac{\| \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}) \|_2}{\| \mathbf{y}_i \|_2} \mathbf{y}_i \quad (18)$$

즉, 제안한 알고리즘은 벡터 \mathbf{y}_i 에 $\| \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}) \|_2$ 와 $\| \mathbf{y}_i \|_2$ 의 비율을 곱함으로써 보정된 벡터 \mathbf{y}_i 을 구하는 것이라고 할 수 있다.

지금 까지 소개한 삼각 측량 모델 및 제안한 알고리즘을 적용한 최종적인 EKF 갱신 식은 아래와 같이 정리할 수 있다.

$$\mathbf{K}_i = \mathbf{P}_{i|i-1} \mathbf{H}_i^T [\mathbf{H}_i \mathbf{P}_{i|i-1} \mathbf{H}_i^T + \mathbf{Q}_i]^{-1} \quad (19)$$

$$\mathbf{y}_i = \frac{\| \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}) \|_2}{\| \mathbf{y}_i \|_2} \mathbf{y}_i \quad (20)$$

$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{h}(i, \hat{\mathbf{x}}_{i|i-1}) \quad (21)$$

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i \mathbf{e}_i \quad (22)$$

$$\mathbf{x}_{i+1|i} = \mathbf{x}_{i|i} + \delta \mathbf{x}_i \quad (23)$$

$$\mathbf{P}_{i|i} = [\mathbf{I} - \mathbf{K}_i \mathbf{H}_i] \mathbf{P}_{i|i-1} \quad (24)$$

$$\mathbf{P}_{i+1|i} = \mathbf{F}_i \mathbf{P}_{i|i} \mathbf{F}_i^T + \mathbf{S}_i \quad (25)$$

식 (20)과 (21)이 제안한 알고리즘에 관련된 부분이며 기존의 EKF와 차이점에 해당한다.

IV. 실험 결과

1. 실험의 구성

본 논문에서 제안한 CSS기반의 이동 로봇의 위치추정 시스템의 평가를 위해 이동 로봇의 위치 추정에 관한 실험을 다음과 같이 진행하였다. 실험은 건물 실내에서 진행되었으며, 그림 5와 같이 바닥에 9m X 9m의 2차원 직교 좌표를 할당하였다. 할당된 좌표의 [4.5m, 4.5m], [-4.5m, 4.5m], [-4.5m, -4.5m], [4.5m, -4.5m] 지점에 4개의 기준 노드를 각각 배치하였다. 또한 그림 5에 나타난 바와 같이 좌표 평면상의 중심을 순회하는 경로 지정하고, 그 경로에 자기 테이프를 부착하여 guideline을 만들었다. 이에 대한 실제 실험 환경은 그림 6에 나타나 있다.

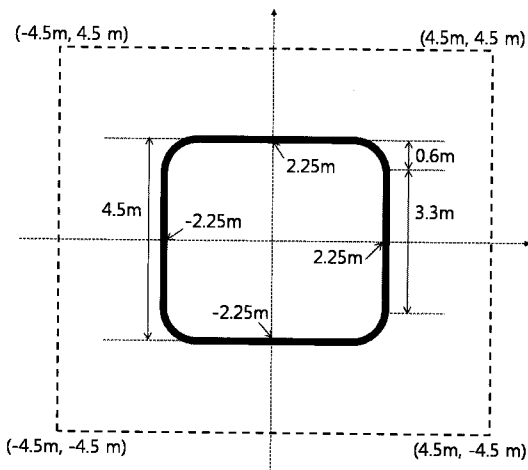


그림 5. 실험을 위한 기준 노드 배치와 로봇의 이동 경로.
Fig. 5. The configuration of anchor nodes and the path of mobile.

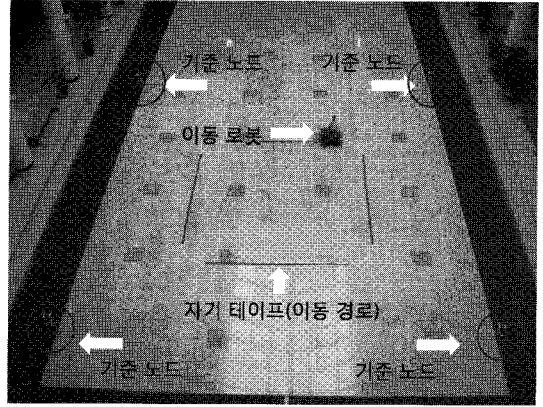


그림 6. 기준 노드와 이동 로봇을 배치한 실제 실험 환경.
Fig. 6. Actual experimental environment with anchor nodes and the mobile robot.

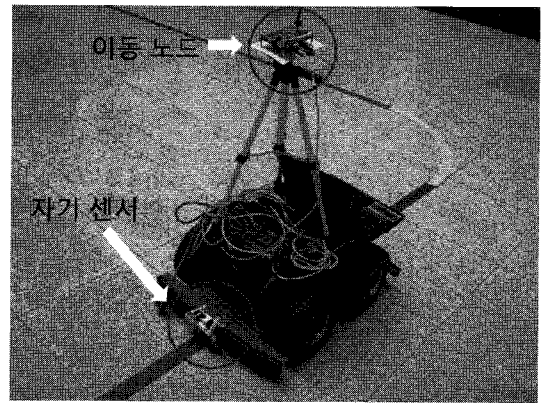


그림 7. 실험에 사용된 Pioneer 3 AT 이동 로봇.
Fig. 7. Pioneer 3 AT mobile robot for experiment.

실험에서 사용된 로봇은 그림 7에 나타난 Mobile Robot사에서 제조한 Pioneer 3 AT가 사용되었으며 해당 로봇에 자기 센서를 부착하여 바닥에 깔려있는 자기 테이프의 자기장을 읽어 로봇이 guideline을 따라갈 수 있도록 하였다. 또한 로봇에는 이동 CSS 노드가 장착되어 있어, 로봇이 경로를 따라 이동하는 동안 지속적으로 고정된 4개의 노드와 순차적인 거리 측정이 가능하도록 하였다. 또한 각 CSS 노드는 Nanotron사에서 만든 CSS 송수신 RF칩 인 nanoLOC TRX Transceiver(NA5TR1)를 사용한 제품을 이용하였다.

그림 8은 실험을 위한 로봇 및 CSS 노드, 자기 센서 간 신호의 흐름을 보여주고 있다. 주 모듈(main module)은 로봇이 자기 테이프를 이루어진 guideline을 따라 이동할 수 있도록 하기 위한 제어 부분과 본 논문의 핵심인 EKF를 이용한 로봇의 위치 추정 알고리즘 부분으로 나누어지는 프로그램이다. 4개의 기준 노드(anchor node)와 이동 노드(tag node) 간 측정된 거리는 EKF의 측정 거리 벡터 \mathbf{y}_i 를 구성하며, 제안된 알고리즘에 의해 그 값이 보정된다. 본 실험에서 사용한 로봇 Pioneer 3 AT는 내장된 펌웨어(firmware)가 있어 로봇의 엔코더로부터 얻은 데이터를 처리하여 자체적으로 자신의 위치 계산한다. 이 값이 곧 dead reckoning에 의한 로봇의 위치가 된다. 사용자는 이 위치값

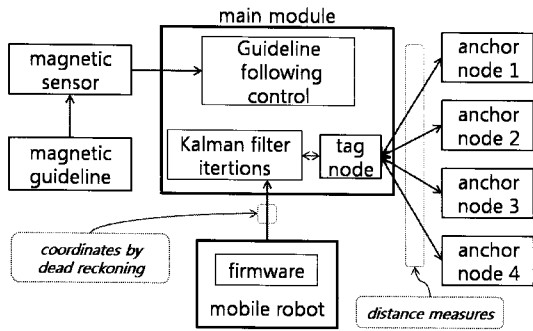


그림 8. 로봇 및 CSS 노드와 자기 센서 간 신호 흐름을 나타낸 블록도.

Fig. 8. Block diagram of signal following among the mobile robot, CSS nodes and a magnetic sensor.

을 로봇이 제공하는 라이브러리 함수를 이용하여 얻을 수 있다. 또한 본 실험에서는 식 (23)의 매 샘플 주기 동안 변화한 로봇의 위치 값 벡터 δx_i 를 firmware가 계산한 로봇의 이전 위치와 현재 위치 값의 차이를 이용하여 계산하였다. 앞서 언급한 바와 같이 일반적인 방법의 δx_i 계산은 엔코더 데이터로부터 얻은 로봇의 속도 및 진행 방향 등과 관계된 로봇의 시스템 모델에 의해 계산되어야 한다. 이 경우 상태 벡터에 로봇의 속도 및 진행 방향과 같은 물리량이 포함되는 것이 보통이며 상태 벡터와 δx_i 는 밀접한 관계를 가진다. 하지만 본 논문에서는 추정된 로봇의 위치값에 초점을 맞추기 위하여 로봇의 2차원 좌표값만으로 상태 벡터 x_i 를 정의하였기 때문에 로봇의 위치 변화 δx_i 는 상태 벡터와 직접적인 관계가 없다. 또한 δx_i 는 매 시각마다 firmware로부터 얻은 값을 바탕으로 계산되므로 상태 벡터를 변화시키는 일종의 시스템 입력으로 간주할 수 있다. 따라서 이 경우 행렬 F_i 는 식 (13)의 정의에 따라 단위행렬 (identity matrix)이 된다.

본 실험에서 EKF의 추정 위치의 초기값으로는 $\hat{x}_{10} = 0$, 추정 오차의 자기상관행렬의 초기값으로는 $\hat{P}_{10} = I$ 를 적용하였다. 시스템 및 측정 잡음의 자기 상관행렬 값으로는 $S_i = \text{diag}(0.01)$, $Q_i = \text{diag}(0.8547)$ 를 사용하였으며 이는 실험적으로 구한 값이다. 여기서 $\text{diag}(\alpha)$ 는 α 로 이루어진 대각행렬을 의미한다. 로봇은 [0m, 2.25m]에서 출발하여 경로를 시계방향으로 회전하도록 하였고, 속도는 25cm/sec로 설정하였다. 우리는 성능 평가를 위해 dead reckoning에 의해 계산된 로봇의 위치와 CSS 거리 측정값을 바탕으로 한 제안한 알고리즘에 의한 위치를 비교하고자 한다. Dead reckoning은 로봇의 초기 위치를 알고 있어야 가능하므로 로봇 출발 시 초기 위치 좌표를 dead reckoning에 제공하였다.

2. 실험 결과

앞서 설명한 실험 환경에서 로봇을 시계 방향으로 경로를 3회 회전하도록 한 실험결과를 그림 9에 나타내었다. 실제 로봇이 이동한 경로는 검은색의 실선으로 표시되어 있다. 이 경로는 실제 자기 테이프가 설치된 경로와 일치한다. 동그라미 기호와 함께 표시된 붉은색 실선은 로봇의 엔

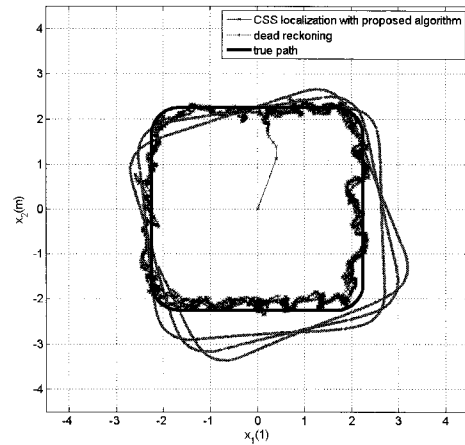


그림 9. 로봇의 실제 경로 및 제안한 알고리즘을 적용한 CSS 시스템에 의해 추정된 위치와 dead reckoning에 의해 계산된 위치.

Fig. 9. The actual path of the mobile robot and the estimated coordinate by the CSS system with the proposed algorithm and by the dead reckoning.

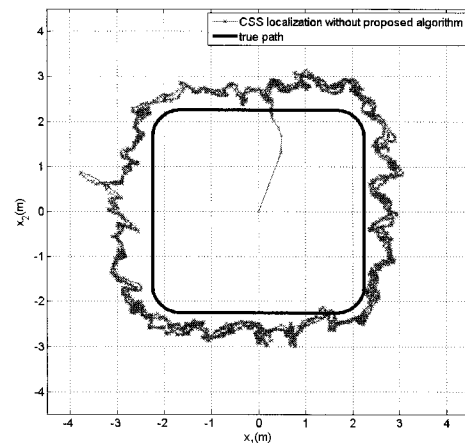


그림 10. 로봇의 실제 경로 및 제안한 알고리즘을 사용하지 않은 CSS 시스템에 의해 추정된 위치.

Fig. 10. The actual path of the mobile robot and the estimated coordinate by the CSS system without the proposed algorithm.

코더를 기반으로 한 dead reckoning을 통해 계산한 로봇의 위치 값을 표시한 것이다. 초기 출발위치가 dead reckoning을 위해서 정확히 제공되었기 때문에 초기에는 비교적 정확한 위치 값을 가진다. 하지만 곡선 경로를 통과하기 위하여 로봇의 진행 각도가 변경될 때 마다 일정한 오차가 누적되는 것을 볼 수 있다. 따라서 경로를 순환하는 회수가 증가할수록 계산된 위치 값과 실제 이동 경로의 차이가 점점 커지는 것을 알 수 있다. 가위표 기호와 함께 표시된 푸른색 실선은 본 논문에서 논의한 측정 거리의 오차를 줄이기 위한 알고리즘을 적용한 CSS 시스템에서 산출한 추정 결과를 나타낸 것이다. 그림 9에서와 같이 EKF가 주어진 초기값 $\hat{x}_{10} = 0$ 에서 로봇의 위치로 수렴하는 부분을 제외

하면 최대 50cm 정도의 오차를 보이며 추정된 위치값은 실제 경로를 중심으로 변동성을 보인다. 하지만 정확한 구간에서는 20cm 수준의 오차를 보이며, dead reckoning에 의한 결과와는 달리 로봇이 실제 경로를 따라 계속 회전하여도 오차가 누적되지 않는 것을 볼 수 있다.

그림 10은 CSS 시스템에서 측정된 거리를 보정하지 않고 EKF에 적용한 후 얻은 추정된 이동 로봇의 좌표를 보여주고 있다. 이 결과는 그림 10에 가위표 기호와 함께 표시된 붉은색 실선으로 나타나 있으며 알고리즘을 적용한 경우인 그림 9의 추정치와 비교하였을 때 오차가 매우 크다는 것을 알 수 있다. 이 경우 실제 이동 경로에서 1~2m 정도 벗어난 위치 값이 계산되는 등 결과가 부정확하다. 따라서 본 논문에서 제안한 측정거리의 오차보정 알고리즘이 효과적임을 알 수 있다.

그림 11은 그림 9와 그림 10에 나타난 결과에서 x_1 방향의 추정된 좌표를, 그림 12는 x_2 방향의 추정된 좌표를 때 iteration에 대해 나타낸 것이다. 그림 11과 12의 각 추정치는 제안한 알고리즘을 적용한 EKF의 추정치, 제안한 알고리즘을 사용하지 않은 EKF의 추정치 그리고 dead reckoning에 의해 계산된 좌표이다. 그림 11과 12의 2.25m와

-2.25m를 표현한 선은 본 실험 세팅에 따른 로봇의 위치 좌표 변화의 최대값과 최소값에 해당한다. 그림 11과 12에 나타난 바와 같이 dead reckoning에 의해 계산된 위치는 시간이 지날수록 최대, 최소값에서 점점 멀어져 오차가 누적된다는 것을 확인할 수 있다. CSS 거리 측정값을 바탕으로 한 EKF에 의한 추정된 좌표는 시간이 지나도 오차가 누적되지 않는다. 하지만 제안한 알고리즘을 적용한 경우의 추정치가 최대 최소값에 더욱 근접하다는 것을 알 수 있다.

지금까지 살펴본 실험 결과를 바탕으로 CSS 위치 추정 시스템은 실내 로봇을 특정 위치에서 다른 위치로 이동시키기 위한 네비게이션에 사용 가능할 것이다. 특정 위치에 이동 로봇을 정확히 멈추는 것은 목표 위치 근방에 beacon을 두고 이를 활용하는 것이 가능하므로, CSS 시스템과 다른 추가적인 방법과 연동할 경우 더욱 활용 가능성을 높일 수 있을 것이다. 또한 본 논문의 실험 결과는 추정치를 여과 없이 나타낸 것으로서, 실제 응용의 경우 로봇의 이동 속도 등을 고려하여 현재 속도로는 실현 불가능한 CSS 거리 측정값 혹은 추정된 위치 값의 변동을 모니터링하여 처리 한다면 더욱 훌륭한 결과를 얻을 수 있을 것이라 생각된다.

V. 결론

본 논문은 CSS 기반 위치 추정 시스템에서의 정확도를 향상시킬 수 있는 알고리즘을 제안하였고 이를 이동 로봇에 적용하였다. 위치 추정 방식은 삼각 측량 모델을 비선형 상태 방정식으로 표현하고 이를 EKF에 적용한 기존의 연구를 참고하였다. CSS 거리 측정값이 가지는 잡음의 평균이 일반적으로 0이 아님을 직접적인 실험과 기존에 발표된 문헌을 바탕으로 보였고, 그러한 거리 측정값을 EKF에 그대로 반영할 경우, EKF에 의해 추정된 위치값이 틀릴 수 있음을 설명하였다. 이에 따라, 거리 측정값의 평균적인 오차에 해당하는 잡음의 평균을 제거하고자 측정 거리 벡터를 정규화하여 EKF의 이노베이션(innovation)을 구하는 알고리즘을 제안하였다. 제안한 방법은 곧 EKF에 의해 추정된 위치를 기반으로 계산된 거리 벡터와 측정된 거리 벡터의 크기 비율을 측정 거리값에 곱하는 방식으로 측정거리를 변경시키는 것과 같음을 수식으로 나타내었다. 또한 이를 이용한 실제 실험을 통하여 추정된 로봇의 위치가 dead reckoning을 이용한 방법에 비해 정확하고 실제 적용 가능성이 있음을 보였다.

참고문헌

[1] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57-66, August, 2001.

[2] 최효식, 황진아, 이장명, "유비쿼터스 모바일 로봇의 강인한 위치 추정 기법," 제어·로봇·시스템학회 논문지, 제14권 제12호, pp. 1139-1145, 2008.

[3] 권지욱, 박문수, 김태은, 좌동경, 홍석교 "간접 칼만 필터 기반의 센서융합을 이용한 실의 주행 이동로봇의 의미 추정," 제어·로봇·시스템학회 논문지, 제14

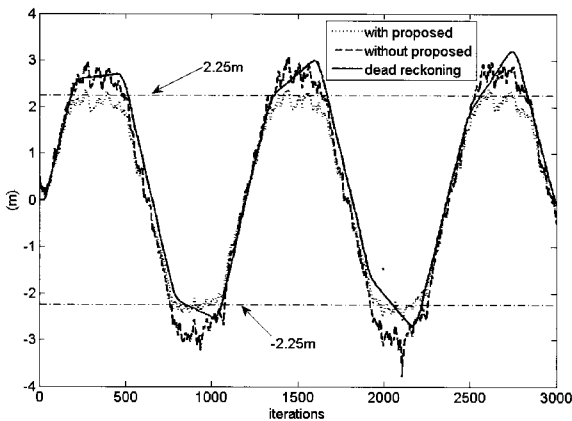


그림 11. 시간에 따른 추정된 x_1 좌표 값의 변화.
Fig. 11. The variation of the estimated x_1 over time.

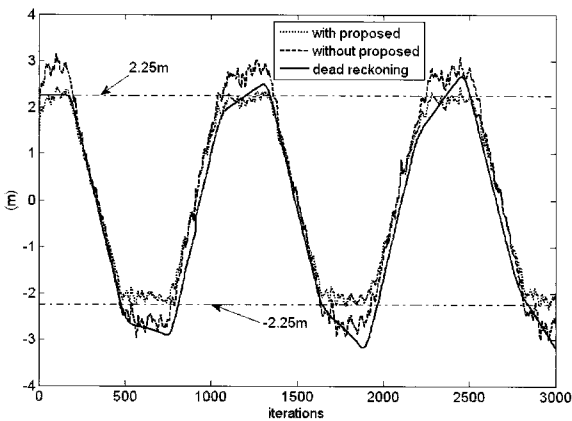


그림 12. 시간에 따른 추정된 x_2 좌표 값의 변화.
Fig. 12. The variation of the estimated x_2 over time.

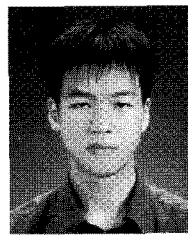
권 제8호, pp. 800-808, 2008.

- [4] 이동환, 김수용, 이만형, "유비쿼터스 컴퓨팅의 실내 측위 기술을 위한 U-SAT 시스템의 적용에 관한 연구," 제어·로봇·시스템학회 논문지, 제11권 제8호, pp. 1-8, 2006.
- [5] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," *Proc. of the 6th Conf. Mobile computing and networking*, pp. 32-43, Aug., 2000.
- [6] *Real Time Location Systems, IEEE Trans. of Automatic Control*, Nanotron Technologies GmbH, Berlin, Germany, White paper NA-06-0248-0391-1.02, 2007.
- [7] J.-E. Kim, J. Kang, D. Kim, Y. Ko, and J. Kim, "IEEE 802.15.4a CSS-based localization system for Wireless sensor networks," *Mobile Adhoc and Sensor Systems 2007*, pp. 1-3, Oct., 2007.
- [8] Y. Jiang and V. C. M Leung, "An asymmetric double sided two-way ranging for crystal offset," *Signals Systems and Electronics 2007*, pp. 525-528, Aug., 2007.
- [9] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Trans. on Robotics*, pp. 93-101, Feb., 2005.
- [10] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499-518, Nov., 2003.
- [11] S. Spieker and C. Röhrig, "Localization of pallets in warehouses using wireless sensor networks," *Proc. of the 16th Mediterranean Conf. Control and Automation*, pp. 1833-1838, Jun., 2008.
- [12] S. Haykin, *Adaptive filter theory 4th ed.*, Prentice Hall, 2002.
- [13] 권대길, 조진웅, 임승욱, 이장연, 이현석, 원윤재, "개인 무선네트워크에서 CSS 방식과 RSSI를 이용한 거리 측정에 관한 연구," 대한전자공학회 하계종합학술대회, 제31권 제1호, pp. 321-322, 2008.



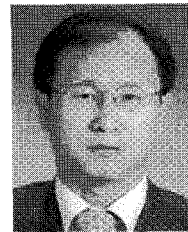
조 현 우

2006년 한양대학교 전자전기공학부(공학사). 2006년~현재 포항공과대학교 전자전기공학과 석박사통합과정 재학.



이 영 훈

2009년 홍익대학교 전자전기공학부(공학사). 2009년~현재 포항공과대학교 전자전기공학과 석사과정 재학.



김 상 우

1983년 서울대학교 제어계측공학과(공학사). 1985년 동 대학원 제어계측공학과(공학석사). 1990년 동 대학원 제어계측공학과(공학박사). 1991년~현재 포항공과대학교 전자전기공학과 부교수. 관심분야는 최적제어, 최적화 알고리즘, 지능제어, 공정자동화.