

이동로봇의 SLAM을 위한 선형화된 측정표현법

Linearized measurement for the SLAM application of mobile robot

*이건용, #도낙주

*K. Lee, #N.L. Doh(nakju@korea.ac.kr)
고려대학교 전기전자전파공학과 로봇 연구실

Key words : Linearized Measurement, EKF SLAM, Point Landmark

1. 서론

SLAM(동시적인 위치인식과 지도그리기)은 센서 정보를 이용해서 주변 환경의 지도를 그리고 로봇의 위치를 예측하는 과정이다. 자율 이동 로봇의 경우 SLAM을 수행하는 능력은 아주 중요하다. 1980년대 후반부터 지금까지 SLAM에 대한 많은 연구가 있어왔다. 그 동안 지속적인 성능향상이 있었고, 현재는 실내 환경에서 레이저 센서를 사용하는 이동 로봇의 경우, 환경을 단순화시킨 상태에서 로봇은 자기 위치 인식과 주변 환경의 지도 작성을 훌륭하게 수행함을 볼 수 있다. SLAM에서 최근의 연구 분야는 로봇의 위치를 예측하는 알고리즘, 인식한 주변 정보를 가지고 등록된 정보와 비교, 판단하는 알고리즘, 센서 데이터를 처리하는 알고리즘으로 크게 나눌 수 있다.[1] 그리고 SLAM에 대한 다양한 해결방법 중 ‘확장된 칼만 필터’ (EKF)는 실시간으로 좋은 성능을 보이는 것으로 증명된 알고리즘이다. 이 알고리즘은 비선형 방정식을 선형화 시킴으로써 처리 속도를 향상시켰고, 로봇의 위치 인식과 지도 작성에 있어서도 뛰어난 성능을 보이고 있다. 하지만 이 알고리즘 안에는 비선형 방정식을 선형화시키는 과정이 있기 때문에 측정된 결과에는 ‘선형화 에러’가 포함되어 있다. 선형화시키는 과정에서 속도 향상을 보였지만, 에러가 포함되기 때문에 이 문제를 해결하기 위한 다양한 방법이 시도되어졌다. ‘입자 필터’ 그리고 ‘Unscented 칼만 필터’ [2]가 대표적인 예에 해당한다.

본 논문에서는 ‘확장된 칼만 필터’의 기본 과정을 유지한 상태에서 선형화 에러를 최소화시킬 수 있는 방법에 대해서 연구, 정리하였다. 단순히 측정표현법만 변형시킴으로써 비선형방정식을 선형 방정식으로 근사화 시키는 과정을 최소화할 수 있음을 보인다.

본 논문은 다음과 같이 구성된다. 두 번째 부분에서는 기존의 ‘확장된 칼만 필터’ 알고리즘의 과정을 간단히 정리하고 선형화 에러가 포함되는 부분에 대해서 설명한다. 세 번째 부분에서는 앞에서 설명한 수식을 약간 변형시킴으로써 선형화 과정을 최소화 시킬 수 있음을 보인다. 그리고 네 번째 부분에서는 컴퓨터 시뮬레이션 결과를 통해서 선형화된 측정 표현법을 이용한 알고리즘의 성능이 더 좋음을 보인다.

2. 확장된 칼만 필터(EKF)

측정 이론에서 사용되는 ‘확장된 칼만 필터’는 비선형 방정식을 위한 칼만 필터이다. SLAM 알고리즘의 모든 과정의 데이터들은 가우시안 분포를 갖는다고 가정을 하고, 각각의 평균과 분산값을 수식으로 표현된다. 그리고 각 데이터들은 전체 좌표에서의 값 그리고 로봇 좌표에서의 값으로 변환되는 과정을 거치게 된다. 이 변환과정은 비선형방정식으로 표현된다. 칼만 필터를 이용하기 위해서는 선형 방정식이 필요하기 때문에, 비선형 방정식을 선형 방정식으로 근사화 시키는 과정이 추가가 되는 데 이 전체 과정을 ‘확장된 칼만 필터’라고 한다.[3]

$${}^r z_{m^1:p} = \begin{bmatrix} {}^r r_{m_1} & {}^r r_{m_2} & \dots & {}^r r_{m_p} \\ {}^r b_{m_1} & {}^r b_{m_2} & \dots & {}^r b_{m_p} \end{bmatrix} \quad (1)$$

$${}^w X_f = s({}^w X_r, {}^r Z_f) = \begin{bmatrix} {}^w x_r + r_f \cos({}^r b_f + {}^w \psi_r) \\ {}^w y_r + r_f \sin({}^r b_f + {}^w \psi_r) \end{bmatrix} \quad (2)$$

$${}^w X_f \approx {}^w S_r {}^w X_r + {}^w S_f {}^r Z_f \quad (3)$$

$${}^w S_r = \frac{\partial s}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \psi} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -r_f \sin({}^r b_f + {}^w \psi_r) \\ 0 & 1 & r_f \cos({}^r b_f + {}^w \psi_r) \end{bmatrix} \quad (4)$$

$${}^w S_f = \frac{\partial s}{\partial Z} = \begin{bmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial b} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial b} \end{bmatrix} = \begin{bmatrix} \cos({}^r b_f + {}^w \psi_r) & -r_f \sin({}^r b_f + {}^w \psi_r) \\ \sin({}^r b_f + {}^w \psi_r) & r_f \cos({}^r b_f + {}^w \psi_r) \end{bmatrix} \quad (5)$$

수식 (1)은 point landmark에 대한 표현이다. 로봇이 센싱한 데이터는 로봇으로부터의 거리와 각도값으로 표현된 값이다. 그리고 수식 (2)의 과정을 통해서 전체 좌표에서의 (x,y)값으로 변환된다. 이 수식은 비선형식이기 때문에, 변환된 분산값을 알기 위해서는 선형화 과정이 필요하다. 수식 (3)은 선형화된 결과를 의미하고, 수식 (4)와 수식 (5)는 선형화 과정에서 유도된 자코비언 행렬이다.

$${}^r Z_f = g({}^w X_r, {}^w X_f) = \begin{bmatrix} \sqrt{({}^w x_f - {}^w x_r)^2 + ({}^w y_f - {}^w y_r)^2} \\ \text{atan}(({}^w y_f - {}^w y_r) / ({}^w x_f - {}^w x_r)) - {}^w \psi_r \end{bmatrix} \quad (5)$$

$${}^r Z_f \approx {}^w H_r {}^w X_r + {}^w H_f {}^w X_f \quad (6)$$

$${}^w H_r = \frac{\partial g}{\partial X} = \begin{bmatrix} \frac{-\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} & \frac{-\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} & 0 \\ \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} & \frac{-\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} & -1 \end{bmatrix} \quad (7)$$

$${}^w H_f = \frac{\partial g}{\partial XY} = \begin{bmatrix} \frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} & \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} \\ \frac{-\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} & \frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} \end{bmatrix} \quad (8)$$

수식 (5)는 등록된 landmark정보를 로봇 중심에서의 거리와 각도값으로 변환시키는 비선형식이다. 그리고 수식 (6)은 이 비선형방정식을 선형화 시킨 결과이다. 수식 (7)과 (8)은 수식 (4), (5)와 마찬가지로 자코비언 행렬을 유도한 결과이다.

이와 같이 ‘확장된 칼만 필터’ 알고리즘 과정에는 총 2번의 선형화 과정이 포함된다. 그리고 이 과정에서 선형화 에러가 측정된 값에 들어가게 된다.

3. 선형화된 측정 표현법

위의 과정에 의해 ‘확장된 칼만 필터’에서는 총 2번의 선형화 과정이 있음을 보았다. 그리고 여기에서는 표현법을 다르게 함으로써 선형화 과정이 한번만 포함이 되도록 할 수 있음을 보인다.

$${}^r z_{m^1:p} = \begin{bmatrix} {}^r r_{m_1} & {}^r r_{m_2} & \dots & {}^r r_{m_p} \\ {}^r b_{m_1} & {}^r b_{m_2} & \dots & {}^r b_{m_p} \end{bmatrix} \quad (9)$$

$${}^r P_m = \begin{bmatrix} r_m \cos(w\psi_r + r b_m) \\ r_m \sin(w\psi_r + r b_m) \end{bmatrix} = f(w X_r, r Z_m) \approx {}^w S_r {}^w X_r + {}^r S_m {}^r Z_m \quad (10)$$

$${}^w S_r = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \psi} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -r_f \sin(r b_f + w\psi_r) \\ 0 & 0 & r_f \cos(r b_f + w\psi_r) \end{bmatrix} \quad (11)$$

$${}^r S_f = \frac{\partial f}{\partial Z} = \begin{bmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial b} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial b} \end{bmatrix} = \begin{bmatrix} \cos(r b_f + w\psi_r) & -r_f \sin(r b_f + w\psi_r) \\ \sin(r b_f + w\psi_r) & r_f \cos(r b_f + w\psi_r) \end{bmatrix} \quad (12)$$

수식 (9)부터 수식 (12)까지는 비선형방정식에 의한 데이터의 형태의 변환과정과 자코비안 행렬을 보여준다. 이 과정에서는 기존의 과정과 마찬가지로 선형화 어려가 포함된다. 그리고 이 변환과정의 결과는 로봇 위치를 중심으로 한 landmark까지의 x, y 거리값이다.

$${}^w P_f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^w X_r + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} {}^r P_f \quad (13)$$

$${}^r P_f = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} {}^w X_r + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} {}^w P_f \quad (14)$$

landmark에 대한 정보를 로봇에 대한 거리와 각도값이 아닌, x와 y에 대한 거리값으로 표현함으로써 식 (13), (14)와 같이 전체 좌표로 등록할 때 그리고 로봇 좌표로 다시 정보를 가져오는 과정이 선형화되었다. 즉, 좌표가 바뀌는 과정에서 비선형식으로 표현되는 것이 아닌 덧셈, 뺄셈의 식으로 바뀌게 된 것이다. 기존의 ‘확장된 칼만 필터’의 과정과 비교를 해 보면, 선형화 과정이 두 번에서 한 번으로 줄어들었다. 즉 알고리즘을 통해 예측한 로봇의 위치 정보에 선형화 어려가 더 적게 포함되게 된다. 다음 파트에서는 이 과정에 대한 컴퓨터 시뮬레이션 과정을 보인다.

4. 시뮬레이션

본 논문에서 제안한 방법의 성능을 테스트하기 위해 컴퓨터 시뮬레이션을 수행하였다. 기존의 방법과 본 논문에서 제안한 방법의 성능 차이를 극대화하기 위해 측정값에 대한 불확실도를 크게 넣어주었다.



Fig. 1 Experiment Environment

그림 1과 같은 환경에서 시뮬레이션을 진행하였다. 로봇은 정확도가 큰 레이저 센서를 사용한다고 가정하였고, 로봇이 인식하는 landmark 정보에 대한 불확실도는 크게 넣어 주었다. 그리고 로봇은 전체 루프를 세 번 반복하였다.

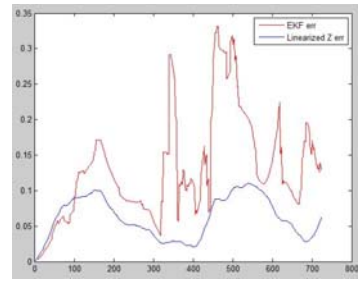


Fig. 2 Experiment Result. Robot position error with respect to the real robot position

그림 2는 로봇의 실제 위치와 각각의 알고리즘을 적용하여 예측한 로봇 위치와의 거리를 그래프로 표현한 것이다. 빨간색은 기존의 알고리즘을 적용한 결과이고, 파란색은 본 논문에서 제안한 방법을 적용한 결과이다. 레이저 센서 노이즈의 표준편차는 0.01m였고, 로봇의 바퀴 노이즈의 표준편차는 0.005m. 그리고 landmark에 대한 노이즈의 표준편차는 1m였다.

5. 결론

본 논문에서는 로봇 네비게이션 알고리즘에서 많이 사용되고 있는 ‘확장된 칼만 필터’에 대한 선형화된 측정 표현법을 제시하였다. 전체 과정은 바뀌지 않지만, 측정값의 형태를 바꿈으로써 비선형 방정식 하나를 제거할 수 있었다. 선형화 과정을 줄임으로써 선형화 어려가 줄어든 것을 수식으로 보였고, 컴퓨터 시뮬레이션을 통해 그 결과를 검증하였다. 그 다음 연구로는, 일반화된 환경에서의 성능향상을 보이는 것이다. 수식상으로 선형화 과정은 선형화 어려움을 포함시키는 것이 분명하게 보이지만, 실제로 그 값은 크지 않고, 특수한 경우에 극대화되어 나타나기 때문이다. 그리고 로봇의 위치값만이 아니라, 로봇의 각도값 그리고 로봇의 불확실도에 대한 표현도 성능이 향상되는지를 보일 수 있어야 한다. 다양한 환경에서 성능 테스트를 하고, 그 결과에 영향을 주는 중요한 파라미터들이 어떤 것들이 있는지를 더 연구할 계획이다.

후기

본 연구는 (고려대학교 로봇자율주행기술 전문인력양성센터를 통한) 지식경제부/한국산업기술진흥원 융복합형로봇전문인력양성사업의 지원으로 수행되었음.

참고문헌

1. Guoquan P. Huang, Anastasios I. Mourikis and Stergios I. Roumeliotis, "Generalized Analysis and Improvement of the Consistency of EKF-based SLAM", IEEE Transactions on robotics, 2007
2. Eric A. Wan and Rudolph van der Merwe, "The Uncented Kalman Filter for Nonlinear Estimation" The IEEE 2000 Adaptive Systems for Signal
3. M. Gautier and Ph.Poignet, "Extended Kalman filtering and weighted least squares dynamic identification of robot" Control Engineering Practice Volume 9, issue 12, December 2001, Pages 1361-1372