

칼만필터를 사용한 자율주행로봇의 항법제어기 설계

최광섭*, 박태형**
충북대학교 제어계측공학과

Design of Navigation Controller for Autonomous Mobile Robots using Kalman Filter

Kwang-Sup Choi, Tae-Hyoung Park
Dept. of Control and Instrumentation Eng., Chungbuk National University

Abstract - When it is used for autonomous mobile robots by using dead-reckoning system, odometry system with encoder is the simplest method as well as well-known in the industry. However, odometry system is reflected slide, friction and mechanical errors of wheels when operating the position estimation. And also in order to minimize errors of direction angle which is the most important factor that it is designed the controller in controlling kinematics and quadratic curve, PID that came into the values of sensor fusion with encoder and gyroscope sensor. After designing, the autonomous mobile robot is produced practically and inspected how it works.

1. 서 론

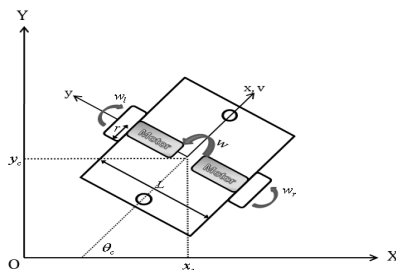
자율주행로봇 시스템은 최근 생산 시스템의 생산성 향상 및 국제 경쟁력을 향상시키기 위하여 연구소와 기업 등에서 집중적으로 연구, 개발되고 있다. 또한 자율주행로봇은 이동성과 조작성이 우수할 뿐만 아니라, 작업 반경이 넓고 동작 반경은 작으며, 작업 환경 변화에 유연하게 대처할 수 있을 뿐만 아니라 이동속도가 빠르고 제어의 용이하다[1]. 그리고 산업현장 뿐만 아니라 가정에서도 사람을 대신할 수 있는 청소로봇, 경비로봇등과 폭탄제거, 원자로와 같이 사람이 작업하기 힘든 상황에서 사용될 수 있는 무한한 잠재력을 갖고 있다. 이러한 자율주행로봇은 스스로 자기 위치를 파악하여 주행하여야 하므로, 별도의 항법제어기가 필요하다. 항법제어기는 엔코더, 자이로스코프, 가속도센서, 지자계센서등의 자율주행로봇의 내부 센서와 비콘,GPS 등의 차량 외부 장치를 활용하여, 실시간으로 차량의 절대 위치를 파악하고 목적한 경로를 따라 목표 위치에 이동로봇을 이동시키는 장치이다[2].

본 논문에서는 자율주행로봇의 위치를 추정하기 위해서 이동로봇의 내부센서인 엔코더와 자이로스코프를 사용하였고, 엔코더와 자이로스코프의 오차를 보정하기 위해서 칼만필터를 이용하여 엔코더와 자이로스코프의 오차를 보정하여 방법을 제안하였다. 또한 경로제어 방법으로는 Quadratic Curve 방법을 이용하여 경로제어를 하였고, 이러한 항법제어기의 성능을 검증하고자 시뮬레이션과 실제 자율주행로봇과 항법제어기를 설계하여 성능을 검증 하였다.

2. 시스템 모델링

2.1 자율주행로봇의 기구학적 모델

자율주행로봇은 그림1와 같이 위치(x_c, y_c)와 자세(θ_c)로 정의하고 다음과 같은 기구학 방정식을 얻을 수 있다.



<그림 1> 자율주행로봇의 기구학

$$\begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} r/2 & r/2 \\ r/L & -r/L \end{pmatrix} \begin{pmatrix} w_r \\ w_l \end{pmatrix} \quad (1)$$

$$x_{c,k+1} = x_{c,k} + v\Delta t \cos(\theta_{c,k} + w\Delta t/2) \quad (2)$$

$$y_{c,k+1} = y_{c,k} + v\Delta t \sin(\theta_{c,k} + w\Delta t/2) \quad (3)$$

$$\theta_{c,k+1} = \theta_{c,k} + w\Delta t \quad (4)$$

여기서 v, w 는 각각 로봇 차체의 선속도와 각속도를 나타내고 w_r, w_l 은 오른쪽, 왼쪽 바퀴의 각속도, L, r 은 로봇의 차축과 바퀴의 반지름을 나타낸다.

2.2 센서 모델링

2.2.1 엔코더

엔코더는 자율주행로봇의 움직인 거리와 속도를 연속적으로 감지할 수 있다는 장점이 있으나, 바퀴의 슬립현상, 로봇의 두 바퀴의 크기 부정확성으로 인한 오차가 발생하게 되어 이동거리가 증가함에 따라 오차가 누적되는 단점이 있다. 이를 보완하기 위해서 엔코더의 이산화된 오차 모델이 필요하고 식(2), (3), (4)는 참값이라 가정하고 오차가 포함된 측정치에 식에 참 값을 빼서 오차 식을 구한다[3].

$$\delta x(k+1) = \delta x(k) + \cos\theta(k) (U_R^m(k) + U_L^m(k))/2 \times \delta\theta(k) \quad (5)$$

$$\delta y(k+1) = \delta y(k) - \sin\theta(k) (U_R^m(k) + U_L^m(k))/2 \times \delta\theta(k) \quad (6)$$

$$\delta\theta(k+1) = \delta\theta(k) \quad (7)$$

여기서 U_R^m, U_L^m 은 각각 오른쪽과 왼쪽의 엔코더로부터 측정된 실제 이동 거리를 나타낸다. $\delta\theta(k)$ 는 매우 작다고 가정하고, $\cos\delta\theta(k) \approx 1, \sin\delta\theta(k) \approx \delta\theta(k)$ 로 표현하여 정리하였다.

2.2.2 자이로스코프

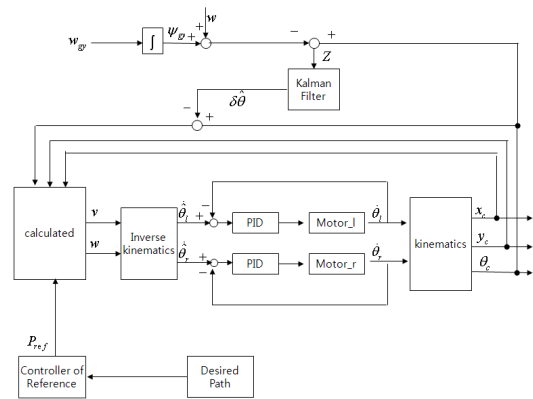
자이로스코프는 외란에 강인한 회전 측정 센서로써 각속도를 측정할 수 있다. 본 논문에서 사용된 자이로스코프는 MEMS형으로 내부의 신호 처리와 필터를 통해서 이러한 드리프트 및 온도, 양자화에 대한 오차를 보정하는 디지털 자이로센서로 시간당 10도 이하의 오차를 나타내므로 각각의 오차에 대한 성분을 무시할 정도로 매우 작다고 가정한다[4]. 따라서 자이로스코프의 오차 값을 구하면 다음과 같다.

$$\delta\psi(k+1) = \delta\psi(k) \quad (8)$$

3. 항법제어기 설계

3.1 위치제어기 설계

자율주행로봇의 위치를 제어하기 위해서 본 논문에서는 Quadratic Curve를 사용하였다[5].



<그림 2> 제어 시스템 블록도

그림2은 자율주행로봇의 제어시스템 블록도이다. 요구된 경로로부터 레퍼런스 포인트를 구하고, Kinematics를 이용하여 자율주행로봇의 위치와 자세를 구하여 레퍼런스 위치와의 오차로부터 Inverse kinematics를 이용하여 자율주행의 오른쪽과 왼쪽 바퀴의 각속도를 구한다. 모터의 구동속도 제어는 PID를 이용하였다.

3.2 칼만필터 설계

위에서 구한 센서오차 모델식으로부터 다음과 같은 상태 방정식을 구할 수 있다.

$$\begin{bmatrix} \delta x_{k+1} \\ \delta y_{k+1} \\ \delta \theta_{k+1} \\ \delta \psi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{U_{R'}^m(k) + U_{L'}^m(k)}{2} & 0 \\ 0 & 1 & -\sin\theta(k) \frac{U_{R'}^m(k) + U_{L'}^m(k)}{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} (a_{\max}(\Delta r)^2)/2 & 0 & 0 & 0 \\ 0 & (a_{\max}(\Delta r)^2)/2 & 0 & 0 \\ 0 & 0 & \Omega_{\max}\Delta t & 0 \\ 0 & 0 & 0 & \Omega_{\max}\Delta t \end{bmatrix} w_k \quad (9)$$

$$z_{k+1} = [0 \ 0 \ 1 \ -1] [\delta x_{k+1} \ \delta y_{k+1} \ \delta \theta_{k+1} \ \delta \psi_{k+1}]^T + v_k \quad (10)$$

여기서 a_{\max} , Ω_{\max} 는 각각 이동로봇의 최대가속도와 최대 각속도를 나타낸다.

4. 시뮬레이션 및 실험

4.1 시뮬레이션 방법 및 결과

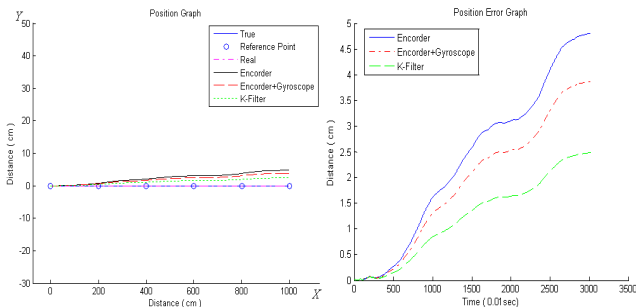
시뮬레이션은 MATLAB을 사용하였고, 표1의 위치 명령(x, y, θ)으로 시뮬레이션 하였다.

〈표 1〉 시뮬레이션의 주행위치 명령

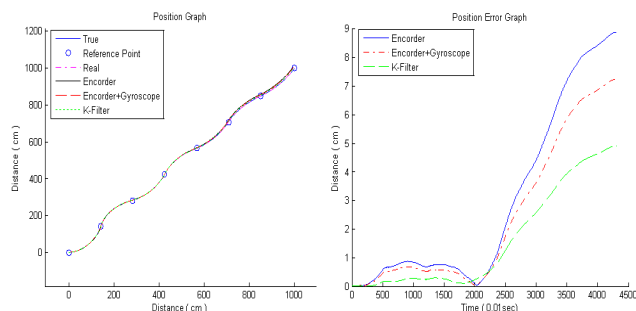
순서	초기점(cm)	목표점(cm)
①	(0, 0, 0)	(1000, 0, 0)
②	(0, 0, 0)	(1000, 1000, 45)

〈표 2〉 시뮬레이션의 목표점 주행위치 오차결과

순서	엔코더(cm)	엔코더+자이로(cm)	칼만필터(cm)
①	4.80	3.87	2.47
②	8.87	7.23	4.91



〈그림 3〉 ①시뮬레이션 위치 및 오차 그래프

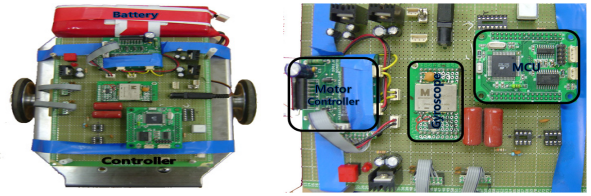


〈그림 4〉 ②시뮬레이션 위치 및 오차 그래프

위의 시뮬레이션 위치 오차의 결과는 표2와 같고, 그림3과 그림4는 주행 위치와 위치오차 결과 그래프를 나타낸다. 엔코더만 사용했을 경우 평균 오차는 6.83cm이고 엔코더와 자이로센서를 같이 사용한 경우의 평균오차는 5.55cm이다. 칼만필터를 사용한 경우의 평균오차는 3.69cm이다. 시뮬레이션 결과 칼만필터를 사용한 경우 위치 오차가 감소됨을 확인할 수 있었다.

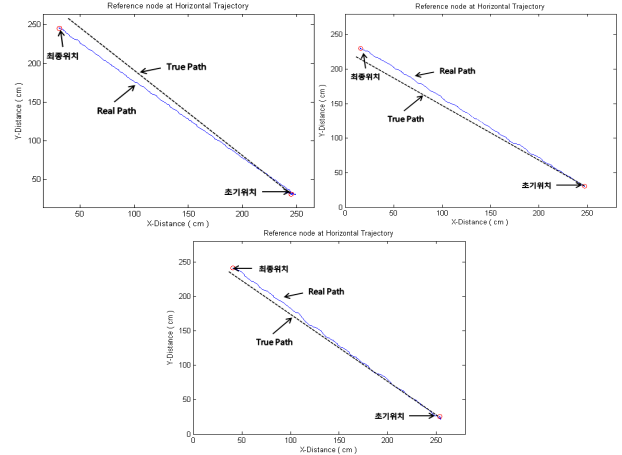
4.1 실험 방법 및 결과

실제 설계한 자율주행로봇과 제어기는 그림5와 같다.



〈그림 5〉 자율주행로봇과 제어기 구성

그림6은 엔코더만을 사용하였을때와 엔코더와 자이로센서를 같이 사용하였을때, 칼만필터를 사용하였을때의 주행 위치 그래프를 나타낸 것이다.



〈그림 6〉 3m 직선 주행 실험 결과

〈표 3〉 실험 결과

순서	위치오차	엔코더	엔코더+자이로	칼만필터
초기위치(X,Y):cm		(245.2, 31.13)	(247.1, 30.06)	(254.0, 25.34)
최종위치(X,Y):cm	TRUE	(17.76, 226.76)	(26.62, 233.51)	(39.15, 234.71)
	REAL	(30.8, 245.6)	(16.15, 230.2)	(40.91, 240.60)
위치오차(cm)		22.91	10.98	6.14
총 주행거리(cm)		303.26	305.60	302.89

실제 주행 실험 결과 표3에서도 보듯이 엔코더만을 사용했을때는 22.91cm의 오차가 발생하였고, 엔코더와 자이로를 사용한 결과에서는 10.98cm, 칼만필터를 사용한 경우는 6.41cm의 오차가 발생하였다. 칼만필터를 사용한 경우는 엔코더만 사용한 경우보다 3.7배, 엔코더와 자이로를 같이 사용한 경우는 1.7배의 오차가 줄어드는 효과 있었다.

3. 결론

본 논문에서는 차륜구동방식의 추측항법 시스템을 이용하는 자율주행로봇의 항법제어기를 설계하여 성능을 평가 하였다. 제한한 항법제어기는 칼만필터를 이용하여 자율주행로봇의 주행 누적 오차를 보정함으로써 자율주행로봇의 핵심장치인 항법제어기의 성능향상에 기여할 수 있으며, 또한 저비용 설계가 가능하므로 상용화에 유리하다. 또한 초음파, RF, 레이저, 카메라등의 센서 등과 결합하여 지능로봇용 위치 제어 시스템에 적용이 가능하다.

[참고 문헌]

[1] 한성현, Hideki Hashimoto, “로봇산업의 국내외 현황과 발전동향”, 한국공작기계학회지, 10(3), 27-37, 2001.
 [2] B. Billur, F. Hugh, Durrant-Whyte, “Inertial Navigation Systems for Mobile Robots”, IEEE Transactions on Robotics and Automation, 11, 328-342, 1995.
 [3] C. Hakyoung, L. Borenstein, “Sensor Fusion for Mobile Robot Dead-reckoning with a Precision-calibrated Fiber Optic Gyroscope”, Proc. 2001 IEEE International Conference on Robotics and Automation, 4, 3588-3593, 2001.
 [4] Microinfinity Co, www.minfinity.com, CruzCore R1001E Technical Manual.
 [5] K. Yoshizawa, H. Hashimoto, “Path tracking control of mobile robots using a quadratic curve”, Proc. IEEE Intelligent Vehicles Symposium, 58-63, 1996.