

이동 로봇 절대위치 추적 제어 시스템

최현승, 현웅근
호남대학교

Mobile robot absolute position tracking system

Hyun-Seung Choi, Woong-Keun Hyun
Honam University

Abstract - This paper represents a absolute position tracking system with sensor fusion and PD-gain. this paper presents an accurate localization method by relative and absolute sensor fusion and PD control for position tracking of mobile robot. we developed a sensor based absolution position tracking and smooth moving algorithm using this algorithm.

서와 Motor controller부, Encoder부는 주제어부와 USB 통신을 하며, 절대위치인식을 위한 StarGazer 등은 AVR2560 센서 제어부를 통한 RS232통신을 하게 된다. 주제어부는 Embedded Linux하에서 센서 리드, TCP/IP통신, Localization 연산, path planning 등을 각각 수행하는 다수 개의 process가 동시에 동작하게 된다.

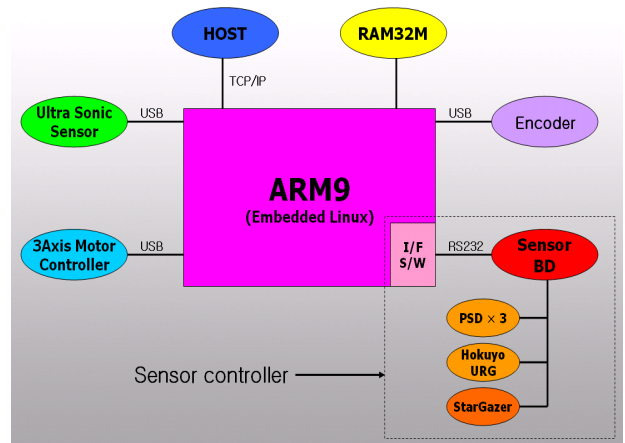
1. 서 론

최근 연구 개발이 활발히 이루어지고 있는 이동로봇은 다른 종류의 산업용 로봇에 비해 이동성과 조작성이 우수한 특성을 갖고 있을 뿐 아니라, 센서기술 및 컴퓨터 기술의 발달에 힘입어 제조업을 비롯한 각종 산업 현장에서 사람의 역할을 대신해 오고 있다. 가사도우미 로봇의 실내주행제어에 관한 연구는 위치추정, 경로계획, 경로추정, 장애물 회피 등이 있으며 모든 연구가 병행되어야 한다. 이러한 연구들은 가사도우미 로봇이 주행과 동시에 가사구성물이 내제된 환경 내에서 자신의 위치를 인지하는 연구가 기반이 되어야 한다. 본 연구에서는 가사도우미 로봇이 사전에 아무런 주위환경에 대한 정보를 갖고 있지 않은 상태에서 자신의 위치를 알고 자신이 가야할 목표점에 정확히 도달할 수 있는 알고리즘을 제안한다.

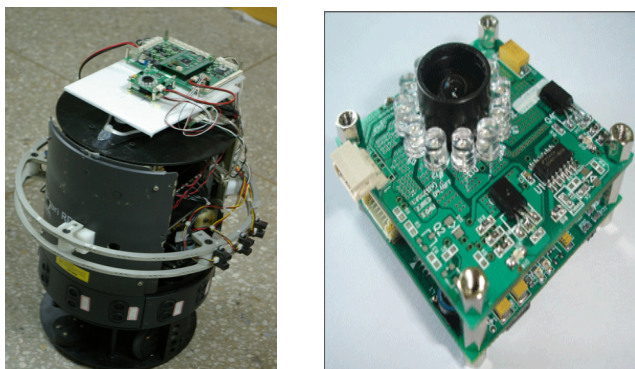
2. 본 론

2.1 로봇 시스템

<그림 1>은 본 논문 연구에 사용된 이동 로봇이다. 로봇은 이동부의 메인프로세서보드(32bit Intel StrongARM SA1110, 2067MHz, RAM 32MB, ROM 32MB)과 모션제어보드, 초음파센서 제어보드로 구성되어지며 메인프로세서는 PCMCIA BUS 방식의 Wireless Lan과 유선 랜 그리고 Serial 통신 및 USB 인터페이스를 갖추고 있고 Embedded Linux OS 운영체제를 사용한다.



<그림 2> 로봇 시스템 구성 블록다이어그램

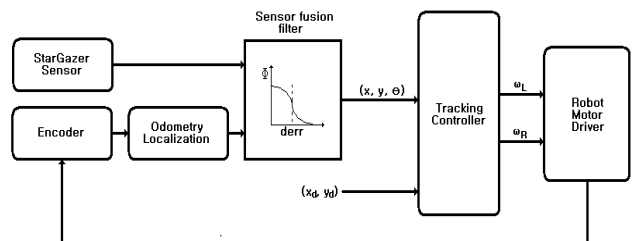


<그림 1> RD로봇의 실제 모습(좌)와 StarGazer Sensor(우)

<그림 2>은 본 논문 연구에서 사용된 Embedded Linux 기반의 로봇의 전체 시스템 구성을 표현하고 있다. 로봇의 위치 인식을 위해서 Encoder, 3Axis Motor Controller를 사용하고 AVR2560을 이용하여 StarGazer 센서의 초기 설정 및 데이터 전송을 한다. Encoder와 StarGazer 센서의 데이터를 검출하여 Linux OS에서 센서융합 알고리즘을 적용하여 이동로봇 정밀 Localization을 수행하게 된다. 로봇 주행을 모니터링 하는 PC와는 TCP/IP방식으로 통신을 하며, 통신시 로봇이 서버, 모니터링 PC가 클라이언트로써 데이터를 주고 받는다. 제어 OS는 범용 embedded Linux를 사용하였으며, time based multi tasking을 위하여 OS에서 제공하는 timer를 동작 시켰다. 로봇에 부착된 초음파 센

2.2 절대위치추적 알고리즘

Encoder(odometry)를 이용한 이동로봇의 절대위치 추적은 전체 주행 구간에 걸쳐 이동로봇의 위치를 연속적으로 측정할 수 있는 장점이 있다. 그러나 단점으로는 Encoder 환산계수 오차, 양 바퀴 사이의 거리 오차 등 규칙적인 오차와 고르지 못한 평면을 주행 하여 발생하는 환산계수 오차, 바퀴의 미끄러짐 때문에 생기는 오차 등 여러 가지 오차요인을 내포하고 있다. 실제 이동로봇의 주행 시 Odometry의 단점을 보완하기 위해 절대센서인 StarGazer를 사용 하였다. StarGazer는 천정에 적외선반을 반사하는 Landmark를 부착하며 Landmark의 범위 안에서 반사된 적외선을 카메라를 통해서 읽어 자신의 위치를 알아낼 수 있는 센서이다. 단점으로는 Landmark의 범위를 벗어나게 되면 위치를 알 수 없는 Dead-zone이 존재한다. 로봇의 Encoder 값과 함께 융합하여 Odometry의 누적오차를 보정하고 StarGazer의 Dead-Zone을 제거하여 더욱 정확한 위치를 측정할 수 있다.



<그림 3> 절대위치 추적 알고리즘의 블록다이어그램

<그림 3>은 절대위치 추적 알고리즘의 블록다이어그램으로 Odometry와 StarGazer를 이용하여 절대위치를 추적하는 순서를 간략하게 나타낸 것이다. Encoder 데이터에 다음과 같은 Odometry Localization 적용하여 현재 로봇의 좌표와 방향을 알 수 있다.

$$\Delta\theta_E = \frac{2\pi R}{ND}(\Delta E_R - \Delta E_L) \quad (1)$$

$$\Delta X_E = \frac{2\pi R}{N}(\Delta E_R + \Delta E_L)\cos\left(\frac{\theta_{(k)} + \theta_{(k+1)}}{2}\right) \quad (2)$$

$$\Delta Y_E = \frac{2\pi R}{N}(\Delta E_R + \Delta E_L)\sin\left(\frac{\theta_{(k)} + \theta_{(k+1)}}{2}\right) \quad (3)$$

Odometry Localization를 통하여 알게 된 로봇의 위치는 StarGazer와 Sensor fusion filter를 통하여 보다 정확하게 알아낼 수 있다. Sensor fusion filter는 다음과 같은 수식에 의해 구현 되었다.

$$X_a = (1 - \Phi)X_{SG} + \Phi \cdot X_E \quad (4)$$

$$\Phi = 1 - \frac{1}{1 + e^{-(perr + derr) \cdot k}} \quad (5)$$

$$perr = \sqrt{(X_{SG} - X_E)^2 + (Y_{SG} - Y_E)^2} \quad (6)$$

Sensor fusion filter를 통하여 알게 된 로봇의 좌표를 이용하여 User가 정해진 절대위치 좌표로 이동이 가능하게 된다. 하지만 일반적으로 단순히 좌표이동을 하게 되면 비교적 부자연스러운 움직임을 보이기 때문에 다음과 같이 Pgain과 Dgain을 이용하여 모터의 속도를 조절하게 되고 보다 자연스러운 움직임으로 목표점에 도달하게 된다.

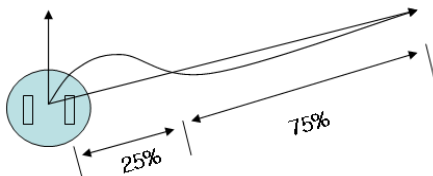
$$w_R = V_d \cdot r_1 + r_2(kr_p\theta_e + kr_d \cdot \dot{\theta}_e) \quad (7)$$

$$w_L = V_d \cdot r_1 - r_2(kr_p\theta_e + kr_d \cdot \dot{\theta}_e) \quad (8)$$

$$r_1 = 1 - \frac{1}{1 + e^{-(\delta l_i/L_0 - \delta L_0/2) \cdot kg_1}} \quad (9)$$

$$kg_1 = 20.0, \delta L_0 = 0.25 \quad \begin{matrix} w_L \leq |w_{L \max}| \\ w_R \leq |w_{R \max}| \end{matrix}$$

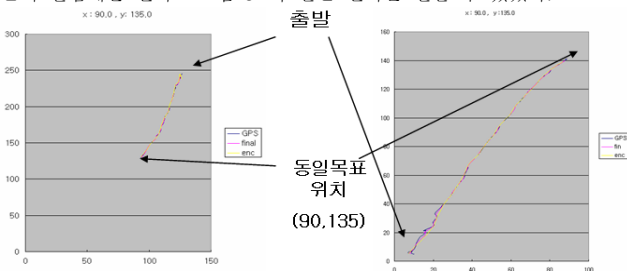
식(7)과 식(8)은 Sensor fusion filter를 통하여 보정된 로봇의 좌표와 목표점의 좌표를 이용하여 방향과 거리를 구하고 그 거리로 이동하기 위한 각 바퀴의 속도를 구하는 수식이다. 로봇은 현재 로봇의 위치에서 목표점까지의 tracking 구간 이동시 25% 이내는 방향각을 우선 제어하는 Rotating control factor이며 식(7)과 식(8)의 r₂ 텀에 의해 이루어지며 나머지 75%는 tracking 제어를 결정하는 Tracking control factor로 식(7)과 식(8)의 r₁ 텀에 의해 이루어지게 된다.



<그림 4> Tracking control factor

3. 결 론

위의 수식을 적용하여 절대위치 좌표로의 이동이 얼마나 잘 이루어지는지 실험해본 결과 <그림 5>와 같은 결과를 얻을 수 있었다.



<그림 5> 실험 결과

실험 결과 초반에는 목표점까지의 방향을 잡기 위해 좌표 변화가 조금 있었지만 후반에 들어서는 자연스럽게 움직인 것을 확인할 수 있다.

[참 고 문 헌]

- [1] Folkesson, J. and H.I. Christensen, " Tree graph based path planning in indoor space," ICRA, pp. 419-427, 2003
- [2] Nakashima and Tokomura, " A new robot path planner using topology map," Proc. of IEEE Intl. Conf. on robotics and automation, Vol. 1, pp704-710, 2002.
- [3] Attawith Sudsang and Jead Ponce, " A new approach to motion planning for disc-shaped robots," Proc. of IEEE Intl. Conf. on Robotics and Automation, Vol.2, pp1068-1075, 2000
- [4] J. Forsberg, U. L. L. and A. Wernersson, " Mobile robot navigation using range-weighted hough transform," IEEE robotics and Automation Magazine, pp18-26, march 1995
- [5] Latombe, Robot Motion planning, Kluwer Academic publishers
- [6] Sebastian Thrun and Wolfram Burgard, Probabilistic robotics, MIT press

 This research was supported by the program for the Training of Graduate Students in Regional Innovation and the program for Regional Strategic Industries which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government.