2006 年 8月碩士學位論文

전자나침반과 엔코더를 이용한 이동로봇의 위치추정

朝鮮大學校 大學院 制御計測工學科 金 光 振

전자나침반과 엔코더를 이용한 이동로봇의 위치추정

Pose Estimation of a Mobile Robot Using
Electric Compass and Encoder

2006 年 8 月 25日

朝鮮大學校 大學院 制御計測工學科 金 光 振

전자나침반과 엔코더를 이용한 이동로봇의 위치추정

指導教授 高 樂 溶

이 論文을 工學 碩士學位申請 論文으로 提出함.

2006年4月日

朝鮮大學校 大學院 制御計測工學科 金 光 振

金光振의 碩士學位論文을 認准함

委員長朝鮮大學校教授<u>金永炯</u>印
委 圓 朝鮮大學校教授<u>崔漢秀</u>印
委 圓 朝鮮大學校教授<u>高樂溶</u>印

2006年5月日

朝鮮大學校 大學院

목 차

List of titles

List of Tables ····· iv
List of Figures ······ v
ABSTRACT vii
제 1 장 서 론1
제 1 절 연구배경과 목적1
제 2 절 연구 방향4
제 3 절 논문의 구성
제 2 장 최근까지 연구된 위치 추정법과 사용된 센서 분석
제 1 절 이동 로봇에 위치 추정이 필요한 이유6
제 2 절 최근까지 위치 추정에 관한 연구들7
제 3 절 기존의 위치 추정법의 장 ● 단점9
제 4 절 위치 추정에 사용된 센서 및 이동 로봇 시스템11
1. 전자나침반 센서11
가. 전자나침반의 구조 및 데이터 획득11
(1) 전자나침반 구조11
(2) 전자나침반의 데이터 획득13
9 자유 주해 근보 시스텐 : PIONFFR 914

가. PIONEER 2의 성능 특징 ······14
(1) 하드웨어 구조14
(2) 프로그램 흐름도16
(3) 환경 설정17
(4) Update20
나. PIONEER 2의 통신 프로그램 ······21
(1) Packet Protocol
(2) 클라이언트 명령코드23
제 3 장 이동로봇의 위치 추정27
제 1 절 이동로봇 센서 시스템 구성27
1. 이동 로봇 초기 환경27
2. 이동 로봇 센서 시스템29
가. 엔코더 데이터29
나. 전자나침반 센서 데이터32
제 2 절 엔코더 데이터를 이용한 위치 추정34
1. 위치 추정 알고리즘의 개요34
가. 로봇 위치 추정34
나. 이산 칼만 필터34
다. 확장 칼만 필터38
2. 이동로봇의 위치추정의 세부사항41
가. 이동 로봇 시스템41
(1) 로봇 이동 위치의 예측43
(2) 실제 이동거리에 대한 분석45
(가) 엔코더 데이터 이동 위치 오차45
제 3 절 전자나침반을 이용하여 로봇의 위치오차 보정48

제	4	장	실	험 및	! 고	찰 …	• • • • • • • • • • • • • • • • • • • •	•••••	•••••	• • • • • • • •	•••••	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	•••••	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	•••••	•••••	49
	제	1	절 -	위치	추정	실현]					•••••	•••••					•••••	•••••		49
	제	2	절 .	고 칠	<u> </u>					•••••						•••••			•••••	•••••	55
제	5	장	결	론						•••••		•••••				•••••			•••••	•••••	56
참	고든	무허								•••••											57

표 목 차

List of Tables

Table	1.	Position estimation research features until now	. 8
Table	2.	Register of CMPS03 ·····	15
Table	3.	Pioneer 2 Packet Protocol ·····	21
Table	4.	Server Information Packet ·····	23
Table	5.	Client Command Packet ····	24
Table	6.	Client Command Set ·····	26

도 목 차

List of Figures

Fig. 1. Requirement of position estimation
Fig. 2. Electric compass Module
Fig. 3. Electric compass Module
Fig. 4. Electric compass Data
Fig. 5. Pioneer 2's physical dimensions and sonar array
Fig. 6. Client-server control architecture
Fig. 7. Direct Mode and Client Mode
Fig. 8. Pioneer 2 Reconfiguration ————————————————————————————————————
Fig. 9. updating Pioneer 2
Fig. 10. Geometric robot moving system ————————————————————————————————————
Fig. 11. Robot structure
Fig. 12. Range data algorism
Fig. 13. Angle data algorism ————————————————————————————————————
Fig. 14. Data of Encoder and Electric compass
Fig. 15. Kalman filter cycle ————————————————————————————————————
Fig. 16. Inertial Navigation System ————————————————————————————————————
Fig. 17. encoder data ——————————————————————————————————
Fig. 18. encoder data graph ————————————————————————————————————
Fig. 19. encoder data
Fig. 20. encoder data graph ————————————————————————————————————

Fig.	21.	Pioneer 2 ——————————————————————————————————
Fig.	22.	Trajectory of Circle Navigation
Fig.	23.	Distance error of Circle Navigation
Fig.	24.	Trajectory of Cross Circle Navigation
Fig.	25.	Distance error of Cross Circle Navigation
Fig.	26.	Trajectory of Straight Navigation ————————————————————————————————————
Fig.	27.	Distance error of Straight Navigation
Fig.	28.	Trajectory of Autonomous Navigation53
Fig.	29.	Distance error of Autonomous Navigation53
Fig.	30.	Trajectory of Autonomous Navigation
Fig.	31.	Distance error of Autonomous Navigation54

ABSTRACT

Pose Estimation of a Mobile Robot Using

Electric Compass and Encoder

Kim, Kwang Jin

Advisor: Prof. Nak Yong Ko, Ph. D.

Dept. of Control and Instrumentation Eng.

Graduate School of Chosun University

This paper propose a method to estimate the pose of an indoor mobile robot.

The method uses the information from electric compass and wheel encoder. Usually

the dead reckoning method using only the encoder information can't avoid the error

due to slip, wheel diameter error, and uneven floor. So the proposed method uses

electric compass to compensate the pose error.

The method regards the heading angle from the electric encoder more reliable

than the heading angle from the dead reckoning. The heading angle from the

electric compass substitutes the dead reckoned heading angle, and is used to

calibrate the dead reckoned position data. This method is tested and evaluated

through simulation and experiments.

- vii -

제1장. 서 론

제1절. 연구 배경과 목적

현재 많은 산업현장에 자율 이동 로봇(autonomous mobile robot)과,자율 운반체(autonomous guided robot)등이 도입되어 왔다. 대부분의 이동 로봇(mobile robot)은 와이어 가이드 시스템(wire guided system),라인 가이드시스템(line guided system),그리고 표지판 시스템(landmark system)등의 방법으로 경로를 유도하는 방식이다^[1].이러한 이동 로봇 시스템은 로봇의 작업 영역 내에 와이어나 특수한 물질을 사용하여 표시해 놓은 라인 및 표지와같은 경로 인식 보조 수단을 사용하는 형태이다.이러한 경로 유도 방식에서는 작업 과정 변경시마다 이동 로봇의 경로를 새롭게 지시해야하고,유도하는 인식 수단의 손상 시에는 로봇의 경로를 이탈하거나 시스템에 치명적인손상을 줄 수 있다.따라서 이동 로봇의 별도의 경로 안내 수단 없이 작업공간에서 자유자재의 이동이 가능한 이동 로봇의 유연한 탐색 주행(flexible navigation)을 위한 연구개발이 이루어 지고 있다.

자율 이동로봇은 다음과 같은 필수적인 기술이 요구된다. 첫째, 로봇에 주어지는 작업 환경과 자율 주행 성능을 보이기에 적합한 이동 로봇 시스템의 하드웨어 구성 기술 및 외부 환경 인식(environment recognition)을 위한센서 구성 기술이다. 둘째, 실시간으로 작업 환경에 존재하는 장애물 회피(obstacle avoidance)기능이 있어야 한다. 셋째, 주어진 작업을 성공적으로 완료하기 위하여 작업 공간에서 가장 효율적인 이동 경로를 선택하여 주행하는경로 계획(path planning) 기술이다. 이 경로 계획의 문제를 대략적으로 분류하면, 작업의 시작 지점에서부터 작업을 최종적으로 끝마칠 때 까지 주행하여야할 전역 경로 설정 문제와, 이미 정해진 전역 경로를 따라 이동하는 과정에서 만날 수 있는 장애물에 대한 충돌 회피와 kidnapping 문제 발생 시

가능한 빠르게 에러 값을 보정 하여 기존의 상태로 복귀할 수 있도록 하는 지역 경로 계획 문제 등의 두 가지로 구별된다. 마지막으로 작업 과정 전반에 걸쳐 이동 로봇의 상시 위치 파악이 가능하도록 하는 위치 추정(position estimation) 기술이다. 이동 로봇이 시작점에서 출발하여 목표점까지 이동하면서, 주어진 경로로부터 어느 위치에서, 어떤 방향으로, 얼마만큼 벗어난 상태인지 로봇 스스로가 판단하는 것이 매우 중요하기 때문에, 로봇의 위치를 추정하기 위한 많은 방법들이 제안되고 있고, 실제로 많이 사용되고 있다.

위치 추정은 크게 상대 위치 추정과 절대 위치 추정으로 나뉘어 진다. 상대 위치추정은 기준 위치로부터 주어진 시간 동안 이동한 거리, 속도, 및 각속도를 계산하여 현재의 위치를 추정하는 방법이 있고, 절대 위치 추정 방 법은 알고 있는 기준 위치들로부터의 거리 혹은 각도를 측정하여 기하학적으 로 위치를 결정하는 방법으로 초음파 센서, 비전 센서 그리고 레이저 레인지 파인더와 같은 센서를 이용하여 거리를 측정한다^[1].

이동 로봇이 계획한 전역 경로내의 이동 경로상의 위치를 파악하는 능력을 갖도록 하기 위해서는 로봇 자체의 상대적인 위치의 인식과 이동 경로상의 외부 환경에 대한 감지 영역이 가능한 광범위한 영역에 대하여 정확한정보 획득 및 신속성이 요구된다. 또한, 센서 별로 고유의 검출 특성을 가짐에 따라 인식할 수 없는 환경 범위나 물리량을 갖기 때문에 이에 대한 센서의 융합 대책도 필요하다. 로봇의 현재 위치 판단 성능은 로봇이 수행하는 작업의 완성도를 높이는데 필수적이며, 장애물 회피나 이동 경로 계획 등의선행 자료로 위치 추정이 필요하며, 센서 정보의 불규칙한 정도, 실시간 검출성능 등에 따라 위치 추정의 성능이 결정된다. 또한 환경 지도가 미리 제공되는 경우에는 센서 감지 능력을 이용해 로봇의 자기 위치를 스스로 확인하고, 현재 위치에서부터 바람직한 목표점까지 경로와 주행을 계획할 수 있다. 그러나, 지도가 미리 로봇에게 준비되지 않았다면 로봇이 추측 항법(dead-reckoning)과 센서에 의한 지리학적 특징에 의해 로봇의 위치를 확인하고 로봇 스스로 경로상의 목표점까지 경로와 주행을 계획할 수 있어야 한

다.

이동 로봇의 위치 추정 문제에서 고려할 수 있는 센서 시스템은 위치 추정의 방법에 적합하게 선정되어야만 한다. 현재 사용되는 위치 추정 방법 은 크게 4가지 방법으로 분류된다. 첫째로는 상대 위치를 추정하기 위한 주 행 기록기와 추측 항법(dead-reckoning)이 있다. 이 방법은 대표적인 상대 위 치 결정 방법으로 이동한 거리, 속도, 각속도를 계산하여 현재 위치를 추정한 다. 일반적으로 구동 바퀴 부분에 휠 엔코더나 자이로스코프, 가속도계 등과 같은 센서를 부착하여 일정 지점에서 경로를 따라 로봇이 이동한 경로의 위 치를 추정한다. 두 번째 방법은 동작하는 신호원(active beacon)을 이용한 방 법이다. 이 방법은 특정 위치에 초음파의 발신기나 적외선 센서의 발광 부분 을 부착하고 로봇에 초음파의 수신기나 적외선 센서의 수광 부분을 부착하여 위치를 추정하는 방법이다. 세 번째로 외부 환경에 일정의 표식을 설치하여 이것을 인식하는 경계표 인식(landmarks) 방법이 있다. 이 방법은 비전 센서 등을 이용하여 경계표를 인식하고, 이 경계표의 위치를 비교하여 위치를 추 정하는 방법이다. 마지막으로, 지도를 기본으로 하는 지도 매칭 방법(map matching)이 포함된다. 지도 매칭 방법은 외부 환경에 인식 표식을 인위적으 로 설치하지 않고, 구조화가 잘된 외부 환경 즉, 평면, 코너 그리고 모서리 등의 환경을 감지하여 위치를 추정하는 방법이다[1~5].

본 논문에서는 연구하고자 하는 방향은 전자나침반을 추가로 이용하여 로봇의 주행 경로에 따라 최대한 오차를 보정하여 상대적인 정확한 위치 추 정을 하여 이동 로봇이 별도의 경로 안내 수단 없이 작업공간에서 자유자재 의 이동이 가능하고 유연한 탐색 주행을 할 수 있도록 정확한 로봇의 위치를 추정하기 위한 방법을 제안한다.

제2절. 연구 방향

위치 추정에 사용된 센서의 종류를 보면, 내부용 센서는 로봇이 운동함에 따라 변화된 위치 및 방향에 관한 물리량을 측정하는 센서로서, 엔코더 (encoder), 포텐셔미터(potentiometer) 등의 회전 변위 센서, 전자 나침반과 같은 방위 센서, 그리고 자이로스코프(gyroscope), 가속도계 등의 관성 센서로 분류된다. 내부용 센서는 이동 로봇의 출발점으로부터 위치 및 방향 정보를 쉽게 얻을 수 있는 장점을 가진 반면, 장시간 주행시 오차가 누적되는 단점을 보완해야만 한다. 일반적인 휠 엔코더 등의 주행 기록계 센서는 기본적으로 사용되고, 외부용 센서는 이동 로봇의 주위 환경에 대한 절대 좌표계 내에서 로봇의 위치나 형상을 알아내기 위해 사용된다. 외부용 센서는 비전 센서, 초음과 센서, 레이저 레인지 파인더 등을 사용한다[78].

최근에는 비전 센서를 이용하여, 복도나 이동 경로 상에 특정 표식을 찾아서 위치를 추정하는 방법이 많이 연구되고 있으나, 스테레오 비전을 환경인식과 거리 측정에 이용하기 위해서는 많은 데이터의 양과 처리 속도문제를 해결해야 하는 단점이 있다^[9].

초음파 센서는 센서 자체가 가지는 큰 개구각(opening angle)과 반사와 온도변화 등에 영향을 받기 쉽고, 획득 시간이 길다는 약점을 가지고 있지만, 이동 로봇의 장애물 회피, 경로 계획 등에 쉽게 적용할 수 있다는 장점은 있 다.

레이저 레인지 파인더는 360°를 회전하면서 환경 테이터를 빠르게 얻을 수 있고, 충분히 환경 특징을 피쳐 드로잉 할 수 있는 스캔 테이터를 제공하지만. 가격이 비싸다.

연구 배경과 목적에서 기술한 바와 같이 본 논문에서 전자나침반 센서를 추가로 사용하여 엔코더만 사용하는 방법과 달리, 보다 정확한 이동로봇의 위치를 추정하는 방법을 제안한다

제3절. 논문의 구성

본 논문의 구성은 다음과 같다. 제2장에서는 최근까지 연구된 위치 추정 방법과 센서들에 대해서 알아보고, 본 연구에서 사용된 전자나침반 센서 시스템과 데이터 획득 알고리즘 및 PIONEER 2 로봇 시스템에 대해 설명한다. 제3장에서는 실제 전자나침반 센서 데이터와 엔코더 데이터를 결합하여두 개의 획득된 데이터를 가지고 위치 추정을 통해 에러 값을 보정하여 이동로봇의 정확한 위치추정 방법을 설명한다. 제4장에서는 위치추정을 수행하는여러 가지 실험을 통하여 보다 정확한 이동로봇의 위치추정을 증명할 수 있도록 한다. 그리고 마지막으로 제5장에서는 전체 논문에 대한 요약과 적용범위, 향후 연구 분야 및 보완 사항에 대해 설명하고자 한다.

제2장. 최근까지 연구된 위치 추정 방법과 센서

제1절. 이동 로봇에 위치 추정이 필요한 이유

이동 로봇에서 위치 추정은 목표점까지의 이동 경로 계획, 이동하면서 필요한 외부 환경 인식, 이동 경로 상에 주어진 장애물 회피 등과 관련하여, 지능적 행동을 위한 필수적인 항목이다. 위치 추정은 다른 기능들을 수행하기위해 기본적으로 요구된다.

이동 로봇이 시작점에서 출발하여 목표점까지 이동하면서, 장애물이나 기타 외부 환경에 의해서 주어진 경로를 어느 정도의 위치에서, 어떤 방향으로, 얼마만큼, 벗어난 상태인지를 로봇 스스로가 판단하고, 현재 위치를 보정 해나가기 위해 위치 추정은 필수적이다.

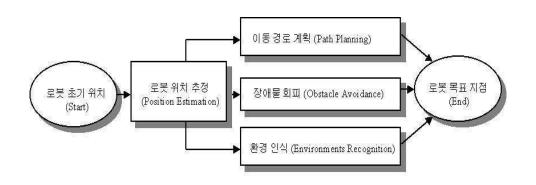


Fig 1. Requirement of position estimation

제2절. 최근까지 위치 추정에 관한 연구들

최근까지 위치 추정과 관련한 연구들을 표 1 과 같이 구성하였다. 구성은 크게 위치 추정 방법과 사용된 센서들의 두 종류로 나눌 수 있다. 위치 추정 분야에서 우리는 크게 odometry 이용 방법, 경계표 인식, 지도 매칭, 기타 방법으로 구별했다. odometry 이용 방법은 엔코더, 자이로 등 내부 주행기록계 등을 이용하여 이동 로봇이 상대적으로 이동한 거리를 계산하여 위치 추정을 하는 방법이고, 경계표 인식 방법은 비전 센서를 많이 사용하는 방법으로, 어떠한 인공 또는 자연적인 표식을 센서로 입력받아서 위치를 추정하는 방법으로 실시간 데이터 처리량이 상대적으로 많으므로 빠른 데이터 처리를 필요로 한다. 지도 매칭 방법은 목표물의 데이터를 위치 추정전에 미리 데이터 베이스화 한 다음, 레이저 레인지 파인더와 같은 센서를 이용하여 입력받은 데이터를 서로 일치하여 위치를 추정하는 방법이다. 그리고 기타의 방법들이 있다.

표 1 안의 센서 부분에서 레이저 레인지 파인더는 라이터 액티브 비전을 갖춘 레인지 센서뿐만 아니라 적외선 레인지 센서를 포함한다. 대조적으로 비전 센서는 카메라 모션을 갖춘 스테레오 비전과 사이트 비전을 갖추고 있다. 비전 센서는 구조화된 빛을 이용하지 않고 단지, 비주얼 사이트에 대해이미지 처리로 이용한다. 그리고 US(Ultrasonic)는 초음파 센서 및 초음파 어레이 센서를 포함한다. 초음파 센서 이용의 초기 단계에서는 초음파의 거울형 반사의 영향은 고려하지 않는다. 그러나 초음파 센서를 이용한 방법은 초음파 센서의 느린 속도에 의해서 실시간 처리 능력을 감소시킨다. 현재 상업용으로 판매되는 대부분의 이동 로봇에는 디폴트 센서로 초음파 센서가 장착되어 있다.

	Localization(position estimation)						Sensors							
ref papers	Odo- metry	Land- marks	Markov process	Map matching	Active beacon	Etc method	Encoder & Gyro	US	Vision	Laser finder	Sensor fusion	Etc sensor		
STOS97						0		0	0					
MAFM97						0	0•							
Einsele97				0						0				
KH97				0						0				
OMB97		0					0•				0			
Thope97		0							0					
MKS97		0				0			0					
YEY97		0							0					
RLK97		0				0			0					
TVG97		0							0					
VPT98				0							0			
GL98				0					0					
BDFC98			0					0				0		
GBFK98			0					0		0				
FLS98				0				0						
TZH98				0					0					
GMR98		0					0•					0		
LD98		0							0					
SD98		0		-					0					
AR98		0							0					
LLM98 MMP98		0					0•		0					
LKH99	-			0				0						
DDPC99	0								0					
KIB99						0			0					
HH99				0		<u> </u>				0				
ARR99	1	0							0	_				
GRS99	0					0	0•					0		
YMOY99	0	0						0						

Table 1. Position estimation research features until now

제3절. 기존의 위치 추정 방법과 장 · 단점

상대 위치 추정(relative localization) 이란 로봇이 출발점으로부터의 자신 의 위치를 측정함으로써 얼마만큼 이동했는가를 예측하는 방법이다. 두 가지 상대 위치 추정방법이 지배적인데, 그 중 첫 번째는 외부 센서를 사용하는 방법이며, 두 번째는 오도메트리(odometry) 정보에 필터를 적용 하는 방법이 다. 이렇게 필터를 적용하는 방법을 데드레크닝(dead-reckoning) 이라고 한 다. 이 방법은 외부의 기준 환경의 제공이 없더라도 로봇 스스로 얼마나 이 동했는지 계산하여 빠르게 자신의 위치를 알 수 있다는 장점이 있다. 반면 단점으로는 바퀴의 미끄러짐이나, 바퀴크기의 차이, 주행환경의 불확실성으로 외란을 극복해야하는 단점이 있다. 예를 들어 로봇이 이동 중에 바퀴의 미끄 러짐이 발생하면, 로봇은 여전히 움직이지만 그 움직임이 기록되지는 않는다. 이것이 오차의 원인이 된다. 이러한 오차는 로봇이 움직임에 따라 지속적으 로 누적이 될 수 있다. 본 논문에서는 로봇이 이동 중 바퀴가 허공에 뜨게 되는 kidnapping 발생 시 엔코더 사용만으로는 로봇의 이동 거리를 정확히 알 수 없으므로 관성센서를 추가로 도입하여 로봇의 이동 각도와 가속도를 측정하여 이동 거리를 구해 보다 정확한 로봇의 위치를 알 수 있도록 연구 하다.

절대 위치 추정 방법은 알고 있는 기준 위치들로부터의 거리 혹은 각도를 측정하여 기하학적으로 위치를 결정하는 방법이다. 절대 위치 추정법도 크게 두 가지로 나눌 수 있다. 먼저 비전센서를 이용하여 작업 환경에서 기준 표식들의 데이터를 입력받아 로봇의 위치를 측정한다. 비선 시스템 사용의 장점으로는 움직이는 표식까지 감지하고 분석하여 위치를 측정할 수 있다. 반면 단점으로는 미리 표식 모델이나 특정량 등을 모델링하고 주행 중영상 내에서 인공 표식을 찾아낸다. 그러나 이와 같이 특정의 형태 정보를이용하는 방식의 경우에는 영상 처리 결과에 매우 의존적이기 때문에, 영상내의 잡음(noise)이나 부정확한 초점에 따른 번점 효과(blurring phenomenon)

등에 의해 낮은 성공률을 보이게 한다. 또한 많은 영상 데이터의 양과 처리속도 등에 따른 문제점이 생기기도 한다. 다른 방법으로는 지도 매칭 방법이 있다. 지도에 의한 위치 추정의 주된 장점은 환경을 재구성하는 것 없이 위치 정보를 유도하기 위해 대표적인 실내의 환경의 본래 구성된 구조물을 사용하고, 외부 환경의 갱신된 지도를 생성한다는 점이다. 외부 환경의 지도들은 또 다른 이동 로봇의 포괄적인 경로 계획이나 어떤 지역 장애물 회피 방법에서 지역 최소한의 장애물의 회피 등을 위해 중요하다. 또한 지도를 근거로 하는 위치 추정 방법은 로봇이 새로운 환경을 학습할 수 있도록 제공하고, 탐색을 통하여 위치의 정확성을 향상시킨다. 반면, 지도에 의한 위치 추정의 단점은 만족스러운 탐색을 위한 특성화된 필수 조건은 외부 환경이 일치화단계에서 필요로 하는 쉽게 구별할 수 있는 형태로 충분히 고정된 것들이 있어야 하며, 센서 지도는 유용성을 위해 충분한 정확성이 필요 되어진다. 또한충분히 정확성 있는 환경 감지와 많은 데이터를 빠르게 처리 할 수 있는 시스템이 사용되어야 한다는 단점이 있다.

제4절. 위치 추정에 사용된 센서 및 이동 로봇 시스템

1. 전자나침반 센서

가. 전자나침반 구조 및 데이터 획득

본 논문에 사용된 전자나침반 센서는 CMPS03 자계 나침반 모듈을 사용한 센서로서 지자계 값을 전압값으로 환산하여 이 값에 의해 로봇의 회전량을 측정한다. 전자나침반 센서의 선능은 $0\sim359$ 도 값으로 전달된다. 분해능은 0.1도 단위이고 절대정밀도는 3도 이다.

(1) 전자나침반의 구조

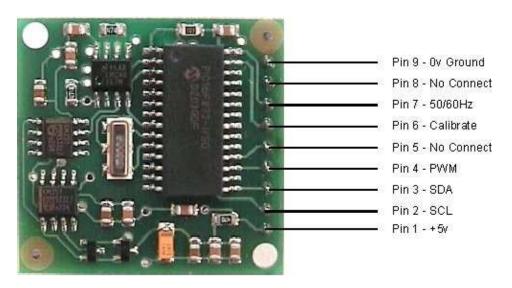


Fig 2. Electric compass Module

그림2는 전자나침반 센서에 장착된 CMPS03 나침반 모듈이다.

Register	Function
0	Software Revision Number
1	Compass Bearing as a byte, i.e. 0-255 for a full circle
2,3	Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees.
4,5	Internal Test - Sensor1 difference signal - 16 bit signed word
6,7	Internal Test - Sensor2 difference signal - 16 bit signed word
8,9	Internal Test - Calibration value 1 - 16 bit signed word
10,11	Internal Test - Calibration value 2 - 16 bit signed word
12	Unused - Read as Zero
13	Unused - Read as Zero
14	Unused - Read as Undefined
15	Calibrate Command - Write 255 to perform calibration step. See text.

Table 2. Register of CMPS03

표2는 CMPS03 전자나침반 모듈의 레지스터이다.



Fig 3. Electric compass Sensor

그림3은 실제 본 논문에 사용된 전자나침반 센서이다.

(2) 전자나침반의 데이터 획득

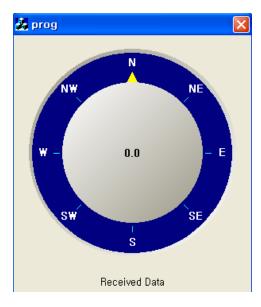


Fig 4. Electric compass Data

그림4는 전자나침반 센서로부터 데이터를 획득하는 모습이다. 시리얼 통 신을 통하여 0.1도 단위로 데이터를 컴퓨터에서 받을 수 있다.

다시 한번 전체적인 전자나침반의 성능을 정리하자면 다음과 같다.

CMPS03 Electric Compass Sensor.

Voltage - 5v only required

Current - 20mA Typ.

Resolution - 0.1 Degree

Accuracy - 3-4 degrees approx. after calibration

Output 1 - Timing Pulse 1mS to 37mS in 0.1mS increments

Output 2 - I2C Interface, 0-255 and 0-3599

SCL speed up to 1MHz

Small Size - 32mm x 35mm

Low Cost - Best Price Compass Module Available

2. 자율 주행 로봇 시스템 : PIONEER 2

가. PIONEER 2의 성능 및 특징

본 논문에 사용된 ACTIVE MEDIA(社)의 이동로봇 Pioneer 2 는 산업용 컴퓨터와 초음파 그리고 접촉식 감지 모듈을 포함한 통합된 자율 이동 로봇시스템이다. 이 제어 시스템은 센서와 모터제어 뿐만 아니라 통신까지도 수행한다. Pioneer 2 의 마이크로컨트롤러는 32K FLASH-ROM과 32K DRAM이 내장된 20MHz Simens 88C166 마이크로프로세서에 의해 제어되며 사용자의 작업환경이나 목적에 따라 내부에 설치된 마이크로컨트롤러로 제어하거나 라디오 모뎀 또는 블루투스를 이용한 무선 중앙제어 시스템도 가능하다.

(1). 하드웨어 구조

본 논문에서 사용된 자율 주행 로봇 시스템인 Pioneer 2의 외형과 구조는 다음 그림(그림5) 와 같다.

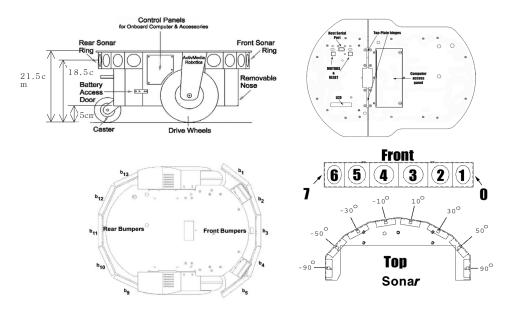


Fig 5. Pioneer 2's physical dimensions and sonar array

DC 12V 배터리 두 개를 병렬로 연결하여 12V의 전원으로 전체 시스템이 동작한다. 두 개의 DC모터를 사용하고 한 개의 보조 바퀴가 부착된 형태이다. 센서는 그림의 중심점(center point)에 부착되도록 설계되었다. 양쪽 바퀴의 직경은 195mm이며 외부 디바이스와의 통신을 위해 RS232 시리얼 통신포트가 두 개 장착되어 있다. 한 개의 시리얼 포트는 내부적인 통신을 수행하고, 나머지 한 개는 외부에 장착된 노트북 컴퓨터와 연결하였다.

Pioneer2의 내부 블록 다이어그램은 다음 그림(그림6)에 나타내었다.

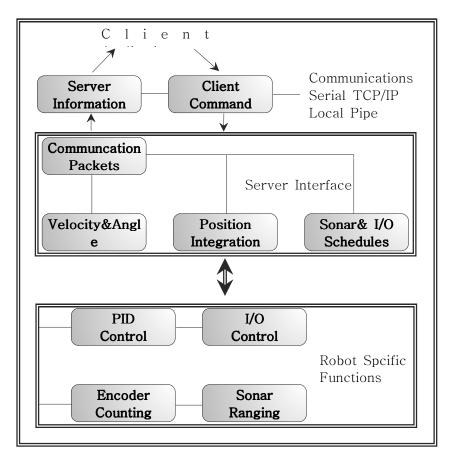


Fig 6. Client-server control architecture

(2). 프로그램 흐름도

Pioneer 2의 제어 프로그램 방법에는 직접 제어 모드와 클라이언트 제어 모드로 두 가지로 구분된다.

직접 제어 모드는 사용자 프로그램이 직접적으로 로봇 데몬(daemon)과 통신하여 로봇 제어용 프로그램으로 사용하는 방법이다. 이 프로그램은 로봇 이 지속적으로 외부에서 데이터를 받아서 이것들을 실행하고, 데이터를 다시 전송하는 방법으로 동작한다.

또한, 클라이언트 제어 모드는 사용자 프로그램의 명령어를 서버가 받아서 서버가 로봇 데몬에 명령어를 전송하는 방법으로 사용자 프로그램을 가상로봇의 시뮬레이터에서 테스트나 디버깅을 할 수 있고, 정확한 프로그램일때 실제 로봇에 사용하는 방법이다.

Pioneer 2의 프로그램 모드에 관한 내용을 다음 그림(그림7)에 나타내었다.

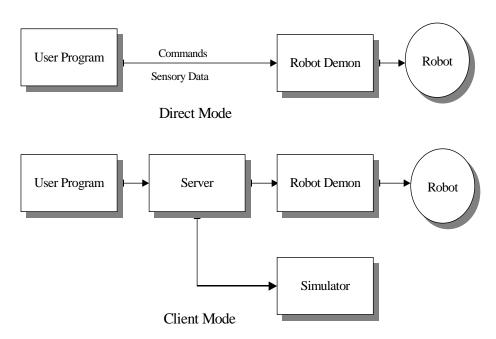


Fig 7. Direct Mode and Client Mode

(3). 환경 설정

- ※ 부트모드 설정
- ① FLASH 스위치를 로봇의 드라이버를 이용해 전진방향으로 위치시킨다.
- ② MOTOR 버튼을 ON 상태로 유지시키면서 RESET버튼을 ON→OFF 한다.
- ③ 마지막으로 MOTOR번튼을 3초이상 ON상태로 유지한 후에 OFF 한다.
- ※ p2os파일을 C:∖에 복사한다.
- ※ MS-Dos를 실행하고 CD명령을 사용해 p2os가 있는 디렉토리로 이동한다.
- ※ p2oscf.exe파일 실행 (실제 실행화면은 다음 그림(그림8) 과 같다.)

c:\>cd p2os			
c:\p2os>p2oscf.exe			
Opening serial port CO	M1		
Sending 187 byte boot	loader		
Monitor is 1765 bytes			
Monitor loaded!			
% Serial port, boot load	der and memory	bus OK.	
Retrieving parameters t	from flash ROM		Pioneer
========	P2OS	Constants	P2PP
=======================================	<u> </u>		<i>BDBB1533</i>
that you shouldn't to	uch		0
Type			360
Subtype			2200
Serial Number			360
FourMotors	0=no,	1=yes	4000
			500
RotVelTop deg/sec			128
TransVelTopmm/sec			
RotAccTop		deg/sec/sec	
TransAccTop		mm/sec/sec	
			CURRENT

Pw mMax	0-500	pw m	counts	VALUES
				Boo-Ki_1533
Encoder counts	/mm			0
========	= <u>1</u>	P20S	Variables	2
=========	=====			0
KEYWORD				0
Name				1
SInfoCycle 0=1	00ms, 1=50n	ns		1
HostBaud 0=96	00, 1=19200,	2=38400		0
AuxBaud 0=960	00, 1=19200,	2=38400		110
HasGripper 0=n	one, 1=user	io 2=genio		2000
FrontSonars 0=	none, 1=ins	talled		33500
RearSonars 0=r				0
AddedSonars 0			sonars	400
Low Battery 1/1	0 volt incre	ements		100
WatchDog ms				0
Rev Count				0
P2Mpacs 0=clas				· ·
StallVal 0-500,	-			0
StallCount 10m				100
Compass 0=non				500
CompX, compas				50
CompY, compas		fset		100
RotVelMax deg				30
TransVelMax n				60
RotAcc deg/sec				0
RotDecel deg/s	ec/sec			100
RotKp				200
RotKv				40
RotKi	, ,			80
TransAcc mm/		0		
TransDecel mm		600		
TransKp TransKv				125

```
TransKi
IovVelMax mm/sec
JoyRV elMax deg/sec
====== P2OS Config Commands ===========
Type: keyword alone to view current value
     keyword and new value to change it
    'c' or 'constants' to view P20S constants
    'v' or 'variables' for current variables
    'a' or 'arm' to for current arm values
    'r' or 'restore' to restore original values
    'q' or 'quit' to exit *without* saving
    'save' to save changes and exit
    '?' or 'help' to see this menu again
command> HostBaud 0
HostBaud is 9600.
command> save
Comparing params with board FLASH. One moment, please...
Saving changes to flash...% Flash bank 3 cleared
% Flash bank 3 erased.
Writing parameters....
Saved to flash...
command> q
Comparing params with board FLASH. One moment, please...
Press <Enter> to quit...
C:\p2os>
```

Fig 8. Pioneer 2 Reconfiguration

- (4). Update
- ※ 부트모드 설정
- ※ p2os파일을 C:∖에 복사한다.
- ※ MS-Dos를 실행하고 CD명령을 사용해 p2os가 있는 디렉토리로 이동한다.
- ※ C:\p2os>p2osdl.exe p2os1 P.hex(실제 실행화면은 다음 그림(그림9)과 같다.)

C:\p2os>p2osdl.exe p2os1_P.hex % Loading Intel hex file p2os1_P.hex Read p2os1_P.hex successfully % 949 32-byte blocks Opening serial port COM1 Sending 187 byte boot loader Monitor is 1765 bytes Monitor loaded! % Serial port, boot loader and memory bus OK. Syscon is 043f FLASH bank 0: data FLASH bank 1: data FLASH bank 2: data FLASH bank 3: data Retrieving parameters from FLASH BANK3..... Downloading flash data... % Downloading 384 blocks to FLASH bank 0 % Flash bank 0 cleared % Flash bank 0 erased. % Downloading 382 blocks to FLASH bank 1 % Flash bank 1 cleared % Flash bank 1 erased. % Downloading 178 blocks to FLASH bank 2 % Flash bank 2 cleared

Fig 9. updating Pioneer 2

나. PIONEER 2의 통신 프로그램

(1). Packet Protocol

P2OS는 표 3과 같이 특별한 명령 패킷을 가지고 클라이언트가 서버에 명령을 내리거나 주어진 명령어에 대한 서버의 상태정보를 클라이언트에게 제공한다.

Component	Bytes	Value	Description
Header	2	0xFA, 0xFB	Packet header; same for client and server
Byte Count	1		Number of subsequent data bytes, including the Checksum word, but not the Byte Count. Maximum 200 bytes.
Data	N	command or SIP	Client command or server information packet (SIP)
Checksum	2	computed	Packet integrity checksum

Table 3. Pioneer 2 Packet Protocol

P2OS는 클라이언트로부터 명령을 받으면 자동으로 호스트 시리얼포트를 통해 패킷 정보를 매 100ms마다 접속되어 있는 클라이언트에게 보내게 된다. 서버패킷은 현재 로봇의 위치, Sonar, Bumper등의 정보를 표4의 명령형식에 맞게 전송한다.

Name	Data Type	Description					
Header	integer	Exactly 0xFA, 0xFB					
Byte Count	byte	Number of data bytes + 2 (checksum); must be less than 201 (0xC9)					
Status/Pac ket	byte = 0x3S; where S =	Motors status					
Type	2	Motors stopped					
	3	Robot moving					
Xpos	unsigned integer (15 ls-bits)	Wheel-encoder integrated coordinates;					
Ypos	unsigned integer (15 ls-bits)	platform-dependent units; multiply by DistConvFactor to convert to millimeters.					
Th pos	signed integer	Orientation in platform-dependent units-multiply by AngleConvFactor‡ for degrees.					
L vel	signed integer	Wheel velocities (respectively Left and Right) in platform-dependent units;					
R vel	signed integer	multiply by VelConvFactor* - currently 1.0 for all - to convert into millimeters per second.					
Battery	byte	Battery charge in tenths of volts					
Stall and Bumpers	integer	Motor stall and bumper accessory indicators. Bit 0 of the lsbyte is the left wheel stall indicator = 1 if stalled. Bits 1-5 of that same byte correspond to the bump switch states (1=on) for the rear bumpers accessory. Bit 0 of the msbyte is the right wheel stall; the bits 1-5 of that same msbyte correspond to the front bumpers switch states.					
Control	signed integer	Setpoint of the server's angular position servo-multiply by AngleConvFactor‡ for degrees					
FLAGS (was PTU)	unsigned integer	b0 - motors flag (1=motors enabled) b1 - sonar flag: enabled if 1.					
Compass	byte	Compass heading in 2-degree units					
Sonar readings	byte	Number of new sonar readings included in information packet; readings follow:					

Sonar number	byte	Sonar number				
Sonar range	unsigned integer	Sonar range; multiply by RangeConvFactor = - currently 0.268 for all - for millimeters				
···rest of th	···rest of the sonar readings···					
Timer	unsigned int	Selected analog port number 1-5				
Analog	byte	User Analog input (0-255=0-5 VDC) reading on selected port				
Digin	byte	User I/O digital input				
Digout	byte	User I/O digital output				
Checksum	integer	Checksum (see previous section)				

Table 4. Server Information Packet

(2). 클라이언트 명령코드

클라이언트는 반드시 명령패킷을 최소 Watchdog(default : 2sec)시간 내에 전송해야하며 그렇지 않으면 로봇은 자동으로 멈추게 된다.

로봇의 연속적인 동작이나 클라이언트의 원하는 동작 수행을 위해서는 최소 명령 사이클 시간 이내에 계속적인 명령이 전송되어야하고 클라이언트 명령 패킷에 맞지 않는 명령어가 전송 될 경우 로봇은 잘못된 작업 수행을 하거나 동작을 멈추게 된다. 클라이언트 명령 전송 방법은 Header, 전송해야 할 명령 (Checksum을 포함)수, 클라이언트의 명령넘버(표 6), 전송되는 명령의 데이터형 그리고 이들 데이터에 대한 Checksum을 구한 후 하나의 명령패킷 (표5) 으로 서버에 전송한다.

Component	By t es	Value	Description
Header	2	0xFA, 0xFB	Packet header; same for client and server
Byte Count	1	N + 2	Number of following command bytes plus Checksum's two bytes, but not including Byte Count. Maximum of 200.
Command Number	1	0 - 255	Client command number; see Table 4-4
Argument Type (command dependent)	1	0x3B or 0x1B or 0x2B	Required data type of command argument: positive integer (sfARGINT), negative integer or absolute value (sfARGNINT), or string (sfARGSTR)
Argument (command dependent)	n	data	Command argument; integer or string
Checksum	2	computed	Packet integrity checksum

Table 5. Client Command Packet

Command	De c	Args	Description	PSO S	P2 OS		
			Before Client Connection				
SYNC0	0	none	Start connection; P2OS echoes				
SYNC1	1	none		3.x	1.0		
SYNC2	2	none	synchronization commands back to client.				
After Established Connection							
PULSE	0	none	Client pulse resets server watchdog	3.x	1.0		
OPEN	1	none	Starts the controller	3.x	1.0		
CLOSE	2	none	Close server and client connection	3.x	1.0		
POLLING	3	string	Set sonar polling sequence	3.9	1.0		
ENABLE	4	int	Enable=1; disable=0 the motors	-	1.0		
SETA	5	signed int	Resets translational acceleration parameter, if positive, or deceleration, if negative; in millimeters per second2	-	1.0		
SETV	6	int	Reset maximum translational velocity,		1.0		

			in millimeters per second		
SETO	7	none	Resets server to 0,0,0 origin	3.x	1.0
MOVE	8	signed int	Translation distance to move in mm		
SETRV	10	int	Resets maximum rotational velocity	4.8	1.0
VEL	11	signed int	in degrees per second Move forward (+) or reverse (-) at millimeters per second	3.x	1.0
HEAD	12	signed int	Turn to absolute heading (+) = counterclockwise; ± degrees	4.2	1.0
DHEAD	13	signed int	Turn relative to current heading (+) = counterclockwise; ± degrees	3.x	1.0
SAY	15	string	As many as 20 pairs of duration (20 ms increments) /tone (half-cycle) pairs; int is string length	4.2	1.0
CONFIG	18	int	1=Request configuration SIP	-	1.4
ENCODER	19	int	Request 1 or continuous stream (>1), or tell to stop sending (0) Encoder SIPs	-	1.4
RVEL	21	signed int	Rotate at ± degrees per second	4.2	1.0
DCHEAD	22	signed int	Heading setpoint relative to last setpoint;		
SETRA	23	signed int	± degrees; (+) = counterclockwise Sets rotational (+)acceleration or (-)deceleration, in mm/sec/sec	_	1.0
SONAR	28	int	1=Enable; 0=disable all the sonar	_	1.0
STOP	29	none	Stops robot (motors remain enabled)	_	1.0
DIGOUT	30	int	Msbits is a byte mask that selects output port(s) for changes; lsbits set (1) or reset (0) the selected port.	4.2	1.2
VEL2	32	signed int	Independent wheel velocities; lsb=right wheel; msb=left wheel; PSOS is in ±4mm/sec; P2OS in ±2cm/sec increments	4.1	1.0
GRIPPER	33	int	Pioneer 1 and Pioneer 2 Gripper server command. See the Pioneer Gripper manuals for details.	4.0	1.3
ADSEL	35	int	Select the A/D port number for analog value in SIP. Selected port reported in SIP Timer value.	-	1.2
GRIPPER VAL	36	int	Pioneer 2 gripper server value. See P2 Gripper Manual for details.	-	
GRIP REQUEST	37	none	Request 1 or continuous stream (>1), or tell to stop sending (0) Gripper SIPs	-	1.E
Ю	40	none	Request 1 or a continuous stream (>1) or tell to stop sending (0) IO SIPs	-	1.E

REQUEST					
PTUPOS	41	int	msb is the port number (1-4) and lsb is the pulse width in 100µsec units PSOS, 10µsec units P2OS. Version 1.J uses RC-servo 40ms	4.5	1.2
TTY2	42	string	duty cycle. Send string argument to serial device connected to AUX port on microcontroller	4.2	1.J 1.0
GETAUX	43	int	Request to retrieve 1–200 bytes from the aux serial channel; 0 flushes the aux serial input buffer.	-	1.4
BUMP_ST ALL	44	int	Stop and register a stall if front (1), rear (2) or either (3) bump-ring contacted and robot motion is in direction of bump.	-	1.5
TCM2	45	int	TCM2 Module commands; see TCM2 Manual for details.	-	1.6
DOCK	46	int	Default is 0=OFF; 1=enable docking signals; 2=enable docking signals and stop the robot when docking power sensed.	-	1.C
JOYDRIVE	47	int	Default is O=OFF; 1=allow joystick drive from hardware port	-	1.G
E_STOP	55	none	Emergency stop, overrides deceleration	-	1.8
E_STALL	56	int	Emergency stop button causes stall	-	1.E
STEP	64	none	Single-step mode (simulator only)	3.x	1.0
ARM	70 - 80	-	Please consult the Pioneer 2 Arm Manual for details.	-	1.H
ROTKP	82	int	Change working rotation proportional PID drive factor (not FLASH default)	-	1.M
ROTKV	83	int	Change working rotation's derivative PID drive factor (not FLASH default)	-	1.M
ROTKI	84	int	Change rotation's integral PID drive factor (not FLASH default)	-	1.M
TRANSKP	85	int	Change working translation proportional PID drive factor (not FLASH default)	-	1.M
TRANSKV	86	int	Change working translation derivative PID drive factor (not FLASH default)	-	1.M
TRANSKI	87	int	Change working translation integral PID drive factor (not FLASH default)	-	1.M
REV COUNT	88	int	Change working revcount (not FLASH default)		1.M
PLAYLIST	90	int	Must be 0; request AmigoBot sound playlist	_	1.E

Table 6. Client Command Set

제3장. 이동로봇의 위치추정

제1절. 이동 로봇 센서시스템 구성

1. 이동 로봇 초기 환경

이동 로봇의 초기 환경 은 로봇의 초기 위치 $\mathbf{x}(0) = [0,0,0]^T$ 에서 임의의시간 $\triangle \nearrow$ 후에 제어 입력 $\mathbf{u} \nearrow$ 와 잡음 분포(noise disturbance) $\mathbf{w} \nearrow$ 대하여 어떻게 다음 위치 $\mathbf{x}(\nearrow) = [x,y,\Theta]^T$ 의 위치로 변화하는지를 나타낸다.

$$x(k+1) = f(x(k), u(k)) + w(k)$$
(3-1)

$$\mathbf{w}(\mathbf{A}) \sim N(0, \mathbf{Q}(\mathbf{A})) \tag{3-2}$$

일반적인 형태로 식(3-1)과 같이 표현된다. 여기서 $f(x(\lambda), u(\lambda))^{-1}$ 비-선형 상태 천이 함수이다.

또한, 시스템 잡음 원천이 공분산 Q♠와 제로-평균(zero-mean) 가우시 안(gaussian)에서 확인되어진 것을 표현하기 위해서, 우리는 식(3-3)과 같은 정규 분포 식의 표시법을 사용한다.

$$\mathbf{w}(\mathbf{A}) \sim N(0, \mathbf{Q}(\mathbf{A})) \tag{3-3}$$

제어 입력 u(A) = [V, Y] 7는 전방 병진(translation) 속도 V와 로봇의 진행 방향을 기준으로 시계 반대 방향으로의 회전각도 Y를 표현하고 있다.

전체적인 시스템의 표현은 다음 그림(그림10)과 같이 초기화 할 수 있다.

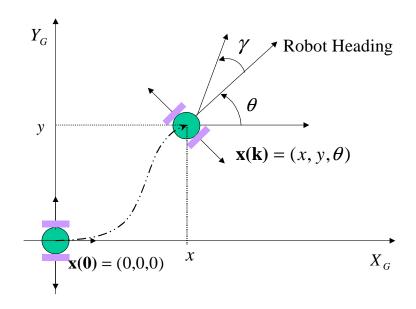


Fig 10. Geometric robot moving system

이동 로봇의 시스템 표현식인 상태 천이 함수 $f(x(\lambda), u(\lambda))^{\leftarrow}$ 식(3-4)와 같은 형태이다.

$$f(x(k), u(k)) = \begin{bmatrix} x(k|k) + \triangle t V \cos(\Theta(k|k) + Y) \\ y(k|k) + \triangle t V \sin(\Theta(k|k) + Y) \end{bmatrix} \Theta(k|k) + Y$$
(3-4)

여기서 상태 천이 함수의 시스템 제어 입력 **2**(A)는 아래 식 (3-5)과 같이 표현된다.

$$\mathbf{u}(\mathbf{k}) = [V, Y]^T \tag{3-5}$$

2. 이동 로봇 센서시스템

가. 엔코더 데이터

로봇에 사용되는 엔코더는 인크리멘탈 엔코더로, 이것은 모터가 회전할때마다 일정한 펄스를 내보내게 된다. 로봇은 이 펄스 수를 세어서 실제로모터가 회전한 각도를 계산하고, 매 샘플마다의 계산된 모터의 회전량을 누적하여 로봇의 위치 및 각도의 변화량을 계산하게 된다. 로봇의 바퀴의 반지름이 R이고, 바퀴 사이의 거리가 D라고 하고, 왼쪽과 오른쪽 엔코더로부터얻어진 펄스수의 차이를 각각 $\triangle counter(A)$, $\triangle counter(A)$ 이라 하면 다음 식(3-6)과 같이 변이 위치와 변이각을 구할 수 있다.

$$\triangle\Theta(R) = \frac{2\pi R}{4N} \frac{(\triangle counter_{s}(R) - \triangle counter_{s}(R))}{D}$$

$$\triangle x(R) = \frac{2\pi R}{4N} \frac{(\triangle counter_{s}(R) + \triangle counter_{s}(R))}{2} \cos(\frac{\Theta(R) + \Theta(R+1)}{2})$$

$$\triangle y(R) = \frac{2\pi R}{4N} \frac{(\triangle counter_{s}(R) + \triangle counter_{s}(R))}{2} \sin(\frac{\Theta(R) + \Theta(R+1)}{2})$$
(3-6)

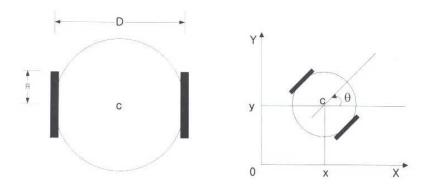


Fig 11. Robot structure

외부 좌표축에 대하여 로봇이 그림(그림11)과 같이 놓여져 있다고 할 때, 로봇의 위치는 다음과 같은 이산 시간 구조학식(kitnematic equations)으로 계산된다.

$$\Theta(\cancel{k}+1) = \Theta(\cancel{k}) + \triangle\Theta(\cancel{k})$$

$$x(\cancel{k}+1) = x(\cancel{k}) + \triangle x(\cancel{k})$$

$$y(\cancel{k}+1) = y(\cancel{k}) + \triangle y(\cancel{k})$$
(3-7)

이때 각 변수들에 대한 설명은 다음과 같다.

 $\triangle counter(R) = (샘플시간 k와 k+1 동안의 왼쪽 바퀴 엔코더 카운터 변화량)$

△ counter (A) = (샘플시간 k와 k+1 동안의 오른쪽 바퀴 엔코더 카운터 변화량)

⊖(★+1) = (샘플시간 k+1에서의 방향각)

△⊖(♠ = (샘플시간 k와 k+1 동안의 방향각 변화량)

x(k+1) = (샘플시간 k+1에서의 X축 좌표값)

△ϰ(ઋ) = (샘플시간 k와 k+1 동안의 X축 좌표값의 변화량)

√(k+1) = (샘플시간 k+1에서의 Y축 좌표값)

△¼(Å) = (샘플시간 k와 k+1 동안의 Y축 좌표값의 변화량)

여기서 π 는 원주율, N은 로봇 바퀴 1 회전시의 엔코더 펄tm 수, R 과 D 는 바퀴 반지름 및 두 바퀴 사이의 거리이다. 이러한 구조학식을 살펴보면로봇의 위치 변수에 따른 상태 방정식 형태로 기술된 것을 알 수 있다. 구조학식에서는 다음과 같은 몇 가지의 가정을 전제로 한다.

- ① 바퀴는 바퀴 축에 수직인 방향에 대해서만 속도 성분을 갖는다.
- ② 바퀴 축에 수직인 방향으로 미끄러짐이 없다.
- ③ 두 바퀴의 반지름은 같고, 반지름 R과 휠베이스 D의 측정오차는 무시한다.

일반적으로 위의 전제조건 ①, ②는 구조학식이 성립하기 위한 가장 기본적인 조건으로 만족한다고 가정하지만, 로봇의 가속도 및 속도에 따라 미끄러짐으로 인해 오차가 발생할 수 있다.

```
short int dTheta = 0;
short int dX, dY;
unsigned char 1_ThetaHighByte;
unsigned char 1_ThetaLowByte;
bool 1 chkSign ;
dX = ((m_pioneer.thXpos[1] << 8) + m_pioneer.thXpos[0]);</pre>
if(dX >0x7FFF)
    dX = (^dX + 1);
m_pioneer.dXpos = dX * 0.4919;
m_Xpos.Format("%4.2f", m_pioneer.dXpos);
SetDlqItemText(IDC XPOS, m Xpos);
dY = ((m_pioneer.thYpos[1] << 8) + m_pioneer.thYpos[0]);</pre>
if(dY >0x7FFF)
    dY = (^dY + 1);
m_pioneer.dYpos = dY * 0.4919;
m_Ypos.Format("%4.2f", m_pioneer.dYpos);
SetDlgItemText(IDC_YPOS, m_Ypos);
```

Fig 12. Range data algorism

그림 12는 이동 로봇에 장착된 엔코더를 통하여 x, y 축으로 이동한 거리를 구하는 알고리즘이다.

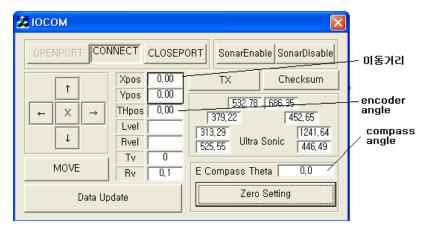
Fig 13. Angle data algorism

그림 13은 이동 로봇에 장착된 엔코더를 통하여 로봇의 각도 변화를 구하는 알고리즘이다.

나. 전자나침반 센서 데이터

본 논문에 사용된 전자나침반 센서는 CMPS03 자계 나침반 모듈을 사용한 센서로서 지자계 값을 전압값으로 환산하여 이 값에 의해 로봇의 회전량을 측정한다.

센서로부터의 측정된 데이터는 RS-232 시리얼 통신 방식으로 컴퓨터와데이터를 주고 받을 수 있다. 측정된 데이터는 센서 자체에 내장되어 있는 12C Interface를 통해 컴퓨터에 전송한다.



실제 로봇통신 프로그램 초기화면

Fig 14. Data of Encoder and Electric compass

그림14는 실제 두 개의 센서로부터 획득된 데이터이다. 엔코더에서 획득한 X , Y축으로의 이동거리와 로봇의 회전 각도 및 전자나침반으로부터 획득한 로봇의 회전 각도를 보여준다.

제2절. 엔코더 데이터를 이용한 위치 추정

1. 위치 추정 알고리즘의 개요

가. 로봇 위치 추정

지능형 로봇에 있어서 이동성은 가장 큰 특징중의 하나로, 산업용 로봇과서비스 로봇을 구분짓는 가장 큰 요소이다. 이동 로봇에 주어지는 임무는 원하는 목적지로 이동하여 원하는 일을 수행하는 것으로, 원하는 위치로 이동하는 작업을 항법(navigation)이라 하며, 이동 로봇 분야의 연구에서 가장 활발히 연구되고 있는 분야이다. 항법은 크게 로봇의 위치 추정 분야, 경로 설계 분야, 경로 추종 제어 분야로 구성된다. 이들 세 분야의 연구 중, 경로 설계와 경로 추종 제어 분야는 오랜 역사를 가지고 연구되어진 반면, 위치 추정 분야는 최근에 활발히 연구 되어지기 시작하였다.

로봇이 이동하기 위해서는 목적지의 위치와 현재의 위치 정보를 바탕으로 하여, 모터를 구동하기 위한 제어 입력이 계산되고, 이 제어 입력에 따라로봇이 이동하게 된다. 따라서 로봇이 주어진 작업을 수행하기 위해서는 기본적으로 로봇이 어느 위치에 있는가에 관한 정보가 필요하게 된다. 로봇의위치를 표현하는 방법은 좌표값을 이용하는 방법과 지형적인 특징 정보 (코너,문앞, 복도 등)을 이용하는 방법이 있다. 이와 같은 표현 방법은 로봇의 사용예에 따라서 달라지게 된다. 이 경우 로봇의 위치가 좌표값으로 표현되는 것이 훨씬 더 쉽게 작업에 이용될 수 있다. 이와 달리, 감시 등의 작업에 사용되는 로봇의 경우에는, 로봇이 미리 정해진 지역을 대략적으로 순회만 하면 된다. 이와 같은 경우에는 로봇의 위치를 정확히 좌표상에 기술하기보다는 대략적으로 로봇이 어느 지역에 있다라는 형태로 표현하는 것이 바람직하다.

위치 추정 방법은 로봇의 위치를 표현하는 방법에 따라 다르다. 좌표값을 이용하여 로봇의 위치를 표현하는 방법은 비교적 손쉽게 제어 입력을 유도할

수 있고 위치 추정 알고리즘의 수렴성이나 안정성 등의 연구에 적합하기 때문에 전통적으로 많이 연구 되어왔다. 이 방법은 로봇이 획득하는 센서 정보로부터 로봇이 동작하는 환경에 대하여 수치데이터를 얻어야 하는데, 다양한형태의 환경에 대하여 적절한 수치데이터를 얻기가 어렵다는 것이 단점이다.특히 예측되어지지 않은 환경에서 동작할 경우, 로봇의 위치를 계산하지 못하고 발산할 가능성이 있게 된다.

위치 추정 분야는 로봇의 초기 위치가 알려져 있느냐의 여부에 따라 크 게 두 가지로 나눌 수 있다. 로봇의 초기 위치가 알려진 경우에는 로봇의 움 직임에 따라 지속적으로 로봇의 위치를 추정하면 된다. 이 분야는 자세 유지 (pose maintenance) 혹은 자세 추종(pose tracking)이라 불리우며, 주로 칼만 필터(Kalman filter) 등을 이용하여 로봇의 움직임을 추적한다. 이 분야는 오 랜 역사를 가지고 있으며 많은 성공적인 연구 결과를 내고 있다. 두 번째 분 야는 로봇의 초기 위치를 전혀 모르는 상태에서 로봇의 위치를 추정하는 분 야로, self-localization 또는 kidnapped robot localization이라는 명칭으로 불 린다. 이 분야의 연구로는 마르코프 위치 추정 (Markov localization)이라 불 리우는 연구가 주로 이루어지고 있다. 이 방법은 로봇의 초기 위치를 모르기 때문에 로봇의 움직임과 센서로부터 획득한 정보를 통해 로봇이 존재할 확률 을 지속적으로 계산해 나가는 방법이다. 이 방법을 적용하기 위해서는 로봇 의 작업 공간을 이산화하고, 각 이산화된 영역에 지속적인 가능성 (likelihood) 함수의 계산이 필요하다. 이 방법은 이산화 개수에 따라 많은 계 산량이 필요한 단점이 있다. 이 분야는 현재 계속해서 연구가 진행 중인 분 야이다.

본 논문에서는 이동 로봇의 위치 추정에 많이 사용되고 있는 엔코더와 전자나침반 센서를 이용하여 로봇의 정확한 위치를 추정하는 알고리즘을 제 안한다. 엔코더와 전자나침반 센서를 이용하여 로봇이 이동한 단위 시간동안 의 변이 거리(혹은 속도)와 변위각(혹은 각속도)을 측정하여 이 값을 적분하 여 위치를 추정하는 방법이다. 우리가 추정하고자 하는 로봇의 상태는 위치 와 속도, 각속도 이다. 로봇 위치 정보는 전자나침반 센서와 엔코더를 이용하여 각각의 정보를 독립적으로 이용하여 로봇의 상태를 추정할 수 있겠지만 보다 더 정확한 위치 정보를 얻기 위해서는 두 개의 센서로부터 얻어진 정보 를 적절히 융합하여야 한다.

본 논문에서는 엔코더 데이터와 나침반센서 데이터를 동시에 획득하여 두 데이터를 비교, 실시간으로 오차를 최소화 하여 보다 정확한 로봇의 위치를 추정하는 알고리즘을 제안한다.

나. 이산 칼만 필터 이론 (Discrete Kalman Filter)

1969년에, 칼만이 이산-데이터 선형 필터링 문제에 대한 반복적인 해를 묘사한 그의 유명한 논문을 출판했다. 그 후, 디지털 계산이 발전하는 큰 분야가 되었고, 칼만 필터는 특히 자율 또는 주행 도움의 분야에서, 광범위한연구와 응용의 주제가 되었다. 칼만 필터는 대표적인 IIR(infinite response filter) 필터로서 반복적인 방법을 이용한 최소 자승법의 효율적인 계산 해를 제공하는 5개의 수학적 방정식의 집합으로 과거와 현재 그리고 미래 상태(state)의 추정 값을 제공한다. 또한 시스템의 정밀한 성질이 알려지지 않을때에도 사용 가능한 필터이다^[15].

칼만 필터는 아래 식(3-8)에 주어진 일반적인 형태의 선형 확률적 차분 방정식에 의해 결정된 이산-시간 공정(process)의 상태 $_{\mathcal{X}} \in \mathbb{R}^{n}$ 과 아래 식(3-9)에 주어진 측정 $_{\mathcal{Z}} \in \mathbb{R}^{m}$ 을 추정하기 위해서 사용된다.

$$x_{k+1} = A_k x_k + B u_k + w_k \tag{3-8}$$

$$z_k = H_k x_k + v_k \tag{3-9}$$

랜덤 변수(random variables) w_k, v_k 는 공정과 측정 노이즈를 각각 표시한다. 이 변수 들은 서로 독립적인 것으로 가정되고, 백색 과 정상 확률 분포를 갖는 것으로 가정되며, 식(3-10)과 식(3-11)과 같이 표현한다.

$$p(w_k) \sim \mathcal{N}(0, Q_k) \tag{3-10}$$

$$p(v_k) \sim N(0, R_k) \tag{3-11}$$

이산 칼만 필터 방정식은 크게 시간 갱신 방정식(time update equation)과 측정 갱신 방정식(measurement update equation)으로 나뉘어 진다. 각각의 방정식은 아래와 같다.

$$\widehat{x}_{k+1}^- = A_k \widehat{x}_k + Bt_k \tag{3-12}$$

$$P_{k+1} = A_k P_k A_k^T + Q_k \tag{3-13}$$

식(3-12)에서 $\widehat{\chi_{k+1}}$ 는 과정 시스템 A_k 와 시스템 입력 B에 대해 전 상태 추정을 표현한 식이며, 식(3-13)에서 P_{k+1} 은 공정 시스템 A_k 와 시스템 노이즈 Q_k 에 대해 에러의 전 상태 추정(priori estimation)을 표현한 식이다.

$$K_{k} = P_{k}^{-} H_{k}^{T} (H_{k} P_{k}^{-} H_{k}^{T} + R_{k})^{-1}$$
(3-14)

$$\widehat{x}_{k} = \widehat{x}_{k+1}^{-} + K(z_{k} - H_{k} \widehat{x}_{k+1}^{-})$$
(3-15)

$$P_{k} = (I - K_{k}H_{k})P_{k+1}^{-}$$

$$= P_{k+1}^{-} - K_{k}H_{k}P_{k+1}^{-}$$
(3-16)

식(3-14)은 칼만 이득을 표현한 식이다. 또한 식(3-15)과 식(3-16)에서 $\widehat{\chi_k}$ 와 P_k 는 정정된 후 상태의 추정(posteriori estimation)을 표현한 식이다.

다. 확장 칼만 필터 이론 (Extended Kalman filter)

확장 칼만 필터는 이산 칼만 필터의 확장된 형태로써, 비-선형 확률적 차분 방정식의 해로 현재의 평균(mean)과 공분산(covariance)을 선형화 시켜서 계산하며, 추정되어지는 공정이나 공정에 대한 측정의 상관 관계가 비-선형적으로 일어날 때 사용된다. 비-선형 관계에 있는 추정을 계산하기 위해 공정과 측정함수의 편도함수(partial derivatives)를 사용하여 추정을 선형화 시킨다^[15]. 시간 스텝 세서 ←1까지 추정된 상태와 공분산을 표현한 시간 갱신 방정식은 아래 식(3-17)과 (3-18)에 나타내었다.

$$\widehat{x}_{k+1}^- = f(\widehat{x}_k \ u_k, \ 0) \tag{3-17}$$

$$P_{k+1}^{-} = A_k P_k A_k^T + W_k Q_k W_k^T$$
 (3-18)

여기서 $f(\cdot)$ 는, A_k 와 W_k 는 스텝 ℓ 에서 과정의 야코비안(jacobians)이고, Q_k 는 스텝 ℓ 에서 동작 노이즈를 표현한다.

측정 z_k 와 함께 평가된 상태와 공분산을 정정하기 위한 측정 갱신 방정식은 아래에 나타내었다.

$$K_{k} = P_{k}^{-} H_{k}^{T} (H_{k} P_{k}^{-} H_{k}^{T} + V_{k} R_{k} V_{k}^{T})^{-1}$$
(3-19)

$$\widehat{x_k} = \widehat{x_k} + K(z_k - h(\widehat{x_k}, 0))$$
 (3-20)

$$P_{k} = (I - K_{k}H_{k})P_{k}^{-} \tag{3-21}$$

여기서 $h(\cdot)$ 는, H_k^{ol} V_k^{ic} 스텝 M서 측정의 야코비안이고, R_k^{ic} 스텝 M서 측정의 야코비안이고, R_k^{ic} 스텝 M서 측정 노이즈 공분산을 표현한다. 확장 칼만 필터의 중요한 형식은 칼만 이득 K_k^{ol} 위한 방정식에서 야코비안 H_k^{ol} 측정 정보의 적절한 요소로 단지 확대 또는 정확하게 지연을 제공한다는 점이다.

확장 칼만 필터 방정식에서 비-선형 확률적 상태 방정식을 근사적으로 선형화하기 위해 사용되는 야코비안 행렬을 아래에 나타내었다.

· A와 W는 각각 x와 u에 관한 $f(\cdot)$ 의 편 도함수 의 야코비안 행렬이다.

$$A_{[i,j]} = \frac{\vartheta f_{[i]}}{\vartheta_{\mathcal{X}_{[i]}}} \left(\widehat{x}_{k} \ \mathcal{U}_{k}, 0 \right) \tag{3-22}$$

$$W_{(i,k)} = \frac{\vartheta f_{[k]}}{\vartheta w_{[k]}} \left(\widehat{x}_k \ u_k, 0 \right) \tag{3-23}$$

 \cdot H와 V는 각각 $_{\chi}$ 와 $_{\chi}$ 에 관한 $_{h}(\cdot)$ 의 편 도함수 의 야코비안 행렬이다.

$$H_{[i,\lambda]} = \frac{\partial h_{[\lambda]}}{\partial x_{[\lambda]}} \left(\widetilde{x_{k}} 0 \right) \tag{3-24}$$

$$V_{\text{Li,l}} = \frac{\vartheta_{\text{M}_{\text{L}}}}{\vartheta_{v_{\text{L}}}} \left(\widetilde{x_{\text{L}}} \right)$$
 (3-25)

칼만 필터는 피드백 제어의 형태를 이용해 과정을 평가한다. 시간 갱신 방정식은 현재 상태와 다음 시간 스텝에 대한 전(priori) 평가를 획득하는 에러 공 분산 평가를 앞에서 반영하는 역할을 한다. 측정 갱신 방정식은 피드백 역할을 한다. 개선된 후(posteriori) 평가를 얻기 위해 전 평가 속에 새 측정을 결합한다. 실제로 추정 알고리즘은 아래 그림(그림15)에 보인 수치 문제해결을 위한 예측자(predictor) - 교정자(corrector) 알고리즘 형태를 취하고 있다^[15].

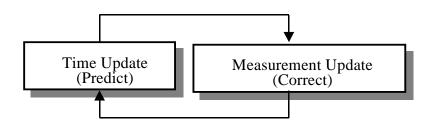


Fig 15. Kalman filter cycle

2. 이동로봇의 위치 추정의 세부사항

가. 이동 로봇 시스템

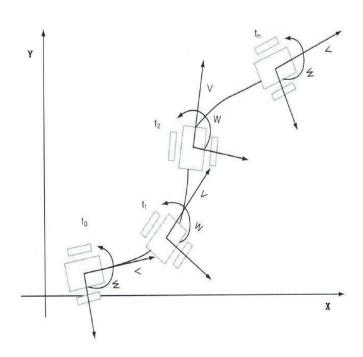


Fig 16. Inertial Navigation System

그림 (그림16)과 같이 이동 로봇이 2차원 평면에서 움직인다고 가정하면, 이동 로봇의 위치는 x축 좌표, y축 좌표, 로봇의 주행 방향각등 3개의 변수로 표현될 수 있다. 이동 로봇의 위치 추정에 많이 사용되고 있는 방법은 관성센서를 이용하여 로봇의 위치를 추정하는 관성 항법 시스템이다. 관성 항법시스템이란 다음 식과 같이, 로봇이 이동한 단위 시간동안의 변이 거리(혹은속도)와 변위각(혹은 각속도)을 측정하여 이 값을 적분하여 위치를 추정하는 방법이다.

$$\chi(\lambda) = \chi(t_0) + \int_{\tau = t_0}^{t} \ell(\tau) \cos \Theta(\tau) d\tau$$

$$\chi(\lambda) = \chi(t_0) + \int_{\tau = t_0}^{t} \ell(\tau) \sin \Theta(\tau) d\tau$$

$$\Theta(\lambda) = \Theta(t_0) + \int_{\tau = t_0}^{t} u(\tau) d\tau$$
(3-26)

관성 항법 시스템에 많이 사용되는 센서로는 가속도 센서, 자이로스코프, 엔코더 등이 있다. 일반적으로 가속도 센서와 자이로스코프는 잡음이 많이 섞여 있기 때문에, 가속도의 변화가 급격한 이동 로봇의 경우 모터에 장착된 엔코더를 주로 사용한다. 그러나 이동 로봇의 경우, 바퀴와 바닥의 미끄러짐이나 마찰력 등에 의해 엔코더 정보에는 약간의 오차가 섞여 있게 된다. 또한 연속적인 측정이 불가능 하여, 다음 식 (3-27)과 같이 일정한 표본 검출시간(△Д)마다 측정하기 때문에, 표본 검출 시간 사이에 로봇의 속도나 각속도가 변화하는 경우에도 엔코더로부터 얻은 로봇의 변위에 오차가 섞이게 된다. 그리고, 로봇의 위치를 추정하기 위해서는 식 (3-26) 이나 식 (3-27)에서처럼 초기위치 (x(Д), y(Д), ӨД))를 정확히 알아야 된다는 것도 문제가될수 있다.

$$x(t) = x(t_0) + \triangle t \sum_{k=1}^{n} t(k \triangle t) \cos^{\Theta}(k \triangle t)$$

$$y(t) = y(t_0) + \triangle t \sum_{k=1}^{n} t(k \triangle t) \sin^{\Theta}(k \triangle t)$$

$$\Theta(t) = \Theta(t_0) + \triangle t \sum_{k=1}^{n} t(k \triangle t)$$
(3-27)

(1). 로봇 이동 위치의 예측

자율 주행 로봇의 주행은 크게 직선 주행과 회전 주행으로 구분할 수 있다. 일반적인 동작의 경우 직선 주행이지만 회전 구간이나 장애물 회피 구간에서는 회전 주행이라 할 수 있다. 이 회전 주행의 경우 원형의 궤적을 그리게 된다. 자율 주행 로봇의 주행에 대한 기본적인 동작방정식의 유도를 통해로봇의 주행 궤적이 어떠한 형태인지 알 수 있다. 방정식의 유도는 로봇의 병진 및 회전 속도를 모두 제어할 수 있다는 가정 하에 정확한 동역학적 법칙에서 시작한다. 방정식을 좀더 실제적으로 만들기 위해, 로봇의 속도를 임의의 시간구간에 대해 일정한 상수 값으로 가정한다. 이와 같이 속도가 일정하다고 가정하면, 로봇이 동작하는 전체 좌표계에서 로봇의 현재 위치와 방향을 임의의 시간 t에 대해 $(x(h),y(h),\Theta(h))$ 로 놓으면, $(x(t_0),y(t_0))$ 와 $(x(t_n),y(t_n))$ 는 시간 t와 t에서의 로봇의 위치를 나타낸다. 그리고 임의의시간 t에 대해 t(t)와 t0가로 로봇의 병진 속도와 회전 속도로 놓으면, t1에서의 로봇의 좌표 t2가 로봇의 병진 속도와 회전 속도로 놓으면, t2가 로봇의 좌표 t3가 t4가 t5가 t6가 t7가 t7가

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} t t(t) \cdot \cos\Theta(t) dt$$
 (3-28)

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} t t(\hbar) \cdot \sin\Theta(\hbar) dt$$
 (3-29)

먼저, x성분에 대하여 수식(3-28) 을 로봇의 초기 위치, 초기 병진 및 회전속도, 그리고 각각의 병진 및 회전 가속도의 성분으로 다시 정리를 하면다음과 같이 쓸 수 있다.

여기서 $t(t_0)$ 와 $n(t_0)$ 는 초기의 병진 및 회전 속도이고, $t(t_0)$ 와 $n(t_0)$ 는 $t\in [t_0,t_0]$ 구간에서의 병진 및 회전 가속도이다. 좀 더 실제적이고 근사화된 모델을 유도하기 위해, 임의의 시간 $[t_i,t_{i+1}]$ 구간에 대해 로봇의 속도를 일정하다고 가정하고, 수식을 다시 정리하면 다음과 같은 동작 방정식을 세울 수 있다.

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} t v_i \cdot \cos(\Theta(t_i) + v_i(\tau - t_i)) d\tau$$
 (3-31)

수식(3-32)에서 적분을 풀고 정리하면 다음과 같이 정리할 수 있다.

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} (F_x^i(t_{i+1}))$$
 (3-32)

여기서

$$(F_{x}^{i}(t_{i+1}) = \frac{tv_{i}}{v_{i}}(\sin(\theta(t_{i+1}) + v_{i} \cdot (t - t_{i+1})) - \sin\theta(t_{i+1})), \quad v_{i} \neq 0$$

$$tv_{i}\cos(\theta(t_{i+1})) \cdot t, \quad v_{i} = 0$$
(3-33)

같은 방법으로 Y좌표에 대해 정리하면.

$$y(t_n) = y(t_0) + \sum_{i=0}^{n-1} (F_y^i(t_{i+1}))$$
(3-34)

$$(F_{y}^{i}(t_{i+1}) = \frac{tv_{i}}{v_{i}}(\cos(\Theta(t_{i+1}) + v_{i} \cdot (t - t_{i+1})) - \cos\Theta(t_{i+1})), \quad v_{i} \neq 0$$
(3-35)

$$tv_i \sin(\Theta(t_i)) \cdot t$$
, $v_i = 0$

수식에서 $v_{i}=0$ 이면, 로봇은 직선의 경로를 따르는 것을 알 수 있다.

(2). 실제 이동거리에 대한 분석

(가). 엔코더 데이터의 위치 오차

이제부터는 로봇에 장착된 엔코더에서 획득한 데이터와 실제 로봇이 이동한 거리와 각도 오차를 알아보기 위해서 실험에 사용된 Pioneer 2 이동로 봇을 가지고 초기상태에서 직접 병진속도와 회전속도를 주어 로봇을 이동 시켜 보았다. 로봇이 실제 주행한 환경은 굴곡이 심하지 않는 건물 내 복도에서 실험해보았다.



Fig 17. encoder data

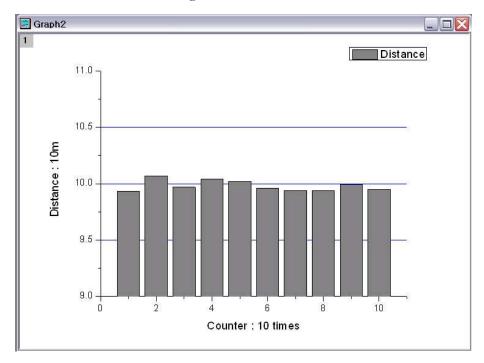


Fig 18. encoder data graph

그림 18은 병진속도만 주어진 실제 이동 로봇이 10m를 직선거리로 이동한 후 엔코더에서 획득한 이동 거리를 나타낸다. 도표에서 보듯이 지면 상태에 따라 바퀴의 미끌림현상 등으로 실제 엔코더에서 획득한 이동거리는 약간의 오차가 발생함을 알 수 있다.

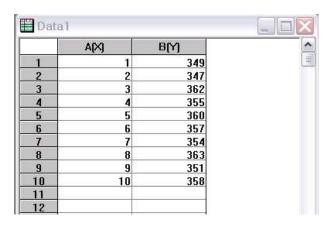


Fig 19. encoder data

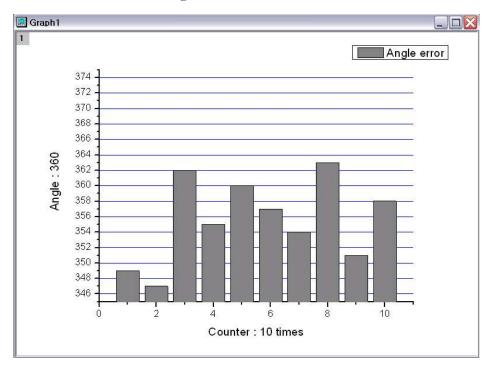


Fig 20. encoder data graph

그림 20은 병진속도는 적게 주고 회전속도를 많이 준 상태에서 실제 이동로봇이 원형으로 360°를 회전 이동 시켰다. 회전 이동을 시켜본 결과 원점으로 다시 되돌아 왔을 때 엔코더에서 획득한 각도 데이터는 상당한 각도 오차

가 발생함을 알 수 있다. 하지만 전자나침반은 절대방위를 나타내므로 각도 오차는 발생하지 않는다. 이것은 이동로봇이 이동함에 따라 각도 오차가 누 적되면 로봇의 정확한 위치를 알아 낼 수 없음을 말해준다. 따라서 이동로봇 의 정확한 위치를 추정하기 위해 각도 오차가 최소한으로 발생하지 않는 전 자나침반을 이용하면 각도보정이 자연스럽게 이루어져 더욱 정확한 이동로봇 의 위치추정을 할 수 있다.

제3절, 전자나침반을 이용하여 로봇의 위치오차 보정

지금까지 이동 로봇의 유연한 주행을 위해 필수 조건 중 하나인 로봇의 위치 추정에 대한 필요성과 위치 추정을 위한 방법들에 대해 논하였다.

로봇이 목표점을 향해 움직이기 시작하는 순간부터 모든 센서들은 데이터를 획득하고 또 장애물을 탐지하며 꾸준히 로봇 자신의 위치를 추정하면서 위치를 저장해 나간다. 이때 두 가지 센서가 동시에 획득한 데이터를 비교하여 좀 더 정확한 로봇의 위치를 추정해 가기 위해 로봇은 꾸준한 업데이트를 통해 주어진 임무를 수행하면서 목적지까지 이동해 간다.

로봇의 초기 위치가 주어졌을 경우, 이동 로봇의 주행에 대한 기본적인 동작방정식을 유도하여 로봇의 주행 궤적이 어떠한 형태인지 알 수 있었다. 방정식의 유도는 로봇의 병진 및 회전 속도를 모두 제어할 수 있다는 가정하에 정확한 동역학적 법칙에서 시작하였고, 로봇이 동작하는 동안에 임의의병진 속도와 회전 속도를 각각 주어서 로봇이 목표점을 향해 이동하면서 획일적이지 않고 자율적으로 이동할 수 있도록 하고서 로봇의 위치 추정 후 오차를 줄이기 위해 엔코더에서 획득한 거리데이터는 사용하고 실제 오차가 크게 발생하는 각도데이터는 전자나침반에서 획득한 데이터를 가지고 로봇의위치를 추정하여 엔코더만을 사용하였을 때보다 이동로봇의 좀 더 정확한 위치를 추정 할 수 있음을 실험을 통해 증명하고자 한다.

제4장. 실험 및 고찰

본 논문에서는 목표점까지 이동하는 로봇이 유연하게 로봇의 임무를 수행할 수 있도록 하기 위해서 선행되어야할 로봇이 자신의 위치를 추정하는 방법에 대해 논했다.

본 논문에서 논의했던 사항들에 의해 정확한 위치 추정을 할 수 있는 방법을 적용하여 실제 실험을 통해 증명해 보이고자 한다. 실험에 사용 된 이동로봇 은 다음 그림21과 같다.



Fig 21. Pioneer 2

제1절. 위치추정 실험

본 실험의 순서는 다음과 같다.

먼저 원운동 주행 실험, 원운동 교차주행 실험, 직선 주행 실험, 자율 주행 실험1, 자율 주행 실험2로 실험을 수행 하였다.

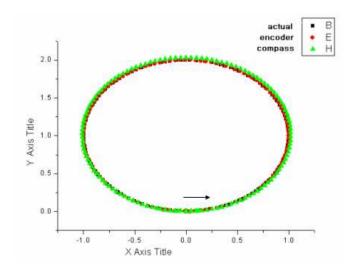


Fig 22. Trajectory of Circle Navigation

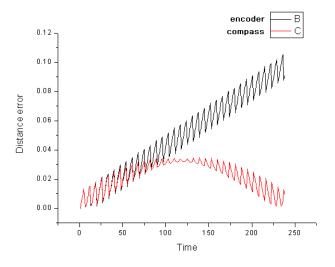


Fig 23. Distance error of Circle Navigation

그림22는 실제 이동로봇이 원운동 주행 실험 후 이동한 궤적이다. 그림23 은 원운동 주행 실험 후 실제 로봇이 이동한 경로를 엔코더 데이터만 가지고 위치추정을 한 결과와 전자나침반을 같이 사용하여 위치추정을 한 결과를 가 지고 실제 로봇이 이동한 경로와의 거리오차를 나타낸다.

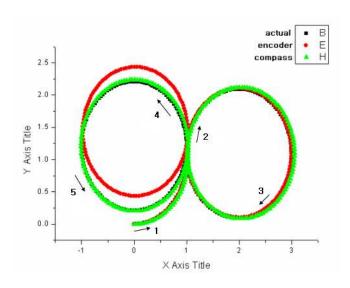


Fig 24. Trajectory of Cross Circle Navigation

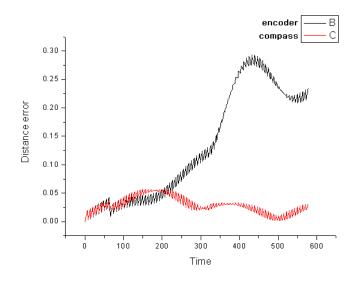


Fig 25. Distance error of Cross Circle Navigation

그림24는 실제 이동로봇이 교차 원운동 주행 실험 후 이동한 궤적이다. 그림25는 교차 원운동 주행 실험 후 실제 로봇이 이동한 경로를 엔코더 데이 터만 가지고 위치추정을 한 결과와 전자나침반을 같이 사용하여 위치추정을 한 결과를 가지고 실제 로봇이 이동한 경로와의 거리오차를 나타낸다.

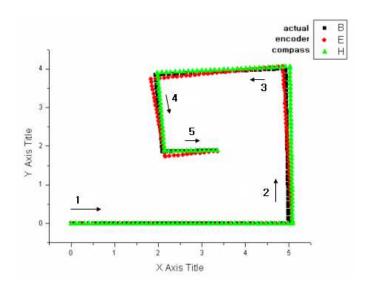


Fig 26. Trajectory of Straight Navigation

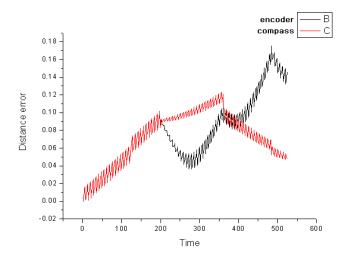


Fig 27. Distance error of Straight Navigation

그림26은 실제 이동로봇이 직진 주행 실험 후 이동한 궤적이다. 그림27은 직진 주행 실험 후 실제 로봇이 이동한 경로를 엔코더 데이터만 가지고 위치 추정을 한 결과와 전자나침반을 같이 사용하여 위치추정을 한 결과를 가지고 실제 로봇이 이동한 경로와의 거리오차를 나타낸다.

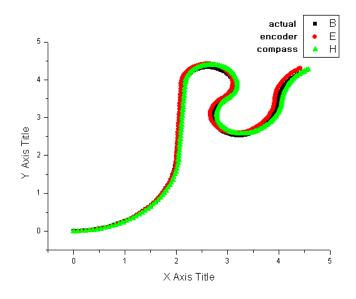


Fig 28. Trajectory of Autonomous Navigation

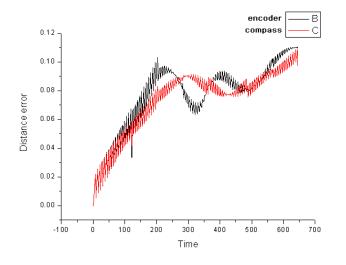


Fig 29. Distance error of Autonomous Navigation

그림28은 실제 이동로봇이 자율 주행 실험 후 이동한 궤적이다. 그림29는 자율 주행 실험 후 실제 로봇이 이동한 경로를 엔코더 데이터만 가지고 위치 추정을 한 결과와 전자나침반을 같이 사용하여 위치추정을 한 결과를 가지고 실제 로봇이 이동한 경로와의 거리오차를 나타낸다.

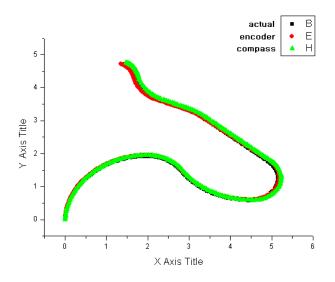


Fig 30. Trajectory of Autonomous Navigation

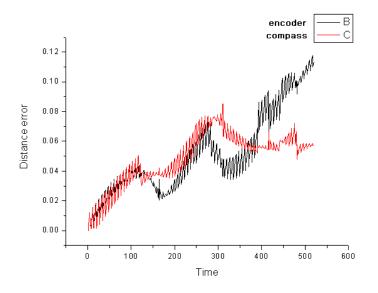


Fig 31. Distance error of Autonomous Navigation

그림30은 실제 이동로봇이 자율 주행 실험 후 이동한 궤적이다. 그림31은 자율 주행 실험 후 실제 로봇이 이동한 경로를 엔코더 데이터만 가지고 위치 추정을 한 결과와 전자나침반을 같이 사용하여 위치추정을 한 결과를 가지고 실제 로봇이 이동한 경로와의 거리오차를 나타낸다.

제2절. 고 찰

지금까지 복도 환경에서 로봇에 주어지는 외부 환경을 이용하지 않고로봇의 위치와 자세를 추정하는 알고리즘을 실험하였다.

로봇이 엔코더와 로봇 외부에 부착된 전자나침반을 이용하여 데이터를 획득하고, 획득된 데이터를 기반으로 현재 위치와 자세를 예측하였다.

로봇이 구동하면서 생기는 이동 과정 오차(process Noise)를 실험을 통해 분석하여 로봇의 구동 방정식인 비-선형 상태 방정식의 식에 포함하여 계산하 였다. 또한, 센서로 입력되는 데이터의 측정 노이즈(measurement noise)를 계 산하여 로봇의 측정 갱신 방정식에 포함하여 계산하였다. 관찰 결과 전자나침 반을 이용하여 위치추정을 한 결과의 오차는 ±5CM 이하로 정확하게 측정되었 다.

제5장. 결 론

본 논문에서는 이전에 사용된 엔코더와 정확한 변위 정보를 획득하기 위해서 전자나침반 센서를 사용하여 정보를 획득하고, 획득한 데이터를 확장 칼만 필터에 적용하여, 이동 로봇이 이동하면서 실-시간으로 로봇의 위치를 추정하는 방법을 제안하였다.

또한, 본 논문에서는 외부 환경에 다른 특정 표식을 설치하지 않고 직선, 코너 그리고 모서리와 같은 구조화된 외부 환경을 이용하지 않고 상대적으로 로봇의 위치를 추정하는 방법을 사용하였다. 구조화된 외부 환경을 사용하기 위해서 기하학적인 방법으로 목표물들을 모델링이 필요하기 때문이다. 또한 반복적인 실험으로 로봇의 입력 값에 대한 위치 변화식인 로봇 구동 방정식을 세워서 칼만 필터에 적용하였다. 센서의 측정 방정식의 에러 공분산 범위를 실험과 가우스 분포를 이용하여 칼만 필터의 에러 공분산에 적용하였다.

제안된 알고리즘을 검증하기 위하여 실제로 복도 환경에서 전자나침반 센서와 Pioneer 2 로봇을 이용하여 실험을 실행하였다. 실험의 결과에서 위치를 추정하고 자세를 보정하는 시간을 단축할 수 있었고, 실-시간으로 처리가능함을 알 수 있었다. 관찰 결과 오차는 ±5cm 이내로 정확하게 나타남을 알수 있었다.

향후 연구 과제로는 본 논문에서 제안한 위치 추정 방법을 레이져 거리 파인더를 이용한 장애물 회피나 경로 계획 설정 또는 외부 환경 인식에 적용하고, 이동 로봇이 어느 위치에서 동작하고 있는지를 특정 지역에 설치된 로봇의 서버 컴퓨터에서 무선이나 네트워크를 이용하여 로봇을 감시하고 제어할 수 있는 통합적인 로봇 제어 시스템 개발이 필요하다.

참고문헌

- [1] Borenstein, J. and Feng, L., where am I? Sensors and Method for Autonomous Mobile Robot Positioning-1995 Edition, 1995c,
- [2] Kang, D. and Hashimoto, H., "Position Estimation for Mobile Robot Using Sensor Fusion," Institute of Industrial Science, Tokyo Univ, 1995.
- [3] Feder, H. S. and Leonard, John. J., "Adaptive Concurrent Mapping and Localization Using Sonar," Proc. of the 1996 IEEE International Conference on Intelligent Robots and Systems, October, 1998, pp 892–898.
- [4] Komoriya, K. and Oyama, E., "Position Estimation of a Mobile Robot Using Optical Fiber Gyroscope(OFG)," Proc. of the 1994 IEEE International Conference on Intelligent Robots and Systems, 1994, pp 143–149.
- [5] 양해용, 정학영, 이장규, "회전 초음파 센서와 지도를 이용한 이동 로봇의 동적 절대 위치 추정," 제12차 한국 자동제어학술회의 논문집, October, 1997.
- [6] Holenstein, A. A., Muller, M. A., and Badreddin, E., "Mobile Robot Localization in a Structured Environment Cluttered with Obstacles," IEEE International Conference on Robotics and Automation, pp. 2576–2581, Nice, France, May 1992.
- [7] Ko, J. H., Kim, S. D., and Chung, M. J., "A Method of Indoor Mobile Robot Navigation Using Acoustic Landmarks," IEEE Conference on Robotics and Automation, pp. 1726–1731, Minneapolis, Minnesota, April 1996.
- [8] Leonard, John J. and Durrant-Whyte, Hugh F., *Direct Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers.
- [9] Borenstein, J., Everett, H. R., and Feng, L., *Navigating mobile robots*, Massachusetts: AK Perters, 1996.
- [10] Hancock, J. A., "Laser Intensity-Based Obstacle Detection and

- Trajacking"Http://www.ri.cmu.edu/pub_files/pub1/ancock_john_1999_1/hancock_john_1999_1.pdf
- [11] Stadler, E., "Definition of telegrams between the user interface and sensor system via RS 422 / RS 232," Part-No. 2017155, SICK. Inc, July 1995.
- [12] Reina, A. and Gonzalez, J.,"Characterization of a radial laser scanner for mobile robot Navigation," Proc. of the 1997 IEEE International conference on Intelligent Robots and Systems, September 1997, pp.579–585.
- [13] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipies in c: The Art Scientific computing*, Cambridge University Press, 1992.
- [14] Brown, R. G. and Hwang, P. Y. C. . 1992. *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, John Wiley & Sons, INC.
- [15] Welch, G. and Bishop, G., "An Introduction to the Kalman Filter" Dept. of Computer Science, University of North Carolina at Chaper Hill
- [16] Borthwick, S. and Durrant-Whyte, Hugh F., "Simultaneous Localization and Map Building for Autonomous Guided Vehicles," Proc. of the 1994 IEEE International Conference Robotics and Automation, May, 1994, pp.761 768.
- [17] Hoppen, P., Knieriemen, T., and Puttkamer, E. von, "Laser-rader based Mapping and Navigation for an Autonomous Mobile Robots," IEEE International Conference on Robotics and Automation, 1990.
- [18] Feng, Lu and Milios, E. " Sharp Registration Using Optimization For Mobile Robot Navigation," Dept. of Computer Science, Toronto Univ, 1995.
- [19] Einsele, T. and Färber, G., "Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder," Proc. of the

1997 IEEE International conference on Intelligent Robots and Systems, September, 1997, Grenoble, France, pp.697 - 702.