



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

# 센서융합을 통한 무인지게차의 주행시스 템 연구

Development for Driving System using Fusion of  
Hybrid Sensors of Automatic Guided Vehicle(AGV)



지도교수 최 형 식

2014년 2월

本 論 文 을 申 熙 永 의 工 學 碩 士 學 位 論 文 으 로  
認 准 함 .

한국해양대학교 대학원

기 계 공 학 과

신 희 영

위 원 장 유 삼 상 (인)

위 원 최 형 식 (인)

위 원 조 종 래 (인)

2014 년 2 월

한국해양대학교 대학원

# 목 차

Abstract

그림목차

표목차

제 1 장 서론.....	1
제 2 장 AGV의 기구부 구성.....	4
2.1 AGV의 기구부 설계.....	4
2.2 AGV의 구동부 설계.....	5
제 3 장 AGV의 전장설계.....	7
3.1 하드웨어 구성.....	8
3.2 전장 및 구동부 전원 구성.....	16
3.3 센서 부 선행실험.....	17
3.3.1 Kinect camera.....	18
3.3.2 RFID module.....	25
3.3.3 IMU sensor.....	29
3.3.4 proximity sensor.....	32
제 4 장 AGV의 주행 시스템.....	34
4.1 주행 시스템 연구.....	34
4.1.1 제어 흐름도.....	34
4.1.2 위치 제어 알고리즘.....	35
4.2 주행 테스트.....	39
제 5 장 결론.....	43
참 고 문 헌.....	44

# Development for Driving System using Fusion of Hybrid sensors of Automatic Guided Vehicle(AGV)

SHIN HEE YOUNG

*Department of Mechanical Engineering*

*Graduate School of  
Korea Maritime University*



## Abstract

This paper is about Driving System using Fusion of Hybrid sensors of Automatic Guided Vehicle for loading and unloading payloads. For the AGV to approach to the target rack for loading and unloading the payload, a way to identify the position and orientation was studied. To identify the position and orientation of the AGV accurately, a complex sensor system composed of RFID, IMU, and limit sensors was developed, and the performance of each sensor was undertaken. A model AGV was constructed, and the good performance of the developed complex sensor system was verified through performance experiments.

## 그림 목 차

Fig. 1.1 Intelligent Logistics Center.....	1
Fig. 1.2 Following the guided path line method.....	2
Fig. 2.1 Appearance of AGV.....	4
Fig. 2.2 Rotation performance of AGV.....	5
Fig. 3.1 Example operation of forklift for warehouse.....	7
Fig. 3.2 Rack surrounding and AGV approaching algorithm.....	7
Fig. 3.3 General arrangement if the AGV.....	8
Fig. 3.4 Embedded system-IEC 667 series.....	8
Fig. 3.5 Driver for motor control.....	9
Fig. 3.6 Kinect camera.....	11
Fig. 3.7 Image of proximity sensor.....	12
Fig. 3.8 RFID module.....	13
Fig. 3.9 IMU sensor.....	14
Fig. 3.10 Lithium Polymer battery.....	15
Fig. 3.11 H/W power system.....	16
Fig. 3.12 Rack surrounding and AGV approaching algorithm.....	17
Fig. 3.13 sensor architecture of kinect camera.....	18
Fig. 3.14 performance of kinect camera.....	18
Fig. 3.15 NUI Library analysis.....	19
Fig. 3.16 Architecture analysis of kinect SDK.....	19
Fig. 3.17 Real camera location.....	20
Fig. 3.18 Experimental environment.....	20
Fig. 3.19 RGB to HSV.....	21
Fig. 3.20 The input image.....	22
Fig. 3.21 Noise reduction and image binarization.....	22
Fig. 3.22 final recognition of Pallet.....	23

Fig. 3.23 pallet recognition.....	23
Fig. 3.24 Palette size and distance-aware.....	24
Fig. 3.25 rack data recognition.....	26
Fig. 3.26 view of Experimental environment.....	26
Fig. 3.27 RFID module GUI.....	27
Fig. 3.28 RFID module algorithm.....	27
Fig. 3.29 Experimental setup for RFID module.....	28
Fig. 3.30 Measurement of IMU sensor.....	31
Fig. 3.31 Application of ultrasonic sensor.....	32
Fig. 3.32 Static test of ultrasonic sensor.....	32
Fig. 4.1 A control flowchart.....	34
Fig. 4.2 A control flowchart for each module.....	35
Fig. 4.3 Trapezoidal velocity profile.....	36
Fig. 4.4 Experimental environment.....	40
Fig. 4.5 Performance test.....	41



## 표 목 차

Table. 2.1 Specification of AGV.....	6
Table. 3.1 Specification of Embedded system-IEC 667 Lite.....	9
Table. 3.2 Specification of DC Motor Driver.....	10
Table. 3.3 Specification of Kinect camera.....	11
Table. 3.4 Specification of ultrasonic sensor.....	12
Table. 3.5 Specification of RFID module.....	13
Table. 3.6 Specification of IMU sensor.....	14
Table. 3.7 Specification of each Lithium Polymer battery for motor control.....	15
Table. 3.8 Specification of Lithium Polymer battery for control board.....	16
Table. 3.9 Recognition results of pallet according to forward movement...25	
Table. 3.10 Recognition distance according to output(1.23mW).....	28
Table. 3.11 Recognition distance according to output(1.59mW).....	29
Table. 4.1 result of the performance test.....	42



# 1. 서론

최근의 주요 물류정책으로 저탄소 녹색성장을 견인하는 녹색물류 강국 실현을 위해 저탄소형 물류산업 구조로 개편, 고효율의 하역장비와 지능형 물류센터 개발을 추진하고 있다.[1] 기후변화협약을 기점으로 물류시설의 자동화 기술, 물류 IT의 지능화, 물류시스템 운영기술, 인터모달 기술 등 첨단 물류기술분야에서의 R&D 개발에 노력하며, 주요 물류 기업에서는 글로벌화 및 녹색물류로의 전환 등으로 새로운 패러다임으로 전환을 시도 중이다.[2][3]

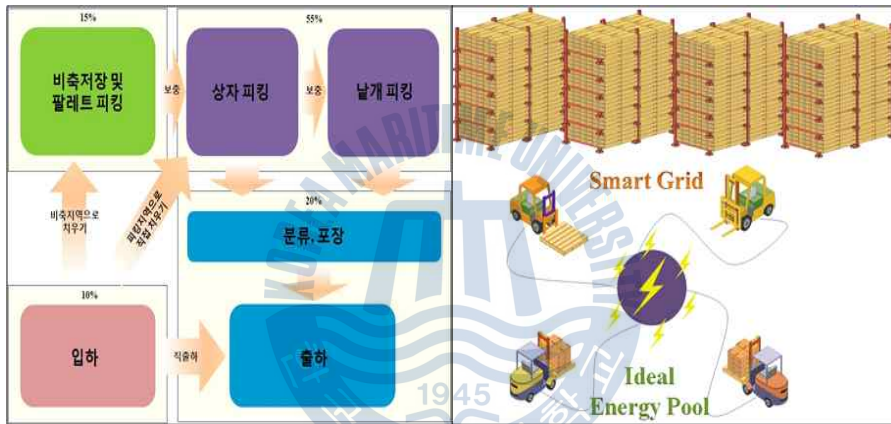


Fig 1.1 Intelligent Logistics Center

고객의 다양한 수요에 의한 물류분야의 다품종 소량 화물의 다빈도 처리가 요구되며, 물류센터에서도 서비스향상 및 물류비용 감소를 위해 지능화를 바탕으로 한 신속한 하역 및 이송처리가 필요하다.[4][5] 본 연구에서는 교통체계효율화 사업을 통해 물류센터내의 주된 작업시간과 비용이 요구되는 화물 하역 및 이송 처리 시 요구되는 자동화기술에 대한 연구개발로 무인지게차(이하 AGV)가 랙에 화물에 이 적재하기 위해 필요한 주행 시스템에 관해 연구한다.

AGV를 이용한 공장 물류자동화 시스템에서 운반차의 이동경로를 설정하는 방법은 운반차에 대한 Guided path line 추종방식이 주로 사용된다. 이러한 유도 시스템은 센서들을 바닥에 매설하는 방법으로 설치 및 유지보수가 어려운 문제

가 있었다. 하지만 완전한 자율주행이 가능한 유도 시스템은 작업 공간의 모든 환경을 인지해야하는 문제가 있기 때문에, 기존의 정해진 경로로 주행하는 시스템으로 변경할 필요가 있다. 기존의 정해진 경로로 주행하는 시스템은 한 작업장에서 다수의 자율 주행 장치가 이동할 시에 제한된 경로로 인해, 자율 주행 장치들 간의 교착 및 충돌 상황이 더욱 빈번히 발생하는 문제가 있다. 이러한 문제점을 해결하기 위해 기존에 사용된 방법으로는 AGV의 다중 경로 계획에 따른 트래픽 제어 방법이 있다.[6] [7][8]

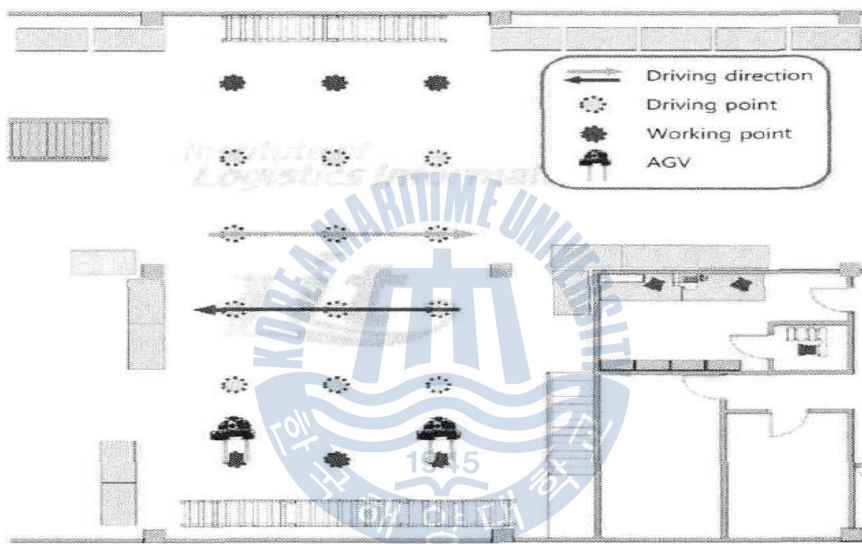


Fig 1.2 Following the guided path line method

위성항법(GPS) 기반의 주행방법은 지상 시설의 설치 없이 위성항법 기반의 위치 정보를 이용하여 주행경로를 설정하므로 기존에 비해 효율적이거나 주변 여건에 따라 위성항법 정보에 오차가 발생하거나 오차값이 발산하는 경우가 빈번하며 이를 보완하기 위해 레이저 내비게이션과 같은 전역위치 측정 센서를 이용한다. 하지만 레이저 내비게이션의 느린 반응속도와 고속 주행 시 주위의 반사체를 제대로 인식하지 못하는 문제로 인해, 실시간으로 위치정보를 받아 작업을 수행하는 AGV에 사용할 경우 안전성에 대한 문제가 나타날 수 있다. 이를 보완하기 위해 다른 부가장치들이 추가될 경우 시스템이 복잡해지고 비용을 증가시키는 단점이 있다.

또 다른 자율주행 기술로 자율주행 장치의 현재 위치를 추정하는 위치측정(localization)과 주위 환경을 인지하는 지도생성(map-building) 기술로 분류된다 [9] 위치측정 기술은 주로 전역위치측정 센서와 지역위치측정 센서의 융합을 통해 연구가 수행 되었으며, 최근에는 위치측정과 지도생성 기술을 함께 수행하는 SLAM(simultaneous localization and mapping)에 대한 연구가 활발히 진행되고 있다[10] 하지만 이러한 방법은 기술구현의 어려움과 센서들의 높은 비용이 문제가 된다.

따라서 본 논문에서는 AGV가 화물 이 적재를 위해 저가의 센서들을 사용하여 랙의 진입을 위한 효율적인 주행방법을 연구한다. Kinect camera, 엔코더, IMU (자이로 센서) sensor, RFID, 초음파 센서 등을 활용한 주행시스템을 설계하고 실험을 통해 이를 확인한다. 각 센서의 특성을 파악하고 성능실험을 거쳐 특성을 파악한다. 이들은 활용하여 주행 시나리오를 작성하고 설계된 주행 시스템의 주행테스트를 거쳐 결과를 도출한다.



## 2. AGV의 기구부 구성

Fig. 2.1은 주행시스템을 개발하기 위해 직접 제작한 차량의 외형을 보여준다. 차량의 크기는 750 X 1100 X 500[mm]이고 2개의 바퀴와 2개의 휠을 사용하여 기구부를 구성하였다.

### 2.1 AGV의 기구부 설계



Fig. 2.1 Appearance of AGV

본 연구에 사용된 차량은 DC24V전원과 평균전력 150W를 낼 수 있는 두 개의 maxon motor를 구동부로 사용하였고 주행제어의 용이성을 위해 두 개의 바퀴가 독립적으로 구동되도록 fig. 2.1과 같이 제작하였다. 제작된 실험용 차량은 주행제어를 위한 센서들이 부착되도록 여유공간을 가지도록 구성되었다.

## 2.2 AGV의 구동부 설계

순간 회전중심(Instantaneous center of rotation)이 양바퀴 축에 위치한 직선 위에서 이동하고, 그 위치는 양 바퀴의 회전 속도비에 의해 결정된다. 위와 같은 자율주행 장치의 오른쪽과 왼쪽 바퀴의 선속도  $v_R$ 과  $v_L$ 은 다음 식과 같이 나타낼 수 있다.

$$V_R = rW_R \cdot V_L = rW_L \quad (2.1)$$

(단,  $r$ 은 바퀴의 반지름,  $w_R$ ,  $w_L$ 은 바퀴의 각속도를 나타냄)

위 식에서  $w_R$ ,  $w_L$ 은 양 바퀴의 각속도를 나타낸다. 식(2.1)을 이용하여 자율주행 장치의 중심 선속도  $v$ 와 각속도  $w$ 는 다음과 같이 표현할 수 있다.

$$w = \tan^{-1}\left(\frac{V_R - V_L}{2L}\right)$$

$$v = \frac{V_R + V_L}{2} \quad (2.2)$$

(단,  $L$ 은 양 바퀴 사이의 거리)

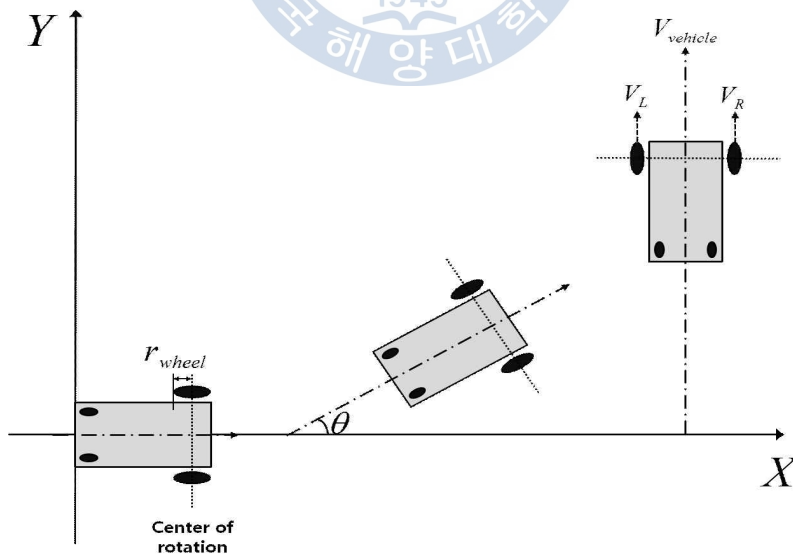
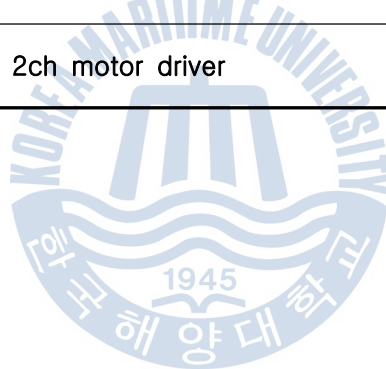


Fig. 2.2 Rotation performance of AGV

Table. 2.1 Specification of AGV

Composition	Specification		EA
Dimension	750 X 1100 X 500 [mm]		-
Drive head	Motor	150W DC maxon motor	2EA
	Wheel	Ball castor	2EA
Sensors	RFID module, IMU sensor, proximity sensor, encoder, Kinect camera		-
Control	Embedded PC, ATXmega		-
Driver	200W 2ch motor driver		1EA



### 3. AGV의 전장설계

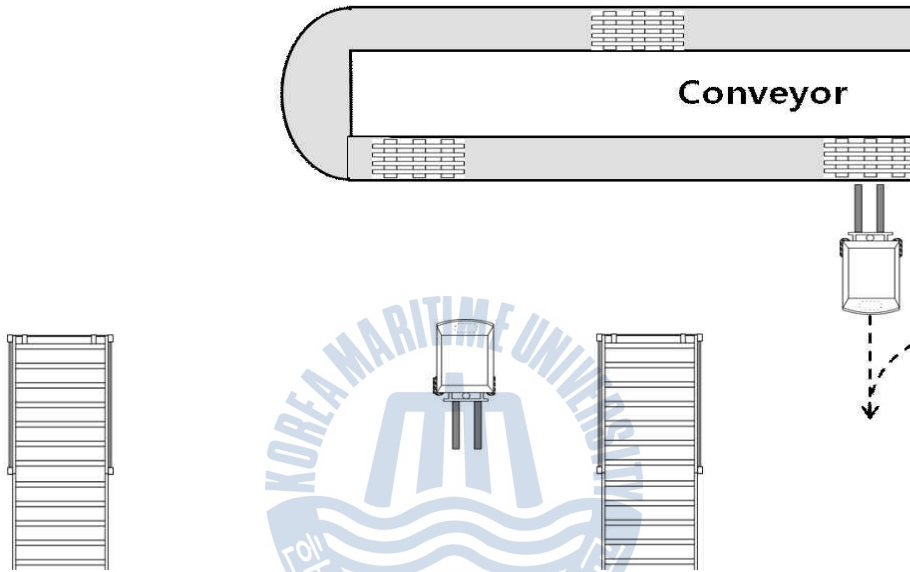


Fig. 3.1 Example operation of forklift for warehouse

Fig. 3.1은 자동화 물류창고에서 운행되는 AGV의 작업 모형도이다. 컨베이어 벨트에서 하역되는 화물의 적재명령을 할당받아 rack파트로 진입하게 되고 이러한 경우의 주행제어는 아래 Fig. 3.2 와 같다.

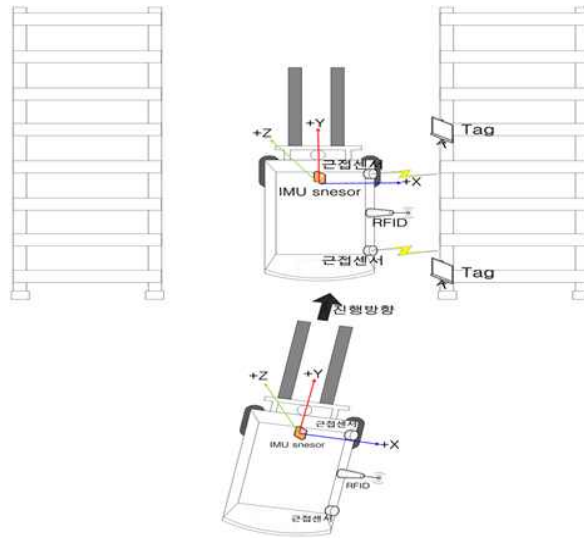


Fig. 3.2 Rack surrounding and AGV approaching algorithm

위와 같은 주행을 위해 실험용 차량에 센서들을 부착하였고 이에 따른 전장구성은 Fig. 3.3과 같다.

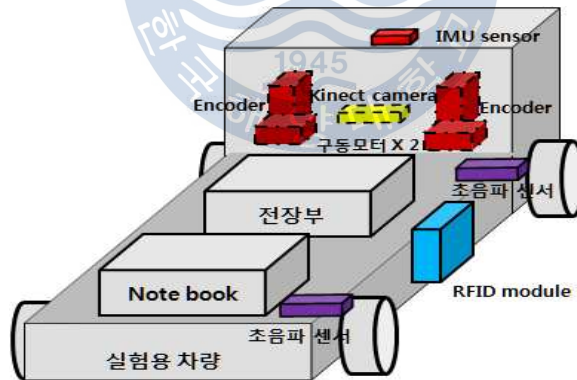


Fig. 3.3 General arrangement of the AGV

Kinect camera는 파렛트의 정확한 인식을 위해 바닥면과 최대한 가깝도록 차량의 전면부 아래에 부착되었다. 옆면에는 초음파 센서와 RFID module이 부착되고 전면 중앙부에는 IMU센서가 부착되고 중앙부에 나머지 제어기와 전장부분이 탑재된다.



### 3.1 하드웨어 구성



Fig. 3.4 Embedded system-IEC 667 series

AGV의 모터 제어를 위해 embedde pc를 사용 하였다. Embedded pc는 Windows CE 6.0 기반으로 Visual Studio series를 이용하여 C언어 기반으로 프로그래밍 할 수 있고 GUI개발 등을 통해 제어알고리즘 작성이 가능하다. 본 연구에서는 메인 제어기인 PC에서 제어 명령을 하달받아 RS-232통신으로 모터 드라이버로 제어명령을 송신하는 역할을 수행한다. 임베디드 컴퓨터의 제원은 Table. 3.1과 같다.

Table 3.1 Specification of Embedded system-IEC 667 Lite

Model		IEC667Lite	
System	CPU	32Bit RISC ARM1179JZF - 667MHz	
	RAM	256MB	
	Flash	NAND Flash : 128MB(OS : 50MB/Storage : 78MB	
	SD Memory	SD 지원	
	RTC	RTC 기능 내장	
	전원	DC 5V	
	동작 온도	-10 ~ 60°C	
Communication	RS232	2 ch	4 ch
	RS485	1 ch	
	TTL	1 ch	
	USB Host	1 ch - USB1.1	
	USB Device	1 ch - USB2.0	
	Ethernet	10 Mbps	

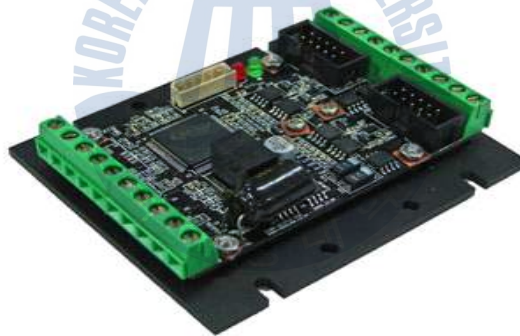


Fig. 3.5 Driver for motor control

본 연구에 사용된 구동 모터는 150W급 DC Motor 2개를 사용하였다. 기어비 10:1, 토크 170mNm 최대 속도 7580RPM으로 AGV의 특성상 높은 속도보다는 높은 토크를 요구하기 때문에 이와 비슷하게 모터 및 기어를 선정하였다.

이의 안정적인 구동을 위해 200W DC모터 드라이버를 적용하였다. AGV는 보통 완전한 평면상에서의 구동을 원칙으로 하나 바닥면의 장애물과 같은 요인에 의해 모터와 제어기에 과부하가 걸릴 위험이 있으므로 보다 높은 스펙의 드라이버

를 선정하였다. 사용된 모터드라이버의 제원은 Table. 3.2과 같다.

Table. 3.2 Specification of DC Motor Driver

SECTION	DETAILS	SPECIFICATION
MOTOR	Number of motor driver	2 ea
ELECTRICAL	Minimum supply voltage	10~30 V
	Nominal supply voltage	30/45/60 V
	continuous current	8 A
	continuous Power	200 X 2 W
	PWM frequency	20 ~ 100 KHz
	motor coil driving MOSFET	DirectFET
OPERATION MODE	speed control	speed range 0 ~ 30000 RPM
	real-time speed command	period of 2 ~ 100 ms
	acceleration/deceleration set	0
SENSOR	encoder	0
	analog hall sensor	0
INPUT/OUTPUT	analog input of digital I/O or fault output	4 X 2 ea
COMMUNICATION	RS232 speed	9600 ~ 115200 bps
	RS485 speed	upto 2.5 Mbps
	CAN speed	upto 1 Mbps
TEMPERATURE	operating ambient temperature	0 ~ 50°C
MECHANICAL	dimension	80 × 50 × 17 mm(L×W×H)
	wight	44g

파렛트 인식을 위해 일반적으로 사용되는 CCD camera가 아닌 Kinect camera를 사용하였다. Kinect camera는 일반적인 RGB영상 정보와 IR센서에서 제공되는 Depth image를 동시에 사용하여 파렛트의 크기 및 거리를 파악하는데 용이하다. IR센서에서 도트 형식으로 뿌려지는 3차원 거리 정보를 이용하여 RGB영상과 Depth image영상을 동시에 만들고 이의 정보를 제공한다. Kinect camera는 Microsoft사와 Asus사에서 제작이 되지만 개발자에 특화된 Asus사의 Xtion PRO 제품을 사용하였고 제원은 다음과 같다.



Fig. 3.6 Kinect camera

Table. 3.3 Specification of Kinect camera

Wavi Xtion Pro Live	
Field of View	58°H, 45°V, 70°D
Sensor	Depth & RGB
Depth Image Size	VGA(640 X 480) : 30fps QVGA(320 X 240) : 60fps
Supported platforms	Intel X86 & AMD
Xtion interface	USB 2.0
Supported OS	XP(Win 32/64) Win7 Linux Ubuntu 10.10 : X86, 32/64bit
Programming Language	C++/C#(windows) C++(Linux)
Operation Environment	Indoor

앞서 설명한 바와 같이 차량이 rack에 진입할 경우의 수평유지 주행을 위해 초음파 센서를 사용하였다. 기본적으로 20mm ~ 3500mm까지 거리파악이 가능하고 15°의 측정각도를 가지고 있다. 이의 제원은 다음과 같다.



Fig. 3.7 Image of proximity sensor

Table. 3.4 Specification of ultrasonic sensor

SECTION	DETAILS
Operating Voltage	DC 5V
Current	15mA
Operating Frequency	40kHz
Distance Range	20mm ~ 3500mm
Resolution	1mm
Measuring Angles	15Degree
Response Time	50ms
Size	45 X 20 X 15mm

랙의 인식시점 제공 및 정지시점 제공을 위해 사용된 RFID module이다. 적용이 용이하도록 안테나 내장형의 모듈을 사용하였고, tag의 감지범위를 출력에 따라 조절이 가능하여 tag의 오 인식을 방지하기 용이하다. 또한 tag가 부착되는 구조물의 재질에 따른 영향이 없어 광범위하게 사용 가능하다.



Fig. 3.8 RFID module

Table. 3.5 Specification of RFID module

SECTION	DETAILS
Scanner Model	R4008
Frequency Range	ISM 902~928MHz
RF Power Output	0 ~ 30dBm
Communication Ports	RS232
Reading Range	Reading Tag >5m
Power	DC 12V, less than 3A
Antenna	Built-in
Avoirdupois	1Kg

현재 차량의 position과 Heading 제어를 위해 사용된 IMU(자이로) 센서는 다음과 같다.



Fig. 3.9 IMU sensor

Table. 3.6 Specification of IMU sensor

SECTION	DETAILS
공급 전압	DC 5V
소모 전류	30mA
사용 온도 범위	-40 ~ 25
추가속도 측정 범위	±2G
비선형성	0.2%FS
최대 측정 각속도	2000 °/sec
자세 연산 대역폭	256Hz
출력 안정화 시간	80ms

전장 및 제어기에 전원 공급을 위해 2가지 종류의 배터리를 사용하였다. 모터 전원 공급용 배터리는 리튬폴리머 12.8V, 23Ah로 총 2개의 배터리를 직렬 연결하여 사용하였고 이는 Fig 3.10과 같다. 제어기 전원 공급용 배터리는 14.8V, 4.5A로 한 개만 사용된다.



Fig. 3.10 Lithium Polymer battery

AGV에서 소모하는 전력은 약 320W이다. 선정된 배터리로 AGV를 구동시 약 2시간 동안 운행이 가능하다. 일반적으로 AGV의 출력을 최대로 하지 않기 때문에 60%의 출력으로 지속적으로 운행 한다고 가정을 하면 2시간 이상 AGV의 운영이 가능하다. 이와는 별개로 제어기에는 안정적으로 제어 시에만 사용되도록 전원부를 따로 구성하여 제작하였다. 사용된 배터리의 제원은 아래의 표와 같다.

Table. 3.7 Specification of each Lithium Polymer battery for motor control

Rated Capacity		Typ. 23Ah
Nominal Voltage		12V
End Of Discharge		5.0V
Max. Charge Voltage		13V±0.03V
Max. Conti. Charge Current		20A
Max. Conti. Discharge Current		20A
Operation Temperature Range	charge	0 ~ 45°C
	Discharge	-20 ~ 55°C
Weight		375g



Table. 3.8 Specification of Lithium Polymer battery for control board

Rated Capacity		Typ. 4.5Ah
Nominal Voltage		3.7V
End Of Discharge		3.0V
Max. Charge Voltage		14.8V±0.03V
Max. Conti. Charge Current		4.5A
Max. Conti. Discharge Current		3A
Operation Temperature Range	charge	0 ~ 45°C
	Discharge	-20 ~ 55°C
Weight		111g

### 3.2 전장 및 구동부 전원 구성

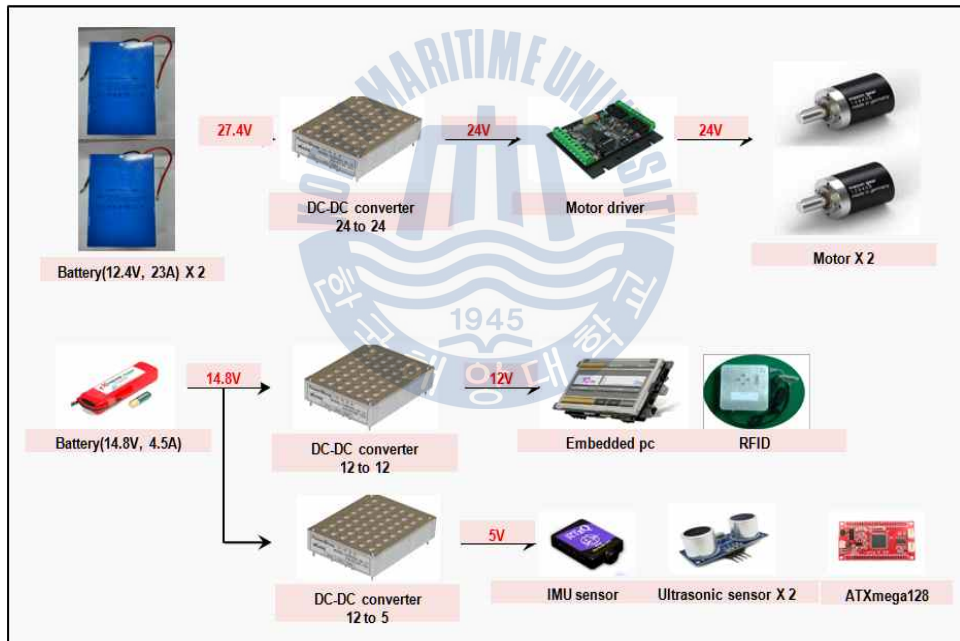


Fig. 3.11 H/W power system

모터 제어부는 12.4V 배터리 2개를 직렬 연결하여 사용하였고 제어기 부에는 14.8V배터리 1개를 사용하였다. 배터리의 전력 특성상 사용량이 증가할수록 전압이 떨어지는 부분을 보완하고자 DC-DC converter를 사용하여 안정적으로 공급되도록 구성하였다. 모터의 경우 24V구동이기 때문에 200W급 converter를 통

해 전원을 공급하였고 제어기는 30W급 converter를 이용해 각 센서부에 인가되도록 구성하였다.

### 3.3 센서 부 선행실험

물류창고에서 무인으로 운영되고 있는 AGV는 여러 가지 제어 기술의 집합체이다. 우리가 목표로 하는 기술은 기존의 주행기술과는 달리 AGV가 파렛트를 인식하는 시점을 시작으로 하여 화물 이 적재 장소까지 센서융합을 통해 이동하는 기술을 연구한다. 올바른 진입에 실패할 경우 랙의 기둥에 부딪혀 차량과 화물이 손상되거나 랙이 파손되는 경우가 생기기 때문이다. 이러한 랙 진입기술을 구현하기 위해서는 여러 가지 센서들을 이용한 AGV의 위치정보 및 방향정보의 파악이 필요하다. 자율주행을 위한 Kinect camera, 랙 정보의 파악 및 정지위치 인식을 위한 RFID 모듈, AGV의 Heading 제어를 위한 IMU센서, 랙 면과의 수평거리 인식을 위한 근접센서로 구성된 복합센서 시스템을 구성하여 직접 제작한 실험용 차량에 설치한다. 아래 그림과 같이 랙에 진입하기 전 IMU센서로 AGV의 현 자세를 파악하고 Heading 제어를 통해 진입을 시도한다. 진입과 동시에 두 개의 근접센서의 거리정보를 통해 수평을 유지하면서 직선주행하게 된다. RFID를 이용해 목표 랙을 찾게 되면 tag의 가장 가까운 인식 시점부터 정해진 거리만큼 이동하여 이 적재 명령이 할당된 목표 랙 위치에 정확하게 도달하게 된다. 본 연구에서는 이러한 복합센서 시스템을 구성하고 주행실험을 통해 이를 검증한다. 본 실험에 앞서 각 센서의 특성을 파악하기 위한 실험이 선행된다.

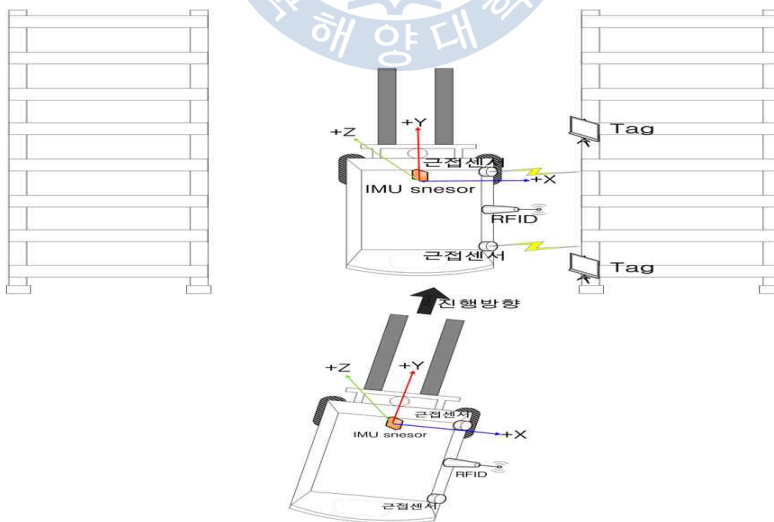


Fig. 3.12 Rack surrounding and AGV approaching algorithm

### 3.3.1 Kinect camera

먼저 적재 명령이 하달된 파렛트의 인식을 위해 Kinect camera가 사용된다. Kinect camera는 RGB와 Depth image를 동시에 사용하여 제어분야에 사용이 용이하고 내부 구조도와 NUI는 다음과 같다.

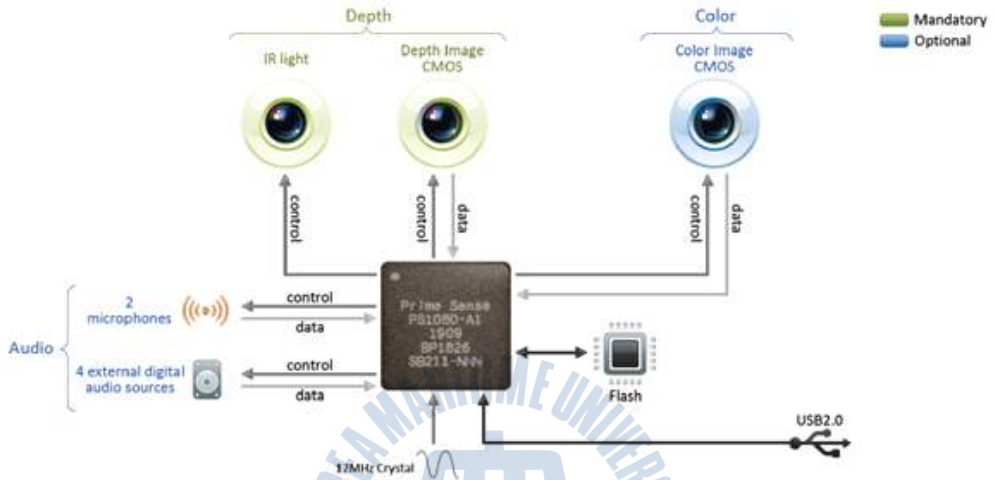


Fig. 3.13 sensor architecture of kinect camera

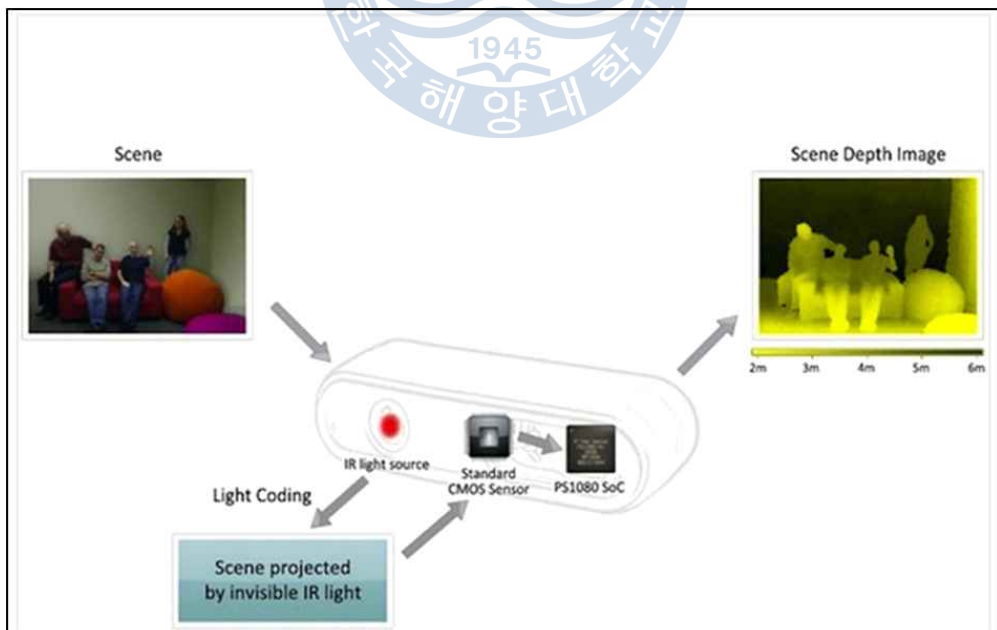


Fig. 3.14 Performance of kinect camera

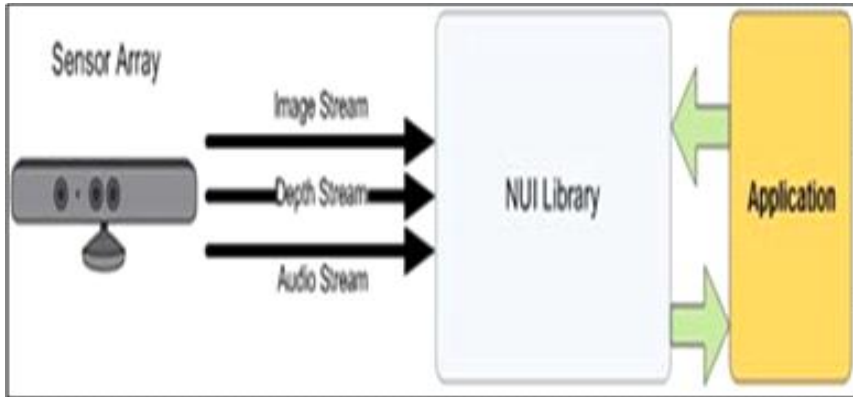


Fig. 3.15 NUI Library analysis

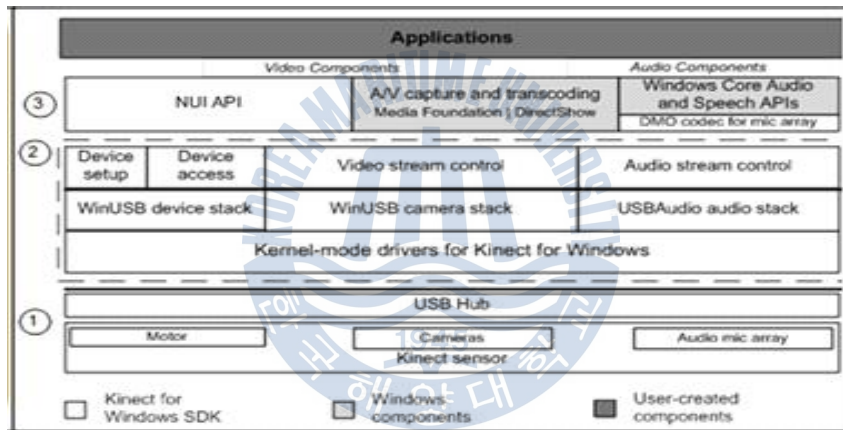


Fig. 3.16 Architecture analysis of Kinect SDK

본 연구에서는 한국 표준 파렛트의 앞면 부분의 형태학적 형상에 해당하는 사각형 성분들을 영상에서 추출하여 추출한 정보와 카메라의 깊이정보를 제공하는 센서를 동시에 사용하여 위치 및 자세를 측정한다. 지게차의 포크 캐리지 중간에 장착된 카메라를 이용하여 파렛트가 놓인 위치에 상관없이 인식 및 측정이 가능하도록 높이가 비슷한 실험용 차량 전면부 하단에 부착하여 측정한다.

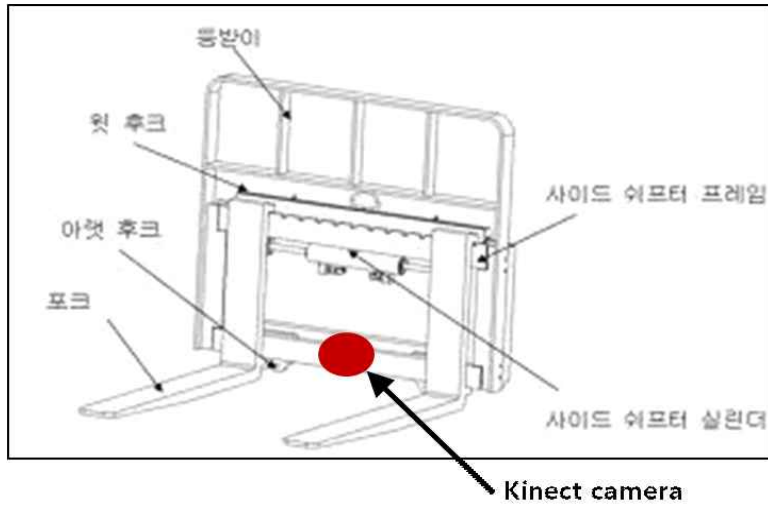


Fig. 3.17 Real camera location

차량에 부착하여 실험에 앞서 Microsoft visual studio 2008과 Open NI, Open CV라이브러리를 이용하여 알고리즘을 개발하고 이의 선행실험을 다음과 같이 수행한다.



Fig. 3.18 Experimental environment

파렛트의 특징 추출을 위해 입력 영상에서 3차원 파렛트를 **인식할여** 이미지 상의 위치와 영역을 추출한다. 파렛트를 인식하기 위해 표준 파렛트가 단색이고 사각형이며 두 개의 홀을 가진다는 것을 이용한다. 먼저 표준 파렛트가 단색임을 이용하여 거리에 따른 색상정보를 추출한다. 또 홀의 음영을 이용하여 어두운 영

역을 수집하고 수집된 컬러 분포를 이용하여 파렛트의 색상을 결정한 후 최종적으로 파렛트를 인식한다.

Color	R	G	B	H	H <sub>2</sub>	C	C <sub>2</sub>	V	L	I	Y <sub>601</sub>	S <sub>HSV</sub>	S <sub>HSL</sub>	S <sub>HSI</sub>
	1.000	1.000	1.000	n/a	n/a	0.000	0.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000
	0.500	0.500	0.500	n/a	n/a	0.000	0.000	0.500	0.500	0.500	0.500	0.000	0.000	0.000
	0.000	0.000	0.000	n/a	n/a	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	1.000	0.000	0.000	0.0°	0.0°	1.000	1.000	1.000	0.500	0.333	0.299	1.000	1.000	1.000
	0.750	0.750	0.000	60.0°	60.0°	0.750	0.750	0.750	0.375	0.500	0.664	1.000	1.000	1.000
	0.000	0.500	0.000	120.0°	120.0°	0.500	0.500	0.500	0.250	0.167	0.293	1.000	1.000	1.000
	0.500	1.000	1.000	180.0°	180.0°	0.500	0.500	1.000	0.750	0.833	0.850	0.500	1.000	0.400
	0.500	0.500	1.000	240.0°	240.0°	0.500	0.500	1.000	0.750	0.667	0.557	0.500	1.000	0.250
	0.750	0.250	0.750	300.0°	300.0°	0.500	0.500	0.750	0.500	0.583	0.457	0.667	0.500	0.571
	0.628	0.643	0.142	61.8°	61.5°	0.501	0.494	0.643	0.393	0.471	0.581	0.779	0.638	0.699
	0.255	0.104	0.918	251.1°	250.0°	0.814	0.750	0.918	0.511	0.426	0.242	0.887	0.832	0.756
	0.116	0.675	0.255	134.9°	133.8°	0.559	0.504	0.675	0.396	0.349	0.460	0.828	0.707	0.667
	0.941	0.785	0.053	49.5°	50.5°	0.888	0.821	0.941	0.497	0.593	0.748	0.944	0.893	0.911
	0.704	0.187	0.897	283.7°	284.8°	0.710	0.636	0.897	0.542	0.596	0.423	0.792	0.775	0.686
	0.931	0.463	0.316	14.3°	13.2°	0.615	0.556	0.931	0.624	0.570	0.586	0.661	0.817	0.446
	0.998	0.974	0.532	56.9°	57.4°	0.466	0.454	0.998	0.765	0.835	0.931	0.467	0.991	0.363
	0.099	0.795	0.591	162.4°	163.4°	0.696	0.620	0.795	0.447	0.495	0.564	0.875	0.779	0.800
	0.211	0.149	0.597	248.3°	247.3°	0.448	0.420	0.597	0.373	0.319	0.219	0.750	0.601	0.533
	0.495	0.493	0.721	240.5°	240.4°	0.228	0.227	0.721	0.607	0.570	0.520	0.316	0.290	0.135

The generic C-code for converting RGB color to HSV color is given below.

```

// r,g,b values are from 0 to 1
// h = [0,360], s = [0,1], v = [0,1]
// if s == 0, then h = -1 (undefined)
void RGBtoHSV( float r, float g, float b, float *h, float *s, float *v )
{
    float min, max, delta;
    min = MIN( r, g, b );
    max = MAX( r, g, b );
    *v = max;
    delta = max - min;
    if( max != 0 )
        *s = delta / max; // s
    else {
        // r = g = b = 0 // s = 0, v is undefined
        *s = 0;
        *h = -1;
        return;
    }
    if( r == max )
        *h = ( g - b ) / delta; // between yellow &
magenta
    else if( g == max )
        *h = 2 + ( b - r ) / delta; // between cyan & yellow
    else
        *h = 4 + ( r - g ) / delta; // between magenta & cyan
    *h *= 60; // degrees
    if( *h < 0 )
        *h += 360;
}

```

Fig. 3.19 RGB to HSV

Fig. 3.19의 변환 알고리즘을 이용해 입력영상의 색상영역을 수집하고 밝기영상으로 변환 후 깊이 정보를 이용하여 대상영역을 제외한 영역 즉, 노이즈에 해당하는 영역을 제거하는 최적화 작업이 이루어진다.



Fig. 3.20 The input image



Fig. 3.21 Noise reduction and image binarization



Fig. 3.22 final recognition of Pallet



Fig. 3.23 pallet recognition








최적화 작업 후 이진화 영상으로 변환하여 형태학적 정보를 이용 및 인식된 부분의 픽셀정보를 수집하여 파렛트 영역을 인식한다. 이 후 Depth image를 이용하여 대상과의 거리를 파악하고 이의 정보를 차량에 제공한다.



Fig. 3.24 Palette size and distance-aware

영상에서 보여지는 카메라와 대상 파렛트 간의 거리오차는 약  $\pm 10\text{mm}$  이내에서 안정적인 모습을 보여준다. 다음은 반복실험을 통한 오차를 나타낸다.

Table. 3.9 Recognition results of pallet according to forward movement

	1	2	3	4	5
Image					
recognition rate	97%	97%	97%	97%	97%
Distance	1m	1.25m	1.5m	1.75m	2m
Real Pallet Width	1.1m	1.1m	1.1m	1.1m	1.1m
Pallet Width	1.1637m	1.1634m	1.1747m	1.1856m	1.1845m
error rate	5.79%	5.76%	6.79%	7.78%	7.68%
Real Pallet Height	0.15m	0.15m	0.15m	0.15m	0.15m
Pallet Height	0.1717m	0.1678m	0.1651m	0.1703m	0.1606m

### 3.3.2 RFID module

물류창고의 랙을 인식하는 방법으로 물체 위치추적 기술인 RTLS(Real Time Location System)와 송수신기를 이용한 방법 등이 있다. 하지만 RTLS기법은 높은 설비비용과 전파방해에 약한 점들로 인해 상용화에 어려움이 따른다.[11] 이에 본 연구에서는 저가의 비용으로 설치가 단순하면서 랙 정보의 파악이 용이한 RFID module을 이용한다. 이러한 RFID module을 이용할 경우 다음과 같이 거리에 따른 RFID tag의 인식범위와 인식율을 고려할 필요가 있다.

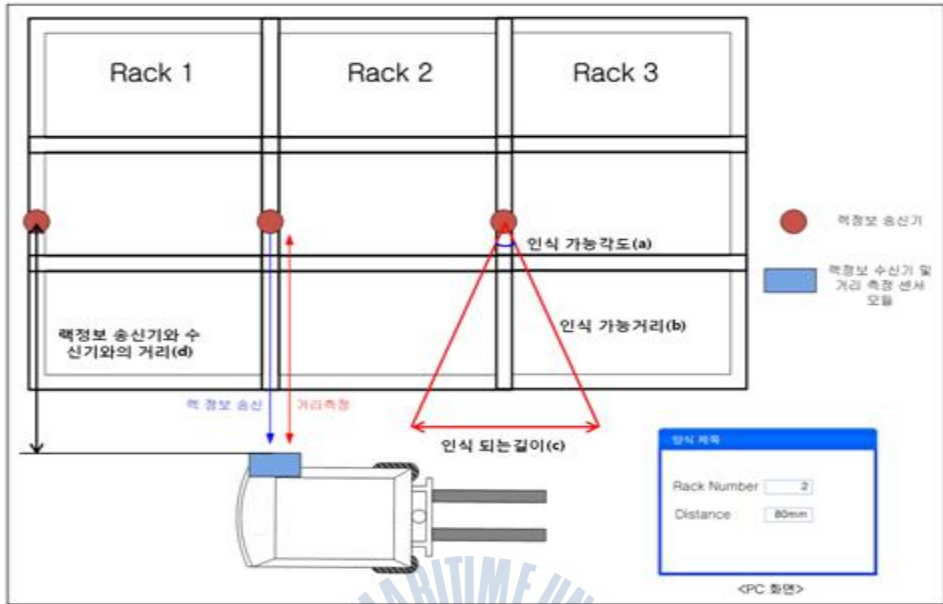


Fig. 3.25 rack data recognition

너무 광범위한 영역을 인식하는 경우에는 정확한 tag의 위치파악에 혼잡을 야기하고 인식범위가 좁다면 원하는 정보를 얻지 못하기 때문이다. 이에 RFID 리더기의 출력 주파수대역에 따른 인식범위를 파악하는 실험을 수행한다. 다음은 RFID module의 인식범위 파악을 위해 구성된 실험 환경을 보여준다.



Fig. 3.26 view of Experimental environment

RFID module의 출력에 따른 인식 범위 조정을 위한 GUI는 visual studio 2008을 이용해 개발하였고 실시간으로 파악이 가능하고 인식된 tag의 정보를 표시하도록 알고리즘 하였다.

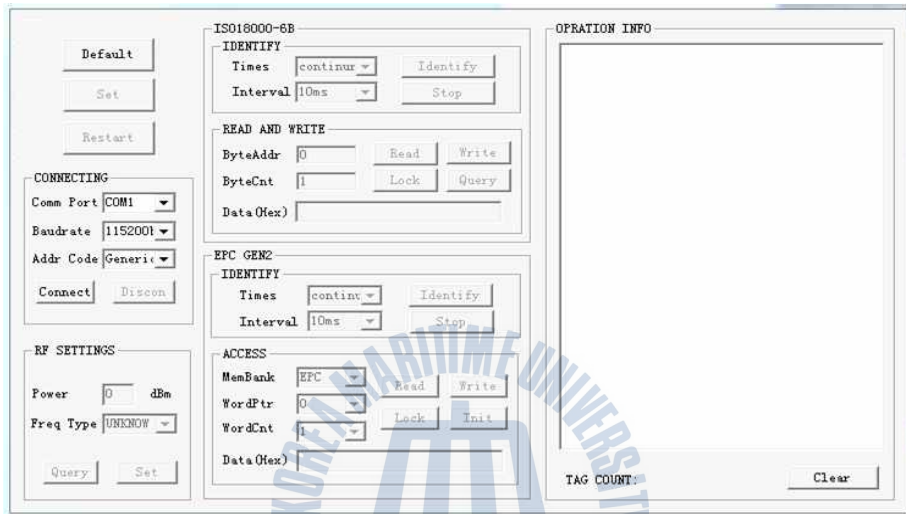


Fig. 3.27 RFID module GUI

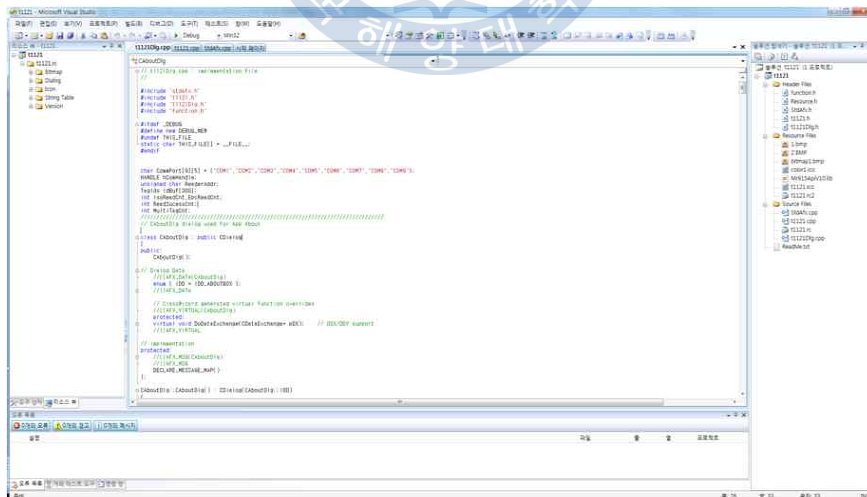


Fig. 3.28 RFID module algorithm

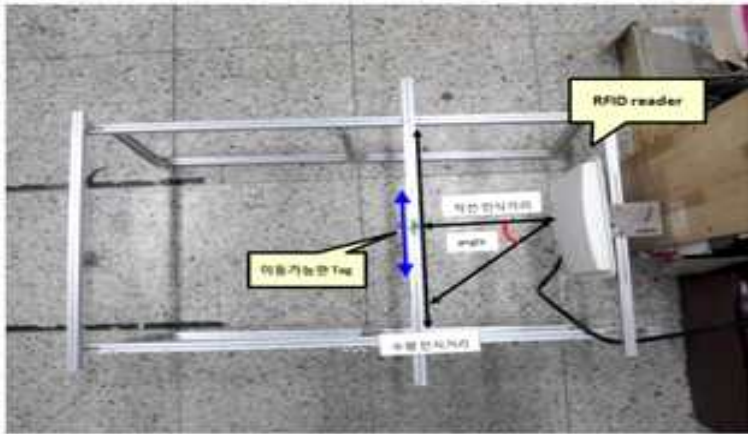


Fig. 3.29 Experimental setup for RFID module

RFID 리더기의 정확한 인식범위 파악을 위해 Fig. 3.23과 같이 눈금이 그어진 측정 장치에 직선 및 수평 이동이 가능하도록 tag를 부착하여 구성하였다. 구성된 장치는 직선거리 10cm - 100cm, 수평거리  $\pm 30$ cm까지 측정이 가능하다. RFID 리더기의 출력 주파수대역 조정과 tag에 정보입력을 위해 C언어 기반의 Microsoft Visual studio 2008을 이용해 프로그램을 제작하였고 RS-232통신을 이용해 PC로 계측한다. 제작한 프로그램을 이용해 RFID 리더기의 출력 주파수를 실시간으로 변경하여 각 거리영역에서 100번의 인식 시도 중 인식 성공 횟수를 기반으로 인식율을 측정하고 각 거리 정보를 이용해 인식 각도를 산출하여 실험결과를 도출하였고 이는 다음과 같다.

Table. 3.10 Recognition distance according to output(1.23mW)

출력 1dBm(1.23mW)			
직선 인식거리	인식율(%)	수평 최대 인식거리(mm)	각도(°)
100mm	100	$\pm 140$	54.46
110mm	100	$\pm 130$	49.76
120mm	100	$\pm 100$	39.85
130mm	100	$\pm 70$	28.30
140mm	100	$\pm 60$	23.19
150mm	53	$\pm 40$	14.93
160mm	37	$\pm 20$	07.12

Table. 3.11 Recognition distance according to output(1.59mW)

출력 2dBm(1.59mW)			
직선 인식거리	인식율(%)	수평 최대 인식거리(mm)	각도(°)
160mm	100	±130	39.09
170mm	100	±130	37.41
180mm	100	±120	35.83
190mm	100	±120	32.28
200mm	100	±120	30.96
210mm	100	±110	27.64
220mm	100	±110	26.56
230mm	100	±100	23.50
240mm	100	±90	20.56
250mm	100	±90	19.80

출력이 1dBm(1.23mW) 일 때, 직선 인식거리를 살펴보면 RFID 리더기와 tag가 140mm이상으로 멀어지면 인식율이 급격히 감소하는 것을 확인할 수 있다. 이 때 RFID 와 tag간의 수평 인식거리는 ±60mm로 실험용 차량이 tag위치에서 약 20°도 앞선 시점부터 인식하는 것을 알 수 있지만 1dBm의 출력 주파수로는 약 140mm 이상 떨어진 지점의 tag인식에는 한계가 있기 때문에 출력 주파수의 증가가 필요하다. 출력이 2dBm(1.59mW)일 때 직선 인식거리의 변화를 살펴보면 200mm이상까지도 인식율이 변함없이 100%임을 확인할 수 있다. 실험용 차량의 RFID 리더기와 모형 랙에 부착된 tag의 간격이 약 200mm - 250mm임을 고려하여 최대 250mm까지 직선거리를 가변하며 실험하였다. 현 실험에서의 최대 직선 인식거리인 250mm일 때 수평으로의 최대 인식거리는 90mm이고 이때의 각도는 약 20°로 실험용 차량이 tag의 약 20°앞선 시점부터 인식이 가능한 것을 확인하였다.

### 3.3.3 IMU sensor

위의 Kinect camera와 RFID module을 이용한 선행실험을 확인하고 본격적인 차량 제어를 위한 센서 구동성능 실험으로 차량의 위치 및 제어를 위한 IMU센서와 초음파 센서에 대해 연구하였다. 유인으로 제어가 가능한 지게차의 경우는 육안으로 지게차의 자세를 제어할 수 있지만 무인으로 구동되는 AGV의 경우는 차량의 위치 인식 및 자세제어는 매우 중요하다. 이러한 차량의 위치정보는 IMU 센서와 AGV 구동모터에 설치된 엔코더를 이용해 얻을 수 있다.

본 연구에서는 IMU 센서(자이로센서)와 엔코더 값을 이용해 목표점까지의 이동 위치 및 방향을 파악한다. 하지만 이러한 IMU센서는 외부영향에 의한 누적 오차가 발생되고 이의 정도를 파악하기 위한 특성파악이 필요하다. 현재 사용 중인 IMU센서는 16bit ADC를 통해 디지털 변환하여 필터링된 관성 데이터를 다중모드 확장 칼만필터(Extended Kalman filter)를 통해 노이즈를 최소화한 roll, pitch, yaw각도 출력 값을 얻도록 알고리즘되어 있다. 확장 칼만필터는 외부 입력인 Measurement matrix와 최종 출력인 Estimate matrix, 시스템 모델 F matrix, H matrix로 구성되며 예측 과정과 추정 과정으로 나뉘어진다. 예측 과정에서는 이전 Estimate matrix와 오차 공분산을 입력으로 받아서 현재 상태의 Estimate matrix와 오차 공분산을 구하게 된다. 예측 과정에서 사용하는 Estimate matrix와 시스템 모델 F matrix의 구성은 식(3.1)과 식(3.2)와 같다. 식(3.1)의 Estimate matrix는 4개의 쿼터니언 원소와 3축 속도 값으로 구성된다.

$$x = [q_0 \ q_1 \ q_2 \ q_3 \ v_x \ v_y \ v_z]^T \quad (3.1)$$

식(3.2)의 F matrix는 이전 상태의 Estimate matrix를 현 상태의 Estimate matrix로 연결 시켜주는 matrix로 구성된다.

$$F = \begin{pmatrix} I_{4 \times 4} & O_{4 \times 3} & O_{4 \times 3} \\ O_{3 \times 4} & I_{3 \times 3} & O_{3 \times 3} \\ O_{3 \times 4} dt & I_{3 \times 3} & I_{3 \times 3} \end{pmatrix} \quad (3.2)$$

추정 과정에서는 예측 과정에서의 Estimate matrix와 오차 공분산, Measurement matrix를 입력으로 받아 시스템 모델 H matrix를 이용하여 최종 Estimate matrix와 오차 공분산을 구하게 된다. 추정과정에서 사용하는 Measurement matrix와 H matrix 구성은 식(3.3)과 식(3.4)와 같다. 식(3.3)의 Measurement matrix는 IMU센서의 자세값과, DVL의 3축 속도 출력 값으로 구성된다.

$$z = (q_0 \ q_1 \ q_2 \ q_3 \ v_x \ v_y \ v_z) \quad (3.3)$$

식(3.4)의 H matrix는 Estimate matrix를 Measurement matrix로 연결 시켜주는 matrix로 구성된다.

$$H = \begin{pmatrix} 1000 & 0 & 0 & 0 & 000 \\ 0100 & 0 & 0 & 0 & 000 \\ 0010 & 0 & 0 & 0 & 000 \\ 0001 & 0 & 0 & 0 & 000 \\ 0000 & \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial z} & 000 \\ 0000 & \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial z} & 000 \\ 0000 & \frac{\partial h_3}{\partial x} & \frac{\partial h_3}{\partial y} & \frac{\partial h_3}{\partial z} & 000 \end{pmatrix} \quad (3.4)$$

구성된 센서의 정밀성을 위해 노이즈의 영향을 파악하였다. 다음의 그래프는 각 roll, pitch, yaw값을 서로 다르게 두고 한 시간 동안 정지시켜 두고 측정한 결과를 나타낸다.

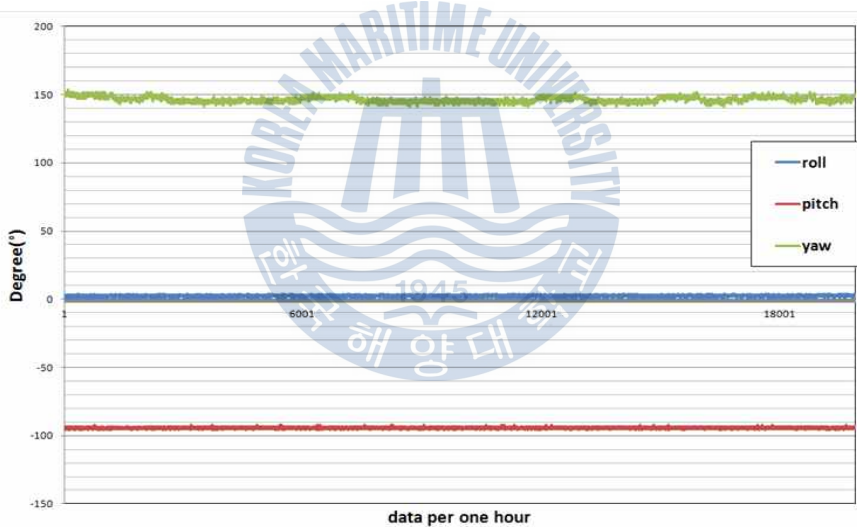


Fig. 3.30 Measurement of IMU sensor

정지 상태에서 roll각도 값은 기준 0.2°에서 ±0.1°오차, pitch각도 값은 -9.40°에서 ±0.1°오차, Yaw값은 -14.90°에서 ±0.2°오차를 보이는 것을 확인할 수 있다. Heading(yaw)값이 약간의 오차를 가지는데 이는 실험용 차량의 구동부 모터의 자기장에 의한 왜곡현상에 의한 것으로 추정되지만 영구적 발산은 일어나지 않아 제어에 큰 영향을 미치지 않는다는 것을 확인하였다.



### 3.3.4 proximity sensor

자동 물류창고 환경에서 AGV가 랙에 진입하여 화물 이 적재를 위해 이동할 때 랙과 수평을 유지하면서 주행하고 tag까지의 거리를 측정해야 한다. 관련 기술로는 비전시스템을 응용하는 방법, 라인트래킹 등의 방법이 있지만 가장 효율적인 기술로는 근접센서를 활용하는 방법이다. 따라서 본 연구에서는 근접센서의 한 종류인 초음파 센서를 이용해 tag까지의 거리정보를 파악하고 랙과 수평을 유지하면서 주행하는 방법을 사용하였다.

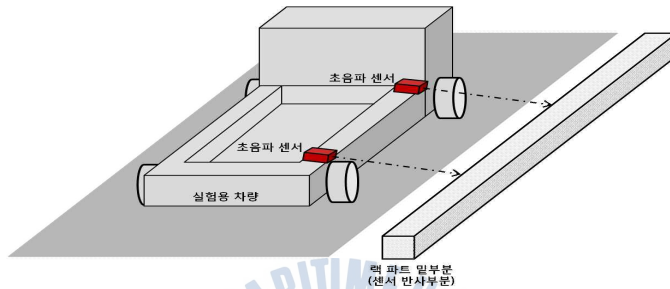


Fig. 3.31 Application of ultrasonic sensor

위와 같이 실험용 차량에 두 개의 초음파 센서를 1m간격으로 부착하고 모형 랙의 밑 부분을 감지하도록 하여 수평을 유지하면서 주행할 수 있는 장치를 구현한다. 이 경우 초음파 센서가 정확하게 랙의 밑 부분을 감지하지 못하면 수평을 유지하면서 주행하는 시스템이 구현되지 못하므로 초음파 센서의 감지범위 파악이 필요하다. 이를 위해 초음파 센서의 정적특성을 파악하였다.



Fig. 3.32 static test of ultrasonic sensor

위와 같이 초음파 센서의 감지범위를 확인하기 위해 초음파 센서가 부착된 센서부, A4용지 크기의 좌표계 그리고 초음파 센서의 거리정보를 계측하는 PC로 실험환경을 구성한다. 정적실험은 초음파 센서를 지면에 두고 센서의 중심부에서 200mm - 250mm사이 거리에서 센서의 감지범위를 파악한다. 큰 차이를 보

이지는 않지만 가로, 세로 약 100mm의 타원형 형상을 가진 인식범위를 가지는 것을 확인하였다.



## 4 AGV의 주행 시스템

### 4.1 주행 시스템 연구

#### 4.1.1 제어 흐름도

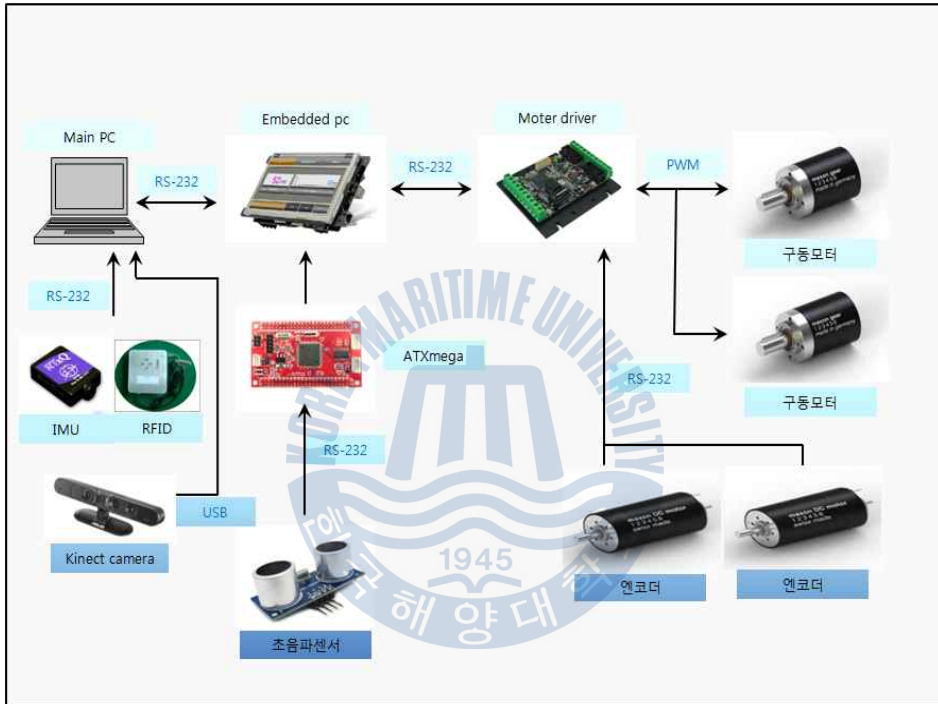


Fig. 4.1 A control flowchart

AGV의 주행 시스템에서 RS-232통신을 사용하는 장치는 IMU, 초음파센서, RFID, 임베디드 컴퓨터 및 모터 드라이버이다. 기본적으로 embedded pc에서 위치 data 및 속도 data를 시리얼 통신으로 모터 드라이버에 전송하면 그에 따른 PWM신호로 모터가 구동되고 엔코더를 통한 피드백으로 현재 AGV의 위치를 파악한다. 주행 시스템에서는 각 센서의 data를 시리얼 통신으로 ATXmega에서 전송받아 이를 메인 제어기인 embedded pc로 전송하게 되고 주행 알고리즘에 따라 모터 드라이버로 모터의 제어명령이 전달되어 모터가 구동된다. 이를 모듈 별로 표현하면 다음과 같다.

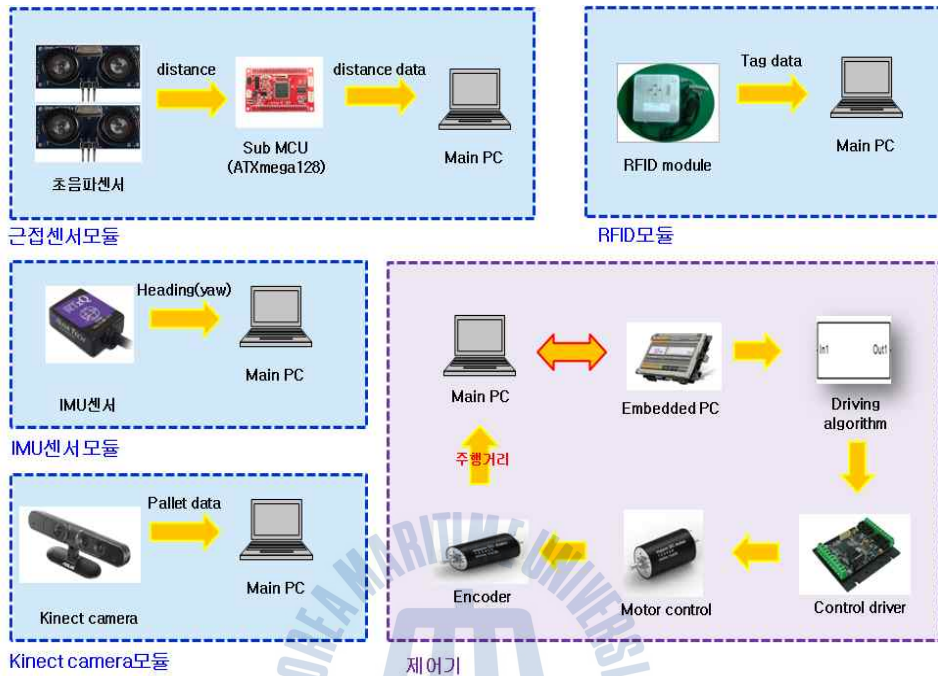


Fig. 4.2 A control flowchart for each module

#### 4.1.2 위치 제어 알고리즘

본 논문에서는 Fig. 4.3과 같이 보편적인 사다리꼴 속도 프로파일의 경로 궤적 방식을 사용하여 이에 상응하는 속도 및 위치 제어를 함으로써 모터를 제어하였다. 또한 초기 속도값이 존재하는 방식으로 경로 추종 중에도 새로운 경로 궤적으로 변경이 가능한 방식이다. 속도 및 위치 제어 알고리즘으로는 일반적으로 많이 사용하는 PID 제어 알고리즘을 사용하였다. 모터 모션 컨트롤러에 Fig. 4.3과 같은 경로 궤적 방식을 적용하기 위해서는 위치 경로 궤적 함수가 필요하다. 본 연구에서 궤적 추적 루프 타임을 1000Hz, 즉 1ms로 하여 이동 경로 궤적을 추종하도록 하였다.

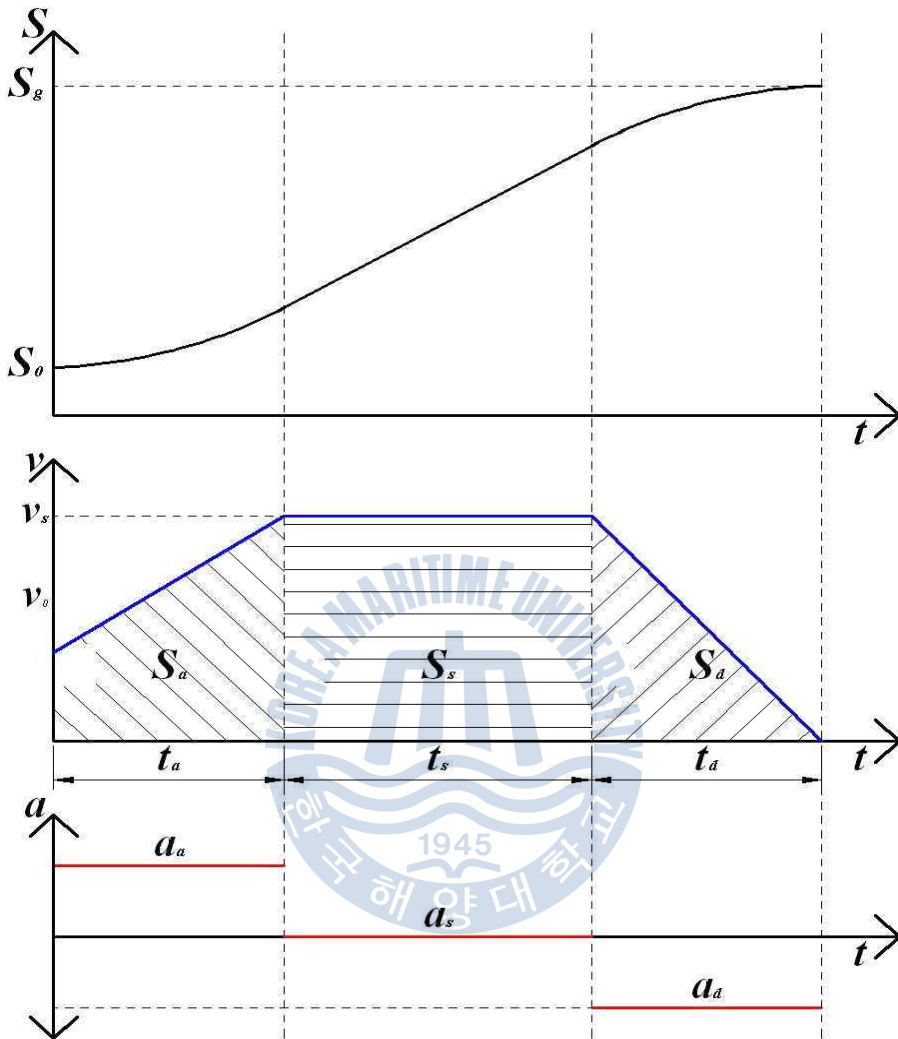


Fig. 4.3 Trapezoidal velocity profile

다음은 Fig. 4.3에서의 변수들의 정의이다.

$S_0$  : 초기 시작 위치,  $S_g$  : 최종 위치,  $S_t$  : 총 변위,

$S_a$  : 가속 기간 변위,  $S_s$  : 등속 기간 변위,  $S_d$  : 감속 기간 변위

$t_t$  : 총 제어 시간,  $t_a$  : 가속 시간,  $t_s$  : 등속 시간,

$t_d$  : 감속 시간,  $v_0$  : 초기 속도,  $v_s$  : 등속도

$a_a$  : 가속 기간 가속도,  $a_s$  : 등속 기간 가속도  $a_d$  : 감속 기간 가속도

최종 목표 위치  $S_g$  와 총 제어 시간  $t_t$  은 기본적으로 주어지는 위치 변경 명령에서 주어지는 값들이다. 또 초기 시작 위치  $S_0$  와 속도  $v_0$  는 현재 상태의 초기값들이다. 여기서 가감속 기간을 일정 비율적으로 정의하는 방법과 일정한 가감속도 정의하는 두가지 방법으로 위치 이동 궤적 함수를 생성할 수 있다. 본 연구에서는 가감속 시간을 일정 비율적으로 정의하는 방법으로 이동 궤적 함수를 정의하여 모션 컨트롤러에 적용하였다.

따라서 가속 시간  $t_a$ , 등속 시간  $t_s$ , 감속 시간  $t_d$  는 총 제어 시간에 따라 비율적으로 정의된 값으로 나머지 변수들에 대해 구해 보면 다음과 같다.

$$S_a = \frac{v_s + v_0}{2} \cdot t_a, \quad S_a = v_s \cdot t_a, \quad S_a = \frac{v_s \cdot t_d}{2} \quad (4.1)$$

$$S_t = S_g - S_0 = S_a + S_s + S_d = \frac{v_s(t_a + 2t_s + t_d) + v_0 \cdot t_d}{2} \quad (4.2)$$

식(4.2.1)과 식(4.2.2)로부터 다음과 같이 등속도가 정의된다.

$$v_s = \frac{2S_t - v_0 \cdot t_d}{t_a + 2t_s + t_d} \quad (4.3)$$

식(4.2.3)으로 구한 등속도로 각 기간별 가속도를 구하면 다음과 같다.

$$a_a = \frac{v_s - v_0}{t_a}, \quad a_d = \frac{v_s}{t_d} \quad (4.4)$$

식(4.2.1)에서 식(4.2.4)의 식들을 통하여 이동 궤적 함수를 구하기 위한 필요한 변수는 모두 구하였다. 이를 바탕으로 각 기간 별 속도함수와 위치함수는 다음과 같다.

1. 가속 기간 :  $0 < t < t_a$

$$v(t) = v_0 + a_a \cdot t \quad (4.5)$$

$$S(t) = S_0 + \frac{v_0 + v(t)}{2} \cdot t = S_0 + \frac{2v_0 + a_a t}{2} \cdot t \quad (4.6)$$

2. 등속 기간 :  $t_a < t < (t_a + t_s)$

$$v(t) = v_s \quad (4.7)$$

$$S(t) = S_0 + S_a + v_s(t - t_a) \quad (4.8)$$

3. 감속 기간 :  $(t_a + t_s) < t < (t_a + t_s + t_d)$

$$v(t) = v_s - a_d(t - t_a - t_s) \quad (4.3)$$

$$S(t) = S_0 + S_a + S_s + \frac{v_s + v(t)}{2} \cdot t = S_0 + S_a + S_s + \frac{2v_s - a_d(t - t_a - t_s)}{2} \cdot t$$

$$\text{또는} \quad S(t) = S_g - \frac{a_d(t - t)^2}{2} \quad (4.10)$$

여기서 가속도가 상수이고 속도 함수가 모두 1차식 이하로 이전 속도  $v_p$ 와 현재 속도  $v_p$ 로 하여 다음과 같이 간단하게 정의 할 수 있다.

1.가속 기간 :  $0 < t < t_a$

$$v_p = v_b + a_a \quad (4.11)$$

2.등속 기간 :  $t_a < t < ( t_a + t_s )$

$$v_p = v_s \quad (4.12)$$

3.감속 기간 :  $( t_a + t_s ) < t < ( t_a + t_s + t_d )$

$$v_p = v_b - a_d \quad (4.13)$$

이전 위치  $S_b$ 와 현재 위치  $S_p$ 에 대해서는 가감속 및 등속 기간 관계없이 다음 식으로 간단하게 정의된다.

$$S_p = S_b + \frac{v_b + v_p}{2} \quad (4.14)$$

모션 컨트롤러에 식(4.11)~(4.14)를 모터 모션 경로 궤적 알고리즘으로 적용하였다.

#### 4.2 주행 테스트

제안한 복합센서 시스템의 성능 분석을 위해 Fig. 7과 같은 공간에서 실험을 하였다. 물류창고의 랙과 유사하게 구현하기 위해 프로파일을 이용해 가로, 세로, 높이가 각각 3300mm, 1200mm, 1000mm로 모형 랙을 제작하였다. 모형 랙의



첫 번째 기동 넓이는 100mm, 밑 기동부의 길이는 1500mm, 두 번째 기동 넓이는 100mm, 두 번째 밑 기동부의 길이는 1500mm이다.



Fig. 4.4 Experimental environment

이동한 경로는 ① ~ ③ 구간으로 총 주행 길이는 6m이다. 실험 방법은 평균 0.1m/s의 속도로 ①구간을 3m 이동 후 ②구간에서 IMU센서 heading(yaw)값을 이용해 90°회전하여 ③구간(모형 랙)으로 진입한다. ② - ③구간에서 첫 번째 RFID tag를 인식함과 동시에 초음파 센서로 측정된 차량과 tag간의 거리정보와 앞서 실험했던 tag의 직선거리에 따른 수평 인식거리 실험값을 토대로 진입 시 tag인식시점부터 수평으로 이동할 거리정보를 제공받게 된다. 제공된 정보와 제어 알고리즘을 통해 차량이 화물을 이 적재하기 위한 곳까지 두 개의 초음파 센서를 이용한 수평유지 주행 후 정지하게 된다. 위의 주행시험을 5회 반복 수행하여 Table. 2의 결과를 얻었다. Table. 4.1은 복합센서 시스템을 이용한 자율주행의 결과 값으로 ① - ③ 구간을 이동해 목표지점인 첫 번째 랙의 중심부까지의 주행결과를 보여준다.



Fig. 4.5 Performance test

가장 이상적인 주행결과로는 화물을 이적재하기 위한 랙의 중심부 즉, 첫 번째 tag와의 거리가 800mm(랙 중심부)인 지점에서 멈추는 것이다. 실험 1은 랙 중심부와 위치오차가 31.6mm로 실험결과 중 가장 큰 오차를 보였고 실험 2는 23.5mm로 매우 작은 오차를 보이는데 이러한 오차는 아주 미소한 tag의 인식시점의 차이로 예상된다. tag의 인식시점이 주변 환경이나 노이즈에 의해 10 - 50mm정도의 변화를 보이면서 이로 인한 정지시점 제공에 오차가 생겨 최종적으로 AGV의 정지위치 또한 오차를 가지게 된다. 위의 반복실험 결과 평균 약 27mm의 오차를 가지는 것을 확인하였다. Fig. 4.4는 실제 주행실험으로 시작점부터 정지시점까지의 이동과정을 보여준다.

Table. 4.1 result of the performance test

	초음파 센서1	초음파 센서2	첫 번째 tag와의 거리	랙 중심부와의 오차율
1	232.3mm	233.1mm	831.6mm	31.6mm
2	233.7mm	234.3mm	823.5mm	23.5mm
3	240.4mm	241.1mm	826.7mm	26.7mm
4	237.5mm	236.0mm	825.6mm	25.6mm
5	239.4mm	238.8mm	827.4mm	27.4mm



## 5 결론

본 연구에서는 RFID/IMU/근접센서로 구성된 복합센서의 융합을 통해 물류창고에서 AGV가 랙에 진입 시 필요한 주행기술을 구현하였다. 실제 자동화 창고에서 운행되는 AGV의 주행기술과는 차이가 있으므로 랙에 진입할 경우를 타깃으로 하여 복합센서 시스템을 구현하였다.

제안된 복합센서 시스템을 설계하기 위해 각 센서들의 특성을 파악하였고 이의 결과 데이터를 기반으로 복합센서 시스템의 성능을 평가하였다. Kinect camera의 depth image의 3차원 거리데이터를 이용하여 파렛트의 홀을 인식하고 거리 정보를 성공적으로 파악하였고 RFID와 초음파 센서를 이용해 랙 정지시점과 수평유지 주행 장치를 완성하였다.

이들을 복합적으로 사용해 AGV가 랙에 수평을 유지하면서 정해진 위치에 정지하도록 알고리즘을 구성하여 주행테스를 거쳤다. 가장 이상적인 결과는 첫 번째 모형 랙 기둥에 부착된 tag위치에서 800mm 떨어진 랙의 중심부에 정지하는 것이다. 하지만 성능시험 결과 평균 27mm의 거리 오차를 가지는 것을 확인하였다. 이는 주변 환경이나 노이즈로 인한 RFID의 인식시점의 차이로 인한 것으로 예상된다. 하지만 이는 목표 랙의 길이에 대해 4%에 해당하는 오차 값으로 목표 랙의 길이에 비교하면 매우 적은 오차 값이다. 또한 RFID의 인식거리에 대한 보정 알고리즘을 통해 오차율을 개선할 수 있을 것으로 예상된다. 이는 실제 상황에서 매우 적은 오차를 갖는 결과로 제안된 복합센서 시스템으로 화물 이적재의 주행기술 적용에 매우 적합함을 확인하였다.

## 참 고 문 헌

- [1] Lee, M. Y.(2011), The Implications Based on Review to Foreign Green Logistics Industry, KRIHS, pp. 5-103.
- [2] Jung, K. H.(2010a), Kim, J. M., Do, J. C. and Kim, S. S., “Sensor Fusion of AGV using Extended Kalman Filter“, Proceedings of KIIS Fall Conference, Vol. 20, No. 2, pp. 287-288.
- [3] Jung, K. H.(2010b), Kim, J. M., Park, J. J., Kim, S. S. and Bae, S. I., “Line Tracking Method of AGV using Sensor Fusion“, KIIS, Vol. 20, No. 1, pp. 54-59.
- [4] Lee, S. Y., Kim, G. S. and Song, J. M.(2010), A Study on Advancement of Logistics Industry in Korea, Korea Maritime Institute, pp. 21-111.
- [5] Korean Intellectual Property Office(2011), Patent Survey Report for Development of Loading & Unloading and Transfer System Technology for Intelligent Logistics Center, pp. 39-45.
- [6] Kim, J. M., Huh, J. M. and Kim, S. S(2009), “Multiple Path-planning of Unmanned Autonomous Forklift using Modified Genetic Algorithm and Fuzzy Inference system”, The journal of the Korea Institute of Maritime Information & Communication Sciences / v.13 no.8, 2009, pp.1483-1490
- [7] Jung, K. H.(2010c), Kim, J. M., Kim, S. D. and Kim, S. S., “Traffic Control Method for AGVs using Path Tracking“, Proceedings of KIIS spring conference, Vol. 20, No. 1, pp. 391-393.
- [8] Yun, D. S.(2007) and Yu, H. S., “Extended Kalman Filter Design for Autonomous Navigation with GPS and INS Sensor System Fusion“, J. of the Korean Sensors Society, Vol. 16, No. 4, pp. 294-300.
- [9] Kim, J. M., Kim, Y. T., Kim, S. S, “Indoor Localization for Mobile Robot using Extended Kalman Filter”(2008) Proceedings of KIIS , Vol. 18, No. 5, pp. 706-711.

- [10] Kim, J. M., Jung, S. Y., Jeon, T. R., Kim, S. S., “SLAM based on feature map for Autonomous vehicle”, The journal of the Korea Institute of Maritime Information & Communication Sciences / v.13 no.7, 2009, pp.1437-1443
- [11] Choi, M. K.(2009), Kim, G. H. and Hong, B. H., “Location Identification Technique for Freight in Warehouse Environment Based on RFID Tag” Proceedings of Korea computer congress, Vol. 36, No. 1, pp. 91-95.

