



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](#)

공 학 석 사 학 위 논 문

장애물 회피를 위한 전방향 주행형
차량의 제어에 관한 연구



2011년 2월

부 경 대 학 교 대 학 원

메카트로닉스공학과

이 경 목

공 학 석 사 학 위 논 문

장애물 회피를 위한 전방향 주행형 차량의 제어에 관한 연구



부 경 대 학 교 대 학 원

메카트로닉스공학과

이 경 목

이경목의 공학석사 학위논문을 인준함.

2011년 2월

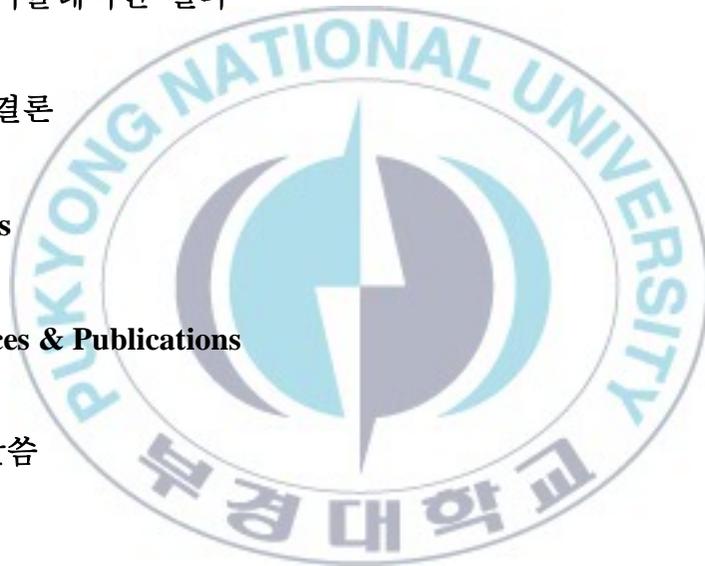


주 심 공학박사 정 영 석 (인)
위 원 공학박사 손 정 현 (인)
위 원 공학박사 김 상 봉 (인)

목 차

목차	i
Abstract	iii
제 1 장 서론	1
1.1 연구 배경 및 동기	1
1.2 연구 목적 및 방법	5
1.3 연구 내용 및 범위	7
제 2 장 천장설치형 카메라 시스템의 구성 및 차량과 장애물 위치인식	9
2.1 천장설치형 카메라 시스템의 구성	9
2.2 천장설치형 카메라 시스템을 이용한 차량과 장애물의 위치인식	12
2.2.1 색상인식 원리를 이용한 차량과 장애물의 위치인식 원리	13
2.2.2 Small Window Algorithm	17
제 3 장 전방향 주행형 차량의 구성 및 모델링	20
3.1 메카넘 휠을 이용한 전방향 주행형 차량의 구성	20
3.2 메카넘 휠의 구조 및 구동원리	24
3.3 전방향 주행형 차량의 제어시스템	27
3.4 전방향 주행형 차량의 모델링	29
3.4.1 전방향 주행형 차량의 운동학적 모델링	30

3.4.1 전방향 주행형 차량의 동역학적 모델링	37
제 4 장 장애물회피를 위한 경로계획 알고리즘 및 경로추종 제어기 설계	41
4.1 장애물 회피를 위한 경로계획 알고리즘	41
4.2 전방향 주행형 차량의 제어기 설계	48
4.2.1 경로추적 제어기 설계	48
4.2.2 속도추적 제어기 설계	52
제 5 장 시뮬레이션 결과	55
제 6 장 결론	63
References	64
Conferences & Publications	78
감사의 말씀	79



A Study on Control of Omni-directional Mobile Vehicle for Obstacle Avoidance

Gyeong Mok Lee

*Dept. of Mechatronics Engineering, Graduate School,
Pukyong National University*

Abstract

This dissertation proposes a tracking controller for obstacle avoidance of an omni-directional mobile robot(OMR) using a ceiling-mounted camera system method. The tracking controller is designed for the OMR to track a desired trajectory based on backstepping method using Lyapunov function. For design the tracking controller, the following tasks are implemented in this dissertation.

First, OMR, obstacles and a goal point must be separated and recognized for the OMR to avoid obstacles and reach the goal point using color recognition method and ‘Small Window(SW)’ algorithm. The SW algorithm has smaller FPS(Frame Per Second) value than the general image processing method.

Second, A* algorithm is proposed to generate a trajectory from the start point to the goal point for the OMR to avoid obstacles and arrive at the goal point using information of the OMR, obstacles and the goal point obtained from the proposed image processing algorithm. Using the cost function of the proposed A* algorithm, a shortest trajectory from a start point of the OMR to a goal point is generated.

Third, modelings for the OMR to track trajectory in 2D space is presented. First, kinematic modeling of the OMR is proposed using pseudo inverse matrix and dynamic modeling considering friction of the OMR's wheels are presented based on Lagrange equation. From the kinematic modeling, a backstepping controller is designed for the OMR to track the planed trajectory obtained from the A* algorithm. And the dynamic controller based on dynamic modeling is designed to make the real velocity of the OMR track the desired velocity that is the output of the kinematic controller.

Forth, to implement the proposed controller, an overall system is established with a ceiling-mounted camera system, OMR and obstacles. The ceiling-mounted camera system consists of ceiling-mounted camera and server computer. The OMR consists of actuator part, sensor part and control system. The actuator part consists of four mecanum wheels and four DC motors to implement omni-directional driving. The sensor part consists of compass sensor and rotary encoder. The compass sensor is used for measuring heading angle of the OMR and each rotary encoder is used for measuring each wheel's velocity. The control system is developed based on 8bit microprocessor PIC18F452 and PIC18F4431. The PIC18F4431 is used as master unit, and two PIC18F452s are used as slave unit.

Finally, simulation results are presented to show the effectiveness of the proposed trajectory planning algorithm and tracking controller for obstacle avoidance of the OMR.

제 1 장 서 론

1.1 연구배경 및 동기

20세기 급속한 산업화 이후, 이동 및 수송수단의 발전은 눈부신 속도로 이루어지고 있다. 현재 다양한 수의 바퀴(2~4륜, 6륜 등)을 이용한 차량이 개발되어 실용되고 있으며, 이와 함께 이동로봇에 대한 연구가 활발히 진행되고 있다.

일반차량의 조향 원리는 전륜(前輪) 및 후륜(後輪)에 장착된 조향 장치를 이용하여 회전과 직선으로 이루어진 경로를 주행하는 것이다. 이를 위한 연구는 현재 포화수준에 이르렀으며, 일반휠을 이용한 조향 시스템의 모델은 정형화되어가는 추세이다. 이에 따라, 보다 획기적인 조향 시스템에 대한 요구가 생겨났고, 최근 새로운 형태의 조향 시스템이 개발되었다. 그것은 전방향으로 주행 가능하도록 제작된 메카넘 휠을 이용한 조향 시스템으로 최근 여러 산업현장에서 적용되고 있다. 전방향 주행형 차량은 작업의 고효율성을 바탕으로 점차 산업현장에서 일반 휠을 대체해 나갈것으로 전망된다.

메카넘 휠을 이용한 전방향 주행형 차량에 대한 연구는 1970년대 초반에 메카넘 휠이 발명되면서 전방향 주행에 대한 연구가 시작 되었다. 스웨덴의 엔지니어인 Bengt Ilon이 메카넘 휠을 1973년에 발명하여 Ilon wheel이라 부르기도 했다. 메카넘 휠은 기존의 바퀴에 롤러를

45°방향으로 원둘레를 따라 정렬해 놓은 것이다. 이로 인해 기존 바퀴의 조향시스템과는 다른 전방향 주행이 가능해진 것이다.

이와 함께 차량의 경로 계획에 대한 연구 역시 활발하게 진행되고 있다. 김달형 등은 모델의 모양과 운동을 고려한 무충돌영역을 나타내는 Motion Space(M-space)의 개념을 이용한 자동 주차를 위한 경로 계획 방법을 제안했다[1]. M-space를 이용하여 제한된 모션의 연속적인 도달 가능 영역을 계산하고, 연속적인 영역을 역 전과 방법을 이용하여 경로 생성 가능 영역을 계산하며, 이 영역에서 하나의 점을 생성하여 최종경로를 생성시킨다. 생성된 경로에 경로의 선호조건인 안전, 조향장치의 조작횟수, 전후진 반복횟수, 경로거리를 고려하여 가중치를 부여하여 각경로마다 경로비용을 부여하여 최소비용의 경로를 선택하는 방법이다. 하지만 알고리즘이 복잡하고 시뮬레이션 결과도 충분치 않다. 김달형 등은 직선과 최소반경을 이용한 간단한 경로생성기법으로 경로를 생성시켜 생성된 목표경로를 운동학적 모델링에 기반을 두어 설계된 Kanayama의 궤적 추종기로 추적하는 방법을 제안하였다[2]. 하지만 초기 출발위치가 목표경로에서 멀리 벗어나 있는 경우에는 목표경로를 추종하지 못하며 또한 2륜 구동 차량에만 적용 가능하다. 손민혁 등은 주차공간탐지를 위한 제어기와 주차공간으로의 이동을 위한 경로생성 및 제어알고리즘을 제안하였다[3]. 제안된 제어알고리즘은 초음파 센서에 의해 탐지된 주차공간의 데이터를 이용하여 조향각과 주행속도를 제어하며, CCD카메라로 차선의 위치를 알아내어 자율 주행을 할 수 있도록 하는 제어시스템을 개발하였다. 이를 모형차량에 시험 적용하여 그 유효성을 검증하였다. 하지만 자율주차 동안의 영상시퀀스만 보여주었으며, 또한 차량의 조향각을 제어하여 주차공간으로 이동하므로 이동경로의 계산을 복잡하게 한다. 위에서 거론된 연구 이외에도 여러 전방향 주행형 차량의 경로생성에 대한 연구들이 보고되고 있다[4-14].

또한 계획된 경로를 정확히 추적하기 위한 전방향 주행형 차량의 제어에 관한 연구도 활발히 진행 중이다. 먼저 N. Tiale 등은 4륜 구동 전방향 주행형 메카닉 휠 이동로봇 대차의 운동학적 모델링과 동력학적 모델링을 제시하고, 2륜 ~ 4륜으로의 회전운동을 하기 위한 개념을 제시하였다[15]. 하지만 엔코더, 3축 자이로미터, 3축 가속도미터를 통합한 센서시스템으로 제시된 수학적 모델이 증명되었지만 구체적인 데이터를 제시하지 못했다. 또한 휠이 다양한 다른 속도를 가질 경우 이동대차의 거동을 제시하지 못했다. 또한 통합된 센서시스템으로 직선운동의 휠회전의 회전방향에 대해서만 제시하였다. S. L. Dickerson 등은 기구학적 모델링을 사용하여 이동공간을 계산하여 전방향 구동을 하였으며, 메카닉 휠의 롤러의 최대 및 최소반지름, 롤러의 수를 설계법을 제시하였다[16]. 하지만 시뮬레이션 결과가 부족하고 실험에 사용된 알고리즘 또한 불명확하다. P. Kiddee 등은 4륜 구동 전방향주행형 차량로봇의 운동학적 모델링과 Lagrange 공식에 의한 동력학적 모델링을 제시하였고, 엔코더와 스테레오 비전시스템을 이용하여 이동물체인 사람을 추종하는 전방향주행형 차량로봇의 제어를 제시하였다[17]. 비전센서를 사용하여 목표점인 사람의 위치를 파악하여 추적하도록 하였는데, 위치오차는 차량이 이동할 때 발생하였으며, 카메라 측정시스템의 불정확성으로 측정된 데이터와 엔코더로 측정된 데이터와 큰 오차가 생기는 결과를 보여주었다. Ahn Kim Tran 등은 천장설치형 카메라 시스템을 이용한 2륜 차량의 제어과 구현을 제시하였다[18]. 하지만 2륜 차량에 대한 모델로써, 메카닉 휠을 사용한 4륜 차량의 모델과는 맞지 않는다.

이외에도 현재 전방향 주행형 차량의 모델링 및 제어에 대한 많은 연구가 이루어져 있다[19-39].

위와 같이 메카닉 휠을 이용한 차량의 제어에 관한 연구와 그 구현에 대해 다양한 방법들이 연구되어 왔다. 위의 많은 연구로부터 메카닉 휠의 전방향 구동성은 입증되었고, 이를 이용한 산업현장에서의 적용범위가 점차 확대되고 있다. 하지만 전방향 주행형 차량에 대해 작업공간 전체를 파악하여 장애물 회피를 구현할 수 있는 천장설치형 카메라 시스템을 적용한 연구는 아직 수행되지 않았다. 따라서 미리 파악된 전체 작업영역에서 안전하게 장애물을 회피하고 계획된 경로를 정확하게 추적할 수 있는 전방향 주행형 차량의 제어에 대한 연구가 절실히 필요하다.



1.2 연구목적 및 방법

지금까지 일반휠을 사용한 차량에 대한 연구는 수없이 많이 이루어진 상태이며[40-53], 그 실용성을 바탕으로 실생활 및 산업현장에서 폭넓게 사용되고 있다. 또한 메카넘 휠을 사용한 차량의 연구 또한 이미 많은 연구가 이루어진 상태이나 실용성이 낮고 안정성의 부재로 인하여 일반휠에 비해 적용 범위가 좁고 활용도가 낮은 것이 사실이다.

위와 같은 단점을 극복하기 위해, 본 연구에서는 전방향 주행형 차량의 이점을 최대한 이용하면서 효율적이고 정확한 경로계획 및 경로추적 방법을 제시한다. 이를 위해 천장설치형 카메라 시스템을 이용한 전방향 주행형 차량, 장애물 및 목표지점의 위치인식법과 전방향 주행형 차량이 출발점에서 장애물을 회피하면서 목표지점까지 도착하기 위한 경로계획 알고리즘을 제시하고, 계획된 경로를 추적하기 위해 Backstepping 기법을 이용한 경로추적 제어를 제시한다.

차량, 장애물 및 목표지점의 위치인식을 위해 천장설치형 카메라 시스템을 이용하며, 다음과 같은 영상처리 알고리즘을 제시한다. 먼저 서로 다른 색상의 표식(Marker)이 부착된 차량, 장애물 및 목표지점을 식별하고 그 위치를 계산하기 위한 색상인식 기법을 제시한다. 또한 계획된 경로를 따라 주행하는 차량의 경로이탈 방지와 경로이탈 시 신속한 경로추종을 위한 'Small Window' 알고리즘을 제시하고, 일반적인 영상처리 기법과 비교해 'Small Window' 알고리즘의 효율성을 검증한다.

제안된 영상처리 알고리즘으로부터 얻은 차량과 장애물의 정보를 바탕으로 경로를 계획하기 위한 A*알고리즘을 제시한다. A*알고리즘을 이용함으로써 차량의 현재위치에서 목표지점까지 최단거리로 도달할 수 있는 경로를 계획할 수 있다.

그리고 A*알고리즘에 의해 생성된 경로를 정확히 추적하기 위해서 차량의 운동학적 모델링과 동역학적 모델링을 제시한다. 제시된 운동학적 모델링을 바탕으로 하여 Backstepping 기법을 이용한 경로추적 제어기를 설계한다. 또한 동역학적 모델링을 바탕으로 실제 차량의 속도벡터가 위치추적 제어기의 출력인 속도벡터를 추적하기 위해 역시 Backstepping 기법을 이용한 속도추적 제어기를 설계한다. 위에서 설계된 제어기를 구현하기 위한 천장설치형 카메라 시스템, 전방향 주행형 차량, 그리고 장애물로 구성된 전체시스템을 설치해야 한다. 첫째, 천장설치형 카메라 시스템은 작업공간의 영상을 확보하기 위한 천장설치형 카메라와 카메라로부터 전송받은 영상에 대해 영상처리를 수행하고, 경로생성 알고리즘을 수행할 Server Computer로 구성된다. 둘째, 전방향 주행형 차량은 다시 구동부, 센서부, 제어시스템으로 구성된다. 구동부는 전방향 주행을 구현하기 위한 4개의 메카넘 휠과 각 휠의 독립적인 구동을 위한 4개의 DC모터로 구성된다. 센서부는 경로추적 알고리즘에 의해 주행하고 있는 차량의 진행 방향각을 알아내기 위한 방위센서와 각 바퀴의 정확한 회전수 측정을 위한 Rotary Encoder로 구성된다. 제어시스템은 I2C 통신방식을 이용한 8bit 마이크로프로세서인 PIC18F4431(Master)과 PIC18F452(Slave)를 사용한다. 마지막으로 메카넘 휠을 이용한 전방향 주행형 차량의 장애물 회피를 위한 경로계획 알고리즘 및 설계된 제어기의 유효성을 검증하기 위해 시뮬레이션 결과를 제시하였다.

1.3 연구내용 및 범위

전방향 주행형 차량의 모델링하고, 이를 바탕으로 전방향 주행형 차량을 제어하기 위한 제어 알고리즘을 제안한다. 이를 천장 설치형 카메라 시스템에 적용시켜 장애물 회피를 위한 경로를 생성 및 그 구현에 대한 시뮬레이션과 실험을 통해 그 효율성을 검증한다. 한편 본 연구의 내용 및 구성은 다음과 같다.

제 1 장 : 본 연구의 배경 및 동기, 연구목적 및 연구방법과 본 연구의 구성 및 그 내용을 개략적으로 제시한다.

제 2 장 : 천장설치형 카메라 시스템의 구성과 이를 이용하여 차량과 장애물 및 목표지점을 인식하고 중심좌표를 획득하기 위한 색상인식 기법과 ‘Small Window’ 알고리즘을 소개한다.

제 3 장 : 전방향 주행형 차량의 구성과 운동학적 모델링과 동역학적 모델링을 제시한다. 또한 메카닉 휠의 구조적 특성 및 전방향 주행원리를 설명한다.

제 4 장 : 차량의 장애물 회피를 위한 경로계획법으로 A*알고리즘을 제시하며, A* 알고리즘에 의해 생성된 경로를 추적하기 위해 운동학적 모델링을 바탕으로 하여 Backstepping 기법을 사용한 경로추적 제어기를 설계한다. 또한 실제 차량의 속도벡터가 경로추적 제어기의 출력인 속도벡터를 추적하기 위하여 동역학적 모델링을 바탕으로 Backstepping 기법을 사용한 속도추적 제어기를 설계한다.

제 5 장 : A*알고리즘을 사용한 전방향 주행형 차량의 경로계획법과 설계된 경로를 추적하기 위한 운동학적, 동역학적 제어기에 대한 시뮬레이션 결과를 통하여 앞서 제안된 경로계획법과 제어알고리즘에 대한 유효성을 보여준다.

제 6 장 : 본 연구의 결론 및 향후 연구에 대한 내용을 간략히 기술한다.



제 2 장 천장설치형 카메라 시스템의 구성 및 차량과 장애물 위치인식

본 장은 천장설치형 카메라 시스템의 구성과 그 시스템을 이용한 차량 및 장애물의 위치인식 방법에 대하여 설명한다. 또한 이동중인 차량의 신속한 위치 추적을 위한 Small Window 알고리즘을 새롭게 제시하고 기존의 Full Window 알고리즘과의 비교를 통해 그 효율성을 검증한다.

2.1 천장설치형 카메라 시스템의 구성

Fig. 2.1은 천장설치형 카메라 시스템의 구성을 나타낸다.

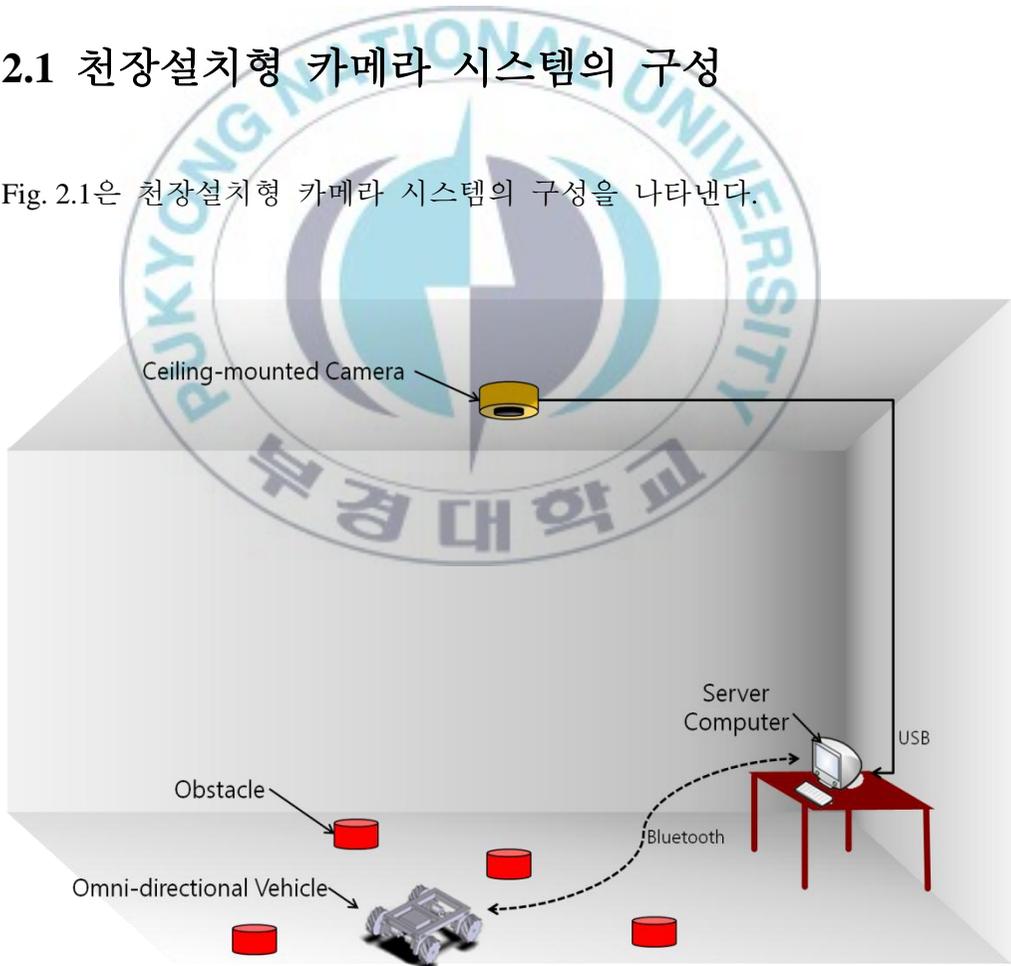


Fig. 2.1 Configuration of Ceiling-mounted Camera System

본 연구에서 제안하는 천장설치형 카메라 시스템의 구성은 다음과 같다.

- **Ceiling-mounted Camera**
: 천장에 설치되어 바닥면을 수직으로 주시하며, 실험공간의 영상을 USB 방식으로 Server Computer로 영상을 전송
- **Server Computer**
: 카메라로부터 전송받은 영상에 대한 영상처리 수행하고, 경로생성을 위한 알고리즘 계산 및 차량에 제어명령 전송
- **Omni-directional Mobile Vehicle**
: 전방향 주행이 가능하도록 메카넘 휠을 장착한 전방향 주행형 차량
- **Obstacle**
: 차량이 목표점을 따라 진행하는 경로상에 위치하는 장애물로써 일정한 크기를 가지며, 그 위치는 임의로 배치된다.

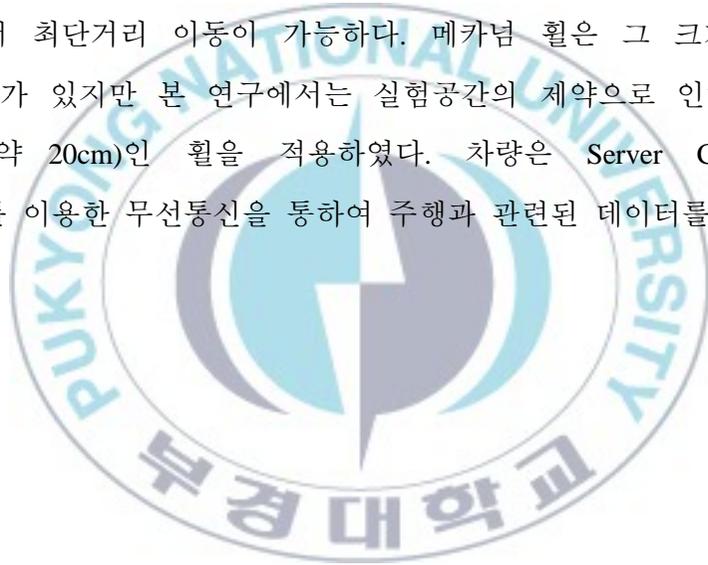
Fig. 2.2는 본 연구에서 사용한 천장설치형 카메라를 나타낸다. 천장설치형 카메라는 천장에서 바닥면을 수직으로 주시함으로써 실제 3차원 실험공간을 2차원 평면으로 단순화시켜 바닥평면만을 포함하는 영상을 획득할 수 있다. 카메라는 일반적으로 사용되는 USB 방식의 CMOS 센서기반 웹캠을 사용하였으며, 8mega의 화소수를 표현할 수 있으며, 1600x1200 pixel의 해상도를 지원하고, 초당 30fps의 영상획득이 가능하다.



Fig. 2.2 Camera for Ceiling-mounted Camera System

Server Computer는 2Ghz의 처리속도를 가지는 AMD 3500+ 프로세서를 탑재하고 있으며, 2Gbyte의 메모리를 장착하였다. 운영체제로 Window XP가 탑재되어 있으며, 카메라와 USB를 통하여 유선으로 연결되어 있다. 또한 카메라로부터 수신한 영상처리 및 알고리즘 계산을 위하여 Visual C++을 이용하여 작성된 프로그램을 구동시킨다.

본 연구의 차량은 목표지점까지 최단거리로 이동할 수 있도록 하기 위하여 일반적인 휠 대신 메카넘 휠을 장착하였다. 메카넘 휠을 장착한 차량은 별도의 조향장치 없이 네 바퀴의 회전방향을 개별적으로 조작해 줌으로써 방향전환이 가능하고, 대각주행 및 측면주행을 가능하게 해 줌으로써 최단거리 이동이 가능하다. 메카넘 휠은 그 크기에 따라 여러 종류가 있지만 본 연구에서는 실험공간의 제약으로 인해 직경이 7.5inch (약 20cm)인 휠을 적용하였다. 차량은 Server Computer와 Bluetooth를 이용한 무선통신을 통하여 주행과 관련된 데이터를 수신한다.



2.2 천장설치형 카메라 시스템을 이용한 차량과 장애물의 위치인식

본 장에서는 차량 및 장애물의 위치인식을 위하여 카메라를 이용한 영상처리 알고리즘을 제시한다.

2.1.1절에서는 색상인식 기법을 적용한 차량 및 장애물의 식별과 그 좌표획득에 관한 내용을 다룬다. 위치인식을 위한 방법으로 초음파 센서를 이용한 기법[54,55], RFID Tag를 이용한 기법[56-58], GPS를 이용한 기법[59-61], 카메라 센서를 이용한 기법[62-67] 등이 있다. 본 연구에서는 이 중 카메라 센서를 이용한 위치인식 기법을 적용하였다.

물체의 위치인식을 위해서는 먼저 목표한 물체를 식별하는 과정이 필요하다. 물체를 식별하는 방법에도 경계선검출 기법[68,69], 색상인식 기법[70,71], 형상인식 기법[72,73] 등이 있지만, 본 연구에서는 차량과 장애물의 색상인식 기법을 적용하였다.

본 연구에서는 장애물은 고정되어 있지만, 차량은 주어진 경로를 따라 주행하는 동적인 물체이다. 차량은 주어진 경로를 따라 주행해야 하기 때문에 매 시간 차량의 현재 위치를 파악할 필요가 있다. 만약 차량이 주행하는 도중에 주어진 경로를 이탈하였다면, 신속히 경로로 복귀할 수 있어야 한다. 이러한 이유로 영상처리 시 차량위치에 대한 신속한 추적이 필요하다. 이를 위해, 2.1.2절에서는 움직이는 차량의 위치를 신속히 추적하기 위한 **Small Window** 알고리즘을 제안한다. 이 알고리즘은 움직이는 물체를 신속히 추적할 수 있을 뿐만 아니라, 영상처리 시 소요되는 연산수를 줄여 제어주기를 단축시켜 주는 효과도 있다.

2.1.1 색상인식 원리를 이용한 차량과 장애물의 위치인식 원리

본 연구에서는 카메라 센서를 이용하여 차량과 장애물의 위치인식을 구현하였다. 차량과 장애물의 위치인식을 구현하기 위해서 다음과 같은 두 가지 과정이 필요하다. 첫째, 영상에서 차량과 장애물 및 목표시점을 식별하기 위해서 색상인식 기법을 이용하여 각각의 색상을 추출한다. 둘째, 추출된 색상의 좌/우, 상/하 경계점을 지나는 직선이 만나 만들어진 사각형(Mask)을 이용하여 그 중심좌표를 구한다.

먼저 차량, 장애물과 목표위치의 색상추출을 위하여 차량과 장애물 및 목표위치에는 각각 서로 다른 색상의 원형표식(Circular Marker)을 부착하고, 이를 카메라로 촬영하게 되면 차량과 장애물 및 목표지점은 부착된 표식의 색상으로 구분할 수 있게 된다. 이와 같은 기법을 색상인식 기법이라 하며, 그 기본적인 원리는 다음과 같다.

- 1) CMOS 센서의 경우 각 프레임의 영상정보는 한 픽셀(Pixel)당 3바이트의 RGB 데이터가 Blue, Green, Red의 순서로 저장된다.
- 2) RGB 각각의 색상은 8bit 수로 표현되며, 0~255로 표시되는 값을 가지며, 이 세 가지 색상의 조합으로 인간이 인식하는 컬러가 표현된다.
예) White(255,255,255), Black(0,0,0), Blue(255,0,0), Yellow(0,255,255) 등.

- 3) 사용자는 차량, 장애물 및 목표지점에 각각 다른 색상의 원형표식을 각각의 중심위치에 부착한다. 이 원형 표식들의 색상에 대한 RGB 데이터를 선정하여 영상처리에서 이 데이터를 이용한다.
- 4) 실제로 한 객체에 대한 표식은 한가지 색상으로 구성되어 있지만, 카메라를 통한 영상에서는 표식의 표면에서 반사되는 빛의 영향으로 표식은 단 하나의 색상이 아닌 비슷한 색상들의 영역으로 나타난다.
- 5) 비슷한 색상들이 포함된 표식을 식별하기 위해, 표식으로 인정할 수 있는 색상들에 대한 Threshold 범위를 지정한다.
- 6) 차량, 장애물 및 목표점이 포함되는 영상에서 Threshold 범위내의 색이 집중적으로 나타나는 영역을 추출해 내어 차량, 장애물 및 목표점의 식별해낸다.

색상인식 기법을 적용한 영상에서 차량과 장애물의 표식을 추출해 내면, 그 위치를 파악하기 위해 중심좌표를 구해야 한다. 이를 위해 영상을 2차원 좌표평면상에 나타낼 필요가 있다. 본 연구에서 카메라의 영상은 640x480 픽셀의 해상도를 적용하였고, 표식은 원형을 사용하였으며, 이를 2차원 좌표평면상에 나타내면 Fig. 2.3 에서 (0,0), (0,480), (640,0), (640,480)을 네 끝점으로 하는 Image Plane 위로 나타낼 수 있다.

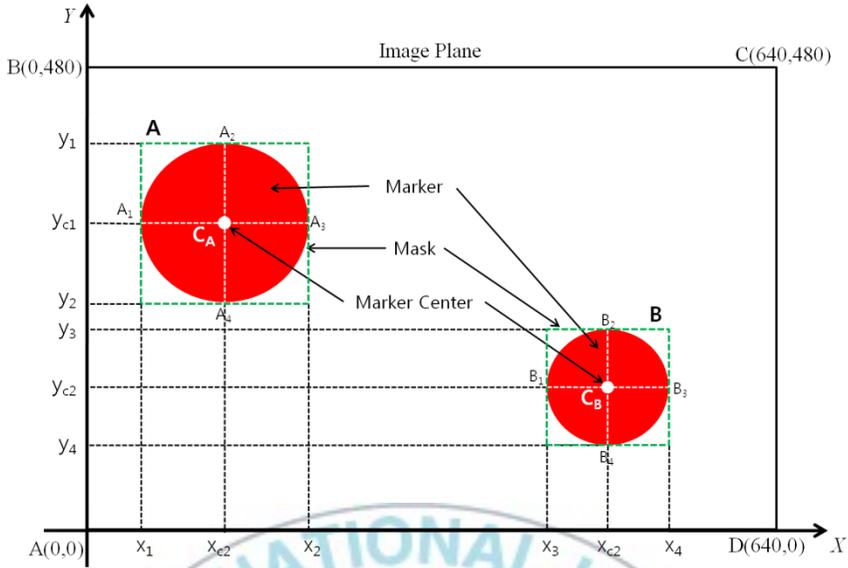


Fig. 2.3 Image Plane on 2D Coordination

Fig. 2.3 에서 표식의 정확한 중심좌표는 다음과 같은 과정을 통해 구해진다.

- 1) Image Plane의 원 A에 대하여, 먼저 상단에서 하단으로, 좌측에서 우측으로 픽셀을 분석하여 Threshold 범위 내의 색상이 연속적으로 나타나는 영역을 추출한다.
- 2) 추출한 영역의 좌/우, 상/하의 경계가 되는 점 A_1, A_2, A_3, A_4 을 구한다.
- 3) 2)에서 구한 경계점 $A_1 \sim A_4$ 를 각각 지나는 4개의 직선을 생성하면, 네 직선이 만나 하나의 사각형인 마스크(Mask)를 생성하게 된다.
- 4) 생성된 마스크의 기하학적 중심좌표 C_A 를 구한다.

5) 원 B도 위와 동일한 과정으로 중심좌표 C_B 를 구한다.

표식은 원형이므로 위의 과정으로 구한 중심좌표 C_A , C_B 는 표식의 중심점과 동일하며, 이 좌표를 이용하여 영상에서 표식의 위치를 알아낼 수 있다.

Fig. 2.4는 실제 실험공간에서 차량과 장애물 및 목표지점에 대하여 색상인식 기법을 적용한 영상을 좌표평면과 합성하여 나타낸 예이다.

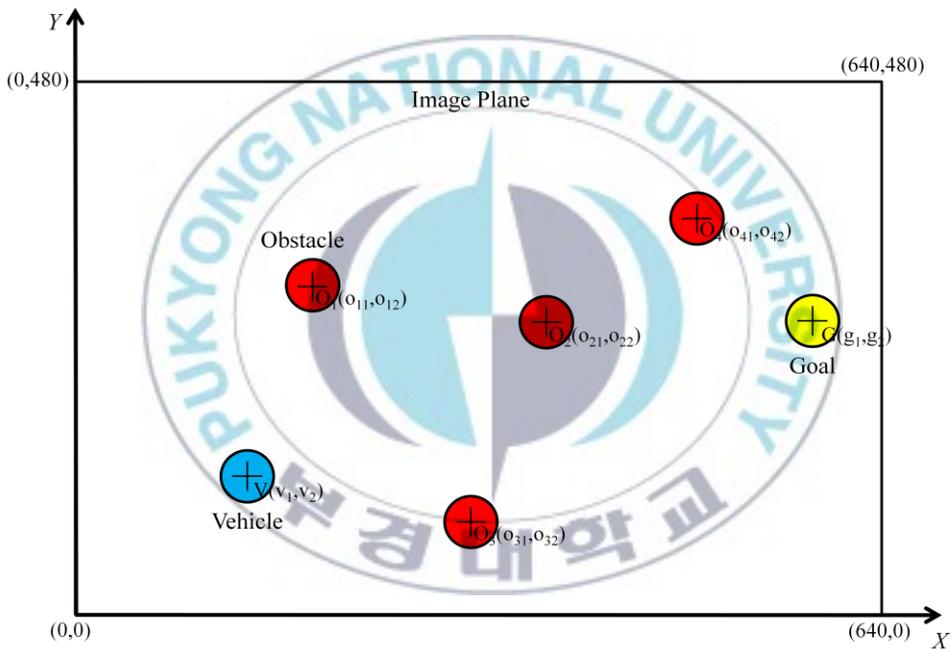


Fig. 2.4 Application of Color Recognition Technique to Ceiling-mounted Camera System

여기서, V 는 차량(Vehicle)의 좌표를 나타내고, $O_1 \sim O_4$ 는 장애물(Obstacle)의 좌표를 나타내며, G 는 목표지점(Goal)의 좌표를 나타낸다.

2.1.2 Small Window 알고리즘

차량이 경로를 이탈하지 않고 올바른 주행을 하기 위해서는 차량의 위치를 신속히 파악할 필요가 있다. 이를 위해 카메라는 매 프레임마다 영상처리를 수행하여야 하며 영상처리에 소요되는 시간은 차량의 경로오차를 줄일 수 있는 중요한 요소라 할 수 있다.

일반적으로 영상에서 원하는 데이터를 얻기 위해서는 각 프레임의 영상 전체를 분석한다. 본 연구에서는 640×480 픽셀의 해상도를 사용하였고, 이 영상전체를 분석하기 위해서는 $307,200(640 \times 480)$ 회의 연산이 필요하다. 또한 Server Computer의 프로세서는 경로생성 알고리즘의 연산을 함께 수행하여야 하기 때문에 연산수는 늘어나고 전체 연산시간은 길어지게 된다. 이는 차량에 지령을 내리는 제어주기와 직접적인 관계가 있다. 즉, 연산시간이 늘어날수록 제어주기는 길어지게 되며, 제어주기가 길어지면 경로를 이탈한 차량의 위치를 신속히 수정하기 어려워 진다.

이러한 문제를 해결하기 위해 본 연구에서는 Small Window 알고리즘을 제안한다. Small Window 알고리즘의 처리순서는 다음과 같으며, Fig. 2.5에 Small Window 알고리즘을 적용한 예를 나타내었다.

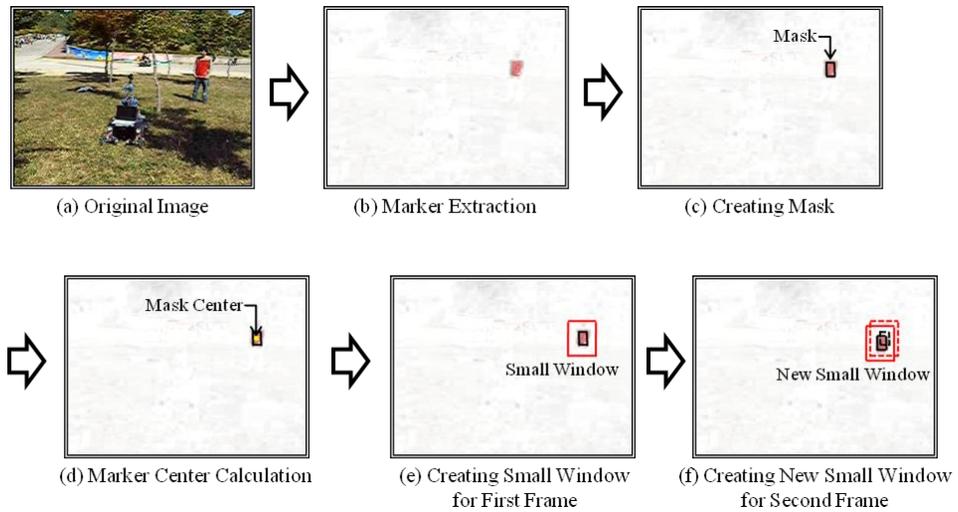


Fig. 2.5 Procedure of Small Window Algorithm

- 1) 카메라가 동작한 후 최초 프레임(First Frame)의 영상(a)은 전체 픽셀에 대하여 2.1.1절에서 제시된 색상인식 기법을 이용하여 차량의 표식을 추출하고(b), 마스크를 형성하여(c), 중심좌표를 구한다(d).
- 2) 1)에서 구한 차량 표식의 중심 좌표를 중점으로 하며, 표식의 가로, 세로 길이의 1.5배 크기의 Window를 새로 생성하여 그린다. 이 Window를 'Small Window'라고 정의한다(e).
- 3) 2번째 프레임(Second Frame)부터는 전체 픽셀이 아닌 Small Window 내부 픽셀을 우선적으로 분석한다.
- 4) Small Window 내부에 표식이 있을 경우, 표식에 대한 색상인식 기법을 적용하여 중심좌표를 구하고 2)의 과정을 반복하여 새로운 Small Window를 생성한다(f). 이전 프레임의 Small Window는 제거한다.

5) 4)에서 Small Window 내부 분석만으로 표식의 좌표를 얻을 수 있었으므로 현재 프레임에서 Small Window내부를 제외한 픽셀에 대한 분석은 생략할 수 있다.

6) 3)에서 Small Window 내부에 표식이 없다면 현재 프레임의 전체 픽셀에 대한 분석을 실시하여 현재 영상에서 차량 표식의 좌표를 파악한 후, 4), 5)의 과정을 반복한다.

본 연구에서 차량에 장착한 실제 표식의 크기는 30(cm) x 30(cm)이며, 3m천장에서 카메라를 통해 본 영상에서는 47 x 47픽셀로 나타난다. 표식을 포함하는 Small Window의 크기는 표식의 가로, 세로 길이의 1.5배로 설정하였으므로 71 x 71 픽셀로 표현되며, 그 면적은 5,041픽셀이 된다. 이는 전체 영상의 면적인 307,200픽셀의 약1.6%에 불과한 크기이다. 즉 전체 영상을 분석하는데 소요되었던 307,200회의 연산을 5,041회만 실시함으로써 표식의 위치를 파악할 수 있게 된다. 이는 경로계획 알고리즘에 대한 연산도 함께 수행해야 하는 Server Computer의 부하를 줄여주고, 차량에 지령을 내리는 제어주기를 단축시켜 줌으로써 신속한 제어가 가능하게 한다.

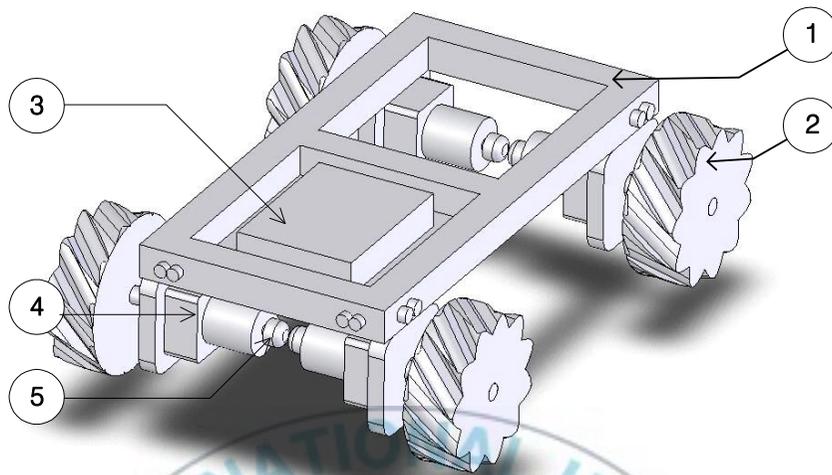
제 3 장 전방향 주행형 차량의 구성 및 모델링

본 장은 전방향 주행형 차량의 구성과 메카넘 휠의 구조적 특성 및 전방향 주행원리를 설명한다. 이를 바탕으로 전방향 주행형 차량의 운동학적, 동역학적 모델링을 제시한다.

3.1 메카넘 휠을 이용한 전방향 주행형 차량의 구성

전방향 주행형 차량의 전반적인 시스템 구성은 Fig. 2.1과 같이 나타낸다. 전방향 주행형 차량의 구성은 기구시스템과 제어시스템으로 나뉜다.

전방향 주행형 차량의 기구시스템은 Fig. 2.1에서 나타내는 것과 같이 직사각형 모양의 차체①에 4개의 구동 메카넘 휠②을 장착한 형태로 구성되어 있다. 제어시스템③은 마이크로컨트롤러부, 센서부, 모터 드라이버부로 구성되어 있다. 마이크로 컨트롤러부는 컴퓨터로부터 받은 경로 데이터에 근거하여 각각의 메카넘 휠에 맞는 속도값을 모터 드라이버부로 전달한다. 센서부의 방위센서는 0.1° 의 분해능을 가지는 CMP03을 사용하였으며, 모터 드라이브는 24V/3A까지 제어가능한 LMD18200을 4개 사용하였다. 또한, 모터④는 50:1의 감속기를 장착한 24V 3000rpm의 DC모터를 각 휠당 1개씩 장착하였고, 엔코더⑤는 3000pulse/rotate의 Incremental type을 사용하였으며, 모터의 회전속도를 측정하는데 사용된다.



-
- 1 . Body Frame
 - 2 . Mecanum Wheel
 - 3 . Control System
 - 4 . DC Motor
 - 5 . Encoder

Fig. 2.1 Configuration of Omni-directional Mobile Robot

Fig. 2.2는 본 연구에 사용된 실제 전방향 주행형 차량을 나타내며, Fig. 2.3은 본 연구에 사용된 실제 모터와 메카넘 휠을 나타낸다. 메카넘 휠은 전·후·좌·우에 4개가 장착되어 각각 DC모터에 의해 구동된다.



Fig. 2.2 Omni-directional Mobile Robot

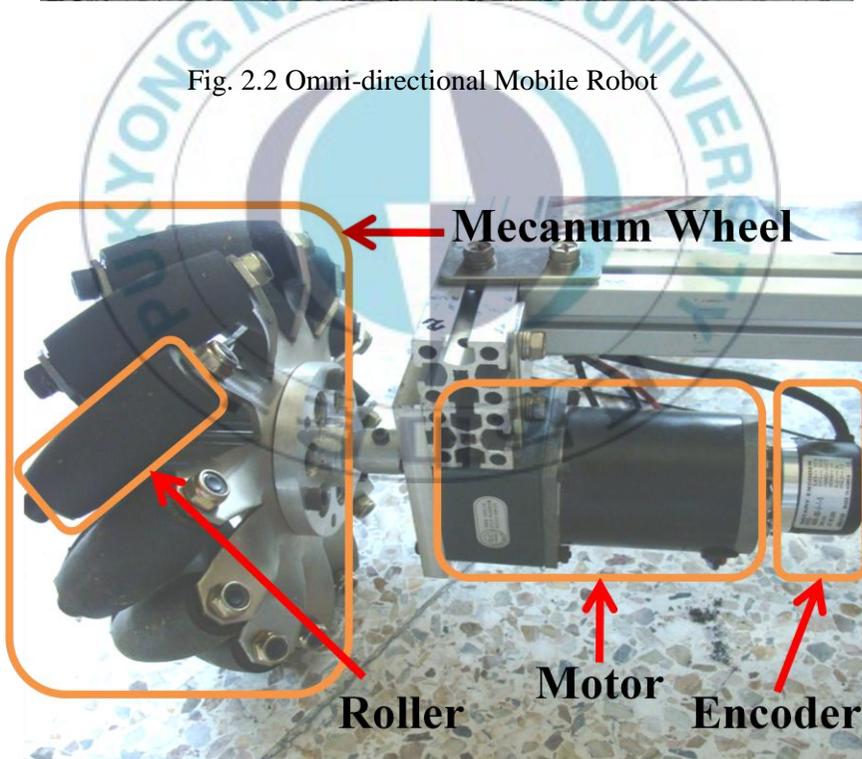


Fig. 2.3 Mecanum wheel and Motor of Omni-directional mobile Robot

메카넘 휠은 각 휠마다 12개의 롤러로 구성되어 있다. 45°의 각도로 위치한 롤러는 전방향 주행형 차량이 어떠한 방향이든 자유롭게 움직일 수 있도록 한다. 즉, 4개 휠은 속도 벡터의 조합에 따라 전방향 주행형 차량이 어떠한 방향이든 자유롭게 움직일 수 있도록 한다. 4개의 메카넘 휠을 구동하여 이동할 때 롤러의 미끄러짐(Slip)을 이용하여 전방향 이동이 가능해 진다.



3.2 메카넘 휠의 구조와 구동원리

본 절에서는 전방향 주행형을 구현하기 위해 사용한 메카넘 휠의 기구적 해석을 제시한다. Fig. 3.4는 다양한 직경의 메카넘 휠의 실제 외형들을 나타낸다. 본 논문에서는 7.5inch(약20cm)의 휠을 사용하였다.



Fig. 3.4 Configuration of Mecanum Wheel

일반적인 바퀴는 하중을 지탱하기 위한 휠과 휠의 바깥에 타이어를 결합한 형태가 일반적이다. 하지만 메카넘 휠의 경우, 휠의 바깥쪽에 있는 타이어 부분을 12개의 Free Roller 로 대체하고, 각 휠은 자신의 회전방향에 대해 45° 를 이루도록 설계되었다. 이는 역학적으로 휠 회전방향의 $\pm 45^\circ$ 방향으로 힘이 발생할 수 있도록 하기 위한 것이다.

또한 메카넘 휠은 차량의 좌측과 우측에 장착되는 휠이 좌우대칭을 이루도록 설계되어 있으며, 각각의 휠은 개별적으로 모터를 장착하고 있다. 각 휠의 회전방향과 회전속도를 독립적으로 다양하게 조절하여 전방향으로 주행할 수 있도록 하기 위함이다.

Fig. 3.5는 메카넘 휠이 X축을 중심으로 Y축 방향 정회전할 경우, 롤러에 의한 휠의 진행방향을 나타낸다.

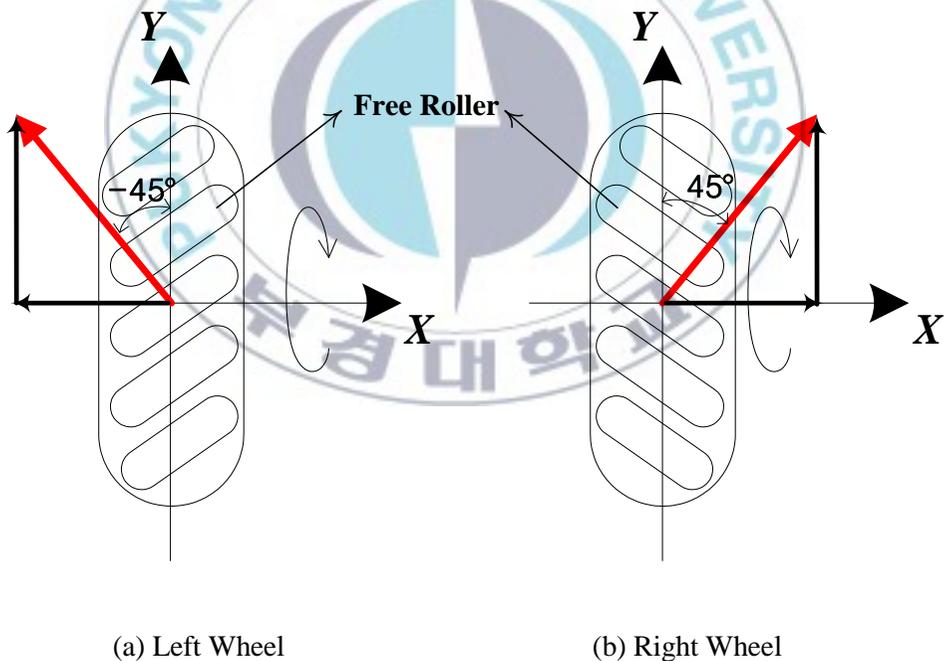


Fig. 3.5 Driving Mechanism of Mecanum Wheel

일반적인 휠의 경우, 액추에이터의 구동 시 발생하는 회전동력은 차량의 진행방향으로만 작용한다.

하지만 메카넘 휠의 경우, 휠의 회전동력은 Free Roller 에 의해 차량의 Y 축을 중심으로 대칭적으로 발생한다. 차량의 좌측 휠에서는 회전방향의 왼쪽 45° 방향으로, 우측 휠에서는 회전방향의 오른쪽 45° 방향으로 작용한다. 이것은 Y 방향의 속도성분뿐만 아니라 X 방향의 속도성분도 발생하게 됨을 의미한다. 이와 같은 원리를 전/후방, 좌/우측 4개의 바퀴에 적용시키게 되면 각 바퀴의 회전속도와 회전방향의 다양한 조합으로 전방향 주행이 가능해진다.



3.3 전방향 주행형 차량의 제어시스템

전방향 주행형 차량의 제어 시스템의 구성은 Fig. 3.6과 같다.

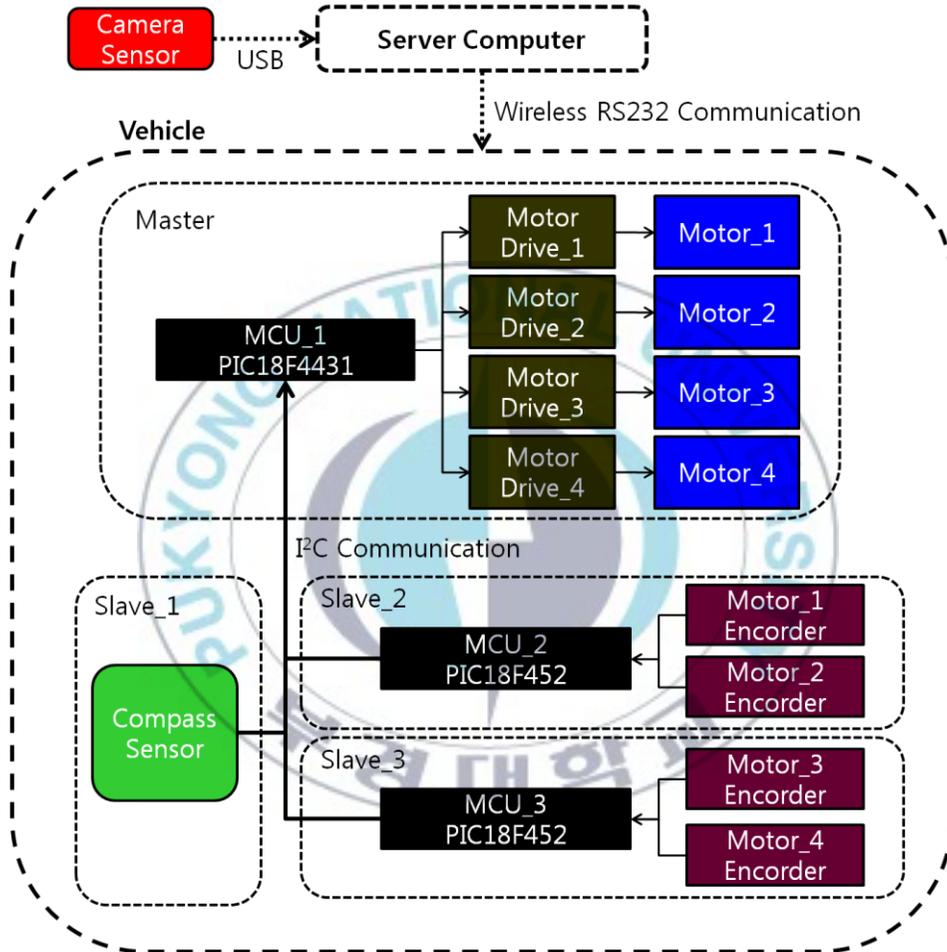


Fig. 3.6 Configuration of Omni-directional Mobile Vehicle Control System

전방향 주행형 차량의 제어시스템은 제어를 위한 Master용 1개의 PIC18F4431과 Slave용 2개의 PIC18F452로 구성된 마이크로 컨트롤러, 차량의 진행방향각을 구하기 위한 Compass 센서, 전방향 주행을 위하여

구현하기 위하여 각 휠마다 장착된 4개의 모터와 그 모터를 구동시키는 모터드라이브, 그리고 각 모터의 회전속도를 계산하기 위한 로터리 엔코더로 구성된다. 시스템은 크게 센서 데이터와 엔코더 데이터를 수집하고, 4개의 모터드라이브를 구동하기 위해 Master와 Slave_1, Slave_2, Slave_3으로 나누어 구성하였으며, Master와 Slave는 I²C 방식을 이용하여 데이터는 데이터를 주고받는다.

Slave_1은 경로추종 알고리즘에 사용될 차량의 진행방향각을 구하기 위해 Compass 센서로부터 데이터를 획득한다. Compass 센서는 0.1°의 분해능을 가지는 CMP03을 사용하였다.

Slave_2와 Slave_3에 사용된 마이크로 컨트롤러는 PIC18F452를 사용하였다. 각 휠에 장착된 로터리 엔코더로부터 휠의 회전수를 받아 샘플링 시간(10ms)동안의 회전속도를 계산한다. 로터리 엔코더는 모터 1회전당 360 펄스(Pulse)를 발생시키며, 모터축에 연결하여 모터의 회전수를 측정한다.

Master에 사용된 마이크로 컨트롤러는 PIC18F4431으로 Slave_1의 차량 진행방향각과 Slave_2, Slave_3에서 계산된 휠의 회전속도를 반영하여 Server Computer에서 계산한 주행경로를 추종하기 위한 제어를 수행한다. 또한 각 모터를 구동시키기 위해 모터드라이브로 PWM 신호를 발생시킨다.

모터드라이브는 24V/3A까지 제어 가능한 LMD18200을 사용하였다.

각 휠에 장착된 모터는 24V/15W 3000rpm의 DC모터로서 모터축과 휠은 별도의 기어와 풀리의 사용 없이 직접적으로 연결되어 있다.

또한 Server Computer는 Camera Sensor와 USB 포트를 통해 유선으로 연결되어 있으며, Camera Sensor로부터 실험구역의 영상을 수신하여 영상처리와 함께 경로생성 알고리즘 계산을 수행한다. 그리고 차량의 주행과 관련된 정보를 무선방식의 RS232통신을 사용하여 차량에 전송한다.

3.4 전방향 주행형 차량의 모델링

본 장에서는 전방향 주행형 차량의 운동학적(Kinematic) 모델링과 동역학적(Dynamic) 모델링을 제시한다. 차량은 비홀로노믹 구속조건(nonholonomic constrain)을 가지기 때문에 이 차량의 운동학적 구속방정식들을 적분할 수 없다는 특징이 있다. 이러한 제약들은 구속방정식의 좌표를 간소화 할 수 없기 때문에 운동학적, 동역학적 해석에 많은 어려움이 있다. 따라서 차량 모델링의 단순화를 위하여 다음과 같이 가정한다.

첫째, 차량중심은 차체의 기하학적 중심에 위치한다. 또한 차량의 질량중심은 $E_0=(X,Y,Z)$ 로 표현되고, 본 연구에서는 2차원상의 평면운동만을 고려하므로 Z 는 고정이다.

둘째, 4개의 메카닉 휠은 차량의 기하학적 중심으로부터 동일한 거리에 위치하고 있으며, 각 휠의 모터는 독립적으로 구동된다.

셋째, 차량의 몸체는 강체이다.

넷째, 외란은 존재하지 않는다.

위와 같은 비홀로노믹 시스템 모델링은 여러 문헌에서 소개되었다[27-45]. 3.4.1 절에서는 메카닉 휠을 사용한 Mobile Vehicle 에 적용 가능한 기구학적 모델링을 제시한다. 또한 본 연구에서 사용되는 메카닉 휠의 특성상 Free Roller와 지면 사이에서 발생하는 마찰력을 무시할 수 없다. 따라서 3.4.2 절에서 마찰력을 고려한 동역학적 모델링을 제시한다.

3.4.1 전방향 주행형 차량의 운동학적 모델링

본 절에서는 메카넘 휠을 적용한 전방향 주행형 차량의 운동학적 모델링을 제시한다. Fig. 3.7은 전방향 주행형 차량의 각 휠에 대응되는 좌표를 나타낸다. 메카넘 휠은 전방향 주행을 구현하기 위해 좌/우측, 전/후방 대칭을 이루도록 장착되었다. 차량의 진행방향은 4개 휠의 정/역회전의 방향과 그 회전 속도의 합력에 의해 결정된다. E_{iW} 와 E_0 는 각각 휠과 차량의 좌표계를 나타내며, Table 3.1은 Fig. 3.7에 표시된 매개변수에 대한 설명을 나타낸다.

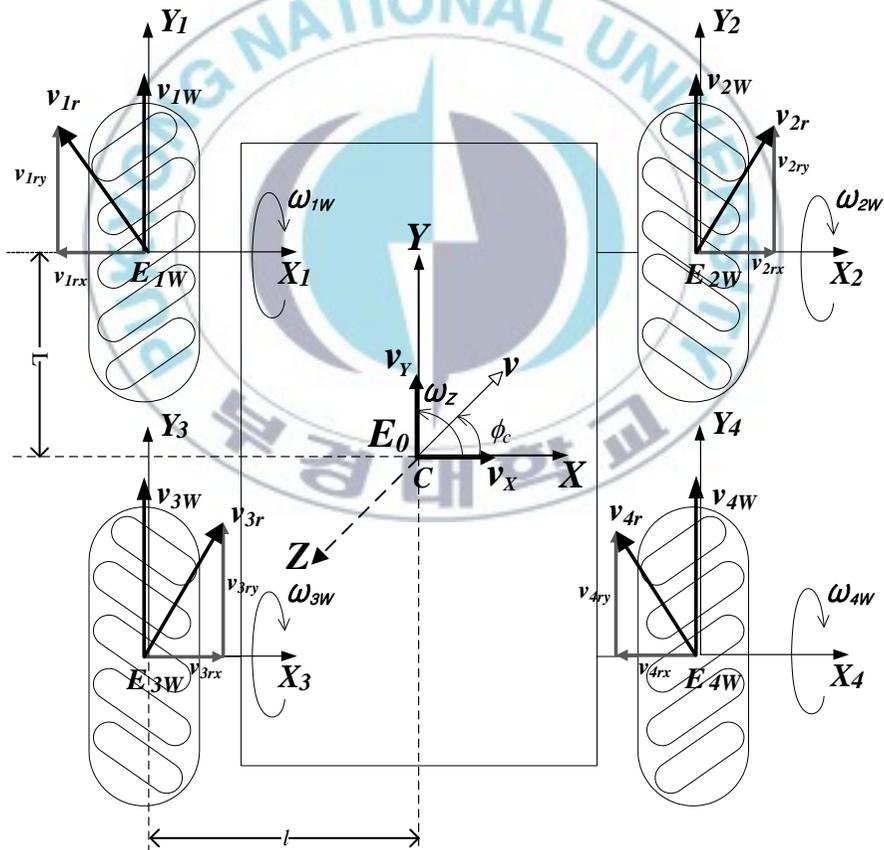


Fig. 3.7 Coordinate of Omni-directional Mobile Vehicle Platform

Table 3.1 Parameters of Omni-directional Mobile Robot

Parameter	Description
E_0	전방향 주행형 차량의 좌표계
E_{iW}	i 번째 휠의 좌표($i=1, 2, 3, 4$)
v	전방향 주행형 차량의 선속도
v_X	전방향 주행형 차량의 X 축 방향의 선속도
v_Y	전방향 주행형 차량의 Y 축 방향의 선속도
v_{iX}	i 번째 휠의 X 축 방향 선속도
v_{iY}	i 번째 휠의 Y 축 방향 선속도
v_{iw}	i 번째 휠의 선속도
v_{ir}	i 번째 휠의 롤러 회전축의 수직방향으로 발생하는 선속도
v_{irX}	i 번째 휠의 v_{ir} 의 X 축 방향 성분
v_{irY}	i 번째 휠의 v_{ir} 의 Y 축 방향 성분
l	좌·우 휠 사이 거리의 1/2
L	휠 중심에서 차량중심 사이의 거리
ω_Z	전방향 주행형 차량의 회전각속도
ω_{iW}	i 번째 휠의 회전각속도
ϕ_c	전방향 주행형 차량의 전진방향각

$i(i=1,2,3,4)$ 번째 메카닉 휠이 Y 축의 정방향으로 X 축의 중심축에 대해 ω_{iW} 의 회전 각속도로 회전할 경우, 휠의 Y 축 방향의 선속도 v_{iW} 는 다음과 같다.

$$v_{iW} = R_W \times \omega_{iW} \quad (3.1)$$

여기서, R_W 는 휠의 반경, ω_{iW} 는 휠의 회전 각속도, θ_{iW} 은 휠의 회전각이다. Fig. 3.8은 식 (3.1)의 파라미터들을 도식화한 것이다.

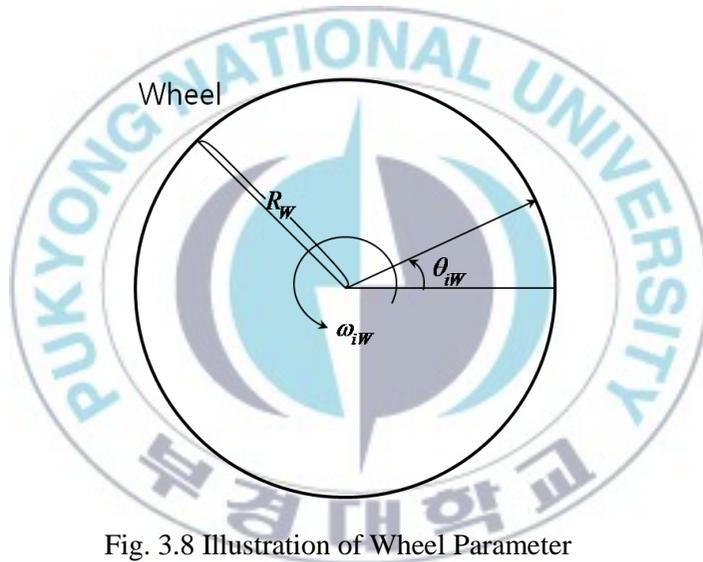


Fig. 3.8 Illustration of Wheel Parameter

휠이 Y 축 정방향으로 회전하면, 지면과 롤러의 접점에서 롤러 회전축의 수직방향으로 발생하는 선속도 $v_{ir}(i=1,2,3,4)$ 이 발생하게 된다. i 번째 휠의 X, Y 방향의 선속도 v_{iX}, v_{iY} 는 v_{ir} 의 X, Y 방향성분 v_{irX}, v_{irY} 에 전방향 주행형 차량의 선속도 v_{iW} 의 영향으로 다음과 같이 구할 수 있다.

$$v_{1X} = -\frac{v_{1r}}{\sqrt{2}}, \quad v_{1Y} = v_{1W} + \frac{v_{1r}}{\sqrt{2}} \quad (3.2)$$

$$v_{2X} = \frac{v_{2r}}{\sqrt{2}}, \quad v_{2Y} = v_{2W} + \frac{v_{2r}}{\sqrt{2}} \quad (3.3)$$

$$v_{3X} = \frac{v_{3r}}{\sqrt{2}}, \quad v_{3Y} = v_{3W} + \frac{v_{3r}}{\sqrt{2}} \quad (3.4)$$

$$v_{4X} = -\frac{v_{4r}}{\sqrt{2}}, \quad v_{4Y} = v_{4W} + \frac{v_{4r}}{\sqrt{2}} \quad (3.5)$$

Fig 3.9는 차량의 중심 E_0 에 대한 각 휠의 선속도 벡터도이다.

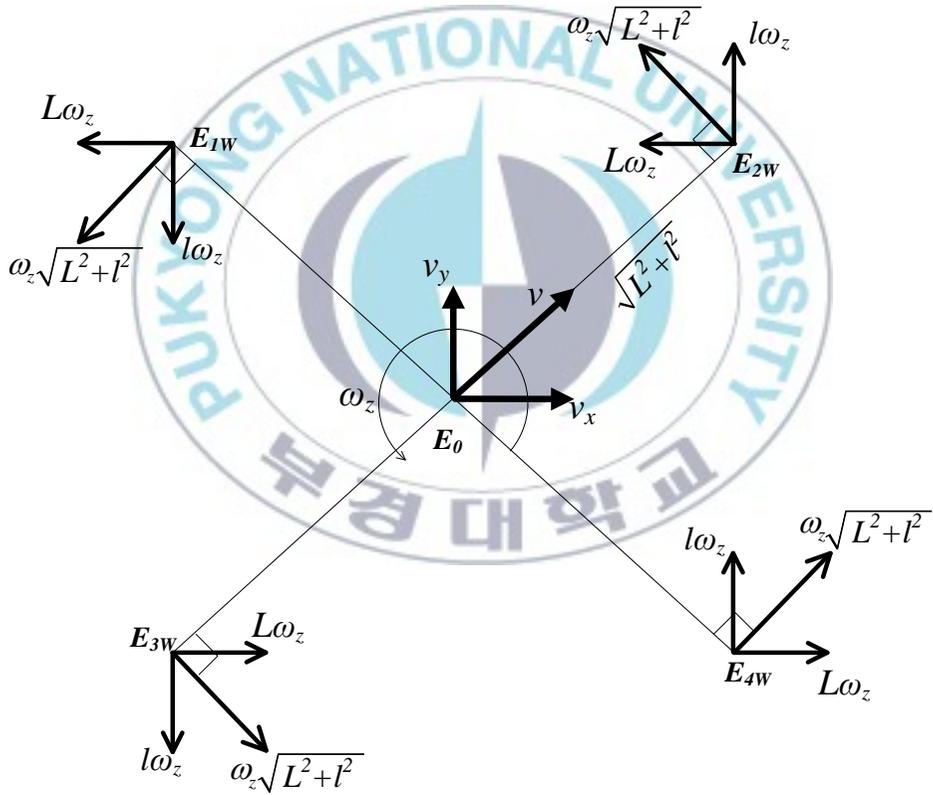


Fig. 3.9 Vector Diagram of Mecanum Wheel with respect to Vehicle Center E_0

또한, Fig. 3.9로부터 v_{iX} , v_{iY} 는 다음과 같이 표현할 수 있다.

$$v_{1X} = v_X - L \cdot \omega_z \quad , \quad v_{1Y} = v_Y - l \cdot \omega_z \quad (3.6)$$

$$v_{2X} = v_X - L \cdot \omega_z \quad , \quad v_{2Y} = v_Y + l \cdot \omega_z \quad (3.7)$$

$$v_{3X} = v_X + L \cdot \omega_z \quad , \quad v_{3Y} = v_Y - l \cdot \omega_z \quad (3.8)$$

$$v_{4X} = v_X + L \cdot \omega_z \quad , \quad v_{4Y} = v_Y + l \cdot \omega_z \quad (3.9)$$

여기서, v_X, v_Y 는 차량의 주행방향의 선속도 v 의 X, Y 방향의 성분이다.

식 (3.2)~(3.5)를 정리하여 식 (3.6)~(3.9)를 대입하면, 다음과 같은 식이 얻어진다.

$$v_{1W} = v_X + v_Y - (L+l)\omega_z \quad (3.10)$$

$$v_{2W} = -v_X + v_Y + (L+l)\omega_z \quad (3.11)$$

$$v_{3W} = -v_X + v_Y - (L+l)\omega_z \quad (3.12)$$

$$v_{4W} = v_X + v_Y + (L+l)\omega_z \quad (3.13)$$

식 (3.10)~(3.13)을 정리하면, 역기구학 방정식 (3.14)을 얻을 수 있다.

$$\mathbf{v}_W = \mathbf{J} \mathbf{v}_0 = \mathbf{R}_W \boldsymbol{\omega} \quad (3.14)$$

$$\mathbf{v}_w = [v_{1w} \ v_{2w} \ v_{3w} \ v_{4w}]^T \in \mathbf{R}^{4 \times 1}$$

$$\mathbf{J} = \begin{bmatrix} 1 & 1 & -(L+l) \\ -1 & 1 & (L+l) \\ -1 & 1 & -(L+l) \\ 1 & 1 & (L+l) \end{bmatrix} \in \mathbf{R}^{4 \times 3}$$

$$\mathbf{v}_0 = [v_x \ v_y \ \omega_z]^T \in \mathbf{R}^{3 \times 1}$$

$$\boldsymbol{\omega} = [\omega_{1w} \ \omega_{2w} \ \omega_{3w} \ \omega_{4w}]^T \in \mathbf{R}^{4 \times 1}$$

여기서, \mathbf{v}_0 는 직각 좌표계의 차량의 선속도 벡터를 나타내고, \mathbf{v}_w 는 휠의 선속도 벡터를 나타내며, \mathbf{J} 는 변환행렬이며, $\boldsymbol{\omega}$ 는 휠의 각속도 벡터이다.

한편, 차량의 속도는 식 (3.14)로부터 Pseudo Inverse Matrix를 이용하여 다음과 같이 구할 수 있다.

$$\mathbf{v}_0 = \mathbf{J}^+ \mathbf{v}_w = R_w \mathbf{J}^+ \boldsymbol{\omega} \quad (3.15)$$

$$\mathbf{J}^+ = \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{(L+l)} & \frac{1}{(L+l)} & \frac{1}{(L+l)} & -\frac{1}{(L+l)} \end{bmatrix} \quad (3.16)$$

여기서 $\mathbf{J}^+ = (\mathbf{J}^T \cdot \mathbf{J})^{-1} \mathbf{J}^T \in \mathbf{R}^{3 \times 4}$ 이며 \mathbf{J} 의 Pseudo Inverse Matrix를 나타낸다.

또한 식 (3.15)와 (3.16)으로부터 다음 식이 구해진다.

$$v_X = \frac{R_w}{4}(\omega_{1W} - \omega_{2W} - \omega_{3W} + \omega_{4W}) \quad (3.17)$$

$$v_Y = \frac{R_w}{4}(\omega_{1W} + \omega_{2W} + \omega_{3W} + \omega_{4W}) \quad (3.18)$$

$$\omega_z = \frac{R_w}{4(L+l)}(-\omega_{1W} + \omega_{2W} + \omega_{3W} - \omega_{4W}) \quad (3.19)$$

한편 차량의 중심 C 의 위치벡터를 다음과 같이 정의한다.

$$\mathbf{q}_c = [X_C \ Y_C \ \phi_C]^T \quad (3.20)$$

식 (3.15)와 식 (3.20)으로부터 차량의 중심 C 에서의 속도벡터 \mathbf{v}_0 에 대한 차량의 선속도와 회전 각속도와의 관계벡터 $\boldsymbol{\eta}$ 및 바퀴의 회전 각속도 벡터 $\boldsymbol{\omega}$ 와의 관계식은 다음과 같다.

$$\mathbf{v}_0 = \dot{\mathbf{q}}_c = \mathbf{S}\boldsymbol{\eta} = R_w \mathbf{J}^+ \boldsymbol{\omega} \quad (3.22)$$

$$\mathbf{S} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \quad (3.22)$$

$$\boldsymbol{\eta} = \begin{bmatrix} v \\ \omega_z \end{bmatrix} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{v}_0 = \mathbf{S}^T \mathbf{v}_0 = \mathbf{S}^T R_w \mathbf{J}^+ \boldsymbol{\omega} \quad (3.23)$$

3.4.2 전방향 주행형 차량의 동역학적 모델링

본 절에서는 운동학적 모델링에서 구한 파라미터들을 사용하여 Lagrange 운동방정식을 이용한 전방향 주행형 차량의 동역학적 모델링을 제시한다. 모델링 전 가정에서 차량은 2차원 평면상에서 운동한다고 하였으므로 역학적 에너지의 총합에서 위치에너지는 무시할 수 있다. 따라서 위치에너지를 제외한 외력 τ_i 가 차량에 작용하고 있고, 휠 구동 시 발생하는 회전 마찰에너지를 고려한 시스템의 Lagrange 운동방정식은 다음과 같다.

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\theta}_i} \right) - \frac{\partial K}{\partial \theta_i} + \frac{\partial D}{\partial \dot{\theta}_i} = \tau_i \quad (3.24)$$

위에너지를 무시한 차량의 역학적 에너지의 총합은 운동에너지의 총합 K 와 같고, 식 (3.25)와 같이 나타낼 수 있다.

$$K = A + B + C \quad (3.25)$$

$$A = \frac{1}{2} m (v_x^2 + v_y^2) \quad (3.26)$$

$$B = \frac{1}{2} J_z \omega_z^2 \quad (3.27)$$

$$C = \frac{1}{2} J_w (\omega_{w1}^2 + \omega_{w2}^2 + \omega_{w3}^2 + \omega_{w4}^2) \quad (3.28)$$

여기서 식 (3.25)는 차량의 직선 운동에너지의 총합이며, m 은 차량전체 질량을 의미한다. 식 (3.26)은 차량의 회전 운동에너지의 총합을

나타내며, J_z 는 차량 중심에서 Z 축에 대한 관성모멘트를 나타낸다. 식 (3.27)는 휠의 회전 운동에너지의 총합을 나타낸다. 여기서 J_w 는 메카넘 휠의 회전중심에 대한 관성모멘트이고, ω_{iw} ($i=1,2,3,4$)는 i 번째 휠의 회전 각속도를 의미한다. 또한, 마찰에 의한 손실에너지 D 는 식 (3.29)과 같이 표현할 수 있다.

$$D = \frac{1}{2} D_\theta (\omega_{1w}^2 + \omega_{2w}^2 + \omega_{3w}^2 + \omega_{4w}^2) \quad (3.29)$$

여기서, D_θ 는 메카넘 휠의 점성마찰계수다.

먼저, 운동학적 모델링에서 구한 v_x , v_y , ω_z 를 식 (3.26)~(3.28)에 대입하여 정리하면 다음과 같다.

$$\begin{aligned} A &= \frac{1}{2} m (v_x^2 + v_y^2) \\ &= \frac{R_w^2}{16} m (\omega_{1w}^2 + \omega_{2w}^2 + \omega_{3w}^2 + \omega_{4w}^2 + 2\omega_{1w}\omega_{4w} + 2\omega_{2w}\omega_{3w}) \end{aligned} \quad (3.30)$$

$$\begin{aligned} B &= \frac{1}{2} J_z \cdot \omega_z^2 \\ &= \frac{R_w^2 \cdot J_z}{32(L+l)^2} (\omega_{1w}^2 + \omega_{2w}^2 + \omega_{3w}^2 + \omega_{4w}^2 - 2\omega_{1w}\omega_{2w} + 2\omega_{1w}\omega_{3w} \\ &\quad - 2\omega_{1w}\omega_{4w} - 2\omega_{2w}\omega_{3w} + 2\omega_{2w}\omega_{4w} - 2\omega_{3w}\omega_{4w}) \end{aligned} \quad (3.31)$$

$$C = \frac{1}{2} J_w (\omega_{1w}^2 + \omega_{2w}^2 + \omega_{3w}^2 + \omega_{4w}^2) \quad (3.32)$$

다음으로, 식 (3.29)~(3.32)를 식 (3.24)에 적용하기 위해 각 휠의 ω_{iW} ($i=1,2,3,4$)에 대하여 정리하여 유도하면 식 (3.33)~(3.37)과 같다.

$$\frac{d}{dt} \left(\frac{\partial A}{\partial \omega_{iW}} \right) = \frac{R_w^2}{8} m \begin{pmatrix} \dot{\omega}_{1W} + \dot{\omega}_{4W} \\ \dot{\omega}_{2W} + \dot{\omega}_{3W} \\ \dot{\omega}_{2W} + \dot{\omega}_{3W} \\ \dot{\omega}_{1W} + \dot{\omega}_{4W} \end{pmatrix} \quad (3.33)$$

$$\frac{d}{dt} \left(\frac{\partial B}{\partial \omega_{iW}} \right) = \frac{R_w^2 \cdot J_z}{16(L+l)^2} m \begin{pmatrix} \dot{\omega}_{1W} - \dot{\omega}_{2W} + \dot{\omega}_{3W} - \dot{\omega}_{4W} \\ -\dot{\omega}_{1W} + \dot{\omega}_{2W} - \dot{\omega}_{3W} + \dot{\omega}_{4W} \\ \dot{\omega}_{1W} - \dot{\omega}_{2W} + \dot{\omega}_{3W} - \dot{\omega}_{4W} \\ -\dot{\omega}_{1W} + \dot{\omega}_{2W} - \dot{\omega}_{3W} + \dot{\omega}_{4W} \end{pmatrix} \quad (3.34)$$

$$\frac{d}{dt} \left(\frac{\partial C}{\partial \omega_{iW}} \right) = J_w \begin{pmatrix} \omega_{1W} \\ \omega_{2W} \\ \omega_{3W} \\ \omega_{4W} \end{pmatrix} \quad (3.35)$$

$$\frac{\partial D}{\partial \omega_{iW}} = D_\theta \begin{pmatrix} \omega_{1W} \\ \omega_{2W} \\ \omega_{3W} \\ \omega_{4W} \end{pmatrix} \quad (3.36)$$

$$\frac{\partial K}{\partial \theta_{iW}} = \frac{\partial A}{\partial \theta_{iW}} + \frac{\partial B}{\partial \theta_{iW}} + \frac{\partial C}{\partial \theta_{iW}} = 0 \quad (3.37)$$

식 (3.33)~(3.37)을 식(3.24)에 대입하여 정리하면, 식 (3.38)와 같이 토크와 회전각속도에 의한 전방향 주행형 차량의 동역학 모델링을 얻을 수 있다.

$$\mathbf{M}\dot{\boldsymbol{\omega}} + D_{\theta}\boldsymbol{\omega} = \boldsymbol{\tau} \quad (3.38)$$

여기서, $\boldsymbol{\tau}$ 는 휠의 토크벡터, $\boldsymbol{\omega}$ 는 휠의 회전각속도벡터, \mathbf{M} 은 관성행렬이고, 각각 다음과 같이 표현된다.

$$\boldsymbol{\tau} = [\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4]^T$$

$$\mathbf{M} = \begin{bmatrix} \alpha + \beta + J_w & -\beta & \beta & \alpha - \beta \\ -\beta & \alpha + \beta + J_w & \alpha - \beta & \beta \\ \beta & \alpha - \beta & \alpha + \beta + J_w & -\beta \\ \alpha - \beta & \beta & -\beta & \alpha + \beta + J_w \end{bmatrix},$$

$$\alpha = \frac{m}{8} R_w^2, \quad \beta = \frac{J_z}{16(L+l)^2} R_w^2$$

앞 절의 운동학 모델링에서 식(3.14)와 식(3.22)로부터 휠의 회전각속도 벡터 $\boldsymbol{\omega}$ 는 다음과 같이 얻어진다.

$$\boldsymbol{\omega} = \frac{\mathbf{J}}{R_w} \mathbf{v}_0 = \frac{\mathbf{J}}{R_w} \mathbf{S}\boldsymbol{\eta} = [\omega_{1w}, \omega_{2w}, \omega_{3w}, \omega_{4w}]^T \quad (3.39)$$

제 4 장 장애물 회피를 위한 경로계획 알고리즘 및 경로추종 제어기 설계

4.1 장애물 회피를 위한 경로계획 알고리즘

전방향 주행형 차량이 목표지점을 도달함에 있어서 주행 경로상에 있는 장애물과 충돌없이 주행하기 위해서는 Server Computer로부터 획득한 장애물의 위치정보를 이용하여 이동로봇이 장애물을 회피할 수 있는 목표경로를 발생시키는 경로계획 알고리즘이 필요하다. 이를 위해 먼저, 전방향 주행형 차량의 시작점과 종점 사이의 최단경로를 획득하고, 생성된 경로상에 있는 장애물의 좌표를 획득한 후 경로계획 알고리즘을 적용하여 장애물을 회피할 수 있도록 한다.

본 연구에서는 경로계획 알고리즘 중에서 널리 쓰이며 그 효율성이 검증된 A*알고리즘을 제안한다. A*알고리즘의 장점은 빠른 시간안에 결과를 도출해 내어 실시간으로 운용되는 시스템에 그 결과를 반영할 수 있다는 것이다. 이것은 최단거리 알고리즘을 기본적으로 적용하기 때문에 다른 알고리즘에 비해서 결과 도출속도가 우수하다. A*알고리즘의 원리는 크게 세 단계로 나눌 수 있으며, 다음과 같다.

첫째, A*알고리즘을 적용하기 위해서는 먼저 차량과 장애물, 목표지점이 모두 포함된 영상에 일정한 크기의 격자로 구성된 격자맵을 생성한다. 격자맵에서 격자 하나는 차량의 주행경로를 생성하기 위한 하나의 노드로 정의한다. 따라서 격자의 크기가 작을수록 정밀한 경로생성이 가능 가능하지만 많은 노드들을 분석해야 하는 단점이 있다.

반대로 격자의 크기가 커지면 분석시간은 짧아지지만 생성된 경로가 정밀하지 못하다는 단점이 있다. 따라서 실험을 통해 환경과 목적에 맞도록 적절한 크기의 격자를 선정해야 한다. 본 연구에서 사용한 영상의 크기는 3:2 비율을 가지는 640 x 480 pixel이므로, 영상안에 정사각형의 격자를 가득 채우기 위해 32 x 32 pixel의 격자를 선정하여 적용하였다. 32 x 32 pixel의 격자를 선정할 경우 20 x 15개의 격자가 생성된다. Fig. 4.1은 격자맵 상에서 차량이 진행할 수 있는 인접 노드들을 표시한 것이다.

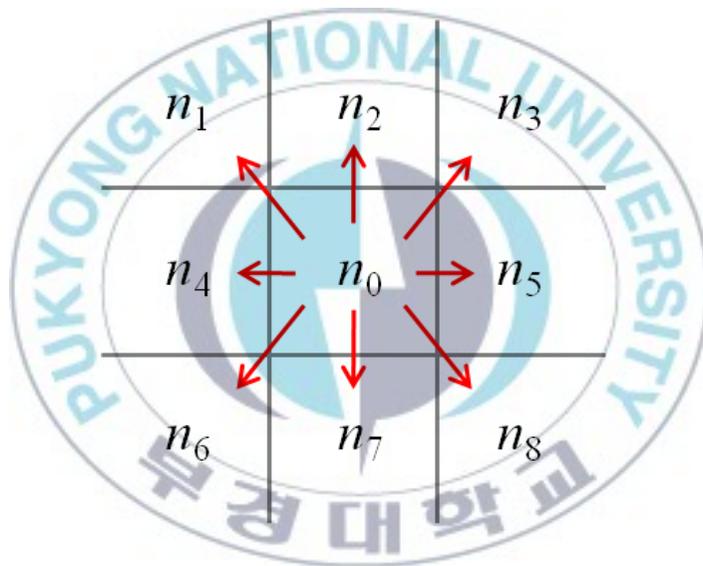


Fig. 4.1 Neighbor Nodes Movable from Center Node to in Eight Directions

Fig. 4.1에서 현재 시간 t 일 때 차량이 n_0 노드에 있을 경우, 다음 시간 $t+1$ 에서 차량은 $n_1 \sim n_8$ 까지 8방향 중 장애물이 없는 한 노드로 진행할 수 있다. 또한 차량의 크기를 고려하여 장애물과 차량 사이의 거리가 차량크기보다 클 경우 진행가능 노드로 판단한다. 만약 특정 노드에

장애물이 있거나 장애물과 차량 사이의 거리가 차량크기보다 작을 경우, 해당 노드는 진행가능 노드에서 제외된다.

둘째, 위와 같은 원리를 적용하여 출발 노드로부터 목표 노드까지의 최적 주행경로를 탐색하기 위해서 각 격자 노드에 대한 비용평가 함수를 식 (4.1)과 같이 정의한다.

$$f(n) = g(n) + h(n) \quad (4.1)$$

여기서 n 은 노드를 의미하며, $f(n)$ 은 비용평가 함수이며, $g(n)$ 은 차량의 출발 노드로부터 현재 노드까지 이동한 총 경로비용을 나타내는 함수이며, $h(n)$ 은 현재 노드부터 목표 노드까지 이동할 거리비용을 나타내는 함수이다. 이 때, 거리비용은 다차원 공간에서 두 점 사이의 거리를 측정하기 위해 일반적으로 사용되는 유클리디안 거리를 이용한다. 노드와 그 인접한 노드들의 중심까지의 거리 비용을 1로 하고, 그 대각 노드거리를 $\sqrt{2}$ 로 한다.

셋째, 경로가 다수 존재하는 경우가 발생하면, 그 경로들 중 비용평가 함수가 최소인 경로를 선택한다. 만약 비용평가 함수가 최소인 경로가 다수 존재하는 경우, 다음과 같은 절차를 따라 하나의 경로만 선택하게 된다.

- 1) 경로상에 방향전환의 횟수가 적은 경로를 우선적으로 선택한다.
- 2) 방향전환의 횟수가 같을 경우, 경로가 최초로 분기되는 시점을 기준으로 방향전환이 먼저 일어나는 경로를 선택한다.

Fig. 4.3은 같은 비용의 경로가 다수 존재할 경우, 1)과 2)의 절차에 의해 하나의 경로를 선택하기 위한 과정을 예로 들어 나타낸다.

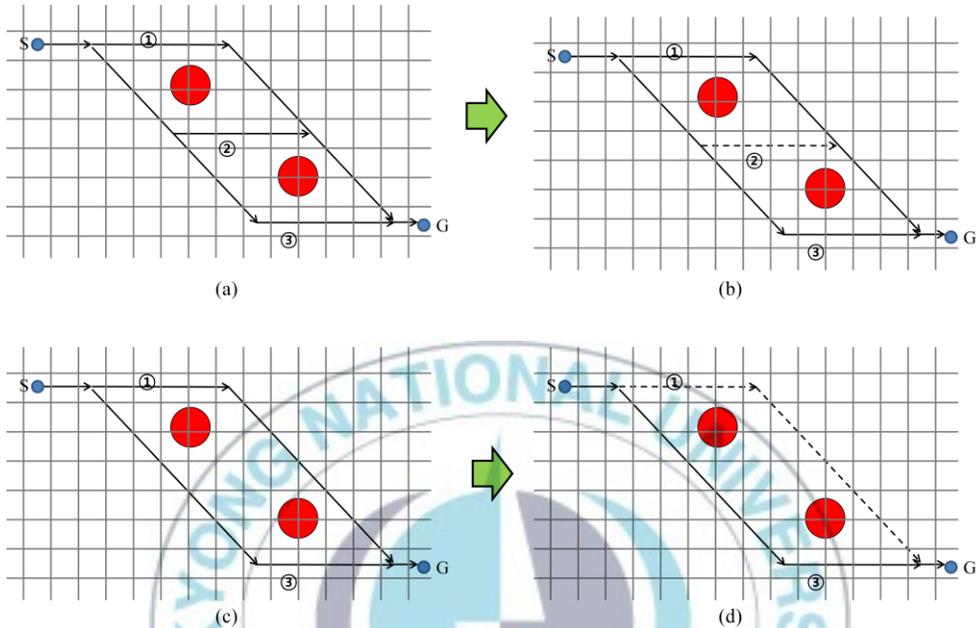


Fig. 4.3 Path Planning Method in Case of the Same Cost Evaluation Function

Fig. 4.3의 (a)는 동일한 비용함수를 갖는 ①~③의 4개의 경로가 존재하는 경우로써, 절차 1)에 따라 방향전환이 많은 경로 ②가 먼저 제거된다(b). 또한, (c)에서 절차 1)을 거친 후 남은 경로 ①, ③에 대해 절차 2)에 따라 방향전환이 먼저 발생하는 경로 ③이 선택되고, 경로 ①이 제거된다(d). 최종적으로 경로 ③이 남게되고, 경로 ③를 차량의 주행경로로 사용하게 된다.

Fig. 4.4는 A*알고리즘을 장애물이 있는 실험환경에 대하여 적용한 예를 나타낸다.

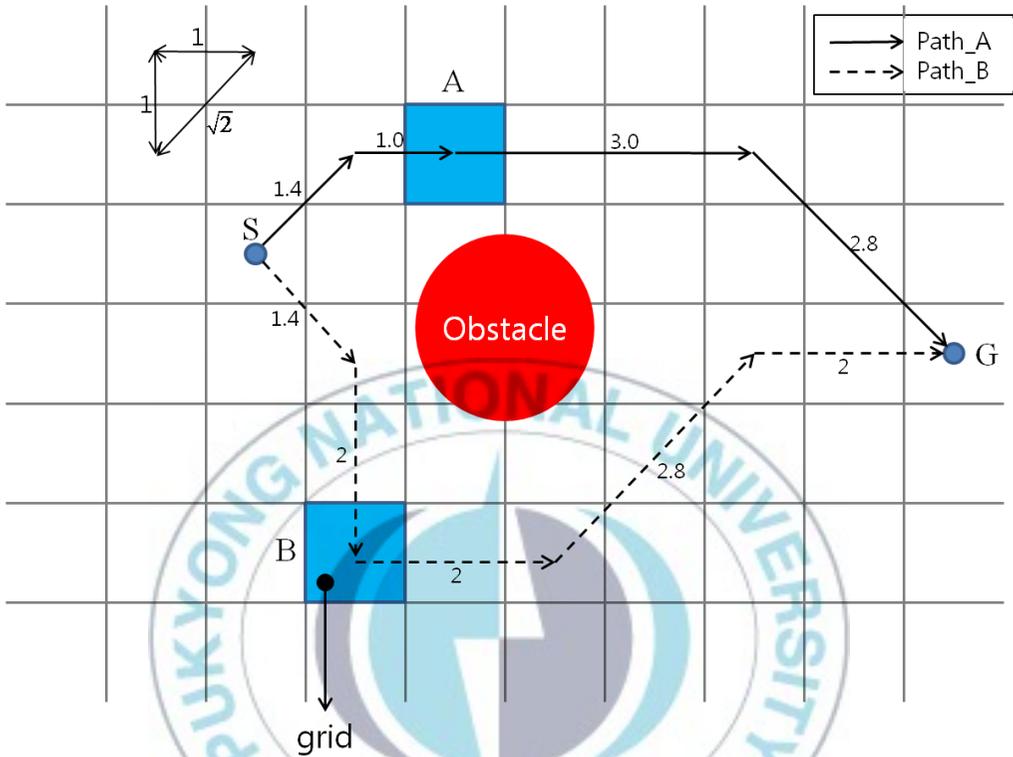


Fig. 4.4 Calculation of Cost Evaluation Function through Two Different Nodes

Fig. 4.4에서 S는 출발노드를, G는 목표노드를 나타낸다. 차량은 S노드에서 출발해 인근 8방향 노드들의 $f(n)$ 값을 비교하여 가장 저렴한 비용을 가진 하나의 인접노드로 이동한다. 이 과정을 G노드까지 반복하게 되면 가장 저렴한 비용의 경로가 생성되고, 차량은 이를 따라 주행함으로써 목표지점까지 최단거리로 도착할 수 있다. 한 노드의 중심으로부터 그 노드와 인접하고 있는 노드의 중심까지의 거리를 1로

가정하면, 그 대각거리는 $\sqrt{2}$ (약 1.4)가 된다. 이 때, 장애물의 위치가 고려된 두 개의 경로 Path_A와 Path_B 위에 있는 노드 A와 B에서의 총 비용함수를 구해 비교해본다. 먼저, 차량이 Path_A를 선택하여 주행할 경우, S로 부터 A까지 이동한 거리비용 $g(n)$ 은 $2.4(=1.0+1.4)$ 이다. 또한 A에서 G까지 이동할 거리비용 $h(n)$ 은 $5.8(=3.0+2.8)$ 이다. 따라서 차량이 Path_A를 선택하여 주행할 때, 노드 A에서의 $f(n)$ 값은 $8.2(=2.4+5.8)$ 로 계산된다. 다음으로, 차량이 Path_B를 선택하여 주행할 경우, B까지 이동한 비용함수 $g(n)$ 은 노드 A와 동일한 방법으로 계산하여 3.4가 되고, $h(n)$ 은 6.8이 된다. 따라서 차량이 Path_B를 선택하여 주행할 때, 노드B에서의 $f(n)$ 의 값은 10.2가 된다. 이것은 노드 B보다 노드 A를 선택할 때 비용이 더 저렴하다는 것을 의미한다. 즉, 노드 A를 통과하는 Path_A가 더 효율적인 경로이므로 최종적으로 Path_A를 차량의 주행 경로로 선정한다.

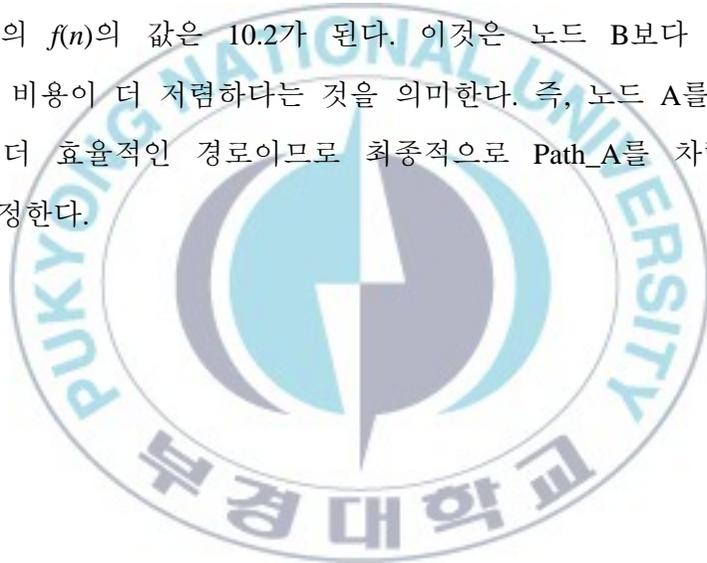


Fig. 4.7은 A*알고리즘의 적용을 3단계로 요약하여 나타낸 것이며, 그 내용은 아래와 같다.

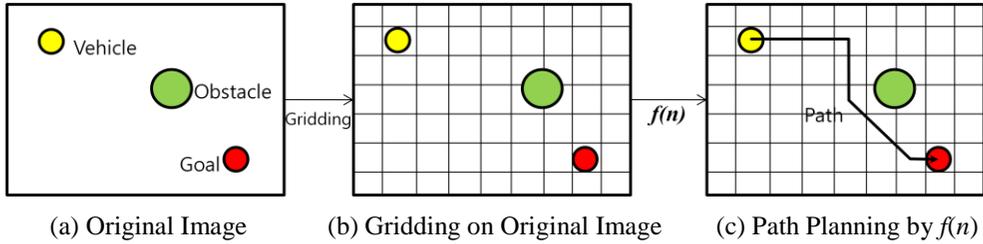


Fig. 4.7 Concept of an A* Algorithm

- (a) : 원본 영상에 대하여 영상처리 과정을 거쳐 차량과 장애물 및 목표 지점을 식별한다.
- (b) : 물체가 식별된 영상에 일정한 크기의 격자로 구성된 격자맵을 생성한다.
- (c) : 격자맵상의 각 노드들의 $f(n)$ 값을 분석해 최단경로를 결정한다.

4.2 전방향 주행형 차량의 제어기 설계

본 절에서는 4.1절에서 제시된 경로계획 알고리즘에 의해 생성된 경로를 추적하기 위한 경로추적 제어기의 설계를 소개한다. 4.2.1절에서 운동학적 모델을 이용한 경로추적 제어기(Path Tracking Controller)를 설계하고, 4.2.2절에서 동역학적 모델을 이용한 속도추적 제어기(Velocity Tracking Controller)를 설계한다.

4.2.1 경로추적 제어기 설계

경로추적 제어기의 목표는 Fig. 4.1에서 차량의 중심 C 가 A*알고리즘에 의해 설계된 경로 위의 점 $R(X_R, Y_R, \Phi_R)$ 을 추종할 수 있도록 하는 것이다.

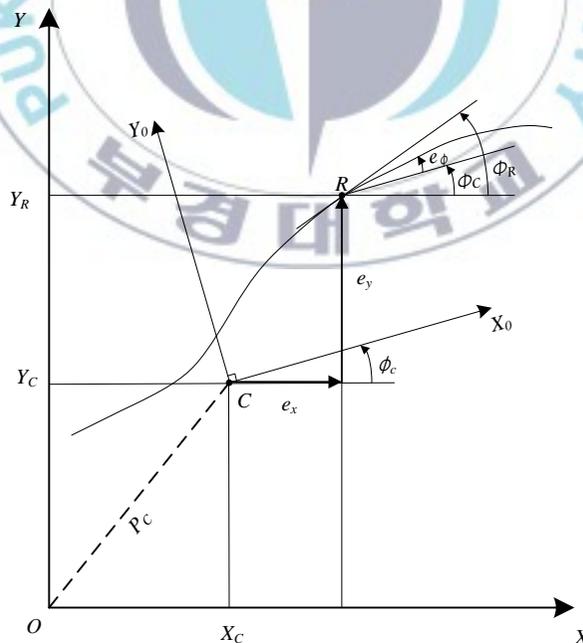


Fig. 4.1 Define Tracking Error Vector e_p

목표경로 위의 점 R 과 차량의 현재 위치 C 의 운동학적 방정식은 각각 다음과 같이 표현할 수 있다.

$$\dot{\mathbf{q}}_R = \begin{bmatrix} \cos \phi_R & 0 \\ \sin \phi_R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_R \\ \omega_R \end{bmatrix} = \mathbf{S}_R \boldsymbol{\eta}_R \quad (4.1)$$

$$\dot{\mathbf{q}}_C = \mathbf{v}_0 = \begin{bmatrix} \cos \phi_C & 0 \\ \sin \phi_C & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega_Z \end{bmatrix} = \mathbf{S}\boldsymbol{\eta} = R_w \mathbf{J}^+ \boldsymbol{\omega} \quad (4.2)$$

여기서 v_R, ω_R, ϕ_R 은 각각 목표경로 위의 목표점에서의 목표 선속도, 목표 각속도, 목표 전진방향이며, $\omega_Z = \dot{\phi}_C$ 이다.

차량의 중심 C 와 목표경로 위의 점 R 사이의 추적오차 벡터는 다음과 같이 정의한다.

$$\mathbf{e}_p = \begin{bmatrix} e_x \\ e_y \\ e_\phi \end{bmatrix} = \begin{bmatrix} X_R \\ Y_R \\ \phi_R \end{bmatrix} - \begin{bmatrix} X_C \\ Y_C \\ \phi_C \end{bmatrix} = \mathbf{q}_R - \mathbf{q}_C \in \mathbb{R}^{3 \times 1} \quad (4.3)$$

여기서 e_x, e_y 는 X, Y 방향의 추적오차를 나타내며, e_ϕ 는 회전각에 대한 오차를 나타낸다. 식 (3.22)와 식 (4.3)로부터 추적오차 벡터 \mathbf{e}_p 의 도함수는 다음과 같이 표현할 수 있다.

$$\dot{\mathbf{e}}_p = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\phi \end{bmatrix} = \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{\phi}_R \end{bmatrix} - \begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{\phi}_C \end{bmatrix} = \dot{\mathbf{q}}_R - \dot{\mathbf{q}}_C = \dot{\mathbf{q}}_R - R_w \mathbf{J}^+ \boldsymbol{\omega} \quad (4.4)$$

운동학적 모델을 이용한 제어칙은 다음과 같으며 차량 바퀴의 목표 각속도 벡터이다.

$$\boldsymbol{\omega}_d = \frac{1}{R_w} \mathbf{J}(\dot{\mathbf{q}}_R + \mathbf{K}\mathbf{e}_p) = \frac{1}{R_w} \mathbf{J}\mathbf{S}\boldsymbol{\eta}_d \quad (4.5)$$

여기서 $\mathbf{K} = \text{diag}(k_1, k_2, k_3) \in \mathbf{R}^{3 \times 3}$ 이며 양의 한정 행렬이며 $\boldsymbol{\eta}_d$ 는 차량의 목표 선속도와 목표 각속도로 구성된 속도벡터이다.

증명 : 리아푸노프 함수를 다음과 같이 정의한다.

$$V_0 = \frac{1}{2} \mathbf{e}_p^T \mathbf{e}_p \geq 0 \quad (4.6)$$

식 (4.4)으로부터 식 (4.6)의 도함수는 다음과 같이 구해진다.

$$\dot{V}_0 = \dot{\mathbf{e}}_p^T \mathbf{e}_p = (\dot{\mathbf{q}}_R - R_w \mathbf{J}^+ \boldsymbol{\omega})^T \mathbf{e}_p \quad (4.7)$$

식 (4.7)로부터 $\dot{V}_0 \leq 0$ 를 얻기 위해 다음 식이 성립해야 한다.

$$\dot{\mathbf{e}}_p = \dot{\mathbf{q}}_R - R_w \mathbf{J}^+ \boldsymbol{\omega} = -\mathbf{K}\mathbf{e}_p \quad (4.8)$$

식 (4.7)에 식 (4.8)을 대입하면 다음과 같다.

$$\begin{aligned} \dot{V}_0 &= \dot{\mathbf{e}}_p^T \mathbf{e}_p = (\dot{\mathbf{q}}_R - R_w \mathbf{J}^+ \boldsymbol{\omega})^T \mathbf{e}_p \\ &= (-\mathbf{K}\mathbf{e}_p)^T \mathbf{e}_p = -\mathbf{e}_p^T \mathbf{K}\mathbf{e}_p \leq 0 \end{aligned} \quad (4.9)$$

식 (4.9)로부터 $t \rightarrow \infty$ 일 때, $\mathbf{e}_p \rightarrow 0$ 임을 알 수 있다.

또한, 식 (4.8)과 식 (3.39)에서 $\boldsymbol{\omega} = \boldsymbol{\omega}_d$ 이고, $\boldsymbol{\omega}_d$ 에 대하여 다음과 같이 정리하면 식 (4.5)의 제어칙을 얻을 수 있다.

$$\begin{aligned}
 \dot{\mathbf{q}}_R - R_w \mathbf{J}^+ \boldsymbol{\omega}_d &= -\mathbf{K} \mathbf{e}_p \\
 \mathbf{J}^+ \boldsymbol{\omega}_d &= \frac{1}{R_w} (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) \\
 (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \boldsymbol{\omega}_d &= \frac{1}{R_w} (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) \\
 \mathbf{J}^T \boldsymbol{\omega}_d &= \frac{1}{R_w} (\mathbf{J}^T \mathbf{J}) (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) \\
 \mathbf{J} \mathbf{J}^T \boldsymbol{\omega}_d &= \frac{1}{R_w} \mathbf{J} (\mathbf{J}^T \mathbf{J}) (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) \\
 \boldsymbol{\omega}_d &= \frac{1}{R_w} (\mathbf{J} \mathbf{J}^T)^{-1} (\mathbf{J} \mathbf{J}^T) \mathbf{J} (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) \\
 \boldsymbol{\omega}_d &= \frac{1}{R_w} \mathbf{J} (\dot{\mathbf{q}}_R + \mathbf{K} \mathbf{e}_p) = \frac{1}{R_w} \mathbf{J} \boldsymbol{\eta}_d
 \end{aligned} \tag{4.10}$$

식 (4.5)를 목표 차량속도벡터 $\boldsymbol{\eta}_d$ 로 변환하면 다음 식이 얻어지며, 식 (3.23)에서 $\boldsymbol{\eta} \rightarrow \boldsymbol{\eta}_d$ 와 $\boldsymbol{\omega} \rightarrow \boldsymbol{\omega}_d$ 일 경우와 같음을 알 수 있다.

$$\boldsymbol{\eta}_d = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T R_w \mathbf{J}^+ \boldsymbol{\omega}_d = \mathbf{S}^T R_w \mathbf{J}^+ \boldsymbol{\omega}_d \tag{4.11}$$

4.2.2 속도추적 제어기 설계

차량 바퀴의 속도오차 벡터 $\mathbf{e}_v = [e_{v1} \ e_{v2} \ e_{v3} \ e_{v4}]^T \in R^{4 \times 1}$ 는 식 (3.35)를 이용해 다음과 같이 정의할 수 있다.

$$\begin{aligned} \mathbf{e}_v &= [e_{v1} \ e_{v2} \ e_{v3} \ e_{v4}]^T \\ &= \boldsymbol{\omega} - \boldsymbol{\omega}_d = \frac{1}{R_w} \mathbf{J} \mathbf{v}_0 - \boldsymbol{\omega}_d \end{aligned} \quad (4.14)$$

여기서 $\boldsymbol{\omega}_d$ 는 차량바퀴의 목표각속도 벡터를 나타낸다.

속도추적 제어기의 설계 목적은 식 (4.14)에 의해 $t \rightarrow \infty$ 일 때, 차량의 실제 속도 $\boldsymbol{\eta}$ 가 목표속도 $\boldsymbol{\eta}_d$ 에 수렴할 수 있도록 하는 것이다. 이것은 식 (3.39)와 식 (4.7)에 의해 차량바퀴의 실제 각속도 $\boldsymbol{\omega}$ 를 목표 각속도 $\boldsymbol{\omega}_d$ 로 수렴시키는 것과 같으며, 추적오차 $\mathbf{e}_v \rightarrow 0$ 이 되도록 설계하는 것과 같다. 속도오차 벡터 \mathbf{e}_v 의 도함수 $\dot{\mathbf{e}}_v$ 는 다음과 같다.

$$\dot{\mathbf{e}}_v = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_d \quad (4.15)$$

식 (3.34)로부터 $\dot{\boldsymbol{\omega}} = \mathbf{M}^{-1}(\boldsymbol{\tau} - D_\theta \boldsymbol{\omega})$ 이고, 이를 식 (4.15)에 대입하면 아래와 같다.

$$\dot{\mathbf{e}}_v = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_d = \mathbf{M}^{-1}(\boldsymbol{\tau} - D_\theta \boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d \quad (4.16)$$

동역학적 모델을 이용한 제어칙은 다음과 같다.

$$\boldsymbol{\tau} = (D_\theta - \mathbf{M}c)\mathbf{e}_v + \mathbf{M}\dot{\boldsymbol{\omega}}_d + D_\theta \boldsymbol{\omega}_d \quad (4.17)$$

증명 : 다음과 같이 리아푸노프 함수를 정의한다.

$$V_1 = \frac{1}{2} \mathbf{e}_v^T \mathbf{e}_v \geq 0 \quad (4.18)$$

식 (4.16)으로부터 식 (4.18)의 도함수는 다음과 같이 구해진다.

$$\begin{aligned} \dot{V}_1 &= \dot{\mathbf{e}}_v^T \mathbf{e}_v \\ &= \left\{ \mathbf{M}^{-1} (\boldsymbol{\tau} - D_\theta \boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d \right\}^T \mathbf{e}_v \end{aligned} \quad (4.19)$$

식 (4.19)에서, $\dot{V}_1 \leq 0$ 을 얻기 위해 다음 식이 성립되어야 한다.

$$\dot{\mathbf{e}}_v = \mathbf{M}^{-1} (\boldsymbol{\tau} - D_\theta \boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d = -\mathbf{c} \mathbf{e}_v \quad (4.20)$$

여기서 $\mathbf{c} = \text{diag}(c_1, c_2, c_3, c_4) \in \mathbf{R}^{4 \times 4}$ 이며 양의 한정 행렬이다.

식 (4.20)을 식 (4.19)에 대입하면 다음과 같다.

$$\begin{aligned} \dot{V}_1 &= \dot{\mathbf{e}}_v^T \mathbf{e}_v \\ &= (-\mathbf{c} \mathbf{e}_v)^T \mathbf{e}_v \\ &= -\mathbf{e}_v^T \mathbf{c} \mathbf{e}_v \leq 0 \end{aligned} \quad (4.21)$$

식 (4.21)로부터 $t \rightarrow \infty$ 일 때, $\mathbf{e}_v \rightarrow 0$ 임을 알 수 있다.

또한, 식 (4.15)와 식 (4.20)으로부터 식 (4.17)의 제어칙은 다음과 같이 유도된다.

$$\{\mathbf{M}^{-1}(\boldsymbol{\tau} - D_{\theta}\boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d\}^T = -\mathbf{e}_v^T \mathbf{c}$$

$$\mathbf{M}^{-1}(\boldsymbol{\tau} - D_{\theta}\boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d = -\mathbf{c}\mathbf{e}_v$$

$$\boldsymbol{\tau} - D_{\theta}\boldsymbol{\omega} = \mathbf{M}\dot{\boldsymbol{\omega}}_d - \mathbf{M}\mathbf{c}\mathbf{e}_v \quad (4.22)$$

$$\begin{aligned} \boldsymbol{\tau} &= D_{\theta}\boldsymbol{\omega} + \mathbf{M}\dot{\boldsymbol{\omega}}_d - \mathbf{M}\mathbf{c}\mathbf{e}_v \\ &= D_{\theta}(\mathbf{e}_v + \boldsymbol{\omega}_d) + \mathbf{M}\dot{\boldsymbol{\omega}}_d - \mathbf{M}\mathbf{c}\mathbf{e}_v \\ &= (D_{\theta} - \mathbf{M}\mathbf{c})\mathbf{e}_v + \mathbf{M}\dot{\boldsymbol{\omega}}_d + D_{\theta}\boldsymbol{\omega}_d \end{aligned}$$

Fig. 4.8은 본 연구에서 제안된 제어기의 블록선도를 나타낸다.

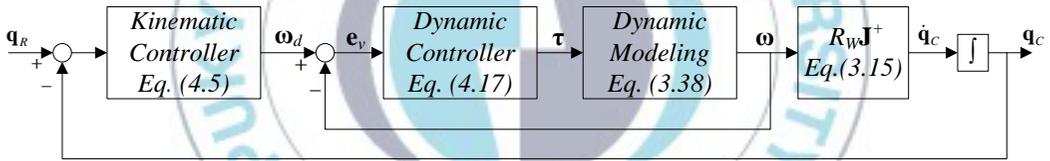


Fig. 4.8 Block Diagram for the Proposed Controller

제 5 장 시뮬레이션 결과

본 장에서는 제안된 전방향 주행형 차량의 경로계획 알고리즘과 경로추적 제어기에 대한 유효성을 보여주기 위해 시뮬레이션 및 실험결과를 보여준다. 시뮬레이션에 사용된 매개변수 값과 변수의 초기값은 Table 5.1과 Table 5.2와 같다.

Table 5.1 Parameter's Values of Omni-directional Mobile Vehicle

Parameter	Description	UNIT
l	0.25	m
L	0.22	m
R_w	0.21	m
D_θ	0.7	m^2/s
J_z	3.7927	$Kg \cdot m^2$
J_w	0.1875	$Kg \cdot m^2$
m	21	Kg

Table 5.2 Initial Values of Variables for Simulation

Parameter	Description	UNIT
X_0	40	<i>pixel</i>
Y_0	400	<i>pixel</i>
ϕ_0	0	<i>deg</i>
v_x	0	<i>m/s</i>
v_y	0	<i>m/s</i>
ω_z	0	<i>deg/s</i>

Fig. 5.1은 2장에서 제안된 A*알고리즘에 의해 계획된 전방향 주행형 차량의 전체 주행경로를 나타낸 것이며, 전체 시스템은 차량, 카메라 시스템, 세 개의 장애물, 시작점 및 목표점으로 구성된다. 4장에서 제안된 경로추적 제어기로 계획된 경로를 추적하며 이 때 전구간의 속도벡터는 속도추적 제어기로 차량이 목표속도 벡터를 추적하도록 제어된다. 차량은 인접한 노드간 0.1m/s의 일정한 속도로 주행한다.

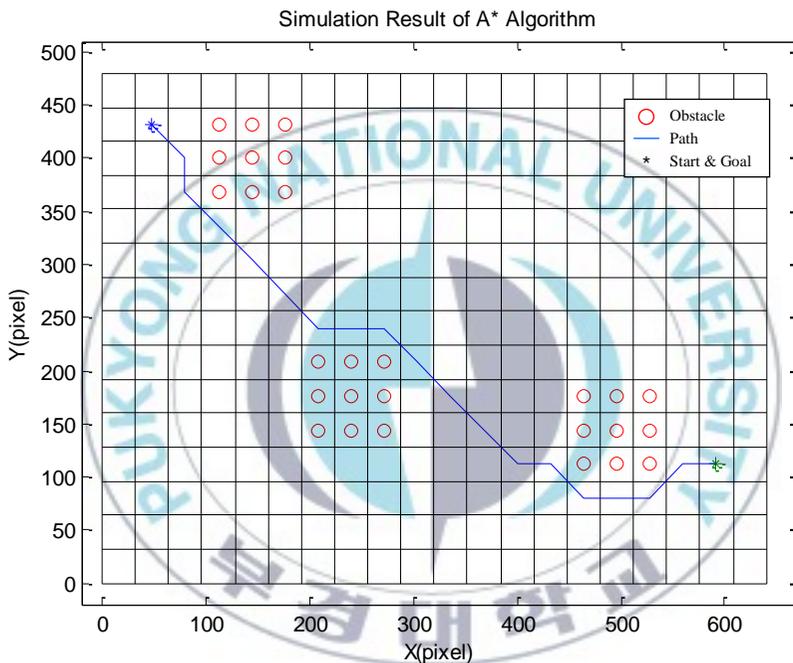


Fig. 5.1 Path Generated by A* Algorithm (Node Size : 32x32)

각 노드는 32x32 pixel 크기로 구성되며 시작지점과 목표지점은 *로 표시된다. 또한, 원으로 표시된 노드들은 영상처리로부터 획득한 장애물을 나타낸다.

맵의 전체 크기는 640x480 pixel이며, 가로 20개, 세로 15개, 총 300개의 노드로 구성되어 있으며, 1pixel은 실제거리 3.77mm로 나타난다. 따라서 각 노드의 중심거리는 32 pixel로 120.75mm이며, 대각거리는

45.25 pixel로 170.59mm이다. 시작지점의 노드좌표는 (2,14)이며, 목표 지점의 노드좌표는 (19,4)이며, 전체경로는 19개의 노드로 구성되어 있다(대각 노드11개, 직선노드8개). 1pixel 당 3.77mm이므로 전체경로의 거리는 753.75 pixel로 2.84m이다.

Fig. 5.2는 X방향의 목표 주행속도들을 나타낸 것이다. X방향 목표 속도는 시작점 노드1에서 노드2로 이동할 때, 감소하고, 노드2에서 노드3로 이동할 때 증가한다. 그 외의 노드에서는 일정하게 됨을 알 수 있다.

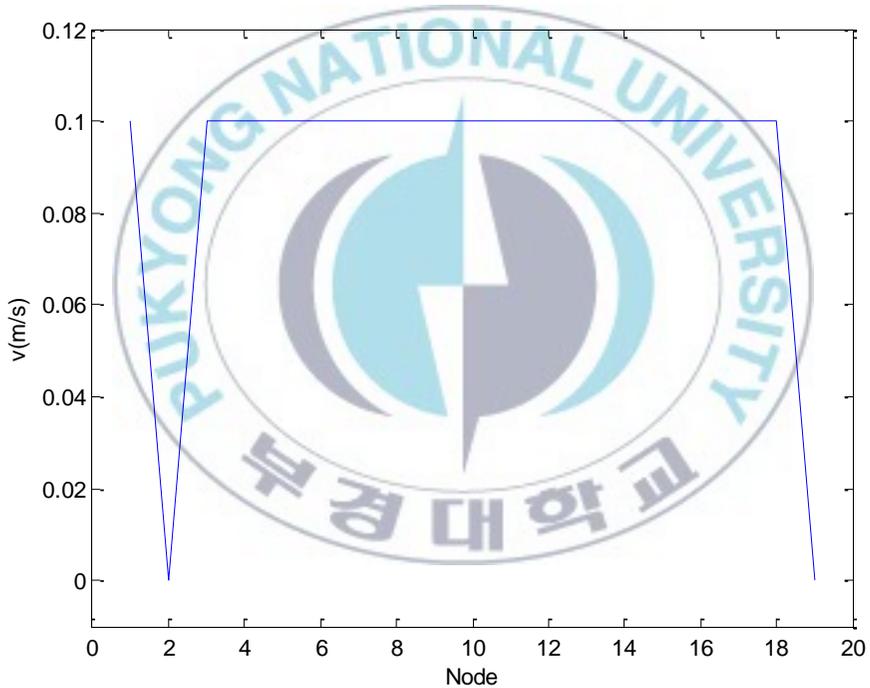


Fig. 5.2 Reference Velocity in X Direction

Fig. 5.3은 Y 방향의 목표 주행속도들을 나타낸 것이다. Y방향의 속도는 노드 6에서 7로 이동할 때 증가하고 노드 8에서 9로 이동할 때 감소한다. 다시 노드 12에서 증가하여 노드 13에서 다시 감소한다. 또한 노드 14에서 증가하여 노드 15에서 일정하게 유지되다 노드 16에서 다시 증가한다. 노드 17에서 감소하여 노드 18에서 일정하게 유지하며 노드 19에서 종료된다. 이와 같이 차량의 목표속도가 증가, 감소 또는 일정하게 유지함으로써 Fig. 5.1과 같은 경로를 생성할 수 있다.

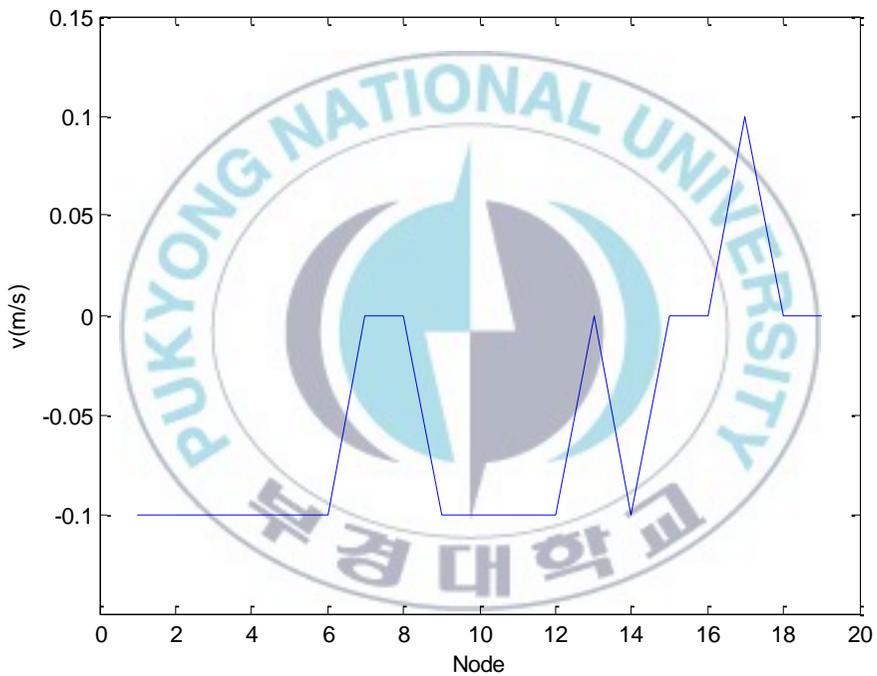


Fig. 5.3 Reference Velocity in Y Direction

Fig. 5.4는 Fig. 5.1에서 A*알고리즘에 의해 생성된 19개의 노드로 구성된 경로에서 각 노드의 진행방향각을 나타낸다.

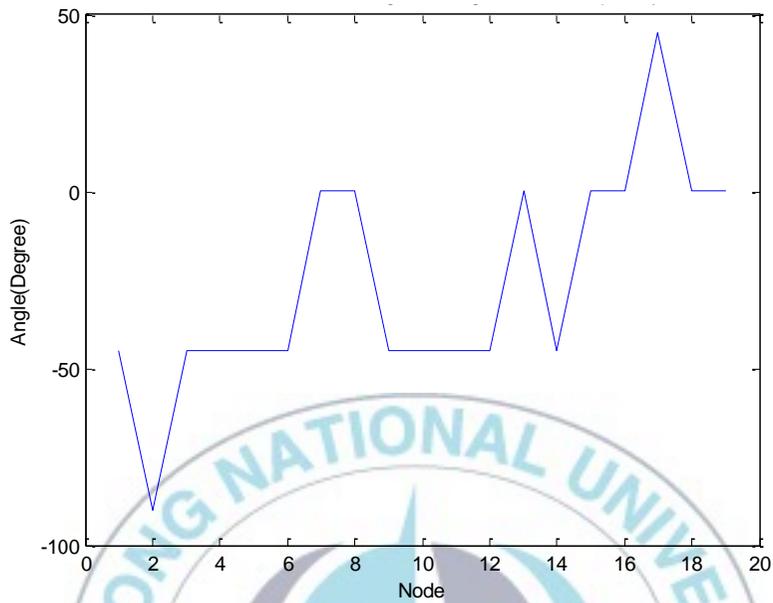


Fig. 5.4 Reference Direction Angle of Path

A*알고리즘에서 차량은 인접한 노드로 수평방향, 수직방향 또는 대각선 방향으로만 이동하기 때문에 진행방향각은 $\pm 90^\circ$, $\pm 45^\circ$, 0° 로 구성된다.

Fig. 5.5는 생성된 경로를 주행할 때 각 바퀴에서 발생하는 토크를 나타낸다.

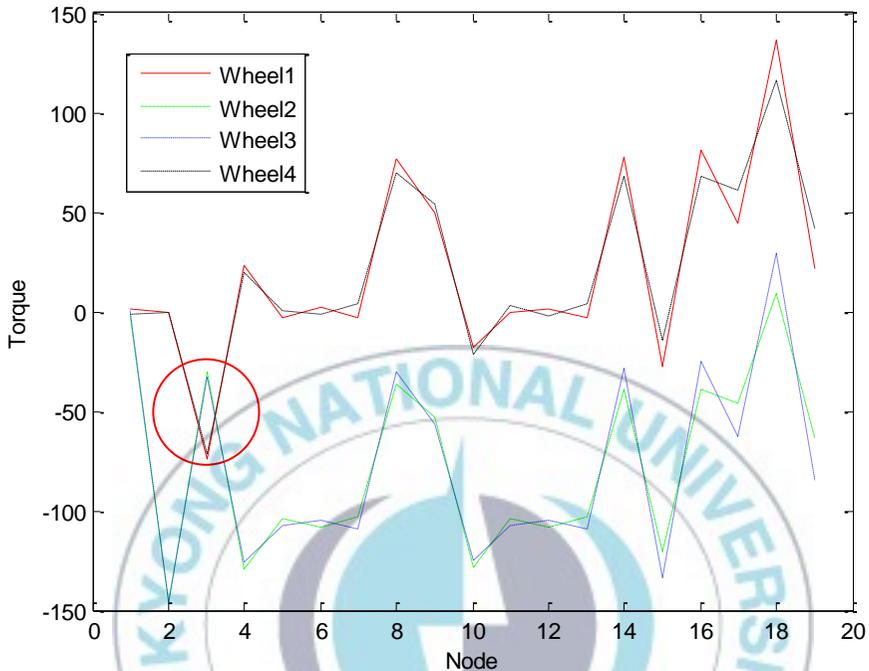


Fig. 5.5 Wheel's Torques

Fig. 5.5에서 대각선 상에 위치한 Wheel1과 Wheel4, Wheel2와 Wheel3의 토크가 각각 짝을 이뤄 비슷한 경향을 보이고 있다. 3번 노드에서 경로는 X축 방향에 수직으로 생성되었기 때문에 차량은 우측 방향으로 측면주행을 하게 된다. 우측방향의 측면주행을 위하여 Wheel2와 Wheel3의 토크가 Wheel1과 Wheel4보다 커짐을 확인할 수 있다. 그 밖의 구간은 우측 45° 방향으로 진행하는 경로가 지배적이고, 따라서 우측 45° 방향으로 진행하기 위해 Wheel1과 Wheel4의 토크가 Wheel2와 Wheel3의 토크보다 항상 큰 것을 확인할 수 있다.

Fig. 5.6는 경로추적 제어기의 시뮬레이션 결과를 나타낸다.

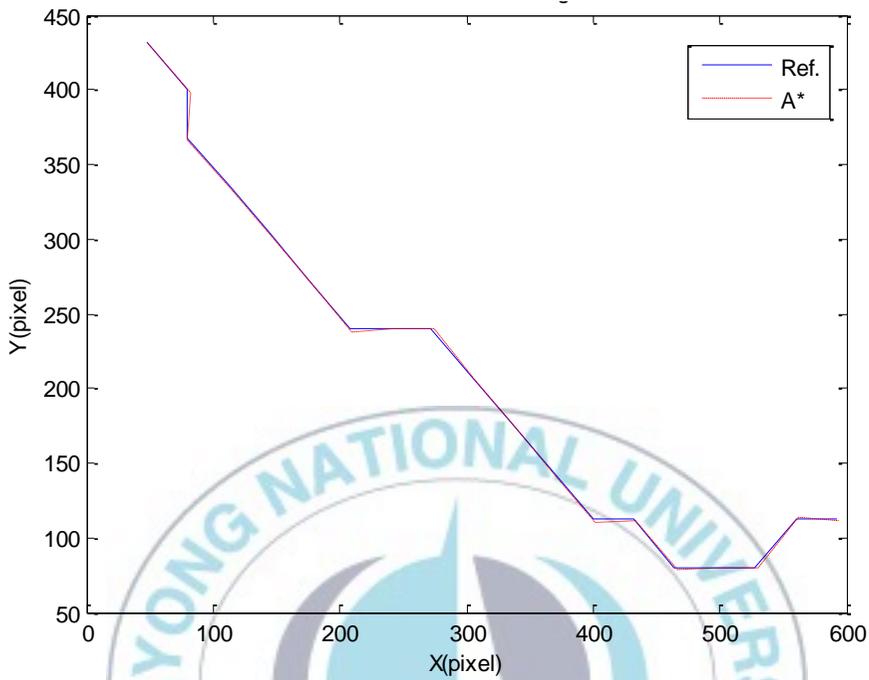


Fig. 5.6 Simulation Result of Trajectory Tracking Controller

차량의 경로가 대각선 방향으로 변경되는 지점에서 약간의 오차가 발생하지만 이내 목표경로에 잘 수렴함을 보이고 있으며, 전체적으로 목표경로를 대체로 잘 추적하고 있음을 보여준다.

Fig. 5.7은 A*알고리즘에 의해 생성된 경로에 본 연구에서 제안된 경로추적 제어기를 적용하여 시뮬레이션 하였을 경우 발생하는 차량의 X축 방향의 위치오차와 Y축 방향의 위치오차를 각 노드별로 나타내고 있다.

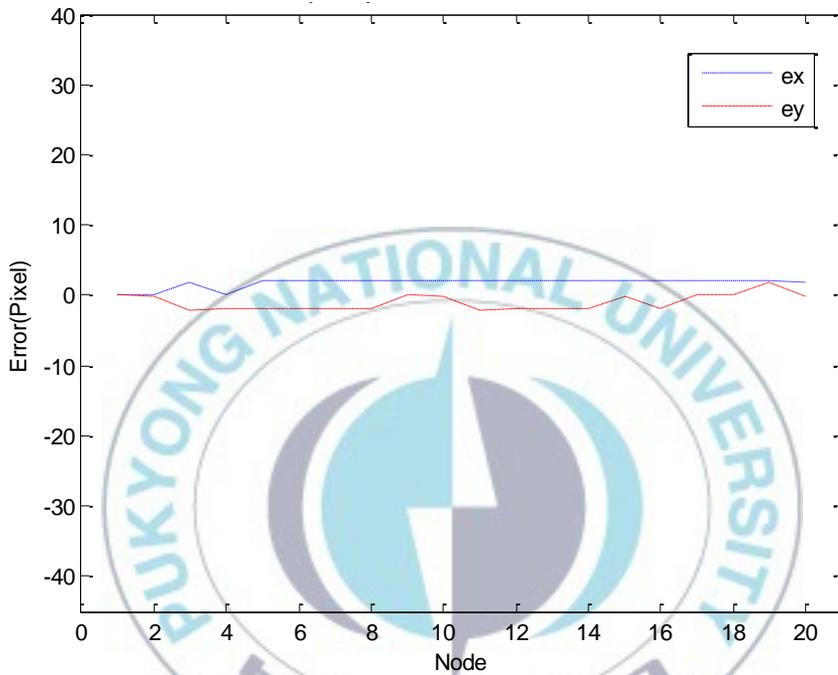


Fig. 5.7 Simulation Result of Path Tracking Error

차량이 2번 노드를 지나면서부터 X축 방향의 오차와 Y축 방향의 오차가 발생하기 시작하지만, 그 크기는 $\pm 5\text{pixel}$ 로 실제거리 오차는 18.85mm 이내이다. 이로부터 차량이 계획된 경로를 각 노드마다 큰 오차없이 대체로 잘 추적하는 것을 확인할 수 있다.

제 6 장 결 론

천장 설치형 카메라 시스템에 적용시켜 장애물 회피를 위한 경로를 생성 및 그 구현에 대한 시뮬레이션을 통해 그 효율성을 검증하였다. 본 연구의 결론은 다음과 같다.

- 천장설치형 카메라 시스템의 구성과 이를 이용하여 차량과 장애물 및 목표지점을 인식하고 중심좌표를 획득하기 위한 색상인식 기법과 ‘Small Window’ 알고리즘을 소개하였다.
- 전방향 주행형 차량의 구성과 운동학적 모델링과 동역학적 모델링을 제시하였다.
- 차량의 장애물 회피를 위한 경로계획법으로 A*알고리즘을 제시하였으며, A* 알고리즘에 의해 생성된 경로를 추적하기 위해 운동학적 모델링을 바탕으로 하여 Backstepping 기법을 사용한 경로추적 제어기를 제시하였다. 또한 실제 차량의 속도벡터가 경로추적 제어기를 출력인 속도벡터를 추적하기 위하여 동역학적 모델링을 바탕으로 Backstepping 기법을 사용한 속도추적 제어기를 제시하였다.
- A*알고리즘을 사용한 전방향 주행형 차량의 경로계획법과 설계된 경로를 추적하기 위한 운동학적, 동역학적 제어기에 대한 시뮬레이션 결과를 통하여 앞서 제안된 경로계획법과 제어 알고리즘에 대한 유효성을 보여주었다.

REFERENCE

- [1] 김달형, 정우진, “M-Space를 이용한 자동 주차를 위한 주차 경로 생성”, 한국지능로봇 하계종합 학술대회 논문집, pp. 551~558, 2001년
- [2] 김달형, 이국태, 정우진, 장효환, 윤달주, “궤적추종제어기를 이용한 무선조종 차량의 주차제어”, 한국지능로봇 하계종합 학술대회 논문집, pp. 448~455, 2002
- [3] 손민혁, 부광석, 송정훈, 김홍섭, “자율적 평행주차 제어시스템의 개발”, Transactions of KSAE, Vol. 11. 5, pp. 176~182, 2003
- [4] Yutaka Kanayama, Yoshihiko Kimura, Fumio Miyazaki and Tetsuo Noguchi “A Stable Tracking Control Method for an Autonomous Mobile Robot” IEEE International Conference on Robotics & Automation, pp. 384-389, 1990
- [5] 백운보, 김원호, “미끄럼 조향 로봇의 강인한 궤적 추종 제어에 관한 연구”, 한국공작기계 학회지, pp. 121-127, 2010

- [6] 김진환, “모바일 로봇의 경로 추종 제어”, 전자공학회 논문지, 46 (4), pp. 73-77, 2009
- [7] T. C. lee, C. Y. Tsai and K. T. Song, “Fast Parking Control of Mobile Robots: A Motion Palnning Approach with Experimental Validation”, IEEE Trasaction on Control Systems Technology, 12 (5), pp. 661-676, 2004
- [8] C. Stachniss and W. Burgard, “An Integrated Approach to Goal Directed Obstacle Avoidance under Dynamic Constraints for Dynamic Environments,” IEEE International Conference on Intelligent Robots and Systems, IROS, 1 (1), pp. 508-513, 2002.
- [9] 이근유, 서진호, 오명석, 김상봉, “격자형 용접선 추적을 위한 용접 이동로봇의 모델링 및 제어”, 2003년도 춘계학술대회 강연 및 논문집, 대한기계학회, pp. 923-928, 2003년
- [10] 임미섭, 임종훈, “경로 관측기를 이용한 차륜형 이동 로봇의 경로 추종”, 전기학회논문지, 48(11), pp. 1448-1454, 1999년
- [11] 김대광, 강동중, “복수의 동적 장애물에 대한 이동로봇의 최적경로설계”, 로봇공학회 논문지, 2 (2), pp. 183-190, 2007.

- [12] W. H. Wang and J.G. Juang, "Application of Localization System to WMR Path Planning and Parking Control", 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1677-1682, 2009
- [13] A. Elfes, "Sonar-based Real-world Mapping and Navigation", IEEE J. Robotics and Automation, 3 (3), pp. 249-265, 1987.
- [14] T. L. Chung, S. J. Ma, H. K. Kim, T. T. Nguyen and S. B. Kim, "Shortest Path Planning and Robust Sliding Mode Control of a Two-Wheeled Mobile Robot", 2006 The International Symposium on Advanced Mechatronics, Cybernetics and Automation, Vietnam, 2006
- [15] N. Tlale and M. D. Villiers, "Kinematics and Dynamics Modelling of a Mecanum Wheeled Mobile Platform", 15th International conference on Mechatronics and Machine Vision in Practice(M2VIP08), pp. 657~662, 2008
- [16] S. L. Dickerson and B. D. Lapin, "Control of an Omni-directional Robotic Vehicle with Mecanum Wheels", IEEE Transaction of intelligent transportation systems, pp. 0323~0328, 1991
- [17] P. Kiddee and A. Shimada, "A Controller Design on Person

Folowing Omni-directional Vehicle Robots”, SICE Annual Conference, pp. 1043~1047, 2007

- [18] 트란안킴, 김광주, 중탄람, 김학경, 김상봉, “천장설치형 카메라 시스템을 사용한 장애물 회피용 이동로봇의 경로설계법과 그 구현”, KSPE 한국정밀공학회, 21 (8), pp. 73-82, 2004년
- [19] S. L. Dickerson and B. D. Lapin, “Control of an Omni-directional Robotic Vehicle with Mecanum Wheels”, IEEE Transaction of intelligent transportation systems, pp. 0323~0328, 1991
- [20] P. Kiddee and A. Shimada, “A Controller Design on Person Folowing Omni-directional Vehicle Robots”, SICE Annual Conference, pp. 1043~1047, 2007
- [21] N. Tlale and M. D. Villiers, “Kinematics and Dynamics Modelling of a Mecanum Wheeled Mobile Platform”, International Conference on Mechatronics and Machine Vision in Practice , pp. 657~662, 2009
- [22] K. L. Han, O. K. Choi, J. W. Kim, H. S. Kim and J. S. Lee, “Design and Control of Mobile Robot with Mecanum Wheel”, ICROS-SICE International Joint Conference, pp. 2932~2937, 2009

- [23] P. Viboonchaicheep, A. Shimada and Y. Kosaka, "Positoin Rectification Control for Mecanum Wheeled Omni-directional Vehicles", Proceedings Of the 1998 IEEE International Conference on Robotics & Automation, pp. 854~859, 2003
- [24] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics", Journal of Robotic Systems, pp. 149-163, 1997
- [25] R. Gonzalez, F. Rodriguez, J.L. Guzman and M. Berenguel, "Localization and Control of Tracked Mobile Robots under Slip Conditions", IEEE Transaction of Mechatronics, pp.978-983, 2009
- [26] C. Samson and K. Ait-Abderrahim, "Feedback Stabilization of a Nonholonomic Wheeled Mobile Robot", IEEE/RSJ International Workshop on Intelligent Robots and Systems(IROS), pp. 1242-1247, 1991
- [27] W. Oelen, H. Berghuis, H. Nijmeijer and C. Canudas de Wit, "Hybrid Stabilizing Control on a Real Mobile Robot," IEEE Rob. Autom. Mag., June, pp. 16-23, 1995.
- [28] 김종수, 문종우, "2자유도 차동 구륜 이동로봇의 기구학 모델링과 Backstepping 기법을 이용한

궀적추적” 한국정보기술학회논문지, 5 (2), pp. 202-208,
2007

- [29] C. Samson, “Velocity and Torque Feedback Control of a Nonholonomic Cart,” in Lecture Notes in Control and Information Science, C. Canudas de Wit, Ed., Springer-Verlag, pp. 125–151, 1991
- [30] S. Koenig and M. Likhachev, "Incremental A*" In Advances in Neural Information Processing Systems 14, MIT Press, 2002.
- [31] P. Kiddee and A. Shimada, “A Controller Design on Person Following Omni-directional Vehicle Robots”, SICE Annual Conference, pp. 1043-1047, 2007
- [32] E. Maalouf, M. Saad and H. Saliah, “A Higher Level Path Tracking Controller for a Four-wheel Differentially Steered Mobile Robot”, Robotics and Autonomous Systems 54, pp. 23-33, 2005
- [33] R. Fierro and F. L. Lewis, “Control of Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics”, Proceedings Of the 34th conference on Decision & Control, pp. 3805-3810, 1995

- [34] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, “A Stable Tracking Control Method for a Nonholonomic Mobile Robot”, IEEE/RSJ International Workshop Intelligent Robots and Systems IROS ‘91, pp. 1236-1241, 1991
- [35] 김영규, “완만한 곡선 경로추적용 용접이동로봇의 비선형 적응제어”, 학위논문, 2005
- [36] 서진호, T. H. Bui, T. T. Nguyen, 김상봉, “용접용 이륜 이동로봇의 모델링 및 적응 추종 제어”, 2003년도 춘계학술대회 강연 및 논문집, 대한기계학회, pp. 786-791, 2003년
- [37] B. D. Novel, G. Bastin and G. Campion, “Dynamic Feedback Linearization of Nonholonomic Wheel Mobile Robots”, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2527-2532, 1999
- [38] T. P. Tran, T. L. Chung, H. K. Kim S. B. Kim and M. S. Oh, “Trajectory Tracking of Mobile Manipulator for Welding Task Using Sliding Mode Control”, The 30th Annual Conference of the IEEE Industrial Electronics Society, 2004
- [39] A. M. Bloch, M. Reyhanoglu and N. H. McClamroch, “Control and Stabilization of Nonholonomic Dynamic Systems,” IEEE Trans. Autom. Control, 1746–1757, 1992.

- [40] J. E. M. Salih, M. Rizon and S. Yaacob, "Designing Omni-Directional Mobile Robot with Mecanum Wheel", American Journal of Applied Science 3 (5), pp. 1831-1835, 1996
- [41] A. Shimada, S. Yajima, Parkpoom and K. Samura, "Mecanum-wheel Vehicle Systems Based on Position Corrective Control", Transactions on IEEE Robotics and Automation, pp. 2077~2082, 2009
- [42] I. Doroftei, V. Grosu and V. Spinu, "Omnidirectional Mobile Robot-Design and Implementation", Bioinspiration and Robotics : Walking and Climbing Robot, pp. 511~528, 2007
- [43] J. E. M. Salih, M. Rizon and S. Yaacob, "Designing Omni-Directional Mobile Robot with Mecanum Wheel", American Journal of Applied Science 3 (5), pp. 1831-1835, 1996
- [44] A. Shimada, S. Yajima, Parkpoom and K. Samura, "Mecanum-wheel Vehicle Systems Based on Position Corrective Control", Transactions on IEEE Robotics and Automation, pp. 2077-2082, 2009
- [45] G. Campion and G. Bastine, "Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots", IEEE Transaction on Robotics and

Automation, 12 (1), pp. 47-62, 1996

- [46] N. Tlale and M. D. Villiers, "Kinematics and Dynamics Modeling of a Mecanum Wheeled Mobile Platform", International Conference on Mechatronics and Machine Vision in Practice , pp. 657-662, 2009
- [47] I. Doroftei, V. Grosu and V. Spinu, "Omnidirectional Mobile Robot-Design and Implementation", Bio-inspiration and Robotics : Walking and Climbing Robot, pp. 511-528, 2007
- [48] J. E. M. Salih, M. Rizon and S. Yaacob, "Designing Omnidirectional Mobile Robot with Mecanum Wheel", American Journal of Applied Science 3 (5), pp. 1831-1835, 1996
- [49] K. Nagatani, S. Tachibana, M. Sofue and Y. Tanaka, "Improvement of Odometry for Omnidirectional Vehicle Using Optical Flow Information", Thesis, Okayama University, 2006
- [50] Z. J. Ill, X. D. Ji, J. W. Kim and Y. B. Kim, "A Study of Autonomous Parking for a 4-wheel Driven Mobile Robot", Proceedings of the 26th Chinese Control Conference, pp. 179-184, 2007
- [51] J. Borenstein and Y. Koren, "The Vector Field Histogram-fast

Obstacle Avoidance for Mobile Robots", IEEE J. Robotics and Automation, 7 (3), pp. 278-288, 1991.

- [52] K. L. Han, O. K. Choi, J. W. Kim, H. S. Kim and J. S. Lee, "Design and Control of Mobile Robot with Mecanum Wheel", ICROS-SICE International Joint Conference, pp. 2932-2937, 2009
- [53] F. G. Pin and S. M. Killough, "A New Family of Omnidirectional and Holonomic Wheeled Platforms for Mobile Robots", IEEE Transactions on Robotics and Automation, 10 (4), pp. 480-489, 1994
- [54] 문우성, "다중 코드 초음파 센서 네트워크를 이용한 군집로봇 위치인식 시스템", 학위논문, 2009
- [55] 강신욱, "레이저 센서와 초음파 센서를 이용한 이동로봇의 강인한 위치 추정 및 지도 작성", 학위논문, 2010
- [56] 임형수, 최승욱, 이장명, "모바일 로봇을 위한 RFID 센서공간에서 효율적인 위치인식 알고리즘", Journal of Control, Automation, and Systems Engineering, pp. 949-955, 13 (10), 2007
- [57] 이현정, 최규천, 이민철, 이장명, "RFID를 이용한

이동로봇의 위치인식 기술”, Journal of Control, Automation, and Systems Engineering, pp. 41-46, 12 (1), 2006

- [58] 이도경, “RFID기술을 이용한 이동로봇의 맵빌딩에 관한 연구”, 학위논문, 2009
- [59] 한영재, 박춘수, 이태형, 김기환, 은종필, “GPS 위치검지시스템 구성에 관한연구”, Journal of the Korean Institute of Electrical and Electronic Material Engineers, pp. 151-155, 21 (2), 2008
- [60] 염정남, 이금분, 박정진, 조범준, “GPS와 가속도계를 이용한 이동 물체의 위치 추정 시스템”, Journal of Korea Multimedia Society, pp. 600-607, 12 (4), 2009
- [61] 정준익, 노도환, “무한원점과 단일 비전 시스템을 이용한 자율주행을 위한 실시간 GPS 위치 데이터 보정”, 전자공학회, pp. 1045-1052, 41 (6), 2004
- [62] 임현, “한 대의 카메라를 이용한 이동 로봇의 동시간 위치인식 및 지도작성”, 학위논문, 2010
- [63] 차정희, “DEM과 산영상을 이용한 비전기반 카메라 위치인식”, 한국컴퓨터정보학회, pp. 177-185, 10 (6), 2005

- [64] 한현석, “모자이크 영상을 이용한 비전 기반의 위치 인식”, 학위논문, 2009
- [65] 이재경, 박영환, “능동형 인공표지를 이용한 이동로봇의 위치 인식”, 한국산학기술학회, pp. 64-69, 9 (1), 2008
- [66] 김병화, “패턴인식에 의한 카메라 위치/방향의 정밀 보정”, 학위논문, 1999
- [67] 유현중, 김우성, 장영범, “경계선 검출에 대한 인접 칼라 영역간 테두리 선 삽입 효과의 분석”, 한국전기전자학회, pp. 87-95, 10 (1), 2006
- [68] 이준영, “표본 기반 학습을 이용한 경계선 검출”, 학위논문, 2009
- [69] 윤지영, “차량 구성 요소 분할 및 색상 인식 알고리즘”, 학위논문, 2008
- [70] 박선미, 김구진, “PCA-SVM 기법을 이용한 차량의 색상 인식”, 정보처리학회, pp. 285-292, 15 (4), 2008
- [71] 이승규, “스테레오 비전을 이용한 물체의 형상인식 알고리즘 개발”, 학위논문, 2009
- [72]곽성환, “Vision 센서를 이용한 물체의 형상인식에 관한

연구”, 학위논문, 2009

- [73] T. Miyasaka, K. Koyama and I. Itoh., “Quantum Conversion and Image Detection by a Bacteriorhodopsin-based Artificial Photoreceptor.”, Science, pp. 255-344, 1992.
- [74] M. Vrhel, E. Saber, and H.J. Trussell, “Color Image Generation and Display Technologies,” IEEE Signal Processing Mag., pp. 23–33, Jan. 2005.



Publications and Conferences

- [1] 이경목, 윤석민, 이도경, 여태경, 박성재, 홍섭, 김상봉, “근해역 채광시스템 성능 실증 시험용 통합운용 소프트웨어 개발”, 한국해양공학회 추계 학술대회, pp. 192-196, 2009
- [2] Do Kyoung Lee, Kyeong Mok Lee, Hak Kyeong Kim, and Sang Bong Kim, “A Study on Map Building of Mobile Robot using RFID Technology”, International Symposium on Mechanical Engineering, pp.61-66, 2009
- [3] 홍섭, 김형우, 최종수, 여태경, 박성재, 윤석민, 이창호, 이도경, 이경목, “심해저 망간단괴 시험집광기와 근해역 성능 실증시험”, 한국해양공학회 추계 학술대회, pp.293-296, 2009
- [4] 박성재, 여태경, 홍섭, 윤석민, 이경목, 김상봉, “부스트로피돈 셀 분해 기법을 이용한 집광기의 집광 경로 계획법에 관한 연구, 한국해양공학회지, 2010
- [5] Nak-Soon Choi, Gyeong-mok, Hak-Kyeong Kim, Sang-Bong Kim, "Optimal Sliding Mode Tracking Controller Design for Mobile Inverted Pendulum" Proceedings of the 2009 International Symposium on Advanced Engineering, pp. 205-208, 2009.03. 19-21
- [6] 김대원, 이경목, 박철한, 김학경, 손정현, 김상봉, 백운경, 변기식, “정찰 및 전투지원용 차량 매니퓰레이터의 개발”, 네트워크 기반 동시.통합전 구현을 위한 제16회 지상무기학술대회 pp. 109, 2008.11.11

감사의 글

먼저 본 논문이 완성되기까지 부족한 저를 시종일관 지켜 봐주시고 지도해주신 지도교수님, 김상봉 교수님께 진심으로 머리숙여 감사드립니다. 제가 학업에 지쳐 힘들어 할 때에는 교수님이 아닌 아버지의 입장에서 제가 살아가야 할 인생에 대하여 깊은 관심과 함께 항상 바른 길로 인도해주신 교수님의 은혜에 다시 한번 깊이 감사드립니다.

논문 심사과정에서 따끔한 질책과 아울러 격려를 아끼지 안으셨던 정영석 교수님과 손정현 교수님께도 깊은 감사의 말씀을 드립니다.

또한 학문적인 부분과 함께 과제를 진행함에 있어서 항상 옆에서 지켜 봐주시며 조언과 함께 제가 실수하지 않도록 저를 챙겨주셨던 김학경 교수님께도 다시 한번 깊이 감사드립니다.

석사과정 중 연구활동에 많은 도움을 주신 해양연구원 해양시스템연구부의 홍섭 박사님, 전봉환 박사님, 김형우 박사님, 최종수 박사님, 여태경 박사님, 박성재 연구원님, 김성순 연구원님, 성기영 연구원님, 이창호 연구원님 및 김대원 연구원님께도 깊은 감사의 말씀을 드립니다.

그리고 제가 4년동안 몸담은 CIMEC Lab.의 선배님이신 김성민 선배님, 김진호 선배님, 서진호 선배님, 감병오 선배님, 이근유 선배님, 김석열 선배님, 신승목 선배님, 이원기 선배님, 김성욱 선배님, 전양배 선배님, 김영규 선배님, 임재성 선배님, 김상찬 선배님, 박바다 선배님, 이희숙 선배님, 김대원 선배님, 이도경 선배님, 최낙순 선배님, 정준호

선배님께 감사 드립니다. 또한 타지에서 공부하며 많은 도움 주셨던 Phoung, Duy, Hung과 직장생활과 학업을 병행하며 많은 도움을 주신 신중훈 형에게 감사 드립니다.

또 현재 CIMEC Lab.에서 더 나은 미래를 위해 밤낮으로 열심히 공부하고 있는 동기, 후배님께도 감사 드립니다. 먼저 CIMEC Lab.에 들어올 수 있었던 계기를 마련해주고, 학부와 대학원 생활 4년동안 동고동락하며 함께 고생하고 많은 도움 주었던 박철한 군에게 고마움을 전합니다. 아울러 논문준비를 위해 함께 밤새가며 고생하고, 힘들 때 서로 의지하며 형제같이 지낸 김대환 형과 김서광, Think에게 감사드립니다. 그리고 실험실 생활에 많은 도움을 주었던 후배 Tuan, Giang, 박유미, 최하연, 배민지양, 한상진, 정우영군에게도 감사 드립니다.

초등학교부터 지금까지 항상 함께하며 힘이 되어준 고우회 친구들 김동훈, 김왕영, 김유현, 김정우, 김주영, 맹경환, 명현우, 손정, 손효섭, 안정진, 양승현, 윤필원, 조민호, 최성균과 김종현, 이상한에게 감사 드립니다.

마지막으로 항상 저를 믿어주시고 응원해주시며 제가 가장 사랑하는 아버지, 어머니 그리고 동생에게도 진심으로 감사의 마음을 전합니다.

2011년 2월

이 경 목

