



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

수중 집광 로봇용 커버리지 경로계획 및
추적 제어기 설계



2016년 2월

수중 집광 로봇용 커버리지 경로계획 및 추적 제어기 설계



2016년 2월

부 경 대 학 교 대 학 원

메카트로닉스공학협동과정

김 진 욱

공 학 석 사 학 위 논 문

수중 집광 로봇용 커버리지 경로계획 및
추적 제어기 설계

지도교수 김 상 봉

이 논문을 공학석사 학위논문으로 제출함.

2016년 2월

부 경 대 학 교 대 학 원

메카트로닉스공학협동과정

김 진 욱

김진욱의 공학석사 학위논문을
인준함.

2015년 12월

주심 공학박사 정영석



위원 공학박사 변기식



위원 공학박사 김상봉



목 차

목차	i
Abstract.....	iv
제 1 장 서론	1
1.1 연구 배경 및 동기	1
1.2 문제 제기	5
1.3 연구 목적 및 방법	7
1.4 연구 내용 및 범위	9
제 2 장 전체 커버리지 경로계획	11
2.1 전체 커버리지 경로계획의 개요	11
2.2 전처리	14
2.3 셀 분할 및 회전 수 최소화	16
2.3.1 모스 셀 분할법	16
2.3.2 셀 분할 최적화	23
2.4 커버리지 순서 결정	30
2.5 셀 간의 이동경로 생성	38
2.6 중복된 경로의 최소화	42
2.7 전체 커버리지 경로 생성	45
2.8 전체 커버리지 경로계획의 시뮬레이션	50

제 3 장 시스템 구성	61
3.1 기구 시스템의 구성	62
3.2 제어 시스템의 구성	64
3.2.1 센서부	65
3.2.2 제어부	71
3.2.3 구동부	74
3.2.4 전원 공급장치	75
 제 4 장 시스템 모델링 및 제어기 설계	77
4.1 AGV 구동 바퀴의 특성	77
4.1.1 운동학적 모델링	77
4.1.2 구동 바퀴(표준 구동 바퀴)	79
4.1.3 수동 바퀴(캐스터 바퀴)	80
4.2 차동 구동 AGV 시스템의 운동학적 모델링	82
4.3 AGV 제어기 설계	88
 제 5 장 시뮬레이션 및 실험 결과	92
5.1 시뮬레이션 결과	93
5.2 실험 결과	100
 제 6 장 결론 및 향후 연구 방향	109
6.1 결론	109
6.2 향후 연구 방향	113

감사의 글.....	114
참고문헌.....	116
학술논문 발표 및 게재.....	121



Coverage path planning and its tracking controller design for an underwater mining robot

Jinwook Kim

Dept. of Interdisciplinary Program of Mechatronics Engineering,

The Graduate School, Pukyong National University

Abstract

This thesis proposes a coverage path planning method for an underwater mining robot to cover a given underwater region effectively and a tracking controller design to track the planned path generated by the coverage path planning method. The coverage path planning method includes preprocessing of a given 3D map, cell decomposition and minimal sum of altitude algorithm for minimizing turning number, an algorithm choosing coverage sequence of each cell, generating moving path between cells by modified sweeping line strategy to reduce an overlapped paths and produce a smooth path. For this work, the following tasks are implemented. Firstly, in preprocessing, a given 3D map is transformed to a binary map. Then, Morse decomposition method is used for decomposing the transformed binary map and minimal sum of altitude algorithm is applied to minimize a turning number. To

define the coverage sequence of each cell, comparison between depth-first-searching algorithm and breath-first-searching algorithm based on spanning tree is proposed. After defining the coverage sequence, an entire coverage path and a moving path between cells using visibility graph algorithm are generated. For minimizing overlapped paths, modified sweeping line strategy is used when the length of overlapped paths is too long. Turning paths of the entire coverage path are smoothened. Secondly, an automated guided vehicle(AGV) with two passive wheels in front and back sides and two active driving wheels in left and right sides of the mobile robot is proposed. Kinematic modeling of this AGV system is presented. Based on the kinematic modeling, a backstepping controller is proposed to track the generated coverage path obtained from the proposed coverage path planning algorithm. Thirdly, to implement the proposed controller, a control system is developed based on microcontroller AVR ATmega128. A laser navigation system is used to measure the rotation angle and its position information of the proposed AGV. Finally, simulation and experiment results are presented to show the effectiveness and the applicability of the proposed coverage path planning algorithm and the proposed tracking controller for AGV system.

Keywords : Complete coverage path, Breadth-first search algorithm, Minimal sum of altitude algorithm, Automated guided vehicle, Backstepping controller design

제 1 장 서 론

1.1 연구배경 및 동기

육상광물자원의 고갈에 따른 대안으로 해저의 광물자원 개발에 대한 관심이 점점 높아지고 있다. 바다와 해저에는 인류가 최소 수백 년 내지 수천 년 이상을 쓸 수 있는 풍부한 광물자원이 다양한 형태로 존재한다. 심해저 광물자원으로 대표적인 것은 해수로부터 화학적 침전에 의하여 형성되는 망간단괴(Manganese Nodules)이다. 이 망간단괴는 다금속 단괴라고도 불리며 망간, 철, 니켈, 구리, 코발트 등의 금속이 집중되어있다. 국내 기간산업과 첨단산업이 발전하면서 금속자원의 대부분을 수입에 의존하고 있는 우리나라로서는 장기적이고 안정적인 금속광물자원의 공급원 확보가 시급한 실정이다. 우리나라는 유엔해양법협약에 따라 1994년 8월 태평양 공해상 클라리온-클리퍼톤 (Clarion-Clipperton, C-C) 해역에서 심해저 망간단괴 광구를 국제해저기구에 등록함으로써 세계에서 7번째 망간단괴 광구를 확보한 국가가 되었으며, 그 동안의 탐사활동 결과를 토대로 2002년 8월 독점적 개발광구를 확보하였다.

망간단괴 집광기술의 연구를 위하여 한국해양과학기술원 부설 선박해양플랜트연구소(KRISO)는 상업용 집광능력의 1/20 규모의 시험 집광로봇 미내로(Minero)를 개발하여 2009년과 2010년에 2차례의 근해역 실증시험을 수행하였다. 그리고 2013년 7월 해저 1,380m 깊이에서 주행에 관한 성능 및 망간단괴를 집광하는 성능 실증시험을 수행하였다[1].

집광로봇은 해저면을 주행하면서 미리 탐사를 통해서 주어진 영역의

망간단괴를 집광하게 된다. 탐사를 통해서 주어진 영역은 매우 넓으며 복잡한 경계와 다양한 장애물을 포함한다. 집광로봇은 원격 자율주행로봇으로 사전에 계획된 경로를 따르게 된다. 그러므로 집광로봇이 망간단괴를 시간적/경제적 효율에 맞추어 집광하기 위해서는 주어진 영역에 대한 커버리지 경로계획이 이루어져야 한다.

최근 이동로봇의 경로계획에 관한 연구가 활발히 진행되면서, 커버리지 경로계획을 위한 다양한 방안들이 연구되고 있다. 이러한 연구들은 잔디 깎기[2], 농작물의 수확[3,4], 지뢰나 자원의 탐사[5,6], 그리고 절삭공구의 경로[7,8] 등 여러 적용분야를 가진다[9]. 경로계획 문제는 그 방식에 따라 몇 가지 형태로 분류될 수 있으며, 셀 분할을 이용한 방식 또한 그러한 방식들 중에서 주요한 하나의 기법으로 분류되고 있다.

일반적으로 이동로봇에 대한 경로계획은 크게 점대점 방식(Point-to-Point, PTP)[10]과 전체 커버리지 경로계획(Complete Coverage Path Planning, CCPP)방식[11]으로 분류된다. PTP 방식이란 우편배달 로봇과 같이 주어진 두 점 사이를 최적의 경로를 이루면서 이동하는 방법을 말하며, CCPP 방식은 청소로봇 또는 지뢰탐사로봇과 같이 이동로봇의 작용기 (Effector)가 주어진 영역의 자유 영역(Free space)을 모두 지나가는 경로를 생성하는 전체 영역을 이동하여야 하는 경우에 적용되는 방법을 말한다.

전체 커버리지 경로계획은 대표적으로 템플릿(Template)을 이용한 방식[12], Potential Field를 이용한 방식[13], 셀 분할을 이용한 방식[14]으로 분류할 수 있다. 템플릿을 이용한 방식은 몇 개의 기본 경로 계획 템플릿과 청소 환경을 이용하여 경로계획을 한다. 로봇과 환경 지도를 기반으로 가장 적당한 템플릿을 이용한다. 하지만 템플릿을 이용한 방식은 주어진 영역 및 분할되는 셀의 형태에 따라 많은 템플릿을 필요로 하는 경우가 발생하기 때문에 많은 계산 시간이 필요하다. M. J. Bae 등은 주어

진 영역을 간단한 형상을 가지는 셀로 분할하고 템플릿 기반 경로를 이용하여 각 셀의 커버리지 경로를 생성한 이후에 인접 그래프를 생성하여 셀 커버리지 순서를 결정하는 방식을 통해 주어진 영역의 경로 계획을 연구하였다[15]. 하지만 주어진 영역이 3D 맵인 경우에 대해서는 다루지 않았다. Potential Field를 이용한 방식에서는 청소 영역을 수치적 값을 가진 셀로 표현한다. 방문한 셀의 값은 증가하고 주위에서 가장 낮은 값을 가진 셀로 이동하는 경로 계획을 말한다. 이때 발생하는 많은 방향전환을 줄이기 위하여 셀의 값으로 거리함수와 방향전환 함수를 이용하였다[16]. Potential Field를 이용한 방식은 지면에 굴곡이 있는 경우 문턱값을 이용하여 처리할 수 있지만, 그로 인해 장애물 및 목표점 사이에 Potential에서 수치적 문제가 발생할 때가 있다. 셀 분할을 이용한 방식은 청소공간을 셀로 나뉘어서 해석하는 경로계획을 말한다. Choset은 장애물을 만나는 임계점에서 셀들을 분리하고, 장애물을 벗어나는 임계점에서 나뉘어진 셀들을 합치는 Boustrophedon 셀 분할방식을 이용해 전체 영역을 방문하는 Morse decomposition 방식을 연구하였다[17]. 이 방법은 주로 각 셀의 커버리지에 대해서만 다루어졌고 셀과 셀 간의 이동경로에 대해서는 다루지 않았다.

또한 커버리지 경로계획 등을 통해 설계된 경로를 정확히 추적하기 위한 이동로봇의 제어에 관한 연구도 활발히 진행 중이다. Y. D. Setiawan 등은 이륜이동로봇의 경로 재계획 문제에서 리아푸노프 안정성을 기초한 백스텝핑(Backstepping) 기법을 사용하여 비선형 제어기를 설계하였다[18]. P. S. Pratama 등은 레이저 측정 시스템을 바탕으로 차동 구동 무인 운반차를 개발하고 칼만필터를 적용하여 이동하는 물체에 대한 추적, 장애물을 회피하는 연구를 제시하였다[18]. 서진호 등은 이륜 용접용 이동로봇의 리아푸노프 함수에 기초한 적응 경로 추적 제어 알고리즘을 설계하였다[19]. R. Fierro 등은 비홀로노믹 2륜 구동로봇의 운동학적 모델링과 동

역학적 모델링에 대해 다루었으며, 주어진 경로를 추적하기 위한 백스텝핑 제어기(Backstepping controller)를 제시하였다[20]. 위와 같은 이동로봇의 모델링 및 제어에 관한 연구는 많이 진행되고 있다.

이와 같이 커버리지 알고리즘과 커버리지 경로의 생성, 이동로봇의 제어와 구현에 많은 연구가 진행되어 왔다. 하지만 장애물을 가지는 영역에서 이동로봇의 영역 커버리지와 효율적인 셀 간의 이동경로를 포함하는 전체 커버리지 경로계획 연구는 아직 수행되지 않았다. 또한 커버리지 알고리즘을 제시한 많은 연구들은 실험이 병행되지 않아 실제 영역에서의 유효성을 입증하기 어려웠다. 따라서 전체 커버리지 경로계획과 이렇게 계획된 경로를 정확하게 추적할 수 있는 이동로봇의 제어에 대한 연구가 필요하다.



1.2 문제 제기

본 연구에서는 주어진 영역에 대한 커버리지 경로계획을 바탕으로 생성된 커버리지 경로에 대하여 이론 이동로봇의 경로 추적을 다룬다. 커버리지 경로계획에서는 모스 셀 분할법(Morse cell decomposition)을 이용하여 주어진 바이너리(Binary) 영역을 셀로 분할하는 방법을 소개한다. 바이너리 영역을 바탕으로 커버리지 경로 생성에 대한 연구는 김시종 등 [21]이 수행한 바 있지만, 장애물 영역의 처리에 대해 상세히 설명하지 않았다.

Huang은 시간적으로 최적화된 커버리지에 초점을 맞춘 영역 구분 방법을 제안했다. 이 방법에서는 이동로봇의 방향 전환 횟수가 최적화의 기준으로 사용된다. 이동로봇의 특성상 로봇이 방향을 전환할 때 많은 시간이 소요되는 것을 고려한 방법이다. 이 방법은 최적의 경로를 생성할 수 있지만 간단한 환경에 대해서도 너무 많은 계산 시간이 필요하다는 단점이 있다[22]. Yao의 방법은 셀 사이의 불필요한 이동 경로를 최소화하는 것을 통해 최적화된 커버리지를 수행하도록 하는 방법이다. 이 방법의 요점은 셀에서의 진입 지점과 최종 이탈 지점이 주변 셀과 가까워야 한다는 점이다[23].

E. U. Acar 등은 주어진 영역을 여러 슬라이스 함수(Slice function)를 이용하여 영역을 분할하고, 각 슬라이스 함수에 의해 분할된 여러 모양을 가지는 셀에 대해 커버리지 경로를 계획하였다[24]. 여러 슬라이스 함수를 통해 셀을 분할하는 방법은 적용분야에 따라 실용적일 수 있으나, 실제 현장에서는 실용적이지 못하고, 셀을 분할하는 방법이 복잡하여 적용이 쉽지 않다. 또한 시뮬레이션을 통해 생성된 경로는 중복이 많으며 영역의 완전 커버리지에 적용하기엔 불충분하다고 판단된다.

본 연구에서는 이륜 집광로봇을 이용하여 망간단괴와 같은 심해저 광물자원의 채광을 위해 주어진 영역 전체를 효율적으로 커버하기 위하여 해결해야 하는 다음과 같은 문제가 있다.

1. 효율적인 커버리지 수행을 위해 주어진 3D 영역을 간단한 2D 영역으로 변환하여 채광 영역과 장애물 영역으로 구분하고, 이 영역들을 셀로 분할하는 방법을 제시하는 것.
2. 격자 형태의 맵을 바탕으로 집광로봇의 회전수를 최소화하기 위한 알고리즘을 제시하는 것.
3. 셀 간의 이동경로를 최소화하고 효율적인 경로를 생성하기 위한 알고리즘을 제시하는 것.
4. 중복된 경로를 최소화하기 위한 알고리즘을 제시하는 것.
5. 전체 커버리지 경로를 생성하는 것.
6. 이륜 집광로봇과 유사한 동작성을 갖춘 차동 구동 무인 운반차의 구성하는 것.
7. 제시된 커버리지 경로를 추적하기 위한 경로 추적 제어기를 설계하는 것.

1.3 연구목적 및 방법

지금까지 영역 커버리지에 대한 연구는 많이 이루어져 왔다[1-10]. 하지만 이러한 연구는 셀 분할 기법을 이용하여 각 셀의 커버리지 경로를 생성하지만 커버리지 경로의 효율성을 보장하지 못하고, 이동로봇의 방향 전환 횟수를 최소화하지만 필요한 계산시간이 상당히 크다. 또한 각 셀의 커버리지 경로 연구에 집중하여 셀 간 이동경로 생성 및 중복되는 경로의 수정에 대한 연구는 미숙하다. 또한 영역 커버리지에 대한 대부분의 연구들은 실제 실험을 수행하지 않고, 시뮬레이션 결과만을 제시한다.

본 연구의 목적은 위와 같은 단점을 보완하고, 실제 실험을 제시하여 제안된 전체 커버리지 경로계획과 경로 추적 제어기의 유효성을 입증하기 위해 셀 분할 기법에 기반한 전체 커버리지 경로계획 및 경로계획에 의해 생성된 경로를 집광로봇과 유사한 동작성을 갖춘 무인 운반차가 추적하는 경로 추적 제어기를 제안하는 것이며, 본 연구의 목적을 달성하기 위해서 다음과 같이 수행한다.

첫째, 각 셀의 커버리지 경로계획과 셀 간의 이동경로 계획을 포함하는 전체 커버리지 경로계획을 제안한다. 셀의 커버리지 경로계획을 위해 주어진 3D 영역을 경사기준값에 따라 2D 바이너리 (Binary)영역으로 변환 후 모스 셀 분할법(Morse cell decomposition) 및 최소 길이 계산 알고리즘(Minimal sum of altitude algorithm, MSA)을 적용한다. 셀 간의 커버리지 순서 결정을 위해 너비우선 탐색 알고리즘(Breadth first search, BFS) 및 깊이우선 탐색 알고리즘(Depth first search, DFS)이 이러한 커버리지 순서는 제안된 신장트리(Spanning tree)를 통해 구해진다. 또한 셀 간의 이동경로를 구성하기 위해 가시성 그래프 알고리즘(Visibility graph algorithm)을 적

용하며 과도한 경로의 중복을 피하기 위해 수정 스위핑 라인법(Modified sweeping line strategy)이 사용된다. 최적 셀 분할 결과, 셀 커버리지 순서, 셀 간의 이동경로, 과도한 중복 방지를 위한 경로를 통합하여 전체 커버리지 경로를 생성하며, 전체 커버리지 경로가 생성된 이후에 회전구간에 대한 처리를 함으로써 연속적인 전체 커버리지 경로를 생성한다. 시뮬레이션을 통해 본 연구에서 제안된 전체 커버리지 경로계획의 유효성을 검증한다. 둘째, 계획된 경로의 추적을 위해, 이륜 집광로봇과 유사한 동작성을 갖춘 앞뒤 2개의 수동 바퀴와 좌우 2개의 구동 바퀴를 가진 무인 운반차 시스템이 구성된다. 셋째, 이러한 시스템의 운동학적 모델링을 제시하고, 운동학적 모델링을 기반으로 백스텝핑 제어기(Backstepping controller)를 설계한다. 넷째, 설계된 제어기를 구현하기 위해 제어시스템을 개발한다. 제어연산을 위해 산업용 PC TANK-800을 이용하고, 로봇의 회전 각도를 측정하기 위하여 가속도 센서 MySen-M과 로봇의 상대좌표의 측정하기 위한 레이저 항법 시스템(Laser Measurement System)을 사용한다. 제안된 경로 추적 제어기의 유효성을 검증하기 위해 레이저 항법 시스템을 장착한 이륜 무인 운반차 시스템을 이용하여 계획된 전체 커버리지 경로를 추적하며 그에 따른 시뮬레이션과 실험을 제시한다.

1.4 연구내용 및 범위

본 연구는 전체 커버리지 경로계획법을 제안하고, 이 제안된 경로계획법에 의해 계획된 경로를 추적하기 위한 이륜 무인 운반차 시스템의 구성 및 제어 알고리즘을 제안한다. 또한 주어진 간단한 영역에 대한 전체 커버리지 경로계획법 및 제어 알고리즘을 구현하기 위해서 실제 로봇을 이용하여 시뮬레이션 및 실험을 통해 그 유효성을 검증한다.

본 연구는 5장으로 구성되어 있으며, 각 장의 내용은 다음과 같이 요약된다.

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

제 2 장 : 주어진 영역의 커버리지를 위한 전체 커버리지 경로계획을 제안하여 목표경로를 생성한다. 이러한 경로계획은 영역 변환, 셀 분할, 셀 분할 최적화, 커버리지 순서 결정, 셀 간 이동경로, 과도한 중복 경로의 수정, 회전 구간의 처리를 통해 수행된다. 또한 시뮬레이션 결과를 제시하여 전체 커버리지 경로계획법의 유효성을 검증한다.

제 3 장 : 경로를 추적하기 위한 집광로봇과 유사한 동작성을 갖춘 이륜 무인 운반차 시스템을 구성한다.

제 4 장 : 이륜 무인 운반차 시스템의 운동학적 모델링을 제시하고, 이를 기준한 백스텝핑(Backstepping) 기법을 사용하여 전체 커버리지 경로계획법에 의해 생성된 목표경로를 추적하기 위한 비선형 경로 추적 제어기를 설계한다

제 5 장 : 전체 시뮬레이션과 실험을 위해 레이저 항법 시스템을 장착한 무인 운반차 시스템을 사용한다. 계획된 전체 커버리지 경로계획법과 제안된 비선형 경로 추적 제어기의 유효성을 검증하기 위해 시뮬레이션 및 실험 결과를 제시한다.

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

제 2 장 전체 커버리지 경로계획

본 장은 영역 변환, 모스 셀 분할, 셀 분할 최적화, 커버리지 순서 결정, 셀 간 이동경로, 중복 경로를 최소화한 수정경로 및 회전구간을 부드럽게하여 전체 커버리지 경로인 목표경로를 생성하는 전체 커버리지 경로계획(Complete coverage path planning, CCPP)법을 제안하며 또한 시뮬레이션을 통해 그 유효성을 검증한다.

2.1 전체 커버리지 경로계획의 개요

본 절은 전체 커버리지 경로계획의 개요에 대해 소개한다. 망간단괴 분포지역 전체를 커버하면서 망간단괴를 채집하는 경로계획과, 집광로봇의 운용비용 및 시간을 줄이기 위해 불필요한 경로 및 중복되는 경로의 최소화를 위한 내용을 포함한다.

커버리지 경로계획이란 집광로봇이 장애물을 회피하면서 주어진 모든 영역을 방문할 수 있는 경로를 계획하는 것이다. 본 연구에서는 다루어지는 영역과 장애물은 간단한 형상을 가지며 3D 영역으로 존재한다. 또한 주어진 영역에 대한 지형적 정보가 주어져서, 주어진 정보 이외에 변수가 없는 전역적 경로계획을 다룬다.

복잡한 경계와 장애물을 가지는 3D 영역의 커버리지 경로계획은 집광로봇이 망간단괴를 채집하기 위한 경로로써 적합하지 않기 때문에, 먼저 전처리로 주어진 3D 영역을 2D 바이너리 영역으로 변환한다. 변환된

2D 바이너리 영역을 바탕으로 채광 영역과 장애물 영역을 구분하며, 채광 영역과 장애물 영역은 간단한 형상을 가지도록 설정한다. 변환된 영역의 최소 단위는 픽셀(Pixel)이라 정의하고, 셀 분할 기법의 결과로 분할된 영역을 셀(Cell)이라 정의한다.

변환된 2D 바이너리 영역을 H. Choset 등이 [3]에서 제안한 방법인 모스 셀 분할법(Morse cell decomposition method)을 사용하여 셀로 분할한다. 주어진 영역을 수직방향 및 수평방향으로 분할한 뒤, 셀 분할 최적화를 위해 최소 고도 계산 알고리즘(Minimal sum of altitude algorithm, MSA)을 적용한다. 이후에 커버리지 기초 경로를 생성하며 이렇게 생성된 커버리지 기초 경로는 각 셀의 커버리지를 보장한다.

분할된 영역에서 각 셀간의 인접한 관계를 나타낸 신장트리(Spanning tree)를 생성하고, 이 신장트리에 기초하여 너비우선 탐색 알고리즘(Breadth first search algorithm, BFS) 및 깊이우선 탐색 알고리즘(Depth first search algorithm, DFS)을 적용하여 셀 커버리지 순서를 결정한다. 셀 간의 이동경로는 가시성 그래프 알고리즘(Visibility graph algorithm)[13]을 적용하여 생성되는데, 이전 셀의 커버리지 완료점으로부터 다음 셀 커버리지 경로의 시작점까지 이동 시 최단경로를 생성하여 셀과 셀 간의 이동경로를 최소화한다. 셀 커버리지 순서와 커버리지 기초경로의 방향을 바탕으로 이동경로의 거리를 가장 짧게 만드는 셀 커버리지 순서와 커버리지 기초경로의 방향을 최종 그래프 경로와 커버리지 기초경로의 방향으로 결정한다. 이 때 셀 간의 이동경로가 이전 셀의 커버리지 기초 경로보다 길게 생성되는 경우 수정 스위핑 라인법 (Modified sweeping line strategy)을 이용하여 셀 간의 이동경로를 재생성한다.

커버리지 기초 경로를 통해 각 셀의 커버리지 경로를 보장하고, 각 셀의 커버리지 순서를 결정하며, 셀 간의 이동경로를 최소화함으로써 주

어진 전체 영역에 대한 경로 생성이 완료된다. 전체 커버리지 경로는 주어진 영역의 완전 커버리지를 지향하며 이동경로의 거리를 최소화하는 것이 최적의 전체 커버리지 경로계획이다.

전체 커버리지 경로계획의 과정은 다음과 같다.

1. 전처리 과정으로 주어진 3D 영역을 2D 바이너리 영역으로 변환한다.
2. 모스 셀 분할법을 이용하여 변환된 2D 바이너리 영역을 분할하고 최소 고도 계산 알고리즘을 이용하여 셀 분할 최적화 결과를 도출한 후 커버리지 기초 경로를 생성한다.
3. 분할된 영역을 바탕으로 신장트리를 생성하고, 너비우선 탐색 알고리즘 및 깊이우선 탐색 알고리즘을 적용하여 셀 커버리지 순서를 결정한다.
4. 가시성 그래프 알고리즘을 바탕으로 셀 간의 이동경로를 생성한다.
5. 수정 스위핑 라인법을 이용하여 과도하게 중복된 셀 간의 이동경로를 수정한다.
6. 커버리지 기초 경로와 셀 커버리지 순서, 셀 간의 이동경로를 결합하여 전체 커버리지 경로를 계획한다.
7. 각 커버리지 경로의 회전 경로를 부드럽게 함으로써 연속적인 전체 커버리지 경로를 생성한다.

2.2 전처리

본 절에서는 전처리로 주어진 가상의 3D 영역을 바이너리 영역으로 변환한다. 가상의 3D 영역은 Fig. 2.1과 같은 수심도 $M(x,y)$ 로 나타난다. 수심도의 각 점 (x,y) 을 통하여 깊이를 알 수 있으며, 식 (2.1)을 통해 각 점의 경사를 계산한다. 식 (2.1)의 $s(x,y)$ 는 경사 분포도를 의미하며, $\vec{i}, \vec{j}$ 는 x,y 축에서 표준단위벡터이다.

$$s(x,y) = \|\nabla M\| = \left\| \frac{\partial M}{\partial x} \vec{i} + \frac{\partial M}{\partial y} \vec{j} \right\| \quad (2.1)$$

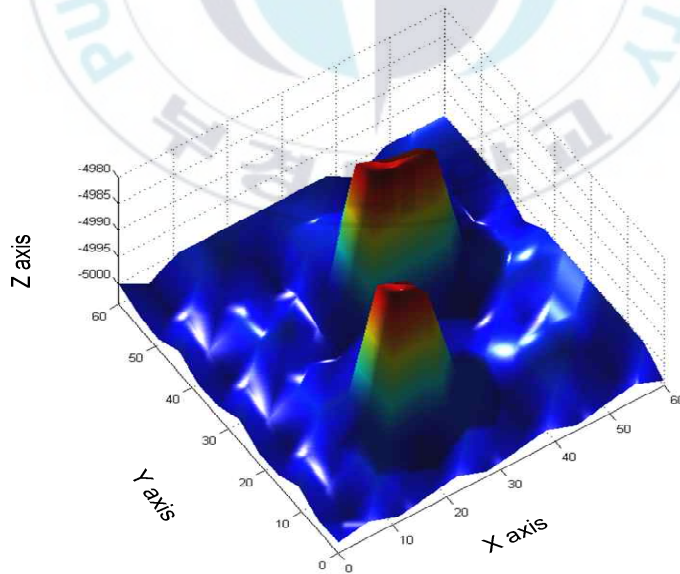


Fig. 2.1 Example of virtual 3D map

식 (2.1)을 통해 경사 분포도를 얻은 후 식 (2.2)와 같이 경사 문턱값 δ_s 를 이용하여 바이너리 영역을 생성한다. 식 (2.2)를 통해 행렬 $B(x,y)$ 에서 1의 값을 가지는 요소들은 장애물 영역으로 정의되며, 0의 값을 가지는 요소들은 채광 영역(자유 영역)으로 정의된다. Fig. 2.2는 Fig. 2.1에 나타낸 가상의 3D 영역을 바이너리 영역으로 변환하였을 때 생성된 맵 (map) 결과이다.

$$B(x,y) = \begin{cases} 1 & \text{if } s(x,y) \geq \delta_s \\ 0 & \text{if } s(x,y) < \delta_s \end{cases} \quad (2.2)$$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 2.2 Example of binary map

2.3 셀 분할 및 회전 횟수 최소화

본 절에서는 Fig. 2.2에 나타낸 바이너리 영역을 바탕으로 모스 셀 분할법을 이용하여 주어진 영역을 셀로 분할하는 방법을 나타내고, 최소 고도 계산 알고리즘을 이용하여 셀 분할 최적화 결과를 도출한다.

2.3.1 모스 셀 분할법

셀 분할은 수평방향 및 수직방향으로 수행되고, 수평 및 수직방향의 셀 분할 결과를 비교·결합하여 최종적으로 셀 분할 결과를 도출한다. 주어진 영역에서 최소 단위를 픽셀이라 정의하며, 셀 분할 기법을 통하여 분할된 영역을 셀이라 정의한다. 모스 셀 분할법은 다음과 같은 단계로 진행된다.

1. 변환된 2D 바이너리 영역에 대하여 행 또는 열 단위로 자유 영역과 장애물 영역을 구분한다.
2. 자유 영역과 장애물 영역을 확실히 구별할 수 있도록 각각에 식별자를 할당한다.
3. 식별자를 할당한 후 자유영역 및 장애물 영역에 대한 식별자 병합을 실시한다.
4. 불필요한 셀 및 갇힌 셀에 대하여 식별자 병합을 실시한다.

Fig. 2.3과 같이 주어진 공간에서 Sweep line을 따라 수평방향 혹은 수직방향으로 이동하면서 자유 영역(Free space segments)을 추출하고, 추출된 자유 영역에 식별자(Identifier, ID)를 할당한다. 여기서 자유 영역은 채광 영역과 같다.

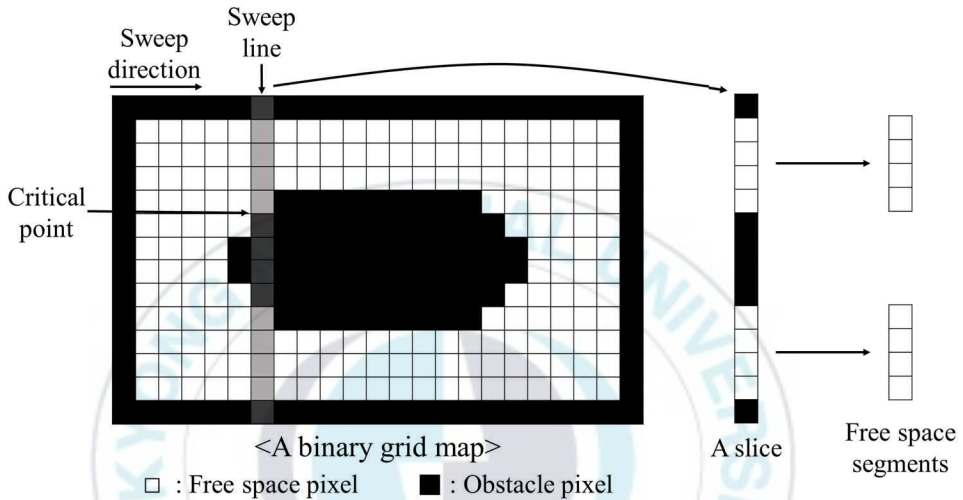


Fig. 2.3 Free space segments extraction of Morse cell decomposition

Sweep line으로 분할된 영역을 슬라이스(Slice)라 하며, Sweep line과 장애물 영역 간의 접점을 임계점(Critical point)이라 한다. 이 임계점에서 위, 아래로 선을 연장하여 경계를 만든다. 이렇게 만들어진 경계의 앞뒤에서 연결성이 바뀐다. 이렇게 주어진 영역을 나누는 방법을 정확한 셀 분할(Exact cell decomposition) 기법이라 한다.

모스 셀 분할법에 대해 자세한 설명을 위해 수평방향으로 셀을 분할할 때의 과정을 소개한다. 1단계는 바이너리 영역에 각 셀로 분할하기 위해 각 픽셀에 식별자를 할당하는 것이다. 수평방향으로 셀을 분할할 경우 주어진 3D 영역을 바이너리 영역으로 변환하여 자유 영역과 장애물

영역을 추출한 후 Fig. 2.4와 같이 영역의 좌측 첫 열에서부터 시작하여 우측 마지막 열까지 각각의 픽셀에 대하여 열 단위로 식별자(Identifier, ID)를 할당한다(수직방향 셀 분할의 경우에는 최상단의 행에서부터 최하단의 행까지 행 단위로 식별자를 할당한다).

하나의 열을 살펴보면 해당하는 열의 상단에서 하단까지 각 픽셀을 따라 식별자를 할당한다. 채광 영역에 대한 식별자는 숫자 1부터 시작되며, 장애물 영역에 대한 식별자는 1001부터 시작된다. 채광 영역의 식별자는 장애물 영역의 픽셀을 만나거나 열이 바뀔 때 마다 그 숫자가 1씩 증가하게 되며, 장애물 영역의 식별자는 다음 픽셀이 채광 영역이거나 열이 바뀔 때 마다 그 숫자가 1씩 증가한다. Fig. 2.4에서 같은 식별자를 가진 픽셀 영역을 셀이라 하며 셀로 표시된 그림이 Fig. 2.5이다.

1	2	4	5	8	10	13	15
1	2	4	1002	1004	1005	1007	15
1	1001	4	6	1004	11	1007	15
1	3	4	1003	1004	1006	1007	15
1	3	4	7	9	12	14	15

Fig. 2.4 Vertical slice and ID assignment

1	2	4	5	8	10	13	15
	1001		1002	1004	1005	1007	
			6		11		
			1003		1006		
			7		9		

Fig. 2.5 Result expressing Fig. 2.4 into cells

2단계는 각 픽셀에 식별자 할당이 완료된 이후에 장애물 영역 및 채광 영역의 식별자를 병합하는 것이다. 장애물 영역의 식별자 병합을 살펴보면, 하나의 장애물 영역 픽셀에 대해 상·하·좌·우 1개의 픽셀에 할당된 식별자를 확인한다. 만약 채광 영역이면 식별자는 변하지 않고, 장애물 영역이면 각각의 식별자를 비교하여 작은 번호를 갖는 식별자로 병합한다. Fig. 2.6은 장애물 영역의 식별자 병합을 실행한 결과이다.

1	2	4	5	8	10	13	15
1	2	4	1002	1002	1002	1002	15
1	1001	4	6	1002	11	1002	15
1	3	4	1002	1002	1002	1002	15
1	3	4	7	9	12	14	15

Fig. 2.6 Grouping of obstacle area's ID

3단계는 장애물 영역의 식별자 병합이 완료 후 채광 영역에 대한 식별자 병합하는 것이다. 수평방향으로 셀 분할을 실행할 때 각 셀은 최상단 및 최하단의 식별자를 확인하여 식별자 병합을 실행한다. 주어진 영역의 바깥쪽(경계 외부)은 Fig. 2.7과 같이 무한대(NaN)로 처리된다. 이 때 식별자 5가 부여된 셀의 양 끝단은 무한대와 1002로 이루어지고, 식별자 8이 부여된 셀의 양 끝단 또한 무한대와 1002로 이루어지므로 두 셀은 가장 작은 식별자인 5로 병합된다. 같은 방식으로 식별자 10, 13이 부여된 셀 또한 식별자 5로 병합된다. 식별자 13과 15를 가진 셀의 경우 각 셀의 양 끝단의 값이 다르므로 식별자 15는 병합을 실행하지 않는다.

NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2	4	5	8	10	13	15
1	2	4	1002	1002	1002	1002	15
1	1001	4	6	1002	11	1002	15
1	3	4	1002	1002	1002	1002	15
1	3	4	7	9	12	14	15
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Fig. 2.7 Values of outside of given area

이러한 과정은 Fig. 2.7에서 식별자 7과 9, 12와 14를 가진 셀에 대해서도 적용되며 이러한 과정을 반복함으로써 채광 영역에 대한 식별자 병합이 완료된다. 채광 영역의 식별자 병합이 완료된 후 전체적으로 각 열의 식별자를 순차적으로 재배치함으로써 Fig. 2.8과 같은 결과를 도출한다.

1	2	4	5	5	5	5	9
1	2	4	1002	1002	1002	1002	9
1	1001	4	6	1002	8	1002	9
1	3	4	1002	1002	1002	1002	9
1	3	4	7	7	7	7	9

Fig. 2.8 Grouping of mining area's ID

4단계는 채광 영역의 식별자를 병합한 이후에 불필요한 셀(Noisy cell)을 병합하는 것이다. 주어진 영역이 매우 복잡한 형태를 지닐 경우 Fig. 2.8에서 식별자 6과 같은 불필요한 셀들이 많이 생겨나게 되며, 이러한 불필요한 셀들이 많이 존재하게 되면 커버리지 경로 또는 셀 커버리지 순서를 생성할 경우 경로가 복잡해지고 시간적으로 매우 비효율적이게 되므로 이러한 불필요한 셀들의 병합 또는 제거가 필요하다. 만약 하나의 픽셀에서 상·하·좌·우 1개의 픽셀들을 확인했을 때 Fig. 2.8과 같이 3개의 장애물 영역으로 둘러싸여 있다면 불필요한 셀로 정의한다. 불필요한 셀은 주위의 채광 영역 셀의 식별자를 확인하여 Fig. 2.9와 같이 식별자를 병합한다. 이러한 과정은 불필요한 셀들이 모두 처리될 때까지 진행되며, 불필요한 셀들에 대한 처리가 끝날 경우 전체적으로 셀의 식별자를 순차적으로 정리한다.

1	2	4	5	5	5	5	8
1	2	4	1002	1002	1002	1002	8
1	1001	4	4	1002	7	1002	8
1	3	4	1002	1002	1002	1002	8
1	3	4	6	6	6	6	8

Fig. 2.9 Grouping of noisy cell

마지막 단계는 갇힌 셀(Trapped cell)을 장애물로 간주하는 과정을 수행하는 것이다. Fig. 2.9에서 식별자 7의 셀은 장애물 영역에 둘러싸여 커버리지 불가능한 셀이다. 이러한 셀들은 커버할 수가 없기 때문에 장애물 영역으로 정의하며, Fig. 2.10과 같은 결과를 얻게 된다. 갇힌 셀에 대한 처리가 끝난 이후에 전체적으로 식별자를 순차적으로 정리한다. 갇힌 셀에 대한 처리가 끝났다면 셀 분할에 대한 모든 과정을 종료하며, Fig. 2.10이 모스 셀 분할법에 의한 수평방향 셀 분할 결과가 된다.

1	2	4	5	5	5	5	7
1	2	4	1002	1002	1002	1002	7
1	1001	4	4	1002	1002	1002	7
1	3	4	1002	1002	1002	1002	7
1	3	4	6	6	6	6	7

Fig. 2.10 Processing of trapped cell

수직방향 셀 분할을 실행할 때의 과정은 수평방향 셀 분할의 과정과 다르게 행 단위로 셀을 분할하며, 나머지 과정은 같다. 수직방향 셀 분할의 결과는 Fig. 2.11과 같다.

1	1	1	1	1	1	1	1
2	2	2	1001	1001	1001	1001	3
4	1002	5	5	1001	1001	1001	3
6	6	6	1001	1001	1001	1001	3
7	7	7	7	7	7	7	7

Fig. 2.11 Vertical cell decomposition result

2.3.2 셀 분할 최적화

수평방향 셀 분할 결과 및 수직방향 셀 분할 결과가 나타나면 두 결과를 비교하여 최적의 셀 분할 결과를 도출한다. 최적의 셀 분할 결과를 도출하기 위해 최소 고도 계산 알고리즘(Minimal sum of altitude algorithm, MSA)을 사용한다. 주어진 영역의 커버를 위해 요구되는 회전 횟수를 최소화한다면, 전체 커버리지 경로에서 시간을 절약할 수 있다. 여기서 고도(Altitude)는 하나의 셀을 회전시킬 때 최소 길이를 의미하며, 즉 셀의 높이(Height) 및 너비(Width)를 비교했을 때 보다 작은 값을 가지는 쪽의 길이를 의미한다.

템플릿 기반 경로계획(Template-based path planning)의 경우 로봇은 모든 셀에 대하여 같은 커버리지 방향을 가지기 때문에 Fig. 2.12(a)와 같이 불필요한 회전이 발생하여 커버리지 효율이 매우 좋지 않을 수 있다. 하지만 커버리지 방향이 Fig. 2.12(b)와 같이 셀의 최단측 방향에 대하여 수직 방향인 b 로 설정된다면 최소한의 회전을 통해 해당 영역의 커버리지를 완료할 수 있다.

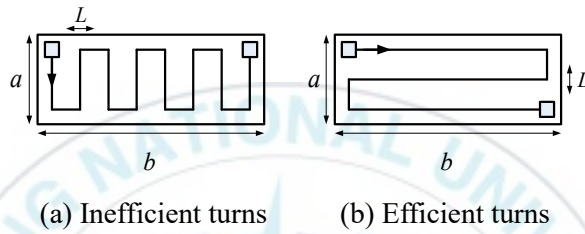


Fig. 2.12 Relation between turning number and coverage direction

Fig. 2.12(b)에 나타난 것처럼, 회전 횟수는 셀의 길이(Altitude of cell)에 비례한다. 그러므로 회전 횟수를 최소화하기 위해서 셀의 길이에 따른 커버리지 방향은 동기화되어야 한다. 사각형 셀에서 i^{th} 셀의 길이 $A(i)$ 는 다음과 같이 계산된다.

$$A(i) = \min(a_i, b_i), \quad \text{for } i = 1, 2, 3, \dots \quad (2.3)$$

여기서 a_i 및 b_i 는 각각 i^{th} 셀의 높이(Height)와 너비(Width)이다.

i^{th} 셀의 커버리지 방향 $Dir(i)$ 는 다음의 식으로 얻을 수 있다.

$$Dir(i) = \begin{cases} \text{Vertical} & \text{if } a_i > b_i \\ \text{Horizontal} & \text{if } a_i < b_i \end{cases} \quad (2.4)$$

j^{th} 셀 배열 방식에서 최소 길이의 합 $S(j)$ 는 다음과 같이 정의된다.

$$S(j) = \sum_{i=1}^n A(i) \quad \text{for } i, j = 1, 2, 3, \dots \quad (2.5)$$

여기서 n 은 셀의 총 개수이고, j 는 분할된 셀 간의 비교 과정에서 생성되는 셀 배열 방식의 개수이다.

MSA 알고리즘의 최종 목표는 최소 고도의 합인 $S(j)$ 의 최소값을 찾는 것이다. 따라서 MSA는 다음과 같이 정의된다.

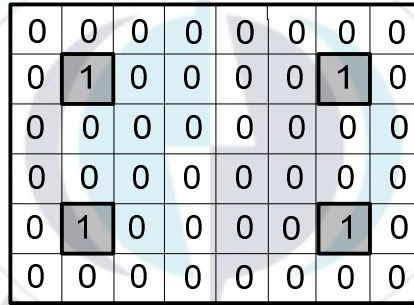
$$MSA = \min(S(1), S(2), S(3), \dots, S(k)) \quad (2.6)$$

여기서 k 는 가능한 셀 배열 방식의 총 개수이다.

마지막으로 주어진 영역을 커버하기 위한 총 회전 횟수 N 을 계산한다. 총 회전 횟수는 로봇의 작용기 너비 w 를 고려하여 다음과 같이 표현된다.

$$N = \frac{1}{w} MSA, \text{ for } i=1,2,3,\dots \quad (2.7)$$

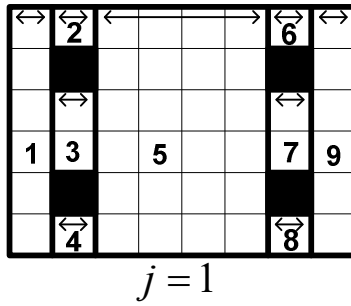
Fig. 2.13은 MSA 알고리즘의 예를 나타내기 위해 생성한 바이너리 영역이다.



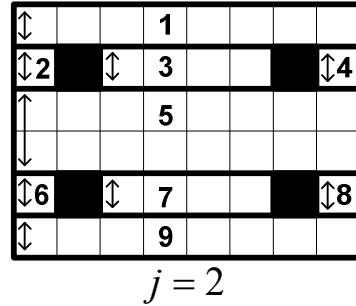
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0

Fig. 2.13 Example of binary map

Fig. 2.13의 바이너리 영역을 모스 셀 분할법을 이용하여 수평방향 및 수직방향으로 분할한 결과는 Fig. 2.14와 같다. 이 때 수직방향 및 수평방향 셀 분할 결과에서 $S(j)$ 를 계산했을 때 그 값이 더 작은 셀 분할 결과를 기본 맵으로 한다. 여기서 수직방향 셀 분할 결과 $S(j)$ 는 12, 수평방향 셀 분할 결과 $S(j)$ 는 10이므로 수평방향 셀 분할 결과를 기본 맵으로 한다. 사각형 내부의 숫자는 셀 번호를 의미한다.



(a) Horizontal



(b) Vertical

Fig. 2.14 Horizontal and vertical cell decomposition

기본 맵이 결정되면 최적의 셀 배열을 찾기 위해 두 셀 분할 결과를 결합하게 된다. 두 셀 분할 결과의 결합이란 Fig. 2.15와 같이 수평방향 셀 분할 결과에 수직방향 셀 분할 결과의 셀 1을 삽입하는 것을 의미한다. 이러한 과정은 수직방향 셀 분할 결과의 셀 개수만큼 진행되며, 이 과정에서 $S(j)$ 가 가장 작은 결과를 최적의 셀 분할 결과로 저장한다.

각 단계마다 전체 셀의 개수 n 및 최소 길이 합 $S(j)$ 가 계산되며, Table 2.1에서 확인할 수 있다. 셀 배열 방식 $j=11$ 일 때 셀 개수 및 최소 고도의 합이 가장 작은 값을 나타내므로 이 결과가 최적 셀 분할 결과가 된다.

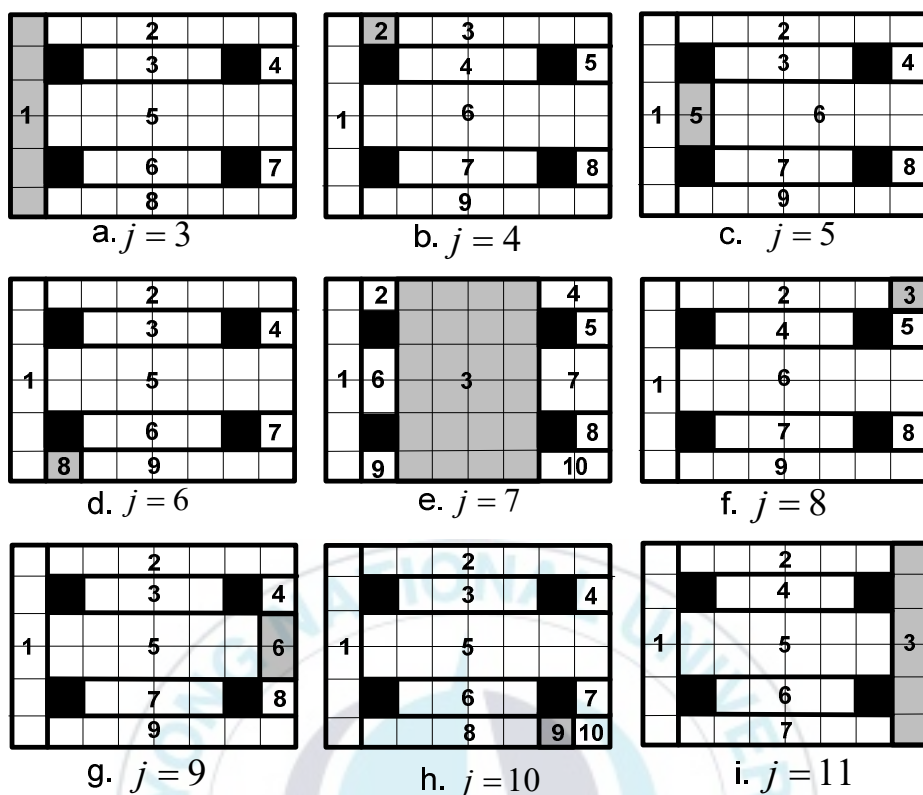


Fig. 2.15 Combination of vertical and horizontal cell decomposition

Table 2.1 Number of cell and the sum of altitude of each step obtained from the proposed combined method

Way to split the graph j	Total number of cell n	Sum of altitude $S(j)$
$j=1$	9	12
$j=2$	8	10
$j=3$	8	9
$j=4$	9	10
$j=5$	9	10
$j=6$	9	10

$j=7$	10	14
$j=8$	9	10
$j=9$	9	10
$j=10$	10	11
$j=11$	7	8



2.4 커버리지 순서 결정

본 절에서는 앞 절의 최적 셀 분할 결과를 바탕으로 셀 커버리지 순서(Cell coverage sequence)를 결정하기 위한 알고리즘을 제시한다. 셀 커버리지 순서는 두 가지 이론을 바탕으로 생성되며 첫 번째는 분할된 영역에서 셀 간의 인접한 관계를 나타내는 신장트리이고, 두 번째는 트리나 그래프를 탐색하는 그래프 탐색 알고리즘이다. 본 연구에서는 그래프 탐색 알고리즘으로 너비우선 탐색 알고리즘(Breadth-first search algorithm, BFS)과 깊이우선 탐색 알고리즘(Depth-first search algorithm, DFS)을 제시한다.

신장트리는 일찍이 어떠한 객체와 그들 간의 연결성으로 표현되는 모든 문제들을 해결하기 위해 가장 많이 사용되는 방법이다. 객체들의 집합을 노드(Node)로 나타내고, 객체들 간의 연결성들의 집합을 엣지(Edge)로 나타내어 문제를 한 눈에 파악할 수 있다. 셀로 분할된 영역에서 두 개의 셀이 경계선을 공유할 때 두 셀은 인접해 있다고 하며, 인접한 셀 간의 관계를 통하여 신장트리를 생성한다. 두 셀이 인접한 관계를 이루고 있다면 두 셀은 엣지로 연결된다.

Fig. 2.16은 일반적인 신장트리를 나타낸 것이다. 노드 A와 같이 뿌리(Root)를 가지며, 뿌리로부터 그래프가 구성된다. 노드 B와 C는 노드 A를 부모(Parent)로 가지는 형제(Sibling)관계이다. 뿌리에서 가장 멀리 떨어져지며, 더 이상 자식(Child)을 가지지 않는 노드 G를 종료노드(Terminal node) 또는 잎(Leaf)이라 한다. 뿌리와 종료노드가 아닌 노드들은 비종료노드(Nonterminal node)또는 내부노드(Internal node)라 한다.

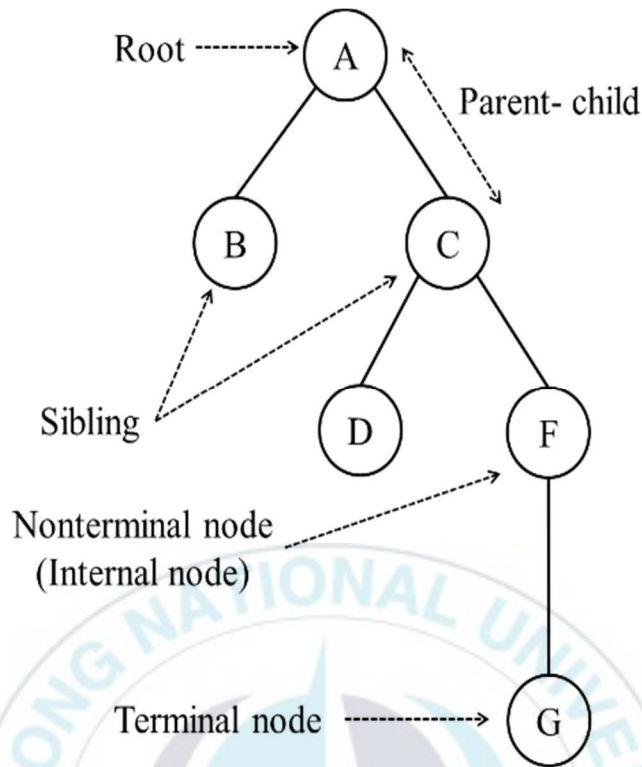
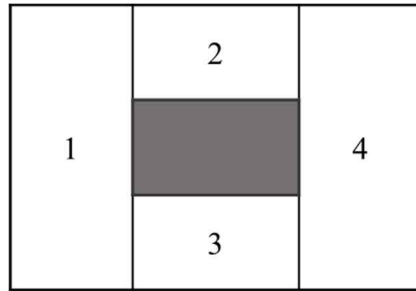
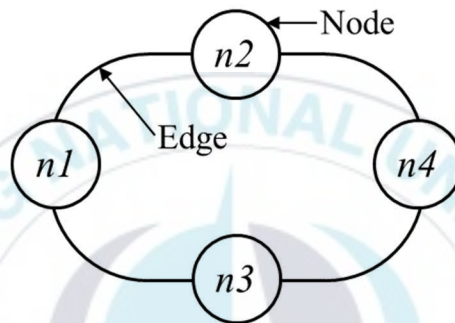


Fig. 2.16 General spanning tree

Fig. 2.17은 영역을 분할된 결과와 그에 따른 신장트리를 나타낸다. Fig. 2.17(a)는 4개의 셀로 분할된 주어진 영역을 의미하고, 분할된 영역을 신장트리로 나타내었을 때의 결과는 Fig. 2.17(b)와 같다. Fig. 2.17(b)에서 n1은 노드 1을 나타내며, 노드 번호는 셀 번호와 같다.



(a) Cell decomposition result



(b) Spanning tree

Fig. 2.17 Cell decomposition and spanning tree

Fig. 2.17(b)에 나타난 신장트리의 노드와 엣지를 방문한다는 의미는 아래와 같다.

- 1) 노드 방문 : 해당되는 셀을 커버한다는 의미이며, 커버리지 경로와 관련이 있음
- 2) 엣지 방문 : 하나의 노드에서 인접하게 연결된 다른 노드로 이동한다는 의미이며, 이동경로와 관련이 있음

신장트리로 생성된 모든 노드를 방문하기 위하여 Fig. 2.18과 같은 그래프 탐색 알고리즘을 이용하며, 그래프 알고리즘에서 핵심적 위치를 차지하는 알고리즘으로는 Fig. 2.18(a)와 같은 너비우선 탐색 알고리즘과 Fig. 2.18(b)와 같은 깊이우선 탐색 알고리즘이 있다. Fig. 2.18에 나타난 노드 번호는 각각의 알고리즘을 이용하여 신장트리를 탐색할 때의 순서를 나타낸 것이며, 화살표는 각각의 알고리즘이 탐색을 진행하는 방식을 나타낸 것이다

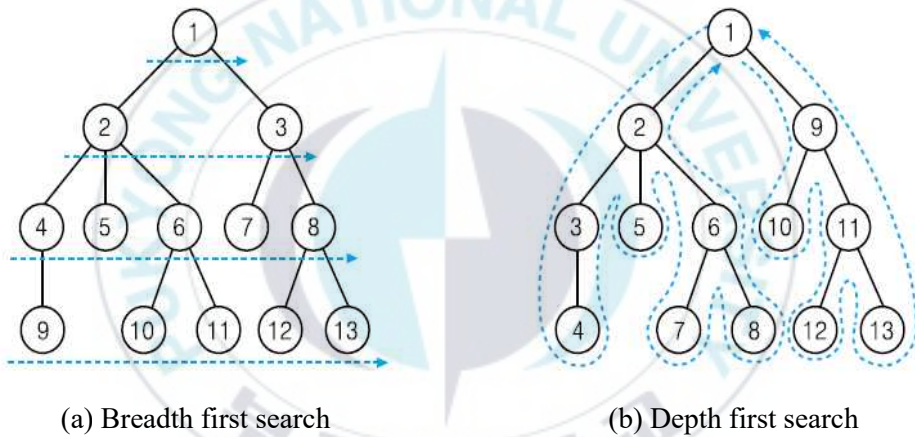


Fig. 2.18 Comparison between BFS and DFS

Fig. 2.18(a)와 같이 너비우선 탐색은 루트(노드 1)를 시작으로 탐색을 한다면 먼저 루트의 자식을 차례로 방문한다. 다음으로 루트 자식의 자식, 즉 루트에서 두 개의 엣지를 거쳐서 도달할 수 있는 노드를 방문한다. 다음으로 루트로부터 세 개의 엣지를 거쳐서 도달하는 노드의 순서로 루트에서의 거리가 가까운 순서대로 방문한다. 즉 너비우선 탐색 알고리즘은 옆으로(너비로) 훑어나가는 형태를 띈다.

Fig. 2.19은 무방향 그래프에 대해 너비우선 탐색 알고리즘을 수행하는 예이다. 그림에서 검은색으로 굵게 표시한 엣지들은 각 노드를 방문할 때 사용한 엣지들이다. 너비우선 탐색 알고리즘을 통해 생성되는 커버리지 순서는 다음과 같으며, Fig. 2.19에서 노드 번호로 나타내었다.

- (a): 시작 노드로 정해진 노드를 방문한다(1로 표시).
- (b): (a)에서 방문한 노드 1에 인접한 노드를 모두 방문한다(2, 3, 4).
- (c): 노드 2는 인접한 노드가 없으므로 방문할 수 있는 노드가 없다. 노드 3에 인접한 노드 중 방문하지 않은 노드는 모두 방문하며 1개의 노드만이 존재한다(5로 표시). 노드 4에 인접한 노드 중 방문하지 않은 노드를 모두 방문하며 2개의 노드가 존재한다(각각 6, 7로 표시).
- (d): 노드 5에 인접한 노드 중 방문하지 않은 노드는 없다. 노드 6에 인접한 노드 중 방문하지 않은 노드를 모두 방문한다. 1개의 노드만이 존재한다(8로 표시). 노드 7에 인접한 노드 중 방문하지 않은 노드는 없다.
- (e): 마지막으로 노드 8에 인접한 노드 중 방문하지 않은 노드가 없어 더 이상 갈 곳이 없으므로 끝낸다.
- (f): 각 노드를 방문할 때 사용되지 않은 엣지들을 제거함으로써 너비우선 탐색 알고리즘의 결과를 나타낸다.

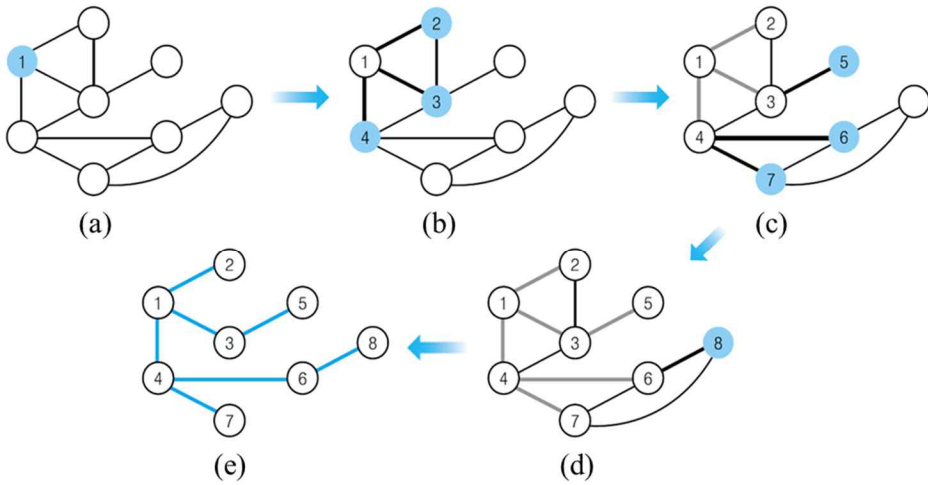


Fig. 2.19 Example of breadth first search algorithm

Fig. 2.18(b)와 같이 깊이우선 탐색은 루트의 자식 노드를 하나 방문한 다음, 아래로 내려갈 수 있는 곳까지 내려간다. 더 이상 내려갈 수가 없으면 위로 되돌아오다가 내려갈 곳이 있으면 즉각 내려간다. 즉 깊이우선 탐색 알고리즘은 아래로(깊이로) 갈 수 있는 데까지 갔다가 막히면 되돌아와서 다시 내려가는 형태를 띤다.

깊이우선 탐색 알고리즘에 대하여 자세히 알아보면 다음과 같다. Fig. 2.20은 무방향 그래프에 대해 깊이우선 탐색 알고리즘을 수행하는 예이다. 깊이우선 탐색 알고리즘을 통해 생성되는 커버리지 순서는 다음과 같으며, Fig. 2.20에서 노드 번호로 나타내었다.

- (a): 시작 노드로 정해진 노드를 방문한다(노드 1).
- (b): 노드 1에 인접한 노드 중 하나를 방문한다(노드 2).
- (c): 노드 2에서 인접하면서 방문하지 않은 정점은 총 세 개다. 이 중 하나를 방문한다(노드 3)
- (d): 노드 3에서 인접한 노드 중 방문하지 않은 노드는 두 개다. 이 중 하나를 방문한다(노드 4).
- (e): 노드 4에 인접한 노드 중 방문하지 않은 노드는 하나뿐이다. 이를 방문한다(노드 5). 5에 인접한 노드 중 방문하지 않은 노드는 없다. 따라서 왔던 길로 되돌아간다. 노드 4, 3, 2로 되돌아가는 과정에 노드 4, 3에 인접한 노드 중 방문하지 않은 노드는 없다.
- (f): 노드 2로 되돌아오면 이에 인접한 노드 중 방문하지 않은 노드가 하나 있다. 이를 방문한다(노드 6).
- (g): 노드 6에 인접한 노드 중 방문하지 않은 노드는 두 개이다. 이 중 하나를 방문한다(노드 7).
- (h): 노드 7에 인접한 노드 중 방문하지 않은 노드는 없다. 노드 6으로 돌아간다. 노드 6에 인접한 노드 중 방문하지 않은 노드가 하나 있으며 이를 방문한다(노드 8). 노드 8에서 인접한 노드 중 방문하지 않은 노드는 없다. 노드 6으로 돌아간다. 노드 6에서도 인접한 노드 중 방문하지 않은 노드는 없으므로 노드 2로 돌아간다. 노드 2에서도 방문하지 않은 인접 노드가 없기 때문에 노드 1로 돌아간다. 노드 1에서도 방문하지 않은 노드가 없다. 따라서 시작 노드(노드 1)에서 더 이상 갈 곳이 없으므로 끝낸다.

(i): 각 노드를 방문할 때 사용되지 않은 엣지들을 제거함으로써 깊이 우선 탐색 알고리즘의 결과를 나타낸다.

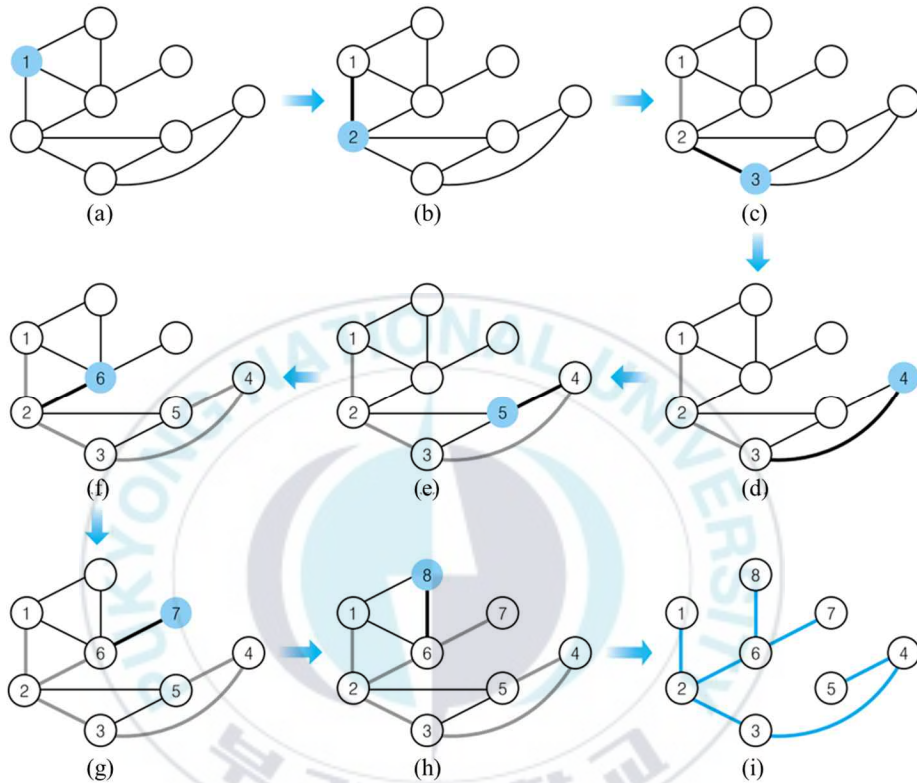


Fig. 2.20 Example of breadth first search algorithm

2.5 셀 간의 이동경로 생성

본 절에서는 셀과 셀 간의 이동경로를 생성하기 위한 가시성 그래프 알고리즘을 소개한다. 셀과 셀 간의 이동경로를 생성하는 것은 시작점에서 목표점까지의 최적경로를 생성하는 알고리즘과 유사하다. 격자 공간에서 시작점과 장애물, 목표점이 있을 때 이동경로를 생성하는 방법에는 A^* 또는 D^* 알고리즘 등이 있지만[25], 격자 공간이 커질수록 분석 시간이 길어진다는 단점이 있다.

가시성 그래프 알고리즘 2차원 공간 상에서 여러 장애물이 존재하고 시작점과 도착점을 잇는 느슨한 줄이 있다고 할 때, 이 줄을 잡아 당겨서 팽팽해지도록 하면 결국에는 줄이 장애물을 걸치게 된다. 더 이상 줄을 팽팽하게 당길 수 없을 때 이 줄이 시작점과 도착점을 잇는 최단 경로가 된다. 이런 기본 생각을 바탕으로 나온 알고리즘이 가시성 그래프 알고리즘이다[26].

일반적으로 가시성 그래프 알고리즘을 얘기할 때, 알고리즘이 적용되는 환경은 다각형의 장애물로만 구성되어있다고 가정을 한다. 이런 가정 하에서는 모든 장애물을 점과 점들을 잇는 선으로 간단히 표현할 수 있기 때문이다.

앞서 설명한 줄을 팽팽하게 당기는 경우를 생각해보면, 시작점과 도착점을 잇는 최단 경로는 시작점과 도착점 사이에 장애물이 존재하는 한 반드시 장애물의 꼭지점을 지날 수밖에 없다. 거꾸로 말하면 모든 장애물의 꼭지점을 연결한 선들 가운데서 최단 경로를 찾아, 이 경로를 통해 시작점에서 꼭지점으로 이동하는 방법이 가시성 그래프 알고리즘이다.

Fig. 2.21은 가시성 그래프 알고리즘을 시작점과 완료점 간의 최단 경로를 생성한 예이다.

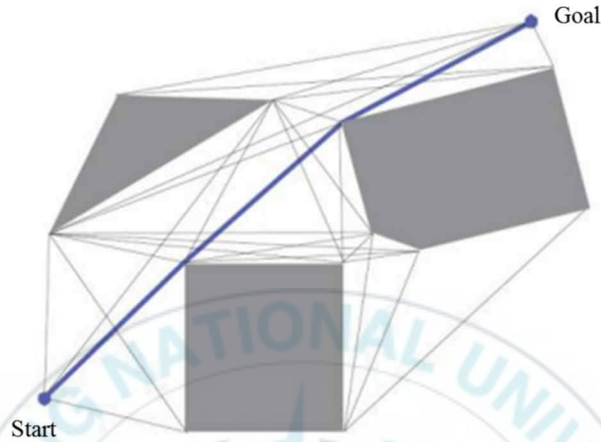


Fig. 2.21 Example of visibility graph algorithm

이 때 선들 가운데 최단거리를 찾는 방법은, 기존에 잘 알려진 그래프 찾기(Graph search) 알고리즘을 이용하면 된다. 가시성 그래프 알고리즘은 2차원 공간 상에서 장애물들이 다각형을 띠고 있다는 가정 하에서는 최단 경로를 찾아주는 알고리즘이다.

가시성 그래프 알고리즘을 이용한 셀 간의 이동경로 생성에 대하여 A^* 알고리즘을 이용한 셀 간의 이동경로와 비교함으로써 그 효율성을 증명하고자 한다. Fig. 2.22는 주어진 영역에서 시작점과 완료점 사이의 경로를 가시성 그래프 알고리즘 및 A^* 알고리즘을 이용하여 생성한 결과이다. 굵은 실선으로 표시된 것이 각 알고리즘을 이용하여 생성한 셀 간의 이동경로이다.

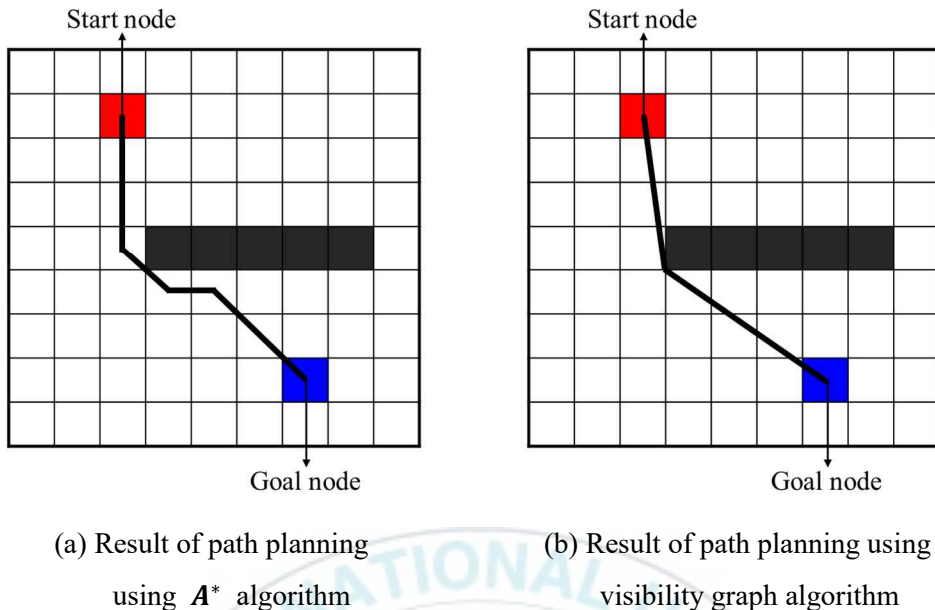


Fig. 2.22 Path comparison between A^* algorithm and visibility graph algorithm

Fig. 2.22에서 픽셀 하나의 길이가 $1m$ 라고 정의하여 이 때 각 알고리즘의 전체 이동거리와 계산 시간을 계산하면 Table 2.2와 같다. A^* 알고리즘을 이용하여 생성된 셀 간의 이동경로(a)를 살펴보면 직선 길이는 $4m$ 이고 대각선 길이 $3\sqrt{2}m(=4.243m)$ 이므로 총 이동경로는 $8.243m$ 가 된다. 가시성 그래프 알고리즘을 이용하여 셀 간의 이동경로(b)를 생성했을 때를 살펴보면 각각의 경로는 $\sqrt{(3.5^2 + 0.5^2)} = 3.536m$ 와 $\sqrt{(2.5^2 + 3.5^2)} = 4.301m$ 이므로 총 이동경로는 $7.837m$ 가 된다. 가시성 그래프 알고리즘을 이용하여 생성된 셀 간의 이동경로는 A^* 알고리즘을 이용한 셀 간의 이동경로보다 전체 이동거리 및 계산 시간에서 더욱 효율적임을 알 수 있다.

Table 2.2 Comparison between A^* algorithm and visibility graph algorithm

Algorithm	Total distance	Calculation time
A^*	8.243m	12s
Visibility graph	7.837m	7s

가시성 그래프 알고리즘을 이용하여 Fig. 2.22(b)와 같이 경로를 생성한다면 집광로봇이 장애물과 부딪히는 경우가 발생하기 때문에 로봇의 중심점을 이용한 설계를 위해 배위공간(Configuration space)를 고려할 필요가 있다. 배위공간은 로봇의 부피를 고려하여 장애물이나 경계에 로봇이 충돌하지 않는 로봇의 중심점에 대한 자유공간이다. Fig. 2.23과 같이 장애물과의 충돌 예방을 위해서 이동로봇의 크기를 고려한 배위공간 경계를 정의하여 가시성 그래프 알고리즘을 적용한다.

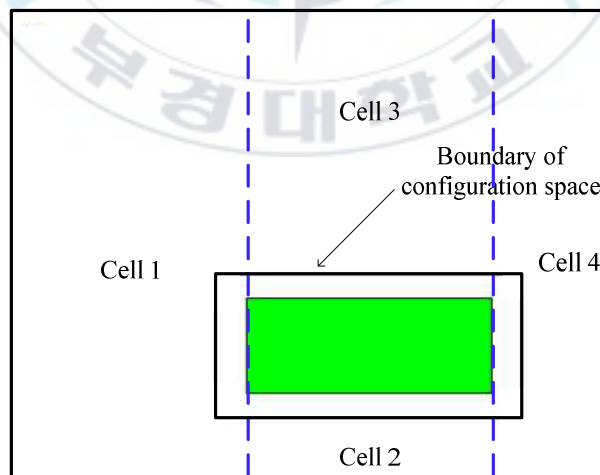


Fig. 2.23 Example of configuration space

2.6 중복된 경로의 최소화

본 절에서는 전체 커버리지 경로 생성 시 과도하게 중복된 경로가 생성될 경우 중복된 경로를 최소화 하기 위한 수정 스위핑 라인법을 소개한다.

커버리지 순서가 결정된 후 전체 커버리지 경로가 생성되는 데, 전체 커버리지 경로를 생성함에 있어서 셀 간의 이동경로는 임의로 생성된다. 이 때 셀 간의 이동경로가 전체 커버리지 경로와 중복을 야기할 가능성이 크기 때문에 수정 스위핑 라인법을 이용하여 전체 커버리지 경로의 중복을 최소화한다.

Fig. 2.24는 셀 간의 이동경로 생성 방법에 따른 커버리지 효율의 차이를 나타낸 것이다. Fig. 2.24(a)는 주어진 영역을 임계점에서 분할하여 점선으로 표시한 것이며, Fig. 2.24(b)는 로봇이 커버리지를 수행할 때 작용기(Effector) 너비를 나타낸 것이다. Fig. 2.24(c)와 같이 커버리지 경로를 생성할 경우 셀 간의 이동경로는 매우 비효율적으로 생성되며, 중복된 경로 및 이동 로봇의 급선회를 필요로 한다. 하지만 Fig. 2.24(d)과 같이 경로를 생성할 경우에 중복되는 경로를 제거할 수 있고, 커버리지를 수행함에 있어서 이동 로봇이 부드럽게 주행할 수 있다.

수정 스위핑 라인법은 Fig. 2.24(d)에 나타난 것처럼 입구점에서부터 시작된 주어진 환경의 일부분을 따라 커버하여 다음 셀의 가장 안쪽까지 이동하는 경로이다. 이후에 이전 셀의 나머지 부분을 커버하고, 입구점과 가까운 곳에서 나와 다음 셀의 커버리지를 수행한다.

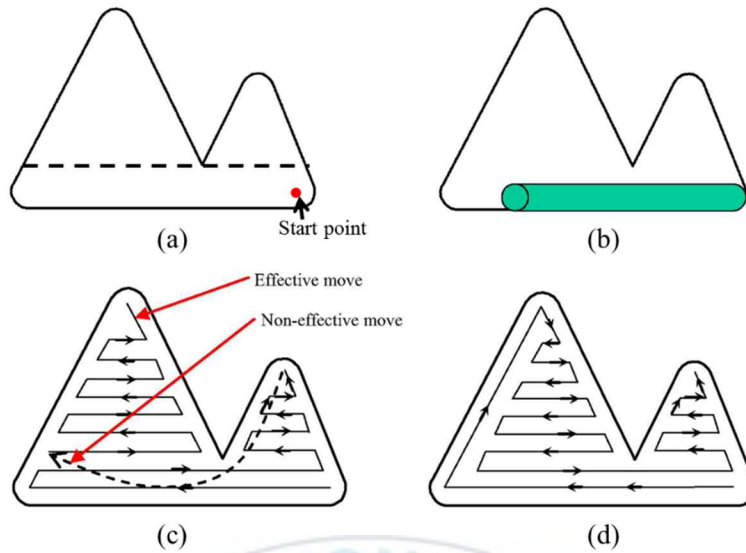


Fig. 2.24 Complete coverage problem using different method

앞서 Fig. 2.24에서 설명한 셀 간의 이동경로는 주어진 영역이 다각형인 경우 꼭지점에 있을 시 이를 바탕으로 셀 분할을 실행하고, 인접한 셀 간의 관계를 바탕으로 셀 간의 이동경로를 생성한다. 본 연구에서는 격자맵을 바탕으로 경로 계획을 실시하기 때문에 다음과 같은 조건을 제시하여 보다 적합하게 구성한다.

- (1) 셀 간의 이동경로가 이전 셀의 커버리지 경로의 길이 이상으로 중복되거나 과도하게 선회하는 경로를 생성할 경우 수정 스위핑 라인법을 적용하여 이동경로를 수정한다.
- (2) 수정 스위핑 라인법을 적용할 때 루트 노드가 최소 2개의 자식 노드를 가진다면 루트 노드는 스위핑 라인법을 적용하기 위하여 자식 노드와 인접한 관계를 이루어야 하며, 루트 노드는 완전히 커버하지 않은 채 최하위 자식 노드로 이동한다.

- (3) 자식 노드로 이동할 때 루트 노드의 바깥쪽 경계를 따라 자식 노드로 이동한다.
- (4) 최하위 자식 노드의 커버리지가 완료되면 루트 노드의 커버리지를 수행한다. 이 때 자식 노드의 커버리지 완료점은 루트 노드로부터 가까운 곳이 되게 한다.

Fig. 2.25는 수정 스위핑 라인법을 이용하여 셀 간의 이동경로를 수정하는 작업을 수행한 결과이다. 붉은 굵은 실선은 셀 간의 이동경로를 나타낸 것이다. Fig. 2.25(a)에서 셀 커버리지 순서가 1→2→3으로 결정된 경우 셀 간의 이동경로는 과도한 선회 경로를 생성한다. 따라서 수정 스위핑 라인법을 이용하여 Fig. 2.25(b)와 같이 셀 간의 이동경로를 수정하여 커버리지 경로를 생성할 경우 매우 효율적인 경로가 생성될 수 있다.

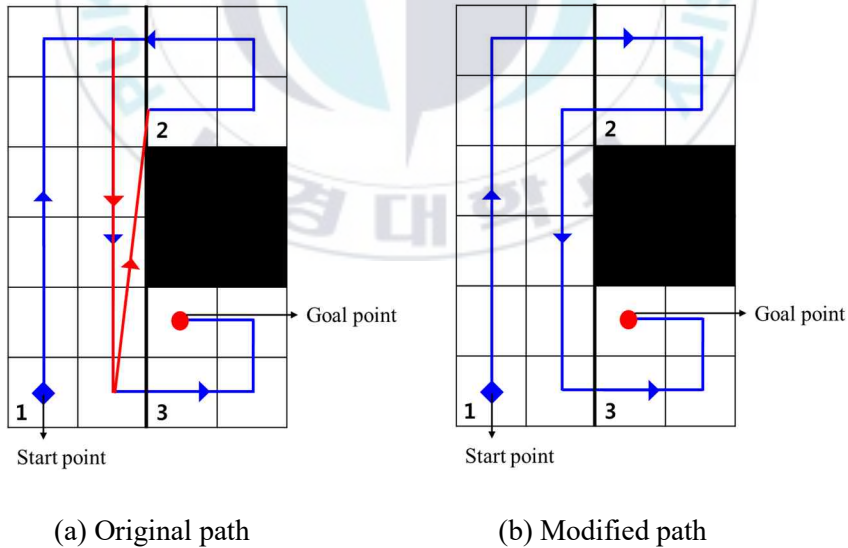


Fig. 2.25 Coverage path using modified sweeping line strategy

2.7 전체 커버리지 경로 생성

본 절에서는 전체 커버리지 경로 생성에 대해 다룬다. 전처리를 통해 얻어지는 2D 바이너리 영역에서 모스 셀 분할법 및 최소 길이 계산 알고리즘을 적용하여 주어진 영역을 셀로 분할 후 너비우선 탐색 알고리즘 또는 깊이우선 탐색 알고리즘을 이용하여 분할된 셀의 커버리지 순서를 결정한 이후에 가시성 그래프 알고리즘 및 수정 스위핑 라인법을 적용한 전체 커버리지 경로를 생성한다. 전체 커버리지 경로 생성 후 회전구간에 대한 처리가 끝나면 최종적인 전체 커버리지 경로가 생성된다.

셀의 커버리지 순서가 정해진 이후에 커버리지 기초 경로를 생성한다. 커버리지 기초 경로는 Fig. 2.26과 같은 템플릿 기반 경로로 이루어지며, 최소 고도 계산 알고리즘에 의해 각 셀의 높이와 너비에 따라 적은 비용이 생기는 패턴을 선택한다.

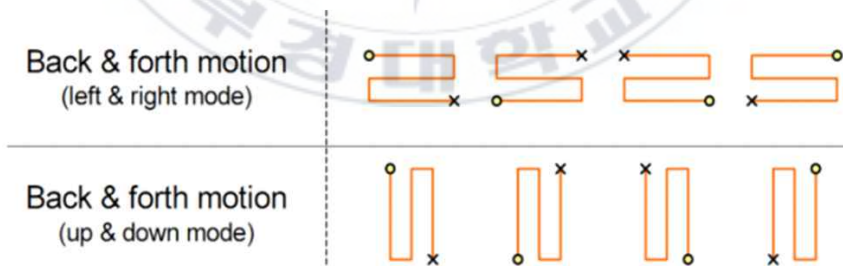


Fig. 2.26 Patterns of coverage path

예를 들어 커버리지 기초 경로는 Fig. 2.27의 실선과 같다. 각 셀의 꼭지점 중 하나가 시작점이 되고, 커버리지 시작점이 정의되면 커버리지 경로 간의 너비는 집광로봇의 작용기 너비가 적용되기 때문에 해당 셀의

커버리지 완료점은 자동으로 선택된다. 커버리지 횟수에 따라 커버리지 완료점이 선택되는 것이다.

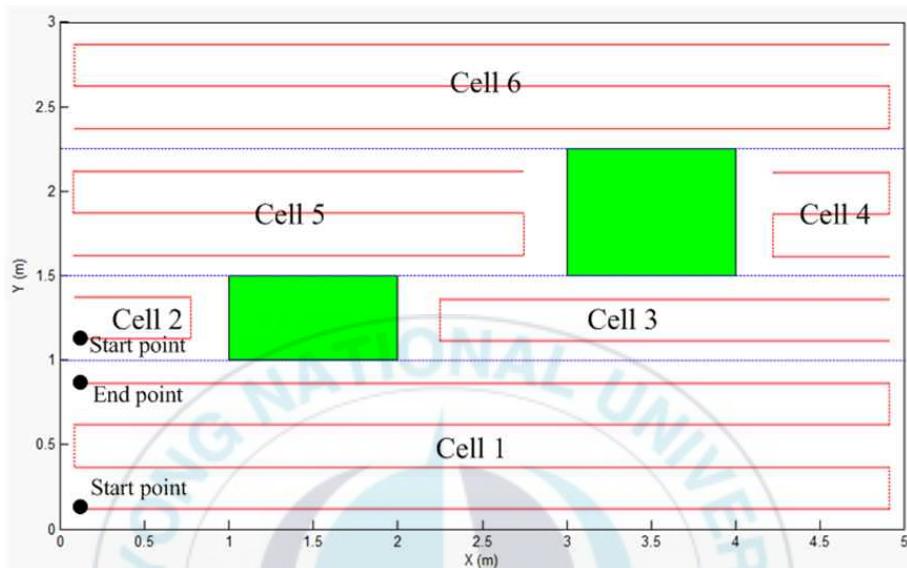


Fig. 2.27 Example of basic coverage path

이후에 셀 커버리지 순서에 따라 이전 셀과 다음 셀 간의 이동경로를 생성한다. 이전 셀의 완료점은 결정되어 있기 때문에 다음 셀 시작점에 대하여 가시성 그래프 알고리즘을 적용하여 셀 간의 이동경로가 생성된다. 이러한 과정은 셀 간의 이동경로가 모두 생성될 때까지 계속되며, Fig. 2.27의 커버리지 기초경로를 이용하여 셀과 셀 간의 이동경로를 Fig. 2.28과 같이 나타낼 수 있다. Fig. 2.28에서 점선으로 나타난 것이 셀 간의 이동경로이며, 화살표로 표시된 것이 다음 셀의 시작점을 나타낸 것이다.

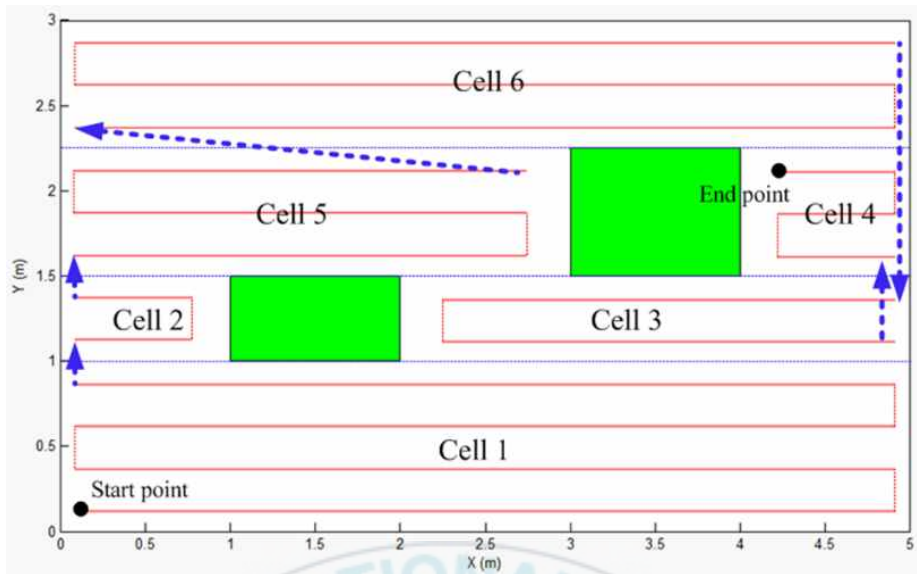


Fig. 2.28 Example of path generation between cells

셀 간의 이동경로가 생성된 후 전체 커버리지 경로를 생성한다. 이때 셀 간의 이동경로가 많이 중복되는 경우 수정 스위핑 라인법을 이용하여 셀 간의 이동경로를 수정한다. 이후에 전체적인 경로가 생성되면 회전구간에 대한 처리를 수행한다. 심해저는 지질이 매우 부드럽기 때문에 집광로봇이 부드러운 움직임을 가져야 커버리지 효율이 상승하기 때문이다. 따라서 Fig. 2.29와 같이 회전구간이 90° 일 경우 $L/2$ 의 원을 배치하여 해당 구간을 보다 매끄럽게 생성한다. 또한 회전구간이 180° 일 경우에도 같은 방법을 적용하여 Fig. 2.30과 같은 부드러운 경로(Smooth path)를 생성한다.

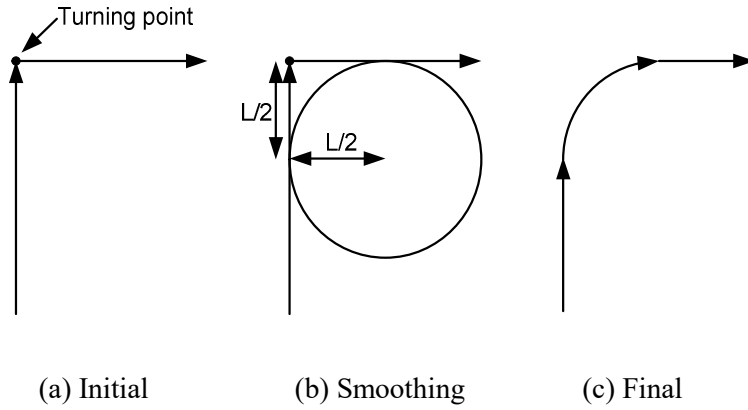


Fig. 2.29 Smoothing 90 degree turning path

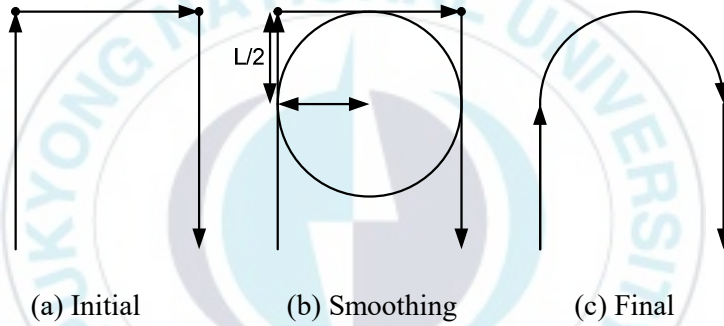


Fig. 2.30 Smoothing 180 degree turning path

회전구간에 대하여 커버리지 경로를 수정한 이후에 생성되는 경로가 최종적인 전체 커버리지 경로가 된다.

생성된 전체 커버리지 경로의 커버리지 효율(Coverage ratio of given area, CR)을 나타내기 위하여 식 (2.4)를 이용한다.

$$CR = \frac{(D_{cp} - D_{op}) \times W_e - A_{eo}}{A_{gv} - A_{ob}} \times 100 \quad (2.8)$$

Table 2.3은 식 (2.8)에 나타낸 각각의 매개변수에 대한 설명을 나타낸다.

Table 2.3 Description of parameters related with coverage ratio

Parameter	Description	Unit
CR	Coverage ratio	%
d_c	Distance of all coverage paths	m
d_o	Distance of all overlapped paths	m
w	Width of mining robot's effector	m
A_e	Covered external area by robot's effector	m^2
A_{gv}	Giver area	m^2
A_{ob}	Obstacle area	m^2

2.8 전체 커버리지 경로계획의 시뮬레이션

본 절에서는 2.2절에서 2.7절까지의 전체 커버리지 경로계획의 유효성을 검증하기 위해 시뮬레이션 결과를 제시한다. Table 2.4는 시뮬레이션에서 쓰인 매개변수의 값을 나타낸다.

Table 2.4 Numeric values for simulation

Parameter	Description	Value	Unit
W	Effector width of robot	0.3	m
W_c	Coverage interval	0.3	m
h_g	Given area height	6	m
W_g	Given area width	6	m
δ_s	Threshold value	0.5	m

Fig. 2.31은 시뮬레이션을 위해 주어진 3D 영역을 나타낸 것이다. X축과 Y축은 각각 주어진 영역의 너비와 높이를 나타낸다. 시뮬레이션에서는 Table. 2.4와 같이 너비와 높이가 6m인 영역을 정해 커버리지 경로를 계획하고자 한다. 주어진 영역은 장애물 3개를 가지며, 경사 문턱값 δ_s 에 의해 2D 바이너리 영역으로 변환된 결과는 Fig. 2.32에 나타내었다. Fig. 2.32에 장애물 영역의 번호 및 각각의 위치를 표시하였다.

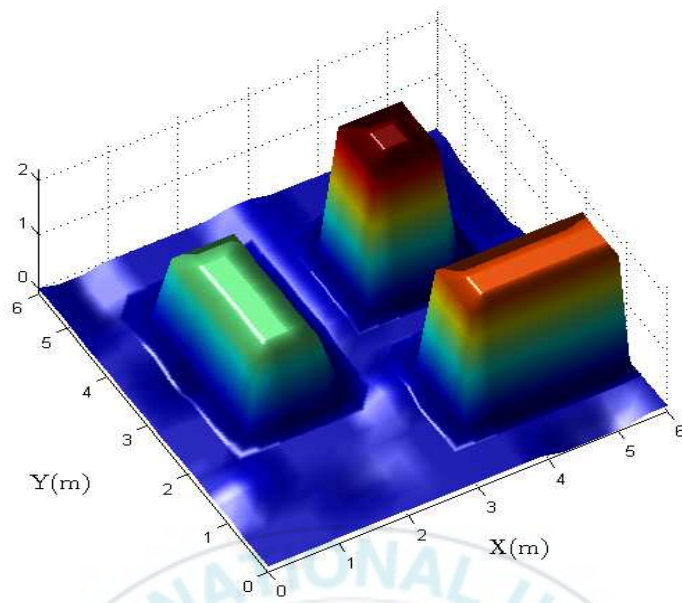


Fig. 2.31 Given 3D area

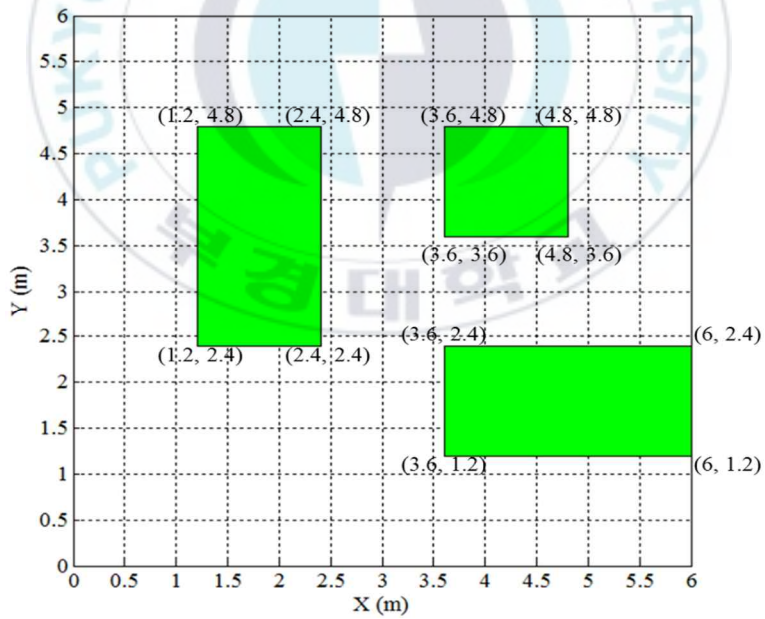


Fig. 2.32 Binarized map

Figs. 2.33~34는 주어진 영역을 수평방향 및 수직방향으로 분할한 결과이다. 굵은 실선은 분할된 경계를 표현하며, 해당 셀에 각 셀의 번호를 표시하였다. Fig. 2.33은 모스 셀 분할법을 이용하여 Fig. 2.32의 주어진 영역을 수평방향으로 셀 분할한 결과이며, Fig. 2.34는 모스 셀 분할법을 이용하여 Fig. 2.32의 주어진 영역을 수직방향으로 셀 분할한 결과이다. Table 2.5는 각각의 셀 분할 방향에 대하여 셀의 개수, 최소 고도의 합을 나타낸 것이다.

Table 2.5 Comparison of cell decomposition results

Direction	Number of cells	Sum of altitude
Horizontal	8	32
Vertical	7	28

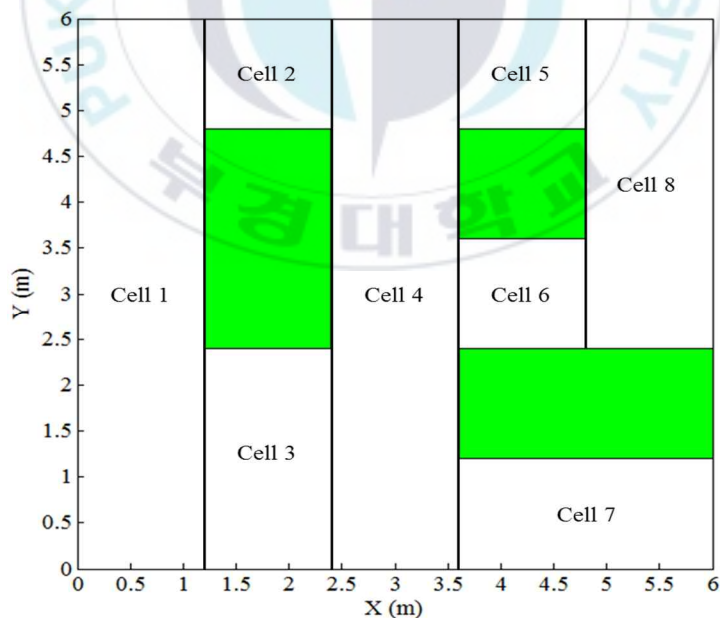


Fig. 2.33 Horizontal cell decomposition result

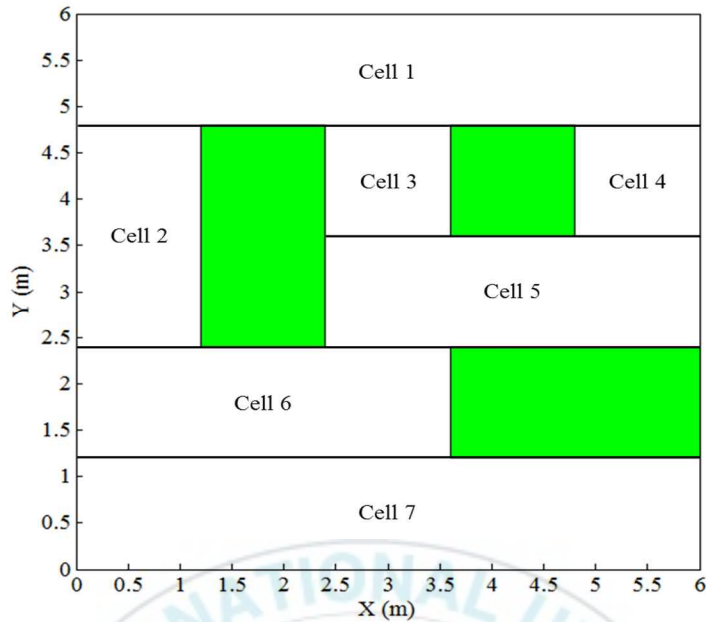


Fig. 2.34 Vertical cell decomposition result

수평방향 셀 분할 결과 및 수직방향 셀 분할 결과를 바탕으로 최적의 셀 분할 결과를 도출한다. 최소 고도 계산 알고리즘을 적용하기 전에 기본 맵을 결정한다. 수직방향 셀 분할 결과가 수평방향 셀 분할 결과보다 셀의 개수 및 최소 고도 합이 작으므로 수직방향 셀 분할 결과를 기본 맵으로 결정한다. 이후에 최소 고도 계산 알고리즘을 적용한 결과는 Fig. 2.35와 같고, 각 단계마다 전체 셀의 개수 n 및 최소 고도 합 $S(j)$ 를 계산한 결과를 Table 2.6에서 확인할 수 있다. $j=10$ 일 때 $S(j)$ 가 최소값을 가지므로 이 때의 결과를 셀 분할 최적화 결과로 결정한다.

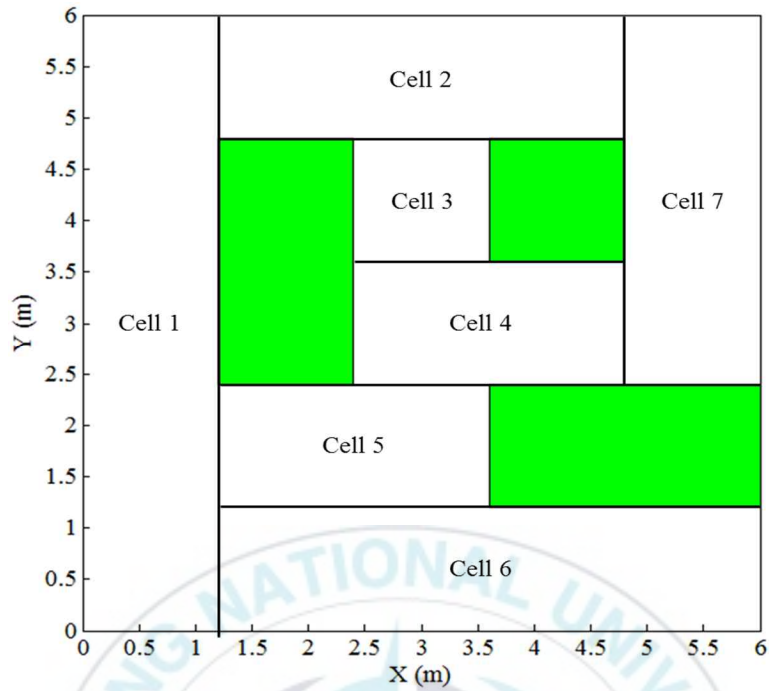


Fig. 2.35 Cell decomposition optimization result

Table 2.6 Number of cell and the sum of altitude of each step obtained from MSA algorithm

Way to split the graph j	Total number of cell n	Sum of altitude $S(j)$
$j=1$	8	32
$j=2$	7	28
$j=3$	7	28
$j=4$	8	32
$j=5$	8	32
$j=6$	9	36
$j=7$	9	36
$j=8$	9	36
$j=9$	8	32
$j=10$	7	28

최적 셀 분할 결과가 나타나면 셀 커버리지 순서를 결정하기 위해 셀과 셀 간의 인접한 관계를 Fig. 2.36과 같이 신장트리로 나타낸다. 각 노드 번호는 셀 번호와 같으며, 각 셀 간의 인접한 관계는 엣지로 나타내어진다.

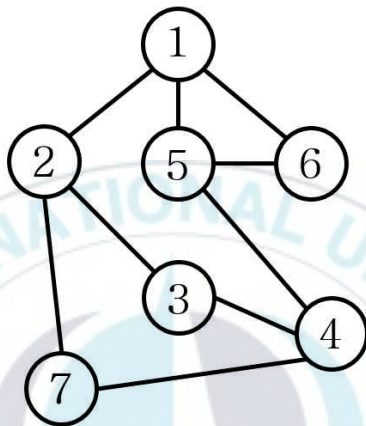


Fig. 2.36 Spanning tree based on cell decomposition optimization result

생성된 신장트리를 바탕으로 너비우선 탐색 알고리(BFS)즘 및 깊이 우선 탐색 알고리즘(DFS)을 적용하여 셀 커버리지 순서를 결정하면 Table 2.7과 같다.

Table 2.7 Cell coverage sequence using BFS and DFS

Method	Cell coverage sequence
BFS	1→6→5→2→4→3→7
DFS	1→6→5→4→3→2→7

너비우선 탐색 알고리즘(BFS)을 바탕으로 생성되는 셀 커버리지 순서를 이용한 전체 커버리지 경로는 Fig. 2.37과 같다. 붉은 점선으로 표시된 부분은 셀 간의 이동경로 및 중복되는 경로이다. 셀 5에서 셀 2로 이동하는 경우, 셀 2에서 셀 4로 이동하는 경우, 셀 4에서 셀 3으로 이동하는 경우 및 셀 3에서 셀 7로 이동하는 경우 중복되는 경로가 발생하지만 각각의 셀이 서로 인접한 관계를 이루지 않기 때문에 수정 스위핑 라인법을 적용할 수 없다. 회전구간에 대한 처리를 할 경우 생성되는 최종적인 전체 커버리지 경로는 Fig. 2.38와 같다.

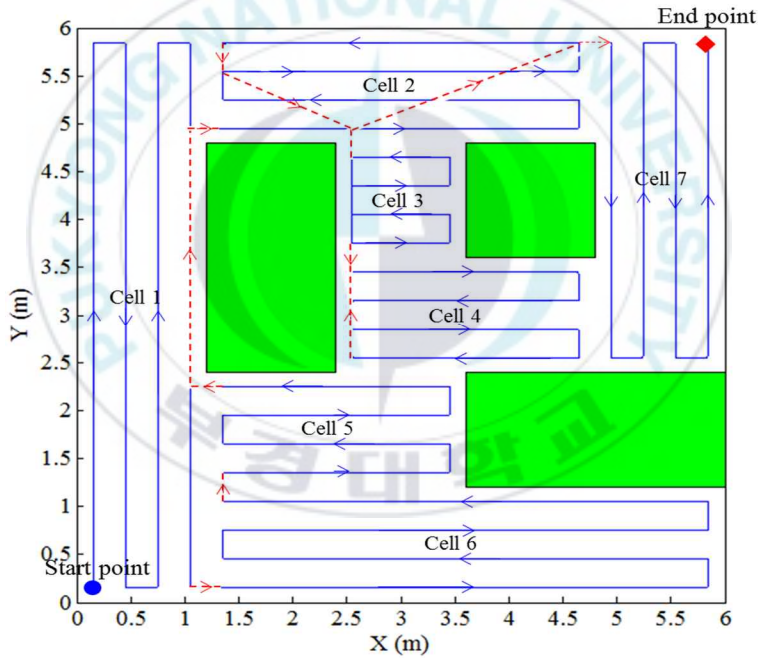


Fig. 2.37 Generated coverage path based on sequence of BFS algorithm

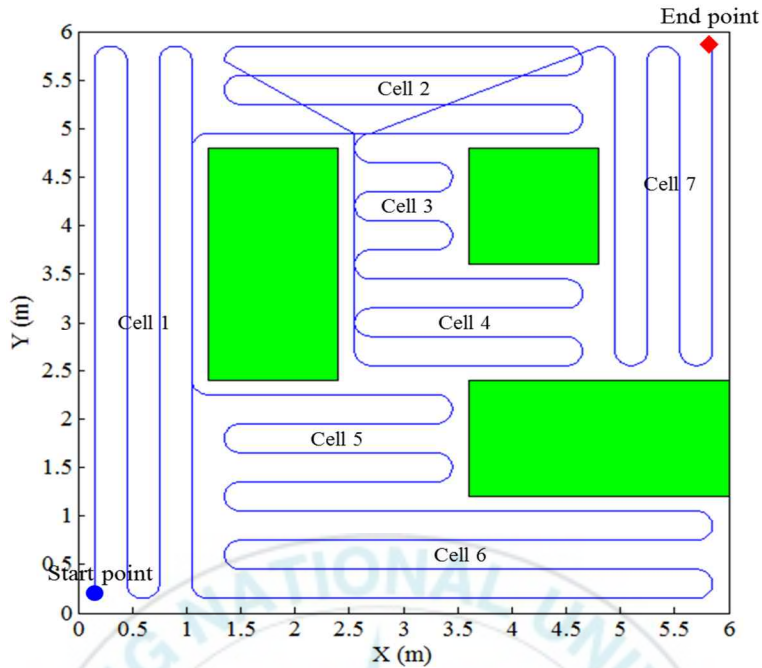


Fig. 2.38 Generated smooth coverage path based on sequence of BFS algorithm

깊이우선 탐색 알고리즘(DFS)을 바탕으로 생성되는 셀 커버리지 순서를 이용한 전체 커버리지 경로는 Fig. 2.39와 같다. 붉은 점선으로 표시된 부분은 셀 간의 이동경로 및 중복되는 경로이다. 셀 5에서 셀 4로 이동하는 경우, 셀 3에서 셀 2로 이동하는 경우 및 셀 2에서 셀 7로 이동하는 경우 중복되는 경로가 발생하며 서로 인접한 관계를 이루고 있기 때문에 수정 스위핑 라인법을 적용할 수 있다. 셀 2에서 셀 7로 이동하는 경우 수정 스위핑 라인법을 적용하여 중복되는 경로를 최소화하고, 회전구간에 대한 처리를 할 경우 생성되는 최종적인 전체 커버리지 경로는 Fig. 2.40과 같다.

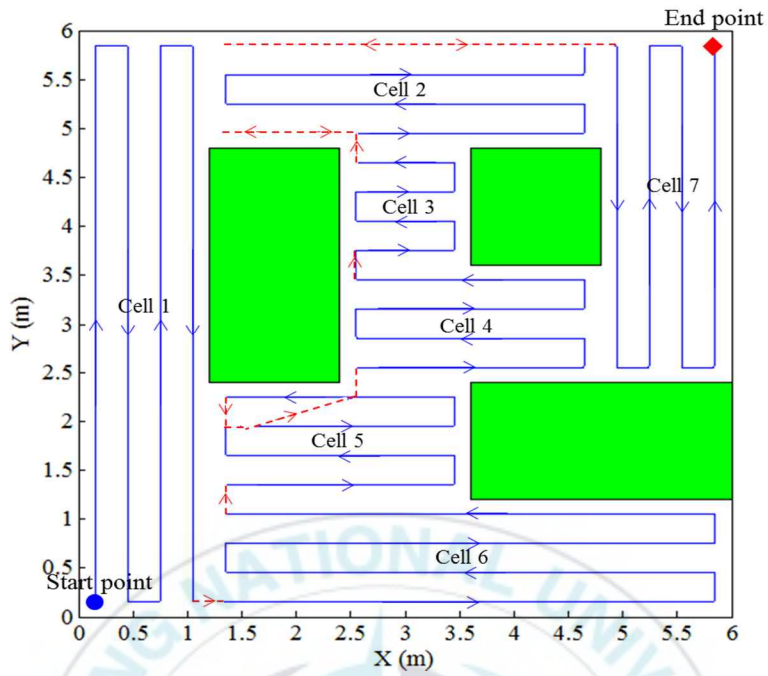


Fig. 2.39 Generated coverage path based on sequence of DFS algorithm

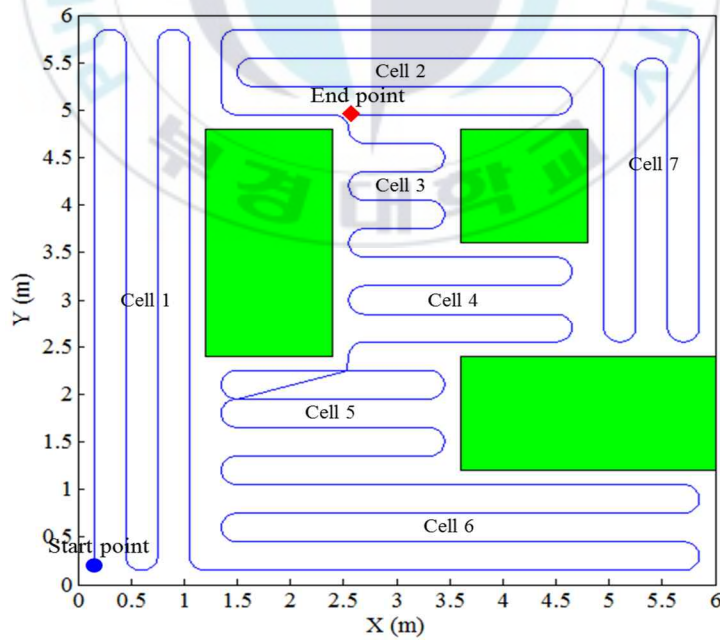


Fig. 2.40 Generated smooth coverage path based on sequence of DFS algorithm

Table 2.8은 너비우선 탐색 알고리즘과 깊이우선 탐색 알고리즘을 통해 생성되는 셀 커버리지 순서를 이용하여 Fig. 2.38과 Fig. 2.40에 나타난 전체 커버리지 경로를 비교한 것이다. 전체 커버리지 경로의 길이는 깊이우선 탐색 알고리즘에 바탕한 전체 커버리지 경로가 너비우선 탐색 알고리즘에 바탕한 전체 커버리지 경로보다 6.983m 더 짧은 것으로 나타난다. 집광로봇이 일정한 속도 0.02m/s로 커버한다고 가정했을 때 전체 커버리지를 완료하는 시간을 Table 2.7에 나타내었다. 집광로봇의 회전수는 같으며, 중복되는 경로는 깊이우선 탐색 알고리즘을 바탕으로 생성된 전체 커버리지 경로가 너비우선 탐색 알고리즘을 바탕으로 생성된 전체 커버리지 경로보다 9.781m만큼 더 작은 것으로 나타난다. 또한 전체 커버리지 경로의 커버리지 효율을 식 (2.8)을 이용하여 계산했을 때, 깊이우선 탐색 알고리즘을 이용한 전체 커버리지 경로가 너비우선 탐색 알고리즘을 이용한 전체 커버리지 경로보다 3% 효율적임을 알 수 있다.

Table 2.8 Comparison of complete coverage path using sequence of BFS and DFS

Method	BFS	DFS	Unit
Distance of all coverage path	99.745	92.762	<i>m</i>
Distance of all overlapped path	11.296	1.515	<i>m</i>
Number of turns	32	32	<i>times</i>
Coverage time	4987.25	4638.1	<i>s</i>
Coverage ratio	92.13	95.13	%

Table 2.8을 통해 깊이우선 탐색 알고리즘이 너비우선 탐색 알고리즘보다 효율적인 것을 알 수 있다. 따라서 이후에 전체 커버리지 경로의 실험에 사용되는 목표경로를 깊이우선 탐색 알고리즘을 바탕으로 생성된 경로인 Fig. 2.40으로 결정한다.



제 3 장 시스템 구성

본 장에서는 이륜 집광로봇과 유사한 동작성을 갖춘 무인 운반차 (Automated guided vehicle, AGV) 시스템의 구성 및 그 제원에 대해 서술한다. 이 AGV는 기구시스템과 제어시스템으로 구성되어 있으며, 제어시스템은 센서부, 제어부, 조작부 및 전원 공급장치로 구성된다.

본 연구에서 사용된 실제 이륜 무인 운반차(AGV)의 시스템은 Fig. 3.1과 같다.

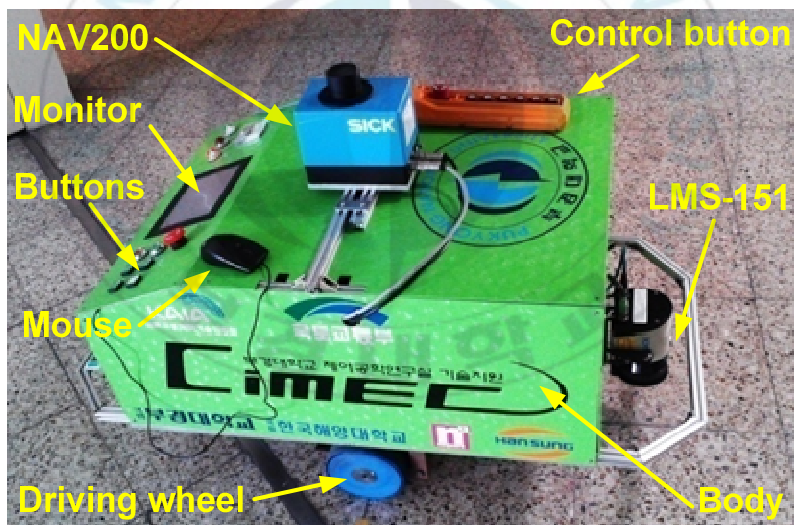


Fig. 3.1 Mechanical design of AGV

3.1 기구시스템의 구성

Fig. 3.2는 본 연구에서 사용된 AGV시스템의 구성도를 나타낸다. AGV 시스템은 가로 60cm, 세로 120cm, 높이 56.5cm로 차동 구동 시스템을 사용한다. AGV는 Fig. 3.2와 같이 본체의 좌·우 중간에 설치되어 있는 차동형 구동 바퀴가 BLDC 모터로 구동되며, 전·후로는 균형을 맞추기 위한 두 개의 수동 바퀴 캐스터 바퀴(Caster wheel)을 사용한다. 본체의 상단에는 위치인식을 위한 NAV-200 레이저 항법 시스템(Laser Navigation System)이 설치되어있으며, 상단 아래쪽에는 Mysen-M 가속도 센서가 설치되어있다.

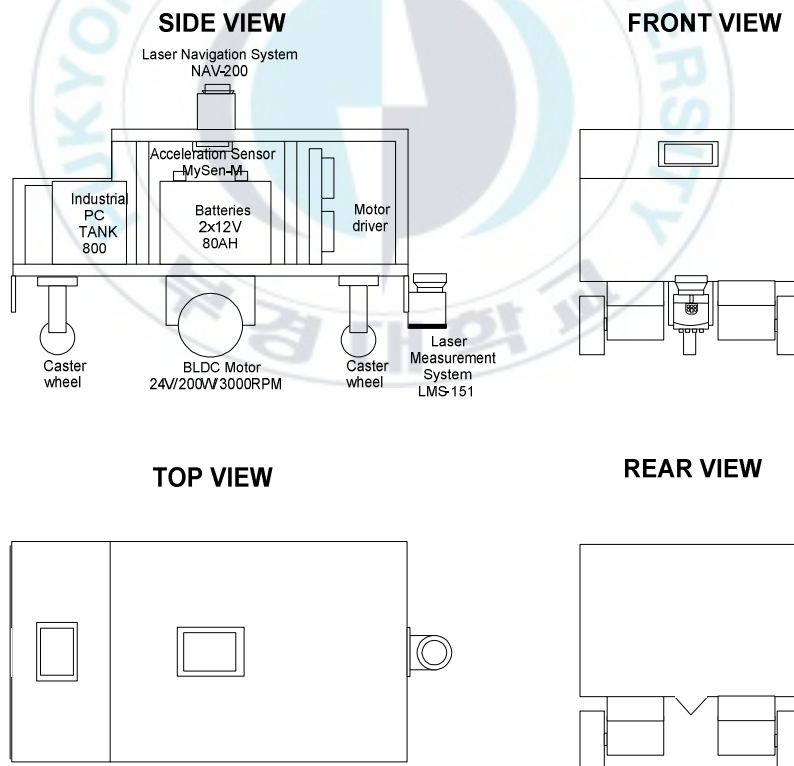


Fig. 3.2 System configuration

AGV 본체 내부에는 산업용 컴퓨터(Industrial PC)와 2개의 12V로 직렬 연결된 24V축전지(Battery)가 설치되어있으며, AGV 본체에는 터치 모니터가 장착되어 산업용 컴퓨터와 연결을 통해 모니터링이 가능하다. Fig. 3.3은 AGV 시스템의 차동형 구동 바퀴 및 수동 바퀴, 가속도 센서 MySen-M의 구성도를 나타낸 것이다.

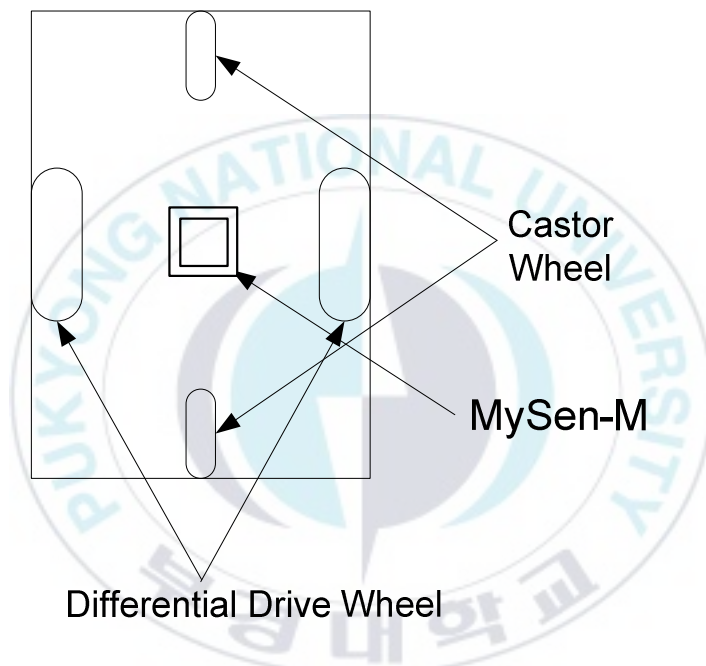


Fig. 3.3 Configuration of differential drive AGV system

3.2 제어시스템의 구성

본 연구에서 사용되는 제안된 제어 시스템은 아래의 Fig. 3.4와 같다. 제어시스템은 센서부, 제어부, 구동부, 그리고 전원 공급장치로 구성되어 있다. 센서부에서는 산업용 컴퓨터와 RS-232통신으로 서로 연결되는 MySen-M 가속도 센서 및 레이저 항법 시스템 NAV-200이 있으며, 레이저 항법 시스템의 전송속도는 19,200Hz이다. 제어부에서는 산업용 컴퓨터 Tank-800이 주 컨트롤러이며 입력을 위하여 블루투스(Bluetooth) 통신 기반 무선 마우스와 키보드가 사용된다. USB로 연결된 터치스크린 모니터(Touch Screen Monitor)를 이용하여 결과 값과 진행 상황을 모니터링 할 수 있다.

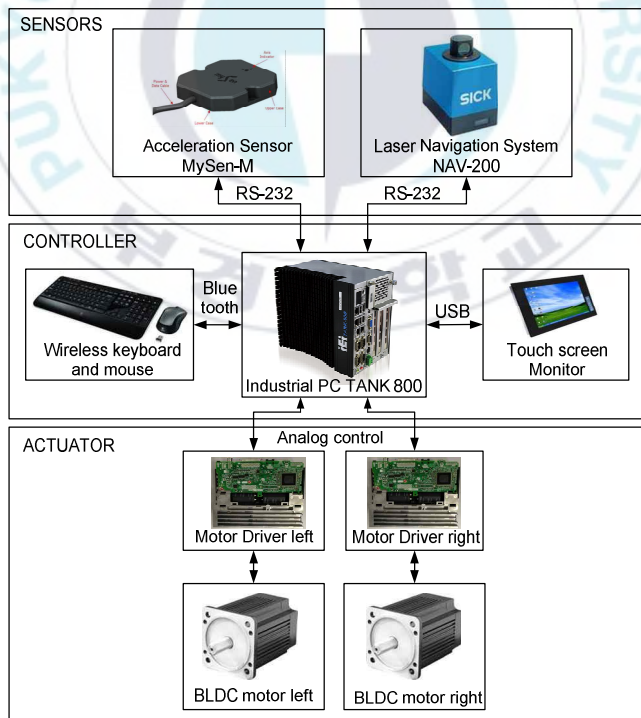


Fig. 3.4 Hardware configuration of AGV

주 컨트롤러인 산업용 컴퓨터로부터 아날로그 신호값과 논리값이 액추에이터로 보내지며, 모터드라이버는 주 컨트롤러에서 참조된 아날로그 신호를 수신하며 이 기준값을 기반으로 BLDC모터의 속도를 제어한다.

3.2.1 센서부

본 연구에서는 두 종류의 센서를 AGV에 적용하였으며 레이저 항법 시스템(NAV-200)과 가속도센서(MySen-M)을 사용하였다.

Fig. 3.5는 레이저 항법 시스템 NAV-200를 나타내며, NAV-200의 기본 원리는 Fig. 3.6과 같다. 무인 운반차에 장착되어 있을 때 NAV-200은 360°로 회전하며 고정된 반사판(Reflector)으로부터 2차원의 좌표를 계속해서 측정 및 계산한다. 위치 인식을 위해 최소한 3개의 반사판이 필요하며, 4~5개의 반사판을 사용하여 최대 인식 범위(28.5m) 내에서 계속해서 반사판을 형성한다면 매우 정확한 위치를 측정할 수 있다. 측정되는 위치 정보는 산업용 컴퓨터로 전송된다.



Fig. 3.5 Laser navigation system NAV-200

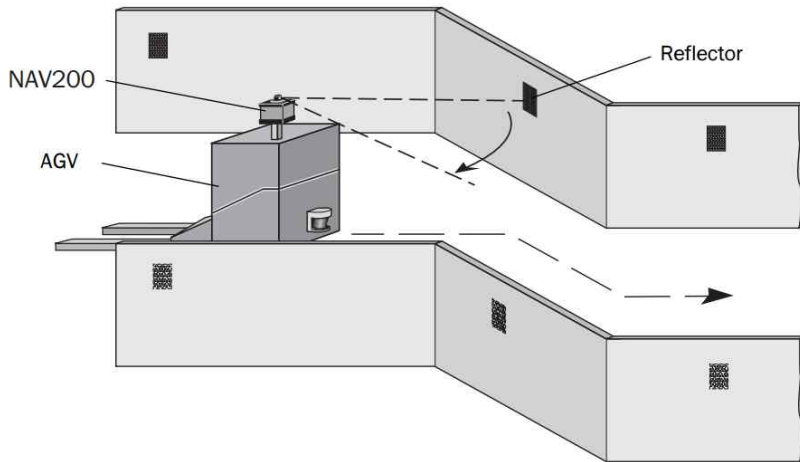


Fig. 3.6 Position measurement of the NAV-200 by
detection of reflector arrangements

Table 3.1은 레이저 항법 시스템의 세부 제원을 나타낸 것이다.

Table 3.1 Specification of laser navigation system NAV-200

No	Parameter	Value
1	Light source	Infrared (855 nm)
2	Laser class	1
3	Field of view	360 °
4	Scanning frequency	8 Hz
5	Angular resolution	0.1°
6	Operating range	1.2 m – 28.5 m
7	Max. range 10 % reflectivity	28.5 m
8	Data communication	Serial (RS-232)
9	Data transmission rate	19200 Hz

10	Operating voltage:	$\geq 24 \text{ V DC} \pm 25\%$
11	Power consumption:	24 W
12	Weight	3.3 kg
13	Dimensions	176 mm x 178 mm x 115 mm

Fig. 3.7은 레이저 항법 시스템의 위치인식 시스템을 나타낸 것이다. $Rn(X_n, Y_n)$ 은 레이저 항법 시스템이 인식할 수 있는 반사판과 그 위치를 나타낸다. d_n 은 각 반사판과 레이저 스캐너까지의 거리이고 이는 레이저가 송신되어 반사판에 의해 반사되어 레이저 스캐너에 다시 수신되기까지의 시간과 레이저의 속도에 의해 구해진다. 또한 θ_A 는 절대좌표계의 X축과 레이저 스캐너의 각도를 나타내며 θ_n 는 레이저 스캐너의 이동 좌표계의 X축과 각 반사판과 레이저 스캐너를 잇는 직선의 각도를 나타낸다.

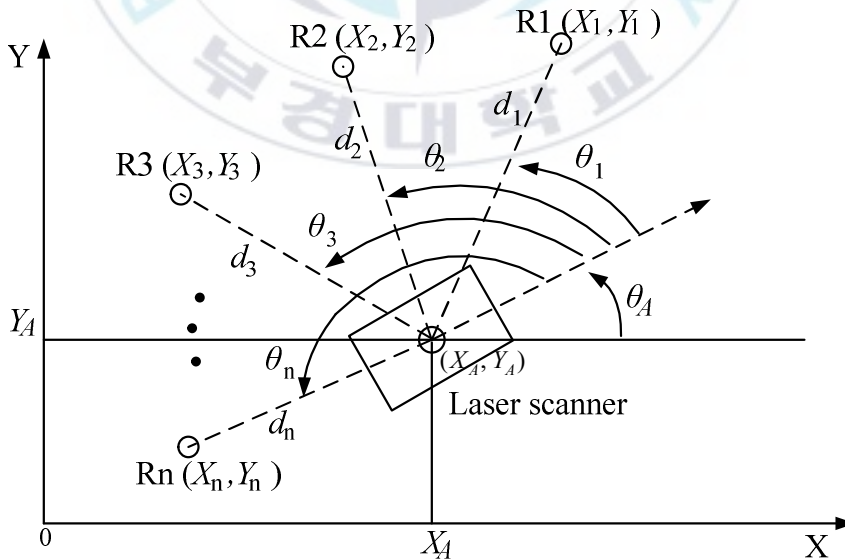


Fig. 3.7 Laser scanner navigation system

AGV의 현재 자세를 인식하기 위한 식은 [34, 35]에서 다음과 같이 나타낼 수 있다.

$$X_A = \left[1 / (m_2 - m_1)(1 + m^2) \right] \{ (m_2 m - 1) \times [(m_1 + m)X_1 + (m_1 m - 1)Y_1] - (m_1 m - 1) \times [(m_2 + m)X_2 + (m_2 m - 1)Y_2] \} \quad (3.1)$$

$$Y_A = \left[1 / (m_2 - m_1)(1 + m^2) \right] \{ (m_1 + m) \times [(m_2 + m)X_2 + (m_2 m - 1)Y_2] - (m_2 + m) \times [(m_1 + m)X_1 + (m_1 m - 1)Y_1] \} \quad (3.2)$$

$$\theta_A = a \tan 2(Y_1 - Y_A, X_1 - X_A) - \theta_1 \quad (3.3)$$

$m_1 = \tan \theta_1$, $m_2 = \tan \theta_2$, $m_3 = \tan \theta_3$ 이며, $m = \tan \theta$ 이다. m 은 아래의 식에 의해 다음과 같이 구해진다.

$$m = \frac{(m_3 - m_1)(Y_1 - Y_2 - m_1 X_1 + m_2 X_2) - (m_2 - m_1)(Y_1 - Y_3 - m_1 X_1 + m_3 X_3)}{(m_3 - m_1)(m_1 Y_1 + X_1 - m_2 Y_2 - X_2) - (m_2 - m_1)(m_1 Y_1 + X_1 - m_3 Y_3 - X_3)} \quad (3.4)$$

두 번째 센서로서는 가속도 센서 MySen-M이며 Fig. 3.8에 나타낸다. MySen-M 가속도 센서는 MEMS 센서 기반의 소형/경량 관성 측정 장치로써 무인항공기(UAV), 무인차량(AGV), 무인잠수정(AUV)등에 적용할 수 있도록 개발된 센서 모듈로써 자세예측 자율제어 및 유도, 위치 인지, 항

법 응용에 적합한 제품이다. 본 센서는 AGV의 좌표 오차를 줄이기 위한 센서이다. 정적가속도(각도성분)와 동적가속도(가속도성분)로 나누어지며 여기서 발생하는 정적가속도를 추출한다. 칼만필터를 이용하여 각도를 추정하며 추출한 정적가속도 성분을 제거하여 오차를 보정하고 가속도 센서를 적분하는 과정에서 발생하는 누적오차를 필터링을 통하여 보정한다. 데이터 형식은 2byte형태로 구성되며 본 연구에서는 RS-232를 연결한 통신 방식을 사용한다.

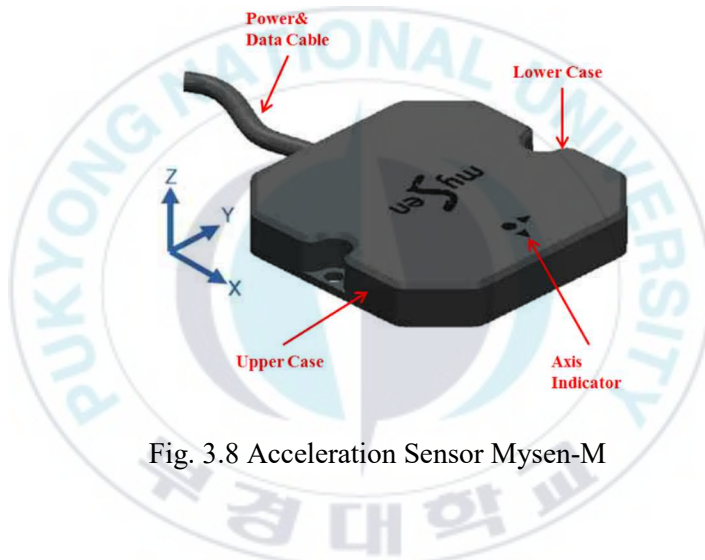


Fig. 3.8 Acceleration Sensor Mysen-M

MySen-M 가속도 센서의 세부 제원은 Table 3.2에 나타낸다.

Table 3.2 Specification of MySen-M

No	Parameter	Value
1	Digital interface	Serial UART(TTL3.3V) / RS232
2	Operating voltage	DC 5V

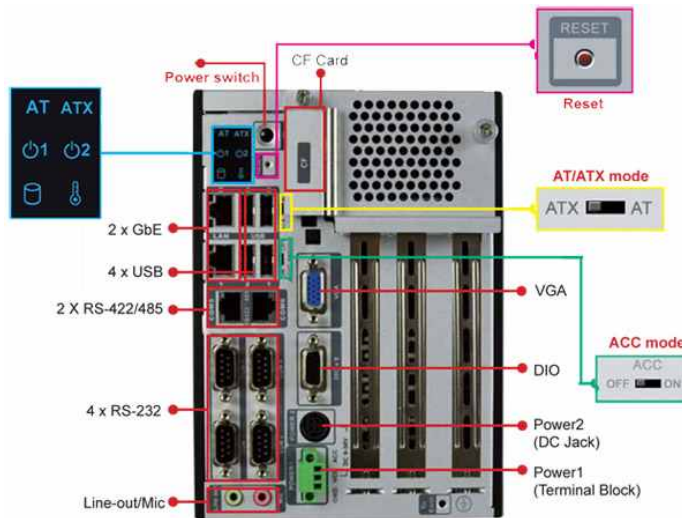
3	Operating temp	-10°C ~ 60°C
4	Power Consumption	1W(Nominal)
5	Data sampling rate	100KHz
6	Dimensions(W x L x H)	55 x 55 x 10(mm)
7	Weight	≤100g

3.2.2 제어부

본 연구에서는 산업용 컴퓨터 TANK-800을 주 컨트롤러로 사용한다. 본 절에서는 컨트롤러로 사용되는 산업용 컴퓨터 Tank-800의 구조는 Fig. 3.9에 표시된다.



(a) Front view of TANK-800



(b) Rear view of TANK-800

Fig. 3.9 Industrial PC TANK-800

산업용 컴퓨터 TANK-800의 세부 제원은 아래의 Table 3.3과 같다.

Table 3.3 Specification of industrial PC TANK-800

No	Parameter		Value
1	Chassis	Dimensions	136 x 219 x 188(mm)
2	Mother board	CPU	Intel® Atom™ D525 1.8GHz dual-core processor
		Chipset	Intel® ICH8M
		Ethernet	Dual Realtek RTL8111E PCIe GbE controllers support ASF 2.0
3	Storage	SATA	2.5 inch SATA HDD bay

4	System Function	USB	4 x USB 2.0
		Ethernet	2 x RJ-45
		RS-232	4 x DB-9
		RS-422/485	2 x RJ-45
		Display	1 x VGA
		Resolution	Up to 2048x1536
		Audio	1 x Mic-in, 1 x Line-out
		DIO	1 x DB-9
		Interior Expansions	One PCIe x4 (physical one PCIe x16 slot) and two PCI slots
5	Power	Power Supply	1.5V(± 0.3 V) ~ 36V
		Power Consumption	33W(Without add-on card)
6	Reliability	Operating Temperature	-20°C ~ 70°C
		Weight	3.0Kg

3.2.3 구동부

구동부는 모터드라이버 MR-J3-20A와 BLDC모터 SWIS BG90-S024로 구성되어 있다. 본 연구에 사용된 모터드라이버는 Fig. 3.10에 나타낸다. 모터드라이버에 의해 허용 가능한 최대 전류는 20A이다.

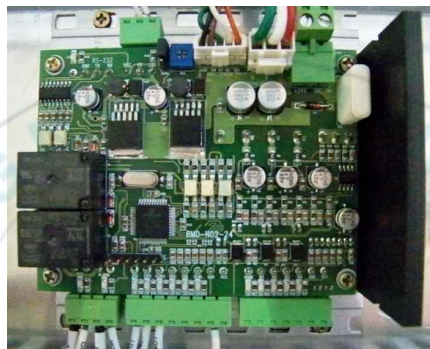


Fig. 3.10 Motor driver

Fig. 3.11은 본 연구에 사용된 BLDC모터 SWIS BG90-S024를 나타낸 것이다. 모터는 24V를 사용하며 세부 제원은 Table 3.4에 나타내었다.



Fig. 3.11 BLDC motor SWIS BG90-S024

Table 3.4 Specification of BLDC motor SWIS BG90-S024

No	Parameter	Value
1	Rated Voltage	5V
2	Rated Output	200W
3	Rated Current	-11.9A
4	Rated Speed	3000RPM
5	Rated Torque	6.3Kgf-cm
6	Weight	2.7Kg
7	Voltage Constant	14.8V/Krpm
8	Torque Constant	1.4 Kgf.cm
9	Rotor Inertia	0.24x10 ⁻³ kg.m ²
10	Armature Resistance	0.9ohm
11	Armature Inductance	0.9mH
12	Mechanical Time Constant	10ms
13	Electrical Time Constant	1.1ms

3.2.4 전원 공급장치

Fig. 3.12는 12V 축전지 Rocket AGM 80-12를 나타내며 Table 3.5는 그 제원을 나타낸 것이다. 2개의 축전지를 직렬로 연결하여 24V를 AGV의 모든 시스템에 공급한다.



Fig. 3.12 Structure of Battery

Table 3.5 Specification of Battery Rocket AGM 80-12

No	Parameter		Value
1	Nominal Voltage		12V
2	Capacity(AH)	10HR 1.80(V/Cell)	80.0AH
		5HR 1.70(V/Cell)	68.0 AH
		3HR 1.67(V/Cell)	61.5 AH
		1HR 1.60(V/Cell)	48.0 AH
		0.5HR 1.60(V/Cell)	37.5 AH
3	Dimensions	Length	332mm
		Width	174mm
		Height	229mm
4	Weight		24.7Kg
5	Terminal Type		Bolt Terminal

제 4 장 시스템 모델링 및 제어기 설계

본 장에서는 무인 운반차의 운동학적 모델링 및 경로 추적 제어기를 제시한다. 무한궤도(Caterpillar)를 이용한 집광 로봇과 이륜 이동로봇인 무인 운반차(AGV) 간의 모델링과 동작성이 유사하므로 본 연구에서는 무인 운반차의 모델링을 제시한다.

4.1 AGV 구동 바퀴의 특성

4.1.1 운동학적 모델링

Fig. 4.1과 같이 본 연구에 사용되는 AGV는 점 C의 좌우에 구동 바퀴와 전후에 AGV의 균형을 맞춰 주기 위해 캐스터 바퀴가 설치되어 있다.

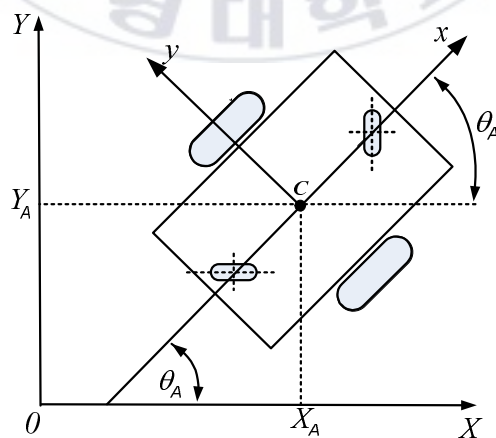


Fig. 4.1 Coordinate for AGV modeling

점 C는 AGV의 기하학적 중심인 좌표이며, 점 C는 AGV의 기준점이다. 전역 고정 좌표계에서 점 C의 위치는 좌표 (X_A, Y_A) 및 전역 고정 좌표계와 이동 좌표계 사이의 각도인 θ_A 에 의해 주어진다. ξ_1 은 전역 고정 좌표계에서 AGV의 위치와 방향각으로 구성된 위치벡터이며, 다음과 같이 표현된다.

$$\xi_1 = \begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix} \quad (4.1)$$

전역 고정 좌표계 XOY 에서 AGV의 위치벡터 및 이동 좌표계 xCy 에서의 AGV 위치벡터 ξ_A 의 관계는 식 (4.2)와 같이 표현할 수 있다.

$$\xi_A = \mathbf{R}(\theta_A)\xi_1 \quad (4.2)$$

여기서 $\mathbf{R}(\theta_A)$ 는 직교하는 3 x 3 회전행렬(Rotational Matrix)으로 식 (4.3)과 같다.

$$\mathbf{R}(\theta_A) = \begin{bmatrix} \cos \theta_A & \sin \theta_A & 0 \\ -\sin \theta_A & \cos \theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

4.1.2 구동 바퀴(표준 구동 바퀴)

구동 바퀴는 조향 장치에 대하여 회전의 수직축(Vertical axis)이 없는 회전형 바퀴로 차체(Chassis)와 각도는 고정되어 있으며, AGV의 전·후 움직임은 지면과 접촉된 점에서 바퀴면과 회전방향에 따라 제한되어 있다. Fig. 4.2는 구동 바퀴를 나타내며, AGV의 이동 좌표계 xCy 에 대한 상대적인 위치를 나타낸다. 바퀴의 위치는 차체에서 바퀴까지 거리 l 과 이동 좌표계의 x 축을 기준한 점 C 로부터 바퀴까지의 각 α 로 구성된 극좌표로 표시된다. β 는 차체와 바퀴면 사이의 각도이며, r 은 바퀴의 반지름을 나타낸다. ϕ 는 바퀴 수평축 주위를 회전하는 회전각도이며 시간함수로 표현된다.

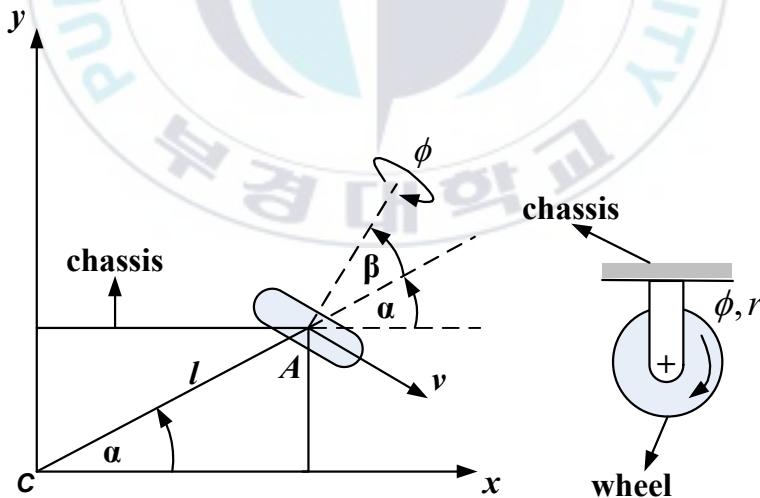


Fig. 4.2 A fixed standard wheel and its parameters

바퀴의 회전 제약조건(Rolling constraint)에 의해 구동 바퀴는 바퀴면의 방향에 따른 모든 운동이 바퀴의 공전이 발생할 만큼의 속도가 동반되어야 하며, 따라서 접점에서 순수 구름(Pure rolling)이 발생해야 하므로, 다음과 같이 표현된다.

$$[\sin(\alpha + \beta) \quad \{-\cos(\alpha + \beta)\} \quad (-l)\cos\beta] \mathbf{R}(\theta_A) \dot{\xi}_1 - r\dot{\phi} = 0 \quad (4.4)$$

바퀴의 미끄럼 제약조건(Sliding constraint)에 의해 구동 바퀴면에 수직인 바퀴의 운동에 대한 요소들은 0이어야하므로, 다음과 같이 표현된다.

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l\sin\beta] \mathbf{R}(\theta_A) \dot{\xi}_1 = 0 \quad (4.5)$$

4.1.3 수동 바퀴(캐스터 바퀴)

수동 바퀴는 수직축을 중심으로 회전할 수 있다. 그러나 구동 바퀴와는 달리 수동 바퀴에서 회전에 대한 수직축은 지면의 접점을 통과하지 않는다. Fig. 4.3은 수동 바퀴를 나타낸 것이며, 수동 바퀴의 위치를 상세히 정하기 위해 필요한 변수들을 나타내었다. 수동 바퀴는 B 의 위치에 있고 길이 AB 의 단단한 지지대에 연결되어있으며, 점 A 의 수직축은 차체에 고정되어있다. 점 A 는 Fig. 4.3에 명시된 무인 운반차의 이동 좌표계에 위치한다. 이를 통해 AB 는 바퀴면과 항상 평행하다고 가정할 수 있다. 구동 바퀴와 마찬가지로 수동 바퀴는 시간의 함수로 나타내는

2개의 매개변수가 있다. $\phi(t)$ 는 시간에 따른 바퀴의 회전각도를 나타내며 $\beta(t)$ 는 시간에 따른 AB 의 조향 각도를 나타낸다.

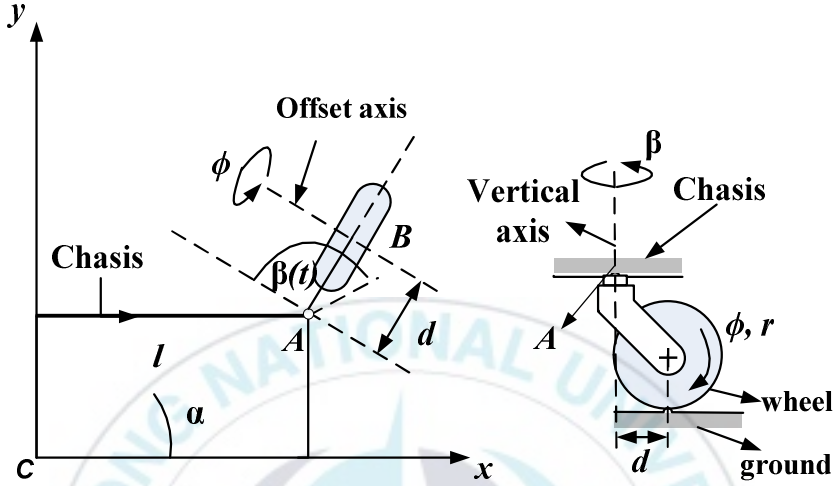


Fig. 4.3 A castor wheel and its parameters

수동 바퀴에서 오프셋 축(Offset axis)은 바퀴면에 나란하며 아무런 역할을 하지 않기 때문에, 회전 제약조건은 다음과 같다.

$$[\sin(\alpha + \beta) \quad \{-\cos(\alpha + \beta)\} \quad (-l)\cos\beta] \mathbf{R}(\theta_A) \dot{\xi}_1 - r\dot{\phi} = 0 \quad (4.6)$$

또한 미끄럼 제약조건은 다음과 같다.

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l\sin\beta] \mathbf{R}(\theta_A) \dot{\xi}_1 = 0 \quad (4.7)$$

4.2 차동 구동 AGV시스템의 운동학적 모델링

차동 구동 AGV 시스템을 모델링하기 위한 좌표계는 Fig. 4.4에 표시된다. AGV의 위치는 전역 고정 좌표계 XOY 에서의 전역 고정 좌표 (X_A, Y_A) 와 함께 전역 고정 좌표계 xCy 의 x 축으로 부터 이동 좌표계의 x 축 사이의 회전 방향각 θ_A 로 표시된다. AGV는 선속도 V_A 와 각속도 ω_A 로 움직인다.

AGV는 반지름 r 인 두 개의 표준 구동 바퀴 W_1 과 W_2 가 무인 운반차 이동 좌표계의 y 축의 위에 위치해 있다. 좌·우 바퀴의 선속도는 v_L 과 v_R 로 각각 표시되어 있다. 두 개의 구동 바퀴 사이의 점 C에서 각 바퀴까지의 거리는 b 로 나타낸다.

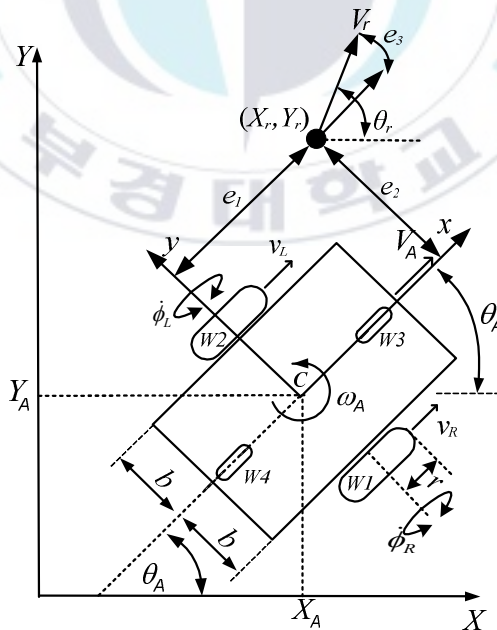


Fig. 4.4 Coordinate of differential drive AGV system

캐스터 바퀴 W_3 과 W_4 는 이동 좌표계 x 축 위에 위치하고 있다. 모든 방향을 자유롭게 움직이거나 힘을 주지 않을 때 캐스터 바퀴는 AGV의 운동식에 영향을 미치지 않으며, 운동학적 제약조건이 없다. 그러므로 오직 기존의 두 개의 표준 구동 바퀴 W_1 과 W_2 만이 무인 운반차의 운동식에 영향을 끼치며, 운동학적 제약조건이 고려되어야 한다.

Fig. 4.4의 우측 바퀴 W_1 의 매개변수는 $\alpha = -\pi/2$, $\beta = \pi$ 이고, 좌측 바퀴 W_2 의 매개변수는 $\alpha = -\pi/2$, $\beta = 0$ 이 된다. 따라서 이들 매개 변수들을 식 (4.4)에 적용하면, 식 (4.8)과 같이 표현된다.

$$\begin{bmatrix} \sin\left(-\frac{\pi}{2}+\pi\right) \left\{-\cos\left(-\frac{\pi}{2}+\pi\right)\right\} (-b)\cos\pi \\ \sin\left(\frac{\pi}{2}+0\right) \left\{-\cos\left(\frac{\pi}{2}+0\right)\right\} (-b)\cos 0 \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 - \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \dot{\Phi} = 0 \quad (4.8)$$

여기서 $\dot{\Phi} = [\dot{\phi}_r \ \dot{\phi}_l]^T$ 는 좌·우 표준 구동 바퀴 W_1 과 W_2 의 각속도로 구성된 표준 구동 바퀴의 각속도 벡터이며, 은 표준 구동 바퀴의 반지름이다. 식 (4.8)은 식 (4.9)와 같이 나타낼 수 있다.

$$\begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \dot{\Phi} \quad (4.9)$$

오른쪽 바퀴의 W_1 의 매개 변수는 $\alpha = -\pi/2$ 와 $\beta = \pi$ 와 같으므로 미끄럼 제약조건은 식 (4.10)과 같은 하나의 독립적 인식으로 나타낼 수 있다.

$$\begin{bmatrix} \cos\left(-\frac{\pi}{2} + \pi\right) & \sin\left(-\frac{\pi}{2} + \pi\right) & b \sin \pi \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 = 0 \quad (4.10)$$

식 (4.10)은 식 (4.11)과 같이 나타낼 수 있다.

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 = 0 \quad (4.11)$$

무인 운반차의 방정식은 표준 구동 바퀴의 회전제약조건 및 미끄럼 제약조건과 무인 운반차의 표준 구동 바퀴의 각속도 벡터 $\dot{\Phi}$ 사이의 관계는 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} \mathbf{J}_1(\beta) \\ \mathbf{C}_1(\beta) \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 = \begin{bmatrix} \mathbf{J}_2 \dot{\Phi} \\ 0 \end{bmatrix} \quad (4.12)$$

여기서, $\mathbf{J}_1(\beta) = \begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \end{bmatrix}$, $\mathbf{C}_1(\beta) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ 와 $\mathbf{J}_2 = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}$ 이다.

식 (4.9)와 식 (4.12)를 이용하여 모든 바퀴의 회전 제약조건은 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}(\theta_A) \dot{\xi}_1 = \begin{bmatrix} J_2 \dot{\Phi} \\ 0 \end{bmatrix} \quad (4.13)$$

식 (4.13)으로부터 얻은 AGV의 운동학적 방정식은 식 (4.14)와 식 (4.15)로 표현할 수 있다.

$$\begin{aligned} \dot{\xi}_1 &= \mathbf{R}(\theta_A)^{-1} \begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} J_2 \dot{\Phi} \\ 0 \end{bmatrix} \\ &= \mathbf{R}(\theta_A)^{-1} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2b} & -\frac{1}{2b} & 0 \end{bmatrix} \begin{bmatrix} J_2 \dot{\Phi} \\ 0 \end{bmatrix} \end{aligned} \quad (4.14)$$

$$\dot{\xi}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \mathbf{R}(\theta_A)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_R}{2} + \frac{r\dot{\phi}_L}{2} \\ 0 \\ \frac{r\dot{\phi}_R}{2b} + \frac{-r\dot{\phi}_L}{2b} \end{bmatrix} \quad (4.15)$$

여기서 $\mathbf{R}(\theta_A)^{-1} = \begin{bmatrix} \cos \theta_A & -\sin \theta_A & 0 \\ \sin \theta_A & \cos \theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 이다.

Y축 방향 AGV 선속도 $\dot{Y}_A$ 는 항상 0이므로 식(4.15)는 식 (4.16)와 같이 표현된다.

$$\dot{\xi}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta_A & \frac{r}{2} \cos \theta_A \\ \frac{r}{2} \sin \theta_A & \frac{r}{2} \sin \theta_A \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \Leftrightarrow \dot{\xi}_1 = \mathbf{S}(\xi_1) \dot{\Phi} \quad (4.16)$$

AGV의 속도와 좌·우 구동 바퀴의 각속도와의 관계식은 식 (4.17) 또는 식 (4.18)과 같이 표현될 수 있다.

$$\begin{bmatrix} V_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \Leftrightarrow \boldsymbol{\eta}_A = \mathbf{T} \dot{\Phi} \quad (4.17)$$

$$\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2b} & \frac{-r}{2b} \end{bmatrix}^{-1} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \dot{\Phi} = \mathbf{T}^{-1} \boldsymbol{\eta}_A \quad (4.18)$$

여기서 $\boldsymbol{\eta}_A = [V_A \ \omega_A]^T$ 는 속도 벡터이고, V_A 와 ω_A 는 각각 선속도 및 각속도이다.

따라서 AGV의 운동학적 방정식 식 (4.16)은 다음과 같이 표현될 수 있다.

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \cos \theta_A & 0 \\ \sin \theta_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \begin{aligned} \dot{\boldsymbol{\xi}}_1 &= \mathbf{J}(\boldsymbol{\xi}_1) \boldsymbol{\eta}_A \\ &= \mathbf{S}(\boldsymbol{\xi}_1) \mathbf{T}^{-1} \boldsymbol{\eta}_A \\ &= \mathbf{S}(\boldsymbol{\xi}_1)_A \dot{\boldsymbol{\Phi}} \end{aligned} \quad (4.19)$$



4.3 AGV 제어기 설계

본 절에서는 백스텝핑법을 사용하여 2장에서 제시된 전체 커버리지 경로계획법에 의해 생성된 AGV의 목표경로를 추적하는 비선형(Nonlinear) 경로 추적 제어기를 설계한다. AGV의 운동학적 방정식은 다음과 같다.

$$\dot{\xi}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \cos \theta_A & 0 \\ \sin \theta_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \quad (4.20)$$

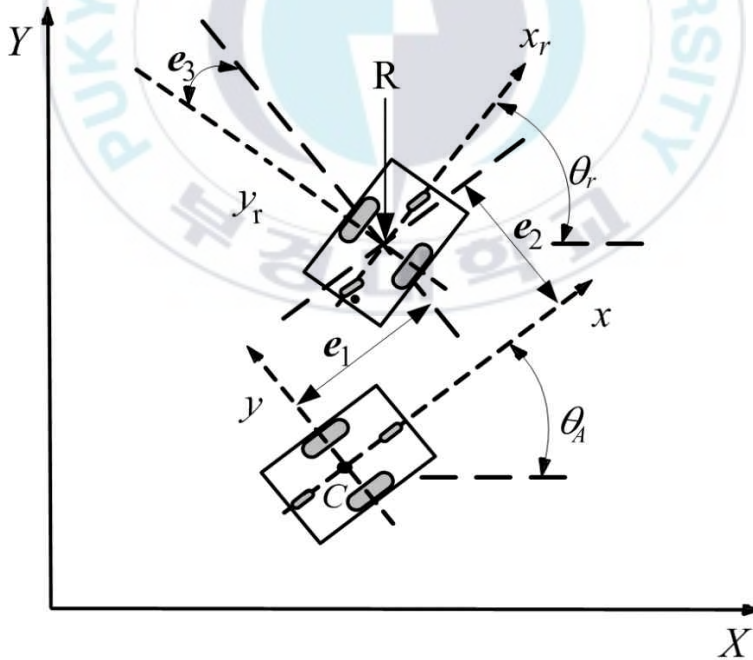


Fig. 4.5 Configuration of tracking error

AGV가 주어진 목표좌표 $(X_r(t), Y_r(t))$ 및 목표각도 $\theta_r(t)$ 를 추적하기 위한 추적오차 벡터(Tracking error vector)는 다음과 같이 정의된다. 식 (4.21)의 추적오차 벡터 $\mathbf{e}(t)$ 는 Fig. 4.5에 나타나있는 이동좌표계에서 추적오차 e_1, e_2 와 회전에 대한 추적오차 e_3 로 정의된다.

$$\mathbf{e}(t) = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta_A & \sin \theta_A & 0 \\ -\sin \theta_A & \cos \theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - X_A \\ Y_r - Y_A \\ \theta_r - \theta_A \end{bmatrix} \quad (4.21)$$

추적오차 벡터 $\mathbf{e}(t)$ 의 도함수는 식 (4.22)와 같이 나타내어진다.

$$\dot{\mathbf{e}}(t) = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \quad (4.22)$$

시스템의 안정성을 판별하기 위하여 Lyapunov function을 식 (4.23)과 같이 나타낸다. 식 (4.24)는 식 (4.23)의 도함수이다.

$$V_0 = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 + \frac{1}{2} e_3^2 \quad (4.23)$$

$$\begin{aligned} \dot{V}_0 &= e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 \\ &= e_1 (-V_A + V_r \cos e_3) + e_2 (V_r \sin e_3) + e_3 (-\omega_A + \omega_r) \end{aligned} \quad (4.24)$$

$\dot{V}_0 \leq 0$ 을 만족시키기 위해서, V_A 와 ω_A 가 식 (4.25)를 성립하도록 선택하면 식 (4.26)과 같은 제어칙(Control law)이 얻어진다.

$$\begin{cases} -V_A + V_r \cos e_3 = -k_1 e_1 & (a) \\ V_r \sin e_3 = -k_2 e_2 & (b) \\ \omega_r - \omega_A = -k_3 e_3 & (c) \end{cases} \quad (4.25)$$

$$\begin{cases} (a) & \rightarrow V_A = V_r \cos e_3 + k_1 e_1 \\ (b)+(c) & \rightarrow \omega_A = \omega_r + V_r \sin e_3 + k_2 e_2 + k_3 e_3 \end{cases} \quad (4.26)$$

식 (4.25)를 식 (4.23)에 대입하면 식 (4.27)이 성립된다.

$$\dot{V}_0 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 \leq 0, \quad (k_i > 0, \quad i=1,2,3) \quad (4.27)$$

식 (4.23), 식 (4.26), 식 (4.27)과 Barbalat's lemma에 의해 $t \rightarrow \infty$ 일 때 $e_1, e_2, e_3 \rightarrow 0$ 이다. 그러므로 식 (4.27)는 시스템을 점근적으로 안정케 한다.

Fig. 4.6은 본 연구에서 제안된 백스텝핑 제어기의 블록선도를 나타낸 것이다.

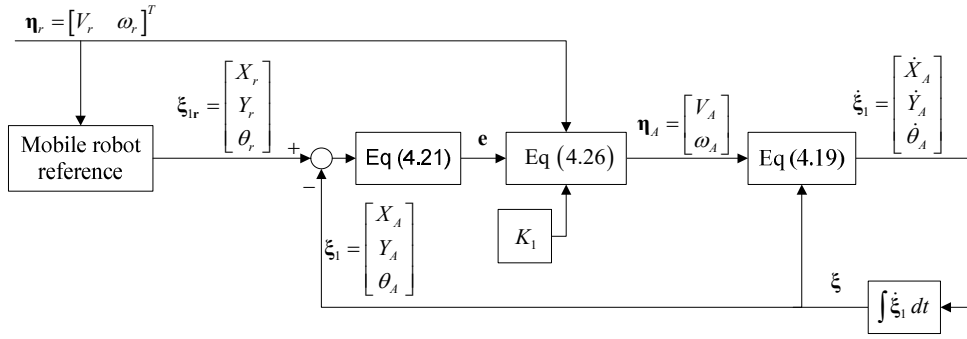


Fig. 4.6 Block diagram of the proposed backstepping nonlinear path tracking controller



제 5 장 시뮬레이션 및 실험 결과

본 장에서는 제 2장에서 계획된 전체 커버리지 경로를 제 3장에서 제안된 AGV 시스템을 이용하여 제 4장에서 제안된 경로 추적 제어기를 적용하여 AGV가 계획된 전체 커버리지 경로를 추적함을 검증하기 위해 시뮬레이션 및 실험 결과를 제시한다. 시뮬레이션 및 실험에 사용되는 전체 커버리지 경로는 Fig. 5.1과 같다.

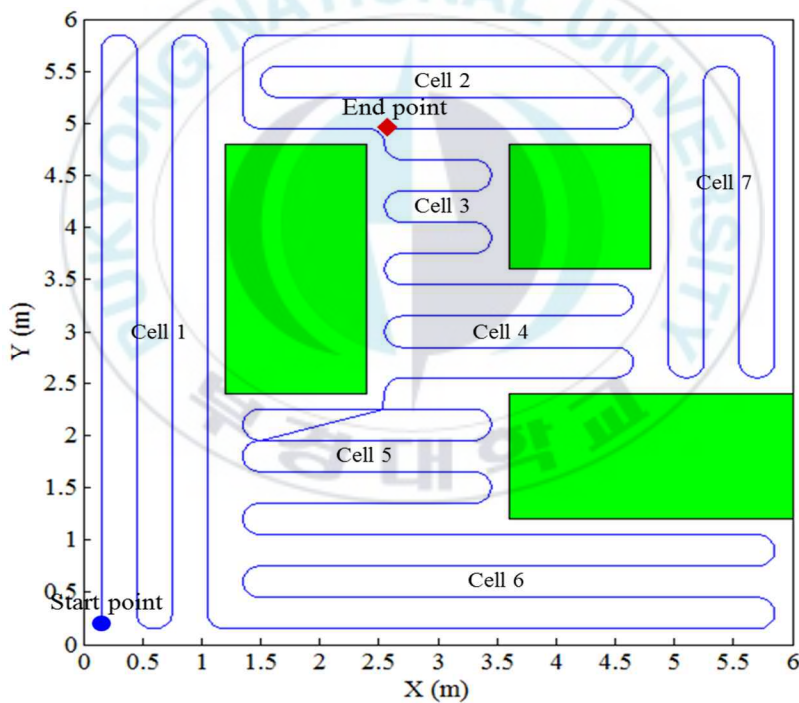


Fig. 5.1 Complete coverage path generated by the proposed complete coverage path planning method

5.1 시뮬레이션 결과

Table 5.1은 시뮬레이션에 사용된 초기값과 매개변수를 나타낸 것이다. AGV의 목표 선속도는 0.096m/s이며, AGV가 구동되는 시간은 약 930초이다.

Table 5.1 Parameter and initial values of the simulation

Parameter	Values	Unit
Effector width of robot	0.3	m
Coverage interval	0.3	m
Given area width	6	m
Given area length	6	m
Initial position of the AGV	(0.15, 0.15)	m
Initial angle of the AGV	$\pi / 2$	rad
Reference linear velocity of the AGV	0.096	m / s
Reference Angular velocity of the AGV	0	rad / s

Figs. 5.2~5.4는 계획된 전체 커버리지 경로를 추적할 때 발생하는 추적오차 e_1 , e_2 , e_3 의 시뮬레이션 결과를 나타낸 것이다. AGV가 계획된 경로를 잘 추적하고 있음을 확인할 수 있다. 셀 5에서 셀 4로 이동할 때 셀 간의 이동경로와 셀 3에서 셀 4로 이동할 때 셀 간의 이동경로는 AGV의 급격한 회전을 요구함으로써, 해당 경로 부분에 대하여 오차가 증

가함을 알 수 있다. 해당 경로를 지난 이후에 오차가 0으로 수렴함을 알 수 있다.

Fig. 5.2는 시간에 따른 추적오차 e_1 을 나타낸 것이며 직선구간에서는 추적오차가 0으로 수렴하고, 각 회전구간에서는 급격한 오차가 발생함을 보여준다. 회전구간의 추적오차는 $-11.33 \sim 7.646\text{mm}$ 범위에서 진동하며 추적하며, 최대 추적 오차는 46.79mm 이다. Fig. 5.2의 (a)~(h)는 Fig. 5.1의 전체 커버리지 경로에 해당하는 부분을 나타낸 것이며 Table 5.2에 설명하였다.

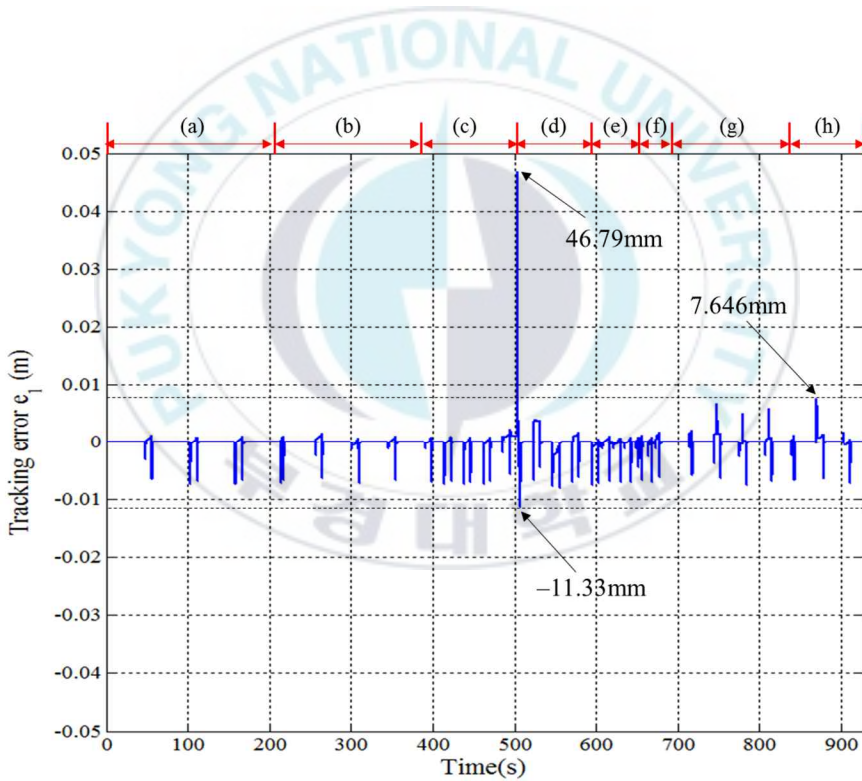


Fig. 5.2 Tracking error e_1

Table 5.2 Parameter and Description of each section in the simulation

Parameter	Description
(a)	Cell 1 coverage path
(b)	Moving path between Cell 1 to Cell 6 and coverage path for Cell 6
(c)	Moving path between Cell 6 to Cell 5 and coverage path for Cell 5
(d)	Moving path between Cell 5 to Cell 4 and coverage path for Cell 4
(e)	Moving path between Cell 4 to Cell 3 and coverage path for Cell 3
(f)	Moving path between Cell 3 to Cell 2 and outside coverage path of Cell 2 for going to Cell 7
(g)	Moving path between Cell 2 to Cell 7 and coverage path for Cell 7
(h)	Moving path between Cell 7 to Cell 2 and coverage path for the rest of Cell 2

Fig. 5.3은 시간에 따른 추적오차 e_2 를 나타낸 것이며, 추적오차 e_1 과 마찬가지로 셀 5에서 셀 4로 이동하는 경우 회전구간을 지날 때 추적오차는 대체로 $-12.0 \sim 11.82\text{mm}$ 범위에서 진동하며 추적하며, 최대 및 최소

추적 오차는 46.88mm 및 -71.11mm으로 급격한 추적오차의 변화가 발생했음을 보여주고 있다.

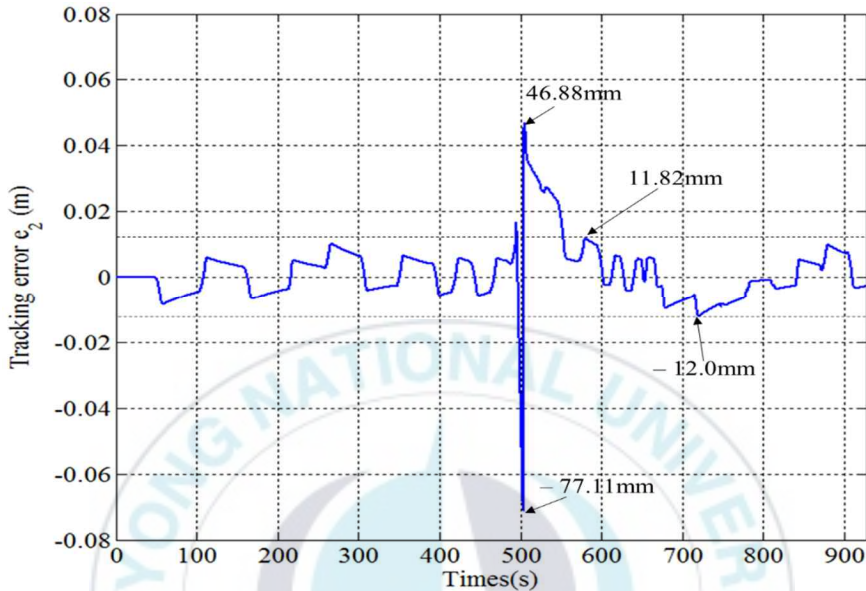


Fig. 5.3 Tracking error e_2

Fig. 5.4는 시간에 따른 AGV의 회전각 추적오차 e_3 를 나타낸 것이며, 회전구간을 지날 때 추적오차는 $-0.06464 \sim 0.03764$ rad 범위에서 진동하며 추적하며, 최대 및 최소 추적오차는 3.838 rad 및 -2.348 rad이다.

Fig. 5.5는 AGV의 선속도를 나타낸 것이며, 직선 경로에서 0.096 m/s로 주행하며 회전구간을 지날 때 0.02576 m/s까지 감속한 이후 다시 0.096 m/s의 속도로 복귀함을 알 수 있다. 셀 4에서 셀5로 이동할 때 최대 및 최소 추적오차 0.1631 m/s 및 -0.07365 m/s의 급격한 추적오차가 발생함을 알 수 있다.

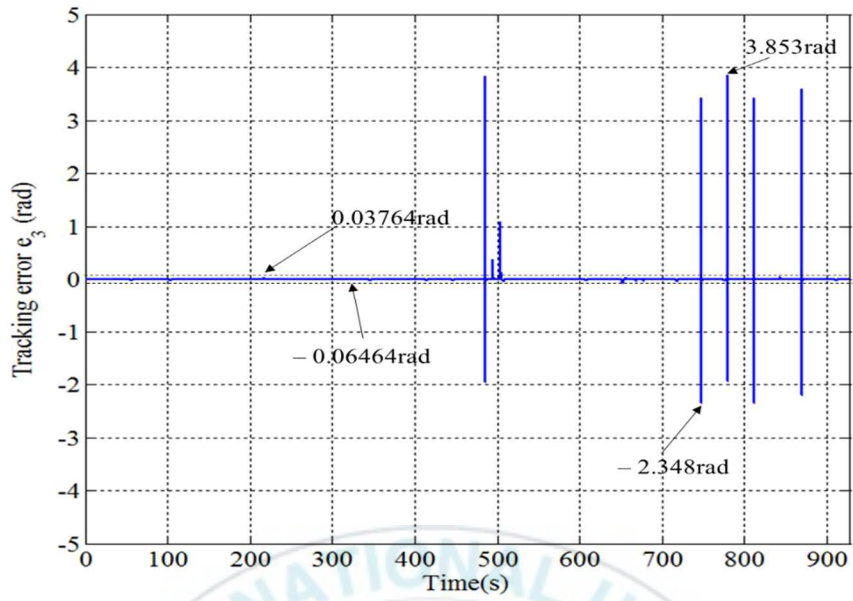


Fig. 5.4 Tracking error e_3

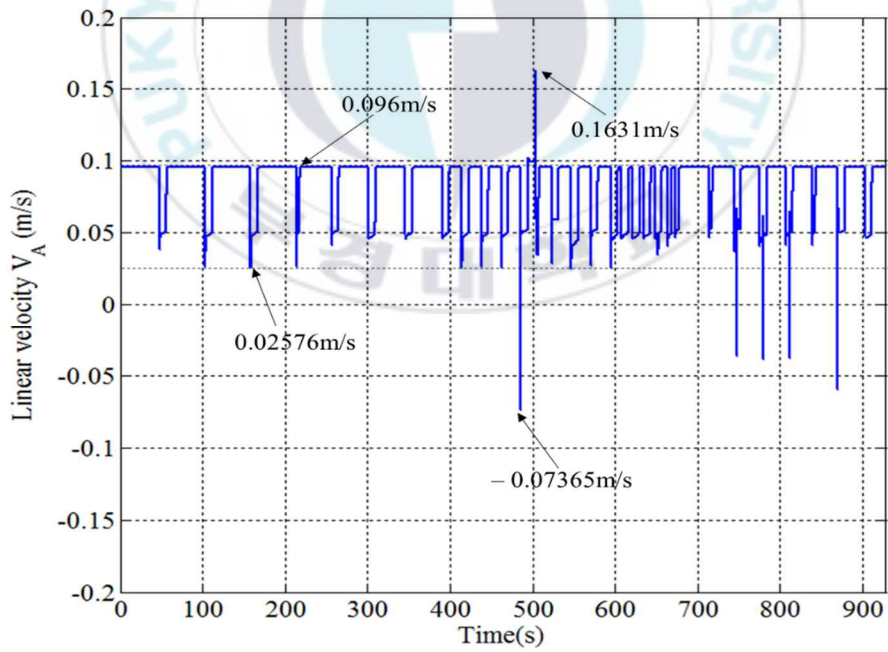


Fig. 5.5 Linear velocity of AGV

Fig. 5.6은 AGV의 회전각속도를 나타낸 것이며, 직선 경로에서 0.000929rad/s 로 거의 0rad/s 로 주행하며 회전구간을 지날 때 $-0.3868 \sim 0.3868\text{rad/s}$ 사이를 진동하며, 최대 및 최소 회전각속도는 2.319rad/s 및 -2.321rad/s 로 급격한 변화가 발생함을 알 수 있다.

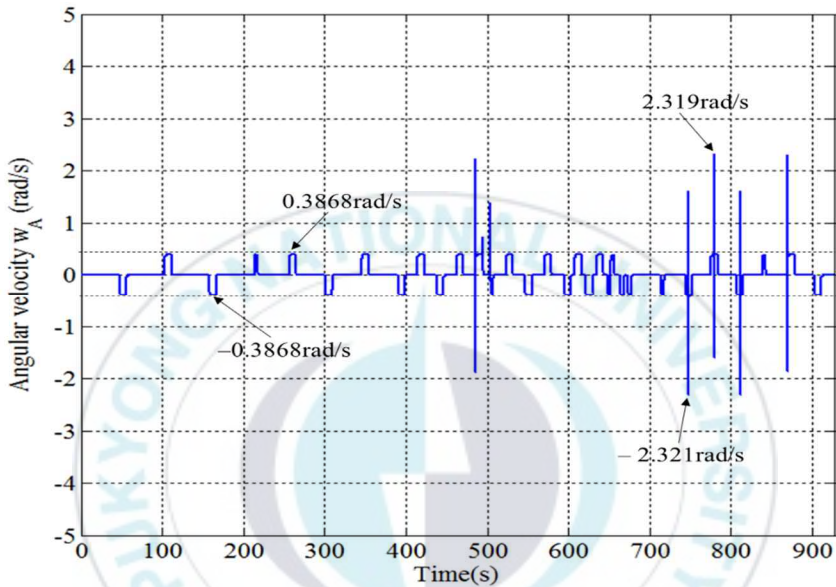


Fig. 5.6 Angular velocity of AGV

Fig. 5.7 계획된 전체 커버리지 경로와 이를 추적하는 AGV의 시뮬레이션 결과이다. AGV가 계획된 커버리지 경로를 잘 추적하고 있음을 확인할 수 있으며, 셀 5에서 셀 4로 이동하는 경우 셀 간의 이동 경로가 회전을 필요로 하는 구간이 많고 급격한 회전 때문에 경로를 조금 벗어나지만, 이후에 원래의 경로로 잘 추적함을 알 수 있다.

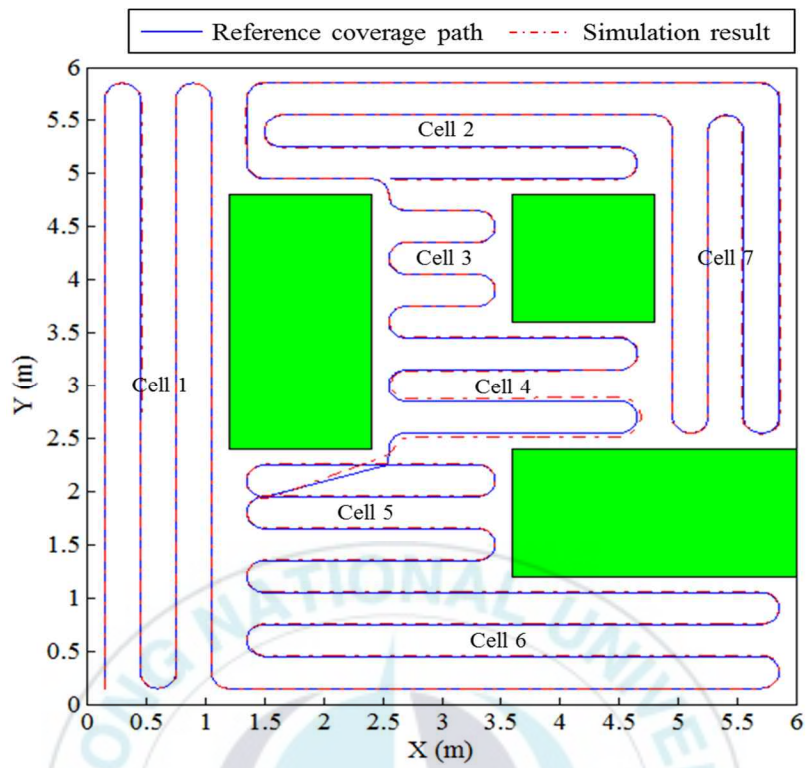
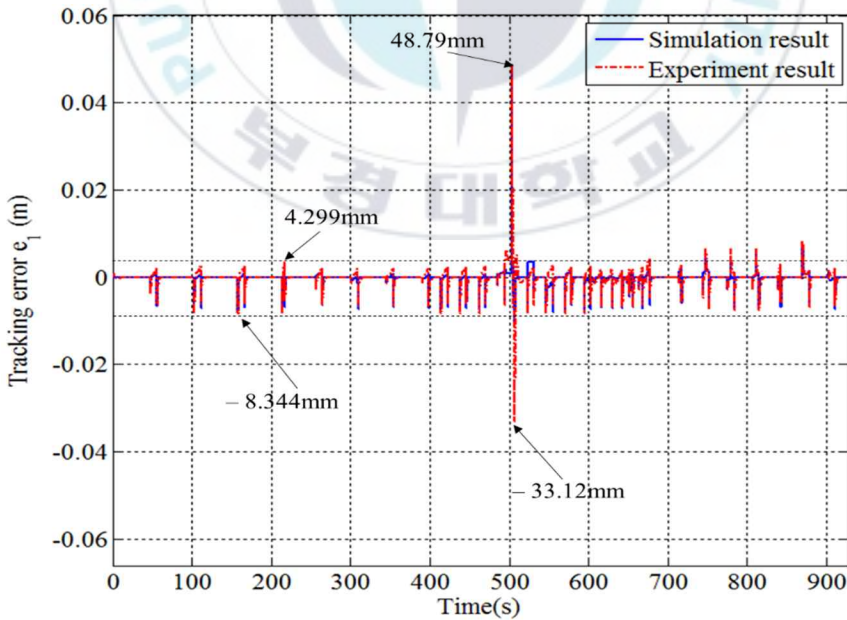


Fig. 5.7 Simulation result of planned coverage path tracking

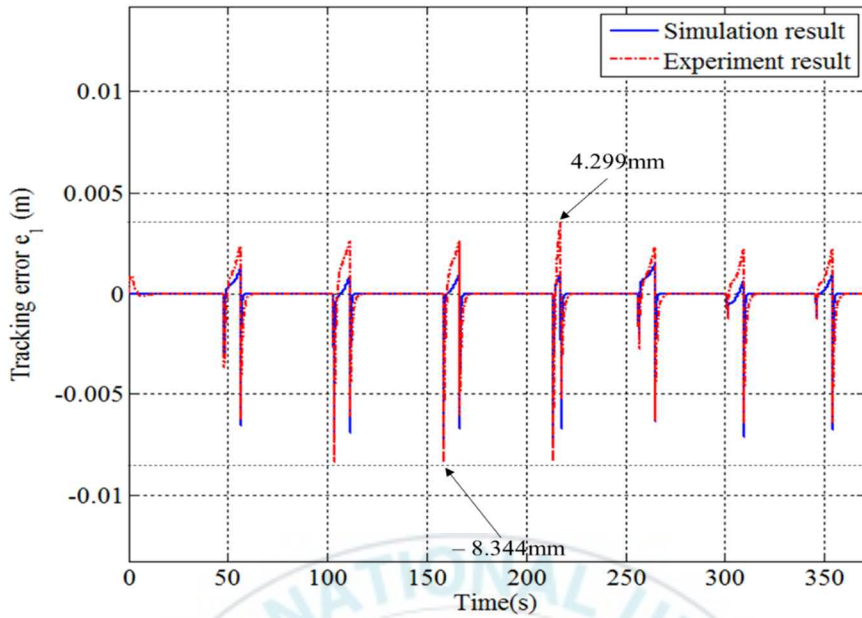
5.2 실험 결과

본 절에서는 AGV를 이용하여 제안된 경로 추적 제어기에 의한 제어 입력을 이용하여 계획된 전체 커버리지 경로를 추적함을 보이기 위해 시뮬레이션 결과에 대응하는 실험 결과를 제시한다.

Figs. 5.8~5.10은 AGV가 계획된 경로를 추적할 때 발생하는 추적오차의 실험 결과를 나타낸 것이다. Fig. 5.8은 추적오차 e_1 의 시뮬레이션 및 실험 결과를 나타낸다. 실험 결과 e_1 은 $-8.344\text{mm} \sim 4.299\text{mm}$ 범위에서 시뮬레이션 결과값의 주위를 진동하며 추적한다. 최대 및 최소 추적오차는 각각 48.79mm 와 -33.12mm 로 셀 5에서 셀 4로 이동하는 경우 발생한다. Fig. 5.9(b)는 Fig. 5.8(a)에서 $t=0\sim360$ 인 구간을 확대한 것이다.



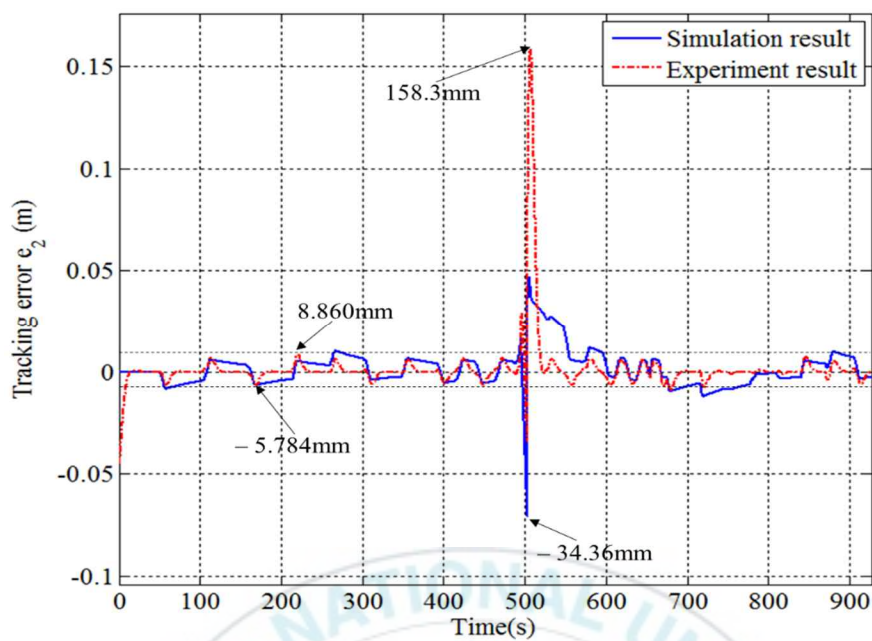
(a) Tracking error e_1 ($t = 0\sim930$)



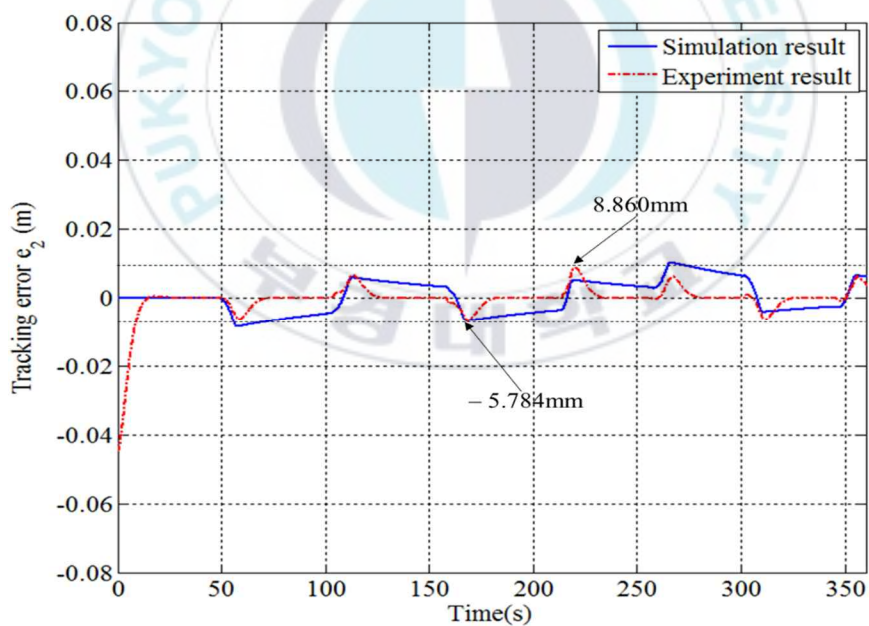
(b) Tracking error e_1 ($t = 0 \sim 360$)

Fig. 5.8 Tracking error e_1 of AGV

Fig. 5.9는 추적오차 e_2 의 시뮬레이션 및 실험 결과를 나타낸다. 실험 결과 e_2 는 $-5.784\text{mm} \sim 8.860\text{mm}$ 범위에서 시뮬레이션 결과값의 주위를 진동하며 추적하며, 최대 및 최소 추적오차는 각각 158.3mm 와 -34.36mm 로 셀 5에서 셀 4로 이동하는 경우 발생한다. Fig. 5.9(b)는 Fig. 5.9(a)에서 $t=0 \sim 360$ 인 구간을 확대한 것이다.



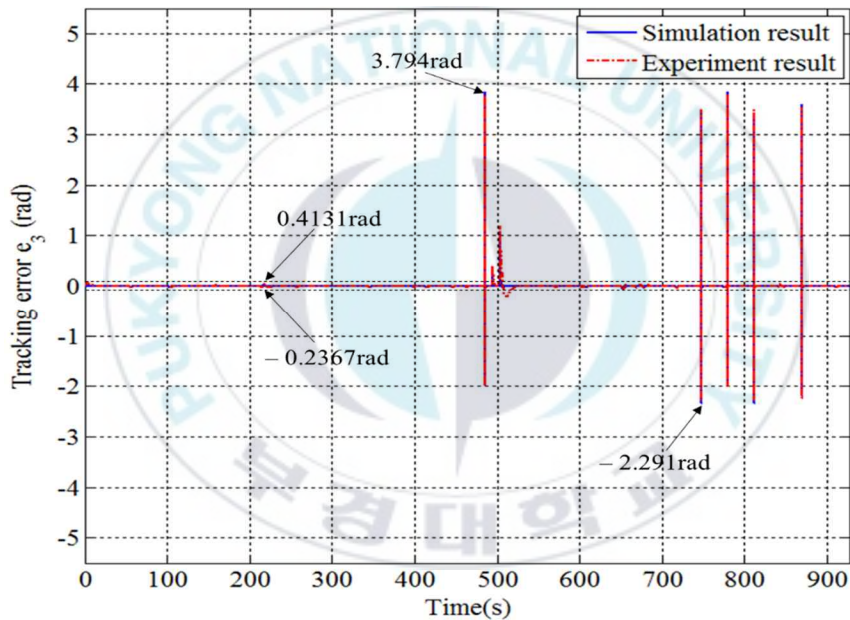
(a) Tracking error e_2 ($t = 0 \sim 930$)



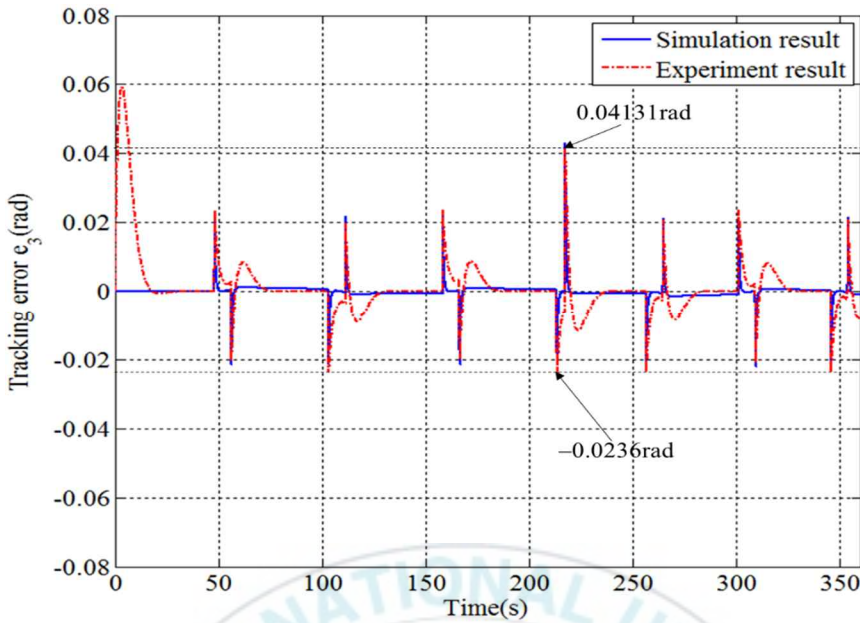
(b) Tracking error e_2 ($t = 0 \sim 360$)

Fig. 5.9 Tracking error e_2 of AGV

Fig. 5.10은 추적오차 e_3 의 시뮬레이션과 실험 결과를 나타낸다. 실험 결과 e_3 는 회전구간마다 $-0.0236\text{rad} \sim 0.04131\text{rad}$ 범위로 시뮬레이션 결과값 주위를 진동하며 추적하며, 직선구간에서는 큰 추적오차 없이 시뮬레이션 결과값을 추적한다. 최대 추적오차는 3.794rad 이며 셀 5에서 셀 4로 이동 시 발생하고, 최소 추적오차는 -2.291rad 이며 최소 셀 2에서 셀 7로 이동 시 발생한다. Fig. 5.10(b)는 Fig. 5.10(a)에서 $t=0\sim360$ 인 구간을 확대한 것이다.



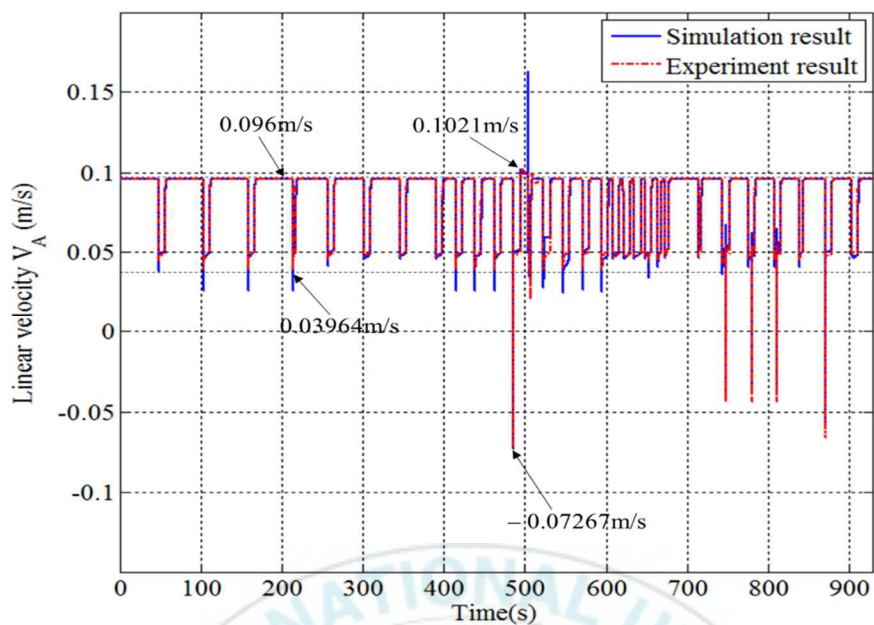
(a) Tracking error e_3 ($t = 0\sim930$)



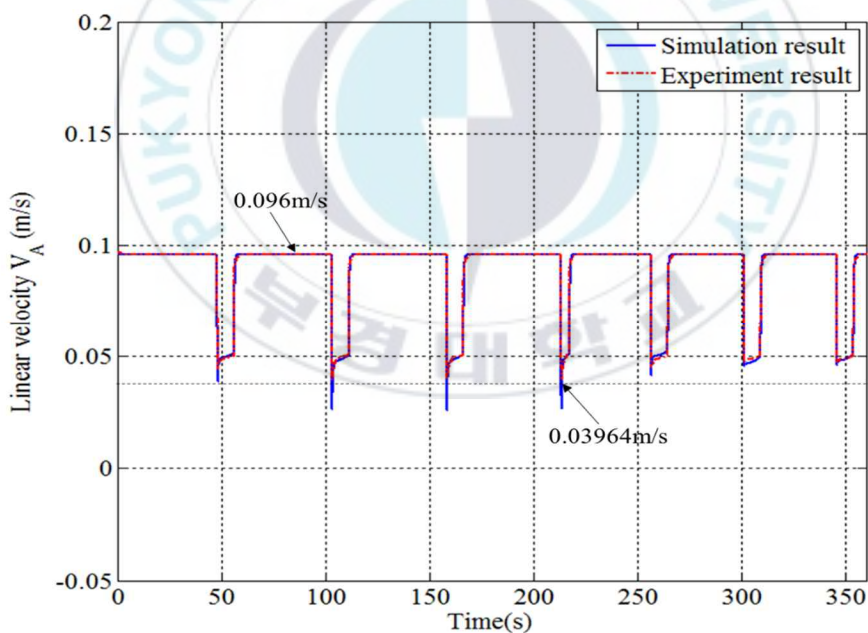
(b) Tracking error e_3 ($t = 0 \sim 360$)

Fig. 5.10 Tracking error e_3 of AGV

Fig. 5.11은 AGV의 선속도에 대한 시뮬레이션 및 실험 결과를 나타낸다. 모든 구간에서 실험 결과값은 시뮬레이션 결과값을 잘 추적하며, 직선구간에서는 0.096m/s로 주행하며, 회전구간의 경우 0.03964m/s까지 감속한 이후 제어입력에 의해 다시 시뮬레이션 값 0.096m/s를 잘 추적함을 보인다. 최대 및 최소 선속도는 0.1021mm/s 및 -0.07269m/s로 셀 4에서 셀 5로 이동할 때 급격한 오차가 발생함을 알 수 있다. Fig. 5.11(b)는 Fig. 5.11(a)에서 $t=0 \sim 360$ 인 구간을 확대한 것이다.



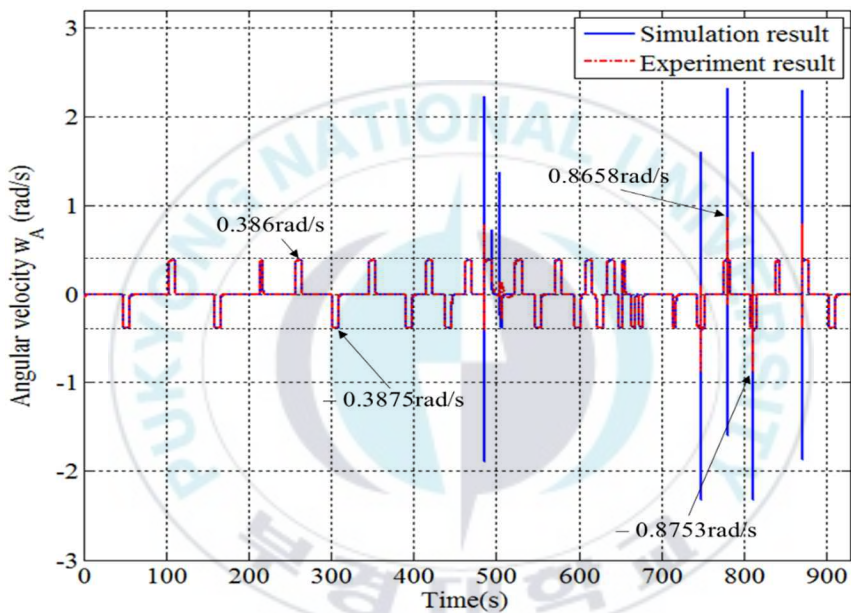
(a) Linear velocity($t = 0 \sim 930$)



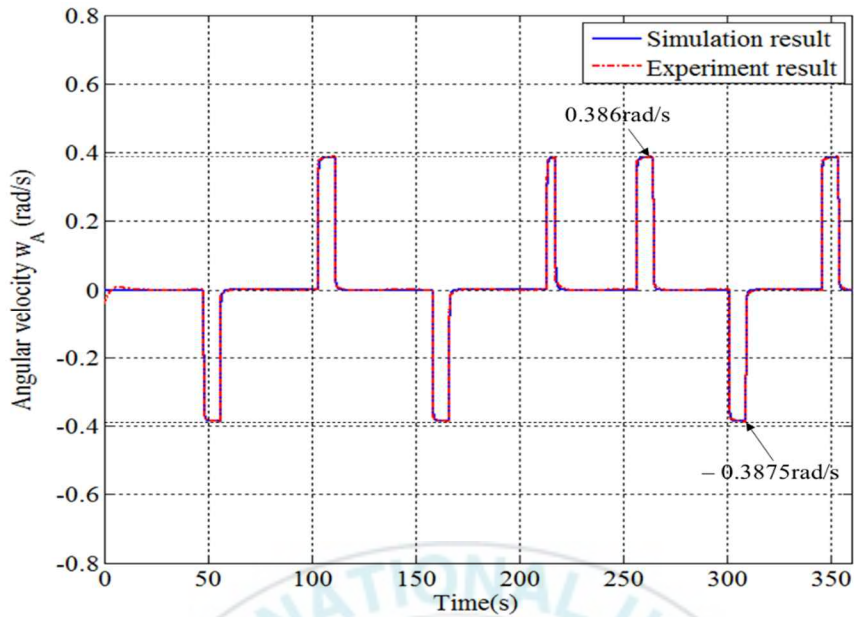
(b) Linear velocity($t = 0 \sim 360$)

Fig. 5.11 Linear velocity V_A of AGV

Fig. 5.12는 AGV의 각속도에 대한 시뮬레이션 및 실험 결과를 나타낸다. AGV는 직선구간에서는 0.000929rad/s 의 결과값으로 거의 0rad/s 를 가지며, 회전구간에서 $-0.3875 \sim 0.386\text{rad/s}$ 의 속도를 가진다. 셀 2에서 셀 7로 이동하는 경우 최대 및 최소 각속도 0.8658rad/s 및 -0.8753rad/s 의 오차가 발생하지만 제어입력에 의해 다시 시뮬레이션 값을 잘 추적함을 보인다. Fig. 5.12(b)는 Fig. 5.12(a)에서 $t=0\sim360$ 인 구간을 확대한 것이다.



(a) Angular velocity($t = 0\sim930$)



(b) Angular velocity ($t = 0 \sim 360$)

Fig. 5.12 Angular velocity of AGV

Fig. 5.13은 AGV의 계획된 전체 커버리지 경로를 추적하는 시뮬레이션 및 실험 결과를 나타낸다. 실험 결과는 AGV의 초기위치에서 목표경로로 수렴하는 결과를 보이며, 초기위치에서 처음 직선구간과 초기의 회전구간은 AGV가 잘 추적함을 보인다. 셀 5에서 셀 4로 이동하는 경우 셀 간의 이동경로가 회전을 필요로 하는 구간이 많고 급격한 회전으로 인해 시뮬레이션 경로를 조금 벗어나지만 이후에 시뮬레이션 경로를 추적하려는 것을 알 수 있다.

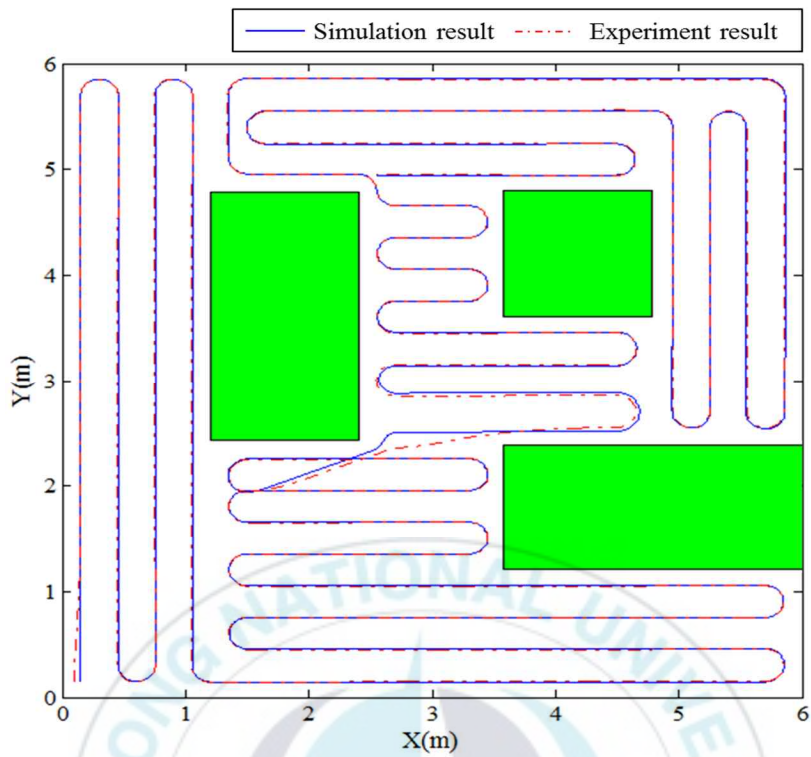


Fig. 5.13 Simulation and Experiment result of proposed coverage path tracking

제 6 장 결론 및 향후 연구 방향

6.1 결론

본 연구에서는 집광로봇의 심해저 망간단괴 채광을 위한 전체 커버리지 경로계획법을 제안하고, 제안된 경로계획법에 의해 계획된 경로를 추적하기 위한 집광로봇과 유사한 동작성을 가진 무인 운반차 시스템의 구성 및 제어 알고리즘을 제안하였다. 또한 주어진 간단한 영역에 대한 전체 커버리지 경로계획법 및 제어 알고리즘을 구현하기 위해서 이룬 무인 운반차를 이용하여 시뮬레이션 및 실험을 통해 그 유효성을 검증하였다. 본 연구의 결론을 요약하면 다음과 같다.

- 제 2장에서는 효율적인 커버리지 수행을 위해 주어진 3D 영역을 간단한 2D 격자 형태의 영역으로 변환하여 채광 영역과 장애물 영역으로 구분하고, 구분된 영역을 모스 셀 분할법에 의해 셀로 분할하였다. 셀 분할 최적화를 실행함으로써 집광로봇의 회전수를 최소화하기 위해 최소 고도 계산 알고리즘을 적용하였다. 주어진 영역이 셀로 분할되면 각 셀의 커버리지 순서를 결정해야 하므로 셀 간의 인접한 관계를 바탕으로 신장트리를 생성하고, 너비우선 탐색 알고리즘 및 깊이우선 탐색 알고리즘을 신장트리에 적용하여 셀 커버리지 순서를 결정하였다. 셀 커버리지 순서가 결정되면 이전 셀의 커버리지 완료점에서 다음 셀의 커버리지 시작점으로 이어지는 이동경로가 필요하므로 가시성 그래프 알고리즘을 이용하여 셀 간의 이동경로를 생성하였다.

셀 간의 이동경로가 이전 셀의 커버리지 경로만큼 중복되는 경로인 경우 수정 스위핑 라인법을 이용하여 충족되는 조건하에 셀 간의 이동경로를 수정함과 동시에 커버리지 경로를 수정하여 중복되는 경로를 최소화하였다. 전체 커버리지가 생성되는 과정에서 집광로봇의 연속적인 경로를 위해 회전구간의 경로가 원형이 되도록 경로를 수정하였다. 너비우선 탐색 알고리즘과 깊이우선 탐색 알고리즘으로 생성된 셀 커버리지 순서에 기반한 전체 커버리지 경로를 시뮬레이션을 통해 생성하여 비교하였을 때, 깊이우선 탐색 알고리즘의 셀 커버리지 순서에 의해 생성된 전체 커버리지 경로가 너비우선 탐색 알고리즘의 셀 커버리지 순서에 의해 생성된 전체 커버리지 경로보다 전체 커버리지 경로의 길이는 6.983m 더 짧고, 중복되는 경로는 9.781m가 더 적으며, 커버리지 시간은 349.15초만큼 더 빠르고, 커버리지 효율도 95.13%로 3%더 높음을 알 수 있었다.

- 제 3장에서는 제 2장에서 생성되는 전체 커버리지 경로를 추적하는 실험을 수행하기 위한 무인 운반차에 대한 시스템을 구성하였다. 무인 운반차는 기구시스템과 제어시스템으로 구성되어 있으며, 제어시스템은 센서부, 제어부, 조작부 및 전원 공급장치로 구성된다. 무인 운반차는 가로 60cm, 세로 120cm, 높이 56.5cm로 차동 구동 시스템을 사용하였고 본체의 좌·우 중간에 차동형 구동 바퀴를 BLDC모터로, 전·후로는 균형을 맞추기 위한 두 개의 수동 바퀴 캐스터 바퀴를 사용하였다. 위치인식을 위하여 레이저 항법 시스템 NAV-200과 가속도 센서 MySen-M을 사용하였고 전원 공급장치로 12V의 축전지 2개를 사용하였으며, 터치스크린과 모니터 및 산업용 컴퓨터 TANK-800을 이용하여 제어부를 구성하였다.

■ 제 4장에서는 무인 운반차의 운동학적 모델링 및 경로 추적 제어기를 제시하였다. 무한궤도를 이용한 집광로봇과 이륜 이동로봇인 무인 운반차 간의 모델링과 동작성이 유사하므로, 무인 운반차의 모델링을 제시하였다. 시스템 모델링을 위해 전역 고정 좌표계에서 무인 운반차의 위치와 방향각으로 구성된 위치벡터를 정의하고, 전역 고정 좌표계에서 무인 운반차 위치벡터 및 이동 좌표계에서의 무인 운반차 위치벡터 사이의 관계를 나타내었다. 또한 운동학적 모델링을 위해 표준 구동 바퀴의 회전 제약조건 및 미끄럼 제약조건과 무인 운반차의 표준구동 바퀴의 각속도 벡터를 통해 무인 운반차의 방정식을 도출하였고, 선속도 및 좌·우 구동 바퀴 각속도를 이용하여 정리하였다. 그리고 무인 운반차가 목표 경로를 추적하기 위한 백스텝핑 비선형 경로 추적 제어기를 설계하였으며, Lypunov function을 이용하여 시스템의 안정성을 판별하였다.

■ 제 5장에서는 제 2장에서 생성되는 전체 커버리지 경로를 제 3장과 제 4장에서 제시된 무인 운반차 및 경로 추적 제어기를 이용하여 추적하는 시뮬레이션 및 실험 결과를 제시하였다. 시뮬레이션 결과 추적오차 e_1 , e_2 , e_3 를 통해 무인 운반차가 계획된 경로를 잘 추적하고 있음을 확인할 수 있었다. 셀 5에서 셀 4로 이동할 때 해당 경로에서 오차가 증가하지만, 해당 경로를 지난 후 오차가 0으로 수렴함을 알 수 있었다. 선속도 및 각속도에 대한 시뮬레이션 결과에서 선속도는 입력 속도 0.096m/s로 이동하며 회전구간에서 0.02576m/s로 감속하는 것을 알 수 있었고, 각속도는 직선 경로에서 거의 0에 가깝고 회전구간을 지날 때 $-0.3868 \sim 0.3868\text{rad/s}$ 사이를 진동함을 알 수 있었다.

실험 결과에서 추적오차 e_1 은 $-8.344 \sim 4.299\text{mm}$, 추적오차 e_2 는 $-5.784 \sim$

8.860mm, 추적오차 e_3 는 $-0.0236 \sim 0.04131\text{rad}$ 의 범위로 진동하며 시뮬레이션 결과값 주위를 추적함을 알 수 있었다. 또한 시뮬레이션 결과와 같이 셀 5에서 셀 4로 이동하는 경우 추적오차 e_1 은 48.79mm, 추적오차 e_2 는 158.3mm, 추적오차 e_3 는 3.794rad로 각각의 오차에 대한 최대 추적오차가 발생함을 알 수 있었다. 실험 결과 선속도는 회전구간에서 $0.3964 \sim 0.096\text{m/s}$ 의 값을 가지며 셀 4에서 셀 5로 이동하는 경우 0.1021m/s 및 -0.07267m/s 의 값으로 큰 오차가 발생하지만, 제어입력에 의해 다시 시뮬레이션 값을 잘 추적함을 보인다. 실험 결과 각 속도는 직선으로 움직일 경우 0.000929rad/s 의 결과값을 가지며, 회전구간에서 $-0.3875 \sim 0.386\text{rad/s}$ 의 값을 가지며 셀 2에서 셀 7로 이동하는 경우 0.8658rad/s 및 -0.8753rad/s 의 값으로 큰 오차가 발생하지만 제어입력에 의해 다시 시뮬레이션 값을 잘 추적함을 알 수 있었다.

그러므로 본 연구의 결과인 커버리지 경로 계획과 경로 추적 제어기는 수중집광로봇의 개발 시 기초 학문으로 적용될 수 있을 것이다.

6.2 향후 연구 방향

본 연구에 대한 향후 계획은 다음과 같다.

- 무한궤도 차량을 이용한 수중용 소형 집광로봇 설계
- 수중 심해저 형태와 유사한 플랫폼 제작 및 수중 커버리지 경로 계획 연구
- 미끄러짐이 발생하기 쉬운 기반에 대한 커버리지 경로 계획 연구
- 수중용 소형 집광로봇의 고장진단에 대한 연구

감사의 글

본 논문이 완성되기까지 아낌없는 조언과 격려로 지도하여 주신 김상봉 교수님께 감사드리며, 고난과 역경 속에 기회가 있음을 가르쳐 주신 은혜를 잊지 않고 살도록 하겠습니다. 논문 심사과정에서 충고와 조언을 주신 정영석 교수님과 변기식 교수님께 깊은 감사의 말씀을 드립니다. 그리고 부족한 논문을 완성하는 데 많은 도움을 주시고, 학문적 지식을 가르쳐 주신 김학경 박사님께 진심으로 감사의 말씀을 드립니다. 항상 어머니의 마음으로 연구실 인원들을 아껴 주시고 사랑하여 주신 사모님께도 감사의 말씀을 드립니다.

석사과정 중 연구활동에 많은 도움 및 조언을 주신 한국해양과학기술원 선박해양플랜트연구소의 홍섭 박사님, 김형우 박사님, 여태경 박사님, 민천홍 박사님, 전봉환 박사님, 그리고 윤석민 선배님과 김진호 선배님, 이경목 선배님께 감사의 말씀을 드립니다.

CIMEC Lab에서 6년 남짓 생활하면서 많은 도움을 주셨던 선배님들 및 후배님들에게 감사의 말씀을 드리며, 특히 실험실에서 오랫동안 함께 생활한 김서광, 박유미, 배민지, 김정근, 정우영, 민중후, Pandu Sandi Pratama, Chetanraj D. Patil, Doan Phuc Thinh, Hoang Giang, Bui Thanh Luan, Duong Van Tu 선배님들께 진심으로 감사드립니다. 석사과정을 함께 보내면서 많은 도움과 힘이 되어준 이미 석사 학위를 받은 Yuhanes Dedy Setiawan, Gulalkari Amruta Vinod와 앞으로 헤쳐나가야 할 것이 많은 박사과정 Trong Hai Nguyen,

Nguyen Huy Hung, 석사과정 정재훈, Sheng DongBo, Gao Tianshui 및 학부생 김진일, 이정웅, 후배님에게도 감사의 말씀을 드립니다. 또한, 연구실 내에서 궂은 일을 맡아 주시고 격려하여 주시며 정신적인 도움을 주신 김현 선생님께 감사의 말씀을 드립니다.

늘 주위에서 희로애락을 함께하며 힘이 되어 주고 정신적으로 큰 도움을 준 신슬기, 류술, 권유창, 공덕희, 김효민, 백진호, 방대혁, 박세하, 박재민, 김두민, 김원준, 고경민, 윤성재, 송호준에게 감사의 말씀을 드립니다.

마지막으로 못난 아들 믿고 묵묵히 지켜봐 주시는 아버지와 항상 사랑으로 보살펴 주시고 아껴 주시는 어머니에게 진심으로 깊은 감사의 마음을 전합니다.

2015년 12월

김 진 욱

참 고 문 헌

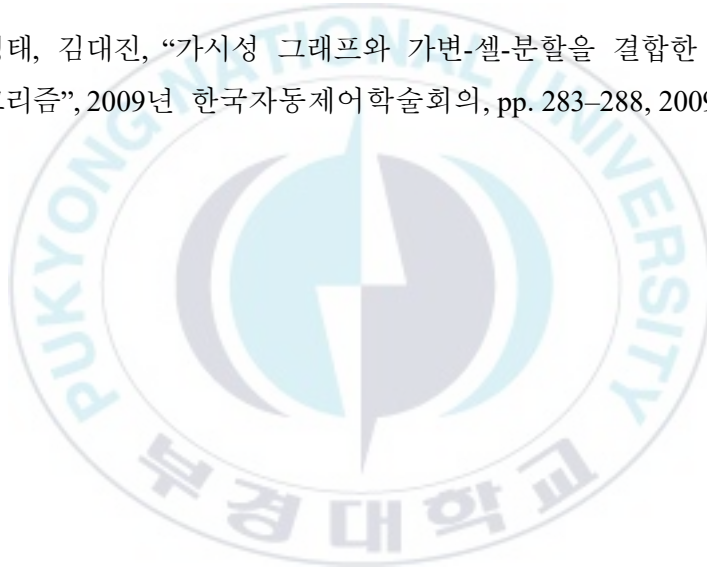
- [1] S. J. Park, T. K. Yeu, S. M. Yoon, S. Hong, and K. Y. Sung, “A Study of Sweeping Coverage Path Planning Method for Deep-sea Manganese Nodule Mining Robot”, Proceeding of Oceans 2011 Conference, pp. 1–5, 2011.
- [2] P. M. Hsu and C. L. Lin, “Optimal Planner for Lawn Mowers”, 2010 IEEE 9th International Conference on Cybernetic Intelligent Systems(CIS 2010), pp. 1–7, 2010.
- [3] A. Barrientos, J. Colorado, J. del Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, “Aerial Remote Sensing in Agriculture: A Practical Approach to Area Coverage and Path Planning for Fleets of Mini Aerial Robots”, Journal of Field Robotics, vol. 28, pp. 667–689, 2011.
- [4] E. Santamaria, F. Segor, I. Tchouchenkov, and R. Schönbein, “Path planning for Rapid Aerial Mapping with Unmanned Aircraft Systems a Fast Complete Coverage Path Planning Algorithm with No Backtracking”, The Eighth International Conference on Systems(ICONs2013), pp. 82–87, 2013.
- [5] C. Fang and S. Anstee, “Coverage Path Planning for Harbour Seabed Surveys Using an Autonomous Underwater Vehicle”, OCEANS 2012 IEEE-Sydney, pp. 1–8, 2010.

- [6] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path Planning for Robotic Demining: Robust Sensor-Based Coverage of Unstructured Environments and Probabilistic Methods", *The International Journal of Robotics Research*, vol. 22, no. 7–8, pp. 441–466, 2003.
- [7] Z. Yao and S. K. Gupta, "Cutter Path Generation for 2.5D Milling by Combining Multiple Different Cutter Path Patterns", *The International Journal of Production Research*, vol. 42, no. 11, pp. 2141–2161, 2004.
- [8] Z. Yao, S. K. Gupta, and D. S. Nau, "A Geometric Algorithm for Finding the Largest Milling Cutter," *Journal of Manufact Process* 2001, vol. 3, pp. 1–16, 2001.
- [9] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, "Principles of Robot Motion: Theory, Algorithms, and Implementations", *Intelligent Robotics and Autonomous Agents*, vol. 12, no. 3, pp. 110–110, 2005.
- [10] J. Jung, R. Jayakrishnan, and D. Nam, "High coverage point-to-point transit: Local vehicle routing problem with genetic algorithms", *14th International IEEE Conference on Intelligent Transportation Systems(ITSC 2011)*, pp. 1285–1290, 2011.
- [11] P. M. Hsu, C. L. Lin, and M. Y. Yang, "On the Complete Coverage Path Planning for Mobile Robots", *Journal of Intellgent and Robotic System*, vol. 74, no. 3, pp. 945–963, 2014.

- [12] Y. M. Y. Mao, L. D. L. Dou, J. C. J. Chen, H. F. H. Fang, H. Z. H. Zhang, and H. C. H. Cao, "Combined Complete Coverage Path Planning for Autonomous Mobile Robot in Indoor Environment", Proceedings of the 7th Asian Control Conference, pp. 1468–1473, 2009,.
- [13] G. Hoang, H. K. Kim, and S. B. Kim, "Path Tracking Controller of Quadraped Robot for Obstacle Avoidance Using Potential Functions Method", International Journal of Science and Engineering, vol. 4, no.1, pp. 1–5, 2013.
- [14] D. H. Kim, G. Hoang, M. J. Bae, J. W. Kim, S. M. Yoon, T. K. Yeo, H. Sup, and S. B. Kim, "Path Tracking Control Coverage of a Mining Robot Based on Exhaustive Path Planning with Exact Cell Decomposition," 14th International Conference on Control, Automation and Systems (ICCAS 2014), pp. 730–735, 2014.
- [15] M. Bae, H. Sup, T. Yeu, and S. Kim, "Exhaustive Path Planning of Coverage of A Mining Robot with Robot Exact Cell Decomposition", Proceedings of 2012 International Symposium of Advanced Mechanical and Power Engineering (ISAMPE 2012), pp. 11–15, 2012.
- [16] A. Zelinsky, "A Mobile Robot Exploration Algorithm" IEEE Transactions on Robotics and Automation, vol. 8, no. 6, pp. 707–717, 1992.
- [17] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition", Autonomous Robots, vol. 9, pp. 247–253, 2000.

- [18] Y. D. Setiawan, P. S. Pratama, J. W. Kim, and D. H. Kim, “Path Replanning and Controller Design for Trajectory Tracking of Automated Guided Vehicles”, 2014 International Conference on Advances in Computing, Communications and Informatics(ICACCI 2014), pp. 771–777, 2014.
- [19] 서진호, T. H. Bui, T. T. Guyen, 김상봉, “용접용 이륜 이동로봇의 모델링 및 적응 추종 제어”, 2003년도 대한기계학회 춘계학술대회, pp. 786–791, 2003.
- [20] R. Fierro and F. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics”, Proceedings of 34th Conference on Decision and Control, vol. 4, pp. 3805–3810, 1995.
- [21] 김시중, 강정원, 정명진, “효율적 커버리지 경로 계획 및 동적 환경에서의 경로 주행”, 한국로봇공학회, 제 2권, pp. 304–309, 2007.
- [22] W. H. Huang, “Optimal Line-Sweep-Based Decompositions for Coverage Algorithms”, Proceedings of IEEE International Conference on Robotics and Automation, vol. 1, pp. 27–32, 2001.
- [23] Z. Yao, “Finding Efficient Robot Path for the Complete Coverage of A Known Space”, Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3369–3374, 2006.

- [24] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks", *The International Journal of Robotics Research*, vol. 21, no.4, pp. 331–344, 2002.
- [25] Y. D. Setiawan, P. S. Pratama, D. H. Kim, S. M. Yoon, T. K. Yeu, S. Hong, H. K. Kim, and S. B. Kim, "Path Planning Based on A* Algorithm and Its Tracking Controller for Underwater Mining Robots", 2013 한국해양공학회 추계학술대회, pp. 2–5, 2013.
- [26] 김정태, 김대진, "가시성 그래프와 가변-셀-분할을 결합한 경로 계획 알고리즘", 2009년 한국자동제어학술회의, pp. 283–288, 2009.



학술 논문 발표 및 게재

- [1] Jin Wook Kim, Min Ji Bae, Suk Min Yoon, Tae Kyeong Yeu, Jin Ho Kim, Sub Hong, Hak Kyeong Kim and Sang Bong Kim, “무인 운반차의 최적 경로 설계 및 경로 추종”, 한국해양공학회 2013년도 추계학술대회(KSOE 2013), pp. 141-144, 2013.

- [2] Jin Wook Kim, Pandu Sandi Pratama, Suk Min Yoon, Tae Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, “Coverage Path Planning Using Modified Sweeping Line Strategy for an Underwater Mining Robot”, 한국해양공학회 2015년도 추계학술대회, pp. 82-85, 2015.

- [3] Giang Hoang, Jung Hu Min, Jin Wook Kim, Hak Kyeong Kim, Yeon Wook Choe and Sang Bong Kim, “Control of Inpipe Inspection Robot in Variable Pipeline”, Proceedings of the 2013 International Symposium on Mechatronics and Robotics, pp. 55-60, 2013.

- [4] Yuhanes Dedy Setiawan, Pandu Sandi Pratama, Jin Wook Kim, Dae Hwan Kim, Young Seok Jung, Sang Bong Kim “Path Replanning and Controller Design for Trajectory Tracking of Automated Guided Vehicles”, 2014 International Conference on Advances in Computing, Communications and Informatics(ICACCI), pp. 771-777, 2014.

- [5] Yuhanes Dedy Setiawan, Jin Wook Kim, Jun Hu Min, Hak Kyeong Kim and Sang Bong Kim, "Pipe Inspection Robot Controller Design Based on Linear Quadratic Regulator", The 7th Vietnam Conference on Mechatronics (VCM 2014), pp. 493-500, 2014.
- [6] Tran Quang Tuyen, Jin Wook Kim, Woo Young Jeong, Gyeong Mok Lee, Hak Kyeong Kim, Bong Huan Jun and Sang Bong kim, "Locomotion Control of Hexapod Robot Based on Inverse Kinematic Algorithm", International Symposium on Advanced Mechanical and Power Engineering (ISAMPE 2014), pp. 114-121, 2014.
- [7] Pandu Sandi Pratama, Jin Wook Kim, Suk Min Yoon, Tae Kyeong Yeu, Sup Hong and Sang Bong Kim, "Path Planning Method and Trajectory Tracking Controller for Coverage Uneven Terrain Structures of the Ocean Floor", International Symposium on Advanced Mechanical and Power Engineering (ISAMPE 2014), pp. 122-132, 2014.
- [8] Pandu Sandi Pratama, Jin-Wook Kim, Jung-Hu Min, Suk-Min Yoon, Tae-Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, "Path Planning Method for Coverage of Uneven Terrain Environment Using Morse Cell Decomposition", 한국해양공학회 2014년도 추계학술대회, pp. 151-154, 2014.

- [9] Dae Hwan Kim, Giang Hoang, Min Ji Bae, Jin Wook Kim, Suk Min Yoon, Tae Kyeong Yeo, Hong Sup and Sang Bong Kim, "Path Tracking Control Coverage of a Mining Robot Based on Exhaustive Path Planning with Exact Cell Decomposition", 14th International Conference on Control, Automation and Systems (ICCAS 2014), pp. 730-735, 2014.
- [10] Amruta Vinod Gulalkari, Pandu Sandi Pratama, Jin Wook Kim, Hak Kyeong Kim, Bong Hwan Jun, Sang Bong Kim, "Moving Object Following Control of Six-legged Robot with Four Joint Legs Based on Backstepping Method", 2015년도 한국수중로봇기술연구회 춘계학술대회, pp. 47-53, 2015
- [11] Pandu Sandi Pratama, Jin-Wook Kim, Hak-Kyeong Kim, Suk-Min Yoon, Tae-Kyeong Yeu, Sup Hong, Sea-June Oh and Sang-Bong Kim, "Path Planning Algorithm to Minimize an Overlapped Path and Turning Number for an Underwater Mining Robot", 2015 15th International Conference on Control, Automation and Systems (ICCAS 2015), pp. 499-504, 2015.
- [12] Trong Hai Nguyen, Phuc Thinh Doan, Jin Wook Kim, Dae Hwan Kim, Yeong Seok Jeong, Yeong Ill Ko and Sand Bong Kim, "Application and Evaluation of a New DSP Microcontroller for Sensorless Control of AC Motor Drive System", 2015년도 한국물류과학기술학회 추계학술대회, pp. 21-22, 2015.

- [13] Pandu Sandi Pratama, Jin Wook Kim, Dong Bo Sheng, Tae Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, “Comparison of Optimization Methods to Minimize the Overlapped Path in Coverage Path Planning for Underwater Mining Robot”, 한국해양공학회 2015년도 추계학술대회, pp. 86-89, 2015.
- [14] Pandu Sandi Pratama, Amruta Viond Gulalkari, Jin Wook Kim, Chung Hwan Lee, Tae Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, “Obstacle Avoidance Control for Four Wheels Independent Steering Automatic Guided Vehicles”, The 2nd International Conference on Advanced Engineering-Theory and Applications 2015(AETA 2015) / Lecture Notes in Electric Engineering 371, pp. 603-618, 2015.
- [15] Sang Bong Kim, Dae Hwan Kim, Pandu Sandi Pratama, Jin Wook Kim, Hak Kyeong Kim, Sea June Oh and Young Seok Jung, “MIMO Robust Servo Controller Design Based on Internal Model Principle Using Polynomial Differential Operator, The 2nd International Conference on Advanced Engineering-Theory and Applications 2015(AETA 2015) / Lecture Notes in Electric Engineering 371, pp. 469-484, 2015.