



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

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

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 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

다중 차량 로봇 시스템의 대형 제어를 위한 관측기 기반 분산 제어



2009 년 2 월

부 경 대 학 교 대 학 원

메카트로닉스공학과

윤 석 민

공 학 석 사 학 위 논 문

다중 차량 로봇 시스템의 대형 제어를 위한 관측기 기반 분산 제어

지도교수 김 상 봉

이 논문을 공학석사 학위논문으로 제출함.

2009 년 2 월

부 경 대 학 교 대 학 원

메카트로닉스공학과

윤 석 민

윤석민의 공학석사 학위논문을 인준함.

2009 년 2 월



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 방법 및 내용	4
제 2 장 다중 로봇 제어 시스템 구성	6
2.1 다중 로봇 제어 시스템의 구성	6
2.1.1 초음파 위치 시스템	7
2.1.2 차량 로봇	8
2.2 제어 시스템 구성	11
2.3 차량 로봇의 모델링	12
2.3.1 차량 로봇의 운동학 모델링	12
2.3.2 모델 매개변수 추정	20
제 3 장 로봇의 위치 인식 및 경로 추종	27
3.1 차량 로봇의 위치 인식 알고리즘	27
3.1.1 3차원 위치 인식 원리	28
3.1.2 위치 인식 보정 알고리즘	31
3.1.3 알고리즘 성능 검증 결과	35
3.2 경로 추종 알고리즘	37
3.2.1 LOS 알고리즘	37
3.2.2 시뮬레이션 및 실험 결과	39

제 4 장 미지 입력 관측기에 기반한 분산 제어	45
4.1 미지 입력 관측기 기반 분산 제어	46
4.1.1 대형에서의 다중 로봇 그룹 모델링	46
4.1.2 미지 입력 관측기 설계	48
4.1.3 분산 제어기 설계	53
4.2 대형 모델링, 제어기 및 관측기 설계	55
4.2.1 대형 모델링	55
4.2.2 제어기 및 관측기 설계	57
4.3 Broadcasting 에 의한 분산 제어	60
4.3.1 기본 개념 및 원리	60
4.3.2 시뮬레이션 결과	63
4.3.3. 실험 결과 및 고찰	68
4.4 선도-추종 분산 제어	73
4.4.1 기본 개념 및 원리	73
4.4.2 시뮬레이션 결과	75
4.4.3. 실험 결과 및 고찰	82
제 5 장 결론	88
References	90
Publications and Conferences	94
부록	96
감사의 글	109

An Observer-based Decentralized Control for Formation Control of Multi-Robot Vehicle System

Suk Min Yoon

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

Abstract

In this dissertation, an observer-based decentralized controller is designed to follow reference path and keep formation of multi-robot vehicle system. To do this, kinematic modeling of four wheeled vehicle robot is presented under nonholonomic constraints. To minimize the error between ultrasonic positioning system(UPS) data and the computed data from the kinematic modeling using encoder, parameters such as wheel radius and distance from wheel to geometrical center of vehicle are estimated. A localization algorithm using the data obtained from UPS and encoder is introduced to obtain accurate position data of the vehicle robot. And, the path tracking algorithm of the single vehicle robot is proposed using the proposed localization algorithm and LOS concept. Next, an unknown input observer is proposed to estimate the unknown states in the system with unknown input. A decentralized observer-based controller is proposed for formation control of the multi-robot vehicle system using leader-following method and broadcasting method. To implement the proposed controller, the control system with embedded controller using S3C2240 microcontroller based Linux is developed. Finally, the simulation and experimental results are shown to prove the effectiveness of the proposed controller.

제 1 장 서 론

1.1 연구 배경 및 동기

분산 로봇공학(Distributed robotics) 분야의 연구는 1980년대 후반을 시작으로 당시의 몇몇 연구가들에 의해 다중 로봇 시스템(Multi-robot system)에 대한 연구가 처음으로 소개되었다. 그 이전의 관련 분야 연구는 단일 로봇 시스템(Single robot system) 또는 실질적으로 로봇이 포함되지 않은 분산 문제 해결형 시스템(Distributed problem solving systems)에 집중되었다. 초기의 분산 로봇공학 분야에서의 주요 관심은 재구성 가능형 로봇 시스템(Reconfigurable robot system)[35], 다중 로봇 운동 계획(Multi-robot motion planning)[33] 그리고 다중 로봇 대형을 위한 구조(Architectures for multi-robot cooperation)[34] 등이었다. 이러한 선행연구들이 수행된 후, 다중 로봇 분야에 다양한 주제들이 나오면서 폭넓은 성장을 이루게 되었다.

오늘날 단일 로봇을 대신하여 다중 로봇에 대한 연구가 활발하게 진행되고 있다. 그 이유는 다양하겠지만, 시스템의 효율성을 극대화 시킬 수 있다는 점이 주된 요인이다. 즉, 다중 로봇은 단일 로봇으로는 힘든 임무를 효과적이고 신속하게 수행할 수 있다. 이는 다중 로봇이 여러 로봇들로 구성되어 있어 가외성(Redundancy)을 가지며, 주어진 문제를 협조하여 해결하기 때문에 신뢰성 높고 신속하게 또는 단일 로봇으로 할 수 있는 것보다 저렴한 비용으로 주어진 임무를 수행할 수 있기 때문이다. 이러한 이유 때문에, 다중 로봇은 해양, 우주 개발, 극한 환경 그리고 서비스 로봇 분야에 많은 이점을 가진다. 현재는 탐색 또는 구조 활동 분야, 지뢰 제거, 원격 탐사 또는 우주 탐험 그리고 무인 항공기 및 위성 자세 제어 등에 적용되고 있으며, 더 많은 분야에서 활발한 연구가 진행되고 있다.

그 중에서 Arai[15]는 지금까지 소개된 다중 로봇 시스템과 관련된 연구들을 정리하여 크게 7개의 분야로 분류하였는데, 그 내용을 살펴보면 다음과 같다.

- 1) 생물학적 영감(Biological inspirations),
- 2) 통신(Communication),
- 3) 시스템 구조(Architecture), 업무 할당(Task allocation), 제어(Control)
- 4) 위치 인식(Localization), 사상(Mapping) 및 탐사(Exploration)
- 5) 물체 수송(Object transport)과 조작(Manipulation)
- 6) 운동 조정(Motion coordination)
- 7) 재구성 가능형 로봇(Reconfigurable robots)

지금까지 다중 로봇과 관련된 연구들의 대부분은 일반적으로 위의 7 가지로 분류된 문제들 중에서 하나만을 다루는 것이 아니라 하나 이상의 주제를 복합적으로 다루고 있다.

지금까지 단일 로봇에 대해 위치 인식(Localization)을 하고 사상(Mapping) 및 탐사(Exploration)하는 방법에 대해서는 상당히 많은 연구가 수행되어 왔다. 그리고 최근에는 이러한 방법들이 다중 로봇 분야로 확대 적용되고 있다. 그 중에서도 이동 로봇이 자신의 위치를 인식하는 것은 로봇이 지녀야 할 가장 기본적인 능력이다. 즉, 주어진 조건에서 자신의 위치를 예측하는 것은 로봇의 필수 요건인 것이다. 그러나 실험 배경이 수중이 되면 이러한 기본적인 능력을 갖추는 것도 매우 복잡한 문제가 된다. 그만큼 수중 로봇 분야에서의 위치 인식은 어렵고 매우 중요하다. 그 이유는 육상에서와 달리 수중에서는 통신의 제약이 매우 크기 때문이다. 현재 수중 통신과 관련된 연구가 활발히 진행되고 있지만, 실제 적용되어 사용되는 것은 극히 제한적이다. 일반적으로 수중에서 로봇의 위치를 계측하기 위한 방법으로 널리 사용하는 것은 음파를 이용하는 LBL 시스템(Long baseline system)이다. 이것은 공기 중보다 물속에서 음파 전달 속도가 빠르고 도달 거리가 멀다는 원리를 이용한 것으로, 군사용, 수중 광물 탐사 등에 사용할 목적으로 처음 개발되었다. LBL 시스템은 해저면(Seabed)과 수중 장비에 신호를 주고 받을 수 있는 트랜스폰더(Transponder)를 각각 설치하여 장비의 위치를 측정하는데, 위치를 측정하기 위해서 음파 전송 시간과 수중 음파 속도 정보를 이용한다. LBL 시스템은 기본적으로 최소 3 개의 트랜스폰더가 해저 지반에 설치되어야만 위치 측정이 가능하다. 그리고 트랜스폰더 사이의 거리는 출력 음파 신호의 크기에 따라서 좌우되며, 최대 14,000m 까지 가능하다. 그러나 이것은 수심, 수온과 같은 환경 변화에 대하여 그 특성이 다르기 때문에, 계측 오차의 발생이 불가피하다. 따라서 이러한 문제에 대한 해결책으로, LBL 시스템의 위치 정보와 더불어 로봇의 속도 및 관성 정보 등을 함께 고려한 센서 융합(Sensor fusion) 항법 알고리즘 개발이 제안되고 있다.

한편 다중 로봇의 운동 조정(Motion coordination) 분야에서 집중적으로 다루어지고 있는 주제로는 경로 계획(Path planning)과 대형 제어(Formation control) 등이 있다. 경로 계획 분야는 다시 경로 생성(Path generation)과 경로 추종(Path tracking)으로 크게 나뉘어진다. 경로 생성은 목표점까지의 로봇이 이동해야 할 최적의 경로를 생성하는 문제를 다루며, 경로 추종은 생성된 경로를 따라 로봇이 이동할 수 있도록 로봇의 움직임을 적절히 제어하는 문제를 다룬다. 그리고 다중 로봇의 대형 제어 문제는 로봇 그룹의 대형을 생성하는 문제와 원하는 형태로 대형을 유지하는 문제, 두 가지로 구분되어 정의된다.

다중 로봇의 대형을 생성하고 유지하는데 있어서는, 한편으로는 정확성(Precision)과 가능성(Feasibility) 사이에서 어느 것에 비중을 둘 것인지 고려하여 타협점을 찾고, 다른 한편으로는 전체 시스템 정보와 통신 능력

사이에서 어느 것에 비중을 둘 것인지 고려하여 타협점을 찾는 것이 일반적이다. 전체 시스템 정보 또는 방송 통신(Broadcast communication)을 요구하는 시스템은 확장성의 부족 또는 시스템 설치에 따른 고비용을 가져올 수 있다. 반대로, 좁은 지역의 통신과 그것의 센서 정보만을 이용하는 로컬 시스템은 대형의 정확성(Precision)과 다양성(Diversity)에는 제한을 가지지만, 확장성이 넓고 강인(Robust)하며 설치가 쉽다는 장점을 가진다. 위와 같은 이유로 다중 로봇의 대형 제어에 관한 연구는 크게 두 그룹으로 분류된다. 첫 번째 그룹은 대형 조정(Coordination)을 위하여 중앙 장치(Centralized unit)가 전체 그룹을 감독할 뿐만 아니라 각각의 로봇에 대한 명령까지 결정한다. 그리고 두 번째 그룹은 각각의 로봇에 대한 지령이 로봇 자체의 제한적인 정보를 바탕으로 결정되는 분산적 방법(Decentralized method)을 이용한다. 이 방법은 최근 다중 로봇과 같은 대규모 시스템에 대하여 폭넓게 연구되고 있다.

한편, 다중 로봇의 대형 제어를 중심 역할을 하는 로봇의 기능에 따라서 구분해 보면 선도-추종(Leader-following), 거동(Behavioral), 가상 구조(Virtual structure)로 구분할 수 있다. 이러한 방법들의 공통점은 중심 역할을 하는 로봇의 기능적 차이는 있지만, 서버(Server) 혹은 선도자(Leader)와 같은 중심 역할을 하는 로봇이 필요하다는 것이다. 선도-추종(Leader-following) 방법은 하나의 행위자(Agent)를 선도자 (Leader)로 지명하고, 나머지 행위자는 추종자(Follower)로 지명한다. 그 기본적인 생각은 추종자가 선도자와 일정한 형상을 유지하도록 정해진 대형을 가지고 선도자의 위치와 방위각을 추종하는 것이다. 가상 구조(Virtual structure) 방법은 전체 시스템의 대형을 하나의 구조, 즉 가상의 구조체로 취급하는 것이다. 만약 이러한 가상 구조에 요구되는 동역학이 각 행위자의 원하는 운동으로 변환된다면, 우리는 전체 시스템이 성능을 발휘하도록 로컬 제어기를 설계할 수 있다. 최근에는 앞서 언급한 3 가지의 장점을 종합한 방법이 다중 로봇의 대형 제어에 있어 유용하다는 사실이 증명되고 있다.

Beard[20]는 대형제어를 위하여 분산 제어 방식을 기반으로 감독 장치(Supervisor unit)와 제어 구조(Control architecture)를 제안하였다. 여기서 로컬 제어기는 이러한 장치의 로컬 실증(Instantiation) 상태를 예측하기 위하여 설계된다. 그러나 대형과 로컬 행위자 사이의 상호연결성(Interconnection)과 감독 장치와 행위 사이의 상호 작용(Interaction)을 고려하지 않았을 뿐만 아니라, 로컬 제어에서 언급된 관측기 설계에 대해서도 자세히 설명하지 않았다. Stilwell and Bishop[21]은 자율 이동 로봇의 대형제어를 위하여, 행위자 사이의 최소한의 정보만 통신을 통해 공유하고 나머지는 비선형 관측기를 통해 상태를 예측하는 분산제어 방식을 제안하였다. 다음으로 Stilwell[13]은 대형 제어에 있어, 전체 특성을 나타내는 대형 수준 함수(Platoon lever function)와 외인성 시스템(Exogenous system)을 통해 측정하는 방법을 제안한다. Nguyen[6]은 대형 수준 함수와 관측기를 기반으로 한 분산 제어 방법을 제안하였다.

여기서는 동일한 대형 수준 함수에 대하여 중앙제어기와 분산제어기의 시뮬레이션 결과를 비교함으로써, 분산제어기의 유효성도 함께 설명하고 있다.

앞서 언급한 바와 같이, 수중에서 단일 또는 다중 로봇을 제어하는 것은 통신의 제약으로 인하여 그 실현이 매우 어려운 실정이다. 수중의 LBL 시스템(Long baseline system)에 의해 측정된 이동 로봇의 위치 정보는 많은 외부 환경적 잡음(Noise)을 가지고, 깊이 또는 거리에 비례하여 시간적 지연 및 오차를 가진다. 이러한 특징을 가지는 위치 측정 시스템으로부터 정확한 이동 로봇의 위치를 알기 위해서는, 위치를 보정할 수 있는 다른 센서가 필요하게 된다. 따라서 이들의 데이터를 상호 보완하여 위치 인식할 수 있는 위치 인식 알고리즘이 꼭 필요하다. 또한, 수중에서는 이동 로봇 사이의 상호 정보 교환이 매우 제한적이며, 교환된 정보 또한 그 신뢰성이 매우 낮다. 그러나 다중 로봇의 대형을 제어하기 위해서는 각 이동 로봇간에 많은 양의 정보 교환이 필요하므로, 수중에서 대형을 제어하는데 문제가 된다. 따라서 정보 교환의 양을 최소로 할 수 있는 다중 로봇의 제어 기술 개발이 필요하다. 마지막으로, 알고리즘 개발 단계에서 그 성능을 검증하기 위해 실험적 시험을 수행하는 것은 위험 부담이 클 뿐만 아니라 많은 인력과 시간, 높은 비용이 요구된다. 이러한 실험적 제약에 따른 어려움을 극복하기 위해서는 수중과 유사한 육상 계측 장비와 육상 로봇을 이용한 다중 로봇의 제어 기술 개발이 절실히 필요하다. 그리고 본 연구에서는 Arai[15]가 분류한 7 가지 분야 중에서 다중 로봇의 위치 인식(Localization)과 이를 기반으로 한 다중 로봇의 운동 조정(Motion coordination) 분야를 중점적으로 다루고자 한다.

1.2 연구 방법 및 내용

수중에서 단일 또는 다중 로봇을 제어하는 것은 통신의 제약으로 인하여 그 실현이 매우 어려우며, 실험적 시험을 수행하는 것은 많은 위험 부담을 가진다. 이러한 실험적 제약에 따른 어려움을 극복하기 위하여, 본 연구에서는 수중과 유사한 육상 계측 장비와 육상 로봇을 이용하여 다중 로봇 제어 방법을 구현한다.

먼저 차량 로봇의 위치를 측정하기 위해, 수중의 LBL 시스템(Long baseline system)과 유사한 육상용 초음파 위치 시스템(Ultrasonic positioning system)을 적용한다. 이것은 정해진 범위 내에서만 측정이 가능하며, 거리에 따른 시간적 지연은 없지만, 위치를 측정하는데 있어 차량 로봇의 제어 주기 시간보다 월등히 긴 최소 0.6 초가 소요된다. 또한, 벽면 반사와 같은 환경적 영향으로 인하여 측정 오차가 불규칙적으로 나타난다. 이것은 수중의 LBL 장비가 많은 외부 환경적 잡음(Noise)을 가지고, 깊이 또는 거리에 비례하여

시간적 지연 및 오차를 가지는 것과 매우 유사한 특성이다. 이러한 특징을 가지는 위치 측정 시스템으로부터 차량 로봇의 위치를 정확하게 알기 위해서, 위치를 보정하기 위한 센서로 엔코더를 선정하였다. 차량 로봇을 제어하기 위해 차량 로봇의 모델링을 수행하고, 유도된 운동학적 방정식을 이용하여 계산된 위치 데이터와 실제 측정된 위치 데이터의 오차가 최소화 되도록 운동학적 방정식의 매개변수(Parameter)를 추정(Estimation)한다. 매개변수 추정을 통한 운동학적 방정식과 초음파 위치 시스템의 위치 데이터를 상호 보완하여, 차량 로봇 제어에 가장 기본이 되는 위치 인식 알고리즘을 제안한다. 그리고 제안된 위치 인식 알고리즘과 LOS(Line of sight) 알고리즘을 이용하여 단일 차량 로봇을 이용한 경로 추종 알고리즘을 제안한다.

다중 로봇의 대형을 제어하기 위한 방법은 크게 중앙 제어(Centralized control) 방법과 분산 제어(Decentralized control) 방법으로 나뉘어 지는데, 본 연구에서는 분산 제어 방법을 선정하였다. 앞서 이야기한 수중에서의 통신 제약으로 인해 주고 받아야 하는 데이터의 수는 적을수록 좋기 때문에, 모든 데이터를 주고 받아야 하는 중앙 제어 방식보다 필요한 데이터만 최소한으로 주고 받을 수 있는 분산 제어 방식이 더 유리하다. 그러나 이러한 분산 제어 방식으로 최소한의 데이터만을 주고 받기 위해서는 서버에서 전송 받은 데이터와 각 로봇에서 알고 있는 데이터만을 이용해서 대형을 유지해야만 한다. 따라서 다른 로봇에 대한 입력 및 상태 정보를 알 수가 없게 되므로, 이러한 미지의 정보를 가지는 시스템에서 상태를 추정하는 미지 입력 관측기(Unknown input observer)와 이를 기반으로 하는 분산 제어를 극배치법으로 설계한다. 그리고 분산 제어를 구현하기 위한 방법으로 본 연구에서는 Broadcasting 방식과 선도-추종 방식을 제안한다. Broadcasting 방식은 미지 입력 관측기를 기반으로 하는 분산 제어 방식으로, 대형 수준 함수(Formation level function)를 감독 컴퓨터(Supervisor computer)에서 계산하여 방송 통신(Broadcasting communication) 방법을 통해 각 차량 로봇으로 전달한다. 미지 입력 관측기는 각 차량 로봇에서 분산 제어기에 사용될 피드백 상태 정보를 예측하는데 사용되며, 여기서의 미지 입력은 다른 차량 로봇의 입력 정보를 의미한다. 선도-추종 방식에서의 선도 차량 로봇은 앞서 제안된 경로 추종 알고리즘에 의하여 주어진 경로를 일정한 속도로 따라 움직이고, 추종 차량 로봇이 Broadcasting 방식과 동일한 분산 제어를 가지고 선도 차량 로봇과 일정한 거리를 유지하면서 대형을 유지한다. 제안된 제어기를 구현하기 위해 리눅스에 기반한 임베디드 컨트롤러를 사용한 제어시스템을 개발한다. 제안된 제어기와 알고리즘의 유효성을 검증하기 위해 시뮬레이션과 실험 결과를 제시한다.

제 2 장 다중 로봇 제어 시스템 구성

2.1 다중 로봇 제어 시스템의 구성

다중 로봇 제어를 위한 시험 기반 장비는 Fig. 2-1 과 같이 초음파 위치 시스템(Ultrasonic positioning system)과 차량 로봇(Vehicle robot) 그리고 리눅스(Linux) 기반의 서버 시스템(Server system)으로 구성되어 있다.

차량 로봇의 위치를 측정하기 위한 장비로 초음파 위치 시스템을 사용하였다. 차량 로봇에 탑재되어 있는 리눅스 기반의 임베디드 컨트롤러는 주변 장치의 데이터를 취합하여 차량 로봇의 위치 및 전진방향각을 인식하며, 이것을 무선 공유기(Wireless router)를 통하여 서버 시스템으로 전송한다. 서버 시스템은 각각의 차량 로봇으로부터 전송 받은 위치와 전진방향각 데이터를 취합하여 다중 로봇 제어에 필요한 데이터를 생성하여 각 차량 로봇으로 전달한다.

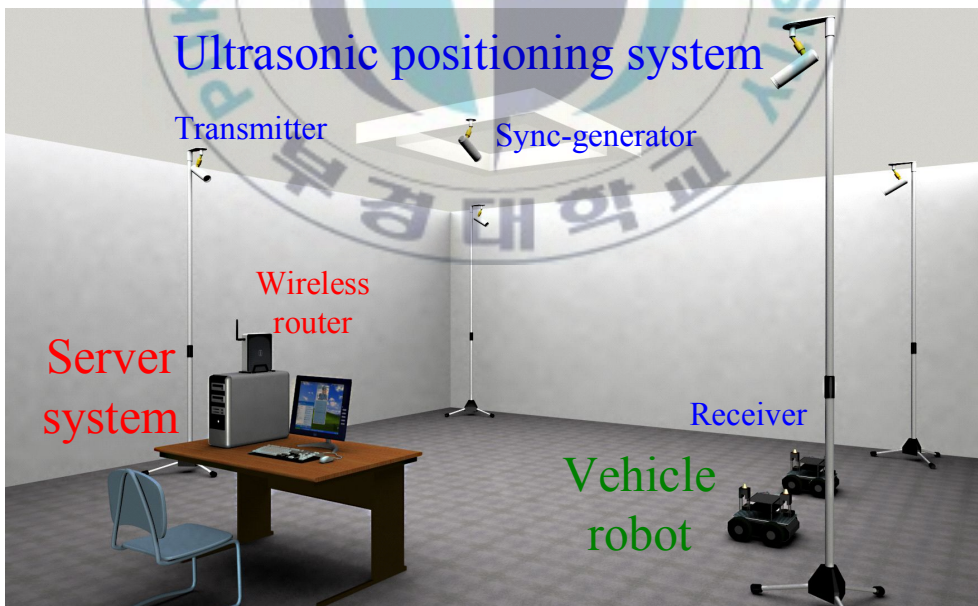
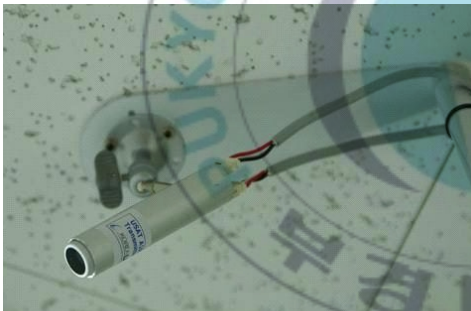


Fig. 2-1 Base experimentation environment

2.1.1 초음파 위치 시스템

초음파 위치 시스템은 Fig. 2-2 (a)의 발신기(Transmitter)와 (b)의 동기기(Sync-generator) 그리고 (c)의 수신기(Receiver)로 구성되어 있다. 물체의 위치를 측정하기 위해서는 최소 4 대의 발신기가 필요하다. 동기기는 RF 무선과 10m 유선 케이블을 이용하여 각각 수신기와 1에서 4 번까지의 발신기에 0.15 초 간격으로 동기신호를 보낸다. 4 대의 발신기로부터 신호를 받은 수신기는 3 장에서 거론될 3 차원 좌표인식 원리를 이용하여 자신의 위치를 계산하고 그 결과를 시리얼 통신을 통해 외부로 전송한다.

초음파 위치 시스템의 위치 측정에 소요되는 시간은 0.6 초이고 실험 공간은 4 대의 발신기가 놓여진 가로 5m, 세로 6m 의 공간이며, 이 공간을 벗어나게 되면 정상적인 위치 측정이 불가능하다. 초음파 위치 시스템은 발신기에서 초음파를 발송하여 수신기에서 받는 형태로 시스템이 구현되어 있기 때문에, 이론상 사용 대수의 제약 없이 무한대의 수신기를 추가해서 사용 가능하며, 수신기를 차량 로봇에 자유롭게 부착할 수 있기 때문에 다중 로봇 제어 환경에서의 위치 인식 시스템으로 적합하다.



(a) Transmitter



(b) Sync-generator



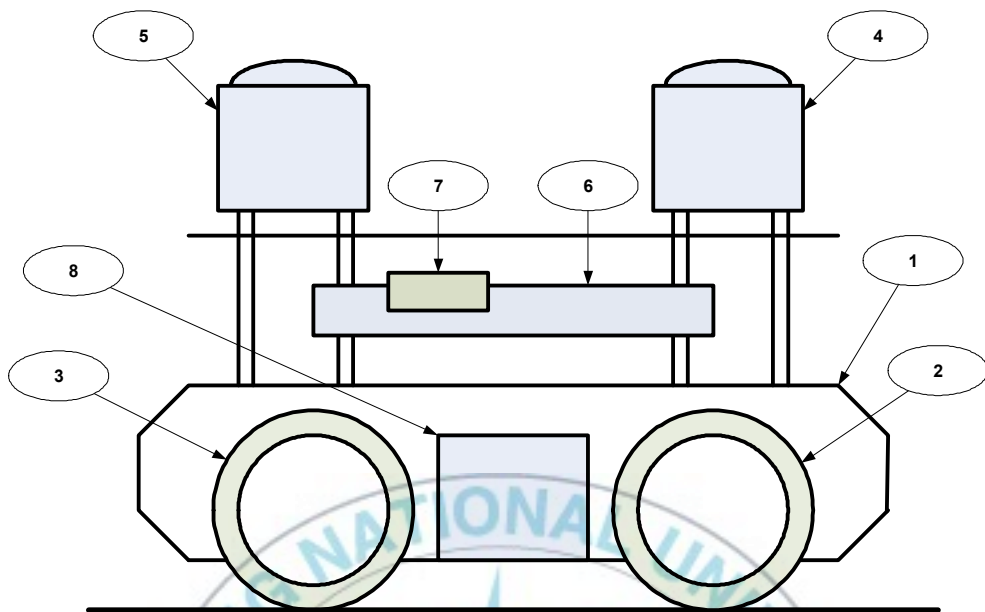
(c) Receiver

Fig. 2-2 Ultrasonic positioning system

2.1.2 차량 로봇

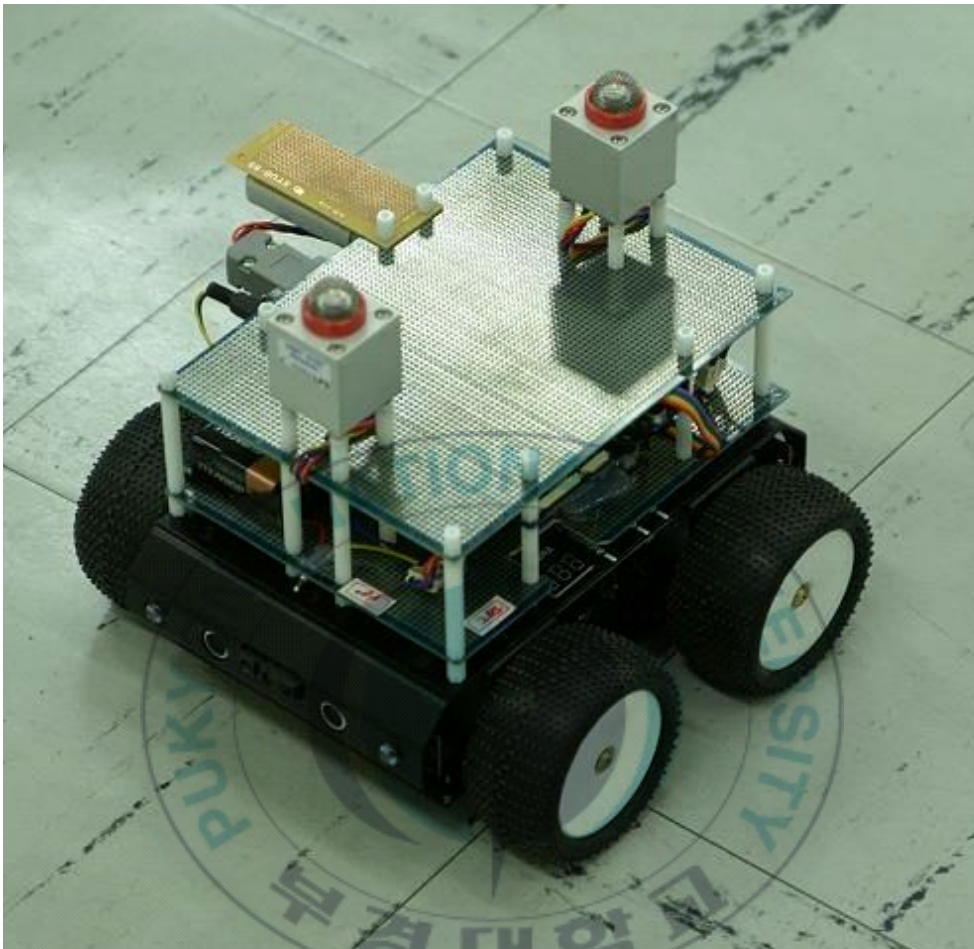
Fig. 2-3 은 차량 로봇의 구성을 나타낸 것으로 좌우 바퀴의 구동력 차이를 이용하여 전륜의 조향 없이 차량의 선회가 가능한 스키드 조향(Skid steering) 방식으로 구동하는 차량 로봇의 실제 사진은 Fig. 2-4 의 (a)이다.

Fig. 2-3 에서 1 번은 4 개의 바퀴를 가지며, 바퀴마다 DC 기어 모터(Geared motor) 장착되어 있는 이동차량을 나타낸 것으로 실제 사진은 Fig. 2-4 (c)와 같다. 이동 차량을 위한 마이크로 컨트롤러로는 Atmega128L 을 사용하고 있으며, 이를 이용해서 이동 차량을 실질적으로 제어한다. 2 번과 3 번은 차량 로봇의 앞뒤 바퀴이며, 차량 로봇 뒤 바퀴의 양쪽 모터에는 650 펄스(Pulse) 엔코더가 내장되어 있다. 4 번과 5 번은 앞에서 소개된 초음파 위치 시스템의 수신기로 차량의 앞뒤에 하나씩 붙어 있어 차량의 위치와 전진방향각을 측정할 수 있다. 6 번은 차량 로봇을 실질적으로 제어하는 임베디드 컨트롤러이며, 리눅스를 기반으로 하는 S3C2440 마이크로 컨트롤러를 사용하였다. 본 연구에서 사용된 임베디드 컨트롤러는 3 개의 시리얼 포트를 사용하여 2 개의 수신기와 이동 차량의 마이크로 컨트롤러로부터 데이터를 수신한다. 임베디드 컨트롤러의 제어 주기는 0.1 초이며 제어 주기마다 이동 차량에 제어 출력을 전달하고 이동 차량의 엔코더 데이터를 받는다. 그리고 초음파 위치 시스템의 수신기로부터 0.6 초마다 위치 데이터를 받으며 이를 이용해서 위치 인식 알고리즘을 실행해 차량 로봇의 위치 좌표와 전진방향각을 계산한다. 7 번은 임베디드 컨트롤러에 연결된 USB 타입의 무선랜이며, 무선 공유기를 통해 리눅스 서버 시스템과 무선 통신을 하게 된다. 본 연구에서는 서버 시스템과 임베디드 컨트롤러는 리눅스 기반으로 구현하였다. 이는 본래 리눅스가 멀티태스킹(Multi-tasking)과 멀티유저(Multi-user)를 기반으로 하여 만들어진 운영체제이므로, 다중 로봇 제어를 위한 통신 서버 시스템을 에러(Error)에 강인하고 컴퓨터 자원(Resource)의 낭비가 없도록 구현하는데 보다 유리하기 때문이다. 8 번은 이동차량을 구동하고 임베디드 컨트롤러에 전원을 공급하기 위한 11.1V 의 5400mA 용량을 가지는 내부 배터리이다. 초음파 위치 시스템의 수신기에는 9V 전원의 건전지를 따로 2 개 사용하고 있다.

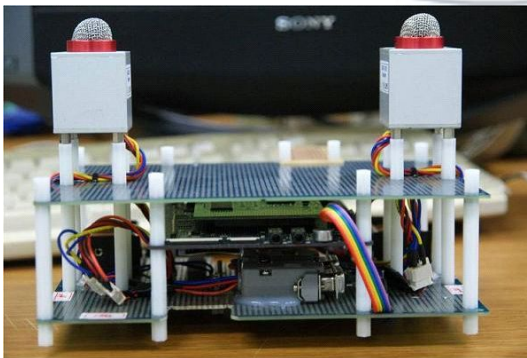


1. Mobile vehicle body
2. Front wheel
3. Rear wheel
4. Front receiver
5. Rear receiver with encoder
6. Embedded controller
7. Wireless LAN
8. Battery

Fig. 2-3 Configuration of vehicle robot



(a) Vehicle robot



(b) Embedded controller



(c) Mobile vehicle

Fig. 2-4 Vehicle robot

2.2 제어 시스템 구성

다중 로봇을 제어하기 위한 제어 시스템의 구성은 Fig. 2-5 와 같다.

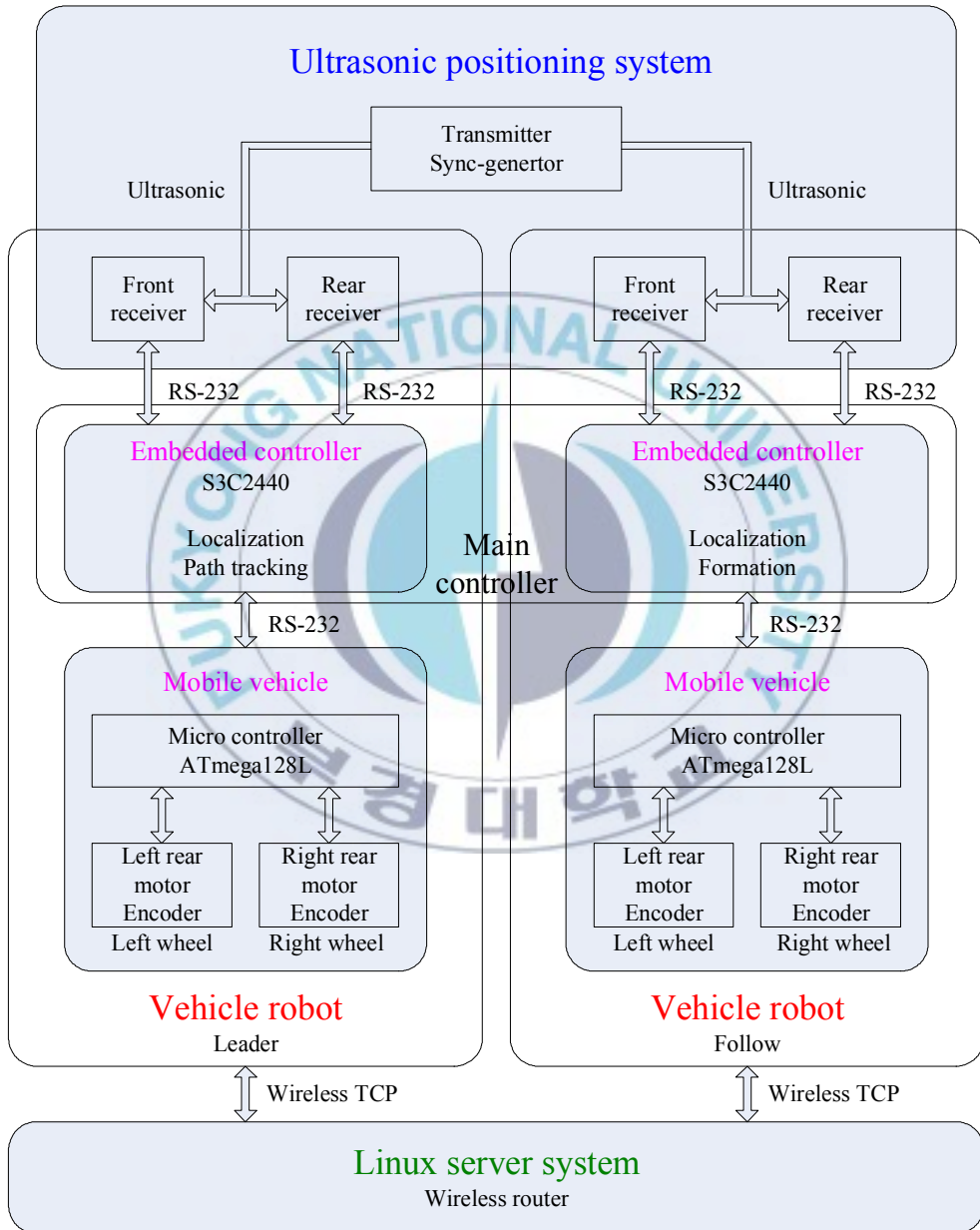


Fig. 2-5 Configuration of vehicle robot control system

리눅스 기반의 서버 시스템은 2 대의 차량 로봇으로부터 다중 로봇 차량을 제어하기 위해서 필요한 각 차량의 위치와 전진방향각 정보를 무선 공유기를 통해 획득한다. 통신 방법으로는 신뢰성 있는 통신을 하기 위해 TCP 통신을 사용하였으며 이를 통해서 각 차량 로봇에 필요한 데이터를 동시에 전송한다. 차량 로봇이 가지는 4 개의 바퀴 중에 뒤 바퀴 2 개에는 엔코더가 각각 달려 있으며, 이 엔코더 값을 마이크로 프로세서인 ATmega128L 을 이용하여 계속적으로 측정한다. 초음파 위치 시스템은 동기기에 의해 동기 신호를 발생시키고 그 동기 신호를 발신기와 수신기에 전달한다. 그리고 발신기에서 쏘아진 초음파를 수신기에서 받아 3 차원 좌표 인식 원리를 이용하여 수신기의 위치를 측정하게 된다. ATmega128L 과 수신기로부터의 엔코더와 위치 데이터는 차량 로봇의 주 컨트롤러 역할을 하는 임베디드 컨트롤러 S3C2440 에서 RS232 시리얼 통신을 이용하여 모두 취합한다. 받아 들인 데이터를 이용하여 현재 차량 로봇의 위치를 인식하고 경로 추종 등의 알고리즘을 실행하게 된다. 그리고 TCP 무선 통신을 통하여 서버 시스템에 차량 로봇의 정보를 전달하는 역할을 수행한다. 서버 시스템과 임베디드 컨트롤러는 모두 리눅스 기반에서 이루어지며, 이는 좀더 효율적으로 시스템을 구성하고 안정적인 서버 시스템을 구현하는 것을 도와준다.

2.3 차량 로봇의 모델링

차량 로봇은 비홀로노믹 구속조건(nonholonomic constraints)을 가지기 때문에 이 차량 로봇의 운동학적 구속방정식들(Kinematics constraint equations)을 적분할 수 없다는 특징이 있다. 이러한 제약들은 구속방정식 좌표를 간소화할 수 없다는 특징이 있기 때문에 운동학적, 동력학적 해석에 많은 어려움이 있다.

2.3.1 차량 로봇의 운동학 모델링

이 절에서는 단일 차량 로봇의 운동학적 모델(kinematics model)만을 제시한다. 1 번째 차량 로봇과 2 번째 차량 로봇의 운동학적 모델은 같으므로, Fig. 2-6 는 운동학적 모델링을 위하여 차량 로봇을 좌표계로 표현한 것이다. 로컬좌표계는 차량 로봇의 질량중심에 위치하며, 이 차량 로봇의 질량중심 위치는 전체 좌표계에서 (X, Y, θ) 이다. 차량 로봇 시스템 모델링에 사용된 매개변수들은 Table 2.1 에 표시된 바와 같다.

차량 로봇은 4 개의 바퀴를 가지며, 각 바퀴는 차체의 기하학적인 중심부터 동일한 거리에 위치하고 있으며, 바닥과의 미끄러짐은 없고 순수 회전 한다고

가정한다. 바퀴마다 하나의 모터가 장착되어 있으며, 왼쪽 2 개와 오른쪽 2 개의 모터는 동시에 동작하도록 회로가 구성되어있다. 또한 차량 로봇은 스키드 조향 방식으로 이동하며, 몸체는 강체이고 2 차원 평면에서만 이동한다고 가정한다. 차량의 질량중심은 $COM = (X, Y, Z)$ 로 표현되고 본 연구에서는 평면 운동만을 고려하므로 Z 는 고정이며, 차량의 질량중심은 차체의 기하학적 중심에 위치한다고 가정한다.

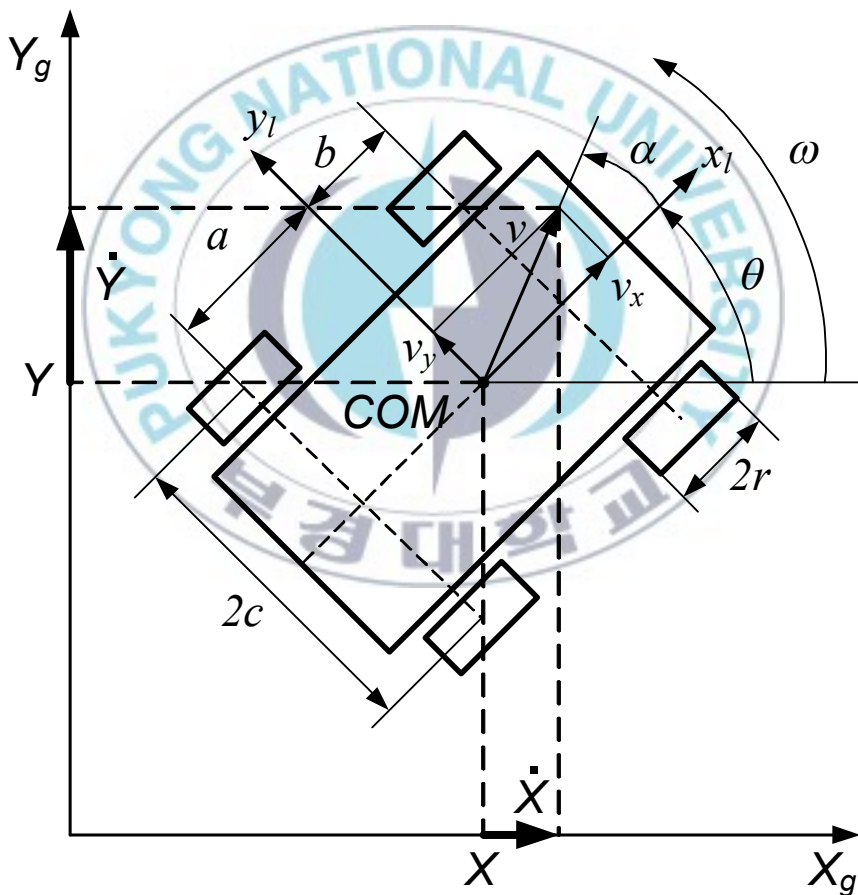


Fig. 2-6 Coordinates of four-wheeled vehicle

Table 2.1 Parameters used for AGV system modeling

Parameter	Description
(X_g, Y_g, Z_g)	전체 좌표계
(x_l, y_l, z_l)	로컬 좌표계
$COM(X, Y, Z)$	차량 로봇의 질량중심
a [m]	뒤 바퀴 축과 y_l 축과의 거리
b [m]	앞 바퀴 축과 y_l 축과의 거리
c [m]	좌우 바퀴 사이 거리의 1/2
r [m]	바퀴의 반지름
r_i [m]	차량 로봇 각 바퀴의 유효구름반경 (Effective rolling radius)
$\mathbf{v}$ [m/s]	차량 로봇의 전진속도벡터
v_x [m/s]	차량 로봇 전진속도벡터의 x_l 축 성분
v_y [m/s]	차량 로봇 전진속도벡터의 y_l 축 성분
v_{wix} [m/s]	차량 로봇 i 번째 바퀴 선속도의 x_l 축 성분
ω [rad/s]	차량 로봇의 회전각속도
ω_{wi} [rad/s]	차량 로봇 i 번째 바퀴의 회전각속도
θ [rad]	X_g 축과 x_l 축의 사이 각
α [rad]	x_l 축과 차량 로봇의 전진속도벡터의 사이 각

차량 로봇의 구속 방정식을 유도하기 위해 차량 로봇의 자세를 고려한 일반화 좌표 벡터를 식 (2.1)과 같이 정의하면, 일반화 속도 벡터는 식 (2.2)와 같다.

$$\mathbf{q} = [X \ Y \ \theta]^T \quad (2.1)$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (2.2)$$

차량 로봇의 움직임을 간단히 하기 위해서 바퀴의 두께는 무시하며, 좌우 구동 바퀴와 바닥은 한 점에서 접촉한다고 가정한다. 그리고 차량의 운동학적 구속조건으로부터 바퀴와 바닥의 차량 로봇의 진행방향 슬립은 없다고 가정한다.

Fig. 2-7 에 의해 i 번째 바퀴의 각속도 ω_{wi} 와 i 번째 바퀴의 전진속도 벡터 v_{wix} 의 x_i 축 성분과의 관계식은 식 (2.3)과 같다.

$$v_{wix} = r_i \omega_{wi} \quad (2.3)$$

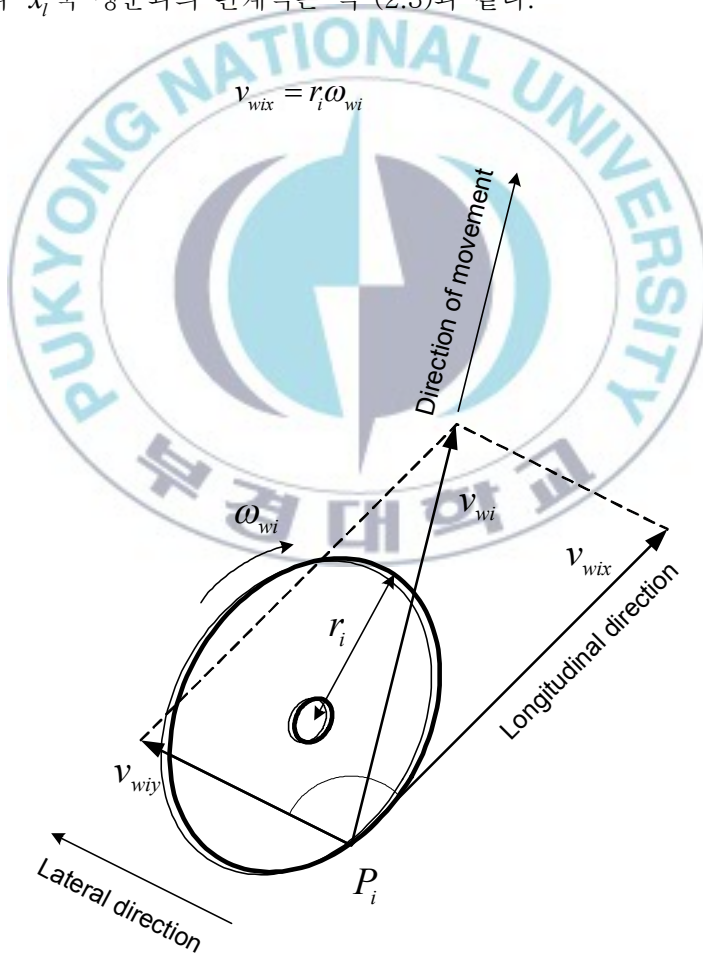


Fig. 2-7 Velocities of one wheel

Fig. 2-8 의 순간 회전 중심(Instantaneous center of rotation : ICR) 에서 i 번째 로컬좌표계까지의 회전반지름 벡터들을 식 (2.4)라고 정의하면 차량 로봇의 속도와 각속도의 관계는 식 (2.5)와 같다.

$$\begin{cases} \mathbf{d}_i = [d_{ix} & d_{iy}]^T \quad \text{for } i=1,2,3,4 \\ \mathbf{d}_C = [d_{Cx} & d_{Cy}]^T \quad \text{for } COM \end{cases} \quad (2.4)$$

$$\frac{\|\mathbf{v}_{wi}\|}{\|\mathbf{d}_i\|} = \frac{\|\mathbf{v}\|}{\|\mathbf{d}_C\|} = |\omega| \quad \text{or} \quad \frac{v_{wix}}{-d_{iy}} = \frac{v_x}{-d_{Cy}} = \frac{v_{wiy}}{d_{ix}} = \frac{v_y}{d_{Cx}} = \omega \quad (2.5)$$

여기서 $\mathbf{v}_{wi} = [v_{wix} \quad v_{wiy}]^T$ 는 i 번째 바퀴의 선속도 벡터이며, $\|\cdot\|$ 는 Euclidian norm 이다.

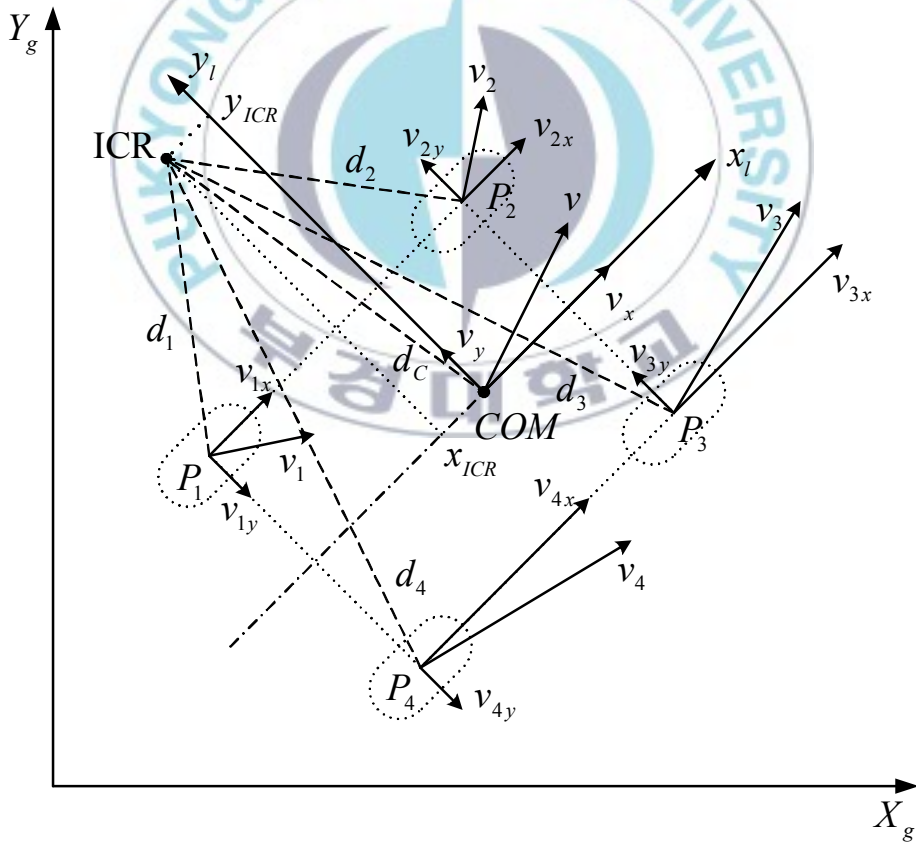


Fig. 2-8 Wheel velocities

로컬 좌표계의 COM 에서 순간 회전 중심의 좌표를 식 (2.6)과 같이 정의하면 식 (2.5)를 식 (2.7)과 같이 표현할 수 있다.

$$ICR = (x_{ICR}, y_{ICR}) = (-d_{Cx}, -d_{Cy}) \quad (2.6)$$

$$\frac{v_x}{y_{ICR}} = -\frac{v_y}{x_{ICR}} = \omega \quad (2.7)$$

여기서 (x_{ICR}, y_{ICR}) 은 COM 에서 ICR 의 위치 좌표이며, (d_{Cx}, d_{Cy}) 는 ICR 에서 COM 의 위치 좌표이다.

Fig. 2-8 으로부터 반지름 벡터 $\mathbf{d}_i$ 의 좌표는 식 (2.8)과 같은 관계를 만족한다. 여기서 a, b, c 는 Fig. 2-6 에 묘사되어진 양의 운동학적 변수들이다. (일반적으로 $b < a$)

$$\begin{cases} d_{1y} = d_{2y} = d_{Cy} + c \\ d_{3y} = d_{4y} = d_{Cy} - c \\ d_{1x} = d_{4x} = d_{Cx} - a \\ d_{2x} = d_{3x} = d_{Cx} + b \end{cases} \quad (2.8)$$

여기서 d_{ix}, d_{iy} 는 x_i, y_i 방향으로의 ICR 에서 각 바퀴까지의 거리이다. 식 (2.5)와 식 (2.8)을 이용하면, 각 바퀴 사이의 관계식은 식 (2.9)와 같다.

$$\begin{cases} v_L = v_{w1x} = v_{w2x} \\ v_R = v_{w3x} = v_{w4x} \\ v_F = v_{w2y} = v_{w3y} \\ v_B = v_{w1y} = v_{w4y} \end{cases} \quad (2.9)$$

여기서 v_R, v_L 은 오른쪽과 왼쪽 바퀴들의 x_i 축 속도이며, v_F, v_B 는 앞쪽과 뒤쪽 바퀴들의 y_i 축 속도이다.

앞의 관계식으로부터 차량 로봇 속도들과 바퀴 속도들 사이의 관계식은 식 (2.10)과 같이 표현할 수 있다.

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & -c \\ 1 & c \\ 0 & -x_{ICR} + b \\ 0 & -x_{ICR} - a \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \quad (2.10)$$

차량 로봇의 유효구름반경을 $r_i = r$ 이라고 하면 식 (2.3)은 식 (2.11)과 같이 쓸 수 있다.

$$\omega_\omega = \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_R \\ v_L \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 \\ \lambda \end{bmatrix} v_R = \frac{1}{r} \begin{bmatrix} 1 & c \\ 1 & -c \end{bmatrix} \eta \quad (2.11)$$

여기서 ω_L , ω_R 은 각각 왼쪽과 오른쪽 바퀴의 각속도이며, λ 선회비로 $\lambda \equiv v_L/v_R = \omega_L/\omega_R$ 와 같이 정의한다.

식 (2.10)과 식 (2.11)을 결합하면, 차량 로봇의 속도와 바퀴의 각속도 사이의 관계식을 식 (2.12)와 같이 표현할 수 있다.

$$\eta = \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2c} & -\frac{r}{2c} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{1+\lambda}{2} \\ \frac{1-\lambda}{2c} \end{bmatrix} v_R \quad (2.12)$$

여기서 η 는 차량 로봇의 전진속도벡터의 x_i 축 성분과 각속도를 성분으로 하는 보조속도벡터이며, 운동학적 모델에서의 새로운 제어 입력이다.

x_{ICR} 가 b 보다 크게 되면 y_i 축을 따라 미끄러지며 제어가 이루어지지 않으므로 x_{ICR} 는 a 보다 작으며, 이로부터 식 (2.13)과 같은 구속 조건을 유도할 수 있다[26].

$$v_y + x_{ICR} \dot{\theta} = 0 \quad (2.13)$$

차량 로봇이 n 개의 상태변수와 m 개의 구속방정식들에 의해 운동이 영향을 받는다고 하면, 이 구속방정식은 식 (2.14)과 같이 표현될 수 있다.

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = 0 \quad (2.14)$$

여기서 $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ 는 구속조건 행렬이다.

바퀴와 바닥의 차량 로봇 진행방향 슬립이 없다고 가정하면, 운동학적 비홀로노믹 구속조건은 식 (2.13)을 고려하여 식 (2.15)와 같이 주어진다.

$$-\dot{X} \sin \theta + \dot{Y} \cos \theta + x_{ICR} \dot{\theta} = 0 \quad (2.15)$$

식 (2.14)과 식 (2.15)으로부터 비홀로노믹 구속조건은 식 (2.16)과 같이 표현된다.

$$\begin{bmatrix} -\sin \theta & \cos \theta & x_{ICR} \end{bmatrix} \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T = \mathbf{A}(\mathbf{q}) \dot{\mathbf{q}} = 0 \quad (2.16)$$

$\mathbf{S}(\mathbf{q})$ 가 $\mathbf{S}^T(\mathbf{q})\mathbf{A}^T(\mathbf{q}) = \mathbf{0}$ 을 만족하는 $\mathbf{A}(\mathbf{q})$ 의 영공간(null space) full-rank 행렬이라고 하면, $\dot{\mathbf{q}}$ 또한 식 (2.14)에 의해 $\mathbf{A}(\mathbf{q})$ 의 영공간이 된다. 따라서 $\dot{\mathbf{q}}$ 와 $\mathbf{S}(\mathbf{q})$ 는 둘 다 $\mathbf{A}(\mathbf{q})$ 의 영공간이므로 $\dot{\mathbf{q}}$ 는 식 (2.17)과 같이 $\mathbf{S}(\mathbf{q})$ 와 보조속도벡터 $\boldsymbol{\eta}$ 의 선형결합으로 표현할 수 있으며, $\mathbf{S}(\mathbf{q})$ 는 식 (2.18)과 같다.

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\boldsymbol{\eta} \quad (2.17)$$

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} \cos \theta & x_{ICR} \sin \theta \\ \sin \theta & -x_{ICR} \cos \theta \\ 0 & 1 \end{bmatrix} \quad (2.18)$$

식 (2.17)은 차량 로봇의 운동학적 운동방정식이며, $\dim(\boldsymbol{\eta}) = 2 < \dim(\mathbf{q}) = 3$ 으로부터 액츄에이터(actuator)의 수가 제어해야 할 자유도(degrees of freedom)보다 작은 underactuated system 이며 비홀로노믹 시스템이다.

x_{ICR} 이 차량 로봇의 기하학적 중심에 위치하면, 즉 $x_{ICR} = 0$ 이라고 가정하면, 비홀로노믹 구속조건을 가지는 차량 로봇의 운동학적 방정식은 식 (2.19)과 같이 주어진다.

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \mathbf{S}(\mathbf{q})\boldsymbol{\eta} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r+r\lambda}{2} \\ \frac{r-r\lambda}{2c} \end{bmatrix} \omega_R \quad (2.19)$$

2.3.2 모델 매개변수 추정

앞서 유도된 운동학 방정식 식 (2.19)를 통해 계산된 정보와 실제 계측된 위치 정보 사이의 오차 정도를 파악하여 운동학 방정식의 적용 가능성을 판단하는 것이 선행되어야 한다.

차량 로봇 운동학 방정식에서의 제어 입력 벡터 식 (2.12)의 매개변수 r , c 는 다음과 같이 $r = 0.044m$, $c = 0.106m$ 실제 측정되었다. 차량 로봇의 엔코더 데이터로 ω_R , ω_L 을 구하고, 식 (2.11)을 이용하여 v_R , v_L 구한 뒤 선회비를 구하고, 식 (2.12)과 식 (2.19)를 이용하여 차량 로봇의 이동 위치를 구한다. 이 선회비에 따른 차량 로봇의 이동 위치를 표시한 결과는 Fig. 2-9 와 같으며, 그래프 각각의 선회비는 (a) 1.1007, (b) 1.1740, (c) 1.2430, (d) 1.3159, (e) 1.3831, (f) 1.4557 이고 $r = 0.044m$, $c = 0.106m$ 값을 사용하였다.

초음파 위치 시스템으로 측정한 위치 벡터 $\mathbf{q}_{ups} = [X_{ups} \ Y_{ups} \ \theta_{ups}]^T$ 와 엔코더로 측정한 위치벡터 $\mathbf{q}_{en} = [X_{en} \ Y_{en} \ \theta_{en}]^T$ 을 이용하여 두 데이터의 위치 오차를 식 (2.20)과 같이 정의하면 Fig. 2-9 의 각 그래프에 대한 오차는 Fig. 2-10 과 같다.

$$\begin{cases} e_{Xd} = X_{ups} - X_{en} \\ e_{Yd} = Y_{ups} - Y_{en} \end{cases} \quad (2.20)$$

Fig. 2-10 으로부터 측정한 실제 r , c 값을 사용하였을 경우, 선회비가 증가 함에 따라 실제 계측된 초음파 위치 시스템의 데이터와 엔코더를 이용하여 식 (2.12)와 식 (2.19)로 계산된 차량좌표의 위치 데이터의 차이가 더 작아짐을 알 수 있다. 이것은 식 (2.12)의 정확도는 v_x 방향의 슬립에 의존하며 이 슬립현상이 지배적이지 않을 때만 유효하기 때문인 것으로 풀이된다. 또한 차량 로봇 바퀴의 표면이 고무로 되어 있기 때문에 이동 시 발생하는 바퀴의 형상 변화와 무게 중심 측정 오차로 인해 영향을 받은 것으로도 보여진다. 그러나 두 데이터의 오차가 $1m$ 가 넘는 큰 값이 가지므로 오차를 보정할 수 있는 방법이 필요하다.

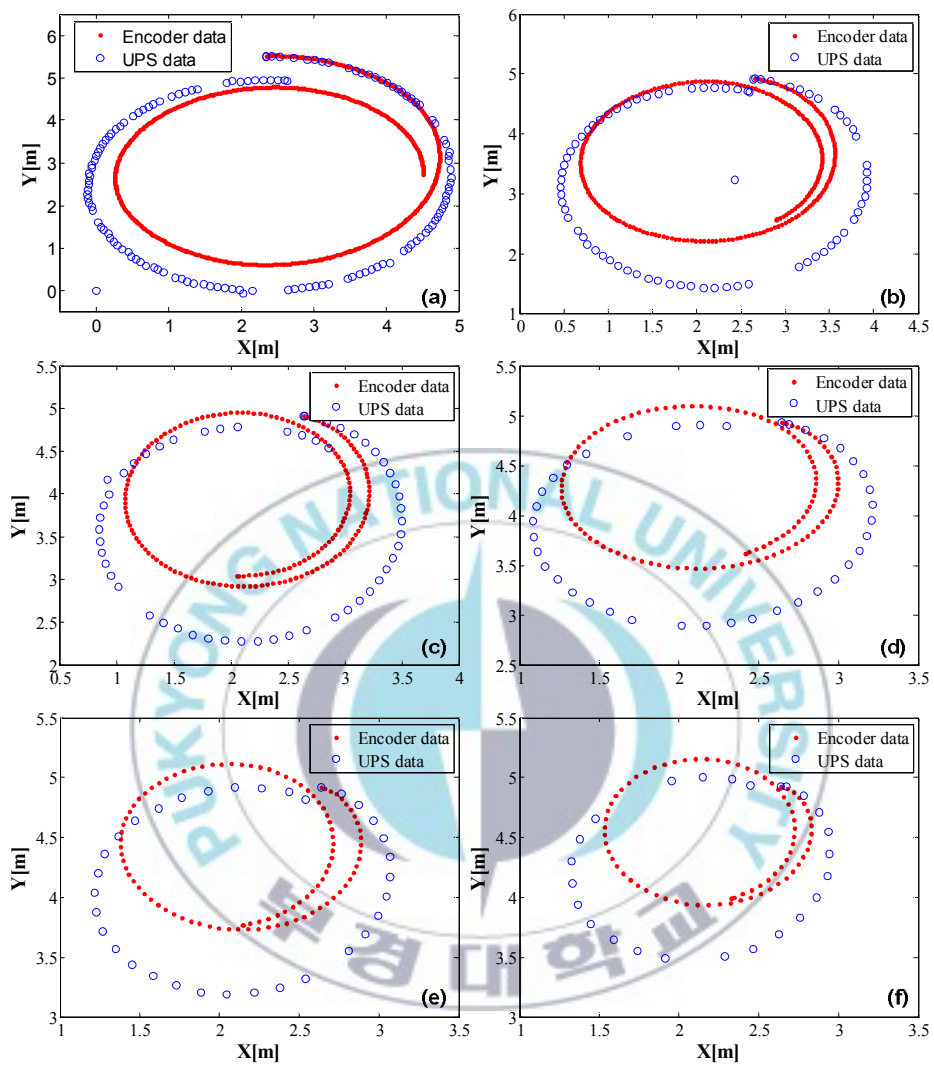


Fig. 2-9 Vehicle movement with real r , c

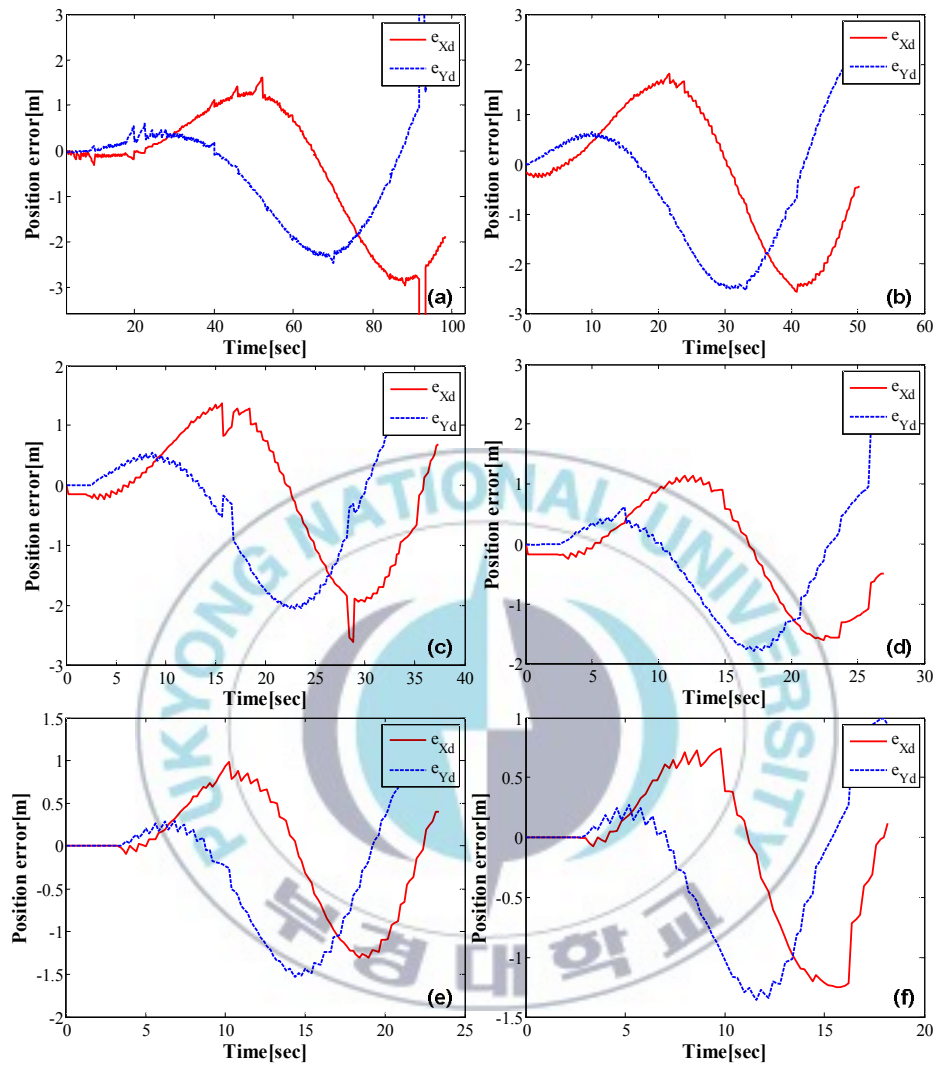


Fig. 2-10 Position error with real r , c

매개변수 r , c 는 바퀴들의 회전 속도에 따라 차량 로봇의 선회 각속도가 결정되므로, 그 유효성을 보증하기 위해 실험적으로 추정될 수 있다.

먼저, 실험적으로 차량 로봇의 매개변수 r , c 을 추정하는 작업을 수행한다.

r 은 $0.03m$ 에서 $0.05m$ 까지 $0.001m$ 단위로 증가시키고 c 는 $0.12m$ 에서 $0.15m$ 까지 $0.001m$ 단위로 증가시키면서 시뮬레이션 하여 식 (2.20) 오차의 절대치를 합한 결과를 최소로 하는 유효값(Effective value) r , c 을 선정하였다.

Fig. 2-11 은 r , c , λ 을 다음과 같이 선정한 경우 초음파 위치 시스템과 엔코더를 이용하여 차량 로봇의 위치를 측정한 결과를 보여준다.

- (a) $r = 0.042m$, $c = 0.125m$, $\lambda = 1.1007$,
- (b) $r = 0.041m$, $c = 0.132m$, $\lambda = 1.1740$,
- (c) $r = 0.041m$, $c = 0.134m$, $\lambda = 1.2430$,
- (d) $r = 0.040m$, $c = 0.133m$, $\lambda = 1.3159$,
- (e) $r = 0.042m$, $c = 0.137m$, $\lambda = 1.3831$,
- (f) $r = 0.042m$, $c = 0.138m$, $\lambda = 1.4557$.

Fig. 2-11 의 결과는 실제 초음파 시스템으로 측정한 차량 위치와 엔코더의 정보를 이용하여 식 (2.12)와 식 (2.19)로 계산된 차량 위치를 보여준다. Fig. 2-10 의 결과로부터 왼쪽과 오른쪽 바퀴의 선회비가 다를 경우 r , c 의 최적값 또한 달라지는 것을 확인할 수 있다. 따라서 본 연구에서는 다양한 선회비에 대해 보편적으로 적용할 수 있는 r , c 값을 반복적인 실험을 통하여 $r = 0.0409m$, $c = 0.13194m$ 와 같이 선정하였다. 이렇게 선정된 값은 (b)와 유사한데 이는 차후 경로 추종과 분산 제어 실험에서 선정될 목표 속도값이 (b)와 유사하기 때문이다.

Fig. 2-12 는 선정되어진 $r = 0.0409m$, $c = 0.13194m$ 을 이용하여, Fig. 2-9 의 각 선회비에 대해서 그래프를 그린 것이고 식 (2.20)에 의한 오차는 Fig. 2-13 과 같다. Fig. 2-13 의 결과로부터 엔코더의 정보를 이용하여 식 (2.12)와 식 (2.19)로 계산된 차량 로봇의 위치와 실제 초음파 시스템으로 측정한 차량 로봇의 위치가 비슷하지만 아직까지 약간의 오차를 가지는 것을 알 수 있다. 따라서 이러한 오차를 줄이기 위한 위치 인식 보정 알고리즘을 3 장에서 소개한다.

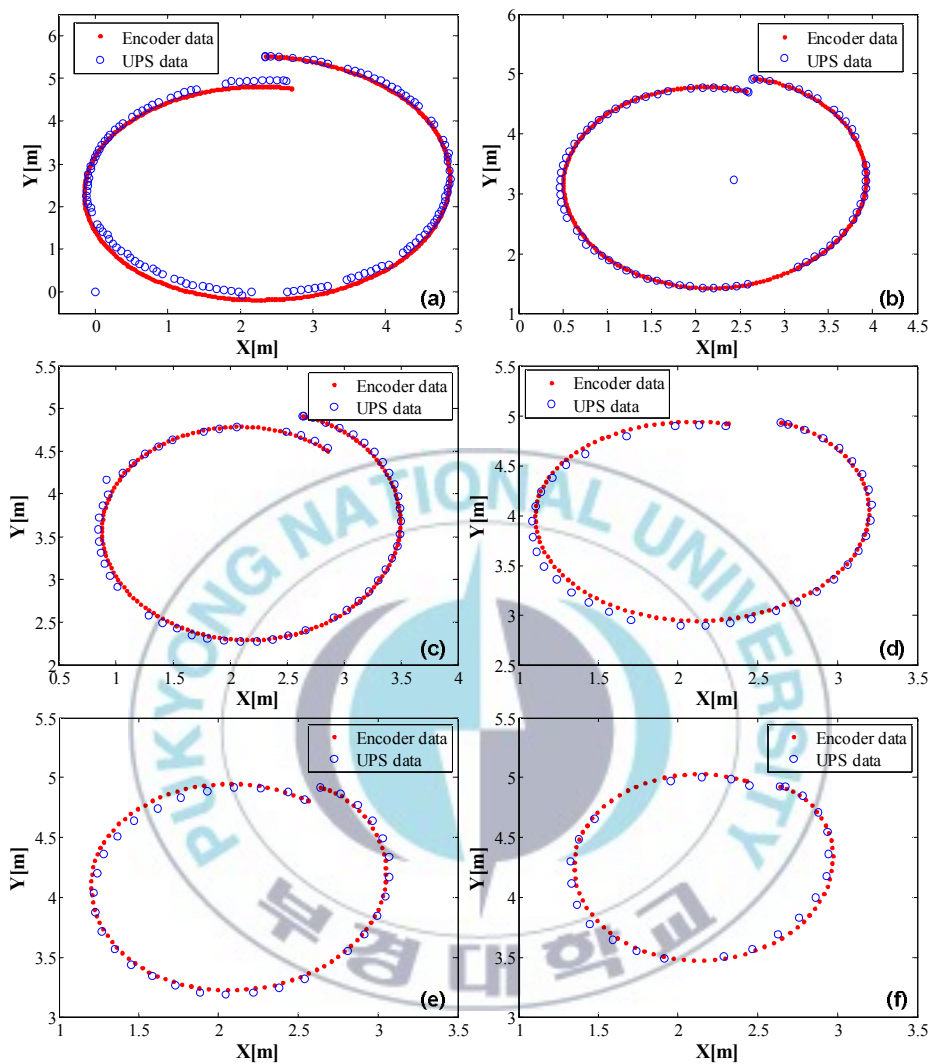


Fig. 2-11 Vehicle movement with optimal r , c

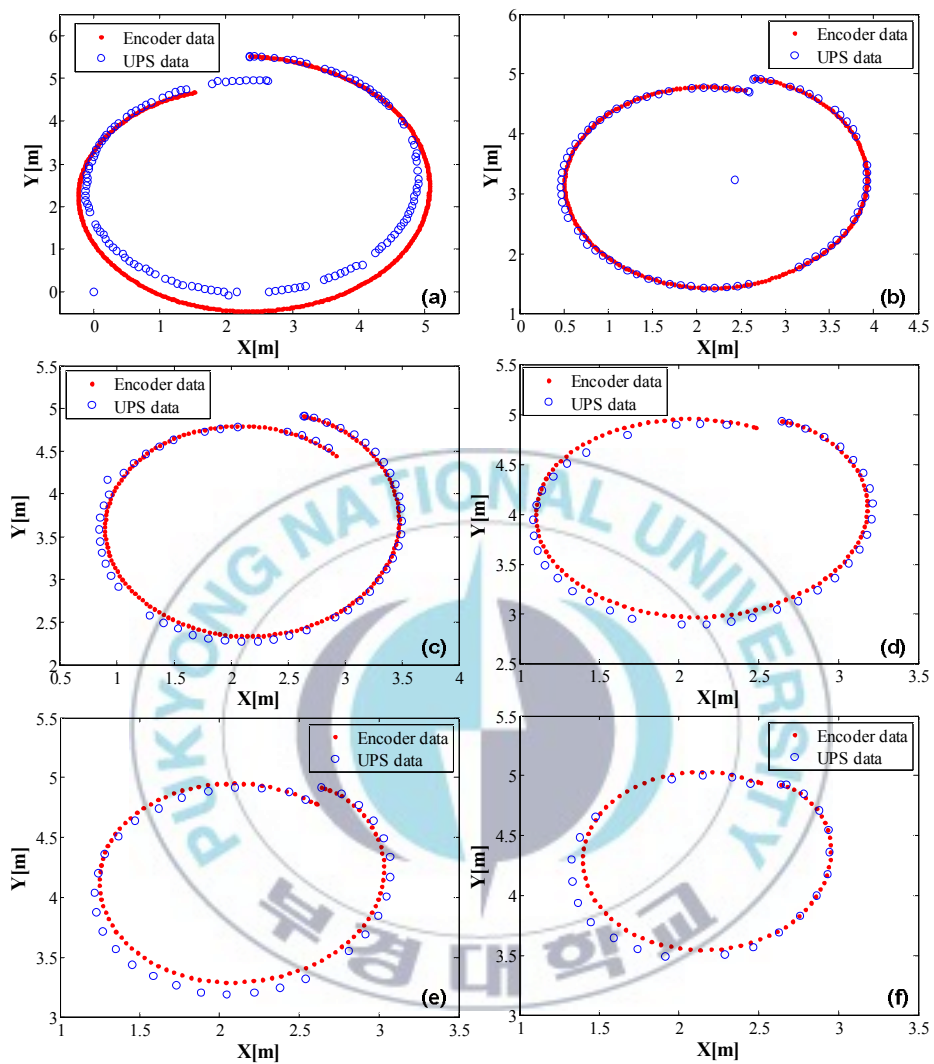


Fig. 2-12 Vehicle movement with $r = 0.0409m$, $c = 0.13194m$

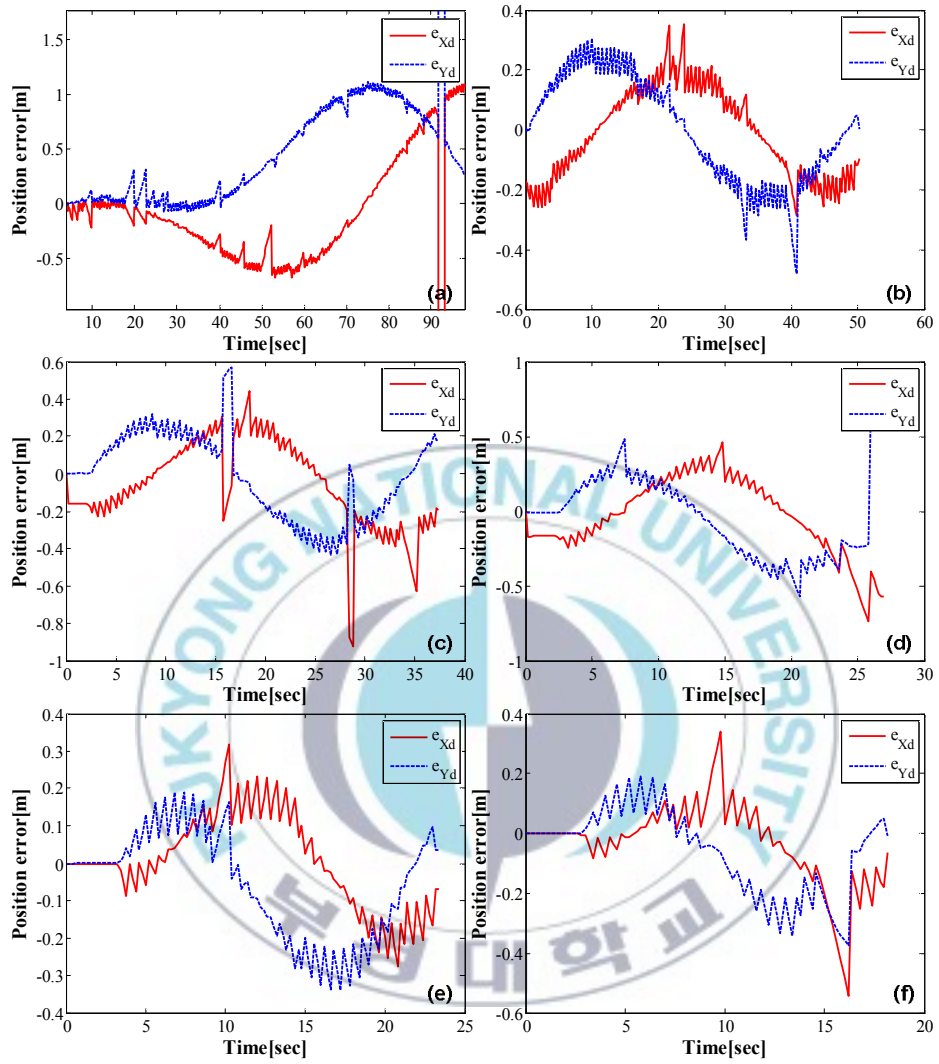


Fig. 2-13 Position error with $r = 0.0409m$, $c = 0.13194m$

제 3 장 로봇의 위치 인식 및 경로 추종

본 장에서는 차량 로봇의 제어에 있어서 가장 기본이 되는 위치 인식 방법을 다룬다. 앞서 언급한 바와 같이, 초음파 위치 시스템은 벽면 반사에 의하여 불규칙한 측정 오차를 가진다. 이로 인하여, 초음파 위치 시스템으로부터의 정보를 차량의 위치로 결정하는 데는 큰 어려움이 따른다. 따라서 차량 로봇의 위치 결정을 위한 위치 인식 알고리즘의 개발이 절실히 필요하다.

본 연구에서는 차량의 양측 바퀴에 부착된 엔코더(Encoder)값을 운동학 방정식에 대입하여 계산된 정보와 초음파 위치 시스템으로부터 측정된 정보 중에서 신뢰성이 높다고 판단된 위치 정보 사이의 가중치를 곱하여 차량의 위치를 결정하는 위치 인식 알고리즘을 제안한다. 제안된 위치 인식 알고리즘을 기반으로, 단일 로봇을 이용한 경로 추종 알고리즘으로 LOS(Line of sight) 알고리즘을 제안한다. 이 LOS 알고리즘은 정해진 경로상의 중간 지점(Waypoint) 정보와 현재 차량의 위치 정보를 통해 순간 목표점을 구하고, 이것을 통해 차량의 순간 움직임을 결정한다. 이 경로 추종 알고리즘은 다음 장에서의 다중 로봇 대형 제어에 있어 선도-추종(Leader-following) 분산 제어 방법에서 선도 차량 로봇의 경로 추적을 위해 적용된다.

3.1 차량 로봇의 위치 인식 알고리즘

위치 판단과 관련하여 일반적으로 3 가지 기술이 주로 사용되는데, 삼각측량법(Triangulation)과 근사치측량법(Proximity) 그리고 분석측량법(Analysis) 방법이 그것이다. 삼각측량법에는 위치를 알고 있는 점들로부터의 정보를 가지고 여러 개의 거리 측정을 사용하는 Lateration 기법이 있고, 위치를 알고 있고 지리적으로 분리되어 있는 점들에 대한 각도나 방향을 측정하여 결정하는 Angulation 기법이 있다. 본 연구에서 사용된 초음파 위치 시스템은 Lateration 기법을 기반으로 하고 있다. Lateration 기법으로 2 차원에서 물체의 위치를 계산하기 위해서는 동일 직선상에 있지 않은 3 개의 점으로부터 거리를 측정해야 한다. 그리고 3 차원에서 물체의 위치를 계산하기 위해서는 동일한 평면상에 있지 않은 4 개의 점으로부터 거리를 측정해야 한다.

본 연구에서 사용된 초음파 위치 시스템은 4 개의 점 역할을 하는 발신기를 이용하여 3 차원 좌표를 인식하게 된다. 그러나 본 연구에서의 차량 로봇은 2 차원 평면에서만 움직인다고 가정하므로 초음파 위치 시스템의 평면 좌표만을 고려한다.

3.1.1 3차원 위치 인식 원리

본 연구에서 사용된 초음파 위치 시스템의 4 대의 발신기 좌표를 각각 $T_i(X_{T_i}, Y_{T_i}, Z_{T_i})$ 라 하고, 위치 측정을 원하는 수신기의 좌표를 $R_r(X_r, Y_r, Z_r)$ 라고 하면, Fig. 3-1 과 같이 수신기와 각 발신기와의 거리 d_1, d_2, d_3, d_4 는 식 (3.1)과 같이 구할 수 있다.

$$\begin{cases} d_1^2 = (X_r - X_{T_1})^2 + (Y_r - Y_{T_1})^2 + (Z_r - Z_{T_1})^2 \\ d_2^2 = (X_r - X_{T_2})^2 + (Y_r - Y_{T_2})^2 + (Z_r - Z_{T_2})^2 \\ d_3^2 = (X_r - X_{T_3})^2 + (Y_r - Y_{T_3})^2 + (Z_r - Z_{T_3})^2 \\ d_4^2 = (X_r - X_{T_4})^2 + (Y_r - Y_{T_4})^2 + (Z_r - Z_{T_4})^2 \end{cases} \quad (3.1)$$

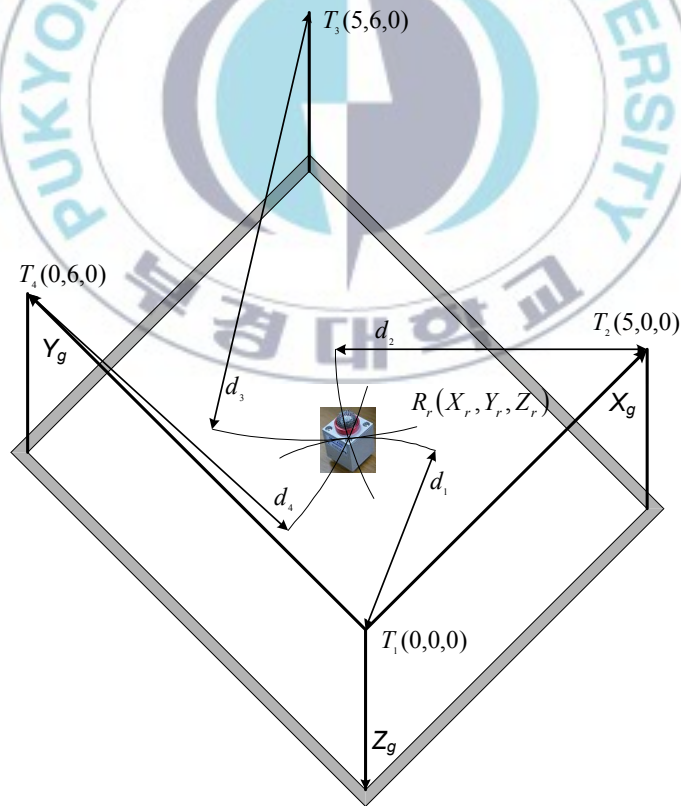


Fig. 3-1 3D position using ultrasonic positioning system

한편 식 (3.1)에서 수신기의 좌표 $R_r(X_r, Y_r, Z_r)$ 을 구하기 위해서는 d_1, d_2, d_3, d_4 값을 구할 필요가 있다. d_1, d_2, d_3, d_4 의 값을 구하는 방법은 수신 신호의 세기(Received signal strength indicator: RSSI)와의 관계를 이용하는 방법과 도착 시간(Time of arrival: TOA) 혹은 도착 시간차(Time difference of arrival: TDOA) 등을 이용하는 방법이 있다.

본 연구에서 사용된 초음파 위치 시스템은 다음과 같은 도착 시간을 이용하는 방법을 적용하였다.

초음파를 이용한 거리측정은 초음파의 전달 속도와 전달시간 정보를 이용한다. 초음파가 공기를 통해 전달되는 시간은 비행시간(Time of flight: TOF)이라고 하며, 발산된 시각 T_t 와 그 신호가 수신된 시각 T_r 의 차이에 의한 값으로 식 (3.2)와 같이 정의된다.

$$TOF = T_r - T_t \quad (3.2)$$

초음파를 이용하여 측정되는 거리 d_i 는 식 (3.3)과 같이 표현할 수 있다.

$$d_i = C \times TOF + d_0 \quad (3.3)$$

여기서 d_0 는 송수신부의 진동판 위치오차로 인한 거리 오프셋(offset)을 나타내며, C 는 음파 속력으로 섭씨온도 T_C 에 대하여 식 (3.4)와 같이 정의된다.

$$C = 331.5 + 0.60714T_C \quad (3.4)$$

이는 초음파의 속도가 온도에 영향을 받아 온도가 높을수록 전파 속도가 빨라짐을 의미한다. 그러므로 초음파를 이용하여 거리를 측정할 경우 온도에 대한 영향도 고려되어야 한다.

본 연구에서는 차량 로봇의 전진방향각과 중심 좌표를 알기 위해서 Fig. 3-2 와 같이 수신기를 차량 로봇의 앞뒤에 하나씩 부착하면, 3 차원 위치 인식 원리에서 구한 수신기 두 개의 위치 좌표 $R_F(X_F, Y_F)$, $R_R(X_R, Y_R)$ 을 이용하여 차량 로봇의 전진방향각 θ 와 중심 좌표 (X, Y) 를 관계식 (3.5)와 (3.6)과 같이 얻어진다. 또한 실제 수신기 사이의 거리는 $d_r = 0.16m$ 이며 d_{re} 는 초음파 위치 시스템 데이터를 이용하여 구한 수신기 사이의 거리로 식 (3.7)과 같다.

$$\theta = \tan^{-1} \left(\frac{Y_F - Y_R}{X_F - X_R} \right) = \sin^{-1} \left(\frac{Y_F - Y_R}{2d} \right) = \cos^{-1} \left(\frac{X_F - X_R}{2d} \right) \quad (3.5)$$

$$\begin{cases} X = \frac{X_F + X_R}{2} \\ Y = \frac{Y_F + Y_R}{2} \end{cases} \quad (3.6)$$

$$d_{re} = \sqrt{(X_F - X_R)^2 + (Y_F - Y_R)^2} \quad (3.7)$$

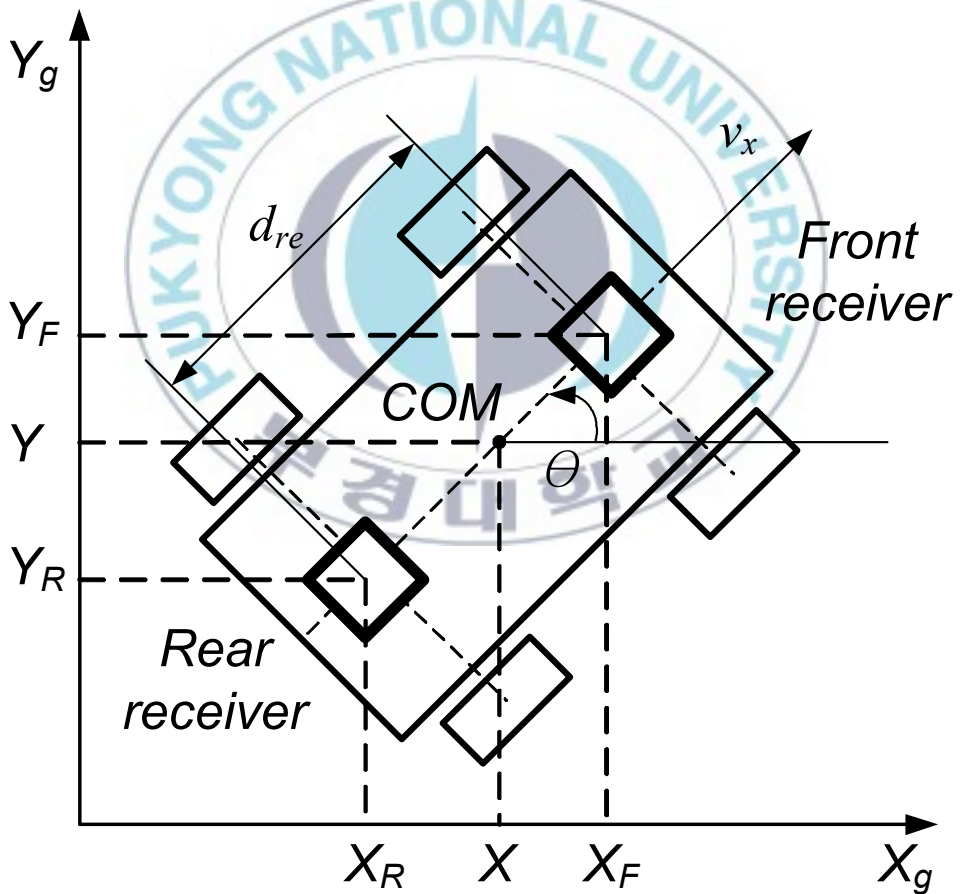


Fig. 3-2 Measurement of COM and heading angle using ultrasonic positioning system

3.1.2 위치 인식 보정 알고리즘

다중 로봇 제어의 경우, 각 차량 로봇의 위치 또는 위치 관계를 알 수 있는 정보를 획득하는 것이 무엇보다 중요하다. 본 연구에서는 각각의 차량 로봇 위치를 측정하기 위해 초음파 위치 시스템을 사용한다.

식 (3.1) - 식 (3.7)을 이용한 초음파 위치 시스템의 같은 원본 데이터는 Fig. 3-3 와 같이 벽면과 바닥 반사 등의 주위 환경에 의해 측정 오차가 발생하여 정확한 차량 로봇의 위치와 전진방향각을 알기가 어렵다. 또한 0.6 초 비교적 긴 샘플링 시간으로 위치를 측정하기 때문에 그 사이 시간에 차량 로봇의 위치와 전진방향각을 알 수가 없다. 그리고 Fig. 2-8 의 결과로부터 엔코더만으로 정확한 위치 인식이 어렵다는 것을 알 수가 있다. 따라서 정확한 차량 로봇의 위치 인식을 위하여 초음파 위치 시스템과 엔코더의 정보를 상호 보완하는 위치 인식 보정 알고리즘이 필요하다.

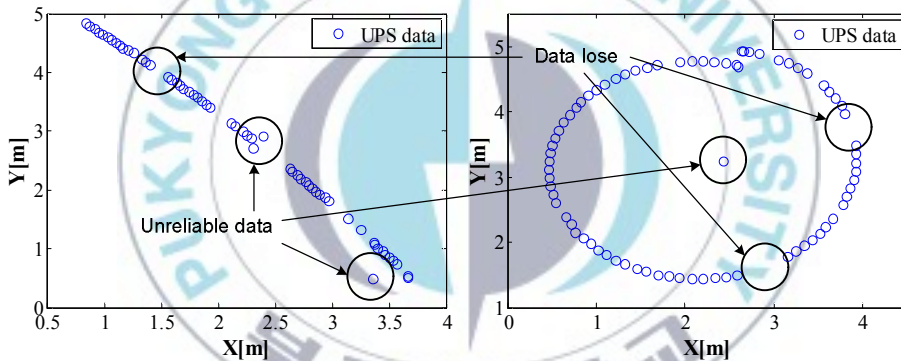


Fig. 3-3 Original data of ultrasonic positioning system

초음파 위치 시스템(UPS) 데이터는 식 (3.1) - 식 (3.7)을 이용하여 매 0.6 초 간격으로 차량 로봇의 위치와 전진방향각 정보를 얻는다. 그러나 여기에는 측정 시간 간격이 길고 신뢰할 수 없는 데이터가 비주기적으로 들어오는 문제점이 있다. 이를 해결하기 위하여 본 연구에서는 차량 로봇의 엔코더(Encoder) 데이터로 ω_R , ω_L 을 구하고, 2장에서 유도된 운동학적 모델링 식 (2.12)과 식 (2.19)를 이용하여 매 0.1 초 간격으로 얻어지는 차량 로봇의 위치와 전진방향각을 계산한다. 이 계산된 값과 신뢰할 수 있다고 판단된 초음파 위치 시스템 정보에 가중치를 가함으로써 최종적으로 차량의 위치를 결정한다.

Fig. 3-4 는 본 연구에서 제안하는 위치 인식 보정 알고리즘을 간략하게 설명한 것이다.

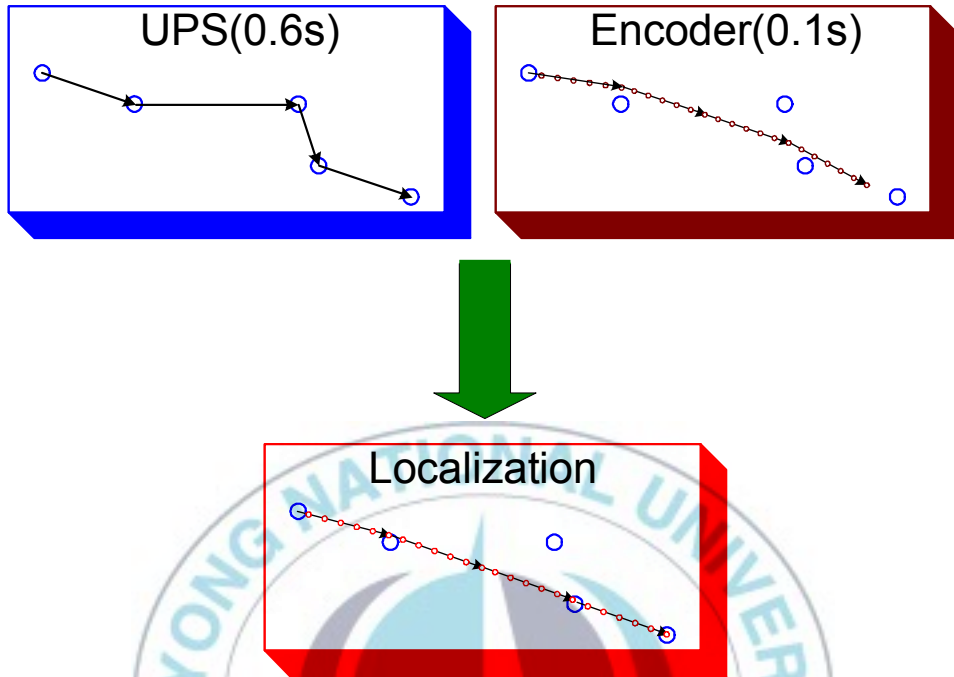


Fig. 3-4 Compensation algorithm in localization of vehicle robot

Fig. 3-4의 알고리즘을 구현하기 위해서는 초음파 위치 시스템에서 Fig. 3-3와 같이 정확한 위치 데이터가 들어오지 않을 수도 있기 때문에, 초음파 위치 시스템의 데이터의 신뢰성 판단 여부를 결정하는 기준이 필요하게 된다. 초음파 위치 시스템 데이터의 신뢰성 판단 기준은 다음과 같다.

- (1) 이전에 수신된 신뢰할 수 있는 데이터와의 시간 간격을 구하고, 그 동안 이동할 수 있는 최대 거리를 계산하여 그 범위 안에 있는지 판단하여 이를 만족하는 데이터만 인정한다. 이동할 수 있는 최대 거리 d_m 은 식 (3.8)과 같다.

$$\begin{cases} d_m = 1.2v_a t_s n_s \\ v_a = \sum_{i=1}^5 v_{k-i} / 5 \end{cases} \quad (3.8)$$

여기서 v_a 는 과거 속도의 산술 평균이며, t_s 는 시스템을 제어하는 주기를 나타낸 것으로 0.1sec이고 n_s 는 신뢰할 수

있는 초음파 위치 시스템의 데이터들 사이의 제어 주기 개수이다. k 는 현 시점의 제어 주기이고 v_{k-i} 는 과거 i 번째 제어 주기의 속도이다. 1.2는 0.2의 허용 오차를 위한 값이다.

- (2) 차량에 장착된 두 개의 초음파 위치 시스템 수신기의 간격은 일정하므로, 식 (3.9)와 같이 그 간격에서 벗어난 정도 d_i 를 판단하여 오차 범위 내의 데이터만 인정한다. 0.2는 허용 오차이다.

$$d_i = |d_r - d_{re}| < 0.2d_{re} \quad (3.9)$$

위의 기준에 부합하는 초음파 위치 시스템으로 측정한 위치 벡터 $\mathbf{q}_{ups} = [X_{ups} \ Y_{ups} \ \theta_{ups}]^T$ 와 엔코더로 측정한 위치 벡터 $\mathbf{q}_{en} = [X_{en} \ Y_{en} \ \theta_{en}]^T$ 의 데이터 사이에 식 (3.11)의 가중치 행렬 $\mathbf{M}_w$, $\mathbf{N}_w$ 을 가하여 최종적으로 차량 로봇의 위치벡터 $\mathbf{q} = [X \ Y \ \theta]^T$ 를 식 (3.10)와 같이 구하게 된다.

$$\begin{cases} \mathbf{q} = \mathbf{M}_w \mathbf{q}_{ups} + \mathbf{N}_w \mathbf{q}_{en} \\ 1 = \mathbf{M}_w + \mathbf{N}_w \end{cases} \quad (3.10)$$

$$\mathbf{M}_w = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1 & 0 \\ 0 & 0 & m_2 \end{bmatrix}, \quad \mathbf{N}_w = \begin{bmatrix} n_1 & 0 & 0 \\ 0 & n_1 & 0 \\ 0 & 0 & n_2 \end{bmatrix} \quad (3.11)$$

여기서 m_1 , m_2 는 각각 초음파 위치 시스템 데이터의 위치와 전진방향각에 대한 가중치이고, n_1 , n_2 는 각각 엔코더 데이터의 위치와 전진방향각에 대한 가중치이다.

Fig. 3-5는 초음파 위치 시스템과 엔코더에서 획득한 회전속도 데이터로부터 위치 인식 알고리즘을 구현하는 과정을 표현한 순서도이다.

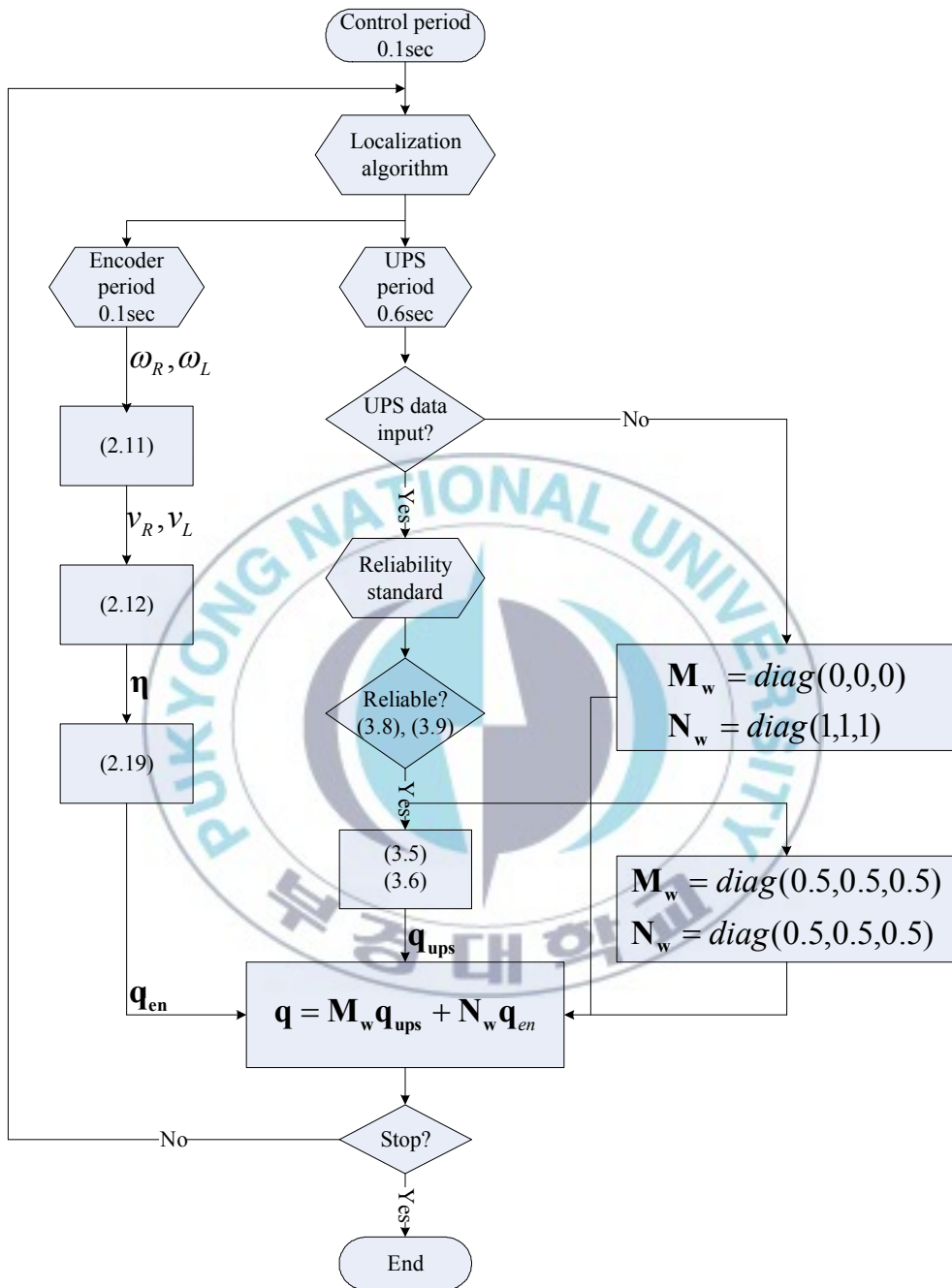


Fig. 3-5 Flow chart of localization algorithm

3.1.3 알고리즘 성능 검증 결과

제안된 알고리즘의 성능을 검증하기 위하여 차량을 자율 주행시킨 후 위치 측정 센서로부터 계측된 값과 알고리즘을 통해 계산된 결과를 비교한다.

Fig. 3-6 은 차량의 자율 주행 실험을 통하여 얻어진 차량 위치와 위치 인식 알고리즘을 적용하여 얻어진 차량 로봇의 위치를 보여준다. 차량의 자율 주행 실험을 통하여 얻어진 위치 인식 알고리즘의 결과를 나타낸다.

- (a) 2장에서 선정한 최적의 $r = 0.0409m$, $c = 0.13194m$ 을 운동학적 모델링 식 (2.12)에 대입하여 얻어진 결과로써 시간 경과에 따라 오차가 크게 발생하는 것을 알 수 있다. 여기서, 가중치는 $\mathbf{M}_w = \text{diag}(0,0,0)$, $\mathbf{N}_w = \text{diag}(1,1,1)$ 이다.
- (b) (a)의 방법에 신뢰성 판단을 통해 인정된 데이터로부터, 전진방향각 정보만을 이용한 것이다. 여기서, 가중치는 $\mathbf{M}_w = \text{diag}(0,0,0.5)$, $\mathbf{N}_w = \text{diag}(1,1,0.5)$ 이다.
- (c) (a)의 방법에 신뢰성 판단을 통해 인정된 데이터로부터, 위치 정보만을 이용한 것이다. 여기서, 가중치는 $\mathbf{M}_w = \text{diag}(0.5,0.5,0)$, $\mathbf{N}_w = \text{diag}(0.5,0.5,1)$ 이다.
- (d) (a)의 방법에 신뢰성 판단을 통해 인정된 데이터로부터, 전진방향각과 위치 정보 모두 이용한 것이다. 여기서, 가중치는 $\mathbf{M}_w = \text{diag}(0.5,0.5,0.5)$, $\mathbf{N}_w = \text{diag}(0.5,0.5,0.5)$ 이다.
- (e) (a)의 방법에 신뢰성 판단을 통해 인정된 데이터로부터, 전진방향각과 위치 정보 모두 이용한 것이다. 여기서, 가중치는 $\mathbf{M}_w = \text{diag}(1,1,1)$, $\mathbf{N}_w = \text{diag}(0,0,0)$ 이다.
- (f) $\mathbf{M}_w = \text{diag}(1,1,1)$, $\mathbf{N}_w = \text{diag}(0,0,0)$ 의 가중치를 가지는 (e)의 그래프에서 그려진 동그라미를 확대한 부분으로 끊어짐 현상이 보인다. 끊어짐 현상은 차량 로봇의 좌표가 갑자기 변할 뿐만 아니라 전진방향각을 알기 어렵게 만든다. 따라서 적절한 가중치를 이용하여 끊어짐 없는 방법으로 위치 인식하는 방법이 필요하다.

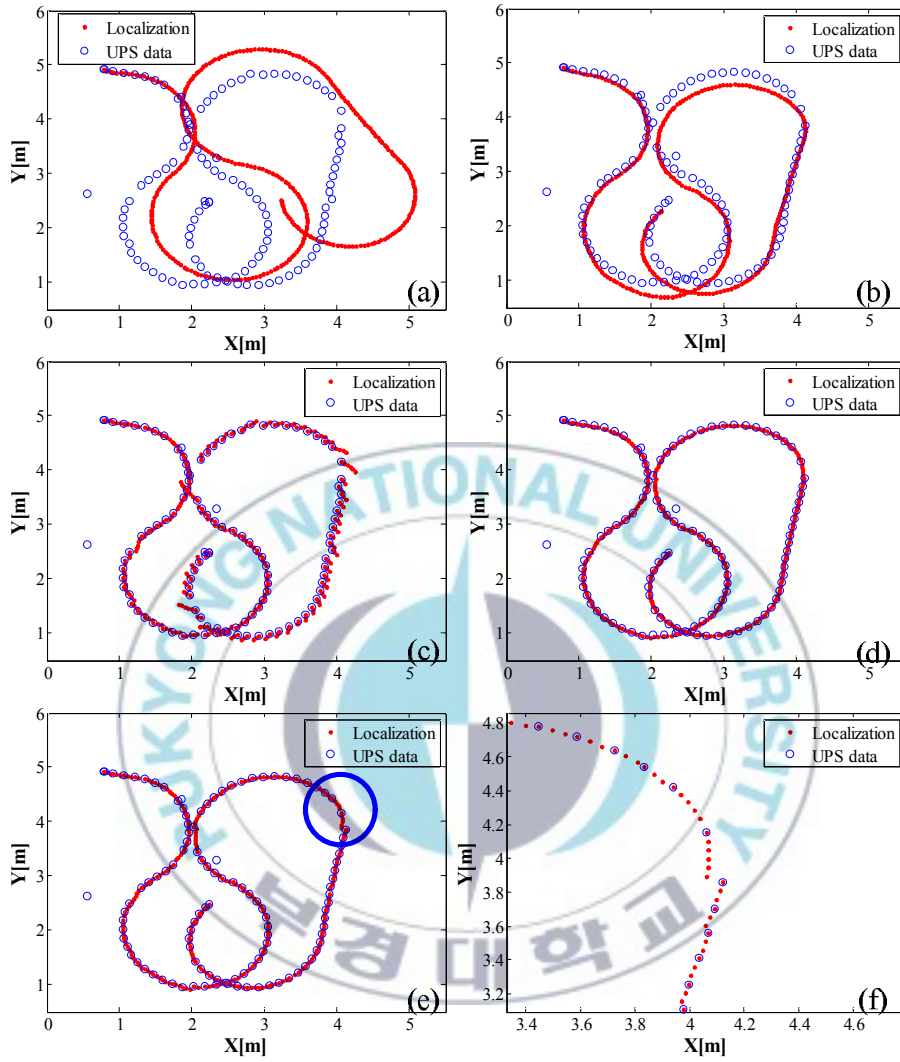


Fig. 3-6 Experimental result of proposed localization algorithm

Fig. 3-6 의 그래프 결과로부터, 가중치를 $\mathbf{M}_w = \text{diag}(0.5, 0.5, 0.5)$, $\mathbf{N}_w = \text{diag}(0.5, 0.5, 0.5)$ 와 같이 가지도록 하는 (d) 방법이 가장 적절하다는 것을 알 수 있다.

3.2 경로 추종 알고리즘

3.2.1 LOS(Line of Sight) 알고리즘

차량 로봇이 목표 경로를 추종하기 위해서 본 연구에서는 일반적으로 널리 사용되고 있는 LOS(Line of sight) 개념을 적용하였다. 이러한 LOS 경로 추종 알고리즘은 다중 차량 로봇의 대형 제어에서 선도-추종 분산 제어의 선도 차량 로봇이 목표 경로를 추종하기 위해 사용된다.

원하는 차량 로봇의 경로는 Fig. 3-7 과 같은 다수의 경로점(Way point)의 집합으로 이루어진다고 가정하며, 목표 경로의 시작점은 P_1 이고 최종 목표점은 P_r 이다. 그리고 차량 로봇의 목표 지향점 $P_{los}(x_{los}, y_{los})$ 와 목표 지향각 ϕ_d 는 LOS 벡터로 결정된다. 또한, 각각의 경로점에서 차량 로봇이 원하는 전진방향 속도 v_n 가 결정되는데, 본 연구에서는 v_n 값이 일정하다고 가정한다.

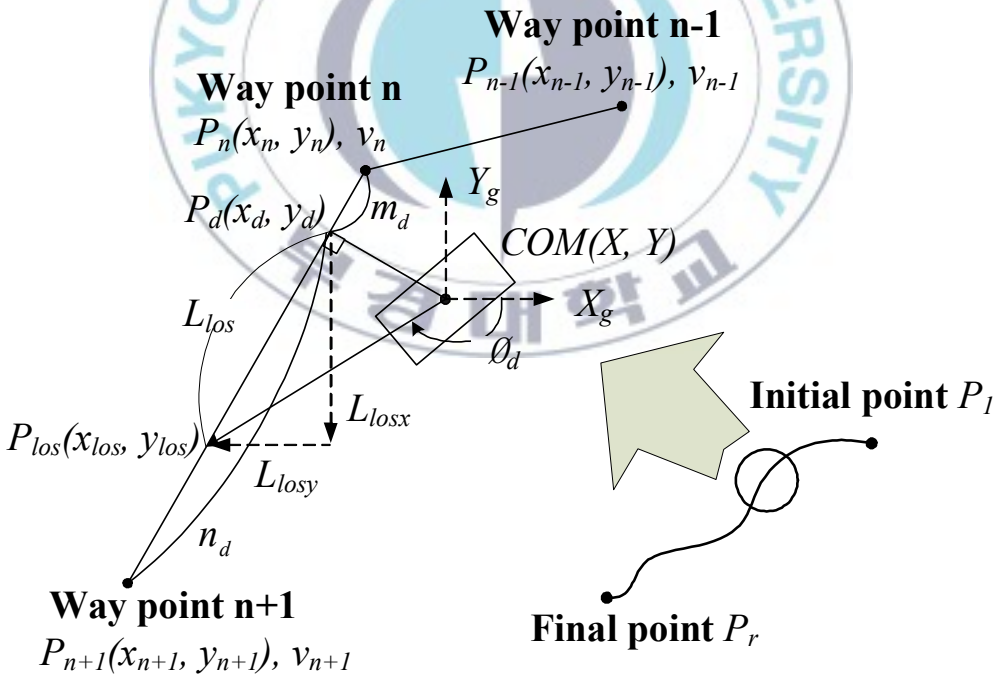


Fig. 3-7 Way point and LOS concept

현재 차량 로봇의 위치가 $COM(X, Y)$ 일 때 차량의 위치와 가장 가까운 경로상의 위치 $P_d(x_d, y_d)$ 가 $\overline{P_n P_{n+1}}$ 을 $m_d : n_d$ 으로 내분한 점이라 하면 벡터의 내적을 찾는 식 (3.11)을 이용해서 식 (3.12)와 같이 구할 수 있다. (부록 A 증명 1 참조)

$$u = \frac{m_d}{m_d + n_d} = \frac{(X - x_n)(x_{n+1} - x_n) + (Y - y_n)(y_{n+1} - y_n)}{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad (3.11)$$

$$\begin{cases} x_d = \frac{m_d x_{n+1} + n_d x_n}{m_d + n_d} = x_n + u(x_{n+1} - x_n) \\ y_d = \frac{m_d y_{n+1} + n_d y_n}{m_d + n_d} = y_n + u(y_{n+1} - y_n) \end{cases} \quad (3.12)$$

식 (3.11)에서 u 는 0 과 1 사이의 값의 내분율이며 이 값이 1 에 가까우면 $P_d(x_d, y_d)$ 가 $P_{n+1}(x_{n+1}, y_{n+1})$ 에 가까이 있음을 선형적으로 나타내는 매개변수다.

지향점 거리 L_{los} 을 식 (3.13)과 같이 정의하면 최종 목표 지향점 $P_{los}(x_{los}, y_{los})$ 과 최종 목표 지향각 ϕ_d 는 각각 식 (3.14)과 (3.15)를 이용하여 구할 수 있다. 그리고 최종 목표 지향점 $P_{los}(x_{los}, y_{los})$ 가 n 번째 경로점과 $n+1$ 번째 경로점 상에 없을 경우 다음 경로점인 $n+1$ 번째 경로점과 $n+2$ 번째 경로점 상에서 찾아 위와 같이 반복 수행하며, 차량 로봇의 좌표 $COM(X, Y)$ 가 최종 목표점 $P_r(X_r, Y_r)$ 에 도달하면 정지한다.

$$L_{los} = \sqrt{L_{losx}^2 + L_{losy}^2} \quad (3.13)$$

$$(x_{los}, y_{los}) = (x_d, y_d) + (L_{losx}, L_{losy}) \quad (3.14)$$

$$\phi_d = \tan^{-1}\left(\frac{y_{los} - Y}{x_{los} - X}\right) \quad (3.15)$$

여기서 (L_{losx}, L_{losy}) 는 $P_d(x_d, y_d)$ 에서 $P_{n+1}(x_{n+1}, y_{n+1})$ 까지 X_g, Y_g 방향으로의 증분 좌표값이다.

3.2.2 시뮬레이션 및 실험 결과

일반적으로 차선변경과 선회만으로 대부분의 차량 로봇의 움직임을 표현할 수 있기 때문에, 본 연구에서는 두 가지 경로에 대해서 시뮬레이션과 실험을 실시한다.

먼저, LOS 알고리즘에 적용을 위해서는 차량 로봇의 전진 속도 v_n 과 지향점 거리 L_{los} 을 적절하게 선정하는 것이 무엇보다 중요하다. 본 연구에서는 경로 실험에서 목표 경로와 실제 차량 로봇이 움직인 경로의 거리 오차를 식 (3.16)라고 정의했을 때, v_n 는 $0.05m/s$ 에서 $0.2m/s$ 까지 $0.0001m/s$ 단위로 증가시키고 L_{los} 는 $0.1m$ 에서 $0.3m$ 까지 $0.01m$ 단위로 증가시키면서 시뮬레이션하여 오차의 절대치를 합한 결과를 최소로 하는 v_n 과 L_{los} 을 선정하였다.

$$\begin{cases} e_{xd} = x_d - X \\ e_{yd} = y_d - Y \end{cases} \quad (3.16)$$

Fig. 3-8 은 P_d 점과 COM의 움직임을 보이는 그래프이고, 각 그래프에 주어진 매개변수는 아래와 같다. Fig. 3-9 는 Fig. 3-8 의 각 그래프의 식 (3.16)으로 표현되는 오차 그래프이다.

- (a) $L_{los} = 0.15m$, $v_n = 0.1234m/s$
- (b) $L_{los} = 0.20m$, $v_n = 0.1158m/s$
- (c) $L_{los} = 0.30m$, $v_n = 0.1153m/s$
- (d) $L_{los} = 0.40m$, $v_n = 0.2391m/s$

Fig. 3-8 과 Fig. 3-9로부터 오차 값은 (b)가 가장 작고, 지속적인 시뮬레이션을 통해 L_{los} 과 v_n 의 최적값으로 $L_{los} = 0.18m$, $v_n = 0.1237m/s$ 을 선정하였다.

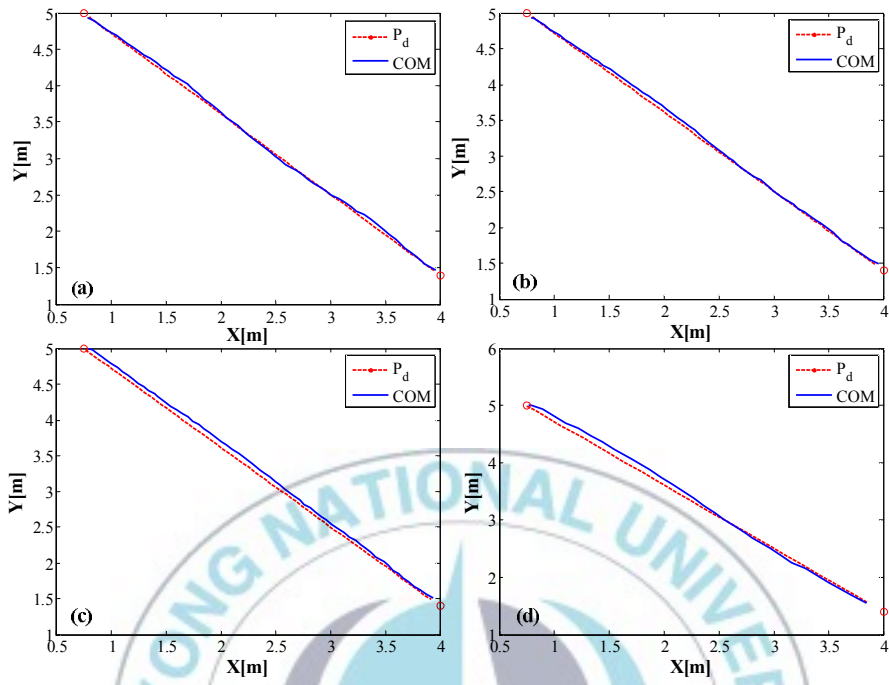


Fig. 3-8 Way point and LOS concept

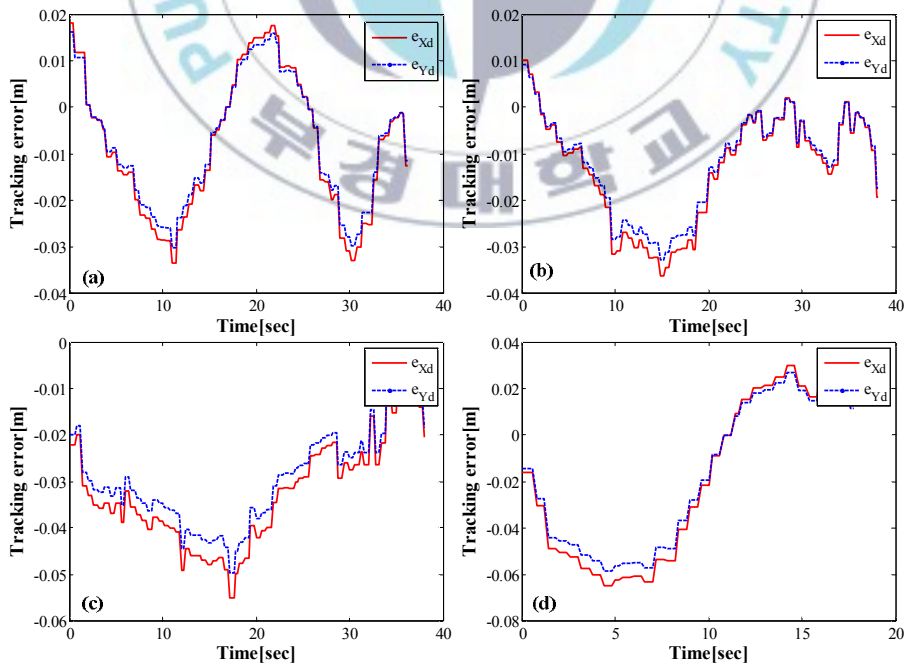


Fig. 3-9 Way point and LOS concept

Fig. 3-10 과 11 은 각각 차선변경 경로에 대한 시뮬레이션과 실험 결과이다. 시작 위치는 $X = 1.0065m$, $Y = 5.571m$, $\theta = -1.5684rad$ 이다. 시뮬레이션과 실험 결과 모두 목표 경로를 잘 추종하는 것을 볼 수 있다. 경로 실험에서 목표 경로와 실제 차량 로봇이 움직인 경로의 거리 오차를 식 (3.16)과 같이 정의하면, 그 결과는 Fig. 3-12 과 같다. (a)시뮬레이션 결과는 $\pm 0.025m$ 이내의 오차 범위를 가지며, (b)실험 결과는 $\pm 0.04m$ 이내의 오차 범위에 있음을 알 수 있다.

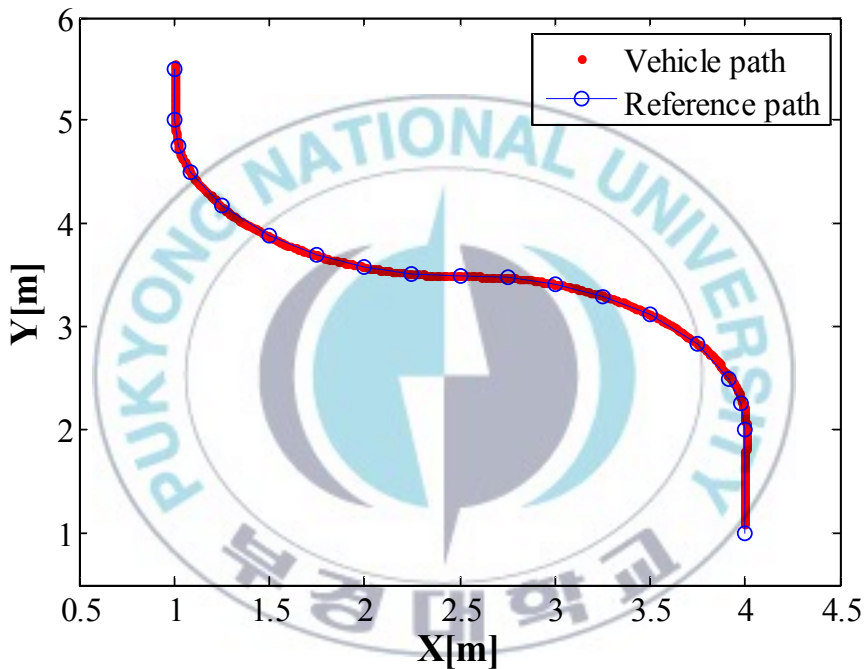


Fig. 3-10 Tacking movement of lane change in LOS simulation

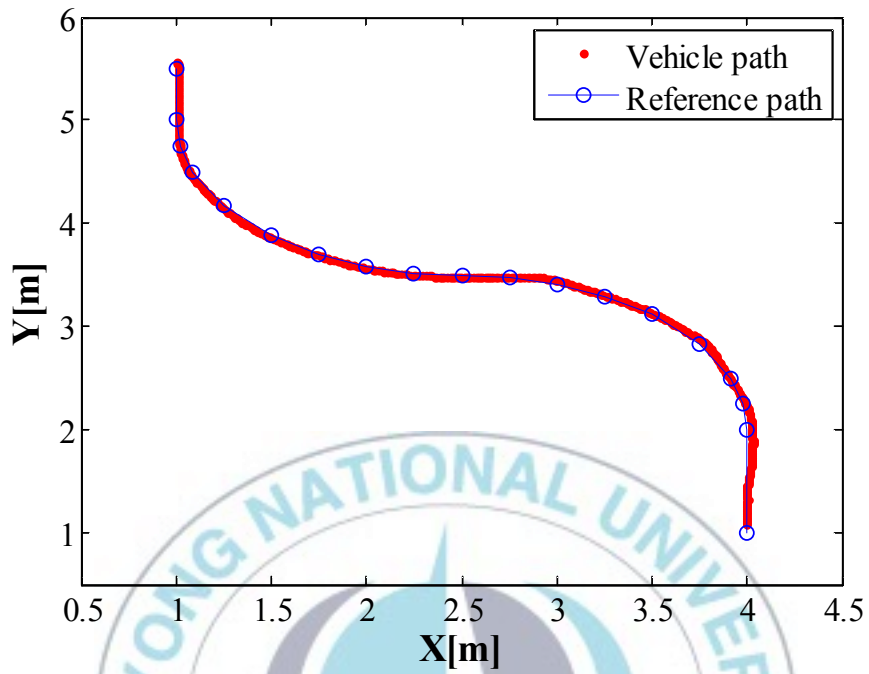
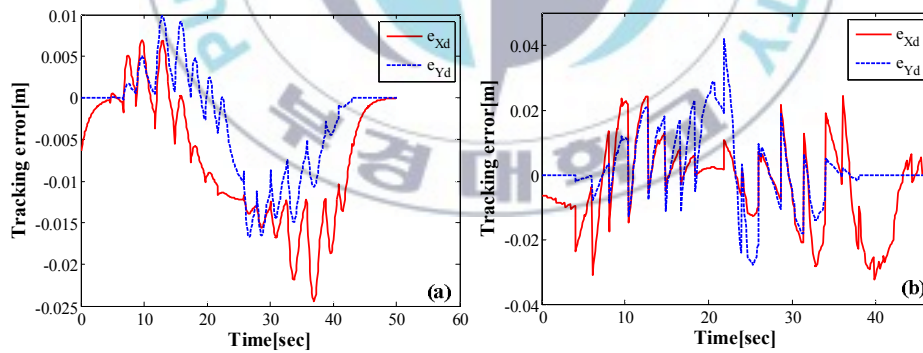


Fig. 3-11 Tacking movement of lane change in LOS experiment



(a) Simulation result

(b) Experiment result

Fig. 3-12 Tracking error of lane change

Fig. 3-13 과 14 는 각각 유턴 경로에 대한 시뮬레이션과 실험이다. 시작위치는 $X = 0.9235m$, $Y = 4.0045m$, $\theta = 0.0091rad$ 이다. 그래프로부터 시뮬레이션과 실험 결과 모두 목표 경로를 잘 추종하는 것을 볼 수 있다. 유턴 경로에 대한 식 (3.16)의 오차 결과는 Fig. 3-15 과 같으며 (a)시뮬레이션 결과는 $\pm 0.02m$ 이내의 오차 범위를 가지며 (b)실험 결과는 $\pm 0.04m$ 이내의 오차 범위에 있음을 알 수 있다.

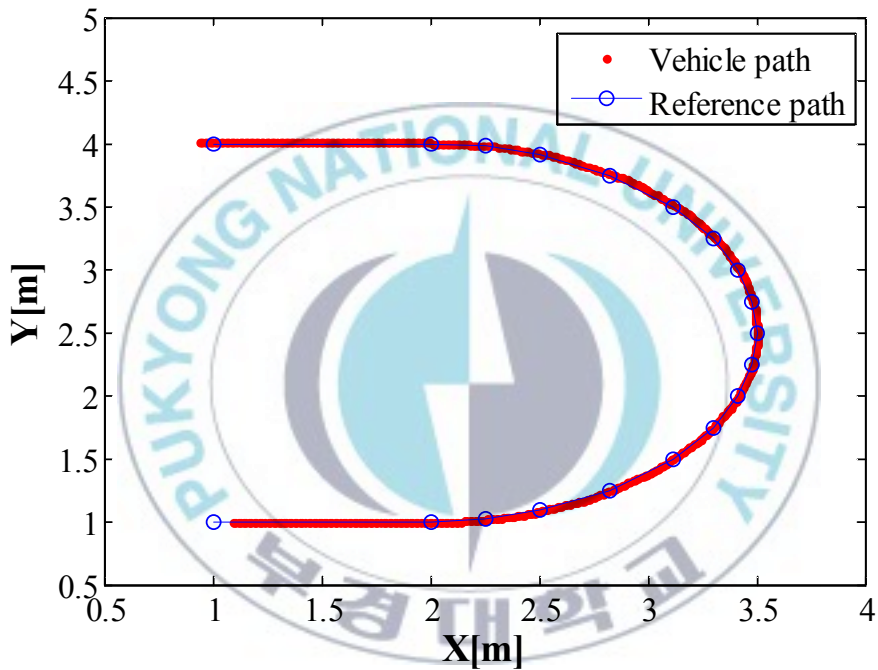


Fig. 3-13 Tacking movement of U-turn in LOS simulation

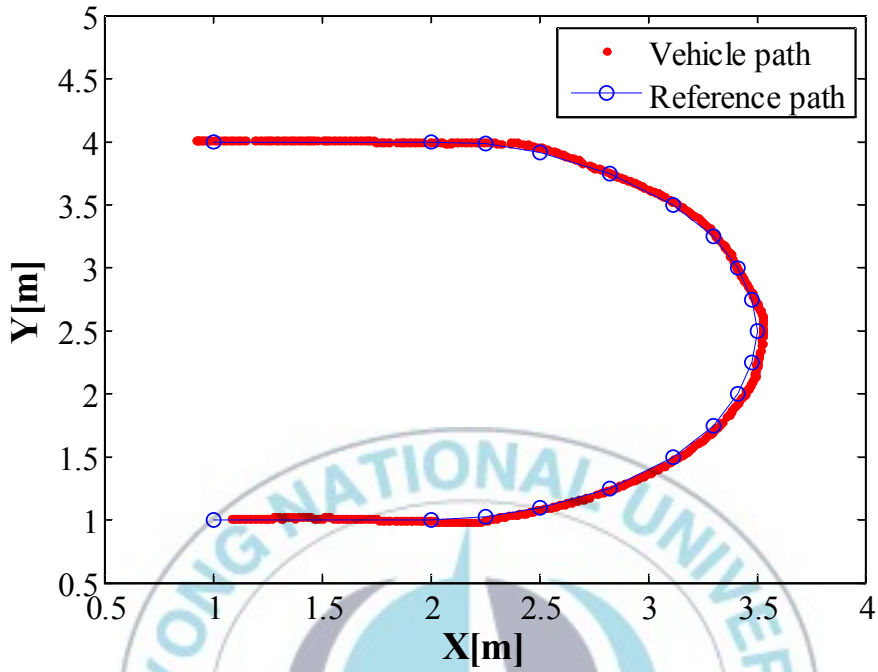
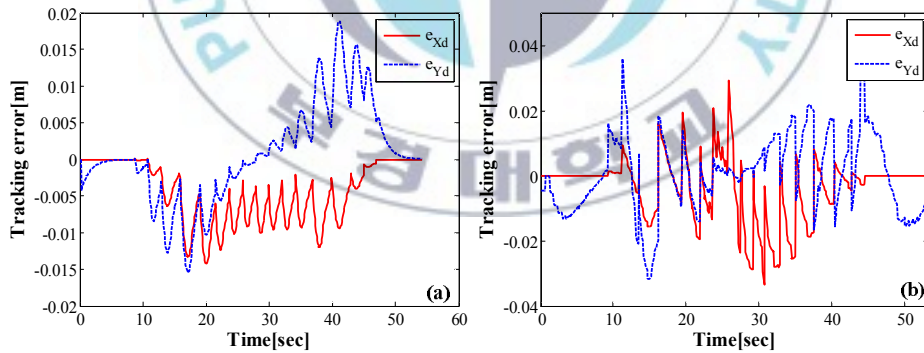


Fig. 3-14 Tacking movement of U-turn in LOS experiment



(a) Simulation result

(b) Experiment result

Fig. 3-15 Tracking error of U-turn

제 4 장 미지 입력 관측기에 기반한 분산 제어

다중 차량 로봇의 대형 제어는 크게 중앙 제어(Centralized control)와 분산 제어(Decentralized control) 방식으로 나뉘어진다. 중앙 제어 방식은 중앙 장치(Centralized Unit)에서 각 차량 로봇의 모든 정보를 계측하거나 수집하여 각 로봇의 제어 입력을 계산한 다음, 그것을 각 로봇에 지령함으로써 제어를 수행하는 방식이다. 중앙 제어 방식이 중앙 장치의 역할이 절대적인 반면, 분산 제어 방식은 그 역할이 비교적 제한적이다. 즉, 중앙 장치의 일부 역할을 각 로봇에 탑재된 로컬 제어기가 나누어서 분담한다.

본 장에서는 앞서 제안된 알고리즘을 활용한 다중 차량 로봇의 대형(Formation) 제어를 위해 미지 입력 관측기를 설계하고, 이를 기반으로 분산 제어기(Decentralized control)를 설계한다. 제안된 미지 입력 관측기를 기반으로 한 분산 제어 방법의 유효성을 검증하기 위하여 시뮬레이션 및 실험을 수행한다.

다중 차량 로봇의 대형 제어를 위해 우선 N 개의 차량 로봇으로 구성된 시스템을 모델링한다. 여기서 다중 차량 로봇은 각각의 제어기를 가지며, 각 제어기는 각 차량 로봇에서 측정된 데이터와 외부 서버로부터 전송 받은 정보를 이용하여 각 차량 로봇에 필요한 제어 신호를 생성하게 된다. 여기서 서버는 모든 차량 로봇의 정보를 알고 있으며, 전체 대형 수준 함수(Formation level function)만을 각 차량 로봇으로 전송한다. 그리고 각 차량 로봇의 제어기는 다른 차량 로봇에 대한 정보를 알 수가 없으므로, 다른 차량 로봇에 대한 정보가 미지 입력이 된다. 따라서 외란 등과 같은 미지 입력이 포함된 시스템의 상태를 관측할 수 있는 미지 입력 관측기(Unknown input observer)를 각 차량 로봇의 대형 제어기에 적용하여 다른 차량 로봇에 대한 정보를 추정 가능하도록 하였다.

본 연구에서는 다중 차량 로봇의 대형 제어를 위해 대형 제어의 최소 단위인 2 대의 차량 로봇을 대형 제어한다. 그리고 대형 제어를 위한 방법으로 분산 제어 방법을 채택하였으며, 중심 역할을 하는 시스템의 기능에 따라 두 가지 분산 제어 방식, 즉 대형 수준 함수와 외부 계측 시스템을 이용하는 Broadcasting 분산 제어 방식과 선도-추종(Leader-following) 분산 제어 방식을 각각 제안한다.

첫 번째 Broadcasting 방법은 미지 입력 관측기(Unknown input observer)를 기반으로 하는 분산 제어 방식으로, 이는 정의된 대형 수준 함수를 감독 장치(Supervisor unit)에서 계산한 후 Broadcasting 통신 방법을 이용하여

각 차량 로봇으로 대형 수준 함수 정보를 전달하는 방법이다. 각 차량 로봇은 대형 수준 함수 정보와 자체적으로 측정한 정보만을 활용하여 미지 입력 관측기로부터 시스템 전체의 상태를 추정한다. 미지 입력 관측기에 의해 추정된 상태에 기반하여 극배치법으로 각 차량 로봇의 상태 피드백 이득을 구하여 제어기를 설계한다. 그리고 각 차량 로봇에 설계된 제어기의 유효성을 검증하기 위해 시뮬레이션과 실험 결과를 제시한다.

두 번째는 선도-추종 방법에 의한 대형 제어 방법이다. 선도 차량 로봇은 주어진 경로를 일정한 속도로 움직이기 위해서 앞서 제안된 경로 추정 알고리즘으로부터 추적 오차를 최소로 하는 제어기 입력을 구한다. 그리고 추종 차량 로봇이 선도 차량 로봇과 일정한 대형을 유지하도록 제어하기 위해서, Broadcasting 방식과 동일한 미지 입력 관측기를 기반으로 하는 분산 제어를 설계한다.

4.1 미지 입력 관측기 기반 분산 제어

4.1.1 대형에서의 다중 로봇 그룹 모델링

본 절에서는 다중 차량 로봇의 대형 제어를 위해 N 개의 차량 로봇으로 구성되어 있는 시스템을 모델링한다. 여기서 다중 차량 로봇은 각각의 제어기를 가지며, 각 제어기는 각 차량 로봇에서 측정된 데이터와 외부 서버로부터 전송 받은 정보를 이용하여 각 차량 로봇에 필요한 제어 신호를 생성한다. N 개의 시스템으로 이루어진 전체 시스템의 상태는 식 (4.1)과 같이 표현된다.

$$\begin{cases} \mathbf{Q} = [\mathbf{X}_\Sigma & \mathbf{Y}_\Sigma & \boldsymbol{\theta}_\Sigma]^T \\ \mathbf{X}_\Sigma = [X_1 & X_2 & \cdots & X_N]^T \\ \mathbf{Y}_\Sigma = [Y_1 & Y_2 & \cdots & Y_N]^T \\ \boldsymbol{\theta}_\Sigma = [\theta_1 & \theta_2 & \cdots & \theta_N]^T \end{cases} \quad (4.1)$$

여기서 $\mathbf{X}_\Sigma$, $\mathbf{Y}_\Sigma$, $\boldsymbol{\theta}_\Sigma$ 는 각각 전체 시스템에 대한 위치 벡터들이며, $\mathbf{q}_i = [X_i \ Y_i \ \theta_i]^T$ 는 i 번째 차량 로봇의 좌표와 전진방향각 정보를 가지는 일반화 좌표 벡터이다. 제어 입력을 식 (4.2)와 같이 표현하면 $\boldsymbol{\eta}_i = [v_{xi} \ \omega_i]^T$ 는 차량의 x_i 방향 전진속도와 회전 각속도를 성분으로 하는 i 번째 차량 로봇의 보조속도벡터이며, 여기서 $i = 1, 2, \dots, N$ 이다.

$$\begin{cases} \boldsymbol{\eta}_{\Sigma} = [\mathbf{v}_{\Sigma} \quad \boldsymbol{\omega}_{\Sigma}]^T \\ \mathbf{v}_{\Sigma} = [v_{x1} \quad v_{x2} \quad \cdots \quad v_{xN}]^T \\ \boldsymbol{\omega}_{\Sigma} = [\omega_1 \quad \omega_2 \quad \cdots \quad \omega_N]^T \end{cases} \quad (4.2)$$

$\mathbf{v}_{\Sigma}$, $\boldsymbol{\omega}_{\Sigma}$ 는 각각 차량의 x_i 방향 전진속도 벡터와 회전 각속도 벡터이다.

대형의 특성은 외부에서 관측될 수 있으며 차량 로봇의 평균 위치와 같은 특성들은 행동 기준(Action reference), 동적 좌표 변수(Dynamic coordination variables) 그리고 대형 수준 함수와 같이 표현된다. 전체 대형 수준 함수는 $\mathbf{h}(\mathbf{Q})$ 으로 표현되며, 선형으로 미분 가능하고, 모든 로봇에 전송 가능하다.

N 개의 차량 로봇으로 구성된 그룹의 대형을 제어하는데 있어, 상태 변수와 대형 수준 함수는 전체 목적에 대한 모든 시스템의 상태를 나타낸다. 전체 차량 로봇의 대형에 관한 시스템 모델은 다음과 같이 표현된다.

$$\begin{cases} \dot{\mathbf{S}}(\mathbf{t}) = \mathbf{A}\mathbf{S}(\mathbf{t}) + \mathbf{B}\mathbf{u}(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) = \mathbf{C}\mathbf{S}(\mathbf{t}) \end{cases} \quad (4.3)$$

여기서, $\mathbf{S}(\mathbf{t}) = [\boldsymbol{\theta}_{\Sigma} \quad \mathbf{h}]^T \in \mathbb{R}^n$ 는 전체 시스템의 상태 변수 벡터이고 $\boldsymbol{\theta}_{\Sigma}$ 와 $\mathbf{h}$ 는 전체 시스템의 상태 변수 벡터의 성분 벡터이며, 각 차량 로봇의 전진방향각 벡터와 전체 시스템의 대형 수준 함수를 나타낸다. $\mathbf{u}(\mathbf{t}) \in \mathbb{R}^m$, $\mathbf{y}(\mathbf{t}) \in \mathbb{R}^r$ 는 입력 벡터와 출력 벡터를 각각 나타낸다. 선형 가능한 조건 하에서, 행렬 $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{r \times n}$ 는 각각 상수 값을 성분으로 하는 시스템 행렬, 입력 행렬, 출력 행렬이다.

만약 $\mathbf{S}(\mathbf{t})$ 의 모든 정보가 중앙 장치뿐만 아니라 각 차량 로봇에서도 이용 가능하다면 중앙 제어 방식이 적용될 수 있다. 그러나 이러한 방법은 시스템의 규모가 커질수록 $\mathbf{S}(\mathbf{t})$ 의 정보 양도 많아 질 것이므로 중앙 제어 장치에 많은 능력을 요구하게 된다. 따라서 전체 차량 로봇의 시스템 모델인 식 (4.3)이 대규모 시스템일 경우에 다음과 같은 분산 제어 방식으로 제어를 설계하는 것이 적합하다.

$$\begin{cases} \dot{\mathbf{S}}(\mathbf{t}) = \mathbf{A}\mathbf{S}(\mathbf{t}) + \sum_{i=1}^N \mathbf{B}_i \mathbf{u}_i(\mathbf{t}) \quad \mathbf{i} = 1, 2, \dots, N \\ \mathbf{y}_i(\mathbf{t}) = \mathbf{C}_i \mathbf{S}(\mathbf{t}) \end{cases} \quad (4.4)$$

여기서 $\mathbf{B}_i \in \mathbb{R}^{n \times m_i}$ 와 $\mathbf{C}_i \in \mathbb{R}^{r_i \times n}$ 는 $\mathbf{B}, \mathbf{C}$ 의 부분 행렬로 각각 i 번째 시스템의 입력 행렬과 출력 행렬이며, $\mathbf{u}_i(\mathbf{t}) \in \mathbb{R}^{m_i}$ 와 $\mathbf{y}_i(\mathbf{t}) \in \mathbb{R}^{r_i}$ 는 i 번째 시스템의 입력 벡터와 출력 벡터이다.

4.1.2 미지 입력 관측기 설계

시스템의 모든 입력을 알고 있을 때 시스템의 상태를 관측하는 관측기를 일반적인 관측기라고 한다면, 시스템이 외란 등과 같은 미지의 입력을 포함할 경우 시스템의 상태를 추정하기 위해서는 미지 입력 항을 고려한 관측기의 설계가 필요하다. 이와 같이 미지 입력 항을 포함하는 시스템에서 상태를 추정하는 관측기를 미지 입력 관측기(Unknown input observer)라고 하며, 본 절에서는 다른 차량 로봇에 대한 정보를 미지의 입력으로 하는 시스템의 미지 입력 관측기 설계 방법을 소개한다.

식 (4.4)와 같은 N 개의 차량 로봇을 가지는 전체 시스템에서, 미지 입력 항 $\mathbf{B}_{\pi}\mathbf{u}_{\pi}(\mathbf{t})$ 을 가지는 i 번째 차량 로봇의 시스템 방정식을 분산 제어 방식으로 표현하면 식 (4.5)와 같다.

$$\sum_{DS} \cdot \begin{cases} \dot{\mathbf{S}}(\mathbf{t}) = \mathbf{A}\mathbf{S}(\mathbf{t}) + \mathbf{B}_i\mathbf{u}_i(\mathbf{t}) + \mathbf{B}_{\pi}\mathbf{u}_{\pi}(\mathbf{t}) \\ \mathbf{y}_i(\mathbf{t}) = \mathbf{C}_i\mathbf{S}(\mathbf{t}) \end{cases} \quad (4.5)$$

여기서, $\mathbf{A} \in \mathbb{R}^{n \times n}$ 는 전체 시스템 행렬이고 $\mathbf{B}_i \in \mathbb{R}^{n \times m_i}$ 와 $\mathbf{B}_{\pi} \in \mathbb{R}^{n \times (m-m_i)}$ 는 각각 i 번째 차량 로봇 시스템의 입력 행렬과 i 번째 차량 로봇을 제외한 다른 차량 로봇들의 입력 행렬을 나타내며, $\mathbf{C}_i \in \mathbb{R}^{r_i \times n}$ 는 i 번째 차량 로봇 시스템의 출력 행렬을 나타낸다. $\mathbf{S}(\mathbf{t}) \in \mathbb{R}^n$ 는 전체 시스템의 상태변수이며, $\mathbf{y}_i(\mathbf{t}) \in \mathbb{R}^{r_i}$ 는 i 번째 차량 로봇 시스템의 출력 벡터이다. $\mathbf{u}_i(\mathbf{t}) \in \mathbb{R}^{m_i}$ 는 i 번째 차량 로봇 시스템의 입력 벡터이고, $\mathbf{u}_{\pi}(\mathbf{t}) \in \mathbb{R}^{m-m_i}$ 는 i 번째 차량 로봇 시스템의 입력 벡터를 제외한 다른 차량 로봇 시스템들의 입력 벡터로 미지의 외란 벡터로 간주된다.

$\sum_{DS}$ 는 다음과 같이 가정한다.

가정 1] $\text{rank} \mathbf{D}_i = m - m_i = q_i$

가정 2] system Σ_{DS} 는 다음 조건을 만족하는 스칼라 λ 가 존재한다.

$$\det(\lambda \mathbf{I} - \mathbf{A}) \neq 0 \quad (4.6)$$

가정 3] system Σ_{DS} 는 R-가관측이다. 즉 다음 조건을 만족한다.

$$\text{rank} \begin{bmatrix} \mathbf{sI}_n - \mathbf{A} \\ \mathbf{C}_i \end{bmatrix} = n \quad (4.7)$$

위의 식 (4.5)에 대한 미지 입력 관측기를 아래 식 (4.8)과 같이 정의한다.

$$\sum_{UIO} : \begin{cases} \dot{\mathbf{z}}_i(\mathbf{t}) = \hat{\mathbf{A}}_i \mathbf{z}_i(\mathbf{t}) + \hat{\mathbf{B}}_i \mathbf{y}_i(\mathbf{t}) + \hat{\mathbf{J}}_i \mathbf{u}_i(\mathbf{t}) \\ \hat{\mathbf{S}}_i(\mathbf{t}) = \hat{\mathbf{C}}_i \mathbf{z}_i(\mathbf{t}) + \hat{\mathbf{D}}_i \mathbf{y}_i(\mathbf{t}) \end{cases} \quad (4.8)$$

여기서, $\hat{\mathbf{S}}_i(\mathbf{t}) \in \mathbb{R}^n$ 와 $\mathbf{z}_i(\mathbf{t}) \in \mathbb{R}^n$ 는 i 번째 차량 로봇 시스템에서 추정되는 상태 벡터와 관측기의 상태 벡터를 각각 나타낸다. $\hat{\mathbf{A}}_i \in \mathbb{R}^{n \times n}$, $\hat{\mathbf{B}}_i \in \mathbb{R}^{n \times r_i}$, $\hat{\mathbf{C}}_i \in \mathbb{R}^{n \times n}$, $\hat{\mathbf{D}}_i \in \mathbb{R}^{n \times r_i}$, $\hat{\mathbf{J}}_i \in \mathbb{R}^{n \times m_i}$ 는 관측기에 관련된 행렬들이다.

정의 1] 위의 식 (4.8)이 다음의 조건을 만족한다면, 식 (4.8)은 식 (4.5)에 대한 미지 입력 관측기라고 한다.

$$\lim_{t \rightarrow \infty} \mathbf{e}_i(\mathbf{t}) = 0, \quad \forall \quad \mathbf{z}_i(0_-), \mathbf{S}(0_-), \mathbf{u}_i(\cdot)$$

여기서 식 (4.9)는 관측기의 추정 오차를 나타낸다.

$$\mathbf{e}_i(\mathbf{t}) = \mathbf{S}(\mathbf{t}) - \hat{\mathbf{S}}_i(\mathbf{t}) \quad (4.9)$$

위의 정의 1 에 의해, 시스템 식 (4.5)와 식 (4.8)사이의 기본적인 관계는 다음의 이론에 의해서 표현된다.

정리 1] 만약, 아래의 식 (4.10)의 조건이 만족하고 식 (4.11)에서 식 (4.14)까지의 관계를 만족하는 행렬 $\mathbf{U}_i \in \mathfrak{R}^{n \times n}$ 가 존재한다면 식 (4.8)은 식 (4.5)에 대한 미지 입력 관측기이다.

$$\lambda_i \begin{bmatrix} \hat{\mathbf{A}}_i \\ \hat{\mathbf{C}}_i \end{bmatrix} < 0, \quad i = 1, \dots, n_i \quad (4.10)$$

$$\hat{\mathbf{A}}_i \mathbf{U}_i + \hat{\mathbf{B}}_i \mathbf{C}_i = \mathbf{U}_i \mathbf{A} \quad (4.11)$$

$$\hat{\mathbf{J}}_i = \mathbf{U}_i \mathbf{B}_i \quad (4.12)$$

$$\mathbf{U}_i \mathbf{D}_i = 0, \quad \mathbf{D}_i = \mathbf{B}_{ii} \quad (4.13)$$

$$\hat{\mathbf{C}}_i \mathbf{U}_i + \hat{\mathbf{D}}_i \mathbf{C}_i = \mathbf{I}_{n_i} \quad (4.14)$$

여기서, $\mathbf{A} \in \mathfrak{R}^{n \times n}$ 와 $\mathbf{D}_i \in \mathfrak{R}^{n \times (m-m_i)}$ 는 행렬이며, $\lambda[\cdot]$ 는 행렬의 고유치(eigenvalue)의 실수부를 나타낸다. (부록 A 증명 2 참조)

미지 입력 관측기를 실현하기 위해, $\hat{\mathbf{C}}_i = \mathbf{I}_n$ 으로 가정하면 식 (4.14)로부터 다음의 식(4.15)가 구해진다.

$$\mathbf{U}_i = \mathbf{I}_n - \hat{\mathbf{D}}_i \mathbf{C}_i \quad (4.15)$$

식 (4.15)를 식 (4.11)에 대입하면, 다음과 같이 식 (4.16)과 식 (4.17)이 계산된다.

$$\hat{\mathbf{A}}_i = \mathbf{U}_i \mathbf{A} - \mathbf{K}_i \mathbf{C}_i \quad (4.16)$$

$$\hat{\mathbf{B}}_i = \hat{\mathbf{A}}_i \hat{\mathbf{D}}_i + \mathbf{K}_i \quad (4.17)$$

여기서, $\mathbf{K}_i = \hat{\mathbf{B}}_i - \hat{\mathbf{A}}_i \hat{\mathbf{D}}_i \in \mathfrak{R}^{n \times r_i}$ 이다.

식 (4.15)를 식 (4.13)에 대입하면 아래의 식 (4.18)이 구해진다.

$$\hat{\mathbf{D}}_i \mathbf{C}_i \mathbf{D}_i = \mathbf{D}_i \quad (4.18)$$

$\text{rank} \mathbf{D}_i = q_i$ 이므로 식 (4.18)을 만족하는 행렬 $\hat{\mathbf{D}}_i$ 을 구하기 위하여, $\text{rank} \mathbf{C}_i \mathbf{D}_i = q_i$ 조건을 반드시 만족하여야 한다. 이는 $\text{rank} \mathbf{C}_i = r_i \geq q_i$ 조건, 즉 출력의 수가 미지 입력의 수 보다 반드시 크거나 같아야만 성립된다
식 (4.18)에 대한 일반 해는 식 (4.19)와 같이 구해진다.

$$\hat{\mathbf{D}}_i = \mathbf{D}_i (\mathbf{C}_i \mathbf{D}_i)^+ + \mathbf{G}_i \left\{ \mathbf{I}_{r_i} - \mathbf{C}_i \mathbf{D}_i (\mathbf{C}_i \mathbf{D}_i)^+ \right\} \quad (4.19)$$

여기서 + 는 일반화된 역행렬(부록 B 정의 1 참조)을 나타내며, $\mathbf{G}_i \in \mathcal{R}^{n \times r_i}$ 는 임의의 적정한 행렬이다. 식 (4.19)를 식 (4.15)에 대입하면, 아래의 식 (4.20)이 얻어진다.

$$\mathbf{U}_i = (\mathbf{I}_n - \mathbf{G}_i \mathbf{C}_i) \left\{ \mathbf{I}_n - \mathbf{D}_i (\mathbf{C}_i \mathbf{D}_i)^+ \mathbf{C}_i \right\} = \mathbf{I}_n - \hat{\mathbf{D}}_i \mathbf{C}_i \quad (4.20)$$

위의 식 (4.20)으로부터 행렬 $(\mathbf{I}_n - \mathbf{G}_i \mathbf{C}_i)$ 을 비특이 행렬(nonsingular matrix)로 만드는 $\mathbf{G}_i$ 가 존재하면, $\text{rank} \left\{ \mathbf{D}_i (\mathbf{C}_i \mathbf{D}_i)^+ \mathbf{C}_i \right\} = q_i$ 이므로 $\text{rank} \mathbf{U}_i = n - q_i$ 이 성립한다.

끝으로 남은 문제는 식 (4.16)을 안정화시키는 행렬 $\mathbf{K}_i$ 을 설계하는 것이다.

정리 2] 만약, 아래의 시스템 관측기와 시스템 출력 $(\mathbf{C}_i, \mathbf{U}_i \mathbf{A})$ 가 가관측 조건을 만족한다면, 식 (4.16)을 안정화시키는 행렬 $\mathbf{K}_i$ 가 존재한다. 즉, 아래의 식 (4.21)과 같은 조건이 성립한다.

$$\text{rank} \begin{bmatrix} s\mathbf{I}_n - \mathbf{U}_i \mathbf{A} \\ \mathbf{C}_i \end{bmatrix} = n, \quad \forall s \in \mathbf{C}, \text{Re}(s) \geq 0 \quad (4.21)$$



정리 2 에서 $(\mathbf{C}_i, \mathbf{U}_i \mathbf{A})$ 가 가관측 조건을 만족한다는 것은 식 (4.16)에 의해, $(\mathbf{C}_i, \hat{\mathbf{A}} + \mathbf{K}_i \mathbf{C}_i)$ 이 가관측 조건을 만족한다는 것과 같다. 이것은 시스템

$(\hat{\mathbf{A}}, \mathbf{C}_i)$ 의 출력 피드백 이득이 $\mathbf{K}_i$ 일 경우의 폐루프 시스템과 같다. 그리고 식 (4.10)에서 $(\hat{\mathbf{A}}, \mathbf{C}_i)$ 는 안정하므로 폐루프 시스템을 안정화시키는 $\mathbf{K}_i$ 가 존재한다. 여기서 $\mathbf{K}_i$ 는 선형 최적 제어 기법(linear quadratic Regulator: LQR Method)으로 설계 가능하다.

정리 2로부터 식 (4.16)을 안정화시키는 행렬 $\mathbf{K}_i$ 가 설계되면 식 (4.5)에 대한 미지 입력 관측기는 설계 가능하다.

정리 3] 만약 다음의 두 가지 조건이 만족되면 식 (4.5)에 대한 미지 입력 관측기는 설계 가능하다. (부록 A 증명 3 참조)

$$\text{rank} \mathbf{C}_i \mathbf{D}_i = \text{rank} \mathbf{D}_i = q_i, \quad r_i \geq q_i \quad (4.22)$$

$$\text{rank} \begin{bmatrix} s\mathbf{I}_{n_i} - \mathbf{A} & \mathbf{D}_i \\ \mathbf{C}_i & \mathbf{0} \end{bmatrix} = n + q_i, \quad \forall s \in \mathbf{C}, \quad \text{Re}(s) \geq 0 \quad (4.23)$$

Fig. 4-1은 미지 입력 관측기를 블록선도로 표현한 것이다.

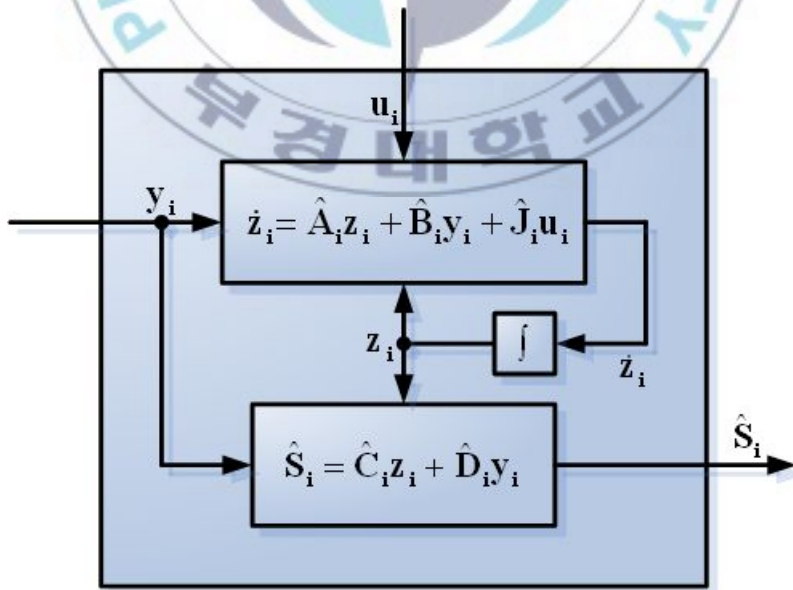


Fig. 4-1 Block diagram of unknown input observer

4.1.3 분산 제어기(Decentralized controller) 설계

본 절에서는 분산 제어 방식을 기본으로 채택한다. 중앙 장치에서 각 로봇의 모든 정보를 계측하고 이를 통해 대형 수준 함수를 계산한 후, 각 로봇에 대형 정보만을 전송한다. 로봇은 대형 정보와 자체 계측 정보만을 가지고 로컬 제어기 즉 분산제어기를 설계한다.

분산된 정보 구조에 대한 제약을 고려한 식 (4.24)의 형태를 가지는 분산 제어기를 설계한다. 설계를 위한 가정들은 다음과 같다.

$$\mathbf{u}_i(\mathbf{t}) = f_i\{\mathfrak{I}_i(t), t\}, i=1,2,\dots,N \quad (4.24)$$

가정 1] 전체 시스템 식 (4.25)의 $(\mathbf{A}, \mathbf{B})$ 는 가제어성(Controllability)과 가관측성(Observability)이다.

$$\dot{\mathbf{S}}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{B}\mathbf{u}(t) \quad (4.25)$$

가정 2] i 번째 제어 시스템에서 획득 가능한 정보 $\mathfrak{I}_i(t)$ 는 i 번째 시스템의 출력과 입력을 포함한다.

$$\mathfrak{I}_i(t) = \{\mathbf{y}_i(t), \mathbf{u}_i(t)\}, i=1,2,\dots,N. \quad (4.26)$$

가정 3] 전체 시스템의 상태 피드백 제어칙은 다음과 같다.

$$\mathbf{u} = -\mathbf{F}\mathbf{S}(t) \quad (4.27)$$

여기서 $\mathbf{F} \in \mathbb{R}^{m \times n}$ 는 상태 피드백 이득이다.



다중 차량 로봇의 대형 제어를 위한 전체 시스템 식 (4.25)는 피드백 제어칙 식 (4.27)에 의해서 안정화 된다. 분산 제어기의 목적은 전체 시스템을 안정하도록 하는 상태 피드백 제어칙 식 (4.27)과 동일한 성능을 가지는 분산 제어 입력 $\mathbf{u}_i(t)$ 을 식 (4.26)의 각 차량 로봇에서 획득 가능한 정보 $\mathfrak{I}_i(t)$ 만을 사용하여 설계하는 것이다. 제어 목적을 달성하기 위해, $\mathfrak{I}_i(t)$ 만을 입력으로

식 (4.29)의 $\mathbf{F}_i \in R^{m_i \times n}$ 는 상태 피드백 이득이며, 본 연구에서는 극배치(Pole assignment) 방법으로 구한다. 결론적으로, 미지 입력 관측기 식 (4.8)을 기반으로 하는 식 (4.29)의 분산 제어기 제어 입력 $\mathbf{u}_i(\mathbf{t})$ 는 식 (4.30)과 같이 제안된다.

$$\mathbf{u}_i(\mathbf{t}) = -\mathbf{F}_i \hat{\mathbf{S}}_i(\mathbf{t}) = -\mathbf{F}_i (\hat{\mathbf{C}}_i \mathbf{z}_i(\mathbf{t}) + \hat{\mathbf{D}}_i \mathbf{y}_i(\mathbf{t})) \quad (4.30)$$

4.2 대형 모델링, 제어기 및 관측기 설계

4.2.1 대형 모델링

2 장에서 유도된 단일 로봇에 대한 선형화된 운동학적 방정식은 다음 식 (4.31)과 같이 표현될 수 있다. (부록 A 증명 4 참조)

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r \sin \theta_r \\ 0 & 0 & v_r \cos \theta_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x - v_r \\ \omega - \omega_r \end{bmatrix} \quad (4.31)$$

여기서, $v_r = \text{const} \approx v_x$, $\theta_r = \text{const} \approx \theta$ 는 선형화를 위한 평형점 (Equilibrium point)을 나타낸다.

본 연구에서는 2 대의 차량 로봇을 평면 상에서만 제어하고자 하며, 1 번째 차량 로봇과 2 번째 차량 로봇의 일반화 좌표 벡터는 각각 $\mathbf{q}_1 = [X_1 \ Y_1 \ \theta_1]^T$ 와 $\mathbf{q}_2 = [X_2 \ Y_2 \ \theta_2]^T$ 이다. 그리고 전체 시스템의 상태 벡터 $\mathbf{S}(\mathbf{t})$ 와 대형 수준 함수(Formation level function)인 수평 대형 함수(Horizontal formation function) h_1 와 수직 대형 함수(Vertical formation function) h_2 을 식 (4.32)와 같이 정의한다.

$$\begin{cases} \mathbf{S}(\mathbf{t}) = [\theta_1 \ \theta_2 \ h_1 \ h_2]^T \\ h_1 = X_1 - X_2 + \alpha(\theta_1 - \theta_2) \\ h_2 = Y_1 + Y_2 - 6 + \beta(\theta_1 + \theta_2) \end{cases} \quad (4.32)$$

여기서, α , β 는 2 대의 차량 로봇들의 선회에 의해 발생하는 거리 측정에서의 섭동(Perturbation) 수준을 나타내는 것으로 시뮬레이션을 통해 구해지는 상수이다.

2 대의 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점인 $y=3$ 으로부터 일정한 거리를 유지하면서 움직인다고 가정하면, 그 관계식은 식 (4.33)과 같다.

$$\begin{cases} X_1 = X_2 \\ \frac{Y_1 + Y_2}{2} - 3 = 0 \\ \theta_1 = \theta_2 = 0 \end{cases} \quad (4.33)$$

각각의 차량 로봇의 x_i 축 속도 값과 목표 속도를 v_{x1} , v_{x2} , v_r 라고 하고, 각각의 차량 로봇의 각속도 값과 목표 각속도를 ω_1 , ω_2 , ω_r 하면, 차량 로봇 2 대의 대형 제어를 위한 전체 시스템 방정식은 식 (4.34)와 같으며, 각 차량 로봇의 제어 입력 $\mathbf{u}_1$, $\mathbf{u}_2$ 는 (4.35)와 같다.

$$\begin{cases} \dot{\mathbf{S}}(\mathbf{t}) = \mathbf{A}\mathbf{S}(\mathbf{t}) + \mathbf{B}_1\mathbf{u}_1(\mathbf{t}) + \mathbf{B}_2\mathbf{u}_2(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \mathbf{S}(\mathbf{t}) \end{cases} \quad (4.34)$$

$$\mathbf{u}_1 = \begin{bmatrix} v_{x1} - v_r \\ \omega_1 - \omega_r \end{bmatrix}, \quad \mathbf{u}_2 = \begin{bmatrix} v_{x2} - v_r \\ \omega_2 - \omega_r \end{bmatrix} \quad (4.35)$$

식 (4.31)의 $\theta_r = 0$, $\omega_r = 0$ 이라고 하면, 식 (4.34)의 행렬 $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$ 는 식 (4.36)과 같다. (부록 A 증명 5 참조)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ v_r & v_r & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & \alpha \\ 0 & \beta \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -1 & -\alpha \\ 0 & \beta \end{bmatrix} \quad (4.36)$$

4.2.2 제어기 및 관측기 설계

본 연구에서는 제어기 및 관측기 설계를 위한 v_r , ω_r , α , β 값을 $v_r = 0.1237$, $\omega_r = 0$, $\alpha = 0.05$, $\beta = 0.03$ 와 같이 선정하였다. 따라서 식 (4.34)의 전체 시스템은 식 (4.37)과 같으며, 식 (4.34)를 안정화시키는 전체 시스템의 페루프 제어기는 극배치법을 이용하여 식 (4.38)과 같이 설계되었다.

$$\left\{ \begin{array}{l} \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.1237 & 0.1237 & 0 & 0 \end{bmatrix} \\ \mathbf{B}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0.05 \\ 0 & 0.03 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -1 & -0.05 \\ 0 & 0.03 \end{bmatrix} \\ \mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \right. \quad (4.37)$$

$$\mathbf{F} = [\mathbf{F}_1 \quad \mathbf{F}_2]^T = \begin{bmatrix} -0.0075 & 0.0075 & 0.2 & 0 \\ 0.3403 & -0.0403 & 0 & 0.3234 \\ 0.0075 & -0.0075 & -0.2 & 0 \\ 0.0403 & 0.3403 & 0 & 0.3234 \end{bmatrix} \quad (4.38)$$

이때, 극은 $\gamma = [-0.2 - 0.2j \quad -0.2 + 0.2j \quad -0.4 \quad -0.3]$ 으로 설정하였고, $\mathbf{F}_1 \in \mathbb{R}^{2 \times 4}$, $\mathbf{F}_2 \in \mathbb{R}^{2 \times 4}$ 는 각 차량 로봇 시스템의 제어 피드백 이득이다.

식 (4.15)으로부터 $\hat{\mathbf{C}}_1 = \hat{\mathbf{C}}_2 = \mathbf{I}_4$ 이며, 식 (4.20)의 행렬 $(\mathbf{I}_{n_i} - \mathbf{G}_i \mathbf{C}_i)$ 을 비특이 행렬(nonsingular matrix)로 만드는 $\mathbf{G}_i$ 을 식 (4.39)와 같이 선정하였다.

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.39)$$

식 (4.16)과 식 (4.17)의 $\mathbf{K}_i$ 는 매트랩(MATLAB)의 LQR 함수를 이용하여 선형최적제어기법(Liner quadratic regulator)으로 구할 수 있으며, 식 (4.40)과 같이 $\mathbf{K}_1, \mathbf{K}_2$ 을 선정하였다.

$$\mathbf{K}_1 = \begin{bmatrix} 10 & -2.9195 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad \mathbf{K}_2 = \begin{bmatrix} -2.9195 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad (4.40)$$

Robot 1 :

이미 알고 있는 $\mathbf{C}_1, \mathbf{D}_1$ 과 선정된 $\mathbf{G}_1$ 을 식 (4.16)과 식 (4.18)에 대입하여 $\hat{\mathbf{D}}_1$ 과 $\mathbf{U}_1$ 을 구한다.

$$\hat{\mathbf{D}}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 33.3333 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{U}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & -33.3333 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

앞서 계산된 $\hat{\mathbf{D}}_1, \mathbf{U}_1$ 과 선형최적제어기법으로 구해진 $\mathbf{K}_1$ 을 식 (4.14)와 식 (4.15)에 대입하여 $\hat{\mathbf{A}}_1, \hat{\mathbf{B}}_1$ 과 $\hat{\mathbf{J}}_1$ 을 구한다.

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} -10 & 0 & 0 & 0 \\ -1.2038 & -4.1233 & 0 & 0 \\ 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & -10 \end{bmatrix},$$

$$\hat{\mathbf{B}}_1 = \begin{bmatrix} 0 & 0 & 0 \\ -8.2467 & 0 & -137.4444 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{J}}_1 = \begin{bmatrix} 0 & 0 \\ 0 & -2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Robot 1 :

이미 알고 있는 $\mathbf{C}_2$, $\mathbf{D}_2$ 와 선정된 $\mathbf{G}_2$ 을 식 (4.16)과 식 (4.18)에 대입하여 $\hat{\mathbf{D}}_2$ 와 $\mathbf{U}_2$ 을 구한다.

$$\hat{\mathbf{D}}_2 = \begin{bmatrix} 1 & 0 & 33.3333 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} 1 & -1 & 0 & -33.3333 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

앞서 계산된 $\hat{\mathbf{D}}_2$, $\mathbf{U}_2$ 와 선형최적제어기법으로 구해진 $\mathbf{K}_2$ 을 식 (4.14)와 식 (4.15)에 대입하여 $\hat{\mathbf{A}}_2$, $\hat{\mathbf{B}}_2$ 와 $\hat{\mathbf{J}}_2$ 을 구한다.

$$\hat{\mathbf{A}}_2 = \begin{bmatrix} -4.1233 & -1.2038 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & -10 \end{bmatrix},$$

$$\hat{\mathbf{B}}_2 = \begin{bmatrix} -7.0429 & 0 & -137.4444 \\ 10 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{J}}_2 = \begin{bmatrix} 0 & -2 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

4.3 Broadcasting 에 의한 분산 제어

4.3.1 기본 개념 및 원리

본 절에서는 2 대의 차량 로봇을 Broadcasting 방식으로 분산 제어하는 방법과 그 시물레이션과 실험 결과를 제시한다.

Fig. 4-3 에 소개되어진 Broadcasting 방식은 2 대의 차량 로봇들 외에도 별도로 서버 시스템이 필요하다. 통신 용어 측면에서 살펴보면 Broadcasting 은 방송이라는 뜻으로 데이터 전송 시 목적지가 하나의 호스트가 아니라, 동일한 네트워크에 연결되어 있는 모든 호스트들이 된다. 즉 단 한번의 데이터 전송으로 여러 클라이언트들에게 데이터를 전달할 수 있다. 그리고 기본적인 데이터 전송 방식은 UDP(User datagram protocol)를 사용한다. 본 연구에서는 이러한 개념을 도입하여 서버는 모든 차량 로봇의 정보를 알고 있으며, 전체 대형 수준 함수만을 Broadcasting 방식을 이용하여 각 차량 로봇으로 전송한다. 그리고 각 차량 로봇은 Broadcasting 방식으로 얻은 대형 수준 함수 정보와 자체적으로 계측한 정보를 활용하여 미지 입력 관측기로부터 시스템 전체의 상태를 추정하고 이를 통해 설계된 제어 입력으로 대형을 제어한다.

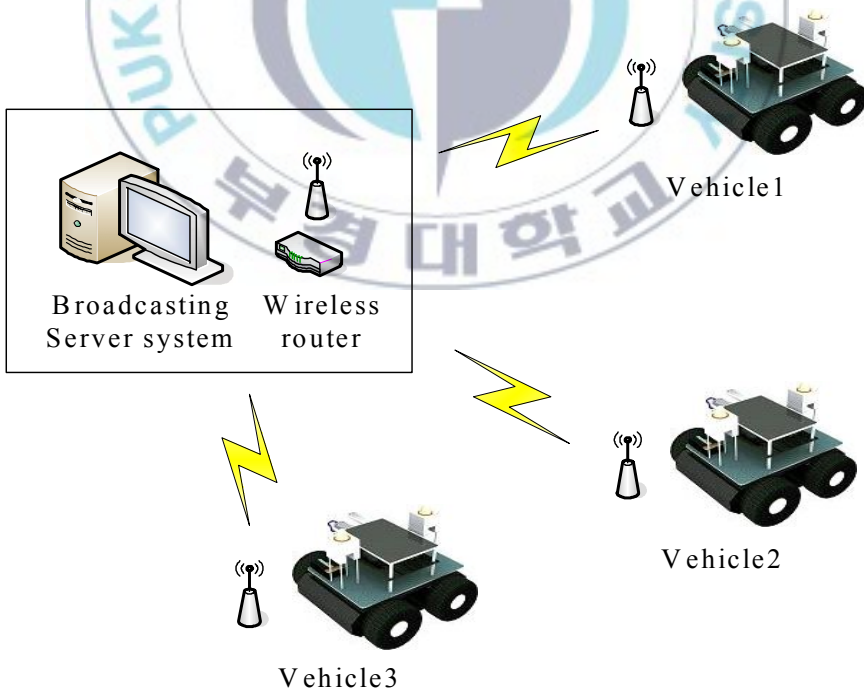


Fig. 4-3 Broadcasting concept

Fig. 4-4 는 식 (4.33)의 Broadcasting 분산 제어 방식의 관계식을 그림으로 표현한 것으로, 2 대의 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점인 $y=3$ 으로부터 일정한 거리 d 을 유지하면서 움직인다.

Fig. 4-4 의 Broadcasting 서버 시스템은 2 대의 차량 로봇으로부터 각각 일반화 좌표 벡터 $\mathbf{q}_1=[X_1 \ Y_1 \ \theta_1]^T$ 와 $\mathbf{q}_2=[X_2 \ Y_2 \ \theta_2]^T$ 을 전송 받아 식 (4.32)에서 정의된 상태 벡터 $\mathbf{S}(t)=[\theta_1 \ \theta_2 \ h_1 \ h_2]^T$ 을 생성한다. 그리고 서버 시스템은 생성된 상태 벡터 중에서 대형 수준 함수 h_1, h_2 만을 각 차량 로봇으로 전송한다.

각 차량 로봇의 제어기는 다른 차량 로봇에 대한 정보를 알 수가 없으므로, 다른 차량 로봇에 대한 정보를 미지 입력으로 가지게 된다. 따라서 각 차량 로봇의 제어기에 미지 입력을 계측할 수 있는 미지 입력 관측기가 필요하게 된다.

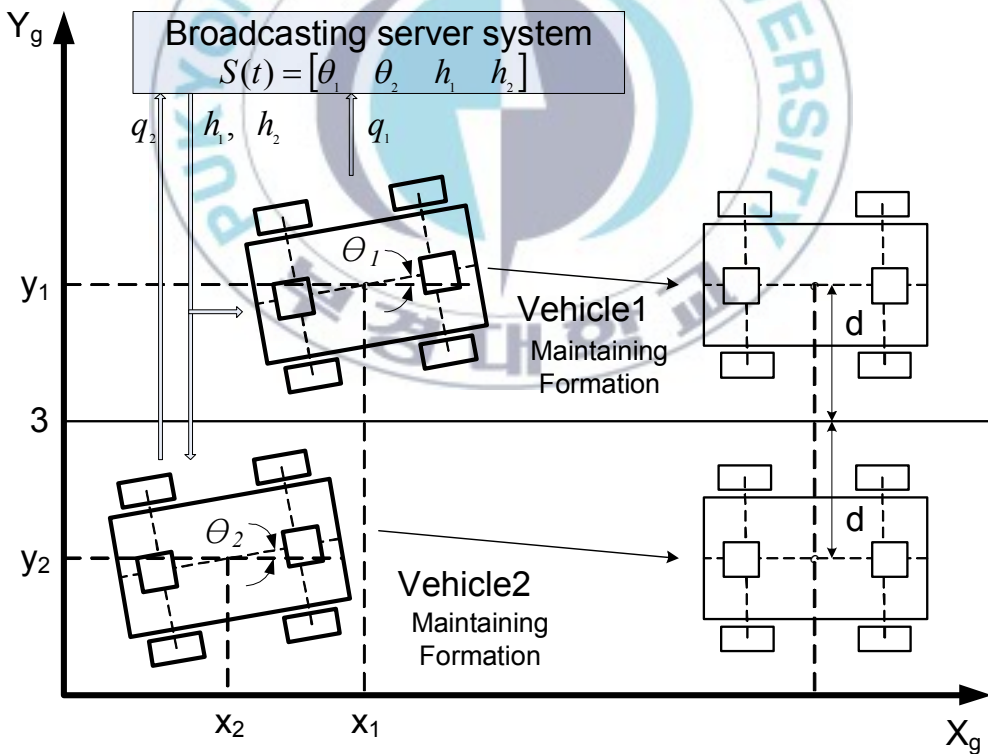


Fig. 4-4 Broadcasting decentralized controller

각 차량 로봇들은 각 차량 로봇의 전진방향각 θ_1 , θ_2 와 대형 수준 함수 h_1 , h_2 을 이용하여 미지 입력 관측기의 입력 벡터 $\mathbf{y}_1 = [\theta_1 \ h_1 \ h_2]^T$, $\mathbf{y}_2 = [\theta_2 \ h_1 \ h_2]^T$ 을 각각 생성한다. 각 차량 로봇들은 4.1 절에서 제안된 미지 입력 관측기에 기반한 분산 제어 방법을 이용하여 서로 다른 차량 로봇의 전진방향각 $\hat{\theta}_{21}$, $\hat{\theta}_{12}$ 을 추정하여 일정한 대형을 유지하면서 이동한다. 여기서 $\hat{\theta}_{12}$ 는 $\mathbf{y}_2$ 를 이용하여 2 번째 차량 로봇의 관측기에서 추정한 1 번째 차량 로봇의 전진방향각이고, $\hat{\theta}_{21}$ 는 $\mathbf{y}_1$ 을 이용하여 1 번째 차량 로봇의 관측기에서 추정한 2 번째 차량 로봇의 전진방향각이다.

Fig. 4-5 는 앞서 소개된 Broadcasting 분산 제어 방식을 블록선도로 표현한 것이다.

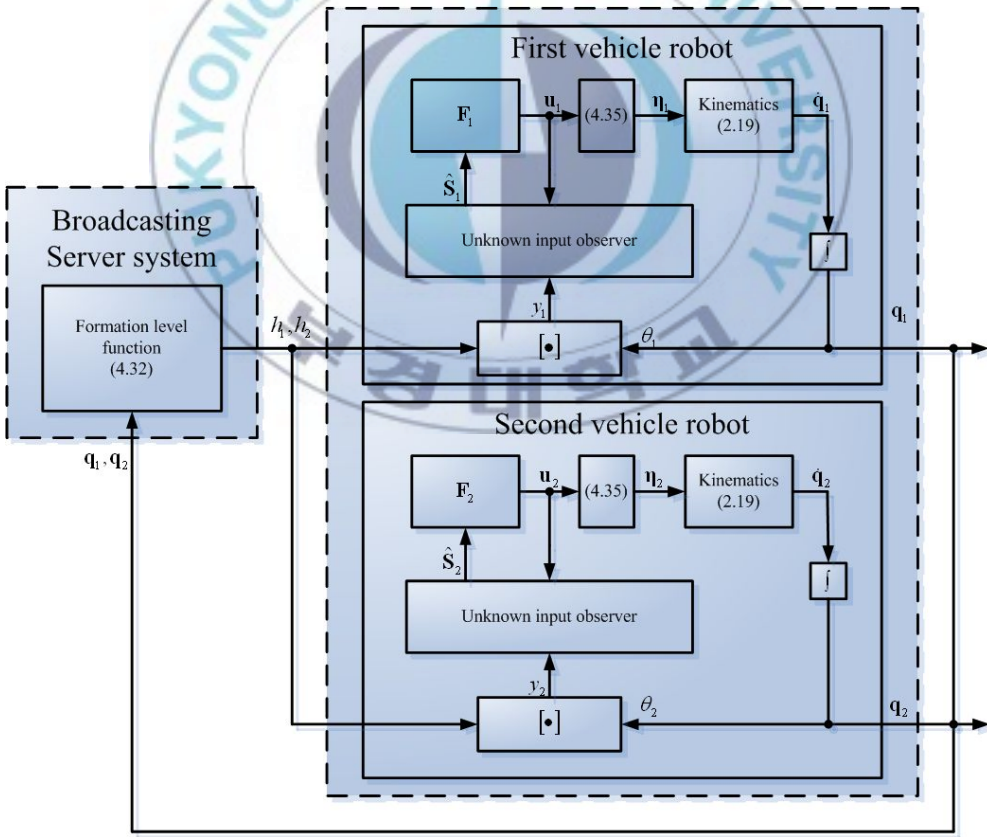


Fig. 4-5 Block diagram of broadcasting decentralized controller

미지 입력 관측기로부터 추정된 상태 $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2$ 값과 극배치법으로 설계한 각 차량 로봇의 상태 피드백 이득 $\mathbf{F}_1, \mathbf{F}_2$ 을 식 (4.27)에 대입하여 분산 제어기의 제어 입력값 $\mathbf{u}_1, \mathbf{u}_2$ 을 구할 수 있다. 그리고 분산 제어기의 제어 입력값과 식 (4.35)을 이용하여 차량 로봇의 보조 속도 벡터 $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$ 을 얻을 수 있다. 마지막으로 보조 속도 벡터를 식 (2.19)에 대입하여 각 차량 로봇의 일반화 속도 벡터 $\dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2$ 을 구하고 이를 적분하여 일반화 좌표 벡터 $\mathbf{q}_1, \mathbf{q}_2$ 을 구한 뒤, 이를 Broadcasting 서버 시스템에 다시 전달하게 된다.

4.3.2 시뮬레이션 결과

시뮬레이션에서는 식 (4.33)에서 표현된 것처럼 1 번째와 2 번째 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점 ($y=3$) 으로부터 일정한 거리를 유지하면서 움직이도록 한다. 1 번째 차량 로봇의 시작 위치는 $X_1=0.507m, Y_1=1.406m, \theta_1=-0.0087rad$ 이고, 2 번째 차량 로봇의 초기 위치는 $X_2=0.452m, Y_2=3.029m, \theta_2=0.0851rad$ 이며, 차량 로봇들의 목표 속도는 $v_r=0.1237m/s$ 이다. 그리고 시뮬레이션의 샘플링 시간은 $0.1sec$ 이며, 500 개의 데이터로 50 초 동안 시뮬레이션 한다.

Fig. 4-6 은 차량 로봇 2 대의 이동 경로를 나타낸 그래프로, 실선은 1 번째 차량 로봇의 이동경로를 점선은 2 번째 차량 로봇의 이동 경로를 나타내고 있다. 차량 로봇 2 대의 거리는 시작부터 종료까지 $1.6m$ 로 거의 일정하게 유지되고 있으며, X_g 방향으로 $3.2m$ 지점부터는 차량 로봇들이 평행하게 움직이고 있음을 알 수 있다.

Fig. 4-7 과 Fig. 4-8 은 각각 $\boldsymbol{\eta}_1=[v_{x1} \ \omega_1]^T, \boldsymbol{\eta}_2=[v_{x2} \ \omega_2]^T$ 으로 표현되는 1 번째와 2 번째 차량 로봇 시스템에 입력되는 보조속도벡터를 나타낸 그래프이다. Fig. 4-7 은 각 차량 로봇의 x_i 축 성분 전진속도를 보여주는 것으로 2 대의 차량 로봇 모두 10 초 후에는 목표 속도인 $0.1237m/s$ 을 유지하고 있음을 보여 준다. Fig. 4-8 은 각 차량 로봇의 회전각속도를 보여주는 것으로 20 초 후에는 차량 로봇 2 대의 회전각속도가 거의 $\omega_r=0rad/s$ 으로 유지됨을 보여주고 있다.

Fig. 4-9 는 1 번째 차량 로봇에서 측정된 θ_1 값과 4.1 절에서 제안되어진 2 번째 차량 로봇에서 설계된 미지 입력 관측기로 추정된 $\hat{\theta}_{12}$ 값을 나타낸

그래프이다. 측정된 θ_1 값과 추정된 $\hat{\theta}_{12}$ 값이 모두 30 초 이내에 식 (4.36)에서 제안된 목표 전진방향각인 $\theta_r = 0rad$ 으로 수렴해 감을 알 수 있다.

Fig. 4-10 은 2 번째 차량 로봇에서 측정된 θ_2 값과 4.1 절에서 제안되어진 1 번째 차량 로봇에서 설계된 미지 입력 관측기로 추정한 $\hat{\theta}_{21}$ 값을 나타낸 그래프이다. 측정된 θ_2 값과 추정된 $\hat{\theta}_{21}$ 값이 모두 30 초 이내에 식 (4.36)에서 제안된 목표 전진방향각인 $0rad$ 으로 수렴해 감을 알 수 있다.

. Fig. 4-11 에서 분산 제어 출력값인 수평 대형 함수 h_1 은 약 10 초 후에 거의 0 으로 수렴되는 것을 볼 수 있다. 수평 대형 함수 h_1 값이 10 초 후에는 0 으로 수렴하기 때문에, Fig. 4-7 에서 차량 로봇들의 x_i 축 성분 전진속도가 10 초 후에는 목표 전진속도인 $0.1237m/s$ 으로 유지되게 된다. Fig. 4-12 에서 분산 제어 출력값인 수평 대형 함수 h_1 은 약 25 초 후에 거의 0 으로 수렴되는 것을 볼 수 있다. 수직 대형 함수 h_2 값이 25 초 후에는 0 으로 수렴하기 때문에, Fig. 4-8 에서 차량 로봇들의 회전각속도가 25 초 후에는 목표 회전각속도 $\omega_r = 0rad/s$ 으로 유지되게 된다.

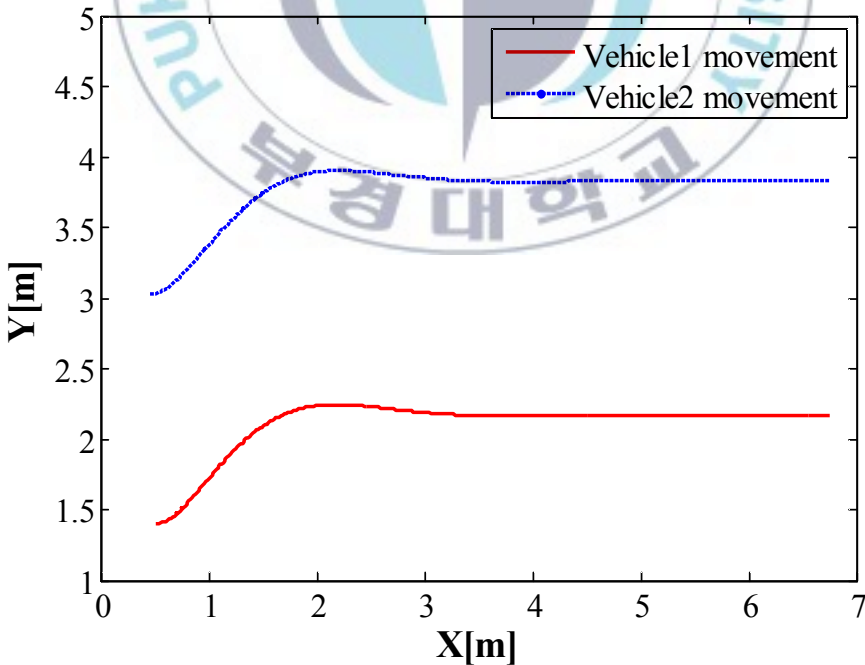


Fig. 4-6 Movements of vehicle1-vehicle2

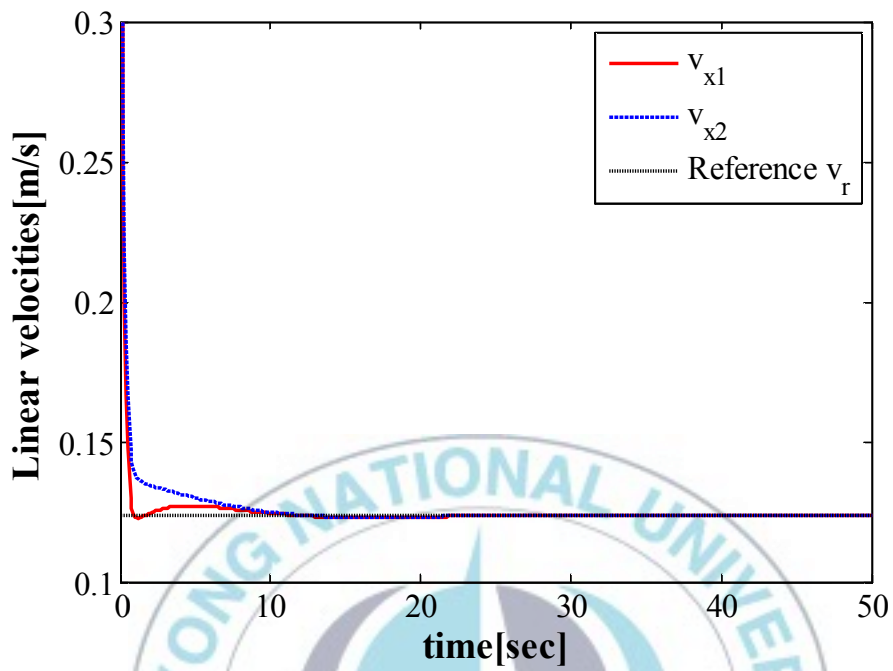


Fig. 4-7 Linear velocities of vehicle1-vehicle2

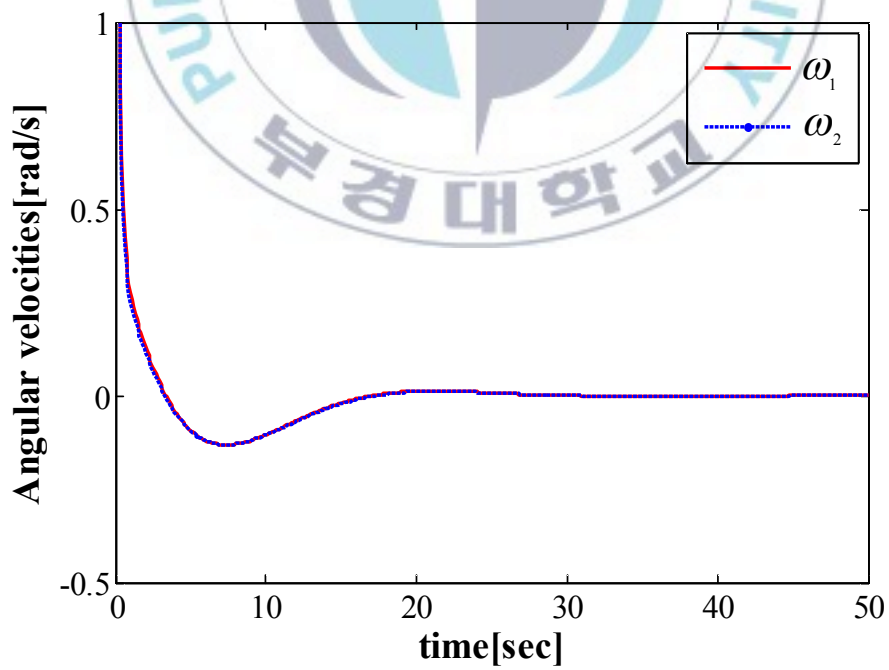


Fig. 4-8 Angular velocities of vehicle1-vehicle2

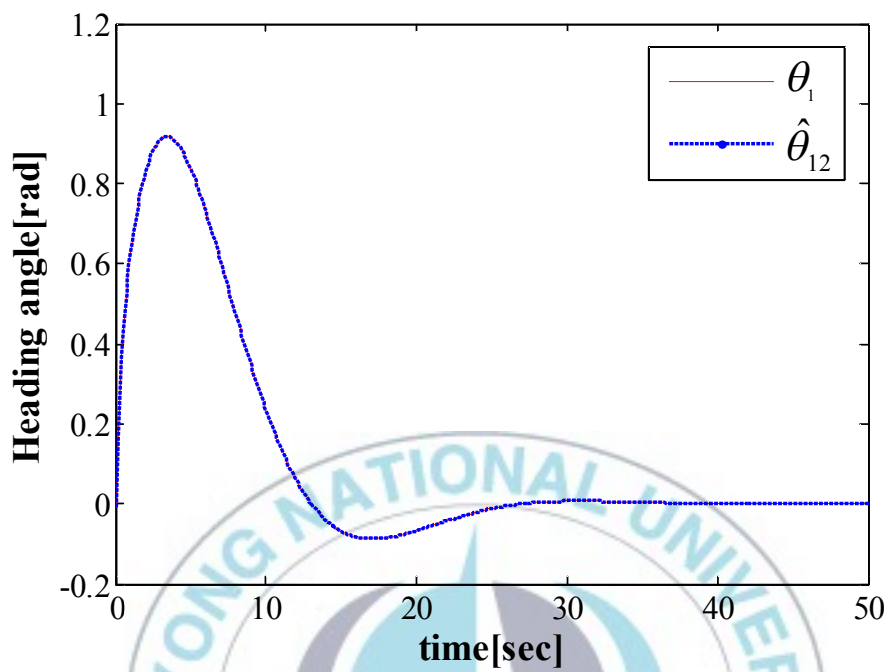


Fig. 4-9 Heading angle of vehicle1

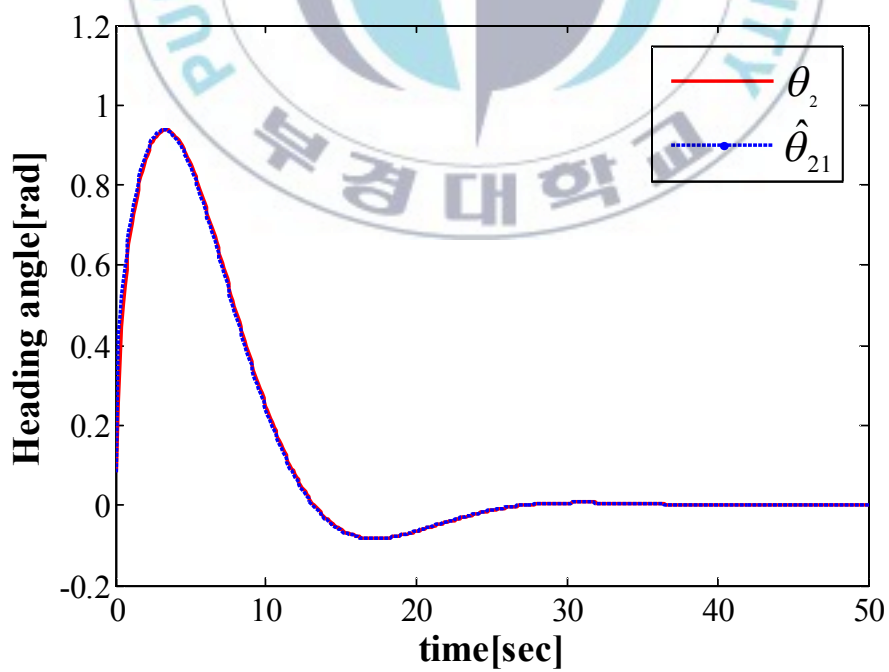


Fig. 4-10 Heading angle of vehicle2

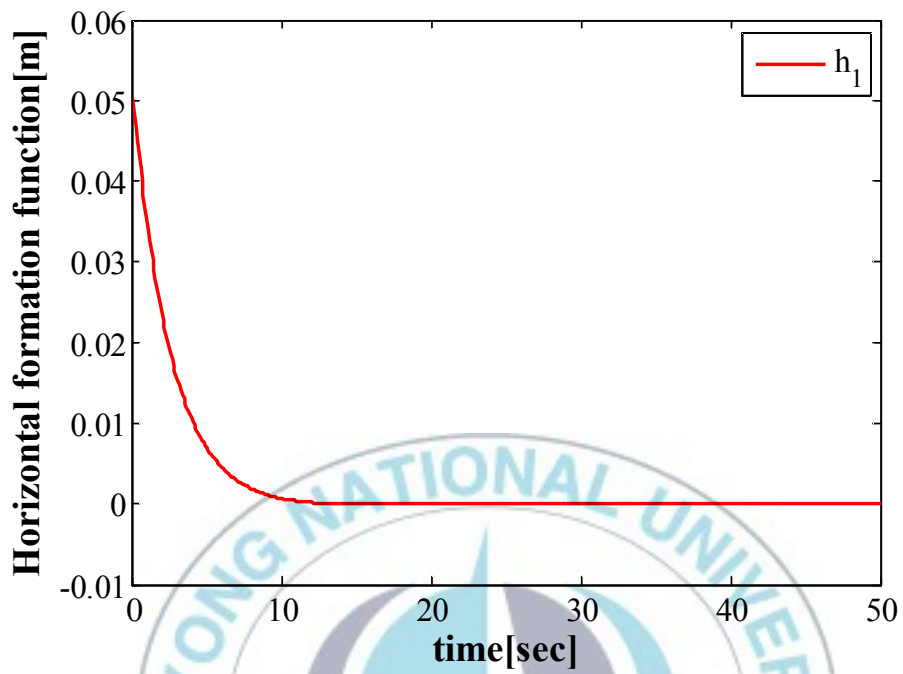


Fig. 4-11 Horizontal formation function

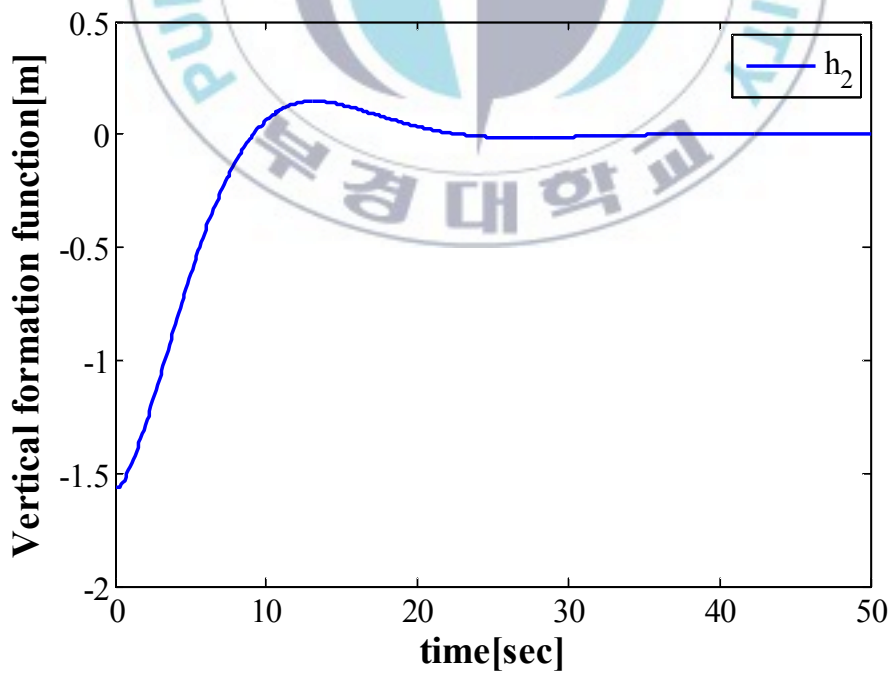


Fig. 4-12 Vertical formation function

4.3.3 실험 결과 및 고찰

실험에서는 식 (4.33)에서 표현된 것처럼 1 번째와 2 번째 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점 ($y = 3$) 으로부터 일정한 거리를 유지하면서 움직이도록 한다. 1 번째 차량 로봇의 시작 위치는 $X_1 = 0.507m$, $Y_1 = 1.406m$, $\theta_1 = -0.0087rad$ 이고, 2 번째 차량 로봇의 초기 위치는 $X_2 = 0.452m$, $Y_2 = 3.029m$, $\theta_2 = 0.0851rad$ 이며, 목표 속도는 $v_r = 0.1237m/s$ 이다. 본 실험에서는 2 대의 차량이 움직일 수 있는 공간이 제한적이기 때문에, $X_g = 4m$ 까지 실험하도록 한다.

Fig. 4-13 은 차량 로봇 2 대의 이동 경로를 나타낸 그래프로 실선은 1 번째 차량 로봇의 이동경로를, 점선은 2 번째 차량 로봇의 이동 경로를 나타내고 있다. 차량 로봇 2 대의 거리는 시작부터 종료까지 $1.8m$ 로 거의 일정하게 유지되고 있으며, X_g 방향으로 $3.5m$ 지점부터는 차량 로봇들이 거의 평행하게 움직이고 있음을 알 수 있다.

Fig. 4-14 와 Fig. 4-15 는 각각 $\eta_1 = [v_{x1} \ \omega_1]^T$, $\eta_2 = [v_{x2} \ \omega_2]^T$ 으로 표현되는 1 번째와 2 번째 차량 로봇 시스템에 입력되는 보조속도벡터를 나타낸 그래프이다. Fig. 4-14 는 각 차량 로봇의 x_i 축 성분 전진속도를 보여주는 것으로 2 대의 차량 로봇 모두 5 초 후에는 목표 전진속도인 $0.1237m/s$ 을 유지하고 있음을 보여 준다. Fig. 4-15 는 각 차량 로봇의 회전각속도를 보여주는 것으로 20 초 후에는 차량 로봇 2 대의 목표 회전각속도가 $\omega_r = 0rad/s$ 주위 값을 유지하고 있음을 보여주고 있다.

Fig. 4-16 은 1 번째 차량 로봇에서 측정된 θ_1 값과 4.1 절에서 제안되어진 2 번째 차량 로봇에서 설계된 미지 입력 관측기로 추정된 $\hat{\theta}_{12}$ 값을 나타낸 그래프이다. 측정된 θ_1 값과 추정된 $\hat{\theta}_{12}$ 값이 모두 약 35 초부터 식 (4.36)에서 제안된 목표 전진방향각 $\theta_r = 0rad$ 주위 값을 유지하고 있음을 알 수 있다.

Fig. 4-17 은 2 번째 차량 로봇에서 측정된 θ_2 값과 4.1 절에서 제안되어진 1 번째 차량 로봇에서 설계된 미지 입력 관측기로 추정된 $\hat{\theta}_{21}$ 값을 나타낸 그래프이다. 측정된 θ_2 값과 추정된 $\hat{\theta}_{21}$ 값이 모두 35 초부터 식 (4.36)에서 제안된 목표 전진방향각인 $0rad$ 주위 값을 유지하고 있음을 알 수 있다.

서버에서 2 대의 차량 로봇으로 전달하는 분산 제어 출력값 h_1 , h_2 중에서 Fig. 4-18 은 수평 대형 함수 h_1 값을 나타낸 그래프이고, Fig. 4-19 은 수직 대형 함수 h_2 값을 나타낸 그래프이다. Fig. 4-18 에서 분산 제어 출력값인 수평 대형 함수 h_1 은 약 5 초 후에 거의 0 으로 수렴되는 것을 볼 수 있으며, Fig. 4-19 에서 분산 제어 출력값인 수직 대형 함수 h_2 은 약 20 초 후에 0 주위 값으로 유지됨을 볼 수 있다.

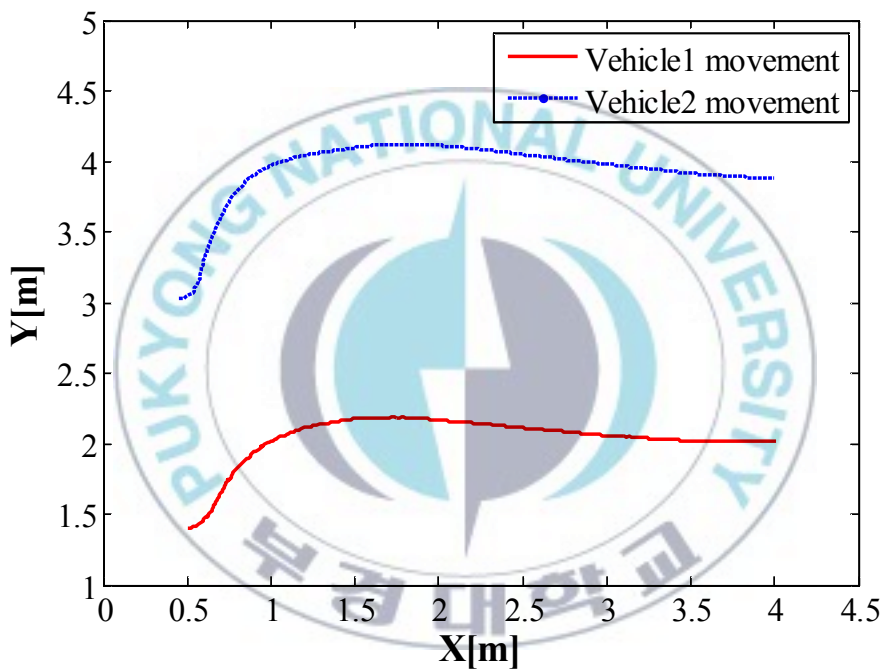


Fig. 4-13 Movements of vehicle1-vehicle2

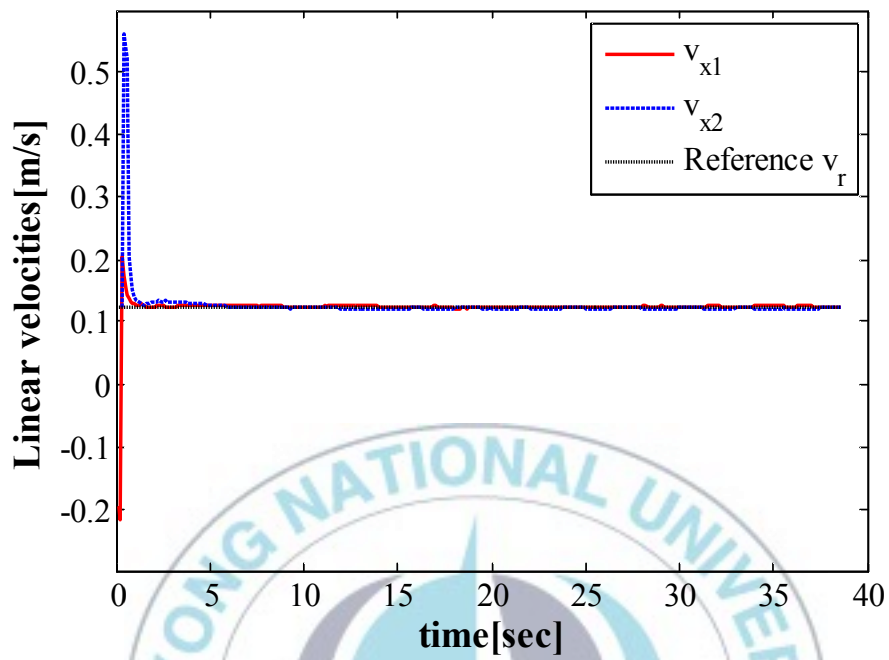


Fig. 4-14 Linear velocities of vehicle1-vehicle2

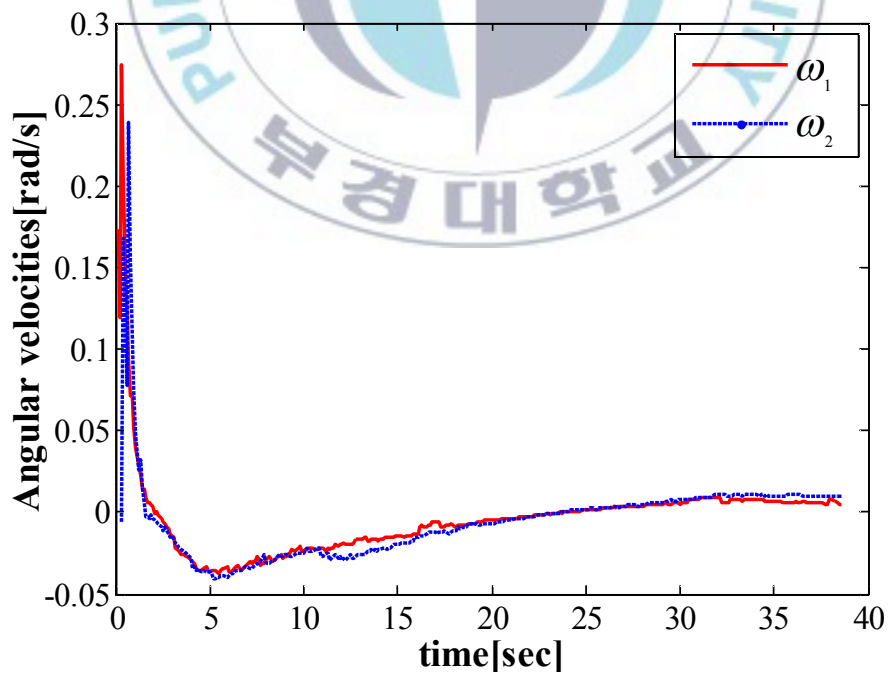


Fig. 4-15 Angular velocities of vehicle1-vehicle2

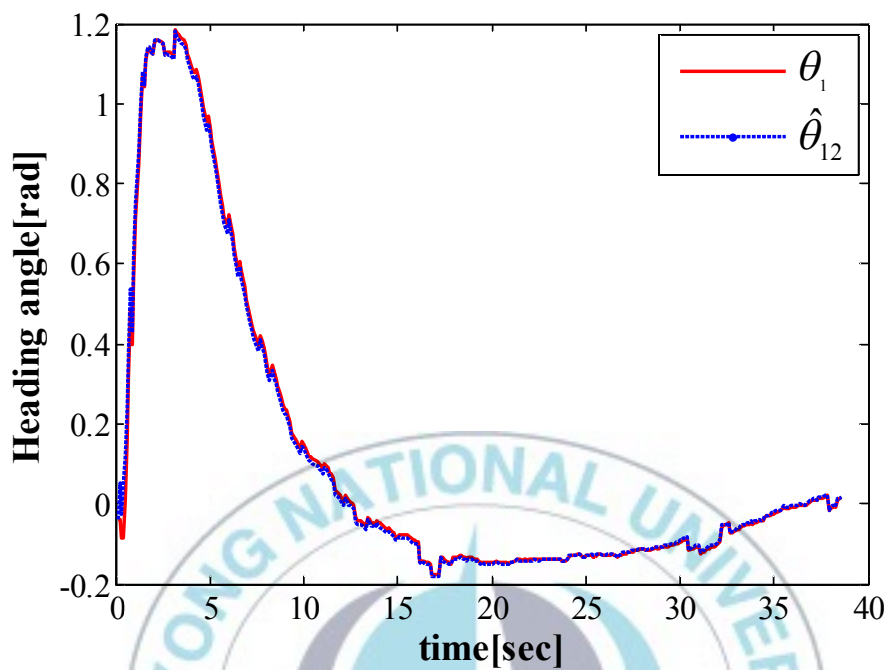


Fig. 4-16 Heading angle of vehicle1

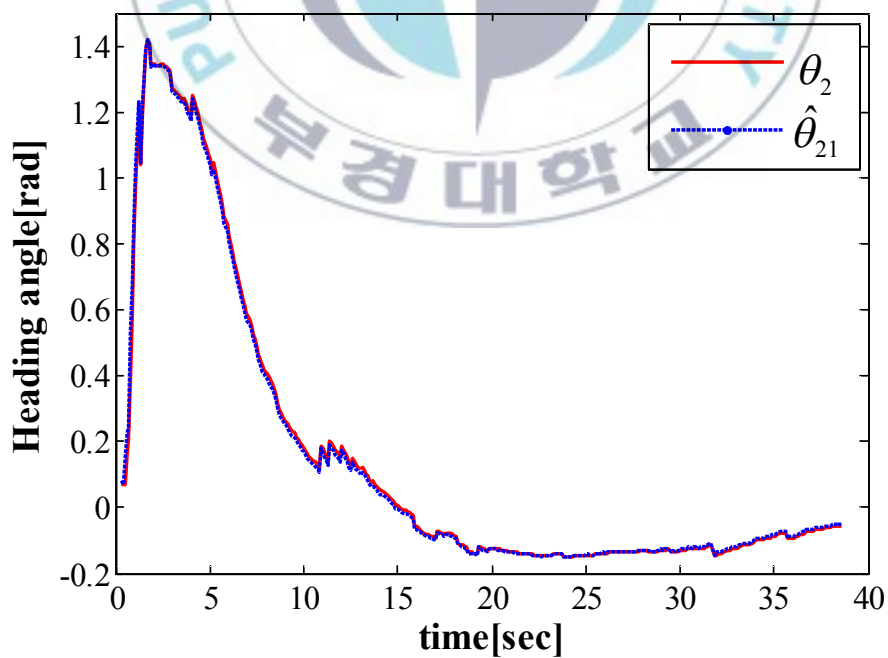


Fig. 4-17 Heading angle of vehicle2

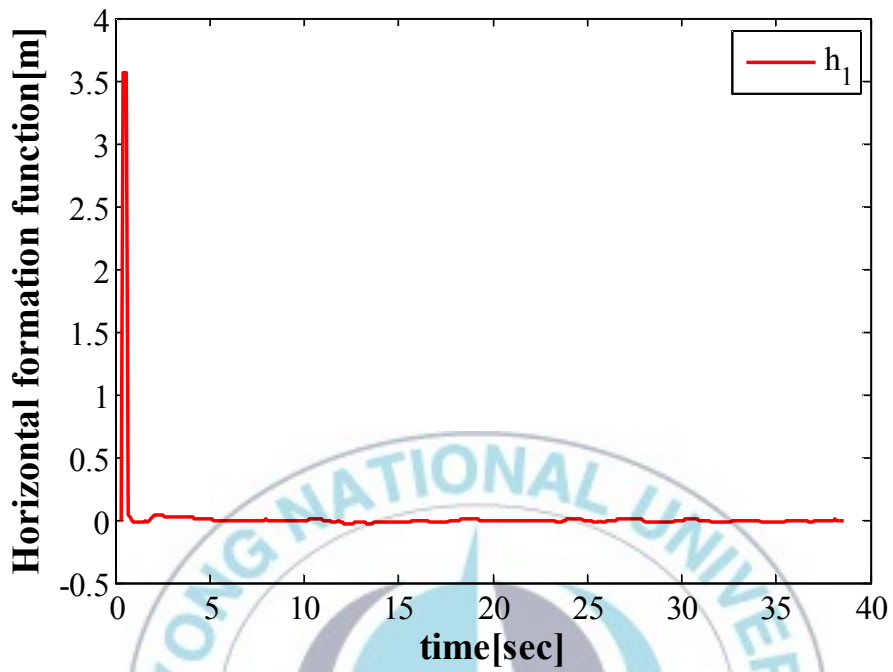


Fig. 4-18 Horizontal formation function

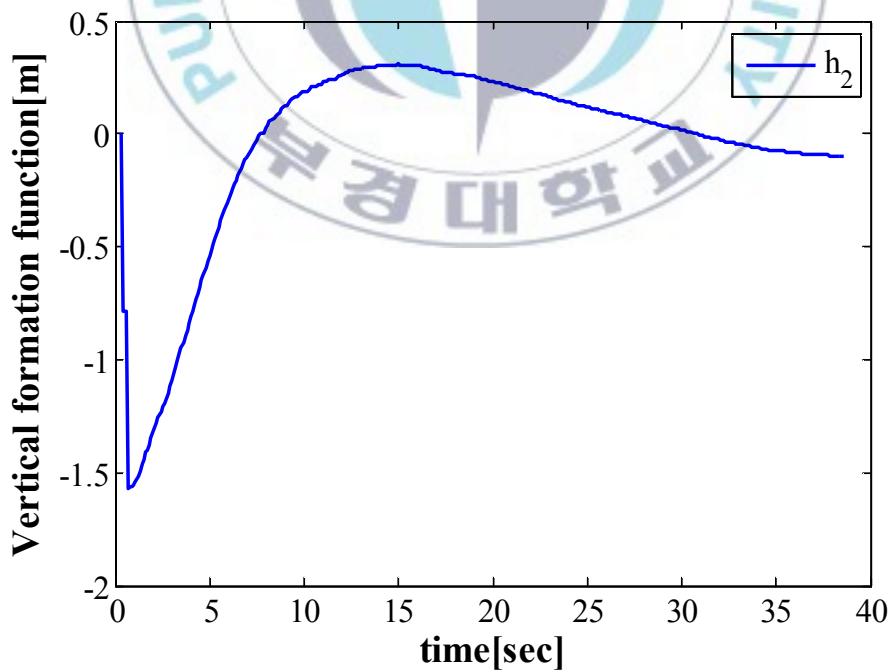


Fig. 4-19 Vertical formation function

4.4 선도-추종 분산 제어

4.4.1 기본 개념 및 원리

본 절에서는 2 대의 차량 로봇을 선도-추종(Leader-Following) 방식으로 분산 제어하는 개념을 소개하고 시뮬레이션과 실험 결과를 제시한다.

선도-추종 방식은 1 대의 선도 차량 로봇과 1 대 이상의 추종 차량 로봇으로 구성되며, 선도 차량 로봇은 다른 차량 로봇에 관계없이 일정한 속도로 목표 경로를 추종한다. 이를 위해 LOS 경로 추종 알고리즘을 이용하여 추적 오차를 최소화시키는 제어 입력 벡터 즉, 선도차량의 전진속도와 회전 각속도를 얻는다. 그러면 나머지 차량 로봇은 추종 차량 로봇이 되어 선도 차량 로봇과 일정한 대형을 이루며 동작한다. 따라서 선도 차량 로봇은 추종 차량 로봇에 대한 정보가 불필요하지만, 추종 차량 로봇은 선도 차량 로봇에 대한 정보가 반드시 필요하다. 따라서 선도-추종 방식의 서버는 모든 차량 로봇으로부터 데이터를 전달 받지만, 추종 차량 로봇에만 대형 수준 함수를 전송한다. 추종 차량 로봇은 전송 받은 대형 수준 함수와 자체 출력값을 이용하여 미지 입력 관측기로 상태를 추정한다. 추정된 상태와 극배치법으로 설계된 상태 피드백을 이용하여 제어 입력을 계산하고, 이 제어 입력으로 선도 차량 로봇과 대형을 유지하도록 추종 차량 로봇을 제어한다.

Fig. 4-20 은 선도-추종 분산 제어 방식에 의한 선도 차량 로봇과 추종 차량 로봇의 관계를 표현한 그림이다. 선도 차량 로봇이 일정한 경로를 따라 움직이기 때문에, 추종 차량 로봇은 식 (4.33)에 의해 $y=3$ 을 기준으로 선도 차량 로봇의 경로와 대칭되는 경로를 주행하게 된다. Fig. 4-20 의 선도-추종 서버 시스템은 2 대의 차량 로봇으로부터 각각 일반화 좌표 벡터 $\mathbf{q}_1=[X_1 \ Y_1 \ \theta_1]^T$ 와 $\mathbf{q}_2=[X_2 \ Y_2 \ \theta_2]^T$ 을 전송 받아 식 (4.32)에서 정의된 상태 벡터 $\mathbf{S}(\mathbf{t})=[\theta_1 \ \theta_2 \ h_1 \ h_2]^T$ 을 생성한다. 그리고 서버 시스템은 생성된 대형 수준 함수 h_1, h_2 을 추종 차량 로봇에만 전송한다.

추종 차량 로봇의 제어기에서는 선도 차량 로봇에 대한 정보를 알 수가 없으므로, 선도 차량 로봇에 대한 정보를 미지 입력으로 가지게 된다. 따라서 이러한 미지 입력을 가지는 추종 차량 로봇은 상태 벡터 $\hat{\mathbf{S}}_2$ 을 측정할 수 있는 미지 입력 관측기가 필요하게 된다. 추정된 상태와 극배치법으로 설계된 상태 피드백을 이용하여 제어 입력을 계산하고, 이 제어 입력으로 선도 차량 로봇과 대형을 유지하도록 추종 차량 로봇을 제어한다.

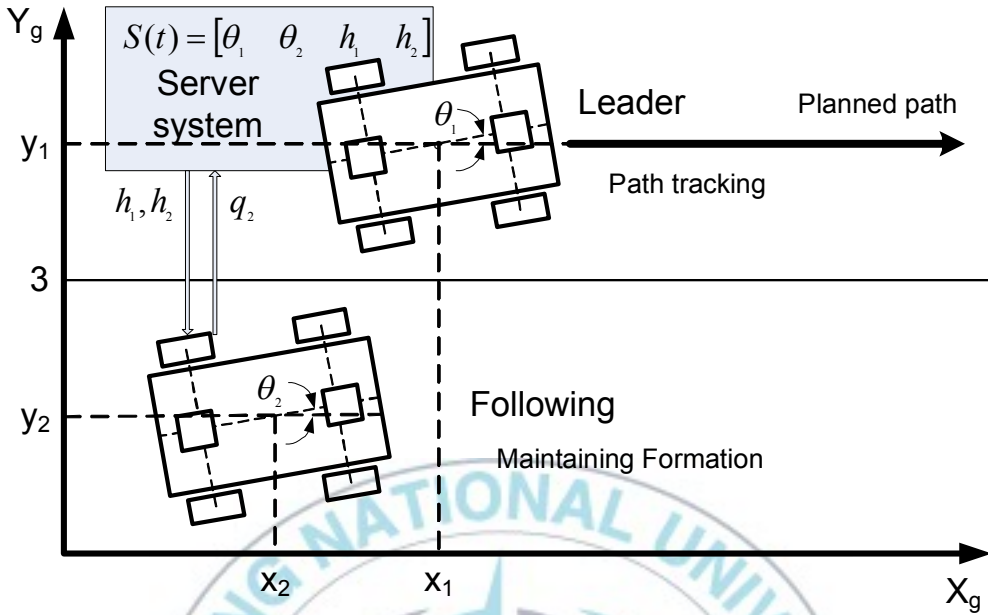


Fig. 4-20 Leader-following decentralized control

Fig. 4-21 은 선도-추종 분산 제어 방식을 블록선도로 표현한 것이다. 추종 차량 로봇은 전진방향각 θ_2 와 대형 수준 함수 h_1, h_2 을 이용하여 미지 입력 관측기의 입력 벡터 $\mathbf{y}_2 = [\theta_2 \ h_1 \ h_2]^T$ 을 생성한다. 추종 차량 로봇은 4.1 절에서 제안된 미지 입력 관측기에 기반한 분산 제어 방법을 이용하여 선도 차량 로봇의 전진방향각 $\hat{\theta}_{12}$ 을 추정하여 일정한 대형을 유지하면서 이동한다. 여기서 $\hat{\theta}_{12}$ 는 $\mathbf{y}_2$ 을 이용하여 추종 차량 로봇의 관측기에서 추정한 선도 차량 로봇의 전진방향각이다.

추종 차량 로봇은 미지 입력 관측기로부터 추정된 $\hat{\mathbf{S}}_2$ 값과 극배치법으로 설계한 각 차량 로봇의 상태 피드백 이득 $\mathbf{F}_2$ 을 식 (4.27)에 대입하여 분산 제어기의 제어 입력값 $\mathbf{u}_2$ 을 구할 수 있다. 그리고 분산 제어기의 제어 입력값과 식 (4.35)을 이용하여 추종 차량 로봇의 보조 속도 벡터 $\mathbf{n}_2$ 을 얻을 수 있다. 선도 차량 로봇은 경로 추종 알고리즘을 이용하여 선도 차량 로봇의 보조 속도 벡터 $\mathbf{n}_1$ 을 구한다. 마지막으로 각 차량의 보조 속도 벡터를 식 (2.19)에 대입하여 각 차량 로봇의 일반화 속도 벡터 $\dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2$ 을 구하고 이를 적분하여

일반화 좌표 벡터 $\mathbf{q}_1, \mathbf{q}_2$ 을 구한 뒤, 이를 선도-추종 서버 시스템에 다시 전달하게 된다.

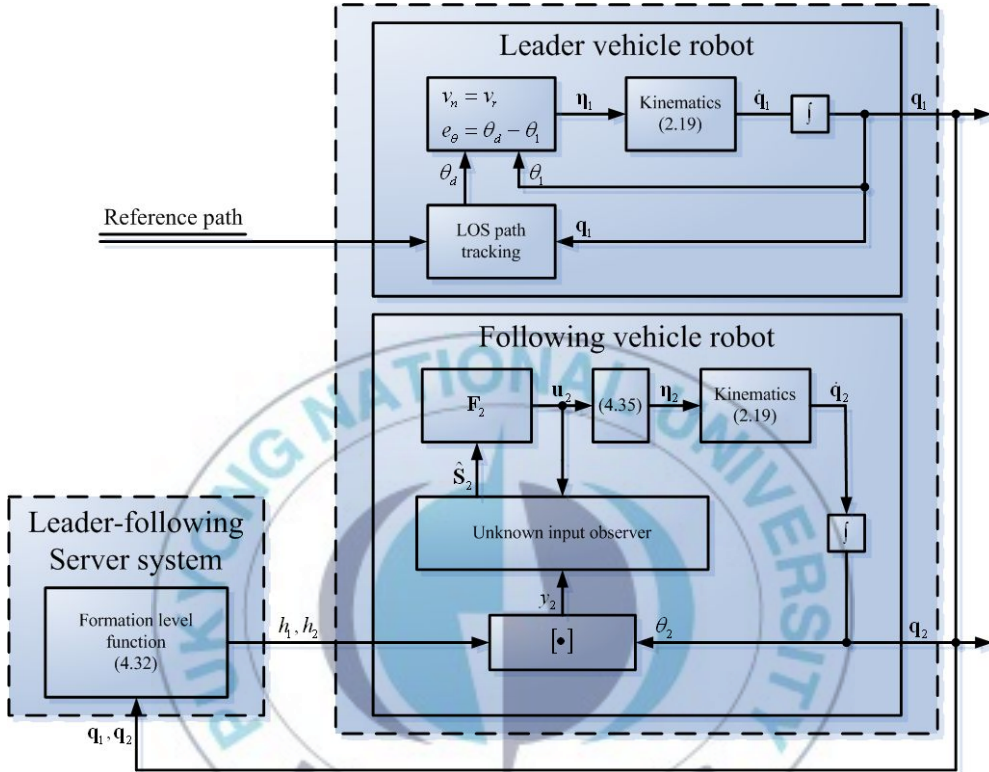


Fig. 4-21 Block diagram of leader-following decentralized control

4.4.2 시뮬레이션 결과

시뮬레이션에서는 식 (4.33)에서 표현된 것과 같이 선도-추종 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점 ($y = 3$) 으로부터 일정한 거리를 유지하면서 움직이도록 한다. 선도 차량 로봇은 3 장에서 소개된 LOS 경로 추종 알고리즘을 이용하여, 좌표 (1,3) 에서 시작되어 좌표 (4,3) 에서 끝나는 총 길이 $3m$ 의 직선 경로를 추적한다. 선도 차량 로봇의 시작 위치는 $X_1 = 0.8847m$, $Y_1 = 3.9739m$, $\theta_1 = 0.0547rad$ 이며, 목표 속도는 $v_r = 0.1237m/s$ 이다. 추종 차량 로봇은 임의의 좌표에서 시작하는 것을 원칙으로 하며, 시뮬레이션에서의 시작점은 $X_2 = 0.3322m$, $Y_2 = 2.4655m$, $\theta_2 = -0.0279rad$ 이다. 시뮬레이션의 샘플링 시간은 $0.1sec$ 이며, 500 개의 데이터로 50 초 동안 시뮬레이션 한다.

Fig. 4-22 는 선도 차량 로봇과 추종 차량 로봇의 이동 경로를 나타낸 그래프이다. X_g 방향으로 $2.5m$ 지점부터는 선도 차량 로봇과 추종 차량 로봇의 거리가 $2m$ 로 일정하게 유지됨을 보여주고 있으며, 추종 차량 로봇이 초기 위치 차이를 줄여서 목표 대형을 만들었음을 알 수 있다.

목표 경로와 실제 선도 차량 로봇이 움직인 경로의 X_g 방향 추적 오차와 Y_g 방향 추적 오차를 식 (4.41)과 같이 정의하게 되면, 선도 차량 로봇의 목표 경로가 X_g 방향과 수평인 직선이기 때문에 $e_{Lx} = 0$ 이 된다. 그리고 선도 차량 로봇과 추종 차량 로봇의 추종 오차를 식 (4.42)와 같이 정의하면 그 결과는 Fig. 4-23 과 같다. 이 그래프에서 25 초 후에는 모든 오차값들이 0 에 수렴함을 알 수 있다.

$$\begin{cases} e_{Lx} = x_d - X_1 \\ e_{Ly} = y_d - Y_1 \end{cases} \quad (4.41)$$

$$\begin{cases} e_{Fx} = X_2 - X_1 \\ e_{Fy} = \frac{(Y_2 + Y_1)}{2} - 3 \end{cases} \quad (4.42)$$

Fig. 4-24 와 Fig. 4-25 는 선도 차량 로봇과 추종 차량 로봇의 시스템에 입력되는 보조속도벡터 $\eta_1 = [v_{x1} \ \omega_1]^T$, $\eta_2 = [v_{x2} \ \omega_2]^T$ 을 나타낸 그래프이다. 추종 차량 로봇의 초기 위치가 선도 차량 로봇보다 뒤에 있었기 때문에 추종 차량 로봇은 계속적으로 선도 차량 로봇을 추종한다. Fig. 4-24 에서 약 25 초 뒤, 즉 추종 차량 로봇이 선도 차량 로봇을 추종했을 경우에 선도 차량 로봇의 속도인 $0.1237m/s$ 값으로 수렴함을 알 수 있다. Fig. 4-25 에서 추종 차량 로봇의 회전각속도 또한 25 초 후에는 $0rad/s$ 으로 수렴함을 알 수 있다.

Fig. 4-26 는 선도 차량 로봇에서 측정된 전진방향각 θ_1 과 4.1 절에서 제안되어진 추종 차량 로봇에서 설계된 미지 입력 관측기에서 추정된 $\hat{\theta}_{12}$ 값을 나타낸 그래프이다. $\hat{\theta}_{12}$ 의 초기 예측 값이 매우 높으며, 약 25 초 후에 0 에 수렴함을 알 수 있다. Fig. 4-27 는 추종 차량 로봇에서 측정된 θ_2 의 그래프로 약 25 초 후에 식 (4.36)에서 제안된 목표 전진방향각인 $0rad$ 으로 수렴해 감을 알 수 있다.

서버에서 2 대의 차량 로봇으로 전달하는 분산 제어 출력값 h_1 , h_2 중에서 Fig. 4-28 는 수평 대형 함수 h_1 값을 나타낸 그래프이고 Fig. 4-29 는 수직 대형 함수 h_2 값을 나타낸 그래프이다.

Fig. 4-28 에서 분산 제어 출력값인 수평 대형 함수 h_1 은 약 25 초 후에 거의 0 으로 수렴되는 것을 볼 수 있다. 수평 대형 함수 h_1 값이 25 초 후에는 0 으로 수렴하기 때문에, Fig. 4-24 에서 추종 차량 로봇의 x_f 축 성분 전진속도가 25 초 후에는 목표 전진속도인 $0.1237m/s$ 으로 유지되게 된다.

Fig. 4-29 에서 분산 제어 출력값인 수직 대형 함수 h_1 은 약 25 초 후에 거의 0 으로 수렴되는 것을 볼 수 있다. 수직 대형 함수 h_2 값이 25 초 후에는 0 으로 수렴하기 때문에, Fig. 4-25 에서 추종 차량 로봇의 회전각속도가 25 초 후에는 $\omega_r = 0rad/s$ 으로 유지되게 된다.



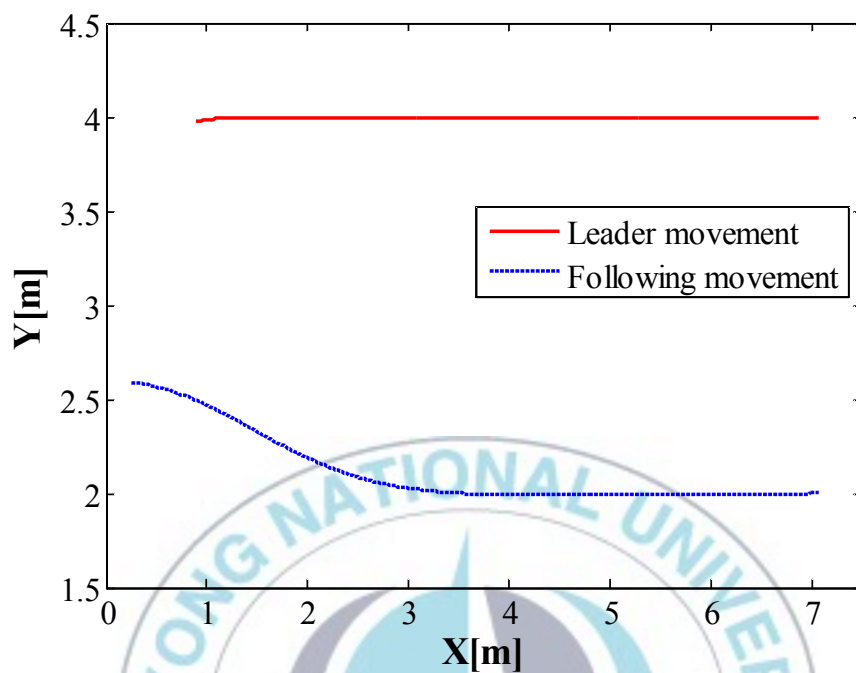


Fig. 4-22 Movements of leader-following vehicle

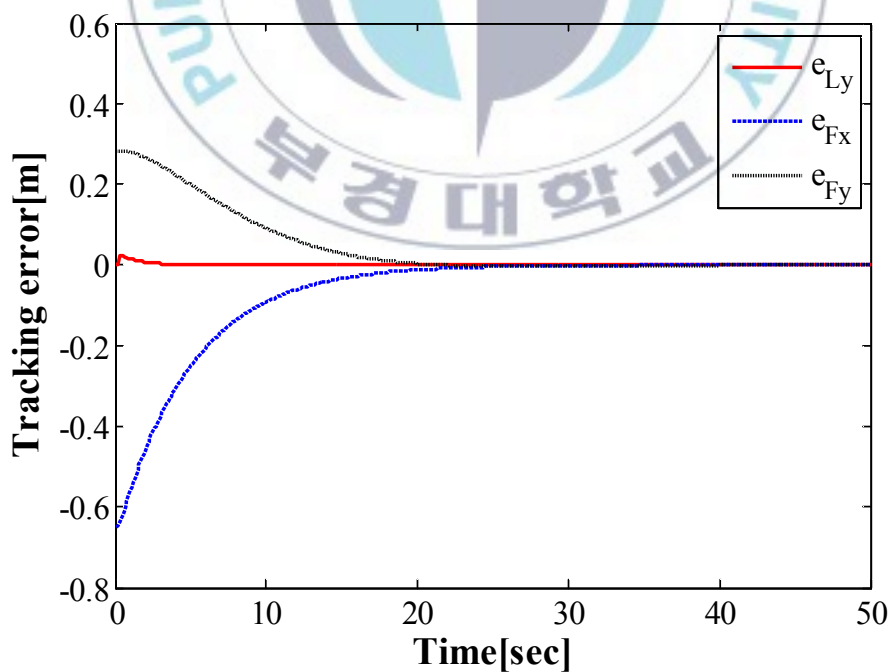


Fig. 4-23 Tracking errors

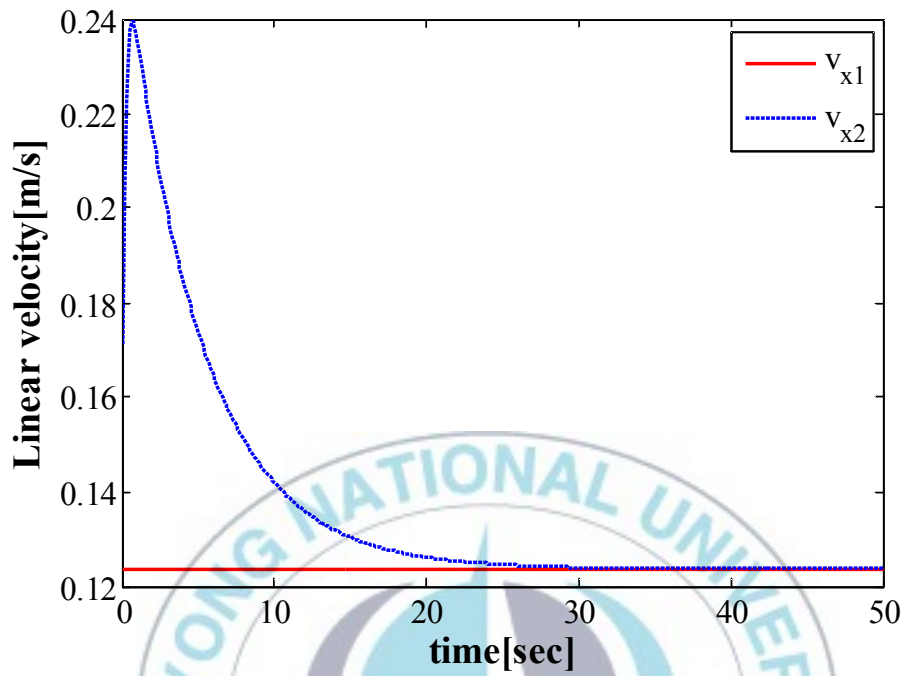


Fig. 4-24 Linear velocities of leader-following vehicle

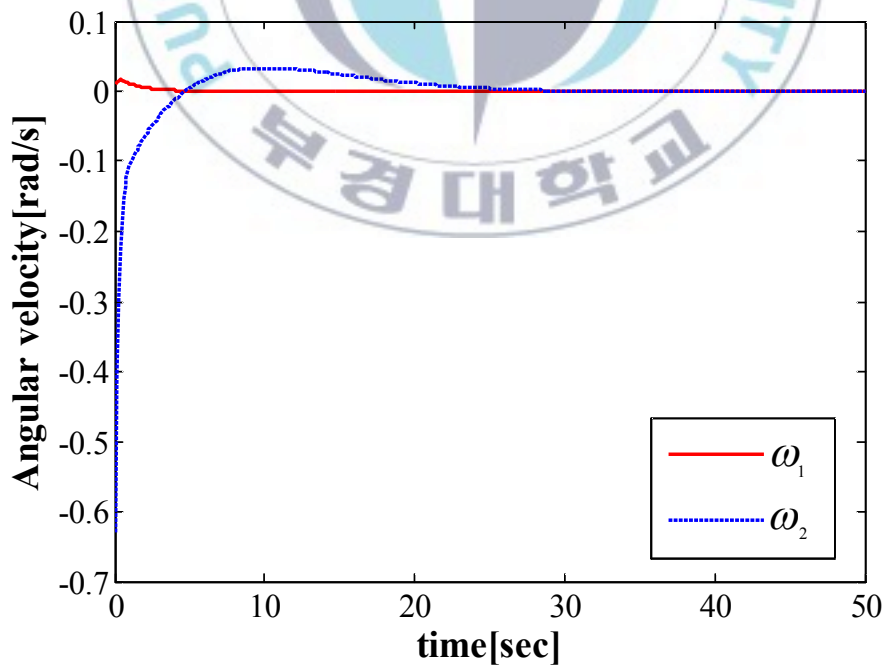


Fig. 4-25 Angular velocities of leader-following vehicle

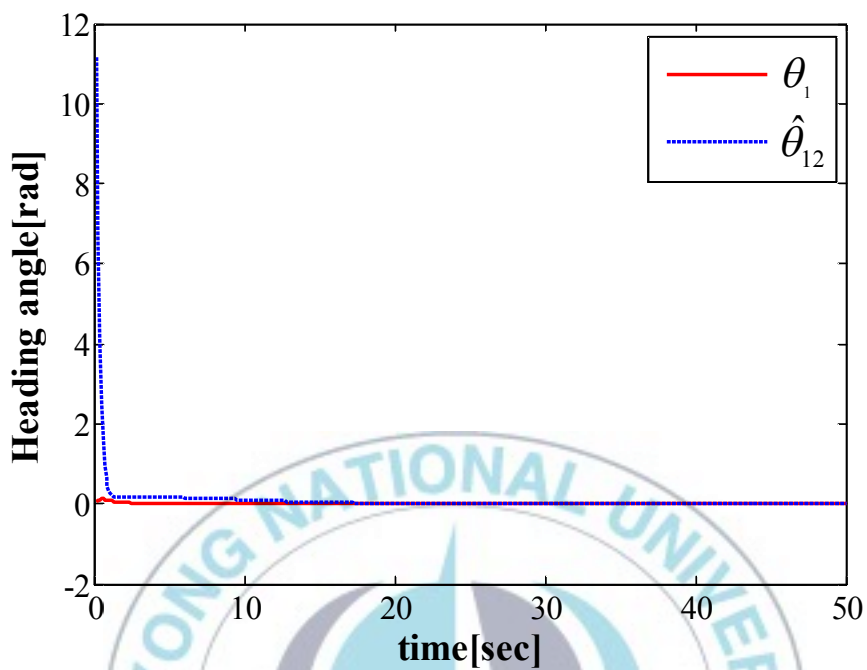


Fig. 4-26 Heading angle of leader vehicle

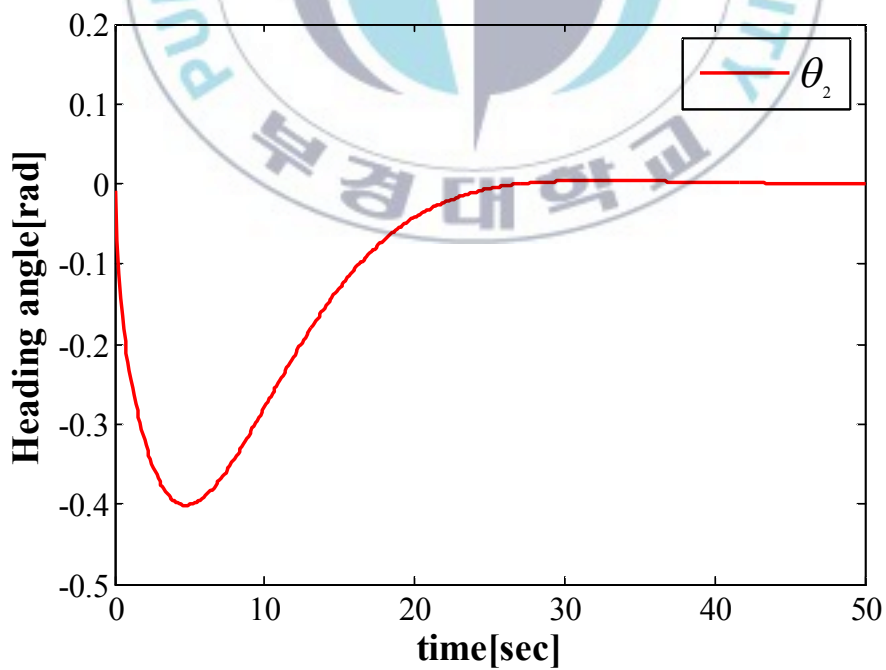


Fig. 4-27 Heading angle of following vehicle

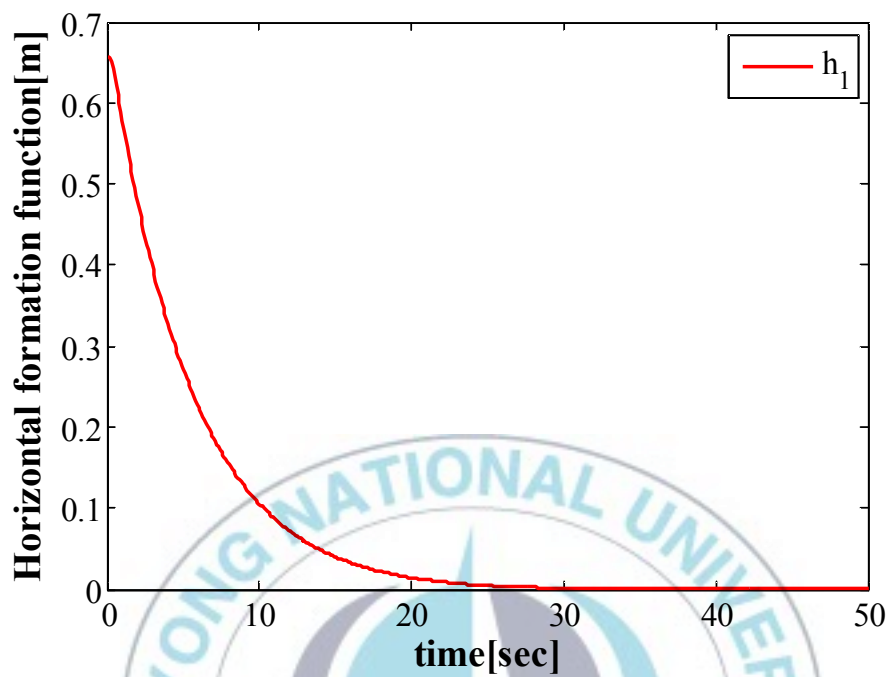


Fig. 4-28 Horizontal formation function

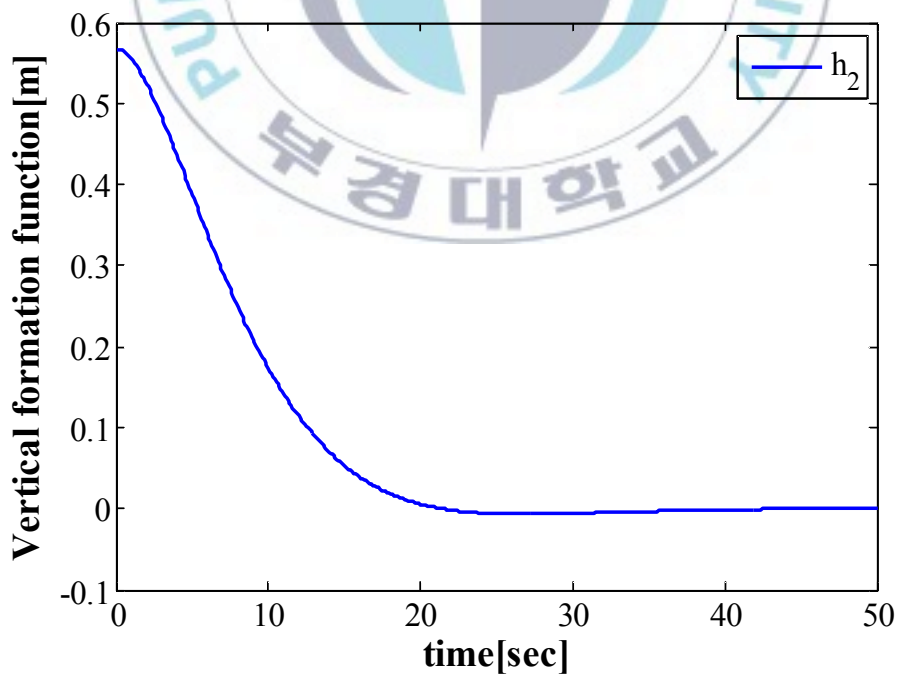


Fig. 4-29 Vertical formation function

4.4.3 실험 결과 및 고찰

실험에서는 식 (4.33)에서 표현된 것과 같이 선도-추종 차량 로봇이 X_g 축에 평행하게 움직이고 Y_g 축의 중간지점 ($y = 3$) 으로부터 일정한 거리를 유지하면서 움직이도록 한다. 선도 차량 로봇은 3 장에서 소개된 LOS 경로 추종 알고리즘을 이용하여, 좌표 (1,3)에서 시작되어 좌표 (4,3)에서 끝나는 총 길이 $3m$ 의 직선 경로를 추적한다. 선도 차량 로봇의 시작위치는 $X_1 = 0.8847m$, $Y_1 = 3.9739m$, $\theta_1 = 0.0547rad$ 이며, 목표 속도는 $v_r = 0.1237m/s$ 이다. 추종 차량 로봇은 임의의 좌표에서 시작하는 것을 원칙으로 하며, 실험에서의 시작점은 $X_2 = 0.3322m$, $Y_2 = 2.4655m$, $\theta_2 = -0.0279rad$ 이다. 실험에서는 2 대의 차량이 움직일 수 있는 공간이 제한적이기 때문에, $X_g = 4m$ 까지 실험하도록 한다.

Fig. 4-30 은 선도 차량 로봇과 추종 차량 로봇의 이동 경로를 나타낸 그래프이다. X_g 방향으로 $1.5m$ 지점부터는 선도 차량 로봇과 추종 차량 로봇의 거리가 $2m$ 로 거의 일정하게 유지됨을 보여주고 있으며, 추종 차량 로봇이 초기 위치 차이를 줄여서 목표 대형을 만들었음을 알 수 있다.

목표 경로와 실제 선도 차량 로봇이 움직인 경로의 X_g 방향 추적 오차와 Y_g 방향 추적 오차를 식 (4.41)과 같이 정의하게 되면, 선도 차량 로봇의 목표 경로가 X_g 방향과 수평인 직선이기 때문에 $e_{Lx} = 0$ 이 된다. 그리고 선도 차량 로봇과 추종 차량 로봇의 추종 오차를 식 (4.42)와 같이 정의하면 그 결과는 Fig. 4-31 과 같다. 선도 차량 로봇은 목표 경로를 $\pm 0.03m$ 이내의 오차로 추적하고 있음을 알 수 있으며, 선도 차량 로봇과 추종 차량 로봇의 추종 오차 e_{Fx} , e_{Fy} 값이 4 초 이후에는 $\pm 0.05m$ 이내로 추종함을 알 수 있다.

Fig. 4-32 와 Fig. 4-33 은 선도 차량 로봇과 추종 차량 로봇의 시스템에 입력되는 보조속도벡터 $\eta_1 = [v_{x1} \ \omega_1]^T$, $\eta_2 = [v_{x2} \ \omega_2]^T$ 을 나타낸 그래프이다. 추종 차량 로봇의 초기 위치가 선도 차량 로봇보다 뒤에 있었기 때문에 추종 차량 로봇은 계속적으로 선도 차량 로봇을 추종한다. Fig. 4-32 에서 약 6 초 뒤, 즉 추종 차량 로봇이 선도 차량을 추종했을 경우에 선도 차량 로봇의 속도인 $0.1237m/s$ 값으로 수렴함을 알 수 있다. Fig. 4-33 에서 추종 차량 로봇의 회전각속도 또한 약 6 초 뒤에는 $\pm 0.01rad/s$ 이내의 오차로 추종함을 알 수 있다.

Fig. 4-34 는 4.1 절에서 제안되어진 추종 차량 로봇에서 설계된 미지 입력 관측기에서 추정된 $\hat{\theta}_{12}$ 값을 나타낸 그래프이다. $\hat{\theta}_{12}$ 의 초기 예측 값이 매우 높으며, 약 6 초 후에 $\pm 0.4rad$ 이내의 오차로 수렴함을 알 수 있다. Fig. 4-35 은 추종 차량 로봇에서 측정된 전진방향각 θ_2 의 그래프로 약 10 초 후에 식 (4.36)에서 제안된 목표 전진방향각인 $0rad$ 으로 수렴해 감을 알 수 있다.

서버에서 2 대의 차량 로봇으로 전달하는 분산 제어 출력값 h_1 , h_2 중에서 Fig. 4-36 는 수평 대형 함수 h_1 값을 나타낸 그래프이고 Fig. 4-37 는 수직 대형 함수 h_2 값을 나타낸 그래프이다.

서버에서 2 대의 차량 로봇으로 전달하는 분산 제어 출력값 h_1 , h_2 중에서 Fig. 4-36 은 수평 대형 함수 h_1 값을 나타낸 그래프이고, Fig. 4-37 은 수직 대형 함수 h_2 값을 나타낸 그래프이다. Fig. 4-36 에서 분산 제어 출력값인 수평 대형 함수 h_1 은 약 6 초 후에 거의 0 으로 수렴되는 것을 볼 수 있으며, Fig. 4-37 에서 분산 제어 출력값인 수평 대형 함수 h_2 은 약 6 초 후에 $\pm 0.1rad$ 이내의 값으로 수렴함을 볼 수 있다.

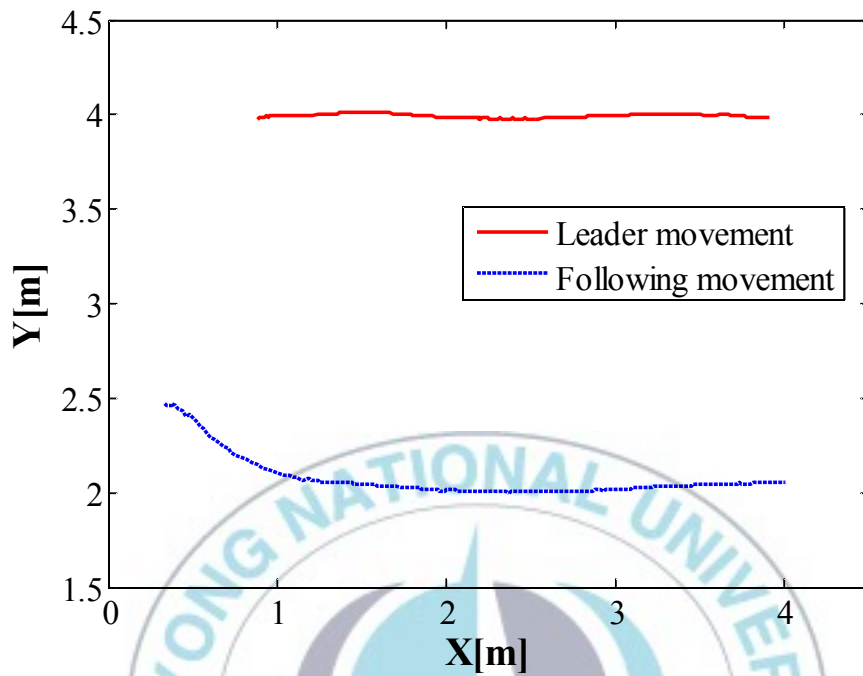


Fig. 4-30 Movements of leader-following vehicle

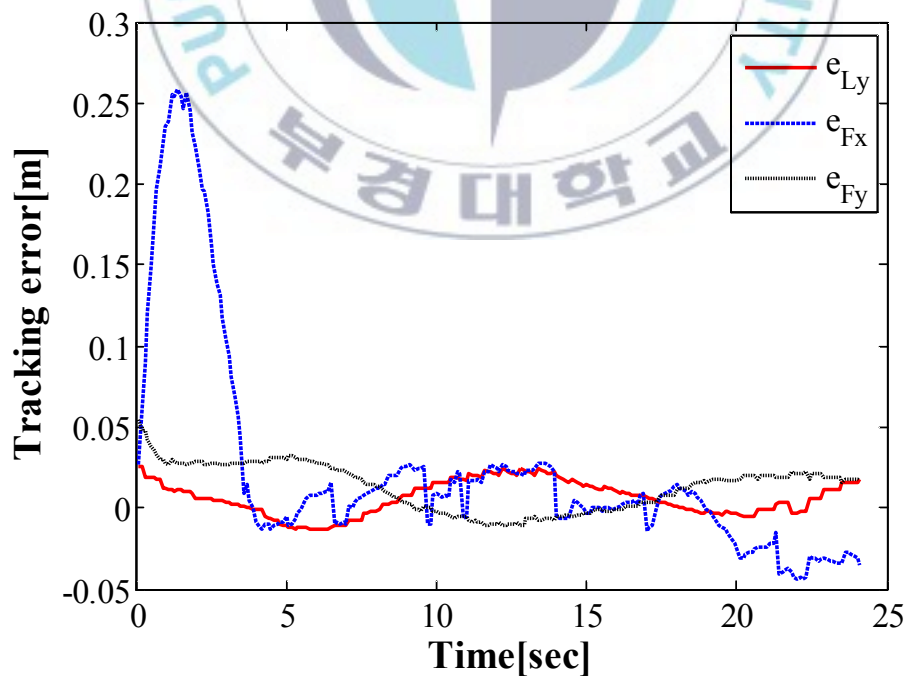


Fig. 4-31 Tracking errors

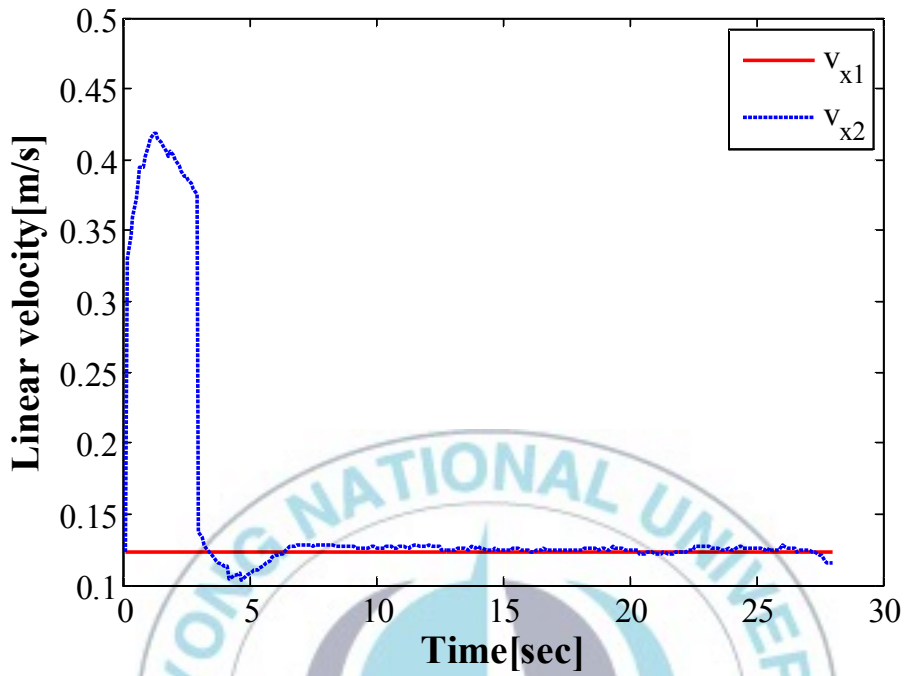


Fig. 4-32 Linear velocities of leader-following vehicle

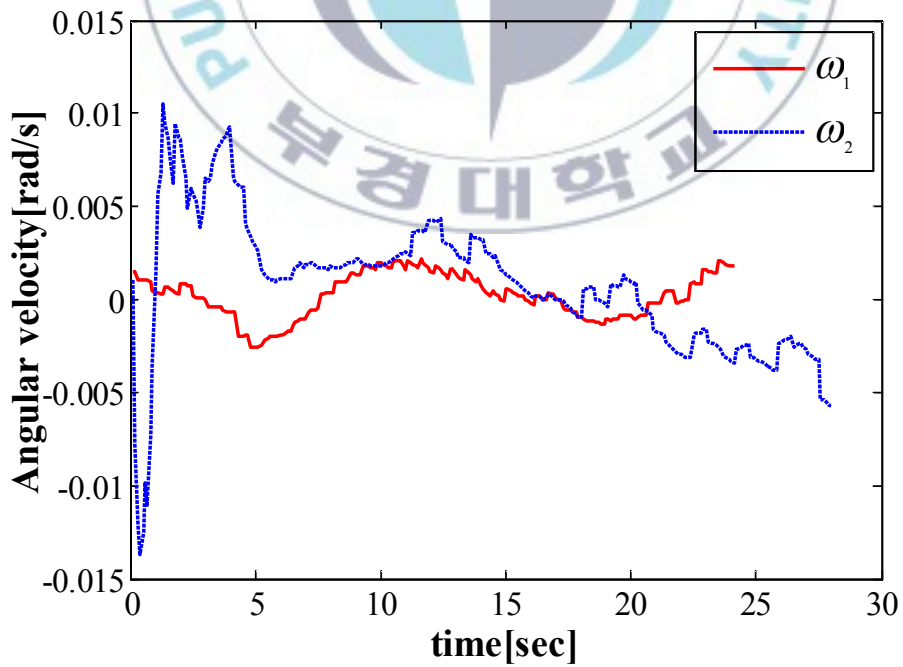


Fig. 4-33 Angular velocities of leader-following vehicle

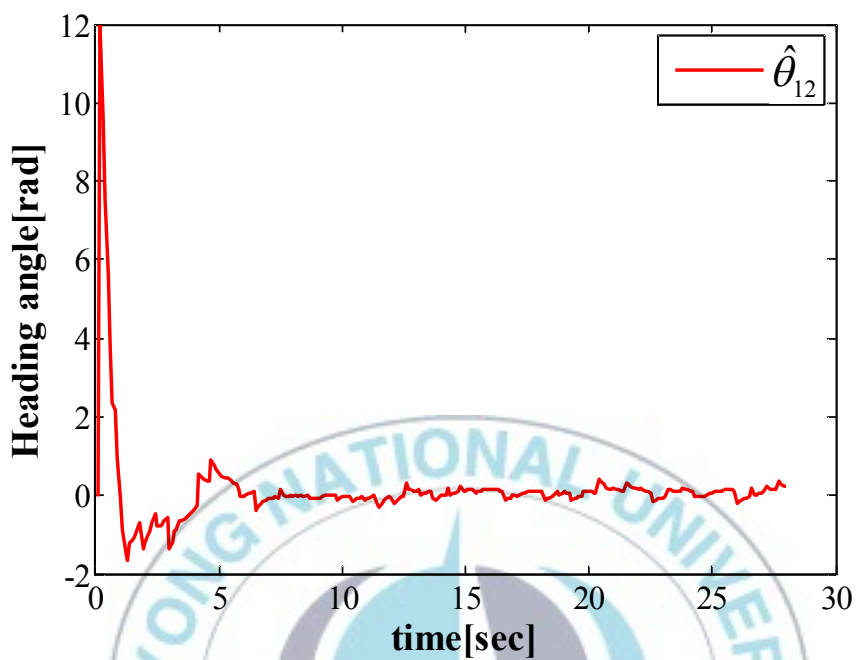


Fig. 4-34 Heading angle of leader vehicle

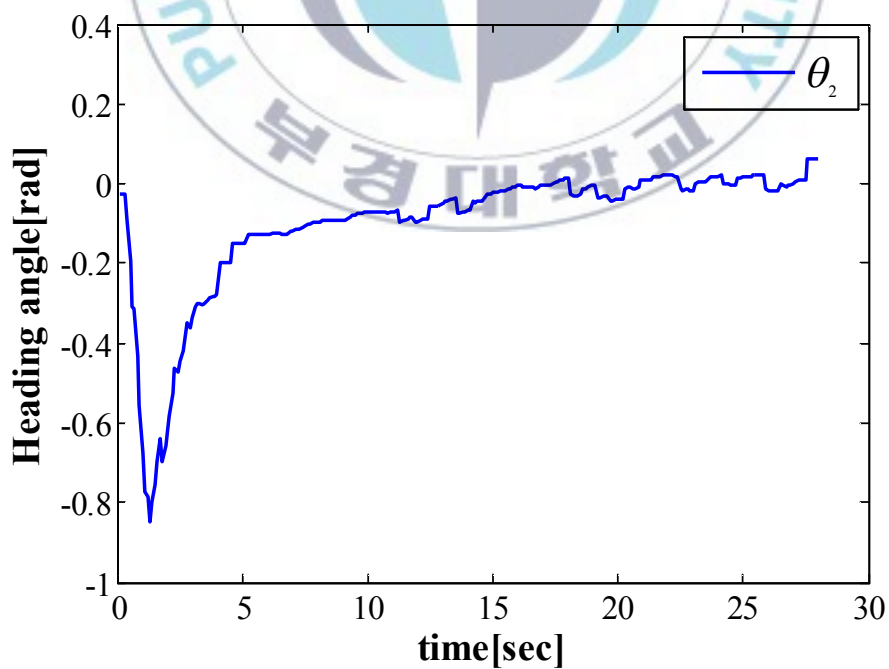


Fig. 4-35 Heading angle of following vehicle

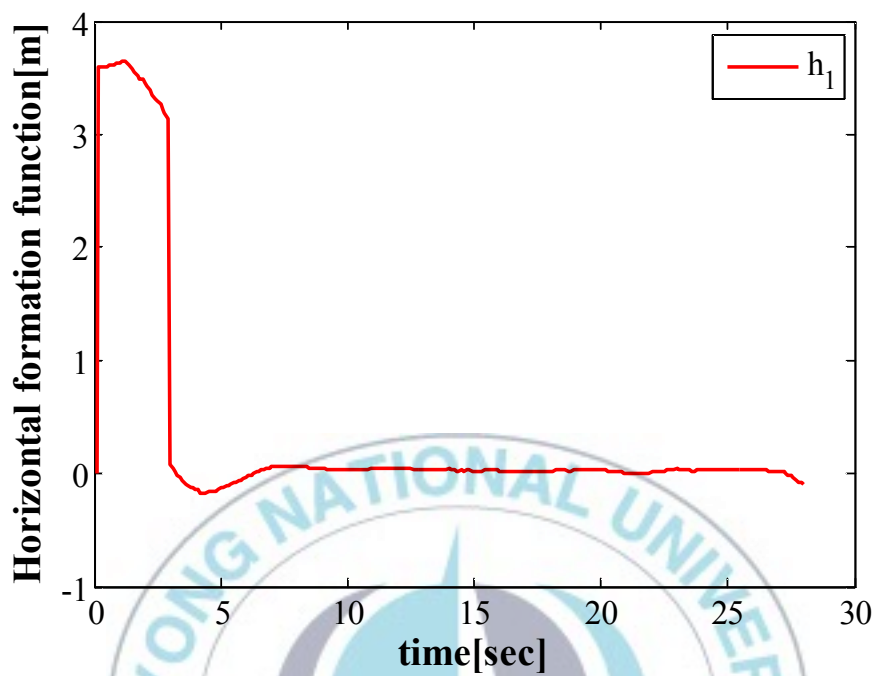


Fig. 4-36 Horizontal formation function

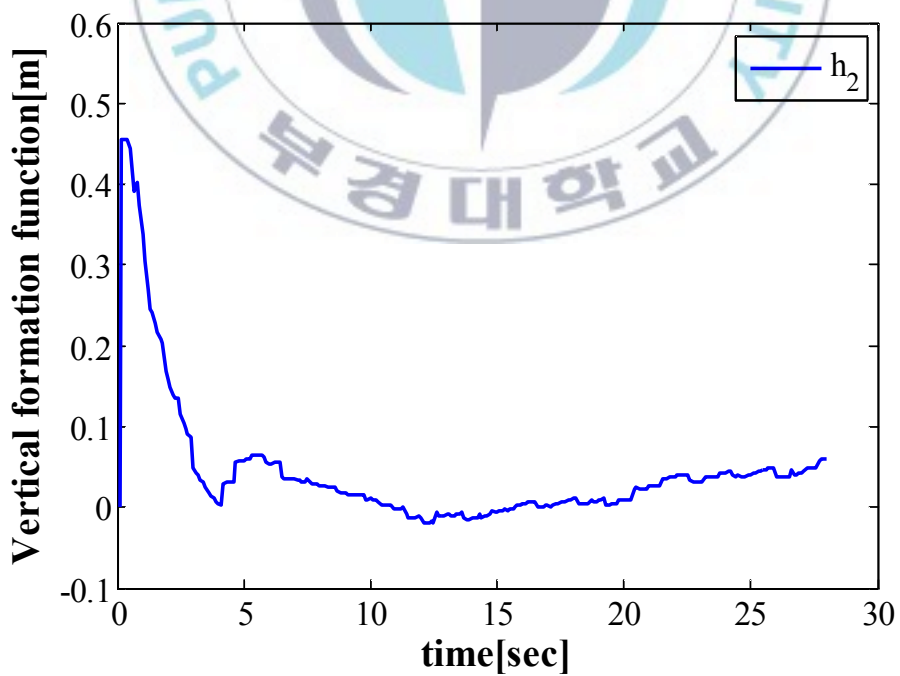


Fig. 4-37 Vertical formation function

제 5 장 결 론

본 연구에서는 수중 로봇의 대형제어 알고리즘을 개발하기 위해, 수중과 유사한 계측 장비와 육상 이동 로봇을 이용하여 미지 입력 관측기에 기반한 분산 제어 방식을 제안하였다. 먼저 단일 차량 로봇에 대한 모델링, 위치 인식 및 경로 추종 알고리즘을 제안하여 위치 인식과 경로 추종 알고리즘에 대해 검증하였다. 그리고 다중 차량 로봇의 대형에 대한 모델링과 이에 대한 관측기를 설계하는 방법을 제안하였다. 또한 이 관측기를 기반으로 한 다중 로봇 시스템의 대형 제어를 위해 Broadcasting 분산 제어기와 선도-추종 분산 제어기를 제시하였다. 제시된 제어기를 구현하기 위해 제어 시스템을 구성하고 개발하였다. 마지막으로 시뮬레이션과 실험을 통하여 제시된 제어기 및 알고리즘의 유효성을 검증하였다. 본 연구의 내용을 요약하면 다음과 같다.

- ✧ 다중 로봇의 대형 제어를 수행하기 위한 실험 기반 장비들을 구성하였다. 우선 리눅스 기반의 임베디드 컨트롤러를 주 컨트롤러로 하는 엔코더가 장착된 2대의 차량 로봇을 제작하고, 각 차량 로봇의 위치를 계측하기 위해 초음파 위치 시스템을 구성하였다. 그리고 분산 제어를 구현하기 위해 리눅스 기반의 서버 시스템과 각 차량의 통신을 위한 무선 네트워크를 구성하였다.
- ✧ 좌우 바퀴의 구동력 차이를 이용하여 선회가 가능한 스키드 조향 방식의 차량 로봇에 대한 운동학적 모델을 제시하였다. 차량 로봇의 바퀴들이 미끄러짐 없이 순수 구동한다는 가정하에서 비홀로노믹 구속 방정식을 구하고 이를 기반한 차량 로봇의 운동학적 방정식을 유도하였다.
- ✧ 차량 로봇의 운동학적 방정식의 매개변수 r , c 는 슬립 및 바퀴의 형상 변화와 무게 중심 측정 오차로 인해 영향을 받는다. 따라서 모델 매개변수 동정을 통하여 오차를 최소로 하는 최적의 r , c 값을 실험을 통해 $r = 0.0409m$, $c = 0.13194m$ 와 같이 선정하였다.
- ✧ 앞서 제안된 운동학적 방정식의 매개변수 r , c 을 이용하여 정확한 차량 로봇의 위치를 인식하기 어렵다. 그리고 초음파 위치 시스템은 벽면 반사에 의한 불규칙한 측정 오차를 가진다. 따라서 차량 로봇의 위치를 정확하게 인식하기 위해서, 초음파 위치 시스템으로부터 계측된 값과 로봇으로부터 획득한 엔코더 값(회전속도)을 운동학 방정식에서 대입하여 얻은 위치 예측 값으로 위치를 보정하는 위치 인식 알고리즘을 제시하고 검증하였다.
- ✧ 앞서 제안된 위치 인식 알고리즘을 기반으로 단일 차량 로봇을 이용한

경로 추종 알고리즘을 제안하였다. 단일 차량 로봇으로 주어진 경로를 움직이도록 하기 위해서 LOS 개념을 적용하였으며, 경로 추종에 필요한 매개변수들을 $L_{los} = 0.18m$, $v_n = 0.1237m/s$ 와 같이 선정하였다. 차선 변경과 유턴 경로의 경로 추종 실험에서 $\pm 0.04m$ 의 오차 범위를 가진다.

- ☆ 앞서 제안된 알고리즘을 활용하여 2대의 차량 로봇으로 구성된 그룹에 대하여 대형 모델을 제시한다. 그리고 대형 제어를 위한 미지 입력 관측기를 설계하고 이를 기반으로 분산 제어기를 설계하였다. 4.3 절에서는 시뮬레이션과 실험을 통해 2대의 차량 로봇이 동시에 움직이면서 일정한 대형을 유지하는 Broadcasting 분산 제어 방법의 특징을 잘 보여주었다. 그리고 4.4 절에서는 시뮬레이션과 실험을 통해 추종 차량 로봇이 선도 차량 로봇으로부터 일정한 대형을 유지하는 선도-추종 분산 제어 방법의 특징을 잘 보여주었다.

본 연구에서 제안된 위치 인식 알고리즘, 경로 추종 알고리즘과 미지 입력 기반 분산 제어 방법은 이동 차량 로봇을 이용하는 많은 경우에 활용될 것으로 판단된다. 특히, 제안된 위치 인식 알고리즘은 수중과 비슷한 계측 장비를 이용하였기 때문에 육상 이동 로봇뿐만 아니라 수중 로봇에도 적용될 수 있을 것으로 기대된다.

차후에는 동역학에 기반한 모델링을 수행하여 보다 정밀한 제어기를 구현하고자 한다. 그리고 차량 로봇의 바퀴와 지면 사이의 슬립을 고려하여 더욱 더 수중 환경에 유사한 모델링을 수행하고자 한다. 그리고 현재의 초음파 위치 시스템보다 더욱 정밀한 시스템을 구축하여 본 연구에서 제안된 위치 인식 알고리즘을 보다 정밀하게 검증하고 보완하고자 한다. 또한 경로 추종 알고리즘에 거리에 따라 차량 로봇의 이동 속도를 변화시키는 개념을 도입하고, 경로를 추종함에 있어서 위치뿐만 아니라 차량 로봇의 전진방향각도 고려하도록 경로 추종 알고리즘을 개선하고자 한다. 마지막으로 이를 토대로 하여 좀더 정밀한 다중 로봇의 대형 제어를 다양한 방식으로 적용하여 분산 제어 연구를 수행하고자 한다.

References

- [1] D. Lhomme-Desages, Ch. Grand and J-C. Guinot, "Trajectory Control of a four-Wheel Skid-Steering Vehicle over Soft Terrain using a Physical Interaction Model", 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 1164-1169, April 2007.
- [2] Tae-Kyeong Yeu, Soung-Jea Park, Sup Hong, Hyung-Woo Kim and Jong-Su Choi, "Path Tracking using Vector Pursuit Algorithm for Tracked Vehicles Driving on the Soft Cohesive Soil", SICE-ICASE International Joint Conference, Korea , Oct 2006.
- [3] 여태경, 박성재, 홍섭, 최종수, 김형우, "Vector Pursuit 알고리즘을 이용한 궤도차량의 경로 추적에 관한 연구", JOURNAL OF SHIPS & OCEAN ENGINEERING, Vol.42, pp. 93-100, December 2006.
- [4] 이동환, 김수용, 이만형, "유비쿼터스 컴퓨팅의 실내 측위 기술을 위한 U-SAT 시스템의 적용에 관한 연구", Journal of Control, Automation, and Systems Engineering Vol. 11, No. 8, August 2006.
- [5] Krzysztof Kozłowski, Dariusz Pazderski, "Modeling and Control of a 4-wheel Skid-steering Mobile Robot", Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 4, 477-496, 2004.
- [6] Nguyen, A. Ha, Q. Huang, S. and Trinh, H. " Observer-Based Decentralized Approach to Robotic Formation Control", Proceedings of the 2004 Australasian Conference on Robotics & Automation, Australian Robotics & Automation Association, Australia, pp. 1-8, 2004.
- [7] Pazderski D., Kozłowski K. and Dixon W.E., "Tracking and Regulation Control of a Skid Steering Vehicle", Amer. Nuclear Soci, 10-th Int. Topical Meeting Robotics and Remote Systems, Gainesville, Florida, USA, pp. 369-376, 2004.
- [8] Q.P. Ha and H. Trinh, "Observerbased Control of Multi-agent Systems under Decentralized Information Structure", Int. J. Sys. Science, in press, July 2004.

- [9] S. Monteiro, M. Vaz, and E. Bicho, "Attractor Dynamics Generates Robot Formations: from Theory to Implementation", Proc. IEEE International Conference on Robotics & Automation, pp. 2582-2587, 2004.
- [10] T. D. Barfoot and C. M. Clark, "Motion Planning for Formations of Mobile Robots", Robotics and Autonomous Systems, pp. 65-78, February 2004.
- [11] Kozłowski K. and Pazderski D., "Control of a Four-wheel Vehicle Using Kinematic Oscillator", Proc. 6-th Int. Conf. Climbing and Walking Robots and their Supporting Technologies, CLAWAR 2003, Catania, Italy, pp. 135-146, 2003.
- [12] Trong Hieu Bui, Tan Lam Chung, Sang Bong Kim, "Adaptive Tracking Control of Two-Wheeled Welding Mobile Robot with Smooth Curved Welding Path", KSME International Journal, Vol. 17, No. 11, pp. 1682-1692, 2003.
- [13] D.J. Stilwell, "Decentralized Control Synthesis for a Platoon of Autonomous Vehicles", Proc. IEEE International Conference on Robotics & Automation, pp. 744-749, 2002.
- [14] S. Carpin and L.E. Parker, "Cooperative Leader Following in a Distributed Multi-Robot System", Proc. IEEE International Conference on Robotics & Automation, pp. 2994-3001, 2002.
- [15] Tamio Arai, Enrico Pagello, Lynne E. Parker, "Editorial: Advances in Multi-Robot System", IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, pp. 665-661, October 2002.
- [16] T. D. Barfoot, C. M. Clark, and S. M. Rock, "Kinematic Path-planning for Formations of Mobile Robots with a Nonholonomic Constraint" In IEEE International Conference on Intelligent Robots and Systems, October 2002.
- [17] Dixon W.E., Dawson D.M., Zergeroglu E. and Behal A., "Nonlinear

- Control of Wheeled Mobile Robots”, London: Springer, 2001.
- [18] J. P. Desai, “Modeling Multiple Teams of Mobile Robots: A Graph Theoretic Approach”, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Maui, Hawaii, USA, pp. 381–386, 2001.
- [19] J.P. Desai, J. Ostrowski, and V. Kumar, “Modelling and Control of Formations of Nonholonomic Mobile Robots”, IEEE Trans. On Robotics and Automation, Vol. 17, no. 6, pp. 905–908, 2001.
- [20] R.W. Beard, J. Lawton, and F.Y. Hadaegh, “A Coordination Architecture for Spacecraft Formation Control”, IEEE Trans. on Control Systems Technology, vol 9, no. 6, pp. 777–790, 2001.
- [21] D.J. Stilwell and B.E. Bishop, “A Framework for Decentralized Control of Autonomous Vehicles”, Proc. IEEE Int. Conf. Robotics & Automation, pp. 2358–2363, 2000.
- [22] H. Trinh and Q. Ha, “Design of Linear Functional Observers for Linear Systems with Unknown Inputs”, International Journal of Systems Science, vol. 31, no. 6, pp. 741–749, 2000.
- [23] M. Ahmadi, V. Polotski, and R. Hurteau, “Path Tracking Control of Tracked Vehicles”, 2000 IEEE International Conference on Robotics & Automation, San Francisco, pp. 2938–2943, 2000.
- [24] M.S. De Queiroz, V. Kapila, and Q. Yan, “Adaptive Nonlinear Control of Multiple Spacecraft Formation flying”, J. Guidance, Control, and Dynamics, vol. 23, pp. 385–390, 2000.
- [25] T. Balch and M. Hybinette, “Social Potentials for Scalable Multi-Robot Formations”, Proc. IEEE Int. Conf. Robotics & Automation, San Francisco, CA, pp. 73–80, 2000.
- [26] W. Kang, N. Xi, and A. Sparks, “Formation Control of Autonomous Agents in 3D Workspace”, Proc. IEEE International Conference on Robotics & Automation, pp. 1755–1760, 2000.

- [27] Caracciolo L., De Luca A. and Iannitti S., "Trajectory Tracking Control of a Four-wheel Differentially Driven Mobile Robot", IEEE Int. Conf. Robotics and Automation, Detroit, MI, pp. 2632-2638, 1999.
- [28] M.A. Lewis and K.H. Tan, "High Precision Formation Control of Mobile Robot Using Virtual Structures", Autonomous Robots, Vol. 4, pp. 387-403, 1997.
- [29] X. Yun, G. Alptekin, and O Albayrak, "Line and Circle Formation of Distributed Physical Mobile Robots", J. Robotic Systems, vol 14, no. 2, pp. 63-76, 1997.
- [30] G. Schoner, M. Dose, and C. Engels, "Dynamics of Behavior: Theory and Applications for Autonomous Robot Architectures", Robotics and Autonomous Systems, Vol. 16, pp. 213-245, 1995.
- [31] Q. Chen and J.Y.S. Luh, "Coordination and Control of a Group of Small Mobile Robots", Proc. IEEE International Conference on Robotics & Automation, pp. 2315-2320, 1994.
- [32] S. Sheikholeslam and C.A Desoer, "Control of Interconnected Nonlinear Dynamical Systems: the Platoon Problem", IEEE Trans. on Automatic Control, vol. 37, no. 6, pp. 806-810, 1992.
- [33] S. Premvuti and S. Yuta. "Consideration on the Cooperation of Multiple Autonomous Mobile Robots", In proceedings of the IEEE International Workshop of Intelligent Robots and Systems, pp. 59-63, Tsuchiura, Japan, 1990
- [34] H. Asama, A. Matsumoto and Y. Ishida "Design of an autonomous and distributed robot system : ACTRESS", In Prodeedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 283-290, Tsukuba, Japan, 1989.
- [35] G. Beni. "The concept of cellular robot", In proceedings of Third IEEE Symposium on Intelligent Control. Pp. 57-61, Arlington, Virginia, 1988.

Publications and Conferences

Conferences

- [1] 여태경, 윤석민, 홍섭, 박성재, “다중 로봇의 경로 및 대형 제어를 위한 실험 기반 구축”, 2008 한국해양학회 선박해양기술, 2008.
- [2] 윤석민, 여태경, 박성재, 홍섭, 김상봉, “초음파 위치 센서를 이용한 차량 로봇의 경로 추종에 관한 연구”, 2008 대한기계학회 추계학술대회, pp. 795-800, 2008.
- [3] 윤석민, 여태경, 박성재, 홍섭, 김상봉, “초음파 위치 시스템 및 차량 파라미터 예측을 통한 위치추정 알고리즘 개발”, 한국해양공학회 2008 년도 추계학술대회 논문집, pp. 251-254, 2008.
- [4] 박성재, 여태경, 윤석민, 홍섭, 김형우, 최중수, “2 대의 집광기를 이용한 심해저 망간 단괴 광구의 최적 Sweeping 기법 연구”, 한국해양공학회 2008 년도 추계학술대회 논문집, pp. 279-282, 2008.
- [5] 이도경, 윤석민, 박성재, 여태경, 김학경, 김상봉, 홍섭, “시험집광기 운용 데이터베이스 시스템 개발 “, 한국해양공학회 2008 년도 추계학술대회 논문집, pp. 231-234, 2008.
- [6] Do Kyoung LEE, Suk Min Yoon, Soung Jea Park, Tae Kyeong Yeu, Sup Hong, Hak Kyeong Kim, Sang Bong Kim, “Development of Database System for Test Miner System”, Proceedings of 2008 International Symposium on Advanced Mechanical and Power Engineering, pp. 104-109, Oct, 2008.
- [7] 박성재, 여태경, 윤석민, 홍섭, 김형우, 최중수, “심해저 집광기의 항법을 위한 센서 필터 설계”, 2008년도 한국해양과학기술협의회 공동학술대회, May, pp. 189-192, 2008.
- [8] 여태경, 윤석민, 홍섭, 박성재, 최중수, 김형우, “시험집광기 운용 프로그램의 구조 설계에 관한 연구”, 2008년도 한국해양과학기술협의회 공동학술대회, May, pp. 185-188, 2008.

- [9] 윤석민, 여태경, 박성재, 홍섭, 이도경, 김상봉, “시험 집광기 개별 센서 및 액추에이터 성능시험 및 교정”, 2008년도 한국해양과학기술협의회 공동학술대회, May, pp. 193-196, 2008.
- [10] Soung-Jea Park, Tae-Kyeong Yeu, Sup Hong, Suk Min Yoon and Sang-Bong Kim, “Development of Simulator of Control and Monitoring System for Deep-seabed Miner”, 2007 International Symposium on Mechatronics and Automatic Control, Oct 2007.
- [11] Suk Min Yoon, Vo Hoang Duy, Nguyen Thanh Phuong, Sang Bong Kim and Myung Suk Oh, “A New Approach for Development op the quadruped Walking Robot”, 2007 International Symposium on Mechatronics and Automatic Control, Oct 2007.
- [12] 윤석민, 박성재, 여태경, 김학경, 김상봉, 홍섭, “시험 집광기 실시간 운용 프로그램 개발”, 한국해양공학회 2007년도 추계학술대회 논문집, pp. 363-366, 2007.
- [13] 박성재, 여태경, 홍섭, 윤석민, 김상봉, “심해저 집광기의 제어·계측용 시뮬레이터와 GUI의 개발”, 2007년도 한국해양과학기술협의회 공동학술대회, pp. 2019-2022, 2007.
- [14] Vo Hoang Duy, Nguyen Thanh Phuong, Suk Min Yoon, Hak Kyeong Kim, and Sang Bong Kim, “A New Approach for Designing Quadruped Robot”, IEEE International Conference on Mechatronics, Japan, May 2007.
- [15] Jae Sung Im, Hee Suk Lee, Suk Min Yoon, Sang Bong Kim, “Acoustic Wave Detection Method for Third Party Damage Monitoring System on Natural Gas Pipeline”, SICE, 2005

Publications

- [1] 박성재, 여태경, 윤석민, 홍섭, 김형우, 최종수, 김상봉, “심해저 망간단괴 집광기의 운영 S/W 및 DBMS 개발”, 2008 한국해양학회 바다지, 2008.

부 록

부록 A 증명

증명 1]

점 P_d 는 $\overline{P_n P_{n+1}}$ 을 $m_d : n_d$ 으로 내분하는 내분점이라고 하면 식 (A1)이 성립한다.

$$\begin{cases} u = \frac{m_d}{m_d + n_d} \\ x_d = \frac{m_d x_{n+1} + n_d x_n}{m_d + n_d} = x_n + u(x_{n+1} - x_n) \\ y_d = \frac{m_d y_{n+1} + n_d y_n}{m_d + n_d} = y_n + u(y_{n+1} - y_n) \end{cases} \quad (A1)$$

$\overline{P_d COM}$ 와 $\overline{P_n P_{n+1}}$ 가 수직이기 위해서는 식 (A2)를 만족해야만 한다.

$$\begin{aligned} 0 &= (x_d - X)(x_{n+1} - x_n) + (y_d - Y)(y_{n+1} - y_n) \\ &= (x_n + u(x_{n+1} - x_n) - X)(x_{n+1} - x_n) \\ &\quad + (y_n + u(y_{n+1} - y_n) - Y)(y_{n+1} - y_n) \\ &= (x_n - X)(x_{n+1} - x_n) + u(x_{n+1} - x_n)^2 \\ &\quad + (y_n - Y)(y_{n+1} - y_n) + u(y_{n+1} - y_n)^2 \end{aligned} \quad (A2)$$

식 (A2)을 u 에 대하여 정리하면 식 (A3)과 같다.

$$u = \frac{(X - x_n)(x_{n+1} - x_n) + (Y - y_n)(y_{n+1} - y_n)}{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad (A3)$$



증명 2]

i 번째 시스템의 추정 오차는 다음과 같이 정의한다.

$$\boldsymbol{\varepsilon}_i(\mathbf{t}) = \mathbf{z}_i(\mathbf{t}) - \mathbf{U}_i \mathbf{S}(\mathbf{t}) \quad (\text{A4})$$

식 (A4)의 도함수는 식 (4.5)와 식 (4.8)에서 식 (4.13)까지를 이용하여 식 (A5)와 식 (A6)과 같이 유도된다.

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}}_i &= \dot{\mathbf{z}}_i - \mathbf{U}_i \dot{\mathbf{S}} \\ &= \hat{\mathbf{A}}_i \mathbf{z}_i + \hat{\mathbf{B}}_i \mathbf{y}_i + \hat{\mathbf{J}}_i \mathbf{u}_i - \mathbf{U}_i (\mathbf{A} \mathbf{S} + \mathbf{B}_i \mathbf{u}_i + \mathbf{B}_{ri} \mathbf{u}_{ri}) \\ &= \hat{\mathbf{A}}_i \mathbf{z}_i + \hat{\mathbf{B}}_i \mathbf{C}_i \mathbf{S} + \hat{\mathbf{J}}_i \mathbf{u}_i - \mathbf{U}_i (\mathbf{A} \mathbf{S} + \mathbf{B}_i \mathbf{u}_i + \mathbf{D}_i \mathbf{u}_{ri}) \\ &= \hat{\mathbf{A}}_i \mathbf{z}_i + (\hat{\mathbf{B}}_i \mathbf{C}_i - \mathbf{U}_i \mathbf{A}) \mathbf{S} + (\hat{\mathbf{J}}_i - \mathbf{U}_i \mathbf{B}_i) \mathbf{u}_i - \mathbf{U}_i \mathbf{D}_i \mathbf{u}_{ri} \\ &= \hat{\mathbf{A}}_i \mathbf{z}_i + (\hat{\mathbf{A}}_i \mathbf{U}_i + \hat{\mathbf{B}}_i \mathbf{C}_i - \mathbf{U}_i \mathbf{A} - \hat{\mathbf{A}}_i \mathbf{U}_i) \mathbf{S} \end{aligned} \quad (\text{A5})$$

$$\begin{aligned} &= \hat{\mathbf{A}}_i \mathbf{z}_i - \hat{\mathbf{A}}_i \mathbf{U}_i \mathbf{S} \\ &= \hat{\mathbf{A}}_i (\mathbf{z}_i - \mathbf{U}_i \mathbf{S}) \\ &= \hat{\mathbf{A}}_i \boldsymbol{\varepsilon}_i \\ \boldsymbol{\varepsilon}_i(\mathbf{t}) &= e^{\hat{\mathbf{A}}_i \mathbf{t}} \boldsymbol{\varepsilon}_i(0) \end{aligned} \quad (\text{A6})$$

식 (A4)와 식 (4.5)를 식 (4.8)의 $\hat{\mathbf{S}}_i(\mathbf{t})$ 에 대입하고 식 (4.14)를 이용하면, 식 (A7)과 같이 구해진다.

$$\begin{aligned} \hat{\mathbf{S}}_i &= \hat{\mathbf{C}}_i \mathbf{z}_i + \hat{\mathbf{D}}_i \mathbf{y}_i \\ &= \hat{\mathbf{C}}_i (\boldsymbol{\varepsilon}_i + \mathbf{U}_i \mathbf{S}) + \hat{\mathbf{D}}_i \mathbf{C}_i \mathbf{S} \\ &= \hat{\mathbf{C}}_i \boldsymbol{\varepsilon}_i + (\hat{\mathbf{C}}_i \mathbf{U}_i + \hat{\mathbf{D}}_i \mathbf{C}_i) \mathbf{S} \\ &= \hat{\mathbf{C}}_i \boldsymbol{\varepsilon}_i + \mathbf{S} \end{aligned} \quad (\text{A7})$$

식 (A7)을 이용하여 식 (4.9)를 다시 표현하면 식 (A8)과 같다.

$$\mathbf{e}_i(\mathbf{t}) = \mathbf{S}(\mathbf{t}) - \hat{\mathbf{S}}_i(\mathbf{t}) = -\hat{\mathbf{C}}_i \mathbf{e}_i = -\hat{\mathbf{C}}_i e^{\hat{\mathbf{A}}_i \mathbf{t}} \mathbf{e}_i(0) \quad (\text{A8})$$

식 (4.10)에서 $\hat{\mathbf{A}}_i$ 가 안정행렬이므로 식 (A6)에서 $t \rightarrow \infty$ 일 때 $\mathbf{e}_i(\mathbf{t}) \rightarrow 0$ 이다. 식 (4.10)을 만족할 때, 식 (A6)에서 $t \rightarrow \infty$ 일 때 $\mathbf{e}_i(\mathbf{t}) \rightarrow 0$ 이므로 식 (A8)에 의해 $t \rightarrow \infty$ 일 때 $\mathbf{e}_i(\mathbf{t}) \rightarrow 0$ 이다. ◀

증명 3]

$\text{rank} \mathbf{D}_i = q_i$ 이기 때문에, $\mathbf{D}_i^+ \mathbf{D}_i = \mathbf{I}_{q_i}$ 인 행렬 $\mathbf{D}_i$ 의 좌역행렬(left inverse) $\mathbf{D}_i^+ \in \mathfrak{R}^{q_i \times n_i}$ 가 존재하고, $\text{rank} \mathbf{U}_i = n - q_i$ 인 조건에 의하여 $\text{Ker } \mathbf{U}_i \cap \text{Ker } \mathbf{D}_i^+ = \{0\}$ 이므로, 식 (4.13)의 $\mathbf{U}_i$ 는 식 (A9) 조건을 만족한다.

$$\text{rank} \begin{bmatrix} \mathbf{U}_i \\ \mathbf{D}_i^+ \end{bmatrix} = n \quad (\text{A9})$$

미지 입력 관측기의 설계 조건을 검토하기 위해 식 (A10)을 고려한다.

$$\left\{ \begin{array}{l} \text{rank} \begin{bmatrix} \mathbf{U}_i & 0 \\ \mathbf{D}_i^+ & 0 \\ 0 & \mathbf{I}_{r_i} \end{bmatrix} = \text{rank} \begin{bmatrix} \mathbf{I}_n & 0 \\ -\mathbf{D}_i^+ (\mathbf{s} \mathbf{I}_n - \mathbf{A}) & \mathbf{I}_{q_i} \end{bmatrix} = n + q_i \\ \text{rank} \begin{bmatrix} \mathbf{I}_n & 0 & \mathbf{s} \hat{\mathbf{D}}_i \\ 0 & \mathbf{I}_{q_i} & 0 \\ 0 & 0 & \mathbf{I}_{r_i} \end{bmatrix} = n + q_i + r_i \end{array} \right. \quad (\text{A10})$$

식 (A11)의 좌변은 식 (4.15)으로부터 식 (A11)의 우변과 같이 계산된다.

$$\begin{bmatrix} \mathbf{U}_i & 0 \\ \mathbf{D}_i^+ & 0 \\ 0 & \mathbf{I}_{r_i} \end{bmatrix} \begin{bmatrix} \mathbf{sI}_n - \mathbf{A} & \mathbf{D}_i \\ \mathbf{C}_i & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & 0 \\ -\mathbf{D}_i^+ (\mathbf{sI}_n - \mathbf{A}) & \mathbf{I}_{q_i} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i (\mathbf{sI}_n - \mathbf{A}) & 0 \\ 0 & \mathbf{I}_{q_i} \\ \mathbf{C}_i & 0 \end{bmatrix} \quad (\text{A11})$$

식 (A10)과 식 (A11)을 이용하여 정리하면 식 (A12)와 같다.

$$\begin{bmatrix} \mathbf{I}_n & 0 & \mathbf{s}\hat{\mathbf{D}}_i \\ 0 & \mathbf{I}_{q_i} & 0 \\ 0 & 0 & \mathbf{I}_{r_i} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i (\mathbf{sI}_n - \mathbf{A}) & 0 \\ 0 & \mathbf{I}_{q_i} \\ \mathbf{C}_i & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{sI}_n - \mathbf{U}_i \mathbf{A} & 0 \\ 0 & \mathbf{I}_{q_i} \\ \mathbf{C}_i & 0 \end{bmatrix} \quad (\text{A12})$$

식 (A10)에서 식 (A12)까지 관계식으로부터 식 (A13)의 조건이 성립한다.

$$\text{rank} \begin{bmatrix} \mathbf{sI}_n - \mathbf{U}_i \mathbf{A} & 0 \\ 0 & \mathbf{I}_{q_i} \\ \mathbf{C}_i & 0 \end{bmatrix} = \text{rank} \begin{bmatrix} \mathbf{sI}_n - \mathbf{A} & \mathbf{D}_i \\ \mathbf{C}_i & 0 \end{bmatrix} = \text{rank} \begin{bmatrix} \mathbf{sI}_n - \mathbf{U}_i \mathbf{A} \\ \mathbf{C}_i \end{bmatrix} + q_i \quad (\text{A13})$$

정리 2 에서 $\mathbf{K}_i$ 가 존재하려면 $(\mathbf{C}_i, \mathbf{U}_i \mathbf{A})$ 가 가관측 조건식 식 (4.21)을 만족해야 하므로, 결국, 미지입력 관측기를 설계하기 위한 가관측 조건식 식 (4.21)은 식 (A14)와 등가이다.

$$\text{rank} \begin{bmatrix} \mathbf{sI}_n - \mathbf{A} & \mathbf{D}_i \\ \mathbf{C}_i & 0 \end{bmatrix} = n + q_i, \quad \forall s \in \mathbb{C}, \quad \text{Re}(s) \geq 0 \quad (\text{A14})$$

식 (A13)과 식 (A14)으로부터 식 (4.21)이 성립함을 알 수 있다. 식 (A14)은 식 (4.5)의 Σ_{DS} 의 불변영점이 안정이어야 한다는 의미이다.



증명 4]

차량 로봇 시스템의 일반화 속도 벡터 $\dot{\mathbf{q}}$ 와 목표 속도 벡터 $\dot{\mathbf{q}}_r$ 는 식 (2.19) 으로부터 식 (A15)와 같이 표현된다. ΔX , ΔY , $\Delta \theta$ 는 차량 로봇의 추적 오차이며 식 (A16)와 같이 정의하며, $\mathbf{q}$ 는 일반화 좌표 벡터이고 $\mathbf{q}_r$ 는 목표 좌표 벡터이다.

$$\begin{cases} \dot{X} = v_x \cos \theta \\ \dot{Y} = v_x \sin \theta \\ \dot{X}_r = v_r \cos \theta_r \\ \dot{Y}_r = v_r \sin \theta_r \end{cases} \quad (\text{A15})$$

$$\begin{cases} \Delta X = X - X_r \\ \Delta Y = Y - Y_r \\ \Delta \theta = \theta - \theta_r \end{cases} \quad (\text{A16})$$

식 (A15)와 식 (A16)으로부터 식 (A17)과 같이 표현할 수 있다.

$$\begin{cases} \Delta \dot{X} = \dot{X} - \dot{X}_r = v_x \cos \theta - v_r \cos \theta_r \\ \Delta \dot{Y} = \dot{Y} - \dot{Y}_r = v_x \sin \theta - v_r \sin \theta_r \\ \Delta \dot{\theta} = \dot{\theta} - \dot{\theta}_r = \omega - \omega_r \end{cases} \quad (\text{A17})$$

식 (A15)의 세 번째 식은 식 (A18)과 같이 표현된다. 그리고 식 (A18)을 식 (A17)에 대입하면 식 (A19)와 식 (A20)을 얻을 수 있다.

$$\theta = \theta_r + \Delta \theta \quad (\text{A18})$$

$$\begin{aligned} \Delta \dot{X} &= v_x \cos(\theta_r + \Delta \theta) - v_r \cos \theta_r \\ &= v_x (\cos \theta_r \cos \Delta \theta - \sin \theta_r \sin \Delta \theta) - v_r \cos \theta_r \\ &= (v_x \cos \Delta \theta - v_r) \cos \theta_r - v_x \sin \theta_r \sin \Delta \theta \end{aligned} \quad (\text{A19})$$

$$\begin{aligned} \Delta \dot{Y} &= v_x \sin(\theta_r + \Delta \theta) - v_r \sin \theta_r \\ &= v_x (\sin \theta_r \cos \Delta \theta + \cos \theta_r \sin \Delta \theta) - v_r \sin \theta_r \\ &= (v_x \cos \Delta \theta - v_r) \sin \theta_r + v_x \cos \theta_r \sin \Delta \theta \end{aligned} \quad (\text{A20})$$

$\Delta\theta$ 가 매우 작을 경우 식 (A19)와 식 (A20)을 선형화하면 식 (A21)과 같이 표현된다.

$$\begin{cases} \Delta\dot{X} = (v_x - v_r)\cos\theta_r - v_x \sin\theta_r \Delta\theta \\ \Delta\dot{Y} = (v_x - v_r)\sin\theta_r + v_x \cos\theta_r \Delta\theta \\ \Delta\dot{\theta} = \dot{\theta} - \dot{\theta}_r = \omega - \omega_r \end{cases} \quad (\text{A21})$$

식 (A21)을 행렬 형태로 표현하면 식 (A22)와 같다.

$$\begin{bmatrix} \Delta\dot{X} \\ \Delta\dot{Y} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_x \sin\theta_r \\ 0 & 0 & v_x \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x - v_r \\ \omega - \omega_r \end{bmatrix} \quad (\text{A22})$$

X, Y, θ 가 매우 작을 경우 $\Delta X \rightarrow X, \Delta Y \rightarrow Y, \Delta\theta \rightarrow \theta$ 이 되고, 선형화를 위한 평형점(Equilibrium point) $v_r = \text{const} \approx v_x, \theta_r = \text{const} \approx \theta$ 을 이용하면, 식 (A22)을 식 (A23)과 같이 표현할 수 있다.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r \sin\theta_r \\ 0 & 0 & v_r \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x - v_r \\ \omega - \omega_r \end{bmatrix} \quad (\text{A23})$$

증명 5]

전체 시스템의 상태 벡터 $\mathbf{S}(\mathbf{t})$ 와 대형 수준 함수(Formation level function)인 수평 대형 함수(Horizontal formation function) h_1 와 수직 대형 함수(Vertical formation function) h_2 는 식 (4.32)와 같이 정의되므로 $\dot{\mathbf{S}}(\mathbf{t})$ 는 식 (A24)와 같다.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \dot{X}_1 - \dot{X}_2 + \alpha(\omega_1 - \omega_2) \\ \dot{Y}_1 + \dot{Y}_2 + \beta(\omega_1 + \omega_2) \end{bmatrix} \quad (\text{A24})$$

식 (A23)의 $\theta_r = 0$ 이라 하면, $\dot{h}_1$ 와 $\dot{h}_2$ 는 식 (A25)와 같이 표현된다.

$$\begin{aligned} \dot{h}_1 &= \dot{X}_1 - \dot{X}_2 + \alpha(\omega_1 - \omega_2) \\ &= -v_r \sin \theta_r (\theta_1 - \theta_r) + \cos \theta_r (v_{1x} - v_r) \\ &\quad - (-v_r \sin \theta_r (\theta_2 - \theta_r) + \cos \theta_r (v_{2x} - v_r)) + \alpha(\omega_1 - \omega_2) \\ &= v_{1x} - v_{2x} + \alpha(\omega_1 - \omega_2) \\ \dot{h}_2 &= \dot{Y}_1 + \dot{Y}_2 + \beta(\omega_1 + \omega_2) \\ &= v_r \cos \theta_r (\theta_1 - \theta_r) + \sin \theta_r (v_{1x} - v_r) \\ &\quad + (v_r \cos \theta_r (\theta_2 - \theta_r) + \sin \theta_r (v_{2x} - v_r)) + \beta(\omega_1 + \omega_2) \\ &= v_r \theta_1 + v_r \theta_2 + \beta(\omega_1 + \omega_2) \end{aligned} \quad (\text{A25})$$

식 (4.31)의 $\theta_r = 0$, $\omega_r = 0$ 이라고 하면, 식 (A25)를 이용하여 식 (A24)를 식 (4.34)의 $\dot{\mathbf{S}}(\mathbf{t}) = \mathbf{A}\mathbf{S}(\mathbf{t}) + \mathbf{B}_1\mathbf{u}_1(\mathbf{t}) + \mathbf{B}_2\mathbf{u}_2(\mathbf{t})$ 으로 표현하면 식 (A25)와 식 (A26)과 같다.

$$\dot{\mathbf{S}}(\mathbf{t}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ v_r & v_r & 0 & 0 \end{bmatrix} \mathbf{S}(\mathbf{t}) + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & \alpha \\ 0 & \beta \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -1 & -\alpha \\ 0 & \beta \end{bmatrix} \mathbf{u}_2 \quad (\text{A25})$$

$$\mathbf{u}_1 = \begin{bmatrix} v_{x1} - v_r \\ \omega_1 - \omega_r \end{bmatrix}, \quad \mathbf{u}_2 = \begin{bmatrix} v_{x2} - v_r \\ \omega_2 - \omega_r \end{bmatrix} \quad (\text{A26})$$

◀

부록 B 정의

정의 1] 일반화된 역행렬

$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$ 와 $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$ 가 성립하면 $\mathbf{A}^+$ 을 $\mathbf{A}$ 의 일반화된 역행렬이라 한다.

다음 식이 만족하면 $\mathbf{A}^+$ 는 $\mathbf{A}$ 의 의사역행렬(Pseudoinverse matrix)이라 하며 일반화된 역행렬을 성립한다. * 는 공액전치복소수(Complex conjugate transpose)이다.

- 1) $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$
- 2) $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$
- 3) $(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}$, $\mathbf{A}^+\mathbf{A} = \textit{Hermitian}$ 행렬
- 4) $(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+$, $\mathbf{A}\mathbf{A}^+ = \textit{Hermitian}$ 행렬

예) $\mathbf{A}^+ = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^* : \exists (\mathbf{A}^*\mathbf{A})^{-1}$, $\mathbf{A}^+ = \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^{-1} : \exists (\mathbf{A}\mathbf{A}^*)^{-1}$



부록 C 하드웨어

C.1 초음파 위치 시스템

본 연구에 적용된 초음파 위치 시스템은 여러 대의 발신기를 고정된 위치에 배치하고 거기서부터 나오는 초음파 신호를 수신기가 수신하여 각각의 초음파 발신기 간의 거리를 측정하고 이를 이용하여 수신기의 위치를 계산하도록 설계되어 있다. 이는 GPS 의 기본 원리와 유사하나 GPS 는 위성에서 자신의 출발 신호와 위치 정보를 전송하지만 본 연구에서 사용한 초음파 위치 시스템은 RF 동기신호로 초음파의 발신 시점을 찾고 발신기는 알고 있는 위치에 고정되어 있다. 또한 여러 대의 수신기가 존재한다고 하더라도 각각의 수신기가 서로 간섭하지 않고 위치를 독립적으로 구할 수 있기 때문에 여러 물체의 위치를 각각 확인할 수 있다.

수신기는 RF 동기신호 수신 시점을 기준으로 발신기에서 발사한 초음파의 TOF 를 계산하여 차례대로 4 대의 발신기와 수신기 간의 거리를 계산한다. 이렇게 일정한 주기로 초음파를 송신하는 이유는 각각의 위성들이 초음파를 송신할 때 서로의 간섭을 없애고 먼저 발사된 초음파가 나중에 반사되어 수신기로 들어와서 잘못된 연산을 수행하지 않도록 하기 위함이다. 일반적으로 GPS 의 샘플링 주기는 1Hz 인데 반해서 초음파 수신기의 샘플링 주기는 이보다 짧은 2.5Hz 이다. 이는 초음파 의사위성이 GPS 보다 좀 더 빠르게 절대 위치를 알 수 있게 하며, 사용되는 환경에 따라서 샘플링 주기의 조절이 가능하므로 좀더 유연하고 빠른 위치 정보를 측정 가능하게 한다.

본 연구에 적용된 초음파 위치 시스템의 사양은 Table C.1 과 같다.

Table C.1 Specification of ultrasonic positioning system

Response Time	0.6sec = 0.15sec * 4개
Resolution	Display unit - 1mm Repeatability - 10~100mm (static)
Cover Range	권장 - 7m * 7m 수신기 천정 지향조건 수신기, 천정과의 높이 차 2m 이상 권장
Power Supply	발신기 - 9~12VDC 수신기 - 9.0VDC
Ultrasonic	25kHz
RF	315MHz (ASK type)
Output	RS232 serial output (115200bps, n, 8, 1)

C.2 이동 차량

본 연구에서 사용된 차량 로봇은 크게 이동 차량과 임베디드 컨트롤러 부분으로 이루어져 있다. 이동 차량은 마이크로프로세서를 이용한 구동체 장비로 컨트롤 디바이스로 ATmega128L 을 사용하여 장비를 제어하고 있으며 ATmega128L 의 사양은 다음과 같다.

- 고성능 저전력 AVR 8bit Micro controller
 - 향상된 RISC 아키텍처(16MIPS @ 16MHz)
 - 133 중 명령세트(in 1cycle)
 - 32 x 8bit 범용 레지스터 + 주변 컨트롤 레지스터/Multiplier(in 2 cycle)
 - 16MHz 에서 16MIPS progress
- 비 휘발성 프로그램과 데이터 메모리
 - 128K Byte 내부 프로그램 가능한 ISP Flash memory
 - 4K Byte EEPROM(10 만번) for Data
 - 4K Byte internal SRAM
 - ISP 를 위한 SPI 인터페이스
- JTAG 인터페이스
 - 내장 메모리의 Programming 과 On-Chip Debug 를 위한 JTAG (IEEE Standard 1149.1) 지원
- 주변 장치 특징
 - 분리된 오실레이터에 의한 Real Time Count
 - 8 채널, 10bit ADC
 - 두 개의 시리얼 UART
 - Master/Slave SPI 시리얼 인터페이스
 - 아날로그 비교기
- 마이크로컨트롤의 특징
 - Power-on Reset, 안정된 전원공급을 위한 Programmable Brown-out Detection
 - 내부 RC 오실레이터
 - 외부와 내부의 인터럽트 소스
 - 6 개의 sleep 모드
 - 소프트웨어적 선택 가능한 클럭 주파수
 - 전체 풀업 Disable
- 2.7 ~ 5.5V Operating Voltages (ATmega128L)
 - 0 ~ 8 MHz Speed Grades

이동 차량은 구동을 위해 DC Geared 모터를 사용하고 있으며, 4 개의 바퀴에 모터가 하나씩 장착되어 있다. 오른쪽 2 개의 모터가 하나의 신호로 제어되어 동시에 구동이 되고, 왼쪽 또한 마찬가지로 구동한다. 4 개 모터 중에서 뒤쪽 2 개의 모터에는 엔코더가 내장된 모터를 사용하고 있으며 모터 사양은 Table C.2 와 같다.

Table C.2 Specification of DC geared motor(RB-35GM series)

Rated volt (V)	12
Rated torque (kg-cm)	2.0
Rated speed (rpm)	102
Rated current (mA)	<470
No load current (mA)	<120
Rated output (W)	3.14

본 연구에서는 사용되지 않았지만 이동 차량에는 구동 상태를 파악하기 위한 다양한 센서가 장착되어 있다. 그 센서를 소개하면 Table C.3 과 같다.

Table C.3 Specification of sensors in mobile vehicle

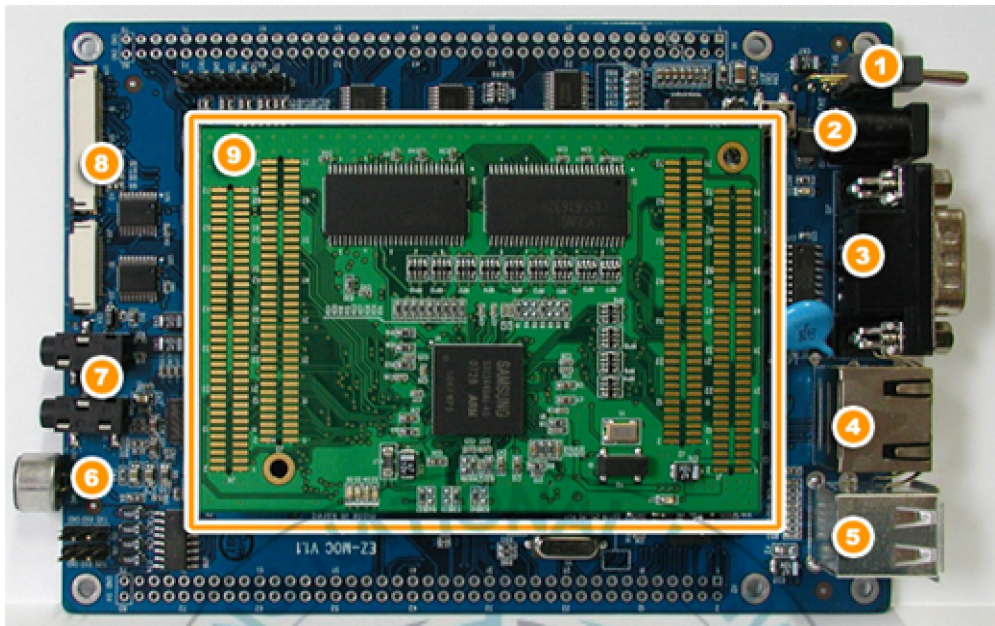
Ultrasonic Sensor (KT-400 series)	Center frequency : 40.0KHz $\pm$ 1.0KHz SPL : 115dB(0dB re 0.0002u bar) Driving voltage : 1 ~ 20V Bandwidth : 2KHz Allowable input power : 0.2W
Distance Measuring Sensors (GP2Y0A21YK)	Distance measuring range : 10 to 80cm Output terminal voltage : 0.4 ~ 2.3V Average Dissipation current : 30mA
Acceleration sensor (ADXL202E)	2-Axis acceleration sensor on a single IC chip Low power < 0.6mA 3V to 5.25V single supply operation Duty cycle Output
Infrared sensor (ST-8L, EL-8L)	Collector Current : 20mA Half angle : $\pm$ 15 deg

C.3 리눅스 기반 임베디드 컨트롤러

본 연구에 적용된 EZ-S3C2440 임베디드 컨트롤러는 삼성의 S3C2449A-400 을 탑재한 보드로서 커널 2.6 을 채택하여 리눅스 기반으로 동작한다. 3 개의 시리얼포트를 가지며 이더넷 통신환경이 구축되어 있으며, 하드웨어를 디버깅 할 수 있는 JTAG 포트가 내장되어 있다. EZ-S3C2440 임베디드 컨트롤러의 사양은 Table C.4 에 표시된 바와 같으며, Fig. C-1 은 실물 사진이다.

Table C.4 Specification of embedded controller based on Linux

Classification	Detailed list
Size	100mm*140mm
MCU	400MHz S3C2440A ARM RISC chip (ARM 920T)
RAM	64 Mbyte SDRAM
ROM2	64 Mbyte NAND-Flash
LED	Debugging 4 bits
UART	Console 1Port, RS232 2prot
Ethernet	10/100 Base-T
USB	USB Host 2 port
Sound	AC'97(Line-out, Head-phone)
SD/MMC	SD/MMC card socket
TFT-LCD I/F	Interface max 640*480
Touch I/F	Interface four-wire system
CMI I/F	Camera module
Internal connector	288-pins board to board connector
Extension connector	160-pins board to board connector
JTAG	On-board JTAG convertor
Power	DC 5V/1A



1. Power switch
2. Adapter jack
3. Console (serial) port
4. LAN port
5. USB connector
6. Microphone
7. Sound jack (Line-out, Head-phone)
8. LCD connector
9. MCU, RAM and ROM2

Fig. C-1 EZ-S3C2440 embedded controller

감사의 말씀

본 논문이 완성되기까지 아낌없는 지도와 격려로서 학문의 바른 길을 이끌어 주신 김상봉 지도교수님께 무한한 감사의 말씀을 드리오며, 교수님께서 베풀어주신 배움에 대한 즐거움과 따뜻한 격려의 말씀에 많은 힘을 얻었기에 다시 한번 깊은 감사의 말씀을 드립니다. 그리고 항상 저희를 생각해 주시고 아껴주신 사모님께도 감사의 말씀을 드립니다.

논문 심사 과정에서 따뜻한 충고와 조언으로 끝까지 세심하게 다듬어 주신 정영석 교수님과 손정현 교수님께 깊은 감사의 말씀을 드립니다. 항상 옆에서 가르침을 주시고 따끔한 질책과 격려를 아끼지 않으셨던 김학경 교수님께도 깊은 감사의 말씀을 드립니다.

석사 과정과 본 논문이 진행되는 동안 아낌없는 지원과 수많은 격려의 말씀을 해주신 해양연구원 해양시스템연구부의 홍섭 박사님, 전봉환 박사님, 김형우 박사님, 최종수 박사님, 여태경 박사님, 박성재 연구원님, 김성준님, 조재철님께 깊은 감사의 말씀을 드리며, 심해저 집광기의 성공을 진심으로 기원합니다.

CIMEC Lab.에서 같이 생활하고 가르침 주신 김동규 선배님, 신승목 선배님, 김영규 선배님, 임재성 선배님, 김상찬 선배님, 박바다 선배님, 마성진 선배님, 김병용 선배님께 감사 드립니다. 학부 시절부터 석사 과정까지 같이 동고동락한 이희숙님, 정준호님, 최낙순님, 이도경님, 김대원님, 이경목님, 박철한님, 박유미님, 최하연님께 감사의 마음을 전합니다. 그리고 타지에서 고생하면서 공부하면서도 많은 도움을 주신 Lam, Dung, Duy, Kang, Phuong, Hung 님께도 감사 드립니다.

초등학교부터 지금까지 제 곁에서 함께하면서 많은 힘이 되어준 모든 친구들에게 감사의 말씀을 전하며, 저를 알고 있는 모든 지인 분들에게도 감사의 말씀을 드립니다.

마지막으로 항상 저를 믿어주시고 응원해주시며 제가 가장 믿고 사랑하는 아버지, 어머니께 머리 숙여 진심으로 감사의 말씀을 드리며, 나의 사랑스러운 동생에게도 감사의 마음을 전합니다.

2009 년 02 월 윤석민