



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

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

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



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

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

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

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

[Disclaimer](#)

碩士學位論文

지능로봇을 위한  
실시간 장애물 회피에 관한 연구



濟州大學校 大學院

機械工學科

趙 守 定

2006 年 12 月

지능로봇을 위한  
실시간 장애물 회피에 관한 연구

指導教授 崔 劉 賢

趙 守 定

이 論文을 工學 碩士學位 論文으로 提出함

2006 年 12 月

趙守定の 工學 碩士學位 論文을 認准함

審査委員長 \_\_\_\_\_ 印

委 員 \_\_\_\_\_ 印

委 員 \_\_\_\_\_ 印

濟州大學校 大學院

2006 年 12 月

# A Study on the Real-time Obstacle Avoidance for Intelligent Robots

Su-Jeong Cho

(Supervised by professor Kyung-Hyun Choi)

A thesis submitted in partial fulfillment of the requirement  
for the degree of Master of Engineering

2006. 12.

Department of Mechatronics Engineering  
GRADUATE SCHOOL  
CHEJU NATIONAL UNIVERSITY

# 목 차

<b>I. 서 론</b> .....	1
1. 연구배경 .....	1
2. 연구 동향 .....	2
3. 연구목표 .....	6
<b>II. 하이브리드 혼합제어구조</b> .....	7
1. 혼합 제어구조 .....	7
2. Reactive layer .....	9
2.1 리소스 .....	10
2.2 장애물 회피 모듈 .....	11
2.3 단위행위 .....	12
2.4 행위조정기 .....	12
2.5 동작제어기 .....	13
<b>III. 모바일-머니플레이터의 기구학 해석</b> .....	14
1. 모바일-머니플레이터 로봇의 구조 .....	14
2. 모바일로봇의 기구학 해석 .....	15
3. 머니플레이터의 기구학 해석 .....	18
3.1 순기구학 해석 .....	18
3.2 역기구학 해석 .....	20
3.3 자코비안 행렬 .....	23
4. 모바일-머니플레이터로봇의 기구학 해석 .....	25
<b>IV. 충돌회피 알고리즘</b> .....	27
1. Elastic Strips .....	27

1.1 내부 힘 .....	27
1.2 외부 힘 .....	28
2. 모바일-머니플레이터로봇의 장애물 회피 .....	30
2.1 장애물 회피 방법 .....	30
2.2 로봇 자세 .....	31
2.3 제어 점 .....	32
3. 내부 힘의 계산 .....	33
4. 외부 힘의 계산 .....	35
5. Elastic Force에 의한 궤적변형 .....	38
<b>V. 시뮬레이션 .....</b>	<b>41</b>
1. 모바일부 시뮬레이션 .....	41
2. 3D Simulation .....	51
<b>VI. 결론 .....</b>	<b>53</b>
<b>VII. 참고문헌 .....</b>	<b>55</b>

## List of Figures

- Fig. 1 BERRA architecture
- Fig. 2 Hybrid control architecture of care-O-bot
- Fig. 3 Control for autonomous robot
- Fig. 4 Proposed hybrid control architecture
- Fig. 5 Overview of integrated navigation system
- Fig. 6 Action coordinator
- Fig. 7 Robot configuration
- Fig. 8 Mobile robot modeling and coordinate system
- Fig. 9 Manipulator modeling and coordinator
- Fig. 10 Joints states
- Fig. 11 Transformation of the  $\theta_3$  value
- Fig. 12 Joint angle after wrist
- Fig. 13 Coordinate of Mobile Manipulator
- Fig. 14 Operation of internal force
- Fig. 15 Potential Field Method
- Fig. 16 Effect of the external force
- Fig. 17 Collision avoidance procedure
- Fig. 18 Trajectory of a mobile manipulator.
- Fig. 19  ${}^jP_i$  represents the control points on the body.
- Fig. 20 Transformation procedure of the trajectory
- Fig. 21 A safety area of control point
- Fig. 22 Compute the position of obstacle
- Fig. 23 Compute the total force
- Fig. 24 Simulation as change of  $k_c$  in the limit of  $k_r < 0.5$

Fig. 25 Simulation as change of  $k_e$  in  $k_r=0.5$  case

Fig. 26 Simulation as change of  $k_e$  in the limit of  $k_r>0.5$

Fig. 27 Simulation as change of  $k_e$  in the limit of  $k_r>0.5$

Fig. 28 Simulation as change of  $k_e$  in  $k_r=0.5$  case

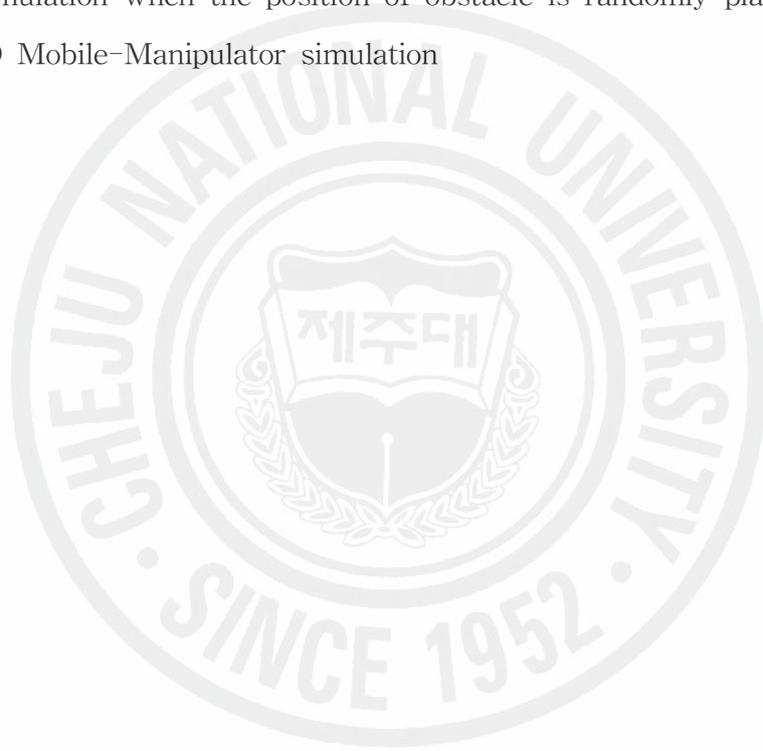
Fig. 29 Simulation as change of  $k_e$  in the limit of  $k_r<0.5$

Fig. 30 Simulation as change of  $k_e$  in  $k_r=0.5$  case when the  $d_{safe}=1000$

Fig. 31 Simulation as change of  $k_e$  in  $k_r=0.5$  case when the  $d_{safe}=100$

Fig. 32 Simulation when the position of obstacle is randomly placed

Fig. 33 3D Mobile-Manipulator simulation



## Summary

The ultimate object of a mobile-manipulator robot in indoor environment is to navigate safely and accomplishes its mission. The robot that assists and helps people has to cope with the rapid change that cannot be predicted in uncertain circumstances and carry out its work.

This paper proposes the Elastic Force application on the obstacle avoidance of Intelligent Robots. The method deals with the problem associated with a Intelligent robot driving to a goal configuration as avoiding obstacles. The initial trajectory of a robot is determined by a motion planner, and the trajectory modification is accomplished by adjusting the control points. The control points are obtained based on the elastic force approach. Consequently the trajectory of a robot is incrementally modified to maintain a smooth and adaptive trajectory in an environment with obstacles. The suggested algorithm drives the robot to obstacle avoid in real-time. Finally, The simulation studies are carried out to illustrate the effectiveness of the proposed approach.

# I. 서 론

## 1. 연구 배경

현재의 로봇은 사용 범위가 다양해지면서 로봇을 이용한 노인 및 장애인을 대신하는 실버산업의 주역으로서의 역할이 기대되어지고 있다. 이러한 인간의 보조적인 작업을 수행하기 위해서는 주행기술 이외에도 지능적인 능력바탕으로 로봇으로 하여금 인간의 행동을 대신하고 더 나아가 인간의 판단이나 지시에 의존하지 않는 자율적 이동 및 다양한 작업수행 능력을 갖추어야 한다.

지능로봇 시스템이 원하는 기능을 얻기 위해서는 현재 자기 위치와 주행목표 위치를 알 수 있는 자기위치인식(localization)기능 과 작업공간에 대한 정보를 스스로 인식 할 수 있는 환경인식(map building)기능, 원하는 목적지를 가기 위한 경로 생성(path planning)기능이 필요하며 추가적으로 동적이며 불확실한 환경에서 효과적으로 안정된 빠른 주행을 위한 정, 동적인 장애물 회피(obstacle avoidance)기능이 필수적이다.

효과적인 장애물 회피기능을 위해서는 최대한 단순하면서 계산이 빠른 알고리즘 개발이 이루어져야 한다. 현재 모바일-머니플레이터 로봇의 경우 장애물 회피기술이 많이 있지만 대부분이 6축 이하의 자유도를 갖고 있는 머니플레이터라는 점과 모바일위에 머니플레이터가 부착된 일반적인 모바일-머니플레이터로봇 형태를 가지고 있다. 따라서 인간의 팔과 닮은 보조 역할 수행 가능한 로봇 시스템에 맞는 알고리즘 개발이 필요하다. 뿐만 아니라 개별 기술들이 효율적으로 연동되기 위해서는 이를 효율적으로 관리하기 위해 하드웨어와 소프트웨어 모듈의 원활한 시스템 통합을 위한 새로운 개념의 제어 구조의 도입이 필요하다.

## 2. 연구 동향

로봇의 행동은 주변의 환경변화를 고려하지 하느냐 않느냐의 관점에서 반응 제어(reactive control)와 인정 제어(deliberative)로 나뉜다. 반응제어는 지역적 주행(local navigation)이라고도 표현하는데 외부환경의 인식 없이 센서의 입력 정보만으로 구현되는 것을 말한다.

반응제어 내에서 많이 사용되는 대표적인 방법으로 다음과 같다.

장애물 표현을 위한 확실성 격자(certainty grid)방법을 사용한 Morayec.H.P.(1985)은 장애물의 존재 가능성을 Certainty Value(CV)값으로서 나타내었다. Khatib.O.(1985)는 장애물 접근 시 공간상의 물체 사이에 가상의 힘이 작용하여 그 힘에 의해 운동하도록 하는 가상 포텐셜 필드(artificial potential field: APF) 방법을 제안하였으며, 모바일 로봇의 실시간 동작 계획을 위하여 인공 전위계 개념에 장애물을 표현하기 위한 확실성 격자 개념을 합쳐서 가상 역장(virtual force field: VFF)(Borenstein and Koren, 1991)을 도입한 방법이 제안되었고, 이것을 발전시킨 벡터장 히스토그램(vector field histogram: VFH)방법이 제안되었다[1].

이 같은 방식의 경우 로봇이 장애물과 충돌하지 않는 경로를 찾아내는 데는 효율적이었으나, 많은 경우 로봇의 동력학적인 제한들을 고려하지 않아서 실제 로봇이 이동할 때 좁은 통로를 빠져나가지 못한다든지 로봇의 실제 이동 궤적에 진동이 발생하여 왔다. 이 방법들은 효율적으로 충돌 회피를 위한 운동 방향을 구해내지만 로봇의 동력학적인 제한 조건을 고려하지 않은 것이 대부분이었고, 동력학적인 제한 조건을 고려하기에는 부적합한 방법들이었다. 이중 벡터장히스토그램 방법은 좁은 통로에서도 부드럽고 성공적으로 로봇을 이동하게 하지만 로봇의 동력학적 제한 조건을 고려하지 못하여 장애물이 많이 흩어져 있는 경우 로봇의 이동에 문제를 발생시켰다.

1997년에는 Khatib.O.의Elastic Strip방법이 제안되었다. 이 방법의 경우 로봇의 궤적(trajjectory)을 탄성물질이라 가정하여 장애물을 안전하게 회피하도

록 하였다. 장애물 접근 시 물체 사이에 가상 힘이 작용한다는 것에서는 포텐셜 필드 방법과 유사하나 목표지점까지의 도달 동안 척력 외에도 궤적이 수축하려는 힘인 탄성 힘이 작용한다는 것에서 보다 최적화된 주행기술을 보인다 [19]. 또한 로봇 중심으로 충돌 회피 영역을 지정하여 장애물에 대한 안전한 최소 회피 영역을 정하여 보다 안정된 주행을 구사 하지만 이 방법 역시 지역 극소점에 빠질 수 있는 단점을 가지고 있다.

현재 지능로봇의 제어구조에 관한 연구동향은 서비스로봇의 필요성을 인식하고 로봇의 다양한 응용을 위하여 실시간 제어 구조에 관한 연구가 본격화하고 있다.

80년대 중반, Brooks[7]는 동물의 반응 행동을 모델링한 행위기반(behavior-based)방식의 포섭구조(subsumption architecture)를 제안하였다. 이 구조는 기능별로 로봇시스템 구성요소를 분류한 기존의 제어방식과 달리 능력별로 분류하였다. 그러나 이 포섭구조의 단점은 계층이 점점 많아질수록 이를 효율적으로 처리할 수 있는 메커니즘을 제시하지 못했다는 점과 주변 환경에 대한 오류 진단 및 복구가 어렵다는 것이었다.

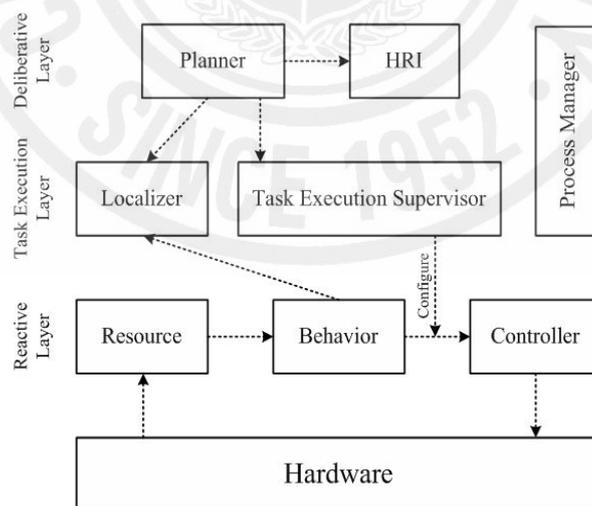


Fig. 1 BERRA architecture

이에 대한 해결책으로 90년대 초반에 인정 층(deliberative layer)과 반응 층(reactive layer)을 동시에 가진 deliberative/reactive 혼합제어구조가 제시되었다[6, 7, 8, 16]. 이 구조는 주어진 임무를 해석하여 적절한 실행계획을 작성하고, 센서부와 구동부를 밀접하게 구성하여 실시간 반응행동이 가능하게 되어 있다. 국내외 연구를 살펴보면, 스웨덴의 왕립기술원에서는 서비스 로봇의 제어를 위해 Fig. 1 과 같이 Deliberate, TaskExecution, Reactive 의 3 개 층으로 구성된 제어구로서 BEERA(Behavior-based Robot Research Architecture)를 제안하였다[20]. 이 구조는 재사용성(reusability)과 유연성(flexibility)을 목표로 설계되었다. 독일의 Fraunhofer IPA 에서는 6 자유도 머니플래이터와 자율주행부가 결합된 서비스로봇인 Care-O-bot 에 Fig. 2 와 같은 하이브리드 제어구조를 제안하였다[10].

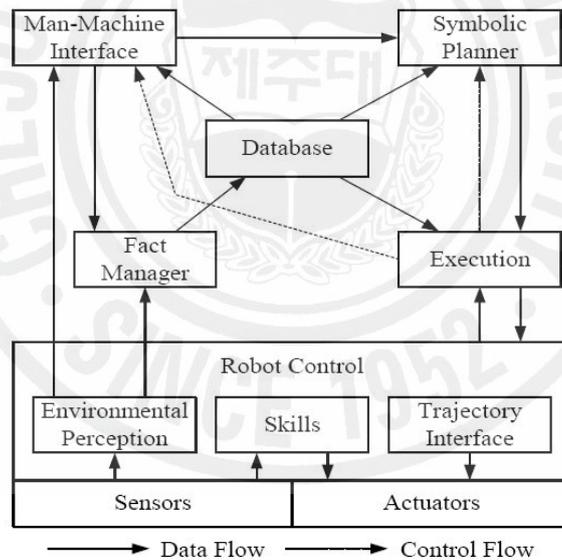


Fig. 2 Hybrid control architecture of care-O-bot

국내의 대학과 한국과학기술원(KIST)이 공동으로 Fig. 3 과 같이 reactive 제어방식을 기반으로 하고 deliberative제어의 계획기능을 일부 채용한 하이브리드 제어구조를 설계하였다[3]. 이 구조는 분산처리 개념을 도입 하여 동작의

수행시간 및 계산 량을 줄일 수 있다.

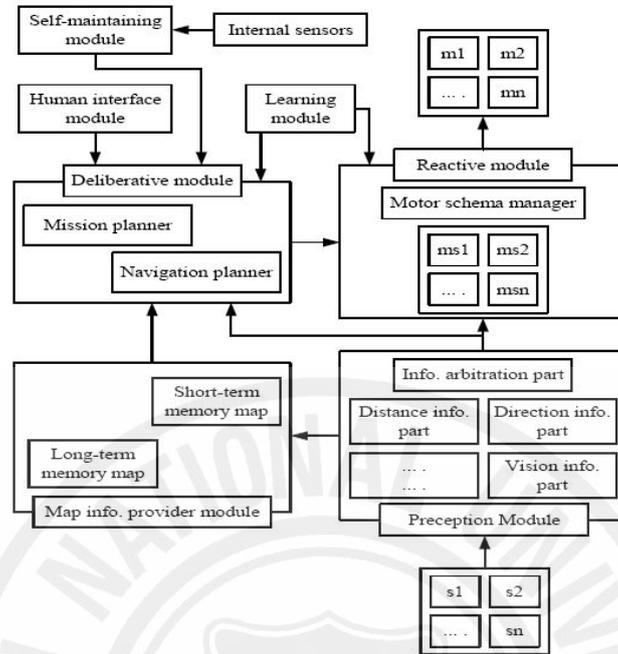


Fig. 3 Control for autonomous robot

그러나 선행연구들을 살펴볼 때 deliberative/reactive 혼합제어구조인 BERRA의 reactive층에 있는 각각의 요소들은 객체지향 접근법으로 모델링되어 재사용성과 유연성을 고려한 측면에서는 우수하나, 실시간성에 대한 고려가 미흡한 실정이다. 따라서 불확실하고 동적인 환경에서 로봇이 다양하고 복잡한 기능을 효율적으로 수행하기 위해서는 reactive층에 대한 제어구조의 개선이 필요하다.

### 3. 연구목적

지능로봇이 가져야 할 기본적인 기능들은 주어진 환경에 대한 인식을 통하여 스스로 판단할 수 있는 기능, 이동해야 할 경로 및 궤적들을 실시간으로 생성하는 기능이다. 이를 위해서는 복잡한 환경에서 다양한 기능을 효율적으로 제어하는 반응 층(reactive layer)의 제어구조를 설계하고 실시간으로 회피하기 위한 장애물회피 알고리즘 개발이 이루어져야 한다.

본 논문에서는 모바일-머니플레이터 형태의 지능로봇을 통합 제어하는 제어구조로부터 반응 층(reactive layer)의 구조를 선정하고 그 구성요소인 컴포넌트를 설계한다. 또한 지능로봇의 기구학 특성을 분석하며 로봇이 주행동안 만나게 되는 장애물에 대한 충돌회피 알고리즘에 대해서 논한다.

## II. 하이브리드 혼합제어구조

### 1. 혼합 제어구조

로봇은 자율적인 실행계획의 작성과 인간과의 상호관계를 유지하며, 이와 동시에 기대하지 않은 환경이나 불확실한 환경에 대해 즉각적인 반응행동을 할 수 있어야 한다. Fig. 4는 3개의 계층으로 이루어진 혼합제어구조(hybrid deliberative/reactive architecture)를 나타낸다. 각 계층은 인정 층(deliberative layer), 결순 층(sequencing layer), 반응 층(reactive layer)으로 구성되어 있다. 인정 층과 결순 층은 Linux의 사용자 영역을 사용하고 반응 층은 실시간 제어를 위해서 실시간(real-time)을 지원하는 RTAI의 커널영역을 사용한다.

인정 층에서는 인간과의 의사소통을 통해 명령을 이해하고 분석하여 주어진 임무를 반드시 완수 할 뿐만 아니라, 임무의 실패를 지각할 수 있어야 하고, 가능하면 실패를 사전에 미리 막을 수 있어야 한다. 인간의 의사를 이해하고 분석하여 실행 계획작성을 통해 하위계층에 세분화된 태스크(task)를 전달함으로써 인간과 로봇의 실행기 사이에서 교량 역할을 하는 것이다.

결순 층은 인정 층과 반응 층 사이의 연결고리를 제공한다. 센서들로부터 온 정보를 가공하여 장기적이고 고차원적인 정보를 생성하여 인정 층으로부터 받은 태스크에 따라 프로세스를 수행하기 위해 필요한 모바일 관리자(mobile supervisor)와 조작 관리자(arm supervisor), 파지 관리자(hand supervisor)를 구동시킨다. 또한, 다양한 단위행위(action)를 효율적으로 수행하기 위한 프로세서를 맡고 있으며 반응 층을 관리한다.

반응 층은 상위계층에서 내려오는 명령을 실행하고 임의의 환경에서 장애물이 나타났을 때 로봇의 안전을 보장할 뿐만 아니라, 인간에게 미치는 위험요소를 줄이기 위해 실시간으로 대응할 수 있어야 한다. 반응 층에 있는 단위행

위들은 센서와 액추에이터 사이에 단단하게 결합되어 행동한다. 그리고, 센서 부를 효율적으로 관리하는 리소스(resource), 정의된 단위행위들을 적절히 융합하기 위한 행위조정기(action coordinator)와 로봇의 액추에이터를 제어하기 위한 조작 제어기(manipulator controller)와 파지제어기(hand controller), 동작 제어기(motion controller)가 있어야 한다.

반응 층의 실시간성을 확보하기 위해서는 각각의 구성요소들이 정해진 시간 내에 주어진 작업을 완수하는 것이 중요하다. 이를 위해서는 먼저 I/O 디바이스를 위한 실시간 드라이브들의 개발이 필요하다. 특히, 직접적으로 로봇의 제어에 관련된 반응 층은 발생된 센서 정보를 실시간으로 취득하여 관리하는 리소스 구성요소들을 구현해야 하고 주기적으로 제어 루프의 계산 작업을 수행해야 함과 동시에 연산 결과를 이용하여 실시간으로 액추에이터들을 움직여야 한다.

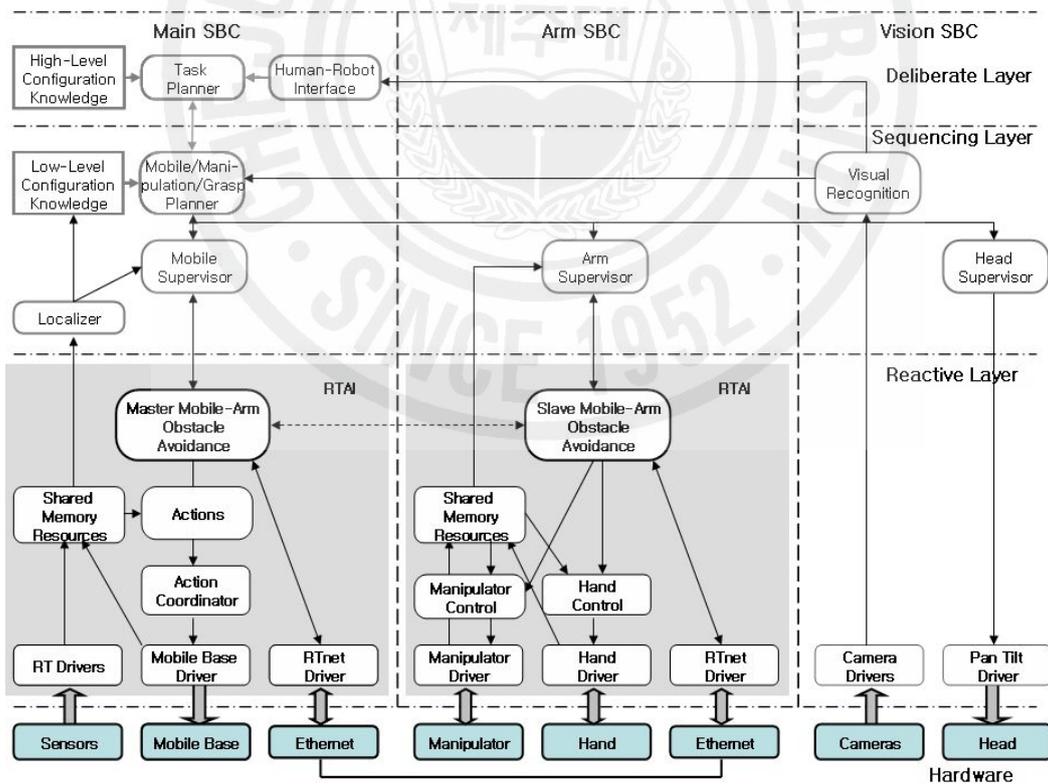


Fig. 4 Proposed hybrid control architecture

## 2. Reactive layer

Fig. 5 는 본 논문에서 개발된 주행 시스템의 개요를 보여주고 있다. 결순 층의 주행 모듈은 많은 시간을 요하는 정교한 알고리즘을 사용하여, 센서 데이터로부터 고차원적인 정보를 생성해낸다. 대표적인 모듈로써 지도(map), 위치추정기(localizer), 경로계획기(path\_planner)가 있다. 결순 층의 자율주행모듈을 살펴보면, 지도작성에는 점유격자지도방법의 알고리즘, 위치추정기는 Monte Carlo Localization 의 알고리즘[9], 경로계획기는 A\* search 알고리즘 [10]이 사용된다. 반응 층은 실시간으로 간단한 계산을 반복 수행하며 로봇의 동작을 제어한다.

반응 층의 구성요소에는 리소스, 단위행위, 행위조정기, 동작제어기, 조작 제어기 및 파지 제어기가 있으며, 장애물 회피 모듈(obstacle avoidance module)이 존재하게 된다. 장애물 회피 모듈은 다시 모바일부와 머니플레이터의 회피를 담당하는 모듈인 주요이동-팔장애물 회피(master mobile-arm obstacle avoidance) 모듈 과 종속이동-팔장애물 회피(slave mobile-arm obstacle avoidance)모듈 로 나뉘며 뒷부분에서 구성요소들의 기능에 대해 상세히 설명되어진다.

결순 층에 있는 자율주행모듈의 위치추정기와 경로 계획기에서 계산된 로봇의 현재 위치 및 로봇이 목표지점에 도착하기 위해서 이동해야 할 경로의 경유점을 계산하여 반응 층의 행위조정기에게 전달하여 동작하게 될 제어모드를 결정하게 된다.

반응 층 내의 구성요소들의 순서는 다음과 같다. 행위조정기에 의해서 활성화된 단위행위는 리소스로부터 정보를 가져와서 로봇이 동작하게 될 속도와 각속도를 계산하여 행위조정기에게 전달한다. 이때 주행동안 장애물을 감지하게 되면 주요이동-팔장애물 회피(master mobile-arm obstacle avoidance)모듈 과 종속이동-팔장애물 회피(slave mobile-arm obstacle avoidance)에서 속도, 각속도를 단위행위에 전달과 동시에 행위조정기에 전달한다. 행위 조정기는

단위 행위들로부터 받은 속도 및 각속도 제어 명령을 융합하여 동작제어기에 전달하며, 융합된 제어 명령을 받은 동작 제어기는 액추에이터를 구동시킨다.

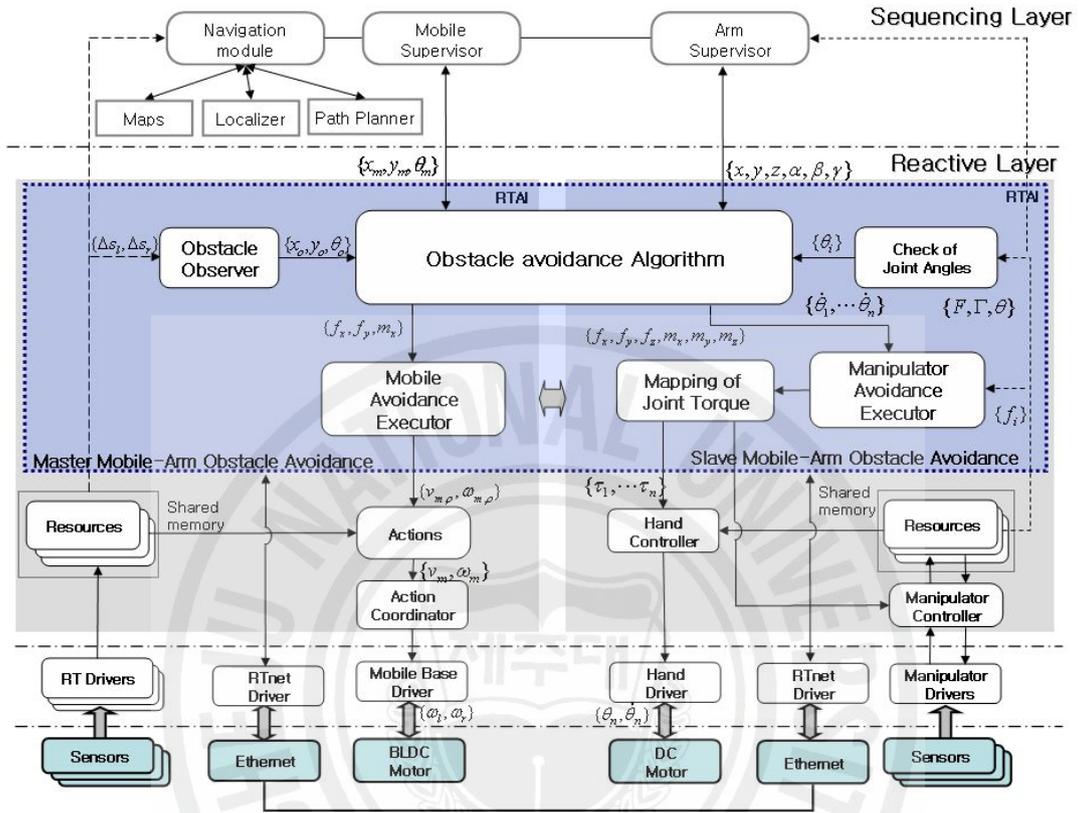


Fig. 5 Overview of integrated navigation system

## 2.1 리소스

리소스는 센서로부터 획득한 데이터를 저장하는 구성요소이다. 각 센서마다 별도의 리소스 구성요소가 존재하며, 저장된 정보는 필요로 하는 여러 단위 행위와 상위 구성요소들에게 제공된다. 리소스는 센서의 데이터 전송방식에 따라 폴링방식(polling type)과 인터럽트방식(interrupt type)으로 분류할 수 있다.

폴링방식의 경우 주기적으로 데이터를 생성하기 위해서는 RTAI 에서 제공하는 주기적인 스레드로 리소스 구성요소를 구현한다. 실시간 스레드는 주기적으로 운영체제에 의해 수행되며 실시간 디바이스 드라이브가 제공하는 함수

를 이용하여 센서로부터 데이터를 획득한 후 이를 공유메모리(shared memory)에 저장한다. Sonar, LRF front, LRF rear, IR front, IR rear, encoder, gyro, battery 의 리소스들이 이 방식에 해당한다.

인터럽트방식은 센서 인터페이스에서 생성되는 인터럽트에 반응하여 센서 데이터를 읽는다. 외부 인터럽트가 발생되었을 때 센서 데이터를 실시간 디바이스드라이브로부터 읽어 공유메모리로 전송한다. Bumper 와 긴급정지 버튼에 이 방식이 사용된다. 제어구조의 실시간성을 확보하기 위해 센서로부터 정확한 시간구간 안에 데이터가 획득되어야 하고, 이 데이터를 인정 층 및 반응 층의 다양한 구성요소들이 효과적으로 사용할 수 있어야 한다[4].

## 2.2 장애물 회피 모듈

주/종속 이동-팔 장애물 회피(Master/Slave Mobile-Arm Obstacle Avoidance)모듈은 반사적, 본능적 회피 모듈이라 할 수 있다. 본 연구에서 제안한 장애물 회피(obstacle Avoidance) 모듈은 센서로부터 측정된 장애물을 감지하여 현재 장애물과 떨어진 거리와 속도, 방향을 Obstacle Observer가 감지하게 되며 현재 위치 추정기(Localizer), 조인트 각 확인(Check of Joint Angles)모듈이 로봇 상태를 확인하게 된다. 이 때 장애물 회피 알고리즘(obstacle avoidance algorithm)이 장애물 회피를 위한 거리 및 회전 방향을 계산하게 되며 실질적으로 로봇의 회피 동작을 위한 병진속도와 회전속도를 Mobile Avoidance Executor에서 생성하여 단위행위로 전달된다. Manipulator 의 경우 Manipulator Avoidance Executor에서 각속도 생성 후 직교 좌표계 값을 각 관절 좌표계의 값으로 변환하는 조인트 토크 매핑(mapping of joint Torques)모듈에 의하여 팔 제어기(hand controller)로 전달된다.

그러나, 로봇이 장애물을 회피하는 동작을 정확히 하기에 앞서, 장애물의 동적 상태를 명확히 알아야 한다. 지역적 주행의 경우, 정지한 장애물은 환경처럼 인식되기 때문에 고려의 대상이 되지 않으나, 움직이는 장애물의 속도와 이동방향을 정확히 예측하여야만 정확한 회피 동작을 수행할 수 있다. 현재 다양한 알고리즘이 연구되고 있으나, 대부분 계산이 복잡하여 시뮬레이션 단

계에 그치고 있거나, 실행되더라도 실시간을 보장할 수 없다. 본 논문에서는 기하학적 표현에 기반한 알고리즘을 제안한다.

### 2.3 단위행위

단위행위는 로봇이 자율적으로 움직이기 위한 최소의 특정 기능을 구현하는 구성요소이다. 단위행위는 여러 리소스 중에서 필요한 몇 개의 정보를 받아서 로봇을 움직이기 위한 속도(translation velocity:  $v$ )와 각속도(angular velocity:  $\omega$ )를 생성한다. 반응 층에 존재하는 단위행위는 결순 층에 있는 자율주행모듈의 위치추정기와 경로계획기에서 계산된 로봇의 현재위치 및 로봇이 목표지점으로 이동시키기 위한 「Goto」와 일반적인 로봇의 위치기반(point-to-point)로 이동하도록 제어하는 「Move」, 로봇의 순간적인 정지를 담당하는 「E-Stop」이 있다. 단위행위는 RTAI 에서 제공하는 함수를 이용하여 실시간 스레드로 구현되며 그 구성은 최대한 단순해야 한다. 단위행위의 기본적인 흐름은 행위조정기에 의해서 수행이 재개되면 연산작업을 위해 필요한 센서의 정보를 읽고 연산 작업을 수행하며 그 결과 정보를 저장한 후, 다시 정지 상태로 들어간다[2, 3].

### 2.4 행위조정기

행위조정기는 특정 기능을 갖는 단위행위의 출력을 효과적으로 융합하여 하나의 출력을 생성하는 구성요소이다. 행위조정기의 특징 중 하나는, 주기적인 수행과 더불어 비주기적인 수행도 가능해야 한다는 것이다.

이것은 행위조정기의 입력에 관여하는 단위행위들이 모두 주기적인 경우도 있지만, 특별한 경우 비주기적인 단위행위도 관여할 수 있기 때문이다. 행위조정기는 주기적 수행과 비주기적인 수행을 포함한 한 쌍의 실시간 스레드로 구현된다. 행위조정기의 역할은 크게 2 가지로 나눌 수 있다. 첫 번째는 액추에이터의 제어 출력을 생성하기 때문에 행위들을 융합하는 방법(command fusion method)과 가중치를 부여하여 선택하는 방법(arbitration method)을 함께 사용하여 새로운 출력을 만들거나 가중치 높은 행위의 출력을 동작제어기

에 전달해 주는 역할이고, 두 번째는 반응 층에서 발생하는 이벤트 등을 감지하여 결순 층의 프로세서 관리자에게 알려주는 역할이다.

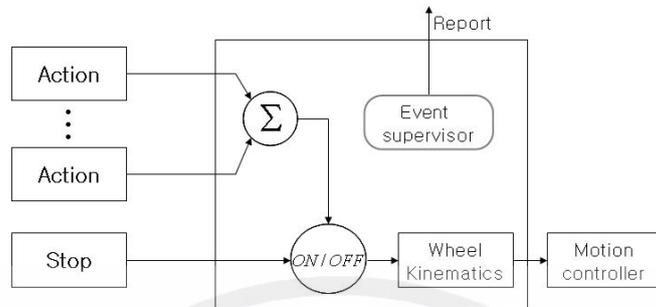


Fig. 6 Action coordinator

예를 들면, 리소스는 각각의 센서들로부터 샘플링시간 내에 데이터 획득 상태를 알려주는 이벤트, 행위들이 주어진 시간 내에 동작여부 및 실패를 알리는 이벤트 등이 있다[2, 3].

## 2.5 동작제어기

동작제어기는 실제로 액추에이터에 제어신호를 전달하는 구성요소이다. 동작제어기에는 행동제어기(motion controller), 조작 제어기(manipulator controller), 파지 제어기(hand controller)가 존재하며, 각 모듈별로 액추에이터가 원하는 형태로 환산한다. 행동제어기의 경우 행위조정기에서 전달되는 양쪽 바퀴속도를 조작 제어기, 파지 제어기는 각각 관절의 상대적 각속도를 액추에이터가 원하는 형태로 환산하여, 사용하는 인터페이스에 맞는 실시간 드라이브를 통해 각 모터에 전달하는 역할을 한다. RTAI 에서 제공하는 함수를 이용하여 실시간 스레드로 구현된다.

### Ⅲ. 모바일 머니플레이터의 기구학 해석

#### 1. 모바일 머니플레이터 로봇의 구조

본 논문에서 사용된 로봇은 Fig.7 에서와 같이 21C 프론티어 연구개발 사업의 일환인, 인간기능 생활지원 지능로봇 사업단에서 노인복지용 로봇을 개발하기 위한 목적으로 설계하여 (주)다사테크에서 제작한 이륜구동 방식으로 주행하는 베이스와 7축의 여유자유도(redundant)를 갖는 머니플레이터(manipulator)가 모바일 양 쪽에 장착되어 있다.

이 매니플레이터는 3개의 링크가 연결되어 있으며 팔목에서 손목사이에 오프셋(off set)이 있어 일반적인 머니플레이터와 기구적 특성이 다르다. 각 조인트 사이에는 위치센서, 관절 토크센서 및 F/T 센서 등이 내장되어 있으며 파지 및 조작작업이 가능하다.

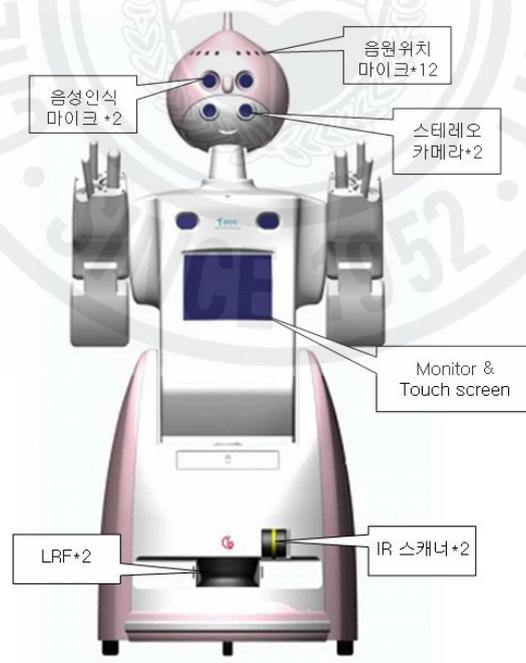


Fig. 7 Robot configuration

## 2. 모바일로봇의 기구학 해석

Fig. 8에서는 본 논문에서 다루고 있는 두 바퀴구동 모바일로봇의 기구학을 보여주고 있다.

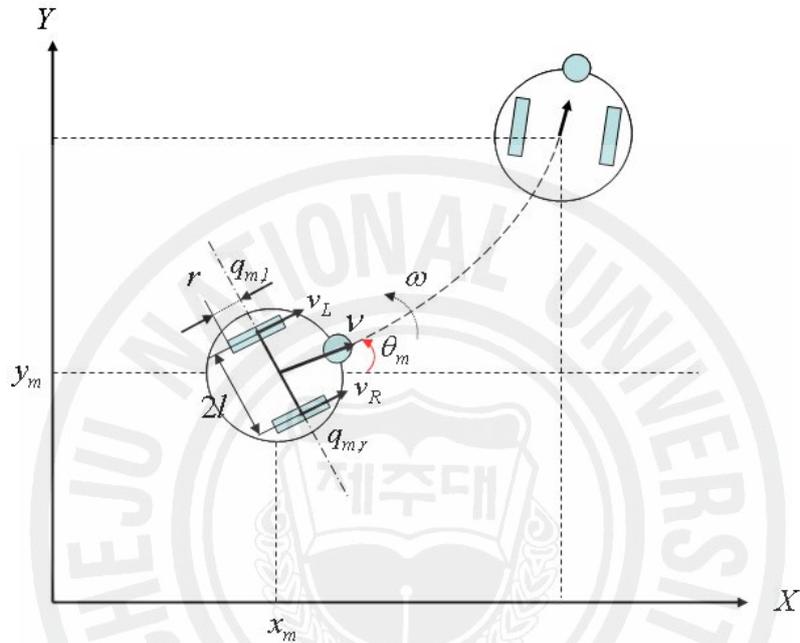


Fig. 8 Mobile robot modeling and coordinate system

여기서,

$X, Y$  : Cartesian 좌표계에서의 world frame

$x_m, y_m$  : Cartesian 좌표계에서의 모바일로봇 중심점의 좌표

$q_{ml}, q_{mr}$  : 모바일로봇의 왼쪽 바퀴와 오른쪽 바퀴의 각속도

$l$  : 모바일로봇 중심에서 바퀴 축까지의 거리

$r$  : 모바일로봇 중심에서의 각속도

$v_R, v_L$  : 모바일로봇의 왼쪽 및 오른쪽 바퀴의 선형속도

$u$  : 모바일로봇 중심에서의 선형속도

$\omega$  : 모바일로봇 중심에서의 각속도

모바일로봇은 실질적으로 2차원 평면을 주행하고 있으며, 고정된 좌표계 즉 모바일로봇 외부에 기준점을 둔 (X, Y)를 절대 좌표계라 하며, 이 절대 좌표계에서의 모바일로봇의 자세는 로봇의 중심  $(x_m, y_m)$ 의 좌표 값 및 모바일로봇의 방향 축과 X축 사이의 각도  $\theta_m$ 로 결정된다.

이  $(x_m, y_m, \theta_m)$ 를 모바일로봇의 자세(posture)라 정의하면 다음과 같이 표현한다.

$$P_m = \begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix} \quad (1)$$

바퀴로 구동하는 모바일로봇의 해석적인 기구학식을 유도하기 전에 가정해야 할 것들이 있다. 로봇의 바퀴는 회전축이 바닥의 수직선분과 직교하고 바퀴와 바닥 사이의 접촉면에서 순수 구름조건(pure rolling)과 미끄러짐 없음 조건(non-slipping)을 만족한다. 순수 구름조건은 미끄러짐이 없는 굴림 운동이라는 뜻으로 바퀴 표면의 점들이 연속적으로 바닥 표면의 점들과 일대일 접촉을 할 때를 말한다. 이때 바퀴와 접촉면 사이의 순간적 이동방향으로의 상대속도가 0이 되어야 하므로 오른쪽 바퀴와 왼쪽 바퀴에 대하여 식 (2), (3)과 같이 나타난다[5].

$$-\cos(\theta_m)\dot{x}_m - \sin(\theta_m)\dot{y}_m - l\dot{\theta}_m + r\dot{q}_{m,r} = 0 \quad (2)$$

$$-\cos(\theta_m)\dot{x}_m - \sin(\theta_m)\dot{y}_m - l\dot{\theta}_m + r\dot{q}_{m,l} = 0 \quad (3)$$

또한, 미끄러짐 없음 조건에 대하여 식 (4)와 같다.

$$-\sin(\theta_m)\dot{x}_m + \cos(\theta_m)\dot{y}_m = 0 \quad (4)$$

식 (2)~식(4)에 의해 Cartesian space의 모바일로봇의 속도와 각속도는 식 (5)와 같이 결정된다.

$$\dot{x} = \frac{r}{2}(q_{m,r} + q_{m,l})\cos\theta_m$$

$$\dot{y} = \frac{r}{2}(\dot{q}_{m,r} + \dot{q}_{m,l})\sin\theta_m \quad (5)$$

$$\dot{\theta}_m = \frac{r}{2l}(\dot{q}_{m,r} + \dot{q}_{m,l})$$

그리고, 모바일로봇을 제어하는데 있어서의 두 제어입력인 선형속도  $u$  와 각속도  $\omega$ 로 달리 표현하면, 식(6), (7)과 같다.

$$\nu = \frac{1}{2}(\nu_R + \nu_L) = r \frac{q_{ml} + q_{mr}}{2} \quad (6)$$

$$\omega = \frac{1}{2l}(\nu_R - \nu_L) = r \frac{q_{ml} - q_{mr}}{l} \quad (7)$$

여기서 주의해야 할 점은  $\nu_L, \nu_R$ 은 바퀴 축에서 본 각속도이며,  $\omega$ 는 좌표 계에서 본 모바일로봇의 각속도라는 것이다.

$[x_m \ y_m \ \theta_m]^T$ 와  $[\nu \ \omega]^T$ 은 자코비안행렬  $J$ 에 의해 식(8)과 같은 기구학 식과 연관 관계를 가지게 된다.

$$\dot{P} = J(p)\dot{q} \quad (8)$$

$$\dot{P}_m = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\theta}_m \end{bmatrix} = J(p) \begin{bmatrix} \nu \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} \quad (9)$$

초기위치를  $x_0, y_0, \theta_0$ 로 두면 모바일로봇의 현재위치는 식(10)과 같이 식 (9)의 적분형태가 된다.

$$P_m = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} \int_t^0 \nu \cos\theta_m d\tau \\ \int_t^0 \nu \sin\theta_m d\tau \\ \int_t^0 \omega d\tau \end{bmatrix} \quad (10)$$

이러한 기구학적 관계로부터 로봇은 매 제어주기마다 위치벡터를 계산하며,  $\nu$  및  $\omega$ 는 각 바퀴의 엔코더 신호로부터 계산된다[5].

### 3. 머니플레이터의 기구학 해석

#### 3.1 순기구학 해석

머니플레이터 로봇의 기구학적 구조는 7개의 축과 그리퍼로 구성되어 있으며 모든 축은 DC서보 모터에 의해 구동되어진다. 본 연구에서는 1, 3, 5축의 회전중심을 Z방향으로 설정하여 해석한다. 이에 대한 머니플레이터로봇의 모델링 및 좌표계 설정은 Fig. 9와 같다.

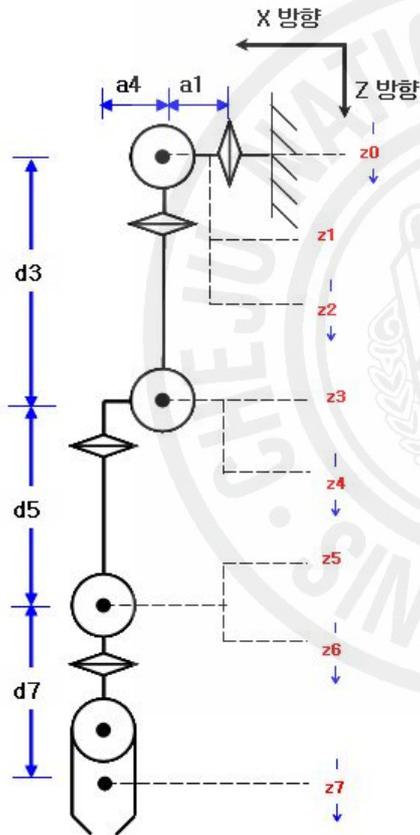


Table 1. DH link parameters.

Joint	Joint Angle( $\theta$ )	Distance ( $d$ )	Offset Angle( $\alpha$ )	Common Length( $a$ )
1	$\theta_1$	0	$\pi/2$	$a_1$
2	$\theta_2$	0	$-\pi/2$	0
3	$\theta_3$	$d_3$	$\pi/2$	0
4	$\theta_4$	0	$-\pi/2$	$a_4$
5	$\theta_5$	$d_5$	$\pi/2$	0
6	$\theta_6$	0	$-\pi/2$	0
7	$\theta_7$	$d_7$	$\pi/2$	0

a1: 0.0m, d5: 0.1858m,  
d3: 0.22m, d7: 0.225m  
a4: 0.0866m

Fig. 9 Manipulator modeling and coordinator

머니플레이터로봇의 기구학 해석을 위한 Denavit-Hartenberg 파라미터는 Table 1과 같다. 그리고, 각 링크의 파라미터로부터 좌표계간의 동차행렬 (homogeneous matrix)을 식(11)과 같이 구할 수 있으며, 이로부터 기저 좌표

계와 마지막 좌표계 사이의 변환관계를 나타내는 동차행렬 식(12)를 얻게 된다.

$$\begin{aligned}
 A_1 &= \begin{pmatrix} \text{Cos}\theta_1 & 0 & -\text{Sin}\theta_1 & a_1 \text{Cos}\theta_1 \\ \text{Sin}\theta_1 & 0 & \text{Cos}\theta_1 & a_1 \text{Sin}\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_2 &= \begin{pmatrix} \text{Cos}\theta_2 & 0 & -\text{Sin}\theta_2 & 0 \\ \text{Sin}\theta_2 & 0 & \text{Cos}\theta_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_3 &= \begin{pmatrix} \text{Cos}\theta_3 & 0 & -\text{Sin}\theta_3 & 0 \\ \text{Sin}\theta_3 & 0 & \text{Cos}\theta_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_4 &= \begin{pmatrix} \text{Cos}\theta_4 & 0 & -\text{Sin}\theta_4 & -a_4 \text{Cos}\theta_4 \\ \text{Sin}\theta_4 & 0 & \text{Cos}\theta_4 & -a_4 \text{Sin}\theta_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_5 &= \begin{pmatrix} \text{Cos}\theta_5 & 0 & -\text{Sin}\theta_5 & 0 \\ \text{Sin}\theta_5 & 0 & \text{Cos}\theta_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_6 &= \begin{pmatrix} \text{Cos}\theta_6 & 0 & \text{Sin}\theta_6 & 0 \\ \text{Sin}\theta_6 & 0 & -\text{Cos}\theta_6 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_7 &= \begin{pmatrix} \text{Cos}\theta_7 & 0 & \text{Sin}\theta_7 & 0 \\ \text{Sin}\theta_7 & 0 & -\text{Cos}\theta_7 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{pmatrix} & & & (11)
 \end{aligned}$$

$${}^0_7T = A_1 A_2 A_3 A_4 A_5 A_6 A_7 = \begin{pmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

$$\begin{aligned}
 x_x &= C_7(C_6(C_5(C_4(C_1C_2C_3 - S_1S_3) - C_1S_2S_4) + (-C_3S_1 - C_1C_2S_3)S_5) \\
 &\quad + (-C_1C_4S_2 - (C_1C_2C_3 - S_1S_3)S_4)S_6) + (C_5(-C_3S_1 - C_1C_2S_3) - (C_4(C_1C_2C_3 - S_1S_3) - C_1S_2S_4)S_5)S_7 \\
 x_y &= C_7(C_6(C_5(C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4) + (C_1C_3 - C_2S_1S_3)S_5) \\
 &\quad + (-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4)S_6) + (C_5(C_1C_3 - C_2S_1S_3) - (C_4(C_1C_2C_3 - S_1S_3) - S_1S_2S_4)S_5)S_7 \\
 x_z &= C_6(-C_1C_3S_2 - (C_1C_2C_3 - S_1S_3)S_4) - (C_5(C_4(C_1C_2C_3 - S_1S_3) - C_1S_2S_4) + (-C_3S_1 - C_1C_2S_3)S_5)S_6 \\
 y_x &= -C_1d_3S_2 - a_4C_4(C_1C_2C_3 - S_1S_3) + a_4C_1S_2S_4 + d_5(-C_1C_4S_2 - (C_1C_2C_3 - S_1S_3)S_4) \\
 &\quad + d_7(C_6(-C_1C_4S_2 - (C_1C_2C_3 - S_1S_3)S_4) - (C_5(C_4(C_1C_2C_3 - S_1S_3) - C_1S_2S_4) + (-C_3S_1 - C_1C_2S_3)S_5)S_6) \\
 y_y &= C_7(C_6(C_5(C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4) + (C_1C_3 - C_2S_1S_3)S_5) + (-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4)S_6) \\
 &\quad + (C_5(C_1C_3 - C_2S_1S_3) - (C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4)S_5)S_7 \\
 y_z &= C_7(C_5(C_1C_3 - C_2S_1S_3) - (C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4)S_5) - (C_6(C_5(C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4) \\
 &\quad + (C_1C_3 - C_2S_1S_3)S_5) + (-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4)S_6)S_7 \\
 z_x &= C_6(-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4) - (C_5(C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4) + (C_1C_3 - C_2S_1S_3)S_5)S_6 \\
 z_y &= -d_3S_1S_2 - a_4C_4(C_2C_3S_1 + C_1S_3) + a_4S_1S_2S_4 + d_5(-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4) \\
 &\quad + d_7(C_6(-C_4S_1S_2 - (C_2C_3S_1 + C_1S_3)S_4) - (C_5(C_4(C_2C_3S_1 + C_1S_3) - S_1S_2S_4) + (C_1C_3 - C_2S_1S_3)S_5)S_6) \\
 z_z &= C_7(C_6(C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5) + (C_2C_4 - C_3S_2S_4)S_6) + (-C_3S_2S_3 - (C_3C_4S_2 + C_2S_4)S_5)S_7 \\
 p_x &= C_7(-C_5S_2S_3 - (C_3C_4S_2 + C_2S_4)S_5) - (C_6(C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5) - S_2S_3S_5) + (C_2C_4 - C_3S_2S_4)S_6)S_7 \\
 p_y &= C_6(C_2C_4 - C_3S_2S_4) - (C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5)S_6 \\
 p_z &= d_1 + C_2d_3 - a_4C_3C_4S_2 - a_4C_2S_4 + d_5(C_2C_4 - C_3S_2S_4) + d_7(C_6(C_2C_4 - C_3S_2S_4) - (C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5)S_6)
 \end{aligned}$$

[단,  $C_n = \cos(\theta_n)$ ,  $S_n = \sin(\theta_n)$ ]

식(12)는 7축 로봇의 기구학을 대표한다. 여기서 4열의  $p_x, p_y, p_z$ 는 손끝(end-effector)의 위치를 나머지 요소들은 말단의 방향을 나타낸다.

### 3.2 역기구학 해석

역기구학은 정지계를 기준으로 하여 손끝의 위치와 방위가 주어진 경우 이 목표한 결과를 이루기 위한 관절각 조합을 어떻게 계산하는지에 관해 다루는 것을 말한다.

관절각을 구하는 문제를 해결하기 위해서는 로봇의 고정 축 에 기준하여 손목(wrist) 좌표계를 찾는 문제로 좌표변환을 수행하고, 역기구계산을 하여 관절각을 풀어야 한다.

기구 방정식을 푸는 것은 비선형 문제가 된다.  ${}^0_7T$ 의 수치 값이 주어질때,  $\theta_1, \theta_2, \dots, \theta_7$ 의 값을 찾기 위하여 식(12)로 주어진 식을 고려하도록 한다.

먼저 Fig. 10의 (a)에서와 같이 머니플레이터의 어깨(shoulder)의 위치를  $P_s$ , 팔꿈치(elbow)의 위치를  $P_e$  이들 조인트 사이에 연결되는 선을  $P_{ew}$ , 팔목의 위치를  $P_w$ 라 정의하겠다. 우선 정지계  $P_s$ 에서  $P_w$ 까지 일 경우를 보면  $\theta_3 = 0$ 일 때  $P_e$ 값을 다음과 같이 구할 수 있다.

7DOF의 조인트 각을 구하기 위해서  $\theta_3$ 를 0으로 놓고 6DOF에 관한 역기구학을 풀도록 하겠다.  $\theta_1, \theta_2$ 를 구하기 위해서  $\theta = 0$  일 때의 어깨부터 손목까지를 고려하여  $P_{ew}$ 의 선분에 대해서 Fig. 10의 (b)를 이용하도록 하겠다.

여기서,  $P$ 는  $P = \text{desirePos}(1:3, 4)$ (desiredPos: 4 by 4 matrix)으로 나타내며 (1:3, 4)는 matrix 4열의 1~3행까지의 값만 뽑아내는 것을 나타낸다.

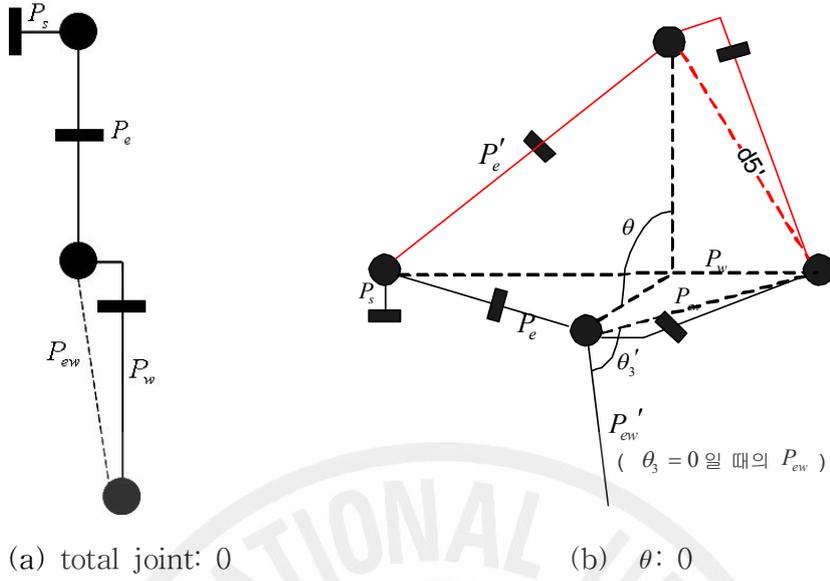


Fig. 10 Joints states

일단  $P'_e$  값의 계산을 위하여 식(13)과 같이  $P_s$ 와 6DOF일 때의  $A'_1 A'_2$  까지 변환 행렬의 위치와  $P_e$ 의 위치에 대한 차를 이용하여 계산한다.

$$P'_e = (A'_1 A'_2)(1:3,4) - P_s \quad (13)$$

이때 계산된 값이 스칼라 값이기 때문에 팔목의 방향 값을 얻어내기 위한  $n = P_w / |P_w|$  값을 계산하며, 최종  $P'_e$  값을 (14)에서 얻어낸다.

$$P'_e = n(n \cdot P'_e) + \{P'_e - n(n \cdot P'_e)\} \cos \theta + (P'_e \times n) \sin \theta \quad (14)$$

$P_e$ 를 이용하여 실제 7DOF상의  $\theta_1, \theta_2$ 를 식(15), (16)로부터 구할 수 있다.

$$\theta_1 = \tan^{-1} \left( \frac{-P_{ey}}{-P_{ex}} \right) \quad (15)$$

$$\theta_2 = \tan^{-1} \left( \frac{\sqrt{P_{ex}^2 + P_{ey}^2}}{P_{ez}} \right) \quad (16)$$

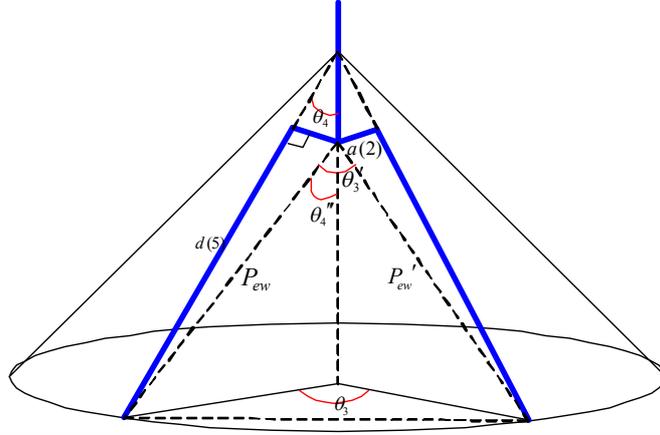


Fig. 11 Transformation of the  $\theta_3$  value

식 (15), (16)에서 세워진 각을 바탕으로 Fig. 11로 나타낼 수 있다.  $\theta_3$ 을 구하기 위해 다른 각을 먼저 구한 후 식(17), (18)으로부터 식(19)를 얻어낼 수가 있다.

$$\theta_3' = \tan^{-1} \left( \frac{|P_{ew} \times P'_{ew}|}{P_{ew} \cdot P'_{ew}} \right) \quad (17)$$

$$\theta_4''' = \frac{\pi}{2} - \cos^{-1} \frac{a_2}{P_{ew}} + \theta_4 \quad (18)$$

$$\theta_3 = 2 \sin^{-1} \left( \frac{2P'_{ew} \sin \left( \frac{\theta_3'}{2} \right)}{P'_{ew} \sin \theta_4'''} \right) \quad (19)$$

다음 팔목부터 손끝까지 고려하여 나머지  $\theta_4, \theta_5, \theta_6$ 을 계산을 한다.

팔목부터 손끝까지의 식을 단순화하기 위해 손목을 기반으로 하여 새로운 desiredPos를 정한다.

$$desiredPos = \begin{pmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{라고 하면}$$

$\neq w \, desiredPos = \begin{pmatrix} n' & o' & a' & p' \\ 0 & 0 & 0 & 1 \end{pmatrix} = inv({}^0A_4)^* \, desiredPos$ 로 재정의 한다.  ${}^0A_4$ 는 DH matrix이며  $A_1A_2A_3A_4$ 에서 4열의 1~3행까지 값을 뽑아낸다. 다음과 같이 각 좌표의 방향을 이해하면 다음과 같이 손목 이후의 조인트각을 구할 수 있다.

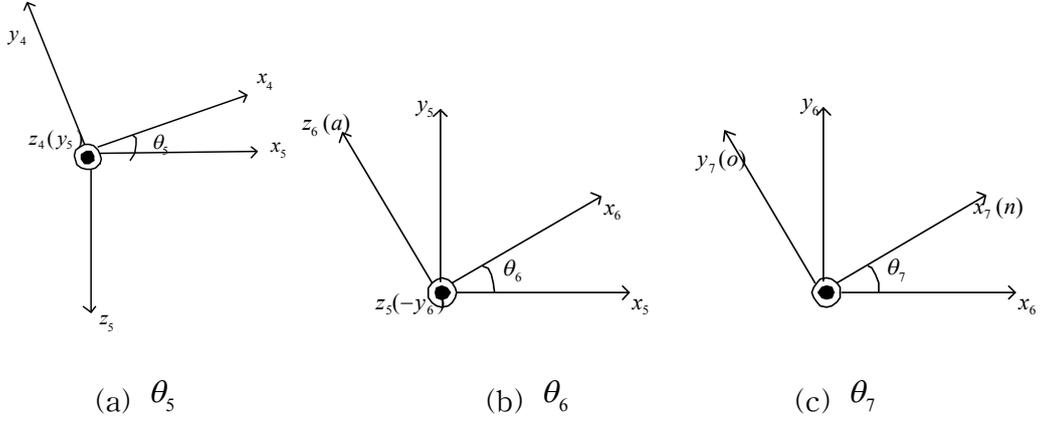


Fig. 12 Joint angle after wrist

Fig. 12는 좌표의 변화에 따른  $\theta_5, \theta_6, \theta_7$  값이며 이를 바탕으로 다음 나머지 조인트 각에 대한 식들을 구할 수 있다.

$$\theta_5 = \tan^{-1} \frac{-z_5 \cdot x_4}{-z_5 \cdot y_4} = \tan^{-1} \frac{-a'_y}{-a'_x} \quad (20)$$

$$\theta_6 = \tan^{-1} \frac{-a \cdot x_5}{-a \cdot y_3} = \tan^{-1} \frac{-a'_x \cos(\theta_5) - a'_y \sin(\theta_5)}{-a'_x} \quad (21)$$

$$\theta_7 = \tan^{-1} \frac{-n'_x \sin(\theta_5) + n'_y \cos(\theta_5)}{-o'_x \sin(\theta_5) + o'_y \cos(\theta_5)} \quad (22)$$

### 3.3 자코비안 행렬

운동중인 머니플레이터를 해석하는데 있어서 시간적으로 변화하는 위치와 방위를 표시하기 위해서는 각 링크의 상대적 선속도와 회전속도를 고려하여야 한다.

손끝의 좌표와 머니플레이터의 관절각도 좌표의 관계는 복잡한 비선형 대수 방정식으로 표현되기 때문에 일반적으로는 이것을 해석적으로 푸는 것은 곤란하다. 이를 위하여 자코비안 행렬을 이용하면 이 복잡성을 피할 수가 있다. 관절형 머니플레이터의 경우 실제로 움직이는 것은 관절각도이기 때문에 머니플레이터의 손끝의 위치 및 자세와 관절각도와 관계가 필요하다.

자코비안 행렬을 구하기 위해서는 식 (12)의 4열의 1~3행 까지의  $P_x, P_y, P_z$  손끝의 위치가 필요하다.

여기서  $P_x = f_1(q), P_y = f_2(q), P_z = f_3(q)$ 으로 정의하도록 하면,

$$\begin{aligned}
 f_1(q) &= C_7(-C_5S_2S_3 - (C_3C_4S_2 + C_2S_4)S_5) - (C_6(C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5) - S_2S_3S_5) \\
 &\quad + (C_2C_4 - C_3S_2S_4)S_6S_7 \\
 f_2(q) &= C_6(C_2C_4 - C_3S_2S_4) - (C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5)S_6 \\
 f_3(q) &= d_1 + C_2d_3 - a_4C_3C_4S_2 - a_4C_2S_4 + d_5(C_2C_4 - C_3S_2S_4) + d_7(C_6(C_2C_4 - C_3S_2S_4) \\
 &\quad - (C_5(C_3C_4S_2 + C_2S_4) - S_2S_3S_5)S_6)
 \end{aligned} \tag{23}$$

식(23)에서 변수와 티끝의 관계가 비선형이므로 이를 갖고 다루는 것은 매우 어렵기 때문에 이를 다음과 같이 시간에 대해 미분하여 선형적인 속도의 관계식으로 표현하여 사용하여야 한다.

조인트 변수가 다음과 같다면  $q_i = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6 \theta_7]$ 은 자코비안 행렬을 구하기 위해서 각각  $f_1(q), f_2(q), f_3(q)$  조인트 변수에 대해 미분되어야 한다.

여기서  $J_{ij} = \frac{\partial f_i(q)}{\partial q_j} \quad i = 1, 2, 3 \text{ and } j = 1, 2, \dots, 7$

$$\dot{P} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} = J \dot{q} \quad J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} & 0 \\ J_{21} & \ddots & J_{24} & & & & 0 \\ J_{31} & & \cdots & 0 & 0 & 0 & \\ J_{41} & & \cdots & & & & 0 \\ J_{51} & & & J_{54} & \ddots & & 0 \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & 0 & 1 \end{bmatrix} \tag{24}$$

식(24)에서  $J$ 는 어떠한 속도의 제한조건도 없는 경우의 모바일 매니플레이터에 대한 자코비안 행렬(Jacobian Matrix)을 나타낸다.

#### 4. 모바일-머니플레이터로봇의 기구학 해석

독립적인 목적으로 설계된 두 개의 로봇을 결합하여 하나의 작업수행을 위하여 각각의 로봇이 동시에 제어되어야 하므로 전체 시스템의 기구학을 해석하여 지능로봇이 제어되어야 한다. 모바일 로봇은 non-holonomic 시스템이고, 머니플레이터로봇은 holonomic 시스템이므로 이동머니플레이터의 기구학은 속도기구학을 통해 모바일로봇과 머니플레이터로봇을 결합한다.

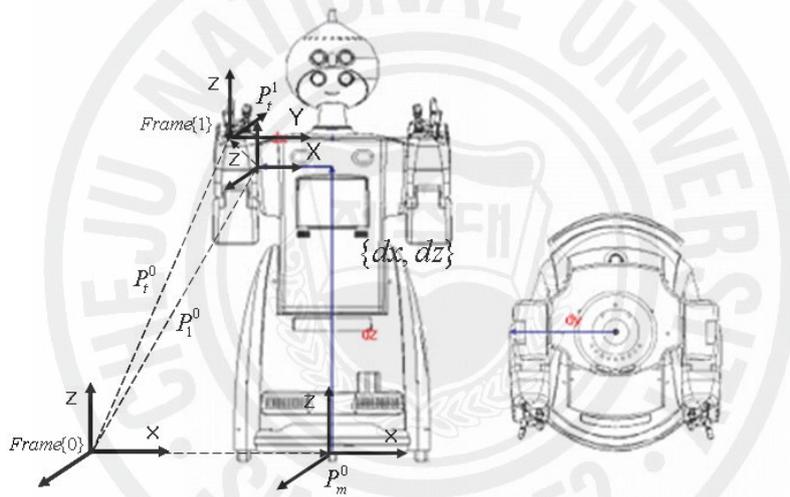


Fig. 13 Coordinate of Mobile Manipulator

Fig.13 에서는 모바일-머니플레이터의 모델링을 나타내었다. 머니플레이터의 조인트 변수를  $q_i = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6 \theta_7]^T$ , 모바일 로봇의 조인트 변수를  $q_m = [x_m y_m \theta_m]^T$ 라 정의하면 모바일-머니플레이터의 변수 식은 식(25)와 같다.

$$q = \begin{bmatrix} q_m \\ q_i \end{bmatrix} = [x_m \ y_m \ z_m \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6 \ \theta_7]^T \quad (25)$$

모바일로봇의 각 축에 대한 조인트 변수에 의해서 형성되는 모바일의 로봇과 머니플레이터와의 고정점의 위치  $P_1^0 = [p_x \ p_y \ p_z \ \theta_m]^T$ 를 프레임 {1}로 설정하였다.  $P_1^0$ 은 고정 좌표계 프레임{0}을 기준으로 프레임 {1}까지의 위치벡터이다. 모바일-머니플레이터의 기구학은 우선 모바일로봇은 3축이므로 조인트 변수  $\dot{\theta}_m$ 은 바닥 면의 고정 프레임에 대한 직교좌표 공간상에서의 선, 각속도  $\dot{P}_m^0 = [V_m^0 \ \omega_m^0]^T$ 와 같으며 머니플레이터 로봇의 조인트 변수들의 벡터  $\dot{\theta}_t$ 의 선, 각속도는  $\dot{P}_t^1 = [V_t^1 \ \omega_t^1]^T$ 과 같다. 각각의 로봇에 대한 자코비안이  $J_m^0, J_t^1$ 으로 주어질 때, 모바일-머니플레이터 로봇의 자코비안을  $J_1^0$ 라고 하면, 바닥면에 대한 말단부의 선, 각속도  $\dot{P}_t^0 = [V_t^0 \ \omega_t^0]^T$ 는 다음 식 (26)과 같다[5].

$$\begin{aligned} \dot{P}_t^0 &= \begin{bmatrix} V_t^0 \\ \omega_t^0 \end{bmatrix} = \begin{bmatrix} V_m^0 \\ \omega_m^0 \end{bmatrix} + \begin{bmatrix} \omega_m^0 + R_1^0 V_t^1 \\ R_1^0 \omega_t^1 \end{bmatrix} \\ &= \begin{bmatrix} J_m^0 & J_t^0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_m \\ \dot{\theta}_t \end{bmatrix} \end{aligned} \quad (26)$$

$R_1^0$ 은 {0}에서 머니플레이터의 {1}까지의 회전변환행렬이다. 즉, 식 (26)에 의해 손끝의 운동은 모바일과 머니플레이터가 함께 관여함을 알 수 있다.

## IV. 충돌회피 알고리즘

로봇의 자율 주행 기술 중 안전성의 평가에서 장애물 회피는 가장 기본적이고 중요한 기술이라 할 수 있다. 센서기반의 충돌회피 알고리즘들은 주어진 환경에서 장애물에 대한 충돌회피하는 동작을 결정하기 위하여 센서정보를 이용한다. 반응제어의 핵심은 실시간으로 장애물을 회피하는 것인데, 이를 위해서는 복잡한 알고리즘보다 빠른 연산을 위해서 간단한 알고리즘이 더 효율적이다. 본 논문에서는 모바일로봇의 Elastic Force를 이용한 장애물 회피 알고리즘을 제안한다.

### 1. Elastic Strips

Elastic Strip(Khatib, 1997) 방법은 Khatib가 제안한 방법으로 로봇의 궤적을 탄성물질이라 가정하여 장애물을 안전하게 회피하도록 한 방법이다[19]. 장애물 접근 시 물체 사이에 가상 힘이 작용한다는 것에서는 인공전위계 개념과 유사하나 목표지점까지의 도달 동안 탄성 힘이 작용한다는 데서 인공전위계 방식보다는 보다 안정된 주행 기술을 구사한다. 또한 여러 자유도를 갖는 로봇에 적용이 용이하며 동적이고 불확실한 환경에서의 장애물회피에 대한 최적화된 회피알고리즘을 보인다는 장점을 가지고 있다. Elastic Strip방법에는 내부 힘과 외부 힘이 존재하는데 다음에 이어 설명한다.

#### 1.1 내부 힘 (Internal Forces)

이 내부 힘은 로봇이 정확한 목표지까지 이동할 수 있도록 도와주는 힘이라 할 수 있다. 이 힘은 로봇의 궤적(trajjectory)상의 형상(configuration)사이에 스프링이 연결되어 있다 가정하였다.

이 궤적으로 이동 중에 궤적을 이탈하게 되면 이 궤적의 이탈을 방지하기 위하여 탄성 힘인 내부 힘이 작용하여 원래의 궤적 찾게 도와준다.

이때 궤적을 많이 이탈하게 되면 내부 힘이 커지게 되며 원래 계획되어진 궤적에 가까워지게 되면 서서히 내부 힘이 감소하게 된다.

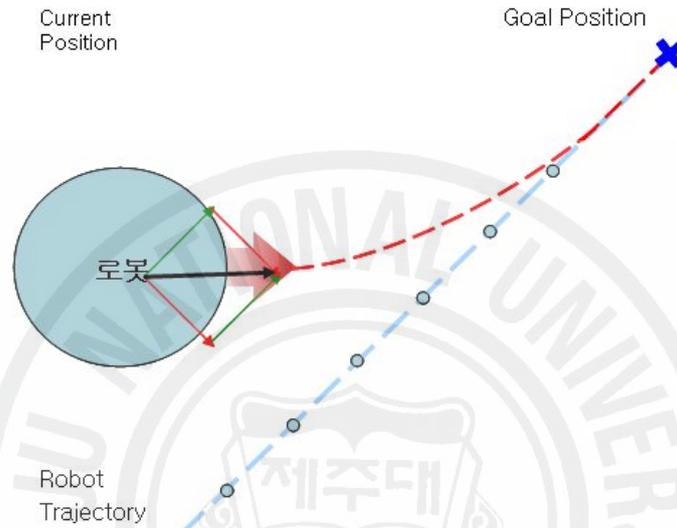


Fig. 14 Operation of internal force

## 2.2 외부 힘(External Force)

외부 힘은 자율이동 로봇의 경로계획이나 작업공간 내에서 장애물과 접했을 때에 충돌없이 목적지까지 이동하도록 하는 힘으로서 포텐셜필드 방식과 유사하다.

포텐셜 필드 방법(Potential Fields Method)은 로봇을 포텐셜 함수의 영향을 받는 물체로 취급하는 것이다. 포텐셜 분포를 보면 장애물을 제외한 모든 모바일로봇의 작업공간 영역에서 도착점으로 기울어져 내려가는 현상을 하게 된다[18]. 따라서, 로봇은 시작점에서 음의 기울기를 따라가면 도착점에 도착하게 된다. Fig. 15는 로봇이 위치한 시작점부터 목표점까지의 포텐셜 필드를 부여해준 것과 장애물에 반발력을 가질 수 있게 높은 에너지 값을 부여해준 모습을 볼 수 있다.

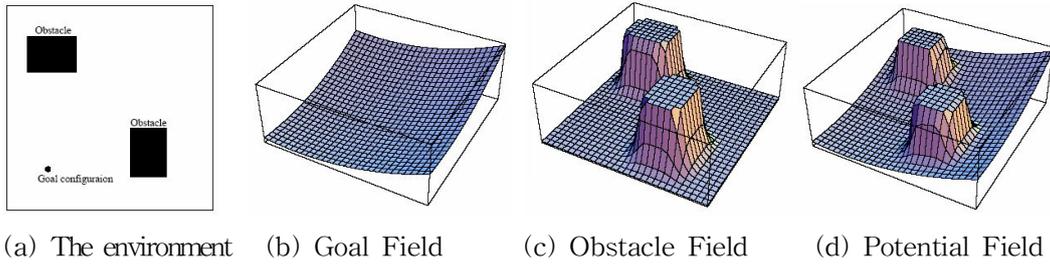


Fig. 15 Potential Field Method

일반적으로 포텐셜 필드 방법은 목표점에 의한 인력에 의해 발생하는 부분  $U_{goal}(q)$ 과 장애물에 의한 척력에 의해 발생하는 부분  $U_{obs}(q)$ 으로 구분되며, 포텐셜필드를 표현하는 기본 식은 다음 (27)식과 같다[18].

$$\begin{aligned}
 U(q) &= U_{goal}(q) + U_{obs}(q) \\
 U_{goal} &= \frac{1}{2} \varepsilon \rho(q, q_{goal})^2 \\
 U_{obs} &= \frac{1}{2} \eta \left( \frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho^0} \right)^2 \\
 F &= F_{goal} + F_{obs} = -\nabla U(q) = - \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{pmatrix}
 \end{aligned} \tag{27}$$

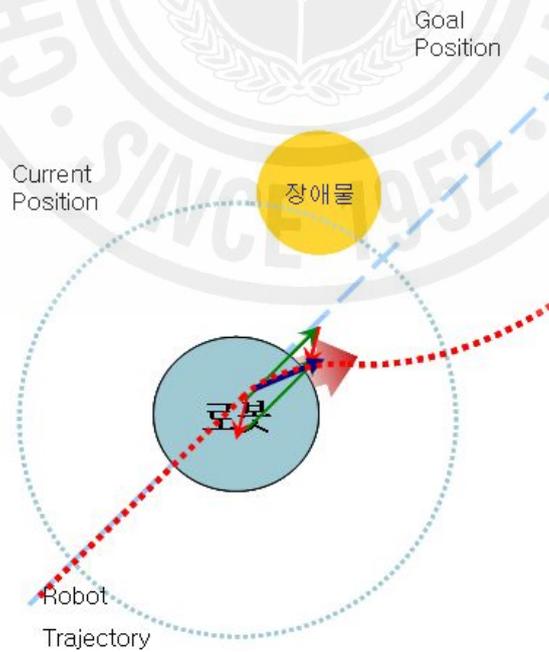


Fig. 16 Effect of the external force

## 2. 모바일 머니플레이터 로봇의 장애물 회피

### 2.1 장애물 회피 방법

매 샘플링 시간마다 이전의 샘플링되어 계획된 궤적을 수정한다. 궤적의 수정은 자세(posture)들의 수정을 의미한다. 과정은 다음과 같다.

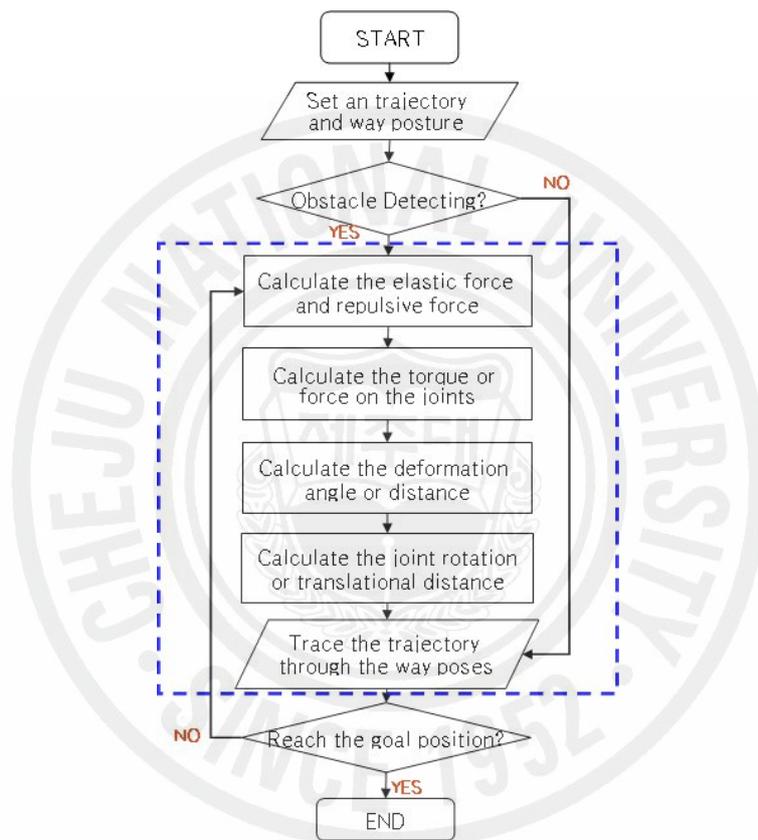


Fig. 17 Collision avoidance procedure

- (1) 초기 궤적이 계획되고 순차적으로 주어진 자세를 따라 궤적이 주어진다.
- (2) 이전의 궤적에 의해 궤적이 수정된다.
  - ⓐ 로봇의 제어 점(control point)에 가해지는 내부 힘과 외부 힘을 계산한다.
  - ⓑ 제어 점들에 가해지는 힘에 대해 로봇 조인트 및 방향의 토크 값을 계산한다.
  - ⓒ 조인트 및 로봇 방향의 토크 값에 대한 이동할 각도 및 거리를 계산한다.

- ④ ㉔에서 계산된 조인트, 로봇 방향의 각도 변형에 의해 이동 및 회전한다.
- (3) 다음 자세를 따라 궤적을 변형한다.
- (4) 만약 로봇이 원하는 목표 자세에 도달하였는지 확인한 후 충분히 목표 자세에 도달하지 않았다면 순서를 멈춘 후 다시 (2)단계로 돌아간다.

## 2.2 로봇 자세(robot posture)

궤적 변형을 위해서는 로봇 자세(robot posture) 설정이 요구된다. 여기서 말하는 로봇 자세는 초기 출발위치의 자세와 목표 점에서의 자세를 포함한 이들 사이에 일정 시간에 따라 설정된 모바일 로봇의 위치 및 팔의 동작위치를 말한다. 이 논문에서는 이것을 로봇 자세(robot posture)라고 부르도록 하겠다 [17, 19].

궤적은 매 시간마다 초기 자세와 목표의 자세사이에 접속하고 있는 연속된 형상으로서 궤적 수정에 관여한다. 초기 자세에서 목표까지 자세사이에 많은 자세들을 설정하는데 각 로봇 자세들 위에 힘을 받게 되는 위치들은 전 단계의 로봇 자세와 다음 단계의 로봇 자세들 사이에 탄력이 있는 스프링과 연결되게 된다.

Fig. 18은 3단계의 로봇 자세를 보인다. (a), (b), (c)는 초기 자세와 목표자세 그리고 그 사이에 여러개 설정된 경유 자세를 보여주고 있다.

여기서 이미 거쳐 온 전 단계의 자세를  $\{P_1\}$ , 현재의 자세를  $\{P_2\}$ , 다음 거쳐 갈 로봇 자세를  $\{P_3\}$ 라 하겠다.

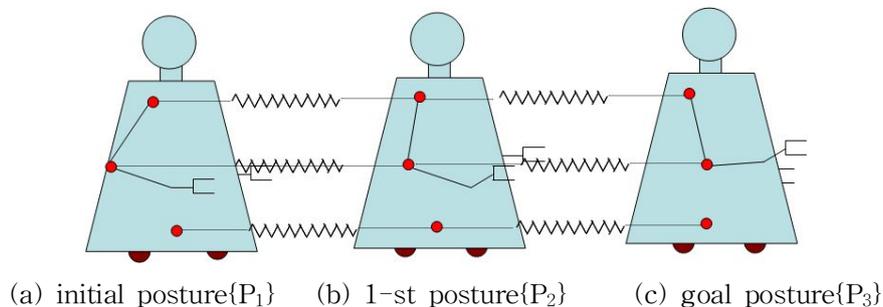


Fig. 18 Trajectory of a mobile manipulator.

이들 로봇 자세사이에 연결된 스프링 힘은 궤적을 이탈 했을 경우 보다 안정적으로 목표 자세에 도달하게 해준다.

로봇 자세는 적절한 샘플링 시간에 대해서 설정을 해주는 것이 좋다. 이들 설정이 많아지게 되면 장애물 반응 시 보다 안정적으로 매끄럽게 회피를 하지만 많은 계산을 요구하게 되며 로봇 자세가 충분히 설정되지 않으면 이동 주행시간은 향상되나 장애물과 겹치지 않는다 해도 약간의 장애물과 충돌 할 수 있으므로 적절한 로봇자세의 설정을 필요로 한다.

### 2.3 제어 점(control points)

로봇 자세에서 장애물에 대한 자세 변형을 위해서 로봇위에 가해지는 힘을 받는 점인 제어 점(control points)이 설정된다. 이들 제어 점에서는 앞서 설명한 내부 힘과 외부 힘이 작용하게 된다.

이들의 점들은 로봇이 동작을 하는 동안 힘의 효과가 균등한 힘을 받을 수 있도록 좌우 균형적인 위치의 제어 점들이 선택되어야 한다.

우선 힘을 받는 위치가 적절한 조인트 위치에 존재해야 하고 하나의 조인트 변형으로 인해 다른 조인트의 로봇 자세 변형을 일으킬 수 있어야 한다.

Fig. 19는 이 로봇에서 Elastic Force계산을 위한 제어 점을 표현하고 있으며 여기서  ${}^jP_i$ 는  $i$ 번째 제어 포인트에  $j$ 번째 로봇 자세라 표현을 하겠다.

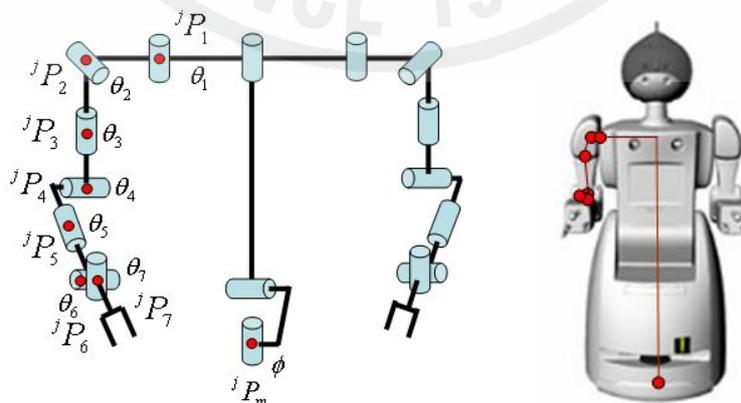


Fig. 19  ${}^jP_i$ represents the control points on the body.

### 3. 내부 힘의 계산

내부 힘 계산을 위해서는 궤적이 미리 설계되어 있어야 하며 설계된 궤적에 의하여 탄성 힘이 계산된다.

로봇 자세는  $\{P_1\}$ ,  $\{P_2\}$ ,  $\{P_3\}$  와의 위치벡터의 차에 의하여 힘의 세기가 결정된다..

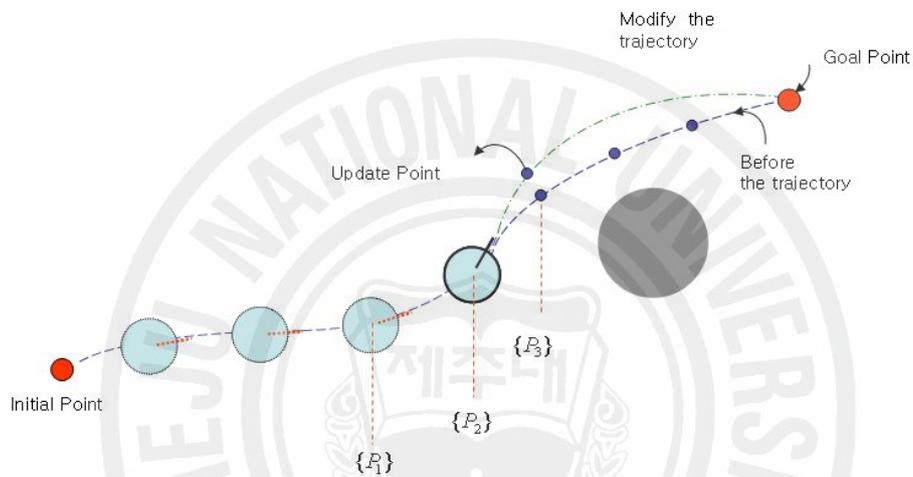


Fig. 20 Transformation procedure of the trajectory

이들 현재  $\{P_2\}$ 가 다음 이동해야 할  $\{P_3\}$ 과 거리가 길면 이 스프링 힘은  $\{P_3\}$ 로 이동을 위해서 더 강한 스프링 힘을 주게 될 것이며 이에 의한 힘이 커지면 미리 계획된 궤적에 가까워지게 될 것이다.

모바일의  ${}^jP_m$ 와 매니퓰레이터의  ${}^iP_i$ 에 가해지는 내부 힘  ${}^e f_m$ 는 다음과 같이 기술된다. 여기서  ${}^jP_m$ 는  $j$ 번째( $\{P_2\}$ ) 로봇 자세에  $i$ 번째 제어 점을 말하며  ${}^{j-1}P_m$ ,  ${}^{j+1}P_m$ 은 각각  $\{P_1\}$ ,  $\{P_3\}$ 의 제어 점에 해당한다.

식 (28)은 내부 힘 계산을 위한 위치 값이며, 모바일은  $z$ 축으로의 이동의 없기 때문에  ${}^jP_m$ 의  ${}^jZ_m$ 를 0으로 넣어 계산을 한다.

$${}^jP_m = [{}^jx_m \ {}^jy_m \ {}^jz_m]^T, \quad {}^jP_i = [{}^jx_i \ {}^jy_i \ {}^jz_i]^T \quad i = 1, 2, \dots, 7 \quad (28)$$

앞서 설명한 것과 같이 내부 힘 계산은  $\{P_1\}$ ,  $\{P_2\}$ ,  $\{P_3\}$ 의 계산에 의하여 목표 자세로 변환을 위한 힘을 계산하게 된다.

$${}^e f_m = k_c \left( \frac{d_i^{j-1}}{d_i^{j-1} + d_i^j} ({}^{j+1}P_i - {}^{j-1}P_i) - ({}^jP_i - {}^{j-1}P_i) \right), \quad i = m, 1, 2, \dots, 7 \quad (29)$$

$$\begin{aligned} d_i^{j-1} &= \sqrt{({}^{j-1}x_i - {}^jx_i)^2 + ({}^{j-1}y_i - {}^jy_i)^2 + ({}^{j-1}z_i - {}^jz_i)^2} \\ d_i^j &= \sqrt{({}^jx_i - {}^{j+1}x_i)^2 + ({}^jy_i - {}^{j+1}y_i)^2 + ({}^jz_i - {}^{j+1}z_i)^2} \end{aligned} \quad (30)$$

여기서  $d_i^{j-1}$ 는  $j-1$ 과  $j$ 번째와  $d_i^j$ 는 다음  $j+1$ 와  $j$ 의 제어 점과의 거리에 해당된다.

$\frac{d_i^{j-1}}{d_i^{j-1} + d_i^j}$ 는 내부 힘에 대한 비례계수이며 이 값의 결정에 따라 계획된 궤적에 빨리 수렴하고자 정도를 결정하게 되며,  $k_c$ 는 내부 힘의 작용에 의하여 수축 값에 의한 수정 궤적의 얼마나 부드럽게 수축하느냐에 따른 값을 결정한다 [19].

#### 4. 외부 힘의 계산

외부 힘 계산을 위해서는 가장 가까운 장애물의 위치 측정을 위한 센서 값의 정보와 센서의 값을 이용한 장애물과의 거리계산으로 인한 장애물의 좌표 및 로봇이 장애물을 어느 영역부터 인지하여 회피할지에 관한 안전거리에 관한 정보가 필요하다. Fig. 21은 모바일 및 머니플레이터의 각각의 제어 점 주위에 안전거리에 관해 표현하였다. 안전영역 안에 장애물 감지했을 경우 제어 점과의 가장 가까운 장애물과의 거리를 측정하게 되는데 이때 거리 측정 시 제어 점과 같은  $z$ 좌표상에서 센싱된 장애물의  $x, y$ 좌표를 고려한 최소 값을 구해야 한다. 안전거리는 작은 값을 넣게 되면 보다 넓은 장애물 회피 영역을 얻을 수 있으나 장애물과의 충돌 가능성 또한 늘어나기 때문에 적절한 값을 지정해야 한다.

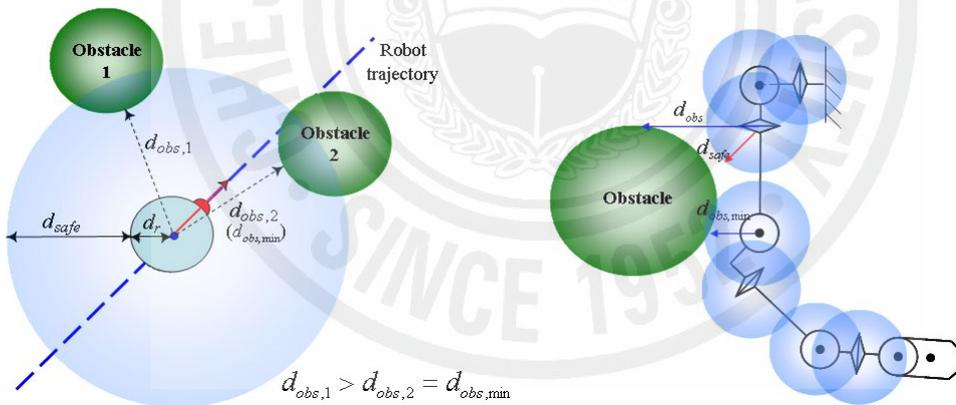


Fig. 21 A safety area of control point

$d_{obs}$ : 로봇과 장애물간의 거리

$d_r$ : 모바일의 반지름

$d_{safe}$ : 장애물 회피를 위한 경계 값

$d_o$ : 로봇 중심점에서의 외부 힘 작용 영역

$$d_o = d_{safe} + d_r$$

$$d_{obs} = \min(d_{obs,i}) = d_{obs,min}$$

이들 안전영역에서 감지된 장애물을 피하기 위해서는 전역 좌표계상에서의 장애물의 위치 값을 계산하여야 한다. 장애물 좌표를 결정하기 위해서 다음과 같이 전역좌표계에 관한 로봇 방향계산으로 인하여 코사인 제2법칙 계산에 의하여 로봇중심으로의 장애물의 위치 및 방향을 얻을 수가 있다. 어느 방향 쪽에 위치하는 지에 관한 정보를 얻을 수가 있으며 장애물의 좌표를  $P_o = [x_o \ y_o \ \theta_o]^T$ 과 같이 지정한다.

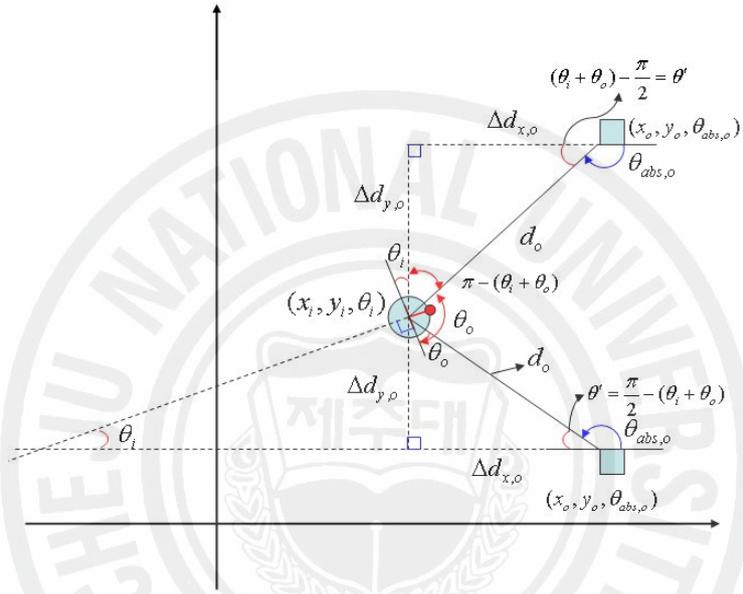


Fig. 22 Compute the position of obstacle

$$\begin{aligned}
 P_o &= [x_o \ y_o \ \theta_o]^T \\
 x_o &= x_i + d_o \sin(\theta_i + \theta_o) \\
 y_o &= y_i + d_o \cos(\theta_i + \theta_o) \\
 \theta_{abs,o} &= -(\pi - \theta')
 \end{aligned} \tag{31}$$

식 (31)은 전역좌표계에 대한 장애물의 위치 및 방향에 대한 식이다.

실제 로봇에 적용할 외부 힘 계산을 위해서 포텐셜필드방식을 이용한 식 (32)을 적용한다. 이 식을 바탕으로 외부 힘을 얻기 위해서  $V_r(p)$ 에서의 그래디언트(gradient)로부터 (33)식이 구해진다.

$$V_r(p) = \begin{cases} \frac{1}{2}k_r(d_o - d_{obs,min})^2 & \text{if } d_{obs,min} < d_o \\ 0 & \text{else.} \end{cases} \quad (d_{safe} + d_r = d_o) \quad (32)$$

$${}^r f_m = -\nabla V_{ert} = k_r(d_o - d_{obs,min}) \frac{\vec{d}}{\|d\|} \quad (33)$$

$$\vec{d} = P_m - P_o, \quad \|d\| = \sqrt{(x_m - x_o)^2 + (y_m - y_o)^2 + (z_m - z_o)^2} \quad (34)$$

여기서  $d_{obs,min}$ 는 측정된 장애물과의 최소거리에 해당되며,  $k_r$ 는 장애물 회피를 위한 외부 힘의 세기에 해당되면  $k_r$  값이 클수록 반발력이 더 커지게 된다.



## 5. elastic force에 의한 궤적 변형

앞서 계산된 힘의 최종 힘은 내부 힘과 외부 힘의 벡터 값의 합으로서 계산된다. 이 식(35)에서 계산되어진 힘에 의하여 형상의 변화를 일으키게 되고 궤적의 변형을 일으키게 된다.

$$F_i = {}^e f_i + {}^r f_i \quad \text{for } i = m, 1, 2, \dots, 7 \quad (35)$$

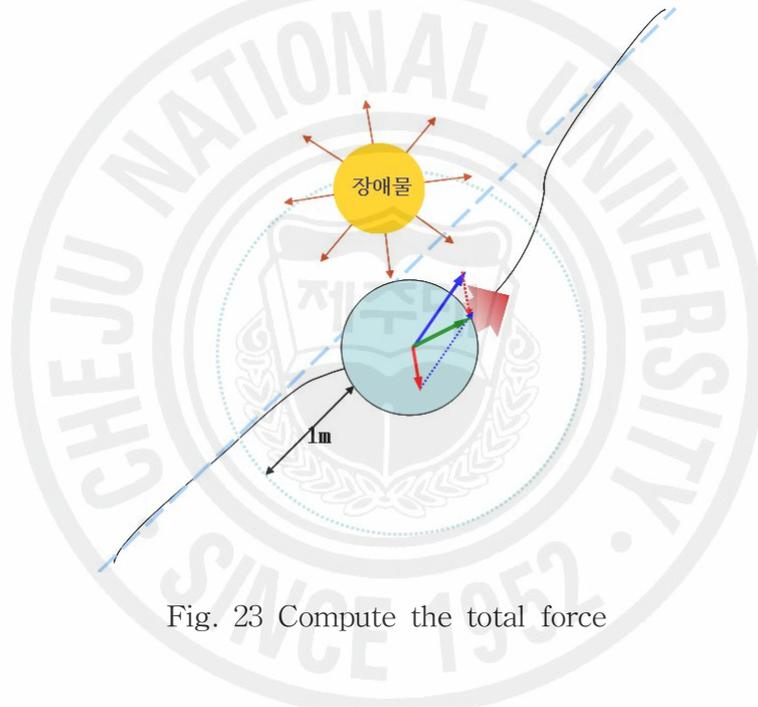


Fig. 23 Compute the total force

Fig. 23은 내부 힘과 외부 힘의 작용에 의한 궤적의 변형을 일으키는 모습을 보여주고 있다.

모바일은 다음과 같이 계획된 궤적  $j, j+1$ 과의 이동 값과 앞서 계산된 최종 로봇에 작용하는 힘과의 합으로서 회피 이동 좌표 값이 업데이트 된다.

$$\begin{aligned} \Delta x &= ({}^{j+1}x_i - {}^jx_i) + f_{x,i} \\ \Delta y &= ({}^{j+1}y_i - {}^jy_i) + f_{y,i} \\ \Delta z &= ({}^{j+1}z_i - {}^jz_i) + f_{z,i} \end{aligned} \quad (36)$$

$x_{cur,i}$ ,  $y_{cur,i}$ ,  $z_{cur,i}$ 가 현재 로봇의 위치라면 최종적으로 모바일을 이동해야 할 좌표는 다음 식(37)과 같다.

$$\begin{aligned}x_{update,i} &= \Delta x + x_{cur,i} \\y_{update,i} &= \Delta y + y_{cur,i} \\z_{update,i} &= \Delta z + z_{cur,i}\end{aligned}\quad (37)$$

모바일의 이동과 동시에 로봇의 방향을 결정해야 한다. 모바일의 이동 방향 결정은 다음 (38)식에 의하여 로봇 방향이 결정된다.

$$\begin{aligned}\Delta\theta_m &= \arctan2\left(\frac{\Delta y}{\Delta x}\right) \\ \theta_m &= \theta_{update,m} = \Delta\theta_m + \theta_m\end{aligned}\quad (38)$$

머니플레이터의 조인트 회전식은 모바일과는 다르게 회전 및 이동에 관한 토크 값을 식(39)에서 자코비안 행렬과의 계산에 의하여 최종 조인트에서의 회전 운동 식을 얻어 낼 수 있다. 머니플레이터의 제어 점  $P_1, P_1, \dots, P_7$ 에 작용하는 힘  $F_i = [f_x \ f_y \ f_z \ m_x \ m_y \ m_z]^T$ 을 식(24)에서 계산된 자코비안 행렬과 계산으로서 각 조인트에 주어지는 토크 값  $\Gamma_i = [\tau_1 \ \tau_2 \ \dots \ \tau_7]^T$ 이 결정된다.

이때 주의해야 할 점은 각 조인트에 토크 값은  $P_1$ 에서 계산된  $F_1$ 에 의하여 머니플레이터의 각 조인트의 정적 평형을 위하여 계산된 토크 값이라는 것이다. 이  $F_1$ 와 같이 나머지  $F_2, F_3, \dots, F_7$ 의 값을 계산하여 얻어진 토크 값은  $\Gamma_2, \Gamma_3, \dots, \Gamma_7$ 이 되며 최종적으로 조인트에서 받는 토크  $\Gamma = [\tau_1 \ \tau_2 \ \dots \ \tau_7]^T$ 는 식 (40)과 같다.

$$\begin{aligned}\tau_i^j &= J_i^T F_i \\ \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \end{pmatrix} &= \begin{pmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & \ddots & & & & J_{26} \\ J_{31} & \dots & \dots & & & J_{36} \\ J_{41} & & & & & J_{46} \\ J_{51} & \dots & \dots & & & J_{56} \\ J_{61} & 0 & \ddots & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} F_{i,x} \\ F_{i,y} \\ F_{i,z} \\ m_{i,x} \\ m_{i,y} \\ m_{i,z} \end{pmatrix}\end{aligned}\quad (39)$$

$$\Gamma_{[7 \times 1]} = \sum J_7^T F_7 = \Gamma_1 + \Gamma_2 + \Gamma_3 + \dots + \Gamma_7, \quad i = 1, 2, 3, \dots, 7 \quad (40)$$

토크 값에 의한 각 조인트가 움직여야 할 각도의 성분은 토크 값의 적절한 계수 값에 의하여 계산이 된다.

머니플레이터의 최종적으로 안전하게 목표 위치까지 도달하기 위해서는 제어 점에 관한 힘과 토크 값의 변형 정도가 결정이 되어야 한다. 다음과 같이  $\alpha_i$  값에 의하여 샘플링 시간마다 제어 점의 회전 및 병진이동의 정도가 결정된다[17].

$$\delta\theta_i = \alpha_i \tau_i, \quad (i = 1, 2, \dots, 7) \quad (41)$$

$\alpha_i$ 는  $i$ 번째 조인트의 토크  $\Gamma_i$ 를 위한 회전의 비례계수이며  $\delta\theta_i$ 는 샘플링 시간마다 이동해야 할  $\theta_i$  값이다.  $\alpha_i$  값이 커지게 되면  $\delta\theta_i$  값이 커지게 되어 그만큼의 매 샘플링 시간마다 빠른 조인트 변환이 가능하다.

## V. 시뮬레이션

본 장에서는 기존에 제시된 Elastic Force를 이용하여 모바일부 및 3D시뮬레이션을 수행하였고, 이를 이용하여 로봇이 장애물을 안전하게 회피하도록 하는 적절한 게인(gain)값에 대한 이동 궤적에 대해 고찰한다.

장애물 회피 주행 실험을 위해서 Matlab을 이용하며 다음과 같이 두 가지의 경우로 시뮬레이션이 수행된다.

첫 번째는 모바일로봇에 관한 2D 시뮬레이션 주행실험을 한다.  $k_r$ 값의 변화에 따른  $k_c$ 값을 달리 주어서 적절한  $k_r$ ,  $k_c$ 의 값과 그 값에 따른 수정된 궤적의 변화를 알아보려고 한다.

두 번째 실험으로는 로봇이 목표 궤적 이동 수행 시 장애물 접근에 따른 모바일과 머니플레이터의 연동 수행 및 주행제어구조에 따른 주행제어 컴포넌트가 정확히 동작하는 지를 확인한다.

### 1. 모바일부 시뮬레이션

모바일 로봇이 주어진 초기 위치에서 중간 장애물을 피해 목표지점으로 가는 실험이다. 이때  $k_r$ ,  $k_c$ 의 변화에 따라 내부, 외부 힘의 작용이 어떠한지 고찰하며 가장 적절한  $k_r$ ,  $k_c$ 의 값을 파악하고자 한다.

실험의 초기 조건은 다음과 같다.

- 초기 위치: (-3000, -3000)
- 목적 위치: (6000, 6000)
- 초기 로봇의 방향:  $30^\circ$
- 장애물의 위치: ① (3600, 1000) ② (-500, 1000)
- 안전거리: 1500mm

모바일에 관한 실험은 다음과 같이 3가지로 수행한다.

**Case 1.** 모바일 궤적의 이동 방향에 따른 적절한  $k_r$ ,  $k_c$  값을 정하도록 한다.

- ① 모바일 이동궤적:  $\pi/6$
- ② 모바일 이동궤적:  $\pi/3$

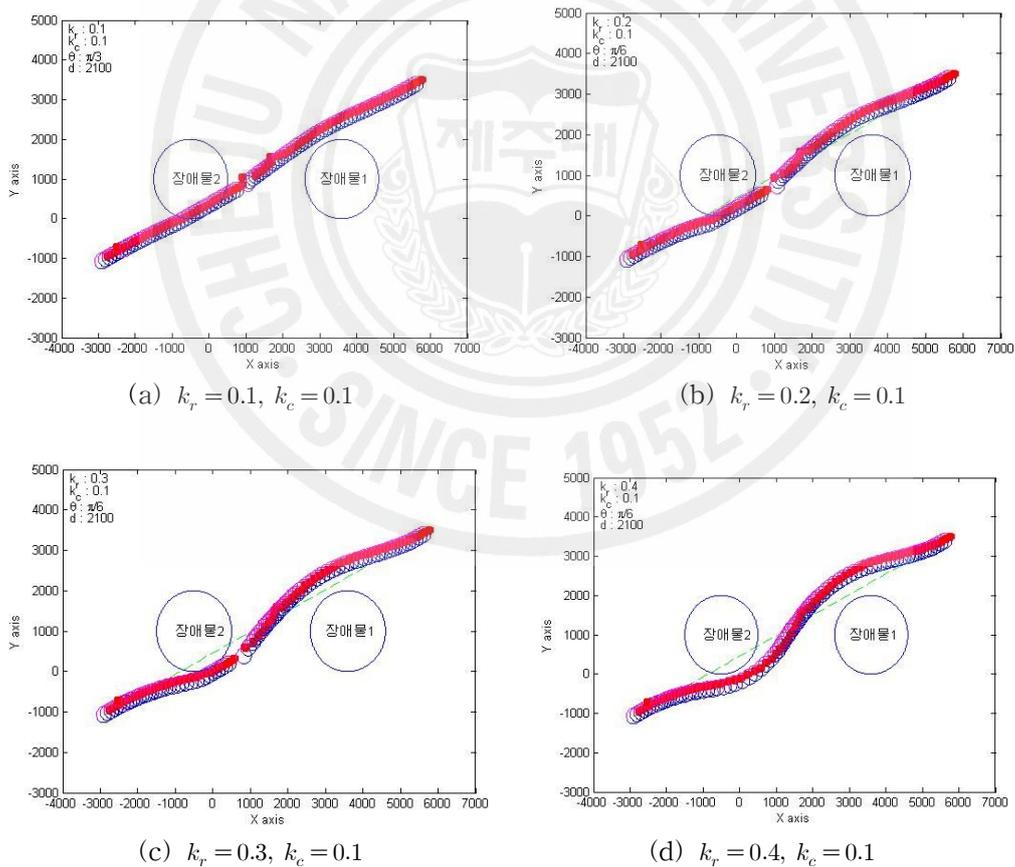
**Case 2.** 장애물간의 거리에 따른  $k_r$ ,  $k_c$  값을 정하도록 한다.

- ① 장애물 간의 간격: 1000mm
- ② 장애물 간의 간격: 100mm

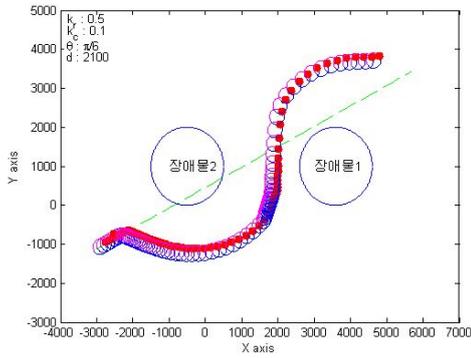
**Case 3.** 불규칙적인 장애물이 있을 때

**Case 1.**

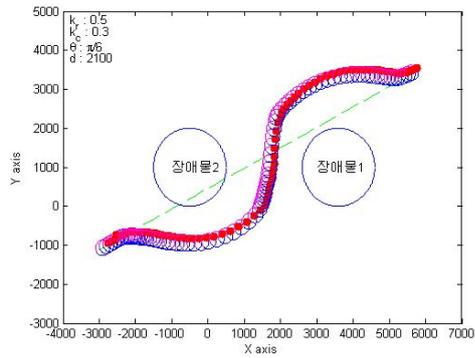
- ① 모바일 이동궤적:  $\pi/6$



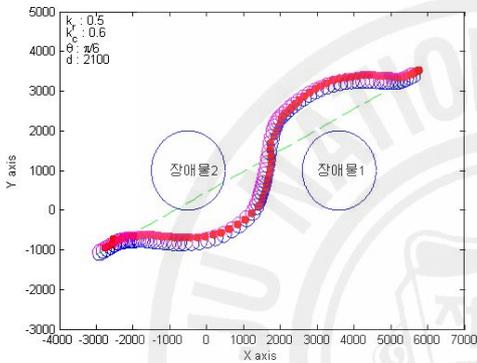
**Fig. 24** Simulations in case of  $k_c = 0.1$ ,  $k_r = 0.1, 0.2, 0.3, 0.4$



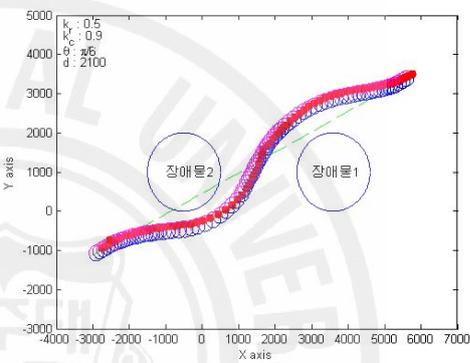
(a)  $k_r = 0.5, k_c = 0.1$



(b)  $k_r = 0.5, k_c = 0.3$

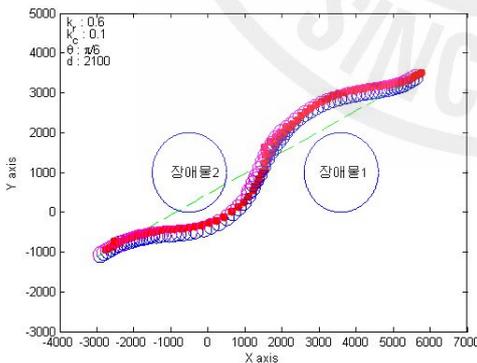


(c)  $k_r = 0.5, k_c = 0.6$

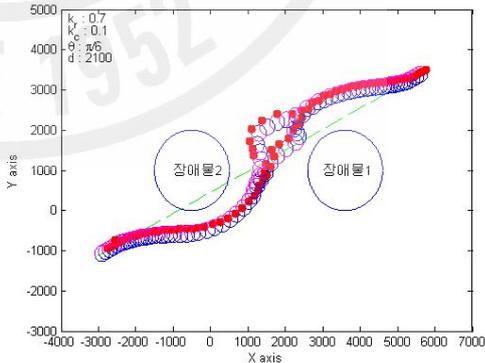


(d)  $k_r = 0.5, k_c = 0.9$

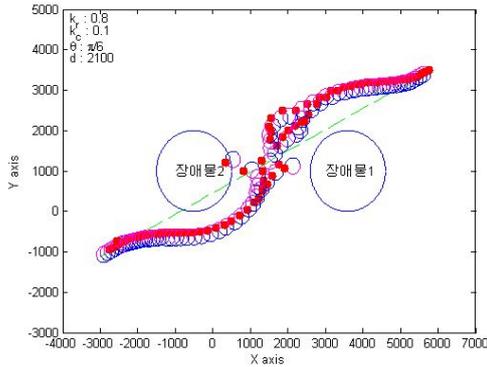
**Fig. 25** Simulations in case of  $k_c = 0.1, 0.3, 0.6, 0.9, k_r = 0.5$



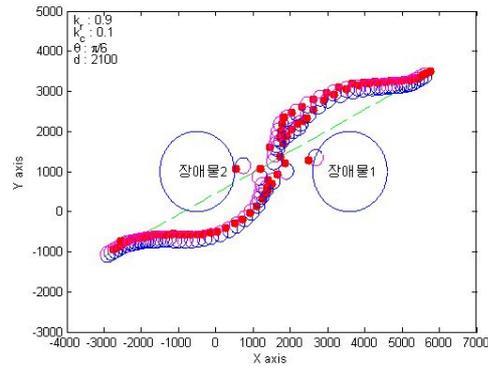
(a)  $k_r = 0.6, k_c = 0.1$



(b)  $k_r = 0.7, k_c = 0.1$



(c)  $k_r = 0.8, k_c = 0.1$



(d)  $k_r = 0.9, k_c = 0.1$

**Fig. 26** Simulations in case of  $k_c = 0.1, k_r = 0.6, 0.7, 0.8, 0.9$

모바일 이동궤적이  $\pi/6$ 일 경우 실험에서 Fig.24의 (a), (b), (c)는  $k_r$ 값이 증가하면서 서서의 장애물과 멀어지고 있지만 장애물과의 거리에 비해 장애물이 밀어내는 힘이 적게 작용하여 장애물과 충돌하고 있다. Fig.24의 (d)는 장애물을 피하고 있으나 실제 로봇 주행 상황을 고려했을 경우 충돌의 여지가 보인다.

Fig.25는 모바일이 장애물과 충돌 없이 주행하고 있는 경우를 보여주고 있다. 이 경우에서 보는 것과 같이  $k_r$  값은 0.5로 가장 안정된 개인 값으로 정할 수 있으며  $k_r$ 의 0.5 값에 따른 (a),(b),(c),(d)으로  $k_c$ 값이 증가함에 따라 장애물 주위로 근접해 가고 있는 것을 알 수 있다. 이는  $k_c$ 값의 커짐에 따라 로봇에 작용하는 내부 힘 즉 수축하려는 힘이 외부 힘보다 크게 작용하기 때문에 앞서 계획된 궤적으로 돌아가려는 경우를 보여주고 있다. 경우(a)는 내부 힘에 비하여 큰 외부 힘이 작용함에 따라 미처 목표궤적을 찾지 못하고 로봇이 정지해 있는 것을 확인할 수 있다. 경우(b),(c),(d)는 모두 주행에는 문제가 없으나 최적의 주행 상태를 고려한다면 불필요한 주행을 보이는 (b)에 비해 경우 (c)가 가장 적절한 이동을 보여주고 있다고 볼 수 있다.

Fig. 26의 경우(a)는 장애물 충돌 회피 및 목적지 도착에는 문제가 없으나 장애물 사이에 약간의 떨림현상이 발생하고 있는 경우이다.  $k_r$ 값이 0.6 이상

증가하면서 (b),(c),(d)에서 실제 로봇이 움직이기 힘들 정도로 움직임이 발생하는 것을 관찰 할 수 있다. 이는  $k_r$  값이 필요이상 작용함에 따라 두 개의 장애물 사이에 양 방향에서 로봇을 밀어내는 외부 힘이 강하게 작용하므로 중간 경로를 찾지 못하여 떨고 있는 지역 극소 점(local minima)에 빠진 것을 보여주고 있다.

② 모바일 이동체적:  $\pi/3$

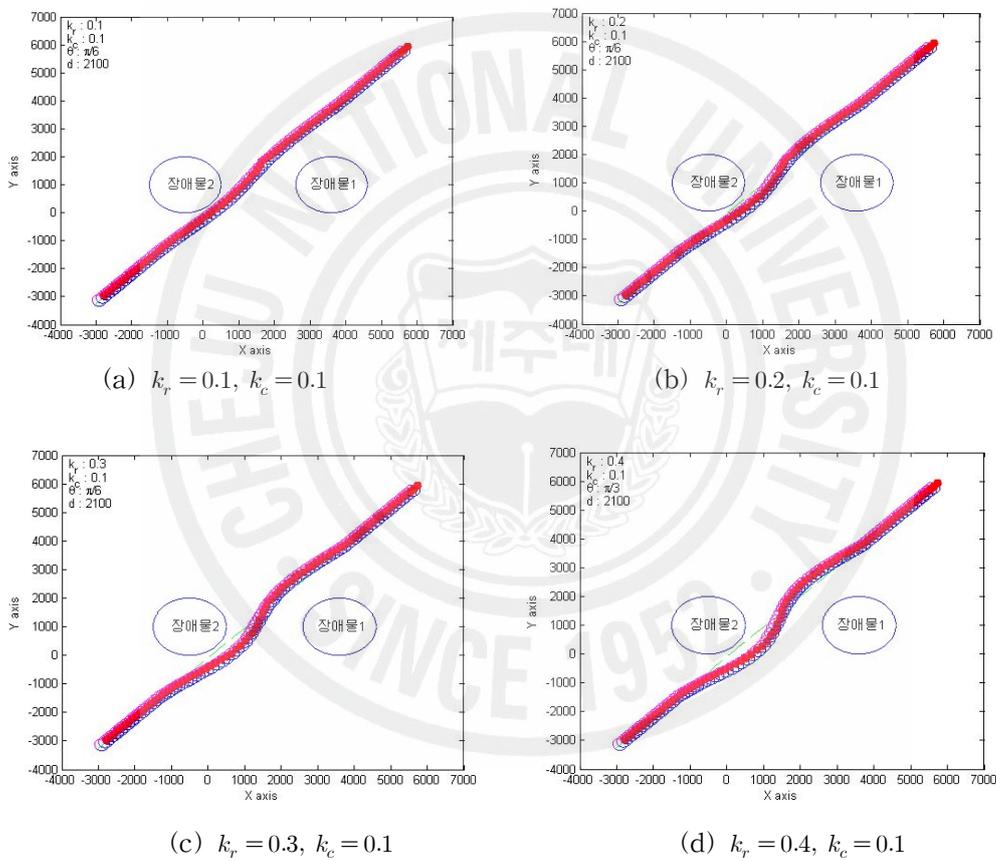
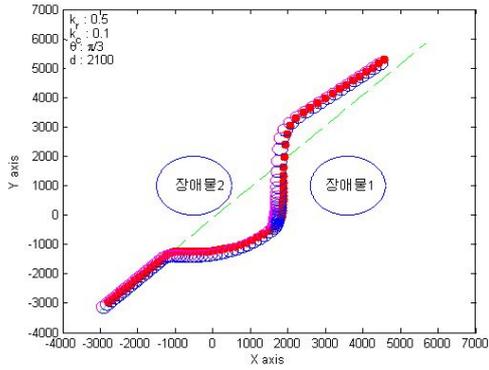
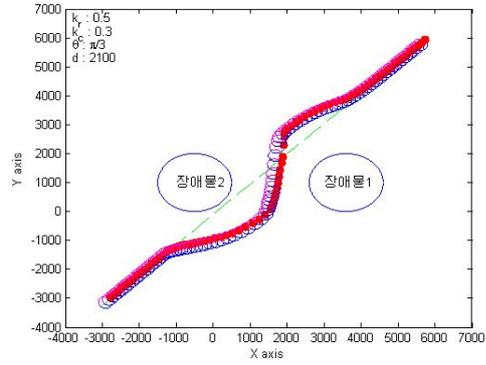


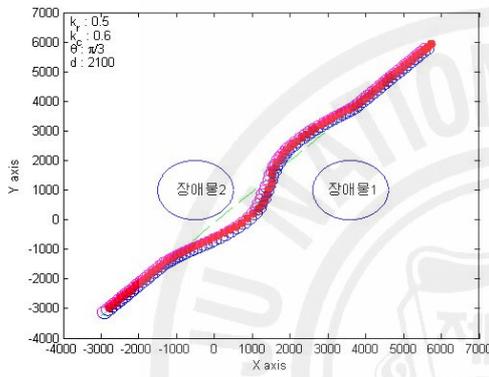
Fig. 27 Simulations in case of  $k_c = 0.1, k_r = 0.1, 0.2, 0.3, 0.4$



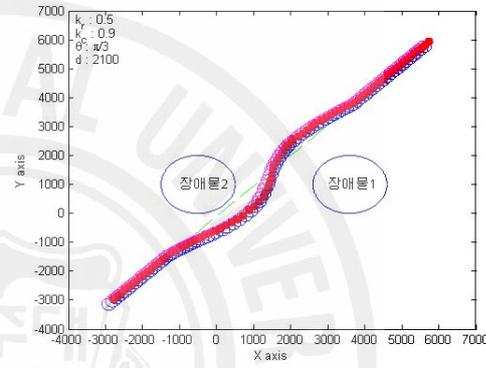
(a)  $k_r = 0.5, k_c = 0.1$



(b)  $k_r = 0.5, k_c = 0.3$

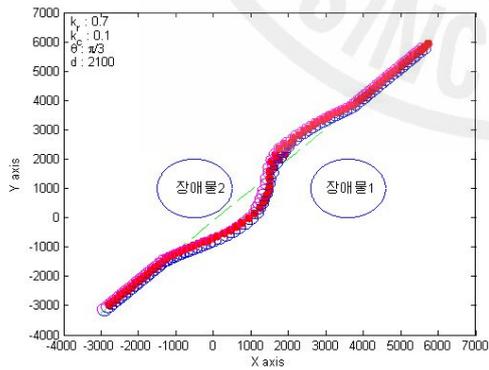


(c)  $k_r = 0.5, k_c = 0.6$

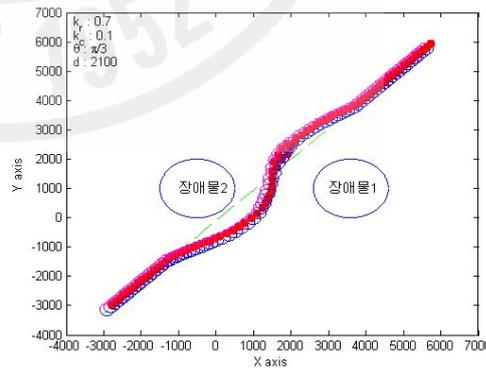


(d)  $k_r = 0.5, k_c = 0.9$

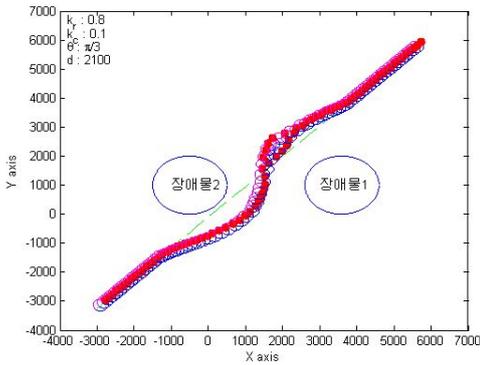
**Fig. 28** Simulations in case of  $k_c = 0.1, 0.3, 0.9, k_r = 0.5$



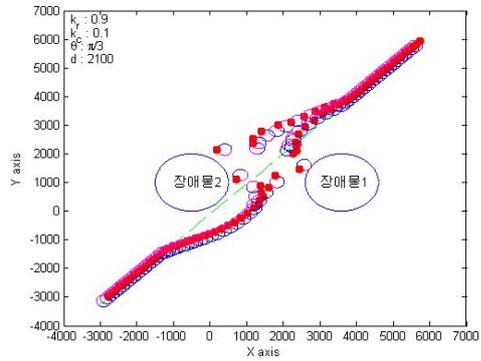
(a)  $k_r = 0.6, k_c = 0.1$



(b)  $k_r = 0.7, k_c = 0.1$



(c)  $k_r = 0.8, k_c = 0.1$



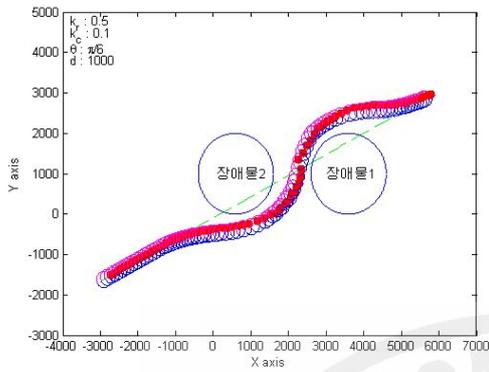
(d)  $k_r = 0.9, k_c = 0.1$

**Fig. 29** Simulations in case of  $k_c = 0.1, k_r = 0.6, 0.7, 0.8, 0.9$

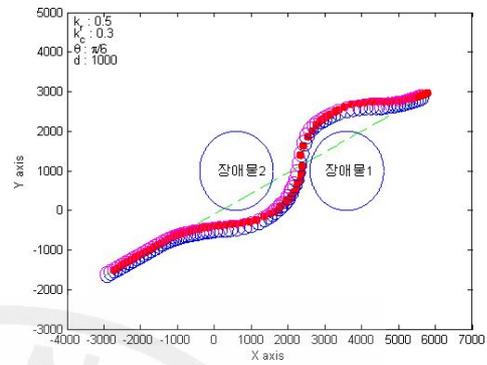
모바일 이동궤적이  $\pi/3$  경우는 앞서  $\pi/6$ 와의 실험과 크게 다르지 않게 주행을 하고 있는 결과를 보여주고 있다. Fig.27의 경우(a),(b),(c),(d)를 보면  $k_r$  값의 증가에 따라 큰 외부 힘의 작용으로 장애물과 적정 거리를 두면서 이동을 하였지만 적은 내부 힘, 외부 힘의 영향으로 목표 궤적과 별 차이 없이 장애물을 통과하고 있다. Fig.28에서 역시  $k_r=0.5$ 에서 가장 안정된 장애물 회피 주행을 보여주고 있으며 경우 (a),(b),(c),(d)에 따라 내부 힘이 서서히 크게 작용하여 장애물로 인한 궤적의 이탈없이 적절한 거리를 유지하며 회피하고 있다. Fig.29에서와 같이  $k_r$  값의 증가에 따라 동시에 외부 힘 커짐으로서 0.7를 지나면서부터 지역 극소점에 빠져 불안정한 주행을 하는 것을 관찰할 수 있다.

Case 2.

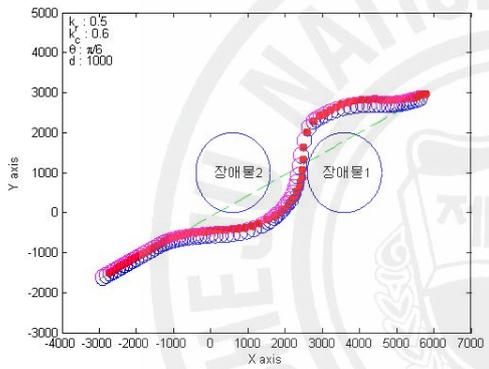
① 장애물 사이의 간격: 1000mm



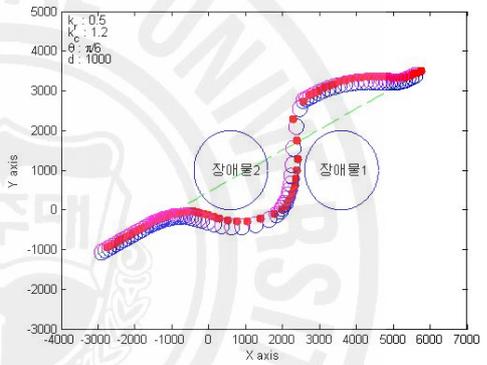
(a)  $k_r = 0.5, k_c = 0.1$



(b)  $k_r = 0.5, k_c = 0.3$



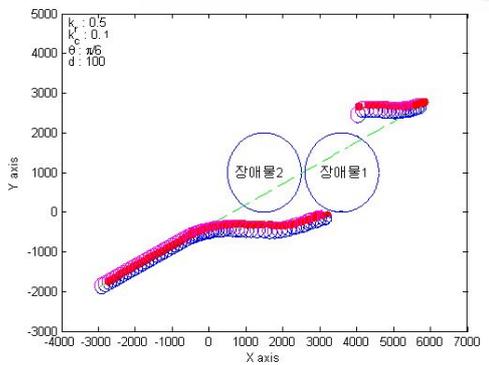
(c)  $k_r = 0.5, k_c = 0.6$



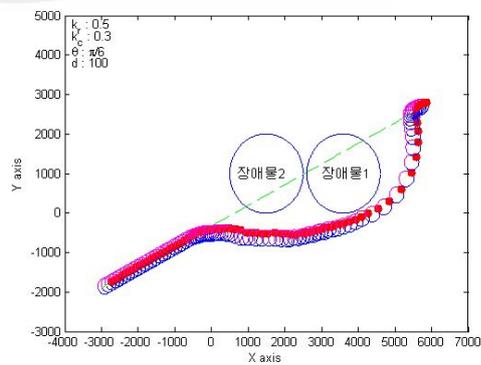
(d)  $k_r = 0.5, k_c = 1.2$

Fig. 30 Simulations in case of  $k_c = 0.1, 0.3, 0.6, 1.2, k_r = 0.5$

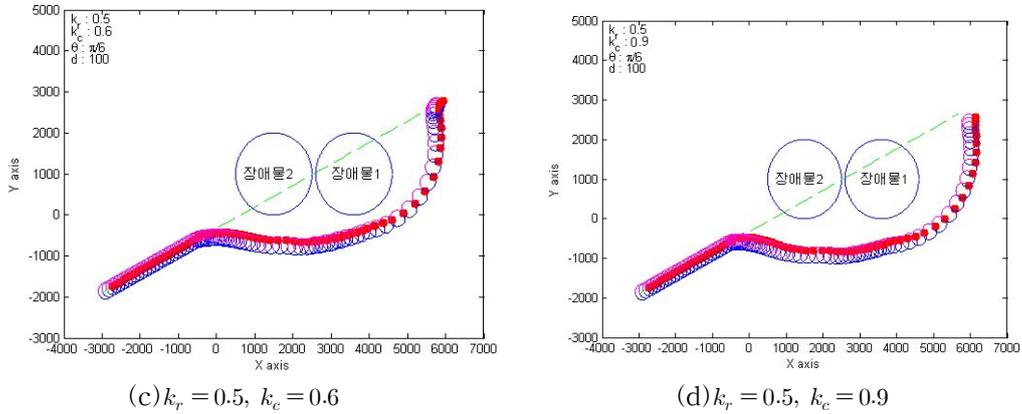
② 장애물 사이의 간격: 100mm



(a)  $k_r = 0.5, k_c = 0.1$



(b)  $k_r = 0.5, k_c = 0.3$



**Fig. 31** Simulations in case of  $k_c = 0.1, 0.3, 0.6, 0.9, k_r = 0.5$

Fig.30는 모바일이 장애물 사이를 지나갈만한 간격이 주어 졌기 때문에 앞서 실험과 별 차이 없이  $k_r=0.5, k_c=0.3$ 값에서 가장 안정된 회피를 보여 주고 있다. 하지만 Fig.31의 경우 모바일이 지나 갈수 없을 정도의 간격 때문에 경우 (b), (c), (d)에서 모바일로봇이 두 장애물을 하나의 장애물로 인식하여 목표지점 까지 문제없이 도착하고 있다. Fig.31의 경우(a)는 외부 힘이 적게 작용하여 장애물과 충돌하는 경우에 해당한다.

Case. 3 불규칙적인 장애물이 있을 때

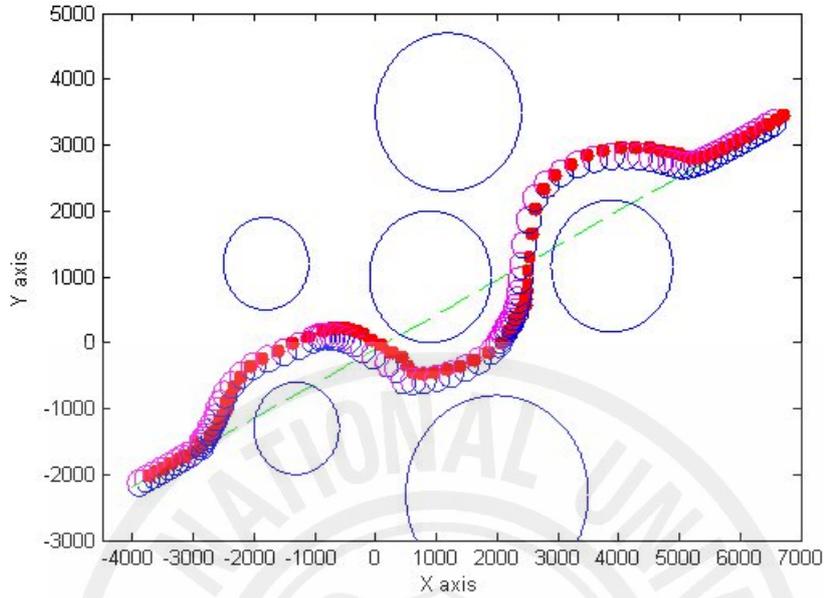


Fig. 32 Simulation when the position of obstacle is randomly placed

Fig.32는 앞서 시뮬레이션을 바탕으로 선택된 계인 값  $k_r=0.5$ ,  $k_c=0.4$ 인 경우에 임의의 장애물이 있는 경우 문제없이 내부 힘과 외부 힘의 적절한 조화를 이루며 장애물을 회피하는 모습을 보여주고 있다.

## 2. 3D 시뮬레이션

모바일 머니플레이터 로봇이 목적 위치까지 주행동안 모바일과 머니플레이터 시스템의 연동된 안전한 장애물 회피가 일어나는지 확인과 동시에 본 논문에서 제안한 주행제어구조가 문제없이 동작하는지를 확인한다.

실험의 초기 조건은 다음과 같다.

- 모바일 로봇의 초기 및 목표위치:  $(-3000, -1500), (6000, 3000)$
- 모바일 로봇의 초기 방향:  $30^\circ$
- 머니플레이터 초기 형상:  $q_i = [18 \ 90 \ 0 \ -90 \ 0 \ -90 \ 0]$
- 머니플레이터 목표 형상:  $q_i = [90 \ 90 \ 0 \ -60 \ 0 \ -30 \ 0]$
- 장애물의 위치: ①  $(3600, 1000)$  ②  $(-500, 1500)$
- 안전거리 및 계인 값:  $1000\text{mm}, k_c: 0.5, k_r: 0.3$

다음 실험은 모바일과 머니플레이터의 연동에 의한 장애물 회피에 관한 실험이다. 실제 로봇 시스템을 위하여 모바일의 LRF 센서 위치에서 앞·뒤로  $180^\circ$  장애물에 대해  $5^\circ$  간격으로 72개의 센싱정보를 획득 하였으며 머니플레이터의 장애물 센싱에 대해서는 앞으로 비전센서를 이용하여 거리 정보를 얻어올 것이므로 여기서는 각 조인트에서 장애물간의 거리를 직접 계산하였다.

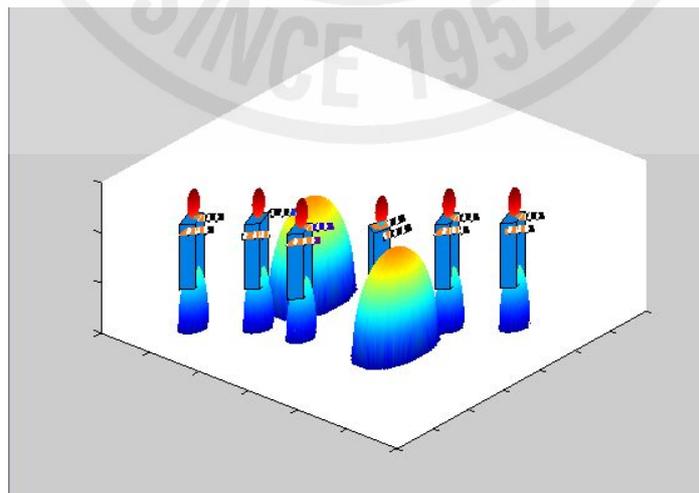


Fig. 33 3D Mobile-manipulator simulation

Fig. 33 에서 3D 시뮬레이션을 통하여 확인한 바와 같이 본 논문에서 제안한 제어구조의 반응 층에서 실시간 회피모듈의 실제 적용이 가능함을 확인하였다. 또한, 실제 모바일 머니플레이터 적용을 고려할 때, IR, Sonar, LRF센서를 이용하여 장애물과의 거리 및 방향을 측정하고 엔코더 정보와 위치 추정기(localizer)로 로봇의 위치를 검출한다면 단위행위들이 실제 적용이 가능함을 시뮬레이션을 통하여 확인되었다.



## V. 결론

본 논문에서는 모바일-머니플레이터의 자율기능을 효율적으로 부여하기 위하여 로봇제어구조를 제안하였으며, 이동성 및 작업성의 효율을 향상시키기 위한 방법으로서 로봇의 기구학 특성에 맞는 장애물 회피 기능을 설계하였다. 이는 시뮬레이션을 통하여 이동성 및 안전성을 검증하였다. 이러한 연구를 통해 다음과 같은 성과를 얻을 수 있었다.

**로봇제어 구조의 설계:** 자율기능을 부여하기 위한 틀로서 세 개의 계층으로 나누었으며 Reactive층에서 장애물 회피 기능을 모듈화 시킴으로써 각각의 컴포넌트들이 실시간으로 동작됨을 확인하였다.

**모바일-머니플레이터의 기구학 해석:** 지능로봇의 모바일 및 여유자유도 머니플레이터의 하드웨어적 특성에 맞는 기구학을 해석하였으며 시뮬레이션 결과로서 모바일-머니플레이터가 안정적으로 연동하여 동작함을 확인하였다.

**장애물 인식 알고리즘 개발:** Elastic Strips를 이용하여 여유자유도를 갖는 지능로봇 기구학 특성에 적합한 알고리즘을 개발하였으며 실시간 구현 가능한 충돌회피(collision avoidance)모듈로 설계하였다. 결과적으로 기존 알고리즘에서와 달리 목적위치에 잘 도착하는 것을 확인하였다. 이는 장애물 회피 외에도 다양한 분야에서 적용가능 할 것으로 예상된다.

본 연구에서 출발하여 추가적으로 도출될 수 있는 연구 및 현 연구 단계에서의 보완점과 추후 진행할 연구내용을 아래와 같이 정리하였다.

**주행제어구조의 실시간 컴포넌트 확장:** 환경지도 작성, 자기위치 인식 및 경로생성과 같은 복잡한 알고리즘도 주행제어구조의 틀 안에 포함시켜 실시간 특성 및 이동 효율성을 증가하기 위함이다.

**장애물 인식 및 회피 알고리즘 보완:** 현재 장애물을 인식하는 방법은 LRF

센서 하나로 장애물의 인식만 고려한 알고리즘인데, 실제로 센서의 부정확한 값을 고려하여, IR, Sonar, LRF, Vision센서의 융합을 통해 복잡한 인간생활 환경에서도 강인하게 주어진 임무를 수행함과 동시에 장애물을 회피할 수 있는 알고리즘 보완 연구가 필요하다.



## VI. 참고문헌

- [1] 고낙용, 김재열, 1998. “이동로봇의 실시간 장애물 회피를 위한 새로운 방법,” 한국공작기계학회지, 제7권, 제4호, pp. 28-34
- [2] 손원중, 홍경태, 홍금식, 2006., “LRF 센서를 이용한 장애물 탐지 및 회피 알고리즘”, 대한기계학회 춘계 학술대회, pp. 3015-3020
- [3] 윤도영, 오상록, 공성학, 서일홍, 박귀태, 2000., “자율로봇을 위한 Reactive-Deliberative 하이브리드 제어구조 설계”, 제어계측/자동화/로보틱스연구회 합동학술발표회 논문집, pp. 63-67.
- [4] 전성용, 2006. 2, “자율이동로봇의 실시간 주행을 위한 반사층의 제어구조”, 부산대학교 석사학위논문, pp. 22-35
- [5] 정식훈, 2003, “최적경로를 이용한 이동매니퓰레이터의 물체이동에 관한 연구”, 부산대학교 석사학위논문, pp. 6-7..
- [6] Arkin, R. C., and Balch, T., 1998, Behavior-Based Robotics, The MIT Press, Cambridge.
- [7] Brooks, R. A., 1986, "Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14-23
- [8] Connell, J. H., 1996, "SSS: A Hybrid Architecture Applied to Robot navigation", Proc. of the IEEE/RSJ International Conference in Intelligent Robots and Systems'96, Vol. 1, pp.1-8.
- [9] Dellaert, F., Fox, D., Burgard, W., and Thrun, S., 1999, "Monte Carlo Localization for Mobile Robots", Proceedings of IEEE International Conference on Robotics and Automation, pp. 419-424.
- [10] Dudek, G., and Jenkin, M., 2000, "Computational Principles of Mobile Robotics", Cambridge University Press.

- [11] E. Tunstel, H. Danny, T. Lippincott and M. Jamshidi, 1997, "Autonomous navigation using an adaptive hierarchy of multiple fuzzy-behaviors", Proc, IEEE international Symposium on Computational Intelligence in Robotics and Automation, pp. 276-281
- [12] Hans, M., and Baum, W., 2001, "Concept of a Hybrid Architecture for Care-O-bot", Proceedings of ROMAN-2001, pp. 407-4211.
- [13] Hans, M., Graf, B., and Schraft, R. D., 2002, "Robotic Home Assistant care-O-bot: Past-Present-Future", Proceedings of ROMAN-2002, pp. 380-385.
- [14] J. Borenstein and Y. Koren, 1991., "Real-time Obstacle Avoidance for Fast Mobile Robots", IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, No. 4, pp. 535-539
- [15] Lindstrom, M., Oreback, A., and Christensen H. I., 2000, "BERRA: A Research Architecture for Service Robots", Proceedings of the IEEE Conference on Robotics and Automation San Francisco, CA, USA, pp. 3278-3283.
- [16] Low, K. H., Leow, W. K., and Ang, M. H., 2002, "A Hybrid Mobile Robot Architecture with Integrated Planning and Control", proceedings of 1st AAMAS, Bologna, Italy, pp.219-226.
- [17] Nak yong Ko and Beom hee Lee, 1997. 4, "Moving Obstacle Avoidance of a Robot Using Avoidability Measure", Journal of Control. Automation and Systems Engineering, Vol.. 3, No. 2, pp. 169~178
- [18] O. khatic, 1986, "Real-time obstacle avoidance for manipulators and mobile robot", The Int. J. robotics Research, Vol. 8, No. 1
- [19] O.khatib, 1996, "Motion coordination and reactive control of autonomous multi-manipulator system", Journal of Robotic System, Vol. 15, No. 4, pp. 300-319
- [20] Ridas, P., Batlle, J., and Carreras, M., 2002, "a New Object Oriented

Control Architecture for Autonomy: the Reactive Layer”, Control Engineering Practice, Vol. 10, pp. 857-873.



## 감사의 글

대학생활을 마치고 대학원 진학한지가 엇그제 같은데 벌써 졸업을 앞두고 있다는 것이 믿어지지 않습니다. 2년이라는 시간은 제게 전공에 대한 적성을 의심 할 정도로 힘든 시간이었습니다. 앞만 보고 달려오다 보니 놓친 것, 잃은 것도 많은 시간이었지만 그동안에 늘 하는 일에 자신 없던 나에게 있어서 나 역시 대한민국의 한 사람으로서 나라에 보탬을 줄 수 있다는 자부심과 사회생활에서의 발판을 미리 마련할 수 있는 기간이었습니다. 이렇게 변화할 수 있게 도와주신 제 주위에 모든 분 들게 감사의 말씀을 드립니다.

가장먼저 4년이라는 긴 시간동안 제 부족한 능력을 끝까지 믿어주시어 가르침을 주신 저희 최경현 교수님께 백 마디를 해도 부족하지만 감사하단 말씀이외에 드릴 말씀이 없습니다. 교수님께서 주신 4년간의 가르침 평생 잊지 않고 가슴속에 새겨 넣고 다니겠으며 새로운 사회에서도 교수님 이름 석자 부끄럽지 않게 열심히 일을 해나가도록 하겠습니다. 그리고 그리 대단하지 않은 논문을 심사한다고 시간 내 주시고 많은 조언을 주신 조경호, 도양희 교수님, 대학 시절부터 여학생이라는 이유만으로 많은 격려를 아낌없이 주신 임종환교수님, 강철웅 교수님, 김상재 교수님께도 감사드립니다.

실험실 생활 때부터 저희 실험실의 만형(?)처럼 많은 조언을 주신 프로그램의 천재이신 양경부 오라버니, 그리고 이미 졸업하여 연락은 되지 않지만 대학원 생활동안 심심찮게 실험실에 들려주어 스트레스 해소를 해 주었던 김영근 오라버니, 팩스 보낼 때 항상 도와주었던 한종욱 오라버니에게도 감사드립니다. 그리고 수업시간 외에 자주 볼 수는 없었지만 대학원 동기이자 학과 선배이신 세분들... 오동훈, 오상율, 김병기 오라버니들이 있었기에 대학생활보다 대학원생활이 더 즐거울 수 있었습니다. 그리고 우리 과 여자 중 나와 외모 1, 2위를 다했던 소영이에게 정말 고맙습니다. 과 들어오고 나서 나 홀로 여학생 이었다면 적응 못하고 전과 생각도 했었겠지만 소영이가 있었기에 대학과 대학원을 무사히 마칠 수가 있었습니다. 그리고 우리 과 얼마 안 되는 여학생 중 총부 맡아서 바빴던 막내 세나 함께 있어주어 고맙다. 역시 대학 1학년 때부터 현재까지도 다른 과이지만 대학 동기 여학생으로서 옆에 있어주기만 해도 힘이 되어 주어주는 한유리, 김은주에게도 고맙다는 말을 올립니다.

4년이라는 긴긴한 실험실 생활동안에 싸우면서도 서로 도우면서 오랜 시간동안 지냈던 창중 오라버니, 올해 있어서 나의 짐을 함께 거들어 주고 로봇 시뮬레이션 돌리느라 고생 많았던 형찬 오라버니, 그리고 과 부회장으로서는 바빠서 얼굴 보기 힘들었던 경찬이, 그래도 일에 있어서 열심히 열정을 보이고 있는 영봉이, 방학이지만 일찍 출근하는 실험실 막내인 상호에게도 정말 고맙고 앞으로 더욱더 열심히 하는 모습 보여주길 바랍니다. 우리 실험실 인원과 못지않게 실험실 일을 같이 도맡아 항상 바쁜

일과에 몸을 사리지 않고 일하는 전자과에 김형찬, 논문 쓴다고 잠깐이나마 로봇에 대해서 같이 연구했던 김수진학생에게도 감사의 말을 전합니다. 그리고 야식 먹을 일 있음 항상 불러 주었던 옆실협실 근현이, 상섭이에게도 고마움을 전합니다.

또한 졸업은 하였어도 같은 로봇 과제 한다고 신경써서 도움을 주신 전성용, 김광 일오라버니 자신의 로봇연구 자료를 아낌없이 내놓아 주신 손원종씨, 그리고 먼 부산에서 우리 실험실의 프로젝트를 도맡아 하시는 최재원오라버니에게도 감사의 말씀을 전합니다.

마지막으로 누구보다 저를 걱정하느라 하루하루 맘 고생을 하시면서도 나의 일이 잘되라고 빌어주었던 나의 사랑하는 아버지, 어머니, 같은 방을 쓰면서도 가끔 얼굴 비추는 나를 구박하지 않고 나의 빨래와 청소를 빼먹지 않고 챙겨주었던 은심언니, 그리고 항상 나에게 건강걱정을 하며 말 한마디라도 따스하게 해주었던 은숙언니에게도 감사의 마음을 전하며 대학원 생활의 결실인 이 논문을 나의 가족에게 받칩니다.

이제 졸업을 앞두고 사회를 나가는 문턱에 서니 설렘과 두려움이 앞섭니다. 4년이라는 시간을 제 집과 같이 실험실에서 지냈기에 앞으로 이곳이 그리울것 같습니다. 하지만 한곳에 정착하지 않고 후회 없는 삶을 살기위해 하루하루 노력하는 사람이 되겠습니다.

2007년 새로운 한 해를 맞이하며  
조 수 정