工 學 碩 士 學 位 論 文

# 다수 이동 로봇을 위한 협동 알고리즘과 경로 계획에 관한 연구

2002년 2월

釜慶大學校 大學院

機 械 工 學 科

洪 成 雨

工學碩士 學位論文

# 다수 이동 로봇을 위한 협동 알고리즘과 경로 계획

指導敎授 安 斗 星

이 論文을 工學碩士 學位論文으로 提出함

2002年 2月

釜慶大學校 大學院

機 械 工 學 科

洪 成 雨

洪成雨의 工學碩士 學位論文을 認准함

2001年 12月 26日

主　　審　　工學博士　　白 雲 景　印

委　　員　　工學博士　　安 斗 星　印

委　　員　　工學博士　　權 純 在　印

# 목차

Abstract

제 1 장 서 론 ..... 1
1.1 기존 연구 고찰 및 연구 배경 ..... 1
1.2 연구 내용 및 구성 ..... 3
제 2 장 로봇축구 시스템 ..... 5
2.1 시스템 제작 ..... 5
2.1.1 로봇 제작 ..... 5
2.1.2 마이크로 컨트롤러 ..... 6
2.1.3 모터부 ..... 8
2.1.4 통신부 ..... 9
2.1.5 맨체스터 코드 방법 ..... 11
2.1.6 영상 처리 시스템 ..... 11
2.2 경로 궤적 계획의 필요성 ..... 13
2.3 협동 알고리즘의 필요성 ..... 17
제 3 장 경로 계획 ..... 19
3.1 경로 계획 ..... 19
3.1.1 단위 벡터장법 ..... 19
3.1.2 동심원 도법을 이용한 경로 계획 ..... 21
3.1.3 퍼지 논리 이용 경로 계획법 ..... 22
3.2 변형된 경로 계획법 ..... 23
3.2.1 동심원을 이용한 경로 계획 ..... 24
3.2.2 퍼지 논리를 이용한 원호 경로 계획법 ..... 27
3.2.3 퍼지 논리를 이용한 경로 추종기 ..... 30
3.2.4 퍼지 논리를 이용한 직선 경로 계획 및 추종기 ..... 33
3.2.5 공이 움직일 경우 경로 계획 ..... 34
3.2.6 슛팅 이외의 경우 ..... 35
3.2.7 골키퍼 경로 계획 ..... 37
3.3 실험 ..... 38
제 4 장 협동 알고리즘 ..... 41
4.1 강화 학습 ..... 41
4.2 용어 ..... 42
4.3 강화 학습의 최적화 모델 ..... 44
4.3.1 Finite-horizon model ..... 44
4.3.2. Infinite-horizon model ..... 45
4.3.3 Average reward model ..... 45
4.4 Q-Learning ..... 45

4.5 역할 분할 알고리즘 ..... 47
4.6 Q-학습에 의한 행동 결정 ..... 47
4.6.1 슛팅 Q-net 구조 ..... 48
4.6.2. 수비 Q-net 구조 ..... 50
4.6.3 패스 Q-net 구조 ..... 51
4.7 전체 제어기 ..... 51
4.8 실험 ..... 52
제 5 장 결 론 ..... 55
참고 문헌 ..... 56
부 록 ..... 59
A. 마이크로 컨트롤러 보드 회로도 ..... 59
B. 모터 구동부 회로도 ..... 60
C. RF통신부 회로도 ..... 60
D. DC 스위스 미니모터 2224 사양 ..... 62

# A Study on the Path Planning and the Cooperation Algorithm for Multi Mobile Robot

*Seong-Woo Hong*

*School of Mechnical Engineering Graduate School*

*Pukyong National University, Busan, Korea*

## ABSTRACT

In this paper, we construct the robot soccer system. Also, we investigate the real time path planning and the action selection strategy of each robot. The system is one of the multi-agent robot systems. Among many works of the system, the most important things are about the motion of each robot and about the action selection strategy. Because each robot is free-ranging mobile robot, it is needed to make the path and to choose a proper action.

In path planning, we suggest three terms. One is to minimize the total distance or time, another is to determine easily, and the last one is to avoid obstacles.

In cooperation behavior algorithm, we propose a reinforcement learning-based method is applied to realize the cooperation of mobile robots. Q-learning, which is one of the popular methods for reinforcement learning, is used to determine what actions to take.

We show result through the simulation and real experiment.

# 제 1 장 서 론

인공 지능에 관계된 분야들은 수십 년간 많은 발전을  이루어 왔다. 그 중 컴퓨터 비전에서는 인간의 시각과 인식 기능을 Machine Learning에서는 인간의 추론, 학습 능력을 컴퓨터를 이용하여 구현하기 위한 노력을 기울여 왔고 근래에 와서는 사람의 얼굴을 인식하는 시스템이나 장기를 두는 시스템 등이 개발되고 있다. 이러한 분야들이 비교적 역사가 오래된 제어나 기계공학 등의 학문과 접목하여 감각, 인식, 사고, 행동 등을 유기적으로 결합시킨 자율 이동 로봇이 등장하였고, 그 중의 한 예가 여러 개의 소형로봇의 협동 작업을 통해 하나의 작업을 수행하는 다중 이동 로봇 시스템이다. 본 논문에서는 다중 이동 로봇의 한 종류인 로봇축구[1][2][3][4]를 구현한다. 로봇 축구 시스템을 구현하기 위해서는 다수 로봇의 운용 알고리즘, 로봇의 전기 기계적인 기술,  카메라에서 들어온 화상을 빠른 시간 내 처리하기 위한 소프트웨어, 로봇 통신을 위한 통신 기술 등 광범위한 기술을 집약해야 한다. 그래서 이 시스템을 개발함으로써 기술적 효과는 매우 클 것 이다.

이 논문에서는 로봇축구 시스템을 구축하고 가장 효과적인 경로 계획과 협동 알고리즘 구현을 목표로 한다.

## 1.1 기존 연구 고찰 및 연구 배경

다중 이동 로봇에 대한 연구는 하나의 완전한 로봇이 작업을 실행하기에는 어렵거나 로봇의 고장이 일어 났을 때 전 시스템에 끼치는 영향과 비용, 시간이 낭비된다는 것을 알게 되면서 그에 대한 대안으로 시작되었다. 작고 비용이 적게 드는 로봇을 여러 개 만들어 어렵을 작업을 서로 협동을 이용하여 지능이 뛰어난 하나의 로봇과 같은 효과를 내고 또한 몇 대의 로봇이 이상이 생기더라도 나머지 로봇으로 인하여 시스템 전체에 끼치는 영향을 최소화 할 수 있으며 그 비용 또

한 절감 효과를 가져 온다[13][24]. 로봇 축구와 다중 이동 로봇은 약간의 차이가 있다. 다중 이동 로봇은 단순히 하나의 작업만을 수행하면 서로 협조만을 수행하는 반면에 로봇 축구에서는 상대 로봇까지 고려하여 경쟁이라는 의미를 고려해야 한다. 따라서 좀더 복잡한 환경에 대한 고려가 이루어 져야 한다. 로봇 축구에 대한 연구를 내용에 따라 분류를 하면, 실제 경기 출전을 위한 시스템 구성에 주안점을 둔 연구[1]와 로봇의 제어에 대한 관점에서 학습의 개념을 포함시킨 연구[2][25] 및 시뮬레이션[3]에 관한 연구 등이 주종을 이루고 있으며 경기를 하는 방식이 MIROSOT과는 크게 다르지만 강화학습 방법을 이용하여 학습을 시도한 연구[4][8][9][12][15]도 이루어지고 있다.

로봇축구는 여러 대의 로봇이 동시에 서로 관계를 가지고 동일한 목표 하에 작동한다는 것이 큰 특징이다. 그러나 지금까지의 연구 중에서는 로봇의 협동 작업의 개념과 관련된 내용은 많지 않고 고전적인 if-then-else 구조의 판단 트리에 의한 접근방식이 주로 제시[2][23]되어왔었다. 최근에는 상황에 따라 적절한 전술을 구사할 수 있도록 학습을 통하여 행동결정을 이루어내고 경험에 의한 시행착오를 학습하는 강화 학습을 사용하여 협력 알고리즘을 제시하는 연구가 많아 지고 있다.

또한, 이동 로봇은 기본적으로 주위 환경을 인식하여 경로를 계획하고, 계획된 경로의 추종 할 수 있는 기능을 갖추어야 한다. 경로를 추종하는 방법으로 현재 실용화 되어 있는 방법 중에는 고정 궤도를 미리 바닥에 설치하고 전자기 또는 광학적 처리를 통해 경로를 주행하도록 하는 방법 등이 있는데 이런 방식은 항상 고정된 궤도를 감지 할 수 있어서 제어가 간단하고 추종 성능이 우수하지만, 경로를 설치 및 유지해야 하므로 시간적 경제적 투자가 필요하고 또 작업 경로가 바뀔 경우 전체적인 변경을 요하는 단점이 있다. 그러므로 원하는 경로나 작업이 변경될 경우에도 단순한 작업만으로 변경 가능한 자율 주행 이동 로봇에 대한 연구들이 필요하게 되었다.

원하는 경로를 추종하는 방법으로는 주로 원하는 위치와 현재 위치를 이용한 궤

환 제어를 하는데, 조향 장치를 따로 가지지 않은 두 바퀴 또는 네 바퀴 이동 로봇의 경우 한 방향으로만 움직임을 가지는 기하학적 제한이 존재하므로 이를 고려하여 좌표계의 변환 등을 이용하여 효과적으로 제어하고자 하는 연구들이 진행되고 있다[16][18][19].

## 1.2 연구 내용 및 구성

자율 이동 로봇의 개발이 활발하여지고, 특히 여러 대의 이동 로봇을 이용하여 시스템을 구성하고자 하는 필요성이 대두됨에 따라 로봇축구에 대한 관심이 높아지게 되었다.

본 연구는 로봇축구 시스템에서 사용된 소형 이동 로봇에 적용 가능한 실시간 경로계획과 협동 알고리즘[20][21]에 관한 것이다.

로봇 축구 대회에 사용되는 로봇은 자율 주행 이동 로봇으로 위에서 말한 기능, 즉 경로를 계획하고 추종하는 기능을 가져야 하고 다수이므로 서로 협동이 되어야 한다. 자율 이동 로봇의 경로 계획에 관하여는 Voronoi Diagram, Cell Decomposition, Potential Field, 신경회로망 등을 이용한 많은 연구들이 있어 왔으나, 대부분 방법들이 오프라인으로 경로 길이를 최적화 하는 방법들이므로 환경이나 작업이 자주 바뀌는 경우 활용하기 어려운 단점들이 존재한다. 특히 로봇축구는 환경과 목표점이 계속적으로 바뀌고 여러 대의 로봇들이 서로 협동하여 목적을 수행해야 하므로 최적성뿐 아니라 실시간성이 요구된다. 최근에는 동심원 도법을 이용한 경로 설정 알고리즘[14][16]을 사용하고 있다. 본 논문에서는 비선형에서도 보간법이 뛰어난 퍼지 논리를 이용하여 경로 설정 및 추적하는 알고리즘을 구현한다. 또, 계획된 경로 따라 움직이는 로봇은 현재 상황에 가장 알맞은 행동을 취하여 좋은 결과가 나올 수 있도록 해야 한다. 예를 들어 공이 같은 편 로봇 2대 사이 중간 지역에 있다고 하자. 환경을 고려하지 않는 다면 2대의 로봇은 모두 공을 향해 돌진하게 될 것이다. 같은 행동을 2대의 로봇이 중복되어 좋지 못한 결과가 일어나게 된다. 특히 현재 3 대 3 경기에서 5 대 5 경기로 바

뀌게 되면 이러한 현상은 더욱 심각해 진다. 그러나 각 로봇들이 상대 골대와 상대 수비수까지 고려하였다면 다른 행동을 선택하게 될 것이다. 왜냐하면 각 로봇에게 주어진 환경이 모든 것이 같을 수는 없기 때문이다. 본 논문은 환경에 따라 가장 좋은 행동을 선택할 수 있는 알고리즘을 제안하고 이를 위한 로봇 축구 시스템을 구현하는 것을 목표로 하였다.

본 논문은 2장에서 로봇 축구 시스템을 구축하기 위한 제반적인 사항과 기본 지식을 서술하고 경로 계획 및 협동 알고리즘의 필요성에 대해 설명하고 3장은 경로 계획과 추종기에 대해 4장은 협동 알고리즘에 대해 서술한 후 그 결과를 시뮬레이션 및 실험을 통해 검증하였다. 마지막으로 5장에서 결론을 정리하고 향후 과제를 언급하였다.

# 제 2 장 로봇축구 시스템

## 2.1 시스템 제작

### 2.1.1 로봇 제작

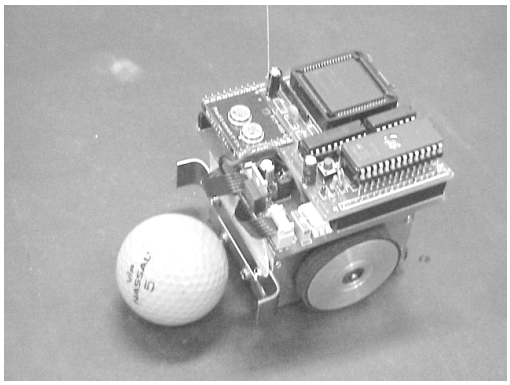


Fig 2.1 Soccer Robot

제작된 로봇은 Fig 2.1에서 보는 것처럼 가로, 세로, 높이가 각각 7.5cm × 7.5cm × 7.5cm이고, 기계부, 제어부, 모터부, 통신부로 구성된다.

기계부는 바퀴, 프레임으로 구성되어 있고, 제어부는 마이크로컨트롤러, 모터부는 모터, 엔코더, 모터 드라이브로 이루어져 있으며 통신부는 RF모듈로 이루어져 있다. 전체 시스템의 사양은 Table 2.1에서 기술되어 있는 바와 같다.

Table 2. 1 The list of hardware

| 구성요소 | 제품명 | 사양 |
|---|---|---|
| PC | PentiumⅡ-450(256MB) | 윈도우98, Visual C++6.0 |
| Frame grabber | Meteor-2/4 standard | 60field/s, 30frame/s |
| Camera | SCC-420(삼성) | 38만 화소 |
| Microcontroller | 80C196KC | 20MHz |

| | | |
|---|---|---|
| Motor driver | L298 | 2 DC 모터 정역전 가능 |
| Communication | RF 모듈 | BIM418, 433 |
| Motor | Minimotor 2224(엔코더 일체형) | 512pulse/rev |
| Body | 알루미늄 | 7.5cm × 7.5cm × 7.5cm |
| Battery | Ni-H Battery | 7.2V 600mA |

### 2.1.2 마이크로 컨트롤러

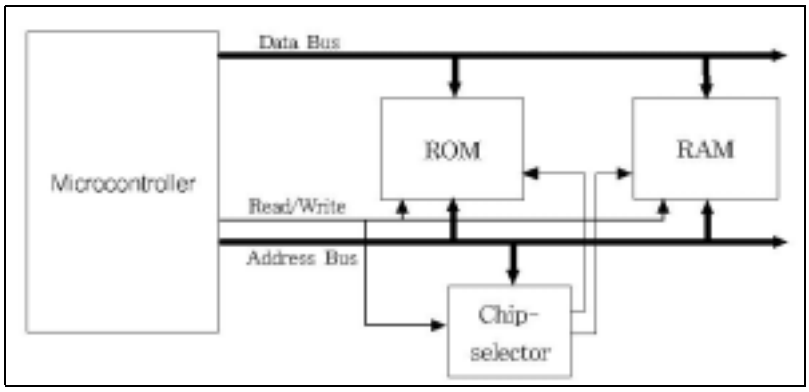



Fig 2. 2 The structure of microcontroller

보드를 설계하기 앞서 먼저 고려해야 할 사항은 바로 마이크로 컨트롤러의 선정이다. 어떤 마이크로콘트롤러를 선택하는가에 따라 성능, 기능, 크기, 용도, 구조 등이 크게 달라지기 때문이다. 마이크로컨트롤러의 기본 구조는 Fig 2.2에서 나타내었다. 원하는 시스템 사양의 속도, 처리 능력, 기능성, 호환성, 편의성 등이 선정 기준을 결정하게 된다. 축구 로봇용 마이크로컨트롤러를 선정하기 위해서는 Fig 2.3와 같은 과정을 따르면 된다.

보통 보편적으로 사용하고 있는 것은 인텔사의 80C196KC나 연산 속도를 향상시킨 80296SA와 8051계열인 ATMEL사의 AT89C52가 있고 최근에는 MICROCHIP사의 PIC계열도 다양한 모델에 호환성을 가지고 많이 사용되고 있다. 본 논문에서는 전용 모터 제어용 칩을 사용하고 있지 않고 자체 PWM 핀을이용하여 모터를 제어하며 직렬통신을 할 수 있는 80C196KC를 선택하였다. 그러나

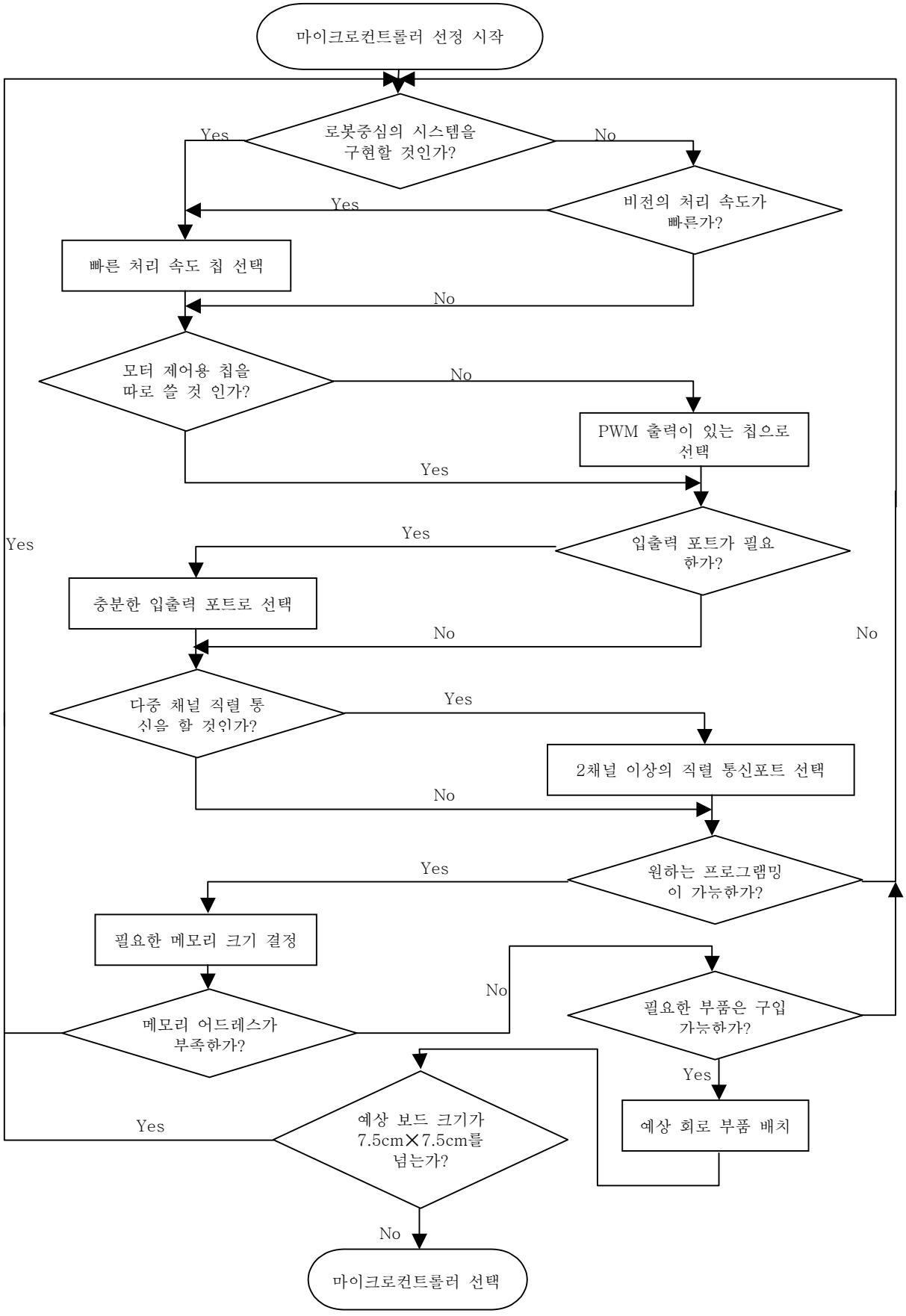



Fig 2.3 The processing selection microcontroller

내부 메모리가 내장되어 있지 않아 부피가 커지고 혼자서 모든 일을 처리하기에는 성능이 모자라는 것이 사실이다. 이를 해결하기 위해 80296으로 업그레이드 시키거나 일을 분담할 수 있도록 작은 CPU을 추가하는 방법이 있다.

본 논문에서 사용된 80C196KC의 경우 외부 메모리와 래치, 칩선택 전용 칩을 모두 사용했기 때문에 공간 활용면에서 매우 비효율적이라 할 수 있다. 이를 해결할 방안으로 현재는 PIC계열로 대치하는 것이 가장 좋은 방법이라 판단한다. 왜냐하면 PIC계열은 4MHz부터 40MHz까지 다양한 모델에 크기, 성능을 지니고 있으며 모두 호환이 가능하고 내부에 메모리를 내장하고 있는 일체형이라 많은 공간을 확보할 수 있다. 또한 기본적으로 마이크로컨트롤러에 대한 개념을 잡고 있거나 처음 시작하는 사람들에게 접근하기 용이한 컨트롤러라 생각한다. 참고로 현재 로봇축구를 하는 대학부의 팀들은 80296SA를 선택을 하고 있다.

### 2.1.3 모터부

모터부는 DC모터 2개를 정역 제어할 수 있는 드라이버로 L298을 선택하였다. L298은 내부에 H-브릿지 회로가 구성되어 있어 컨트롤러에서 발생한 PWM 신호를 H-브릿지의 스위칭 신호에 인가하여 정역 제어를 한다. PWM(Pulse Width Modulation)이란 디지털 회로의 특성을 이용하여 아날로그 처럼 DC모터의 속도를 제어하기 위해 펄스 폭을 조정하여 평균 전압값을 조정하므로써 아날로그와 똑같은 효과를 내도록 하는 방법을 말한다. 아래 Fig 2.4은 펄스 변조에 따른 효과를 보여주고 있다.

Fig 2.5은 H-브릿지 회로로 하나의 모터를 정역전 제어하기 위해 4개의 트랜지스터가 연결되어 있으며 정역전시키는 원리는 Fig 2.6, Fig 2.7에서 보여주고 있다. 참고로 L298에 대한 사양은 부록을 참조하기 바란다.

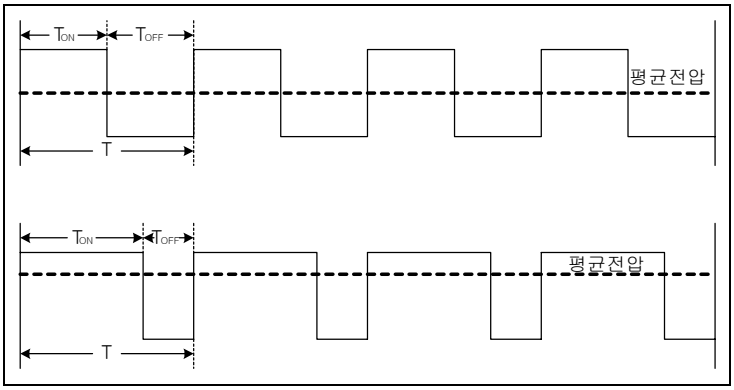


Fig 2.4 PWM method

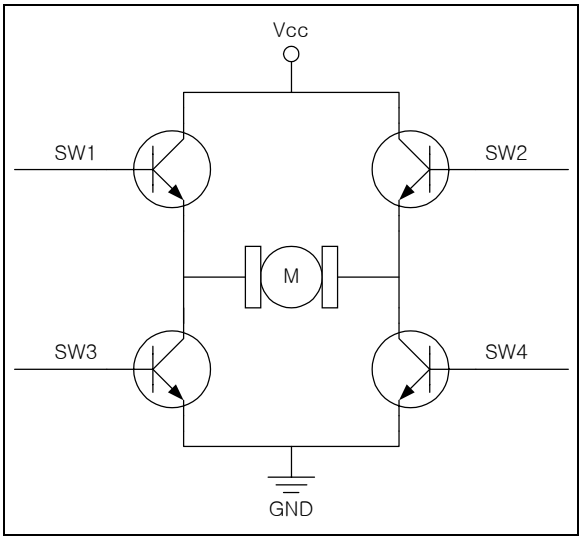


Fig 2.5 H-bridge circuit

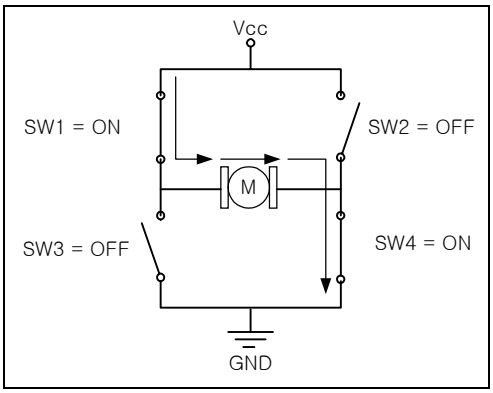


Fig 2.6 The forward direction

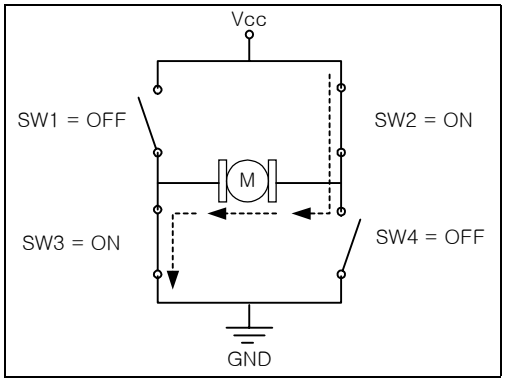


Fig 2.7 The backward direction

## 2.1.4 통신부

통신부는 대부분 팀이 상용 모듈을 이용해서 통신을 하고 있고 주로 사용되는

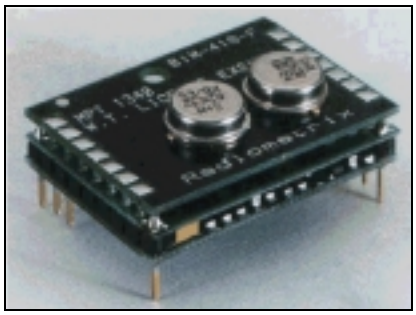


Fig 2.8 RF module

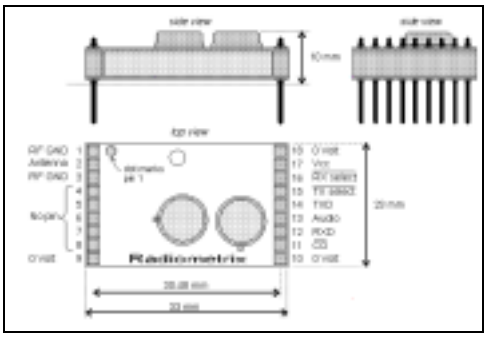


Fig 2.9 Size

모듈은 RF solutions의 RF-418, 433로서 많이 알려진 제품으로 40Kbps까지 전송이 가능하다. 현재 로봇 축구 경기에서는 보통 로봇과 호스트 컴퓨터 사이에 양방향 통신을 하지는 않는다. 대개의 경우 호스트 컴퓨터에서 로봇이 필요한 자료를 로봇에게 무선 통신으로 보내주기 때문에 송신기는 호스트 컴퓨터에만 필요하다. 호스트 컴퓨터에서 나온 명령 패킷은 RS232의 규격을 갖기 때문에 이를 CMOS/TTL에 넣을 수 있는 신호로 바꿔야 한다. 이를 위해서 MAX232를 사용한다. 이 신호는 통신 모듈을 통해 안테나로 송신되고 로봇에서는 수신부에서 이 신호를 통신 모듈을 통해 신호화하고 안정을 위해 NOT 게이트를 거쳐 데이터를 얻는다.

무선 통신은 유선 통신과 다른 점이 선이 있느냐 없느냐의 차이다. 하지만 그 외에 통신상에 문제점이 발생하게 된다. 첫번째 문제점은 송신기에서 아무런 데이터를 보내지 않을 때이다. 무선의 경우 송신기에서 데이터를 보내지 않더라도 수신기측에서는 동기 타이밍을 놓치면서 주변의 잡음 영향에 의한 이상한 데이터를 수신한 것 처럼 동작하게 된다. 이 문제를 해결하기 위해서는 평소에 동기를 맞출 수 있는 신호를 3ms이상 송신해야 한다. 이를 위해 1과 0의 비율이 차례대로 발생시킬 수 있는 0xAA나 0x55를 사용한다. 두 번째 문제는 타이밍을 맞추기 힘든 비트 변화가 없는 부호의 연속적인 송신시 타이밍을 잃어버리는 문제이다. 예를 들어 0x00를 연속으로 보낸다고 가정하면 현재 사용하고 있는 방식은 비동기 방식이므로 동기를 맞추기 위한 별도의 클럭을 서로 주고 받지 않는다. 따라

서 수신된 데이터의 비트 변화를 이용하여 동기의 차이를 수정하는데 0x00과 같이 비트 변화가 없는 같은 부호가 연속적으로 들어온다면 이러한 동기의 차이를 맞출 기회가 없는 것이다. 이런 오작동을 막기 위해 일정 구간 안에 '1'이 되는 구간과 '0'이 되는 구간을 50:50으로 맞추어 주는 코드 밸런싱을 해야 한다. 이 통신부분은 시스템을 구성하며 가장 힘들었던 부분 중 하나이다. 본 논문에서는 이를 해결하기 위해 맨체스터 코드를 사용하였다.

### 2.1.5 맨체스터 코드 방법

맨체스터 부호는 신호 '1'과 '0'을 각각 '10', '01'에 해당하는 부호로 치환함으로써 부호의 중앙에 변화가 있도록 만드는 것이다. 이렇게 되면 0이나 1이 연속되는 부호가 들어오더라도 맨체스터 코드를 사용하면 항상 부호의 변화가 생기는 것을 알 수 있다. 단 전송해야 하는 데이터가 2배가 되므로 동일한 정보량을 보내려면 통신 속도를 두 배로 올려서 전송해야 하고 속도를 올릴 수 없다면 전송량은 절반으로 떨어지는 것을 감수해야 한다.

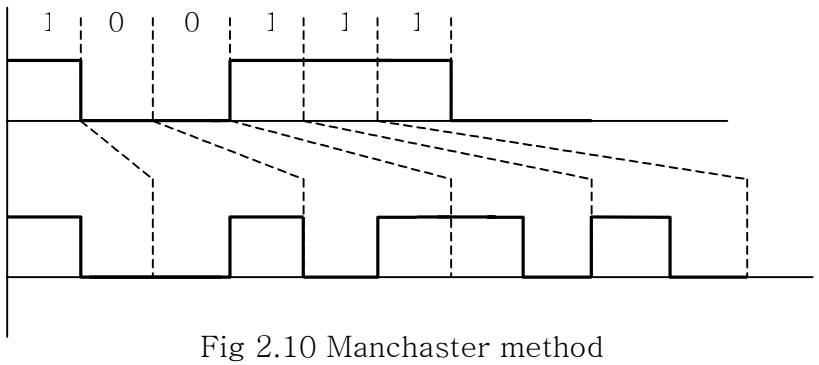



Fig 2.10 Manchaster method

### 2.1.6 영상 처리 시스템

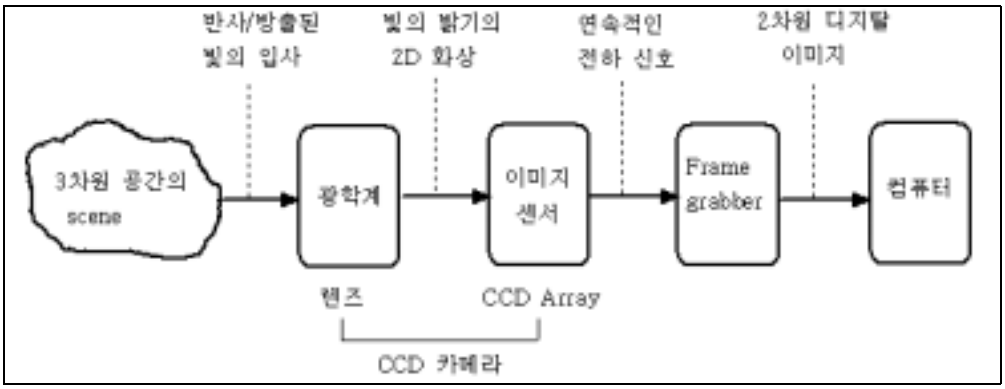



Fig 2.11 Vision System

Fig 2.11은 비전 시스템의 구성을 나타내었다. 광학계는 3차원 장면을 2차원으로 투사시킨다. 이미지 센서는 빛의 밝기를 전기적인 신호로 변환하는 것으로, 사용되는 센서로는 CCD(Charge Coupled Device)를 이용한 카메라가 보편적으로 사용된다. 카메라는 광학계와 이미지 센서로 구성된다. 프레임 그래버는 아날로그 신호를 디지털 신호로 바꾸어 화상 신호의 한 프레임을 메모리에 저장하는 장치이다. 컴퓨터는 프레임 그래버로부터 얻어진 화상을 처리하기 위한 여러 가지 알고리즘을 수행하고 그 처리 결과 등을 저장한다. 이것을 다시 알고리즘으로 표현한다면 Fig 2.12와 같고 내용은 CCD 카메라로부터 NTSC 신호를 입력받아 Frame Grabber를 통해 RGB로 변환하고, 그것을 다시 YUV의 칼라포맷으로 선형 변환하여 Thresholding을 시킨다. 순차 라벨링을 통해 분류시킨 후 필터링을 거쳐 공과 각 로봇을 찾아서 위치 및 각도 정보를 알아 낸다.

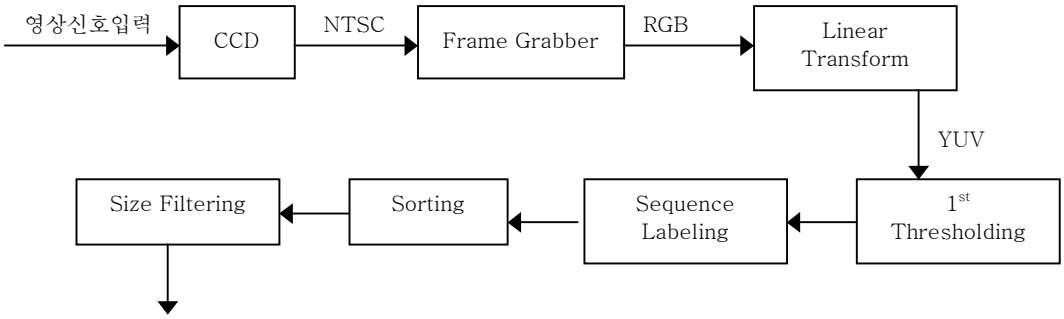



Fig 2.12 Vision algorithm

로봇 축구에서 영상 처리는 매우 중요하다. 이는 로봇의 눈에 해당하고 이 성능

에 의해 1차적인 모든 성능이 좌우 된다고 해도 과언이 아니다. 먼저 카메라에서 비전의 성능은 좌우된다. 들어오는 영상에 노이즈가 있으면 당연히 목표물을 찾는 것이 힘들기 때문이다. 또한 렌즈의 줌 성능에 따라 2m 높이에서 넓은 운동장을 왜곡없이 잡을 수 있도록 셋팅을 해야 한다. 사실 시스템을 구성하면서 무선 통신과 함께 가장 힘들었던 것이 카메라 문제였다. 2m 높이에서 보통 3 대 3 경기를 한다면 운동장 크기가 150cm×130cm로 일반 CCD 카메라에 6mm 렌즈를 사용하면 충분하게 잡을 수 있다. 하지만 5대5경기 일 경우 운동장의 크기는 약 2배의 면적인 220cm×180cm가 되고 4mm렌즈를 써도 전체 운동장을 잡을 수 없다. 또한 가장자리의 왜곡은 전술에서 심각한 오류를 가져 올 수 있다. 성균관대의 킹고팀이나 포항공대의 마스카라팀의 경우 플릭스의 TMC-7을 사용하는데 이는 매우 고가의 제품이라 사용하기가 힘들고 최대한 잡을 수 있도록 렌즈를 맞춰준 다음에 가장자리의 왜곡을 고려한 전술을 구현할 수 밖에 없다.

## 2.2 경로 궤적 계획의 필요성

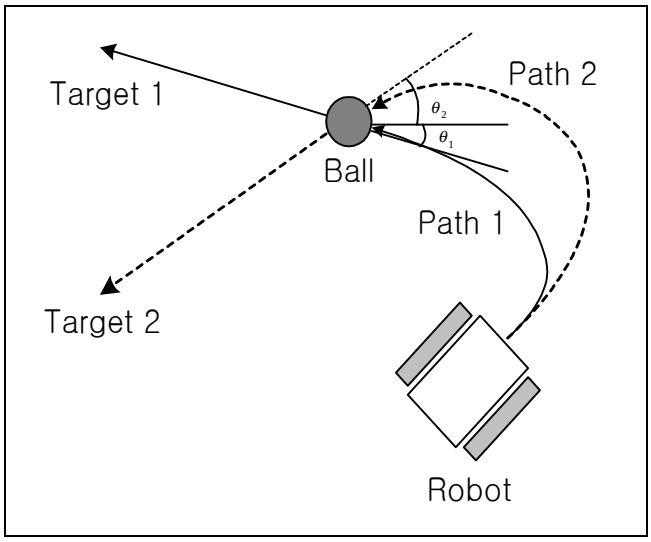



Fig 2.13 The path of robot

로봇 축구 시스템에서 로봇이 해야 할 일 가장 중요한 것 중 하나는 가장 짧은 시간에 공을 골대로 몰아가거나 공격자를 막는 것이다. 이때 로봇이 취할 수 있

는 기본 운동 형태는 회전과 직선으로 구성된 형태다. 공은 속도가 일정치 않고 어느 방향으로 갈지 알 수 없다. 또한 상대 로봇보다 빨리 움직여야 하고 로봇 자신의 속도에 맞게 경로를 계획하지 않고 동심원을 그리다 원심력에 의해 밖으로 튀어 나갈 수도 있다. 따라서 로봇의 운동이 불필요한 움직임을 만들지 않고 알맞은 속도로 정확하게 공을 잡을 수 있는 경로 계획을 세우는 것이 무엇보다도 중요하다. 주로 경로 계획은 kick을 위주로 이루어져 있다. 이는 최단시간에 정확하게 상대방 골대로 공을 몰아 넣는 것이 시합의 승패를 좌우하기 때문이다.

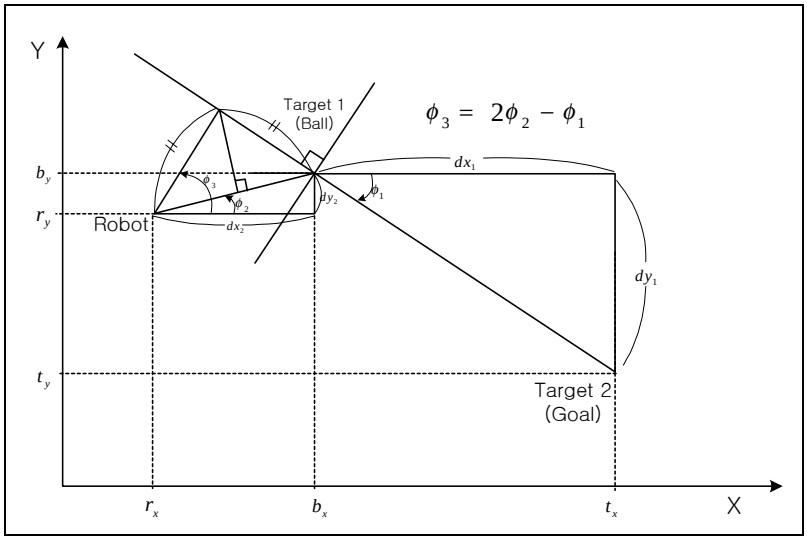



Fig 2.14 the degree of robot

Fig 2.13을 살펴보면 로봇이 목표1로 공을 차기 위해서는 경로1를 따라 이동해야 하고 목표2로 공을 차기 위해서는 경로 2를 따라 움직여야 한다. 이때 로봇이 공의 위치에 도달할 때의 각도는 각각 $\theta_1$, $-\theta_2$이다. 즉, 로봇이 공을 차기 위해서는 로봇이 공의 위치로 도달할 때의 각도로 고려해야 한다. 이 각도를 고려하여 로봇의 왼쪽, 오른쪽 바퀴의 속도를 계산한다.

Fig 2.14에서 Target1을 공이라고 하고 Target2를 공을 보내고자 하는 위치라고 하자.  공을 잡기위해 먼저 각도 $\phi_2$로 로봇이 이동하고 공에 도달 후에는 Target2에 도달하기 위한 각도 $\phi_1$로 바뀌어야 한다. 이를 위해 $\phi_3$으로 취하면

Target1로 수렴하면서 그 순간 방향각이 $\phi_1$이 된다. 매순간 로봇이 향해야 하는 각도는 $\phi_3$이라고 하고 실제 로봇의 방향각을 $\theta_r$이라고 하면 각도오차를 다음과 같이 정의할 수 있다.

$$\theta_e = \phi_3 - \theta_r \qquad (2.1)$$

로봇중심 속도 $v_c$를 고정했을 경우 오른쪽 바퀴와 왼쪽 바퀴의 속도 입력은

$$v_R = v_c + K_a \cdot \theta_e \qquad (2.2)$$

$$v_L = v_c - K_a \cdot \theta_e \qquad (2.3)$$

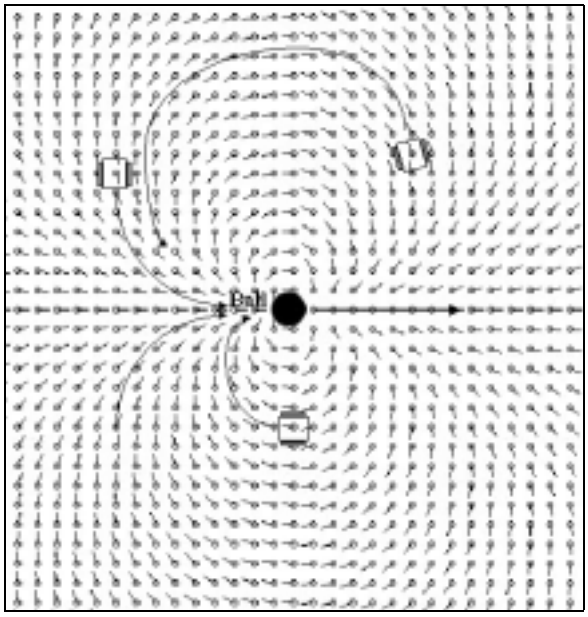
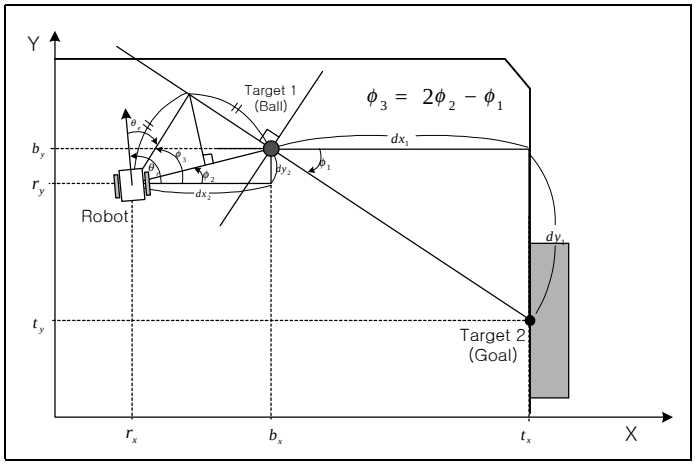


Fig 2.15 The result of simulation

여기에서 $K_a$는 각도 오차에 대한 비례상수로 제어기 변수이다. 이를 이용하면 $\theta_e$를 0으로 수렴시켜서 로봇이 $\phi_3$를 따라가도록 할 수 있다. Fig 2.15은 위의 방법으로 각 위치에서 $\phi_3$을 표현한 Fig이다. Fig 2.16은 경기장에서 각 로봇과 공의 위치와 각도를 보여준다. 그러나, 이 방법은 로봇이 공에 수렴할 때 방향각이




심

Fig 2.16 The calculation of degree

하게 변한다.

즉, 로봇이 공을 차기 위해서 공보다 위쪽에서 접근할 때는 로봇이 향해야 하는 각은 음(-)의 각이다. 그러다가 로봇이 공보다 아래쪽에 위치하면 로봇이 향해야 하는 각은 양(+)의 각이 된다. 결국, 로봇이 먼 곳에서 공을 차기 위해 접근해오다가 공과 같은 선상에 위치하여 접근해 올 때는 위 아래로 흔들림 현상이 생길 수 있다. Fig 2.17은 로봇의 흔들림 현상을 표현한 것이다.

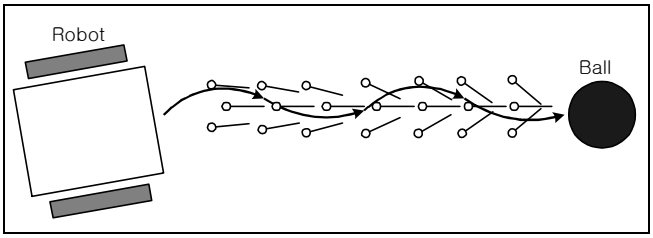



Fig 2.17 Robot wave

이를 해결하기 위해 Fig 2.18는 공에 접근되는 수평선상의 영역을 확장시켜 로봇이 영역 안으로 들어오면 모두 공을 향하는 방향각을 갖도록 변형시킨 Fig이다.

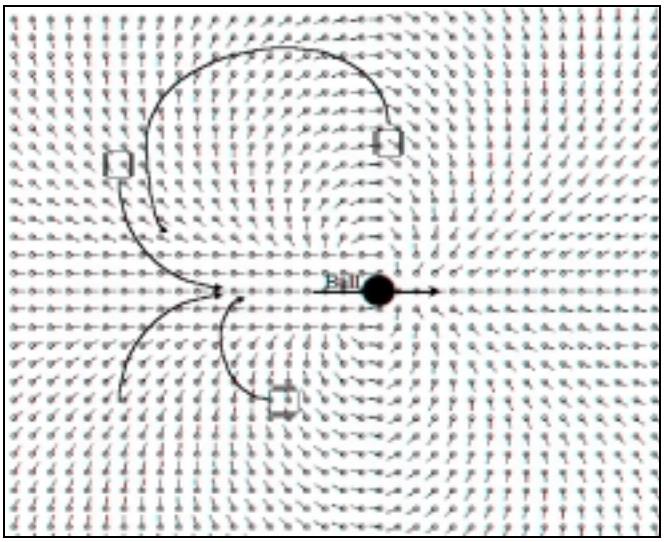


Fig 2.18 The expansion of path

지금까지 로봇이 볼을 잡기 위해 가장 가까운 경로를 계획하기 위해 경로를 설정하고 각 경유점을 잡아서 경로 계획하는 과정을 살펴 보았다. 이 방법들은 가장 기초적인 방법들로 쉽게 구현이 가능하지만 볼에 접근하는데 필요이상의 경로를 돌아 가게 된다. 이것을 극복하기 위해 많은 경로 계획법들이 연구되고 있고 또한 계속 연구되어야 할 부분이다.

## 2.3 협동 알고리즘의 필요성

로봇 축구는 로봇 개체의 속도제어와 위치제어를 하위 개념으로 제어를 기반으로 로봇축구 경기 가짐에 있어 매 시간마다 일어나는 현상을 고려하여 능동적인 전술을 구사해야 한다. 이것은 축구라는 특별한 환경상 많은 변수와 다변적인 환경 변화로 사용자가 어떤 상황을 고려하여 행동을 설정해 두었다 하더라도 그와 다른 환경이 무수히 많이 생기게 되고 또한 사용자도 모든 상황을 고려한 행동을 설정해 주는 것은 거의 불가능하다고 하겠다. 특히, 5 대 5 경기는 이전보다 훨씬 넓어진 운동장과 많아진 로봇 수로 변수와 상태 값들이 증가하였고 상황에 따른 행동 설정은 그 만큼 더 힘들어 졌다. 따라서 실시간 적으로 제어하기 위해서는 입력 변수 값을 일반화 시키고 간단한 제어 알고리즘으로 각 로봇의 행동을

정의할 수 있는 방법이 요구된다.

가장 간단한 알고리즘은 슛팅과 막기다. 각 로봇에게 역할을 부여하고 공격수에게는 슛팅 만 수행하고 수비수는 상대 공격수를 막는 역할을 부여한다. 이는 가장 기초적이며 협동은 배제한 알고리즘이다. 실제로 처음 참가하거나 3 대 3 경기에서는 공격수, 수비수, 골키퍼로 나누어 지기 때문에 이런 알고리즘이 많이 쓰였다. 하지만 5 대 5 경기부터는 골키퍼를 제외하고 4대가 경기를 수행하고 한 두 대 정도가 고정 임무를 수행한다 해도 다른 로봇들이 능동적인 역할 변화를 시키며 여러 가지 전술을 구가할 수 있는 여건이 되었다. 현재 각 로봇 축구 팀들은 초창기 하드웨어와 하위 개념의 제어에 주력하여 위치 및 속도 제어와 하드웨어 측면에서는 크게 차이가 없다. 로봇의 경우 가장 큰 차이를 낼 수 있는 모터는 거의 모든 팀들이 스위스 미니모터 2224를 사용하고 있고 비전 성능을 좌우하는 카메라 또한 외국제품을 사용하는 몇몇 팀을 제외하고는 성능에서 별반 차이가 없으며 프레임 그래도 또한 일반적으로 초당 30frame, 초당 60field에서 크게 차이가 없다. 경기에서 승패를 좌우할 요건은 비전 처리 알고리즘과 전술 알고리즘인 것이다. 비전 처리 알고리즘은 본 논문에서 주 관심사가 아니기 때문에 전술 알고리즘만을 본다면 여러 로봇이 자기편이 공격에 유리 할 때 적극적으로 도와 주면서 공격을 활발히 수행하고 수비가 불리할 때 수비 가담하여 상대의 공격을 차단하는 능동적인 로봇의 행동 변화가 필연적이다 하겠다.

본 논문에서는 협동 알고리즘으로 로봇들이 상황에 따른 행동에 따라 상벌을 부여하여 가장 적합한 행동을 수행할 수 있도록 하는 강화학습법을 이용하여 구현한다.

# 제 3 장 경로 계획

앞 장에서 밝힌 바와 같이 빠른 시간 내에 움직이는 목표점에 도달하기 위해서는 경로의 계획이 필요하다. 따라서, 경로 계획은 다음과 같은 사항을 고려해야 한다.

1) 구현의 용이성

2) 가능한 한 짧은 경로

3) 움직이는 목표물의 연속 추종

4) 정지 또는 이동하는 장애물 회피

로봇축구에서는 목적지까지 빨리 갈 수 있게 하기 위한 경로 생성을 실시간으로 해야 하기 때문에 연산 시간이 적게 걸려야 한다. 그리고 이미 만들어진 경로를 빠른 속도로 오차 없이 잘 추종되어야 한다. 이 장에서는 가장 많이 사용되는 경로 계획 알고리즘의 종류와 장단점을 알아보고 본 논문에서 사용하는 변형된 경로 계획 알고리즘에 대해 알아 보겠다.

## 3.1 경로 계획

### 3.1.1 단위 벡터장법

경로 계획 구현 방법으로 가장 많이 사용되는 것 중 하나는 단위 벡터장법이다. 단위 벡터란 크기가 1인 각도 정보만을 가지는 벡터를 뜻한다. 이러한 단위벡터로 구성되는 장을 단위 벡터장이라고 한다. 즉, 단위 벡터장 내의 모든 벡터는 크기가 1이고 각도 값만을 갖는 단위 벡터이다. Fig 3.1는 크기가 1이고 각도 값을 갖는 단위벡터로 구성된 단위 벡터장을 나타낸다. 여기서

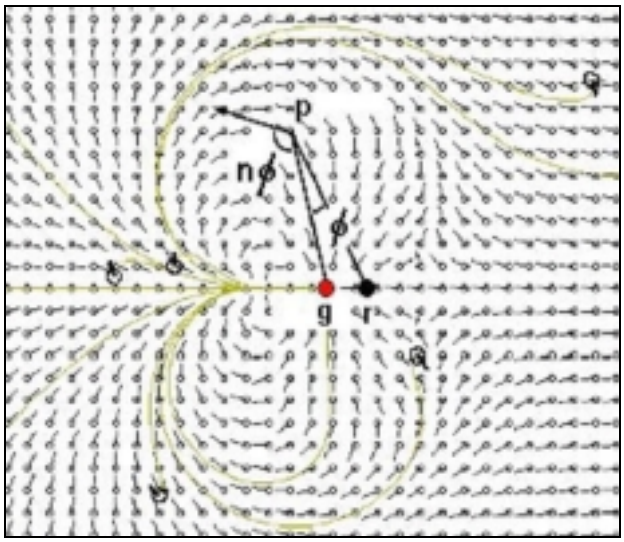



Fig 3.1 The unit vectors

작은 원과 선분은 로봇의 현재 위치와 그 지점에서의 움직여야 할 로봇의 방향각을 나타낸다. p 는 현재 로봇의 위치이고, g 는 공의 위치, r 은 공 앞에 설정된 임의의 지점이라고 한다면, 이때, 위치 p에서 단위 벡터를 $F(x,y)$또는 $F(p)$ 라고 하면 로봇의 위치 p에서 생성된 벡터는

$$|F(p)|=1 \qquad (3.1)$$

$$\angle F(p)=\angle\overrightarrow{pg}-n\phi \qquad (3.2)$$

로 표현할 수 있다. 여기서 $\phi=\angle\overrightarrow{pr}-\angle\overrightarrow{pg}$ 이고, n은 적절히 선정된 양의 상수이다. Fig 3.1에서 작은 원들은 로봇들의 현재 위치이고 선분은 그 위치에서 로봇이 향해가야 하는 방향각을 나타낸다. 그러나 Fig 3.1을 보면 알 수 있듯이 수평선 중심 영역의 벡터장 흐름이 좁아서 이 영역에서 로봇은 좌우로 비틀리는 현상이 일어난다. 이를 보완하고자 확장된 단위 벡터장법을 사용한다. Fig 3.2은 확장된 단위 벡터장을 보여준다. 즉, 로봇이 위의 부분에 있을 경우 확장 영역에서의 g 와 r 점은 공보다 위쪽 위치로 설정하고 아래쪽 영역에 위치할 때는 그에 대칭되도록 설정하여 구한다. 이와 같이 구현하면 로봇이 목표점으로 수렴하다 확장된 영역 안에서 실제 공의 방향을 취하게 되어 단위 벡터장이 부드럽게 변하여 흔들

림 현상이 사라지게 된다. 이 단위 벡터장은 구현이 쉽다는 장점이 있으나 로봇이 공으로 수렴 시 필요이상의 원을 그리게 된다는 단점이 있다.

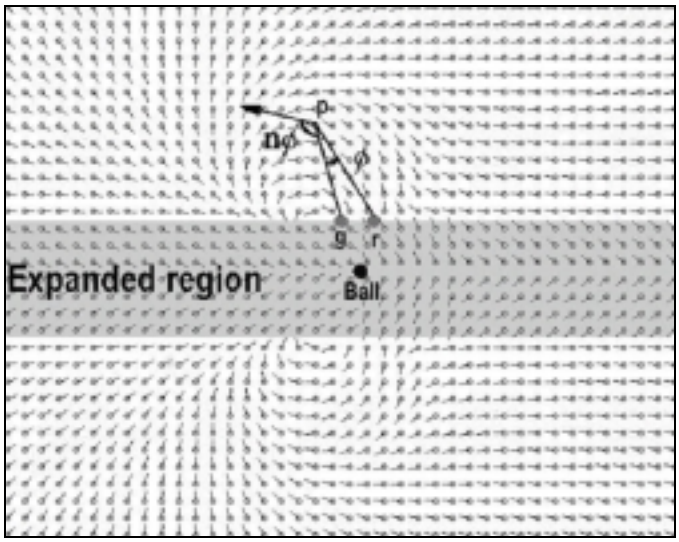



Fig3.2 The expansion of unit vectors

### 3.1.2 동심원 도법을 이용한 경로 계획

최근에 많이 사용하는 방법은 원과 직선을 이용하는 방법이다. 즉, 로봇의 위치와 공의 위치에 내접하는 원을 구한 후 이를 직선과 호로 연결했을 때 가장 짧은 거리의 경로를 찾는다. Fig 3.3에서 $\theta_h$ 는 초기 위치에서의 로봇의 방향각이고, $\theta_d$ 는 최종점에서의 로봇이 향해야 하는 방향각이다. 이런 방법을 적용하기 위해서는 먼저 로봇이 거쳐야 할 경로를 만들어 낸다. 그리고 로봇이 향해야 할 각과 매 순간 로봇의 방향각과의 오차를 이용하여 바퀴 속도값을 만들어 낸다. 이 방법은 최단기간에 원과 직선으로 이동하게 된다는 장점을 가지고 있으나 구현이 어렵다는 것과 로봇이 회전반경 외부에 있을 경우는 문제가 안되지만 로봇이 공에 내접하는 원 안에 위치하는 경우에는 문제가 된다.

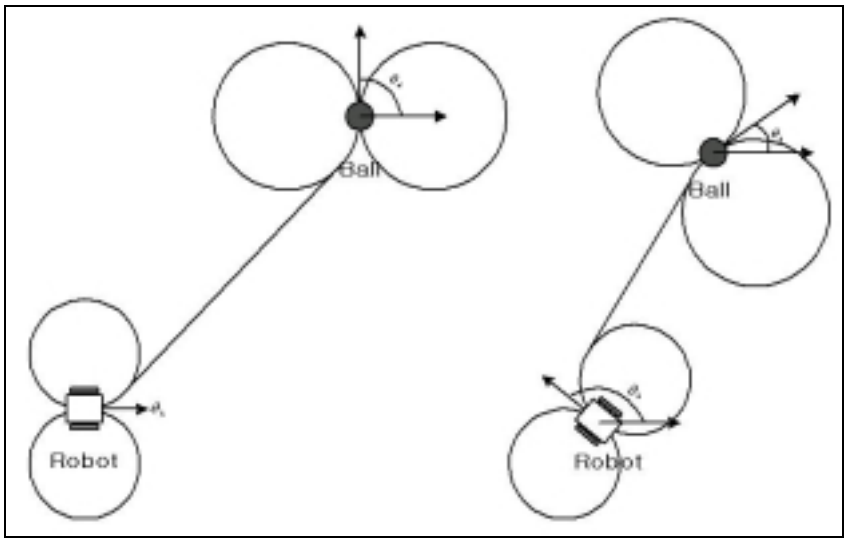



Fig 3.3 the method of circle and line

### 3.1.3 퍼지 논리 이용 경로 계획법

퍼지 논리는 비선형 특성을 가지고 보간법이 우수하여 임의의 계획을 모사할 수 있다. 즉 로봇이 어떤 점에서 다른 점으로 이동하는 경유점을 알고 싶을 때 이를 유추할 수 있어야 하는데 이를 보간법이라 하고 보간법이 우수한 퍼지 논리를 이용하여 원과 직선으로 만드는 경로 계획을 모사할 수 있다. 경로 계획은 현재 로봇의 위치에서 목표점까지 가는 궤적을 생성하게 된다. 이를 따라가기 위해서는 그 궤적으로 로봇의 방향이 위치해 있어야만 한다. 따라서 로봇의 위치에서 원과 직선으로 만들어진 경로를 따라가기 위해서 로봇이 현재 위치에서 취해야할 각을 구해야 한다. 보통 로봇의 위치와 목표점을 나타낼 때는 일반적으로 직교좌표계를 사용하게 되지만 경로계획을 퍼지로 모사할 때는 목표점을 기준으로 하여 극좌표계로 바꾸어 나타내어야 한다. 퍼지에서는 입력 값을 퍼지 논리에 맞게 값의 특성에 따라 퍼지화을 해야 하는데 입력 변수에 잡음이 없고 신뢰성이 높으면 sigletone방식을 취하고 그렇지 못할 때는 gaussian 형태 또는 삼각파 형태의 소속함수를 사용하는 것이 좋다.

## 3.2 변형된 경로 계획법

위에서 기하학적 경로 계획법과 퍼지 논리를 이용한 방법을 살펴 보았다. 기하학적 경로 계획법에서 가장 좋은 방법은 목표지점까지 직선 경로로 이동하는 것이다. 그래서 주로 원과 직선을 이용한 방법을 사용한다. 이는 거리가 가깝게 계획할 수 있으나 반면 목표점이 공이 되는 데 무작정 공을 향하게 된다면 자기 진영에서는 자살골이 될 수 있다. 따라서 상대 골 쪽을 바라보며 다가가도록 하거나 어느 지역에서는 다른 알고리즘을 쓰도록 연구를 해야 한다. 또한 직선과 원을 이용할 경우 원호의 반지름을 고려해야 하는데 이 반지름은 목표점에서 최종속도와 함수관계를 가진다. 왜냐하면 로봇이 회전할 때 그 원의 구심력 $F_c$을 넘지 않아야 하기 때문이다. 그렇지 못하면 미끄러짐이 생겨 원하는 목표점으로 갈 수 없다. 따라서 최단 반경 R은 다음과 같은 조건을 만족해야 한다.

$$R \le \frac{mv^2}{F_c} \qquad (3.3)$$

위 식에서 원의 구심력 $F_c$는 바퀴와 접한 면의 마찰력과 관계를 가진다. 이러한 마찰력을 정확하게 측정하기 힘들기 때문에 속도를 변화시키면서 그에 따른 반경을 실험으로 구하게 된다. 또한 이렇게 구한 반경을 유지하며 로봇을 제어 할 경우 방법에 따라 문제가 생길 수 있다. 앞에서 언급한 것 같은 일정이상의 속도로 높일 경우 바퀴의 미끌림 현상이나 trade-off 현상이다. 아무리 실험치로 적당한 반경을 구했다 하더라도 순간 바퀴와 바닥과의 미끌림으로 타원형을 그리거나 작은 반원으로 구동되기도 한다. 따라서 매 순간 로봇의 경로를 벗어나지 않도록 제어할 수 있어야 한다. 퍼지의 경우 비선형 특성에 대해 보간법이 우수하여 경유점 유추가 가능하고 바퀴의 속도 제어에서도 퍼지는 모든 변화를 고려할 수 있는 특징이 있다. 위에서 퍼지 논리를 이용한 경로 계획이 있었으나 그 방법의 경우 극좌표계를 이용하여 로봇과 공을 기준으로 일정범위를 설정하여 거리와 각도

에 따른 방향을 단위 벡터로 설정하여 구현되었다. 이 경우 경로 추종시 로봇의 목표방향과 자신의 방향과의 오차와 거리를 이용하여 속도 구현이 가능하고 퍼지 규칙으로 모든 곳에서의 자세 표현이 가능하지만 일정 거리의 범위가 한정되어 있고 로봇과 공의 위치에 따른 최단 거리를 표현하는 것에 단점이 있다. 여기에서는 기하학적 경로 계획법인 동심원을 이용하는 방법의 장점인 최단거리 경로 구현과 퍼지의 장점인 경유점 유추를 적절히 이용하여 가장 효율적인 방법을 연구해 보았다.

### 3.2.1 동심원을 이용한 경로 계획

우선 슛 동작을 일반적인 동심원 기법을 기본으로 활용한다. 로봇이 경로를 선택하기에 앞서 슛을 하기에 좋은 조건인지를 확인한다. 슛을 하기 좋은 조건이란 Fig 3.4에서처럼 일정한 거리 내에 공과 골대와 수평선상에 놓여 있는 경우다. 만일 조건을 만족하면 상대 골대를 향하여 직선으로 슛을 하면 된다. 이때는 직선 경로에 대한 제어기를 사용한다.

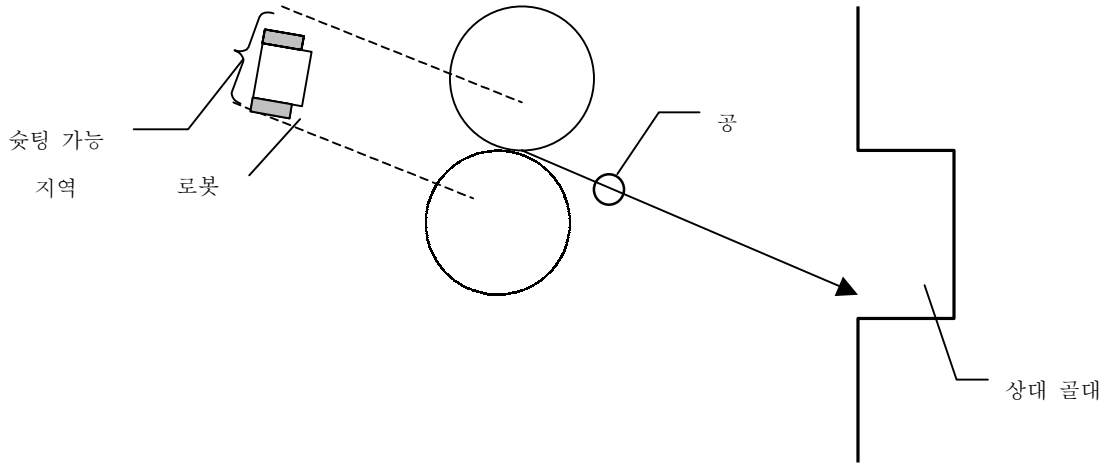



Fig 3.4 the area of possible shoot

Fig 3.5에서는 로봇과 공이 떨어져 있어 슛 가능지역으로 로봇이 이동 후에 슛을 하기위해 경로를 계획하는 과정이다. 본 논문에서는 최종 목표인 상대 골대로 가지 이전에 중간 경유지점을 지정하여 바로 앞의 목표 지점을 달성해 가는 방식으

로 구현하였다. 이는 최종 목표만을 보고 이동하는 것보다는 그 전의 경유점을 경유해 가는 것이 좀더 정확히 경로를 추종하기 때문이다. 동심원법을 이용하여 로봇을 중심으로 내접하는 두 원과 공을 중심으로 슛팅 각도에 수평을 유지하고 일정 거리 내에 $g_3$을 설정하고 이점에 내접하는 두 원을 생성한다. 그리고 경로를 그려보면 로봇이 최종적으로 가고자 하는 목표 지점은 $g_4$이고 목표 각도는 $\theta_4$이다. 최종 목표 지점에 수렴하기 전에 로봇은 그 전의 경유점 중에 자신의 목표지점을 삼는다. 그리고 그 목표 지점에 도달 했을 경우 그 다음 경유지점을 다음 목표 지점으로 삼는다. 여기에서는 최초의 로봇 목표 지점은 $g_1$, 목표 방향은 $\theta_{g1}$이다. $\theta_{g1}$는 로봇의 내접 원과 공의 내접 원의 중심간의 각도인 $\theta_c$와 같고 최초 목표 지점인 $g_1$의 좌표 ($r_x$, $r_y$)를 구하는 방법은

$$r_x = circle_x - radius \times \cos(90° - \theta_r) \qquad (3.4)$$

$$r_y = circle_y - radius \times \sin(90° - \theta_r) \qquad (3.5)$$

이다. 두번째 목표 지점인 $g_2$의 좌표 또한 같은 방법으로 구할 수 있고, 직선 방향으로 이동한다. $g_3$까지는 $g_1$으로 이동 할 때 처럼 원호를 따라 이동하고 $g_4$를 향해 직선으로 밀어준다.

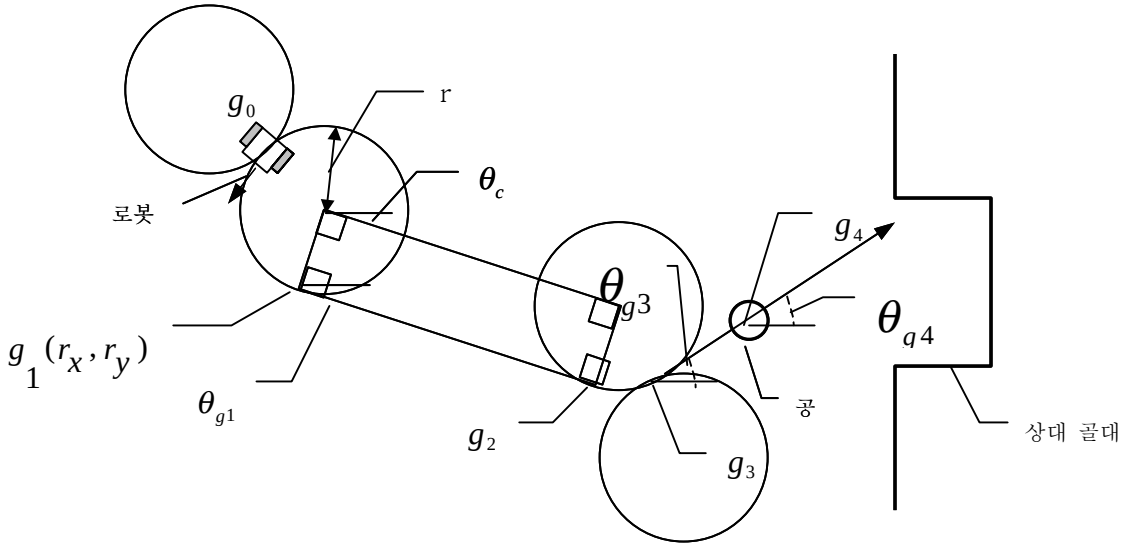



Fig 3.5 The situation 1

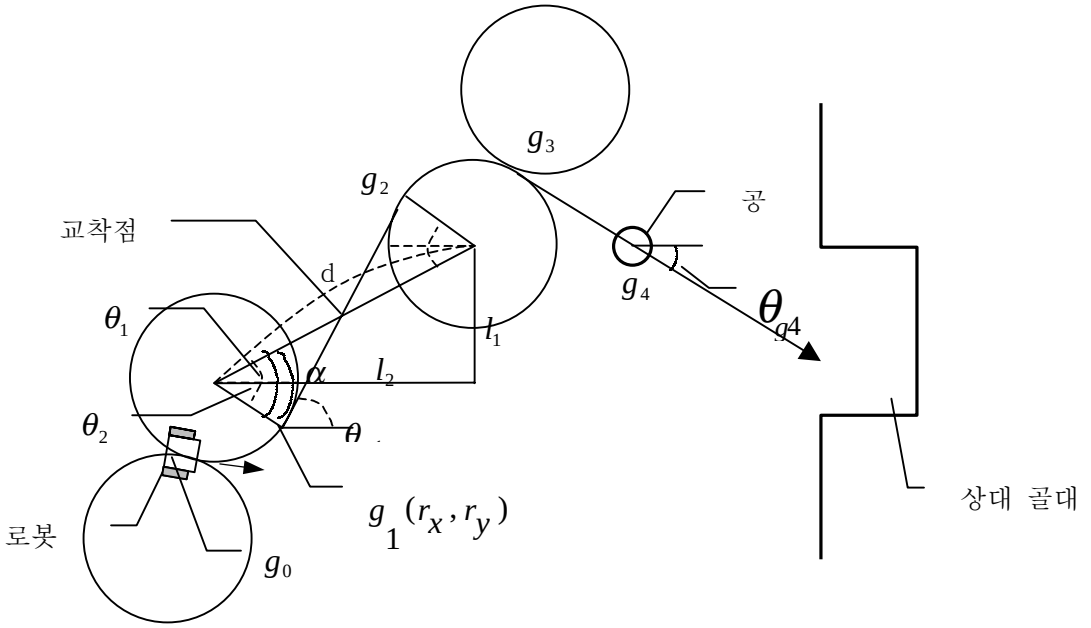



Fig 3.6 The situation 2

Fig 3.6의 경우는 로봇이 슛팅하려는 방향이 로봇 위치에 대해 교착점을 형성시킨 경우다. 이 경우는 이전의 경우보다 로봇의 목표 지점과 목표 각도를 구하는 것이 조금 복잡한 식을 통해 구할 수 있다. 최초 목표지점 $g_1$의 $(r_x, r_y)$과 목표각도 $\theta_{g1}$은

$$\sin\alpha = \frac{2r}{d} \quad \alpha = \sin^{-1}(\frac{2r}{d}) \tag{3.6}$$

$$\theta_1 = \tan^{-1}(\frac{l_1}{l_2}) \tag{3.7}$$

$$\theta_{g1} = 180° - (90° + \theta_2) \tag{3.8}$$

$$r_x = c_x + r \times \cos\theta_2 \tag{3.9}$$

$$r_y = c_y - r \times \sin\theta_2 \tag{3.10}$$

와 같다. 여기에서 ($c_x$, $c_y$)와 r 는 로봇이 회전하는 원의 중심 좌표와 반경을 말한다.

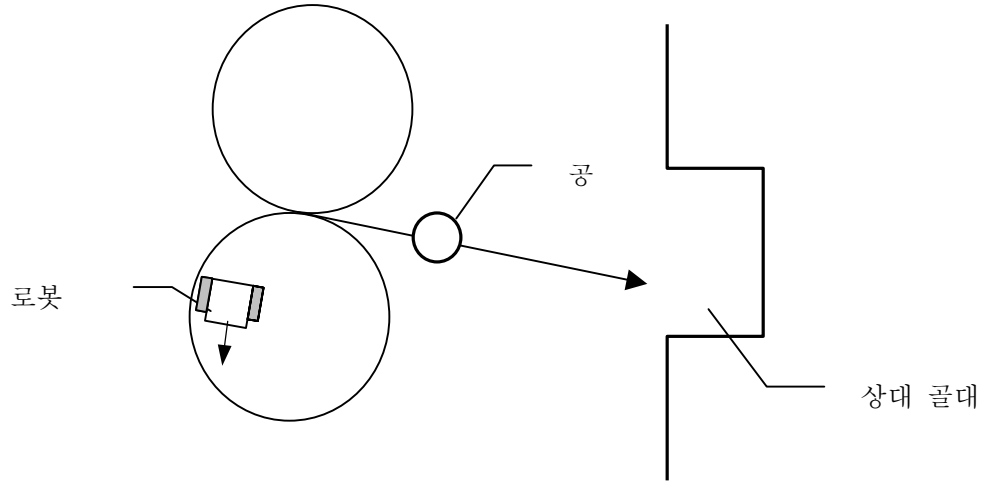



Fig 3.7 The situation 3

Fig 3.7은 로봇이 원의 반지름 내부에 있을 경우를 보여준다. 다음 장에서 구현될 원호 경로 계획법에 의해 생성된 방향에 따라 적절한 속도로 부드러운 곡선을 그리며 나오게 된다.

### 3.2.2 퍼지 논리를 이용한 원호 경로 계획법

동심원법을 이용하여 경로를 생성한 후 퍼지를 이용하여 그 경로에서 로봇이 거쳐야 할 경유점과 그 곳에서 목표 방향을 구한다.

위에서 퍼지를 이용한 경로 계획법에 대해 소개를 했었다. 여기에서 퍼지 이용과 앞에서 소개한 것과는 차이가 있다. 예전에 연구된 퍼지 논리를 이용한 경로 계획은 공과 로봇의 관계를 극좌표계로 표현하며 각 위치에서의 방향값을 설정해서 경로를 계획하였다. 이 방법은 공과 로봇의 거리에 따라 멀수록 큰 회전 반경을 그리게 되어 시간적으로 손실이 많게 된다. 본 논문에서 생성하는 방법은 중간 경유점을 기준으로 원호 경로 계획과 직선 경로 계획을 구분하여 설정한다. 먼저 원호 경로 계획 방법을 살펴 보면 Fig 3.6에서처럼 회전할 원호에서 각 위치와 각도에 따른 방향을 정해 주고 그에 따라 퍼지 논리를 거쳐서 최종적으로 로봇의 위치에 따른 목표 방향을 찾아낸다.

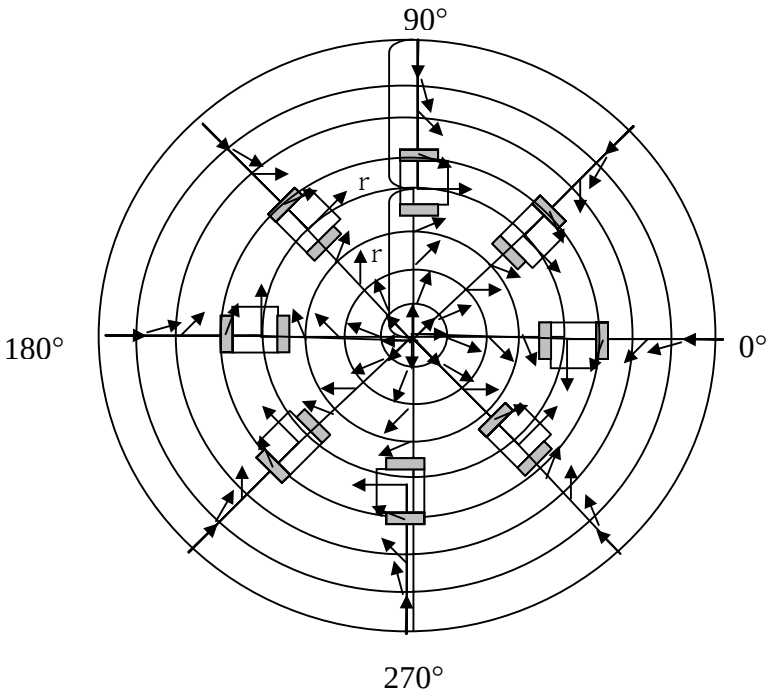



Fig 3.8 The plan path

### 3.2.2.1 입력값의 퍼지화

입력값은 원호 중심에서의 로봇까지 길이와 그 위치에서 각도를 말한다. Fig 3.7 은 입력값을 퍼지화하

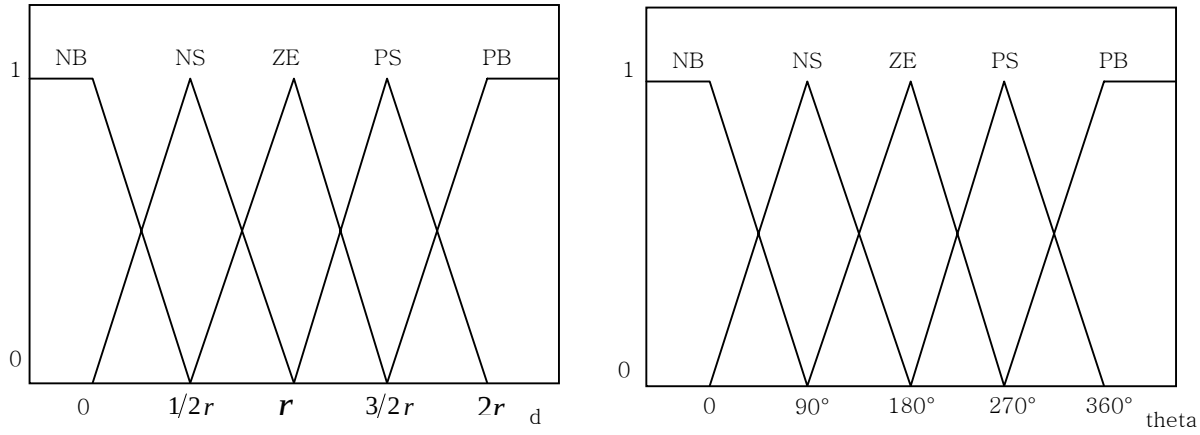



Fig 3.9 The membership function

기 위한 소속 함수를 보여준다. 여기에서 사용된 퍼지화에 사용된 방식은 잡음에 강하고 계산량이 적은 삼각파 형태를 취한다. 여기에서 잡음이란 비전에서 들어

오는 로봇의 정보에 섞여있는 약간의 편차를 말한다.

### 3.2.2.2 퍼지 규칙

퍼지 제어 규칙은 “if-then” 형식의 퍼지 조건문으로 구성된다. 이러한 조건문은 일반적으로 여러 개의 복수입력□ (MIMO)으로 구성되게나 다음과 같이 복수입력□ (MISO)로 나타낸다.

$R_1$ : If $x_1$ is $A_1$ and $x_2$ is $B_1$ … then z is $C_1$

$R_2$ : If $x_1$ is $A_2$ and $x_2$ is $B_2$ … then z is $C_2$

$R_n$ : If $x_1$ is $A_n$ and $x_2$ is $B_n$ … then z is $C_n$

일반적으로 제어 규칙을 추출하는 방법은 전문가의 경험적 지식, 숙련자의 조작 행위로부터 추출하는데 여기서는 이미 입력과 출력 데이터 쌍을 알고 있으므로 쉽게 알 수 있다. Table 3.1는 제어 규칙을 나타내었다.

Table 3.1 lookup table

| $\theta$ \ d | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|
| NB | 0° | 45° | 90° | 135° | 180° |
| NS | 90° | 135° | 180° | 225° | 270° |
| ZE | 180° | 225° | 270° | 315° | 360° |
| PS | 270° | 315° | 360° | 45° | 90° |
| PB | 360° | 45° | 90° | 135° | 180° |

### 3.2.2.3 의사 결정 단계

퍼지 입력과 퍼지 제어 규칙으로부터 퍼지값을 추론하기 위해서는 의사 결정 논리가 필요하다. 추론법 결론부는 singletone과 삼각형 형태가 있다. 주로 시스템 제어에는 singletone 형태의 추론법이 많이 사용되는데 이는 일반적인 퍼지 논리

를 사용하는 것과 거의 동일한 특성을 갖고 있으며 비교적 연산이 간단하다는 장점을 가지고 있다. 따라서 프로그램으로 구현하기가 용이하다. 추론법으로 사용되는 것들은 Mamdani의 방법과 Larsen, Tsukamoto의 방법 등이 있으나 여기에서는 Mamdani의 min-max의 연산을 사용한다.

#### 3.2.2.4 비퍼지화

추론의 결과가 나왔을 때 이를 직접 조작량으로 출력하면 그 시스템을 정상적으로 움직일 수 없다. 즉, 퍼지화된 값이 “조작량을 조금 작게 하라”라고 한다면 실제 시스템은 이 값을 알지 못한다. 이러한 애매한 출력 값을 명확한 값으로 변화해야 할 필요가 있다. 이를 비퍼지화라고 한다. 비퍼지화 방법으로는 최대값 방법, 최대 평균법, 그리고 무게 중심법이 있다. 일반적으로 무게 중심법이 우수하고 일반적으로 많이 구현되고 있다. 본 논문도 이 방법을 이용한다.

$$u = \frac{\sum_{i=0}^{n} W_I C_i}{\sum_{i=0}^{n} W_i} \qquad (3.11)$$

각 규칙에서의 최소 소속정도를 $W_i$라 하고 퍼지 출력부의 값은 $C_i$가 된다. $u$는 원하는 각도가 된다.

### 3.2.3 퍼지 논리를 이용한 경로 추종기

위에서 로봇이 가야 할 경로를 설정하였다. 하지만 정해진 경로를 특히 원호의 경우 속도를 높이면 경로를 벗어나 정확도가 떨어지게 되고 속도를 낮추면 정확도는 높아지나 시간이 많이 걸리게 된다. 또한 로봇 바퀴와 바닥과의 미끌림으로 전혀 다른 경로로 가는 경우도 있다. 이를 위하여 로봇의 움직임을 포착하여 일정 경로를 벗어나지 않도록 가장 적당한 속도를 유지토록 해야 한다. 여기에서 원호를 따라 갈 경우 왼쪽과 오른쪽 바퀴의 속도를 어느 정도 설정한다. 이 값은 이미 테스트를 거쳐 가장 적당한 값을 정한다.

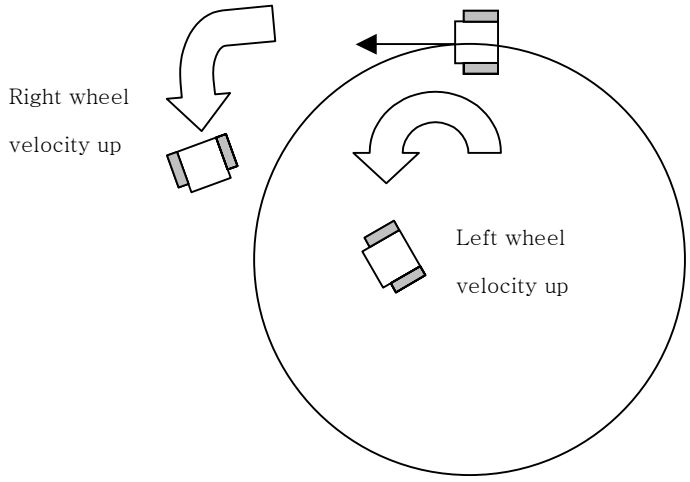



Fig 3.10 The circle tracking

원호에서는 이미 정해진 속도로 이동한다. 가장 적당한 속도는 실험치에 의해 구해진 20cm/s 차이를 가지는 40cm/s와 60cm/s이다. 이때 회전 반경은

$$R = \frac{L}{2}\frac{V_R + V_L}{V_R - V_L} = \frac{7.5cm}{2}\frac{60cm/s + 40cm/s}{60cm/s - 40cm/s} = 18.75cm$$

가 된다. 물론 마찰력을 고려하지 않았으니 어느 정도의 차이는 존재한다. 실제 실험결과 반경 R은 14cm이다. R은 회전 반경, L은 로봇의 길이, $V_R$과 $V_L$은 로봇의 오른쪽, 왼쪽 바퀴 속도값이 된다. 시계방향으로 회전할 경우 왼쪽 바퀴가 오른쪽 바퀴보다 20cm/s의 속도가 클 것이고 반대 방향일 경우는 속도값이 바뀔 것이다. 이 로봇이 회전하다 반경의 외부로 나가게 된다면 기준 속도에서 오른쪽 바퀴의 속도값을 변화시키고, 반경 내부에 위치하게 되면 왼쪽 바퀴 속도값을 변화 시킨다. 그 이유는 양 바퀴의 값을 동시에 보정하는 것보다는 간단하고 예측이 가능하기 때문이다. 속도 변화량을 구하기 위해 입력값은 현재 위치에서 로봇의 각도 $\theta_r$와 목표각도 $\theta_d$의 차이인 $\theta_e$와 현재의 $\theta_e$와 과거의 $\theta_e$의 차이인 에러 변화율 $\Delta\theta_e$로 설정한다.소속 함수로 Fig 3.11에서 나타내었다.

$$\theta_e = \theta_d - \theta_r \qquad (3.12)$$

$$\Delta\theta_e = \theta_{enow} - \theta_{eprev} \quad (3.13)$$

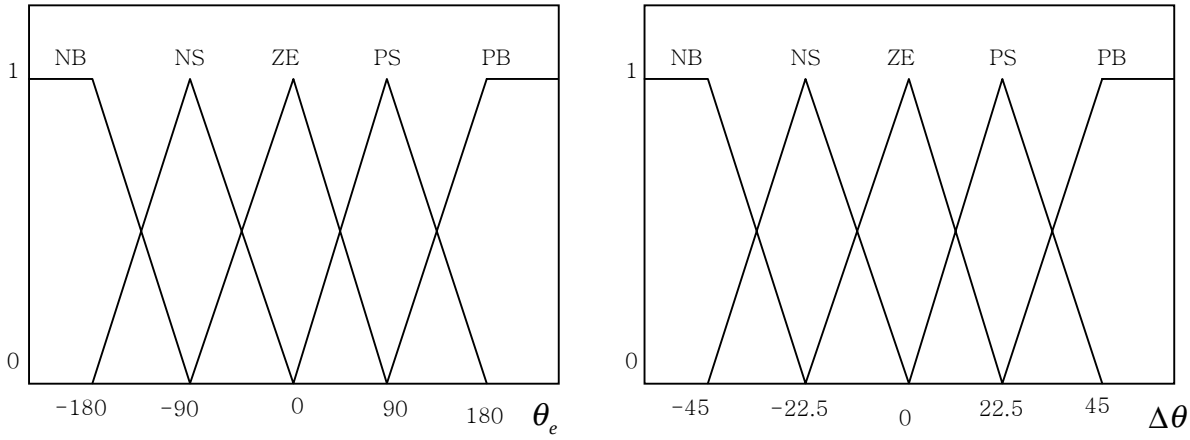



Fig 3.11 the $\theta_e$ and $\Delta\theta_e$ Membership Function

제어 규칙은 왼쪽 바퀴와 오른쪽 바퀴에 상관없이 한번만 구현하고 다만 조건식을 통하여 로봇이 반경의 외부냐, 내부냐에 따라 가감이 되는 바퀴를 찾아 실행한다. 각 규칙에 따른 lookup table은 다음과 같다. Table 3.2의 값은 바퀴의 속도에 가감이 되는 속도값을 나타낸다.

Table 3.2 lookup table

| $\theta$ \ d | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|
| NB | -30 | -20 | -10 | 0 | 10 |
| NS | -25 | -15 | -5 | 5 | 15 |
| ZE | -20 | -10 | 0 | 10 | 20 |
| PS | -15 | -5 | 5 | 15 | 25 |
| PB | -10 | 0 | 10 | 20 | 30 |

이 경우도 singletone을 사용하였는데 앞에서와 같은 이유에서이다. 비퍼지화도 이미 앞에서 소개 되었던 무게 중심법을 사용한다.

### 3.2.4 퍼지 논리를 이용한 직선 경로 계획 및 추종기

직선의 경우 거리에 따라 속도를 조정하면 된다. 이때 미끌림 등으로 방향 오차가 생길 수 있는데 이를 위하여 각도 오차를 고려한다. 따라서 입력 변수를 아래와 같이 소속함수를 구할 수 있다. 변수 d는 로봇과 목표 지점까지의 거리이고 $\theta_e$는 로봇 각도와 목표 각도의 차이를 말한다. Fig 3.12은 이에 대한 소속함수를 나타낸다. 이를 바탕으로 제어규칙을 표현하면 좌 우 바퀴들은 각각 아래와 같이 표현될 수 있다.

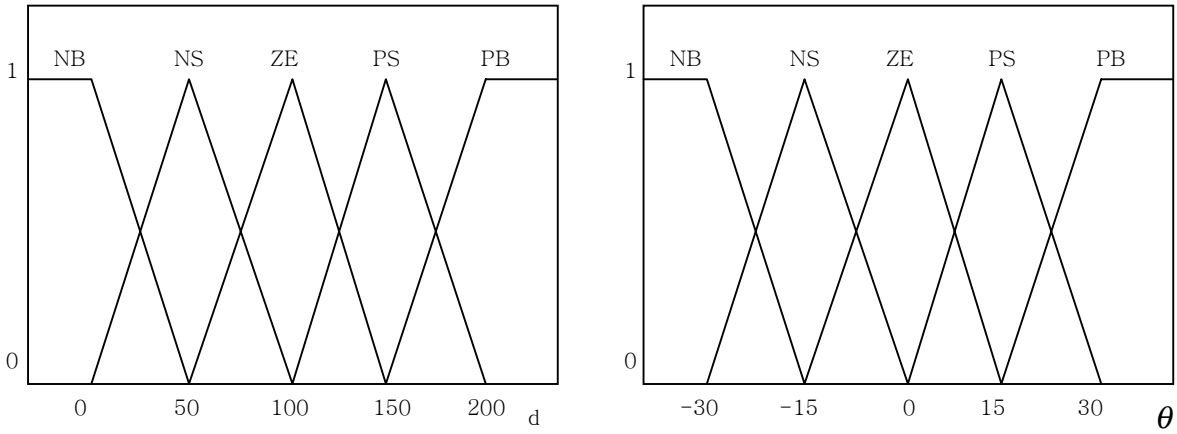



Fig 3.12 Membership Function

Table 3.3. Left Wheel Velocity Lookup Table

| θ \ d | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|
| NB | 50 | 70 | 90 | 110 | 110 |
| NS | 40 | 60 | 80 | 100 | 110 |
| ZE | 30 | 50 | 70 | 90 | 110 |
| PS | 20 | 40 | 60 | 80 | 100 |
| PB | 10 | 30 | 50 | 70 | 90 |

Table 3.4 Right Wheel Velocity Lookup Table

| $\theta$ \ d | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|
| NB | 10 | 30 | 50 | 70 | 90 |
| NS | 20 | 40 | 60 | 80 | 100 |
| ZE | 30 | 50 | 70 | 90 | 110 |
| PS | 40 | 60 | 80 | 100 | 110 |
| PB | 50 | 70 | 90 | 110 | 110 |

### 3.2.5 공이 움직일 경우 경로 계획

로봇 축구에서는 공이 정지해 있을 때보다 움직일 때가 많고 그 움직임을 예측하기란 매우 어렵다. 따라서 경로를 계획할 때도 정지시의 경로 계획법을 세웠다간 공을 놓치게 될 것이다. 본 논문에서는 움직이고 있는 공의 위치를 예측하기 위한 방법으로 예전에 공의 움직임을 데이터로 받아서 이를 바탕으로 다음 위치를 예측하는 것을 사용한다. 공의 이전의 위치를 데이타 베이스화 한다.

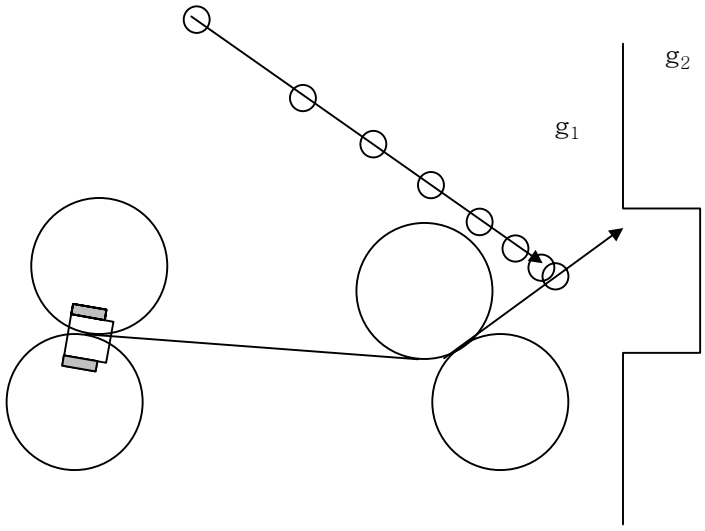



Fig 3.13 The moving ball situation

공의 이전부터 저장하고 있는 데이터 위치 데이터 중 최근 10개의 데이터를 참

고 하여 위치마다의 거리를 구하고 시간은 알고 있으므로 속도를 구할 수 있다. 이를 이용하여 감가속도를 구하고 따라서 공의 예상 경로 및 위치를 추정이 가능하다. 그에 따라 로봇은 $g_2$까지 걸리는 시간을 예상 공의 위치에 대해 거리에 비례하여 시간을 구하고 공과 만나는 것이 가능한 지역으로 이동한다. 공의 위치 예측 지점을 구하는 방법은 Fig 3.13에 나타내었다.

### 3.2.6 슛팅 이외의 경우

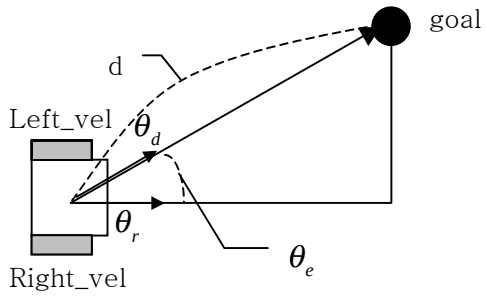



Fig 3.14 The case of exclusion shoot

Fig 3.14은 슛팅 이외의 경우 볼에 접근하기 위해 또는 목표지점에 접근하기 위하여 사용되는 간단한 방법을 표현하고 있다. 여기에서는 목표지점에서 자세보다도 목표 지점까지 도달하는 것이 우선 과제이다. 이런 동작은 막기, 공잡기, 멀리차기가 있다. 이때는 거리와 각도을 입력값으로 고려한 직선에 대한 추종기를 약간 보정하여 사용한다. 여기에서 우리는 특별히 고려해야 할 사항을 알아보자.
로봇축구 중에는 공이 벽에 붙는 경우가 많이 발생한다. 이를 처리하기 위한 특별한 알고리즘이 필요하다. 하지만 복잡한 여러 가지 알고리즘은 오히려 더 좋지 않은 결과를 가져오고 간단한 방법이 복잡한 환경에서는 더욱 좋은 결과를 가져오는 경우가 자주 발생한다. 그것은 많은 변수에 대해 감독자가 모든 것을 고려할 수 없기 때문이다. 여기에서는 공이 벽에 붙었을 경우 원하는 목표지점은 벽과 공의 거리를 고려하여 적당한 지점을 설정한다. 로봇이 목표지점에 도달한 후 회전하고자 하는 방향에 바퀴를 중심으로 급회전하여 공을 튀어 나가도록 한다.

물론 로봇 앞에는 돌출부가 있고 이때 공의 방향은 확신할 수가 없다. 하지만 특정한 곳에 공이 머물러 있거나 로봇이 공을 구석으로 몰아 나오지 못하는 경우보다는 좀더 나은 상황이 될 것이다. 목표지점 $g_1$은 벽과 공과의 거리를 통해 구하고 로봇의 관성을 고려한 offset 값을 준다.

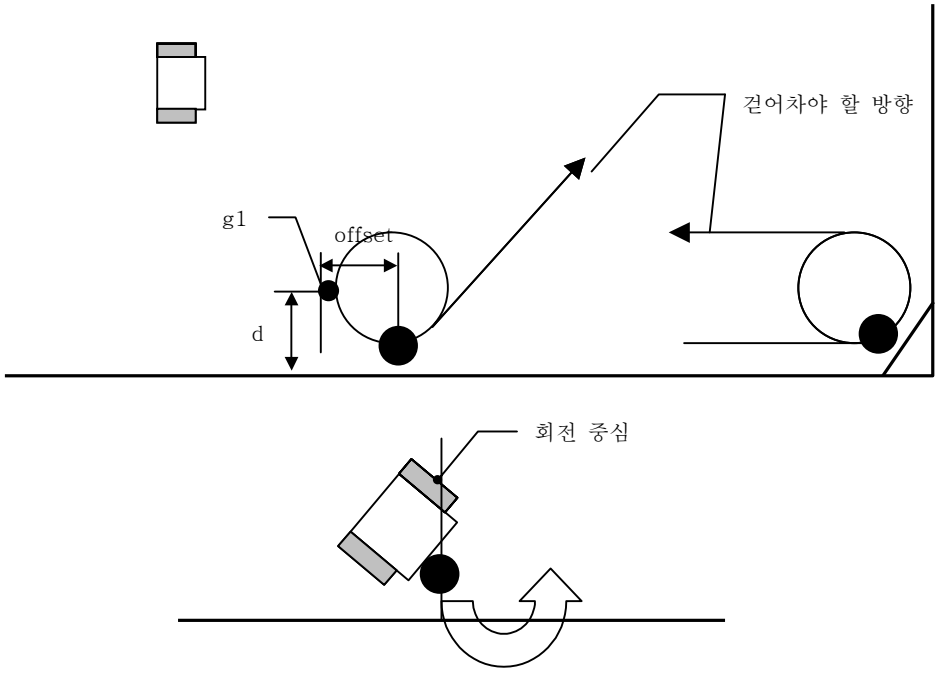
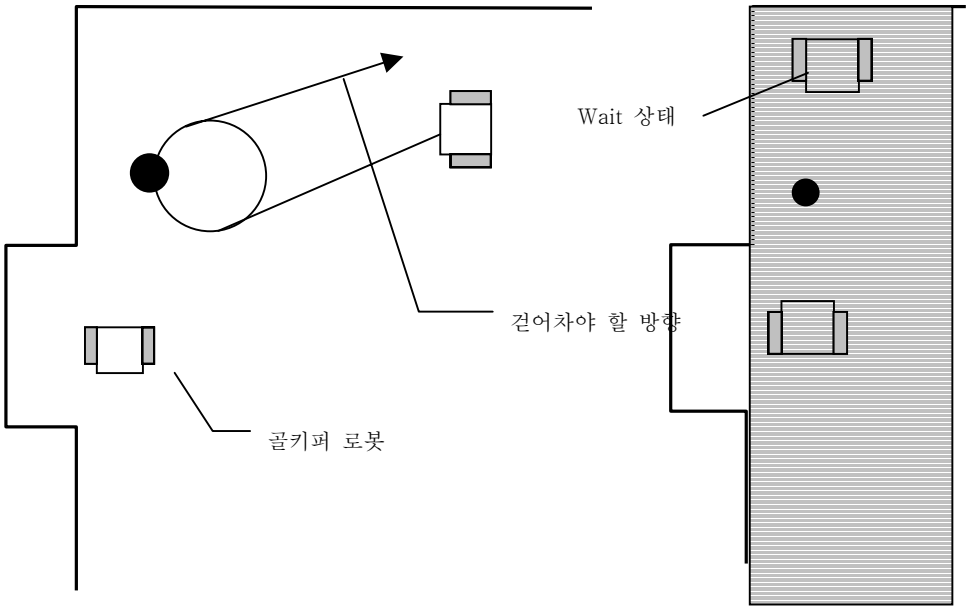



Fig 3.15 the ball kick

Fig 3.16의 경우는 우리 골대 근처에서 수비수의 경우를 보여준다. 수비수는 공과 자기편 골대와의 관계를 확인해야 한다. 그림의 경우 수비수 로봇이 무작정 공을 차려 한다면 자칫 자살골로 연결되는 가능성이 있다. 따라서 자살골이 되지 않도록 공이 자신과 골대 사이에 위치할 경우는 걷어 찰 때 골대의 반대방향으로 나갈 수 있도록 목표지점을 잡는다. 즉, 골대-공-수비수 순의 (a)와 같을 때 위험이 최대한 작은 위치를 선택한다. 골대와 공과 수비수가 (b)처럼 어느 일정 영역, 즉 공을 건드렸을 때 위험 부담이 많을 때는 골키퍼에게 처리를 맡기고 자신은 대기 상태로 들어간다.




(a) (b)

Fig 3.16 the kick of defense

### 3.2.7 골키퍼 경로 계획

골키퍼의 경우 로봇 축구에서 가장 중요성 위치라고 할 수 있다. 강한 팀일수록 골키퍼의 성능은 뛰어나다. 골키퍼는 공과 상대 로봇의 움직임에 민감해야 한다. 특히 공이 골대 근처에 위치할 때 처리하는 것과 상대 로봇이 볼을 몰고 들어 올 때 처리 사항은 상당히 중요하다. 여기에서는 공과 상대 공격수의 움직임을 파악하는 골키퍼 알고리즘을 구현한다. Fig 3.17에서 보듯이 공을 붙잡은 상대 로봇의 방향을 구하여 골키퍼의 경로와 만나는 점을 구하여 이동한다.

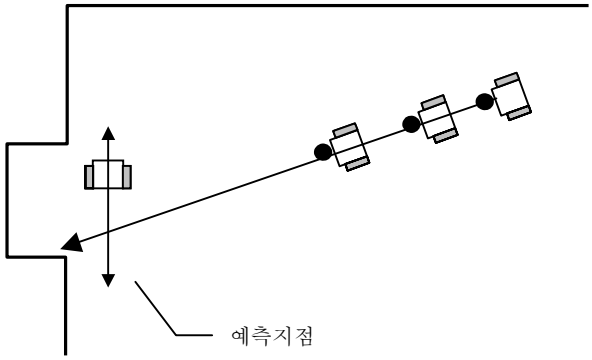



Fig 3.17 goal keeper

## 3.3 실험

이 장에서는 로봇의 경로 추종을 제안된 알고리즘을 적용한 결과로 샘플링 시간마다의 위치값을 바탕으로 실험 결과를 보여준다. Fig 3.18은 원호 궤적을 생성하는 과정을 보여준다. 기존의 방법에 의한 원호 궤적 생성에서 로봇은 나선형으로 돌며 원호 궤적에서 벗어나지만 제안된 방법은 위치에 따른 수정으로 원호 궤적을 잘 생성하고 있다. Fig 3.19은 시작위치 $g_0$에서 목표지점 target까지 추종하는

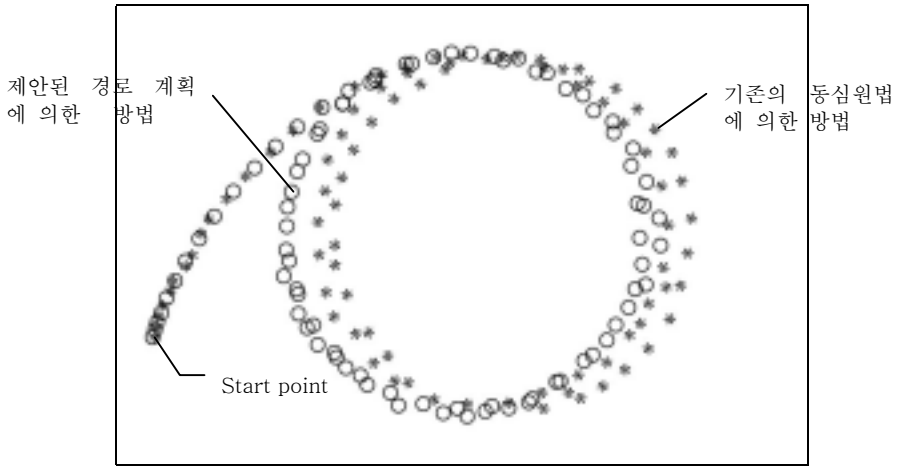



Fig 3.18 the circle path tracking

과정을 보여주고 있다. Path1은 기존의 동심원도법에 의해 구현한 경로이고 path2는 본 논문에서 제시한 방법에 의해 구현된 것이다. Fig 3. 20, 21, 22은 각각의 경로 계획 방법들을 구현하여 비교 한 것 이다. 그러나 이것은 큰 의미를 가지지는 못한다. 왜냐하면 추종하는 기준 자체가 틀리기 때문이다. 현재 그래프에서 말하는 에러는 현재 위치에서 로봇의 목표 방향과 현재 방향과의 오차 $\theta_e$를 나타낸 것이다. Fig 3.21은 움직이는 공에 대해 추종하는 과정을 보여준다.

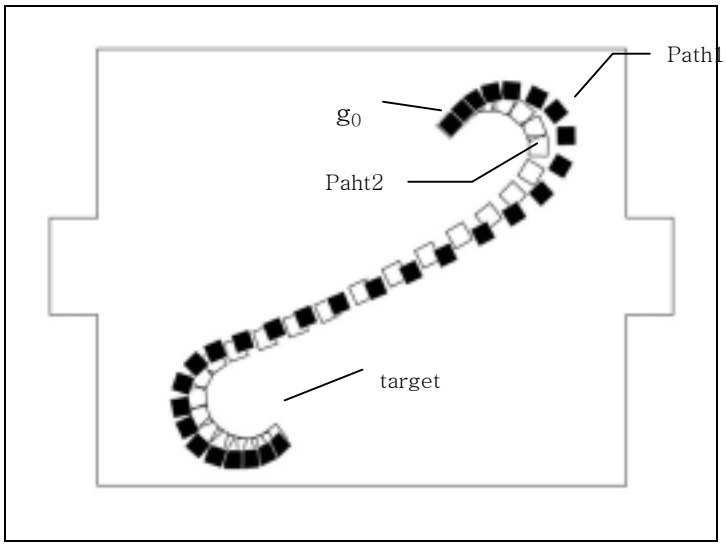



Fig 3.19 The tracking path

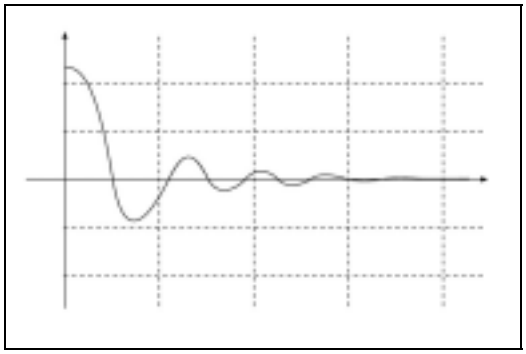


Fig 3.20 the circle and line method

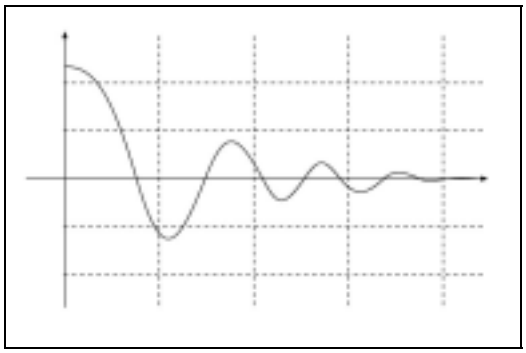


Fig 3.21 the unit vector method

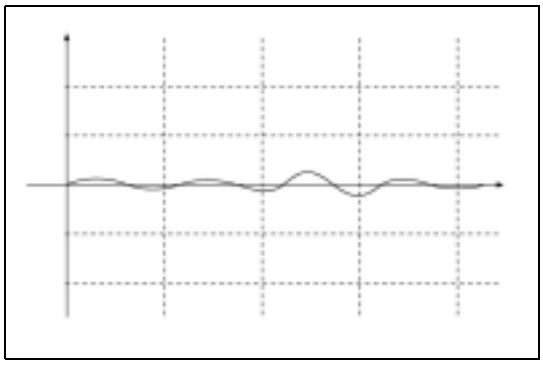


Fig 3.22 new method

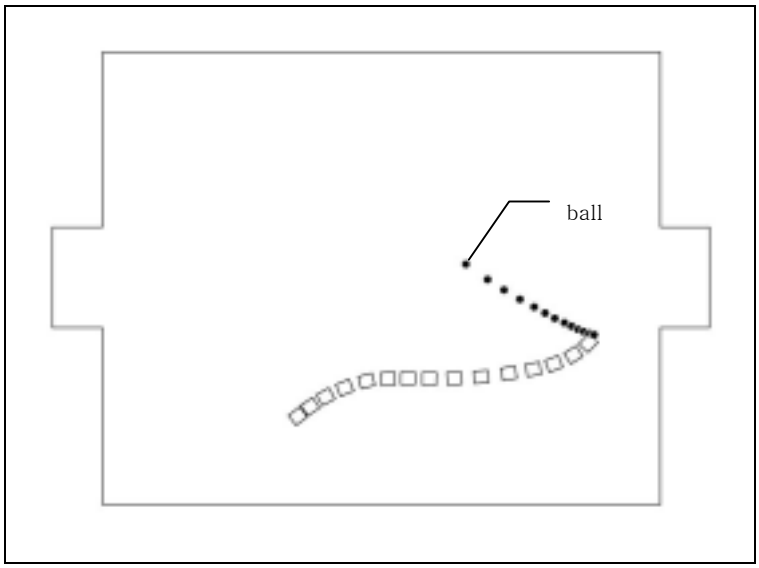



Fig 3. 23 moving ball tracking

# 제 4 장 협동 알고리즘

각 로봇은 앞 장에서 설명한 것 같은 경로계획에 따라 움직임을 취하게 되고 전체적인 상황을 판단하며 어떤 동작을 수행해야 가장 좋은 결과를 가져오는지 결정해 주는 것이 필요하다. 특히 로봇 축구는 여러 대의 로봇들이 동시에 동작하면서 서로 영향을 주고 받으며 협동 작업이 요구되는 동작이다. 로봇의 위치와 공의 위치의 조합이 무수히 많이 존재하므로 각 경우를 모두 모델링하여 그에 따른 행동을 결정하는 것은 매우 어렵다. 따라서 수많은 시행 착오를 시켜 축적된 경험을 바탕으로 결정을 내리는 강화학습 기법을 사용하였다.

## 4.1 강화 학습

강화 학습은 생물의 학습 방법을 응용한 것으로 인간은 출생 이후 자신의 센서(눈, 코, 귀 등)로부터 입력 받은 자료를 바탕으로 구동 기관(손, 발 등)을 가지고 원하는 작업을 행하려고 할 때 수많은 시행 착오를 거쳐 그 대응 관계를 학습하게 된다.

어떤 행동에 의해 목적이 달성되면 그 때의 행동은 보상을 받게 되어 강화하는 과정을 겪는다. 즉 작업이 성공하는 경우의 행동은 강화되고 실패하는 경우는 억제 시킨다. 인간의 경우 이러한 강화는 본능적인 자극, 예를 들어 쾌감이라든가 즐거움 또는 고통이나 슬픔 등의 보상을 바탕으로 이루어 진다. 개체가 환경을 인식하고 그에 맞는 행동을 함으로써 환경을 바꾸게 되고 각각의 상호 작용 단계에서 개체는 환경이 주는 상태를 입력으로 받으며 출력으로 행동을 선택한다. 이 행동은 환경의 상태를 바꾸게 하고 상태의 변화는 개체에서 강화 신호로 전달하게 한다. 개체의 행동들은 오랜 기간 동안의 강화 신호 값의 합을 전달하게 한다. 강화 학습에서 에이전트의 목표는 최종적인 보상을 최대로 하는 정책을 찾는 데에 있다. 보상을 정의하는데 있어서 주의해야 할 점은 문제 해결에 대한 최종적

목표에 도달했을 때만 보상을 주어야 한다는 것이다.

강화 학습은 현재 대부분의 연구에서 쓰이는 기계 학습이나 패턴 인식, 그리고 신경회로망과 같은 교사 학습과는 다르다. 교사 학습은 외부의 총명한 감독자로부터 주어지는 예로써 학습한다. 이것은 학습에서 중요한 것이지만 이것만으로는 상호작용을 통한 학습에는 적당하지 않다. 강화 학습에서는 입력과 출력의 연결을 나타내는 표현을 쓰지 않는다. 대신에 행동을 위한 개체는 포상과 다음의 상태를 알게 된다. 개체는 처할 수 있는 상태와 행동들, 상태의 변화, 그리고 최적이 되도록 행동을 하게 만드는 포상에 유용한 경험들을 수집해야 한다. 다른 학습법과 다른 강화 학습의 또 다른 특징은 탐험(exploration)과 탐색(exploitation) 사이에서의 균형이다. 강화 학습에서의 개체는 많은 포상을 받기 위해 과거에 경험했던 행동 중에서 더 효과적인 행동을 선호한다. 그러나 그런 행동을 발견하지 위해서 전에 선택하지 않았던 행동을 시도해야 한다. 결국 개체는 포상을 얻기 위해 이미 알고 있는 행동들을 선택하기도 하고 미래를 위해 더 나은 행동 선택을 위해서 탐험을 하기도 한다.

강화 학습의 문제를 풀기 위한 기본적인 세 가지 분야는 동적 프로그램밍, 몬테카를로 방법, 시간차 학습이 있다. 동적 프로그램밍은 수학적으로는 잘 발달되었지만 환경에 대한 완벽하고 정확한 모델이 필요하다. 몬테 카를로 방법은 모델을 필요로 하지 않고 개념적으로 단순한 장점이 있지만 단계별 증가 연산에는 적합하지 않다. 시간차 학습방법은 모델이 필요 없는 장점이 있지만 분석하기에 복잡한 단점이 있다. Q-학습은 시간차 학습의 한 종류이다.

## 4.2 용어

● 개체(agent)

학습자 또는 의사 결정자라고 한다. 개체는 환경과 연속적으로 상호 작용을 하면서 행동을 선택하고 환경은 이 행동에 상응하는 새로운 상태를 개체에게 정해준다. 또한 환경은 시간이 지남에 따라 최대로 만들려는 특별한 수치 값을 제공한

다.

● 환경(environment)

개체와 상호 작용을 하는, 개체 외부의 모든 것을 말한다. 개체가 수행한 행동에 대해 개체에게 새로운 상황과 포상을 제공한다. 환경을 나타내는 특정한 것을 일(task)이라고 정의한다.

● 행동과 상태(action and state)

개체의 결정에 의해서 행하는 모든 것 또는 개체가 환경을 변화시키는 모든 작용을 행동이라 한다. 행동은 결정적인 또는 비결정적인 결과를 가진다. 상태는 환경을 정의하는데 필요한 변수들의 집합을 말하며 상황이란 용어를 사용한다.

● 포상(reward)

개체가 수행한 행동에 대한 평가량이다. 개체가 목적을 얻기 위한 수단으로 사용하며 환경이 개체에게 주는 것으로 강화 학습의 특징 중에 하나이다.

● 정책(policy)

각각의 시간 간격에서 각 상태에서 가능한 행동을 선택할 수 있는 확률적인 사상이 존재하는데 이를 개체의 정책이라 한다. $\pi_t$ 라고 표기하며 강화 학습 방법은 개체 자신의 경험의 결과로써 어떻게 정책을 변화시킬 것인가를 특정화하는 것이다.

● 정책, 행동, 상태, 포상의 관계

S는 가능한 상태들의 집합이고 $A(s_t)$ 는 시간 t에 대해 상태 $s_t$ 에서 개체가 행할 수 있는 행동들의 집합이라고 할 때 상태 $s_t$ 에서 개체가 $a_t$ 의 행동을 수행하면 다음 시간 간격 후에 개체는 $r_{t+1} \in R$ 인 포상과 새로운 상태 $s_{t+1}$ 을 받는다. 환경상태가 $s_t \in S$ 일 때 행동 $a = a_t$ 의 확률을 $\pi_t(s,a)$ 라 한다.

● Q-함수와 Q 값

강화학습에서는 주어진 상태에서 개체에게 얼마나 좋은가를 예측하는 함수가 있는데 이를 상태 값 함수이라 한다. Q-학습에서는 이와 비슷한 함수로 Q-함수가

있다. 정책 $\pi$를 따를 때, 상태 s에서 행동 $a$를 선택할 때의 값을 Q 값이라 하고 이를 나타내는 함수를 Q-함수라고 하며 $Q^{\pi}(s,a)$라고 표기한다.

● 마르코프 결정 과정(Markov decision process)

수학적으로 간단한 접근을 위해서 유한개의 상태와 포상 값이 존재한다고 가정하자. 시간 t에서 개체가 행동을 취했을 때 시간 t+1에 환경으로부터 어떤 응답을 받을지를 본다. 보통, 응답은 과거에 있었던 모든 것에 의존하는 인과관계의 특성이 있다.

## 4.3 강화 학습의 최적화 모델

에이전트가 내린 결정에 대해 받은 보상을 어떻게 평가할 것인가에 따라 다음 3가지 모델로 나뉜다.

1) Finite-horizon model

2) Infinite-horizon discounted model

3) Average reward model

### 4.3.1 Finite-horizon model

항상 유한한 개수의 행동으로 목표에 도달할 수 있는 경우 h만큼의 스텝 후에 받을 보상의 기대값은

$$E(\sum_{t=0}^{h} r_t)$$

이다. 여기서 $r_t$는 시간 t에서의 순간적인 보상값을 의미한다. 위의 값을 최대로 하도록 학습한다면 에이전트는 h스텝 후에 가장 큰 보상을 받을 수 있도록 행동하게 된다.

### 4.3.2. Infinite-horizon model

강화 학습의 목표가 최종적인 보상을 최대화하는데 있기 때문에 보통 무한히 계속되는 행동에서 받는 보상을 기준으로 학습하는 것이 유리하다. Infinite-horizon discounted model은 이러한 상황에서 무한히 많은 회수만큼 보상을 누적하기 때문에 누적된 값이 발산하지 않도록 하기 위한 수학적 기술이 필요하다.

$$E(\sum_{t=0}^{h}\gamma^t r_t)$$

여기서 $\gamma$는 절감율로 시간에 따른 보상의 비중을 나타낸다. 즉 t스텝 후의 보상은 $\gamma^t$만큼 절감된 비중으로 계산한다. 일반적으로 강화 학습을 구현하는 데 있어서 가장 많이 사용되는 모델이다.

### 4.3.3 Average reward model

학습이 무한히 계속될 수 있는 경우에 보상 값의 합을 무한히 커지지 않게 하기 위한 방법 중 다른 하나는 보상 값을 시간에 따라 계속 평균을 취하는 것이다.

$$\lim_{h\to\infty}E\left(\frac{1}{h}\sum_{t=0}^{h}r_t\right)$$

## 4.4 Q-Learning

강화 학습에서 최적의 정책을 얻기 위해서는 상태 전이 함수와 포상 함수가 필요하다. 이 두 가지를 강화 학습에서의 모델이라고 하고 각각 $T(s_t, a_t, s_{t+1})$ 와 $R(s_t, a_t)$로 나타낸다. $T(s_t, a_t, s_{t+1})$는 상태 $s_t$에서 행동 $a_t$를 수행할 때 다음 상태 $s_{t+1}$이 될 확률함수이고 $R(s_t, a_t)$는 상태 $s_t$에서 행동 $a_t$를 선택할 때 받는 예상 포상이다. Q 학습이란 이런 모델이 알려지지 않더라도 지연된 포상으로부터 최적의 정책을 얻을 수 있는 강화 학습 방법이다.

시간 t에서 개체는 상태 $s_t$에서 행동 $a_t$를 선택하고 확률적인 포상 $r_t$를 받는다. 절감된 예상 포상 $R(s_t, a_t)$는 개체가 있는 상태와 선택한 행동에 의존하고 개체의 목표는 절감된 예상 포상을 최대로 만드는 정책을 찾는 것이다. 절감된 예상 포상은 다음과 같다.

$$R(s_t, a_t) = E\{\sum_{t=0}^{\infty} \gamma^i r_{t+i}\} \quad (4.1)$$

여기서, $\gamma$는 절감 계수이며 $0 \le \gamma < 1$의 범위를 가진다. 절감계수는 t 시간 간격에서 받은 포상이 현재 받은 포상에 비해 $\gamma^t$ 만큼 적다는 것을 뜻한다. 상태 s에서 시작해서 정책, $\pi$를 따를 때의 예상 포상을 정책에 대한 상태 값 함수이라고 하고 다음과 같이 정의한다.

$$V^{\pi}(s) \equiv R(\pi(s), a) + \gamma \sum_{s \in S} T(\pi(s), a) V^{\pi}(s') \quad (4.2)$$

최적의 정책 $\pi^*$가 존재할 때 최적의 상태 값 함수는 다음과 같다.

$$V^*(s) \equiv \max_a \{R(s, a) + \gamma \sum_{s \in S} T(s, a, s') V^*(s')\}, \ \forall s \in S \quad (4.3)$$

Q학습에서는 정책 $\pi$에 대해 Q값을 다음과 같이 정의한다.

$$Q^{\pi}(s, a) \equiv R(\pi(s), a) + \gamma \sum_{s \in S} T(\pi(s), a) V^{\pi}(s') \quad (4.4)$$

Q학습에서는 Q값을 최적의 정책에 대해 근사 하는 것이 목적이다. Q값과 $V^*$의 관계식은 $V^* = \max_a Q^*(s, a)$이다. 위 식은 결정적인(deterministic) 포상과 행동인 경우와 그렇지 않은(nondeterministic) 경우 두 가지로 나누어진다. 결정적인 포상과 행동인 경우에 현재의 상태와 행동을 각각 s, $a$라 하고 다음 상태와 그 상태에서의 행동을 $s'$, $a'$이라 할 때 Q 학습의 식은 다음과 같다.

$$Q(s, a) \leftarrow r + \gamma \max_{a'} Q(s', a') \quad (4.5)$$

비결정적인 포상과 행동인 경우 Q 학습의 식은 다음과 같다.

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (4.6)$$

여기서 $\alpha$는 학습률을 나타낸다. 현재 취한 행동에 의해 다음 상태에 바뀌었을

때 그 상태에서의 Q값 중에서 최대를 나타내는 값이 있다. 이 값과 함께 전의 Q 값과 학습률, 그리고 절감률로 새로운 Q값을 정한다.

## 4.5 역할 분할 알고리즘

먼저 각 로봇들은 상황에 따른 자신의 임무를 결정해야 한다. 임무는 수비와 공격으로 나누어 질 수 있다. 임무를 결정한 후에는 자신에게 가장 알맞은 행동을 취해야 한다. 대표적으로 수비의 경우는 공을 가지고 있는 상대 로봇을 가로 막기와 공을 강하게 차는 것이 있겠고 공격수의 경우는 슛팅이 있겠다. 그리고 멀티 로봇 시스템의 가장 큰 특징인 협력을 수행하는 패스와 도움을 주기 위한 로봇의 행동을 구현할 수도 있을 것이다.

여기에서 현재 상태를 알 수 있는 방법으로 여러 가지 변수를 지정하여 환경을 결정하고 그 상태에 따른 행동을 취하여 현재에 가장 알맞은 행동을 로봇이 강화학습에 의해 찾는 것을 구현한다. 그리고 이 실효성을 실제 축구 경기를 통하여 증명한다.

## 4.6 Q-학습에 의한 행동 결정

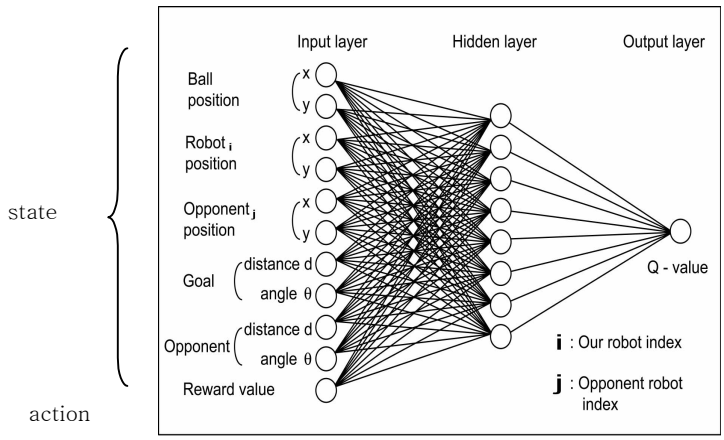



Fig 4.1 Q-net

이전 장에서 설명했듯이 Q값은 상태 s와 행동 a의 함수이다. Fig 4.2 같은 함수

관계를 구현하기 위해 s와 a에 대해 배열로 저장하기 위해서는 $S \times A$의 용량에 큰 기억 장소가 필요하다. 이렇게 저장한다면 모든 Q값이 s와 a에 대해 독립적으로 존재하므로 모든 값들은 충분히 학습시키고 설정한 모든 상황에 최선의 행동을 취할 수 있다. 하지만 이들 값들은 이산화된 상태로 구성되어야 하고 천문학적인 크기의 기억장소가 필요하다. 그러므로 본 논문에서는 상태 변수 s와 행동값 a을 입력으로 하는 신경회로망을 사용한다. Fig 4.1은 그 예를 보여준다. 각각의 값들은 0과 1사이의 정규화된 값들이다. 이 구조에서 환경을 나타내는 상태값과 행동에 대한 action값이 입력값으로 사용되고 그에 따라 Q-net을 구한다. 여기에서 환경값의 변수가 로봇의 수와 비례하여 증가한다. 예전의 3대3 경기의 경우보다도 5대5 경기는 더욱 상태값이 증가하여 상대적으로 행동값의 영향이 Q값에 적게 미치고 계산량 또한 많아 진다. 따라서 이 입력값의 수를 가능한 줄이고 정량화 될 수 있어야 한다. 본 논문에서는 입력의 수를 행동에 따라 차별화하여 구성 방법을 연구한다. 왜냐하면 신경회로망이 출력값 Q(s,a)가 각 s와 a에 대해 모두 독립적으로 계산되지 않기 때문에 어떤 $a_0$행동에 대한 학습값의 갱신에 의해 인접한 $a_1$의 행동값이 영향을 받을 수 있기 때문이다. 상태 s또한 서로 비슷한 상황 $s_0$, $s_1$에 대해 다른 상황임에도 비슷한 상황으로 인정되어 결국 학습의 효과는 비슷한 상황에 대한 것으로 효율이 떨어지게 된다.

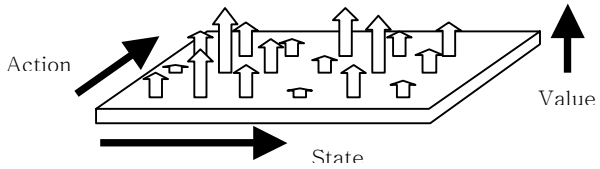


Fig 4.2 Q-value about the state and the action

### 4.6.1 슛팅 Q-net 구조

공격수의 경우 가장 중요한 상황 판단 요소는 공과 상대 골키퍼, 골대, 수비수라고 할 수 있다. 공격수는 슛만을 하도록 임무를 부여한다. 상황이 공격수-공-골대 순으로 위치하고 있다면 절대적으로 슛에 유리한 조건이다. 슛팅의 좋은 조건이 같은 편 로봇이 있다면 역할을 순간 바꾸는 방법을 쓸 수 있다. 고려되는

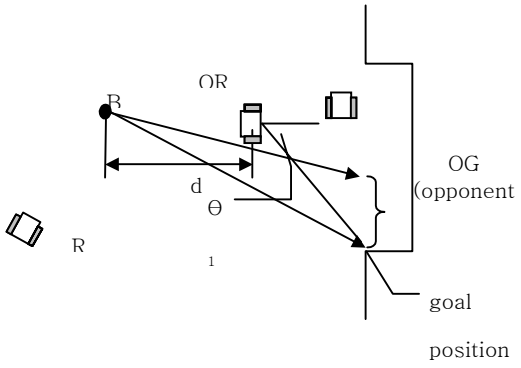
변수들은 아래와 같다.

1. B-OR-OG 또는 B-R-OG 상태 여부 및 슛팅 방향 결정

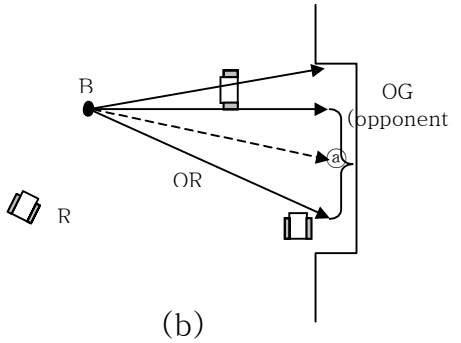
이 경우는 슛을 할 때 상대 로봇에 막힐 염려가 있다. 먼저 로봇들이 슛팅 방향 쪽 위치 한 것들만 고려한다. 그리고 나서 슛팅 방향을 잡는다. 슛팅 방향을 잡는 방법은 Fig 4. 에 나타냈다. (a)에서 슛 가능 지역은 ①과 ② 사이 영역이 되고 (b)에서는 ①과 ②, ③과 ④ 사이 영역이 된다. 슛 방향을 선택하는 기준은 최대한 장애가 될 수 있는 것이 없어야 한다. 이 조건을 만족하기 위해서는 골대 모서리 부분을 향하는 것이 가장 이상적이며 여의치 않을 때는 장애물과 상대 골키퍼 사이를 향하는 것이 좋은 방향이 된다. (a)에서는 ②를 선택하고 (b)에서는 ⓐ를 선택한다.

2. 슛 경로 장애 여부

위 조건을 만족하는 로봇이 있으면 상대 수비수나 같은 편 로봇이 슛 경로를 막고 있는지 아닌지를 판단해야 한다. 이는 슛에서 가장 중요한 상황이다. 이를 판단하는 방법으로 슛 방향과 장애물과 골 목표지점과의 방향 그리고 공과 장애물과 거리를 고려한다. Fig 4.3(a)에서 공과 장애물과의 거리, 방향을 표현하고 있다.




(a)




(b)

Fig 4.3 shoot

Fig 4.4에서는 슛팅일 때 Q-net을 보여준다. 경기가 진행되면 행동들을 선택하여 각 로봇에게 전달하는데 로봇은 이미 앞에 명령에 의해 명령을 수행하고 있을 것이다. 따라서 새로운 명령을 선택하여 보낼 시기를 결정해야 하는데 이를 상황의 변화를 감지하는 event을 정하여 지정된 event가 발행하면 새로운 상태에 대한 평가를 내려 새로운 행동을 지시한다. event값은 Table4.1에서 표현하고 있다.

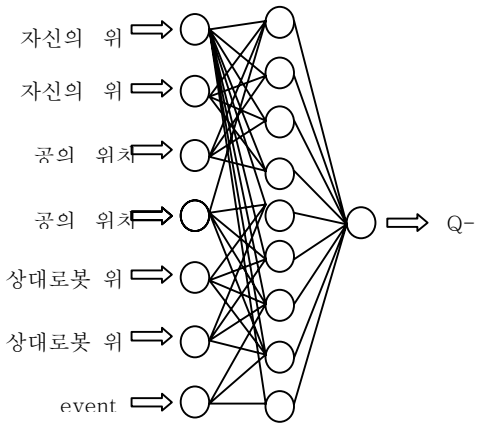


Fig 4.4 Shoot Q-net

Table 4.1 Definitions of events

| 명칭 | 설명 |
|---|---|
| hitball | 로봇과 공 충돌 |
| hitrobot | 로봇과 다른 로봇충돌 |
| arrive | 목표지점 도착 |
| Ballnear | 공이 일정거리 내에 있음 |

## 4.6.2. 수비 Q-net 구조

수비수는 공과 같은 편 골대, 같은 편 골키퍼, 상대 공격수가 주 관심 변수 값이라고 할 수 있다. 이때 고려되는 상대 로봇은 공을 잡고 있는 로봇으로 입력될 때는 공과 가장 가까운 상대 로봇이 된다.

### 4.6.3 패스 Q-net 구조

패스에서는 상대 로봇보다는 자기편 로봇의 위치가 중요하다. 즉, 고려되는 변수 대상이 달라 지는 것이다. 따라서 여기에서 Q-net의 입력값은 자신과 가장 가까운 로봇의 좌표가 사용된다. Fig 4.6에서 표현하였다.

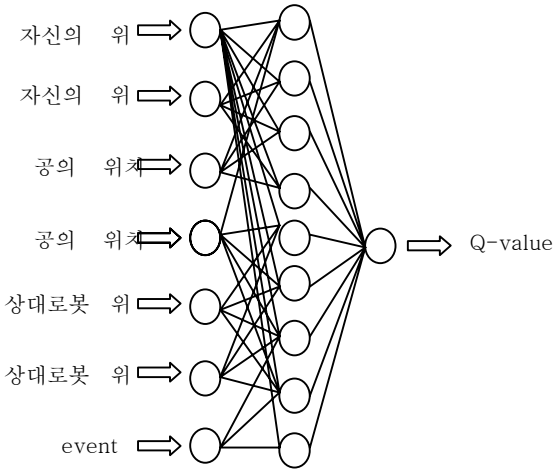



Fig 4.5 Defense of Q-net

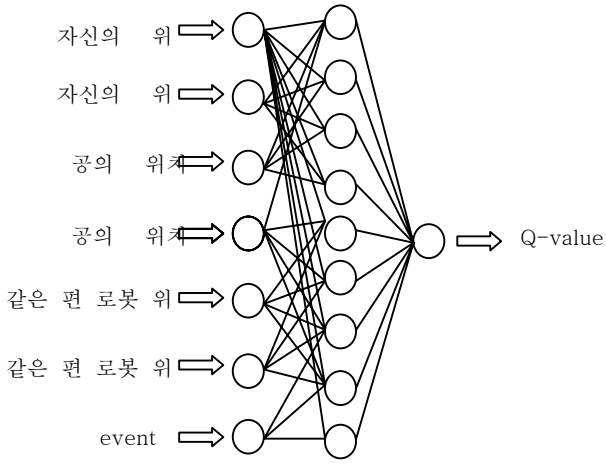



Fig 4.6 Pass Q-net

## 4.7 전체 제어기

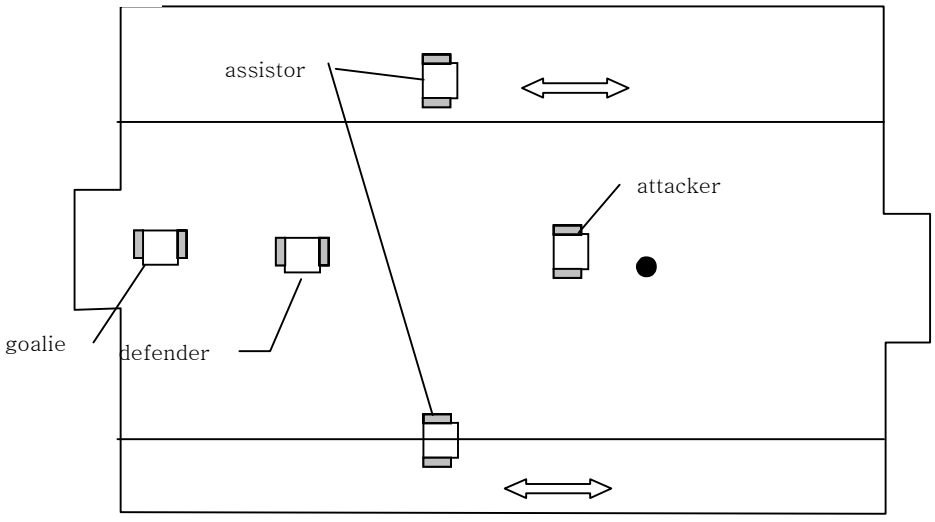



Fig 4.7 Situation ground

현재 구성된 시스템은 5 대 5 경기를 기준으로 한다. 따라서 로봇들의 협력 관계가 이전에 시스템보다 더 중요하다고 하겠다. 여기서는 공격과 수비 전용 로봇을 한대씩 지정하고 2대는 자유로이 역할을 분담하며 나머지 한대는 골키퍼를 고정으로 지정한다. Fig 4.8은 새명령이 전달 되기 전 이전 명령 수행 과정을 나타낸 것이다. 흐름도에서 4번째에서는 구해진 Q값들로부터 행동 a를 결정하기 위해서 Q값을 a에 따라 더한 누적 확률 분포 함수(cumulative probability distribution function)를 얻은 후 난수를 발생하여 이 값이 확률 분포 함수 내에 위치한 구간을 찾게 된다.이것은 강화 학습의 exploration 및 exploitation특성을 구현하기 위한 것이다. 마지막 단계에서 식 (4.6)을 적용하여 새로운 Q값을 계산한 후 이 값을 신경 회로망의 출력 층의 참 값으로 생각하고 역전파를 수행하여 회로망의 가중치를 갱신한다.

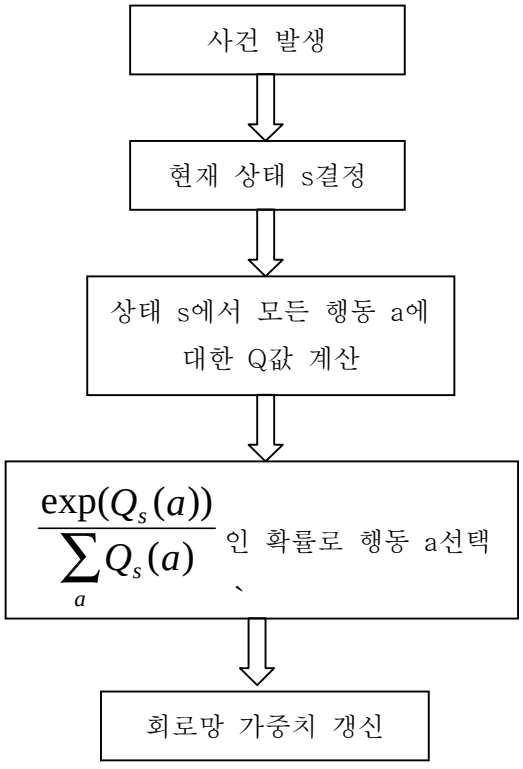



Fig 4.8 Flow chart of decision making and strategic learning

## 4.8 실험

우리는 실험을 위하여 2 대 2로써 먼저 학습시키기 위한 셋팅을 하였다. 기호로 x팀과 y팀으로 나누겠다. 로봇은 0과 1로 표시한다. 우리의 학습 대상은 x-0이다. 일단 시작되면 기초 행동 중 한가지를 선택하여 수행하게 된다. 어떠한 한 행동을 선택되어 지는 확률은 처음에는 모두 같다. 하지만 학습이 시작된 후 로봇 x-0는

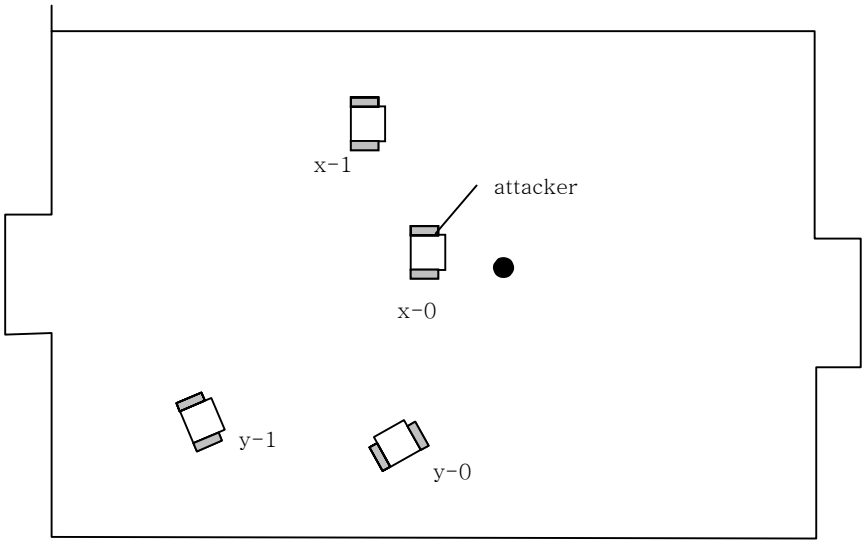
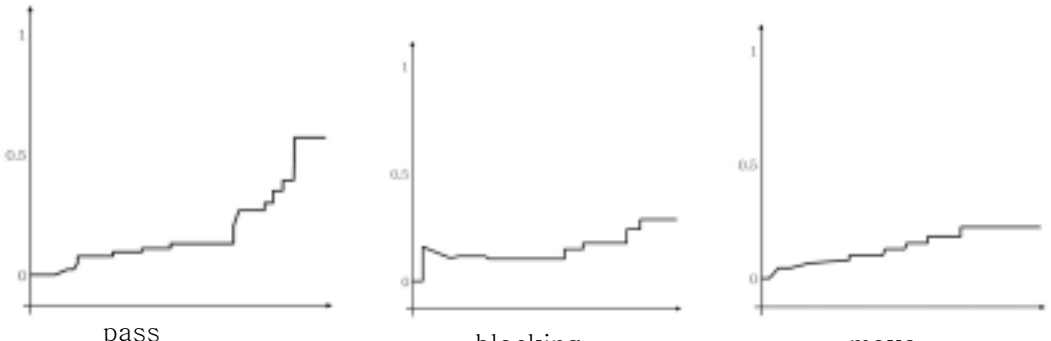



Fig 4.9 Situation shooting

직접 슛을 하여 골이 성공하면 보상 값(reward) 1을 실패하면 0을 받기 때문에 식  에 의해 새로운 Q값을 얻어진 후 회로망을 학습시키게 되므로 행동 슛팅에 대한 행동 값은 계속 커지고 Fig 4.10처럼 슛을 선호하게 된다. 결과적으로 이런 상황에서는 슛이 선택될 확률이 커지게 되어 나중에는 이 상황은 슛을 해야 하는 상황이라는 해석이 가능하게 된다.



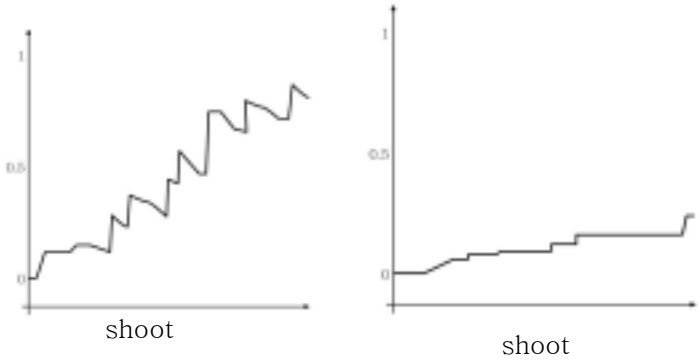



Fig 4.10 Action values versus learning epochs

# 제 5 장 결 론

본 논문에서는 로봇 축구 시스템의 각 부분별 제작, 개발하면서 이에 적용하기위한 소형 자율 이동 로봇의 효율적인 실시간 경로 계획과 이러한 기초 행동 위에 보다 고차원적인 팀전술에 대해 연구하였다. 경로 계획은 로봇의 관성과 공과의 위치 및 속도 등을 고려하고 구현하기 쉽게 하기위해 퍼지를 이용하였다. 경로 계획을 확립하여 다개체 로봇 시스템의 기초작업으로 발전된 전략 수립을 가능토록 하는 바탕이 되도록 하였다. 팀전술을 위해서는 행동의 대상이 되는 물체와 활동 공간의 특성에 따라 기초 행동을 결정하기 위해 슈팅, 이동, 수비 등의 환경에 따라 적절한 행동을 취할 수 있는 알고리즘을 제안하였다. 현 단계에서는 공과 로봇이 적절히 배치된 제한된 상황에서 축구 로봇은 자신이 처한 상황에 따라 적절한 행동을 선택할 수 있었다.

향후 경로 계획을 위해 좀더 체계적으로 퍼지 등을 통해서 구현되어야 할 것이다. 또한 강화 학습에 의한 팀전술의 학습은 실제 경기 중에 발생할 수 있는 무수히 많은 상황들에 대해 대처할 수 있어야 하기 때문에 일반적인 공, 로봇의 배치에 대해 학습이 가능하여야 하며 또한 그에 따라 오랜 시간이 소요되는 학습 속도를 빠르게 하는 것과 수렴성의 확보가 과제로 남는다. 그리고 팀전술의 학습은 기초 행동의 성능에 따라 실패할 수 있으며 Q학습을 구현하기 위한 신경 회로망을 본 문제에 맞게 구성하는 것 또한 중요한 향후 과제이다.

# 참고 문헌

[1] H.S.Shim, M.J.Jung, H.S.Kim, I.H.Choi, W.S.Han, J.H.Kim, J.H.Kim, "*Distributed Control Architecture for Cooperative Multiagent Systems*", Proc. Of Micro-Robot world cup soccer tournament(MIROSOT), 1996

[2] Peter Stone, Manuela Veloso, Sorin Achim, "*Collaboration and Learning in Robotic Soccer*", Proc. Of Micro-Robot world cup soccer tournament(MIROSOT), 1996.

[3] John Harvey, Chao Cheng, Dennis Michaelson, "*High-Level Design of a MIROSOT simulator*", Proc. Of Micro-Robot world cup soccer tournament(MIROSOT), 1996.

[4] Eiji Uchibe, Minoru Asada, Koh Hosoda, "*Strategy Classification in Multi-agent Environment-Applying Reinforcement Learning to Soccer Agent*", ICMASS'96 workshop 2: RoboCup Workshop: Soccer as a Problem for Multi-Agent System, 1996.

[5] A.H.Cai, T.Fukuda, F.Arai, and T.Ueyama, A.Sakai, "*Hierarchical control architecture for cellular robotics system*", IEEE Proc. Int.Conf. on R&A 1191-1196, 1995.

[6] Noreils, F.R "*Integrating Multi Robot Coordination in a Mobile-Robot Control System*", Proc. IROS '90, 43/49, 1991.

[7] Pape, C.L "*A Combination of Centralized and Distributed Methods for Multi-Agent Planning and Scheduling*", Proc. 1990 IEEE Int. Conf. Robotics and Automation. 488/493, 1990.

[8] Kui-Hong Park, Yong-Jae Kim, Jong-Hwan Kim, "*Modular Q-learning based multi-agent cooperation for robot soccer*", Robotics and Autonomous Systems, pp109-122, 2001.

[9] Kostas Kostiadis, Huosheng Hu, "*Reinforcement Learning and Co-operation in a Simulated Multi-agent System*", IEEE Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 990-995, 1999.

[10] Takayuki Nakamura, "*Development of Self-Learning Vision-Based Mobile Robots for Acquiring Soccer Robots Behaviors*", Proc. of the 1998 IEEE Int. Conf. on Robotics & Automations Leuven, Beigium, May 1998.

[11] R. Polesel, R.Rosati, A.Speranzon, C.Ferrari, E.Pagello, "*Using Collision Avoidance Algorithms for Designing Multi-robot Emergent Behaivors*", Proc. of the 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 1403-1409, 2000.

[12] Ou Haitao, Zhang Weidong, Zhang Wenyuan, Xu Xiaoming, "*A novel multi-agent Q-learning algorithm in cooperative multi-agnent system*", Proc. of the 3$^{th}$ World Congress on Intelligent Control and Automation, 2000.

[13] Sjerd Van Der Zwaan, Jose A. A. Moreira, Pedro U. Lima, "*Cooperative Learning and Planning for Multiple Robots*", Proc. of the 15$^{th}$ Int. Symposium on Intelligent Control(1SIC2000), Rio, Patras, Greece, 17-19 July, 2000.

[14] 김종환, "로봇 축구 시스템", 대영사

[15] Dong-Han Kim, Yong-Jae Kim and Jong-Hwan Kim, "*Nonlinear Field based Path Planning and Petri-net based Role Selection Mechanism with Q-learning for the Soccer Robot System*", FIRA 로봇축구공학 워크샵 논문집, 2000.

[16] Sung Ho Kim, Jong Suk Choi, and Byung Kook Kim, "*Development of BEST Nano-Robot Soccer Team*", Proc. of the 1999 IEEE Int. Conf. on Robotics & Automation, Detroit, Michigan, May 1999.

[17] Byoung-Ju Lee and Gwi-Tae Park, "*A Robot in Intelligent Environment : Soccer Robot*", Proc. of the 1999 IEEE/ASME Int. Conf. on Advanced

Intelligent Mechatronics, Atlanta, USA, September 19-23, 1999.

[18] M.-J. Jung, H.-S. Kim, H.-S. Shim and J.-H. Kim, "*Fuzzy Rule Extraction for Shooting Action Controller of Soccer Robot*", Proc. of the 1999 IEEE Int. Conf. on Fuzzy Systems, August 22-25, 1999.

[19] Byoung-Ju Lee, Sung-Oh Lee, and Gwi-Tae Park, "*Trajectory Generation and Motion Tracking Control for the Robot Soccer Game*", Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1999.

[20] 김광춘, 김동한, 김용제, 김종환, "축구 로봇을 위한 제어기 설계", Proc. of the 12th KACC, October 1997.

[21] 심현식, 정명진, 김홍수, 최인환, 김종환, "다개체 협력 시스템을 위한 비전 기반 축구 로봇 시스템의 개발", Proc. of the 12th KACC, Octorber 1997.

[22] 한국현, 서보익, 이택헌, 오세종, "MIRAGE I 로봇 시스템의 개선", Proc. of the 12th KACC, October 1997.

[23] Xiaoxhu Wang, Masanori Sugisaka, "*Learning Soccer Robot Using Genetic Programming*", Proc. of the 14th KACC, October 1999.

[24] S.-W. Hong, S.-W. Shin, and D.-S. Ahn, "*Formation Control Based on Artificial Intelligence for Multi-agent Coordination*", Proc. of the 2001 IEEE Int. Symposium on Industurial Electronics Proceedings, June 12-16, 2001.

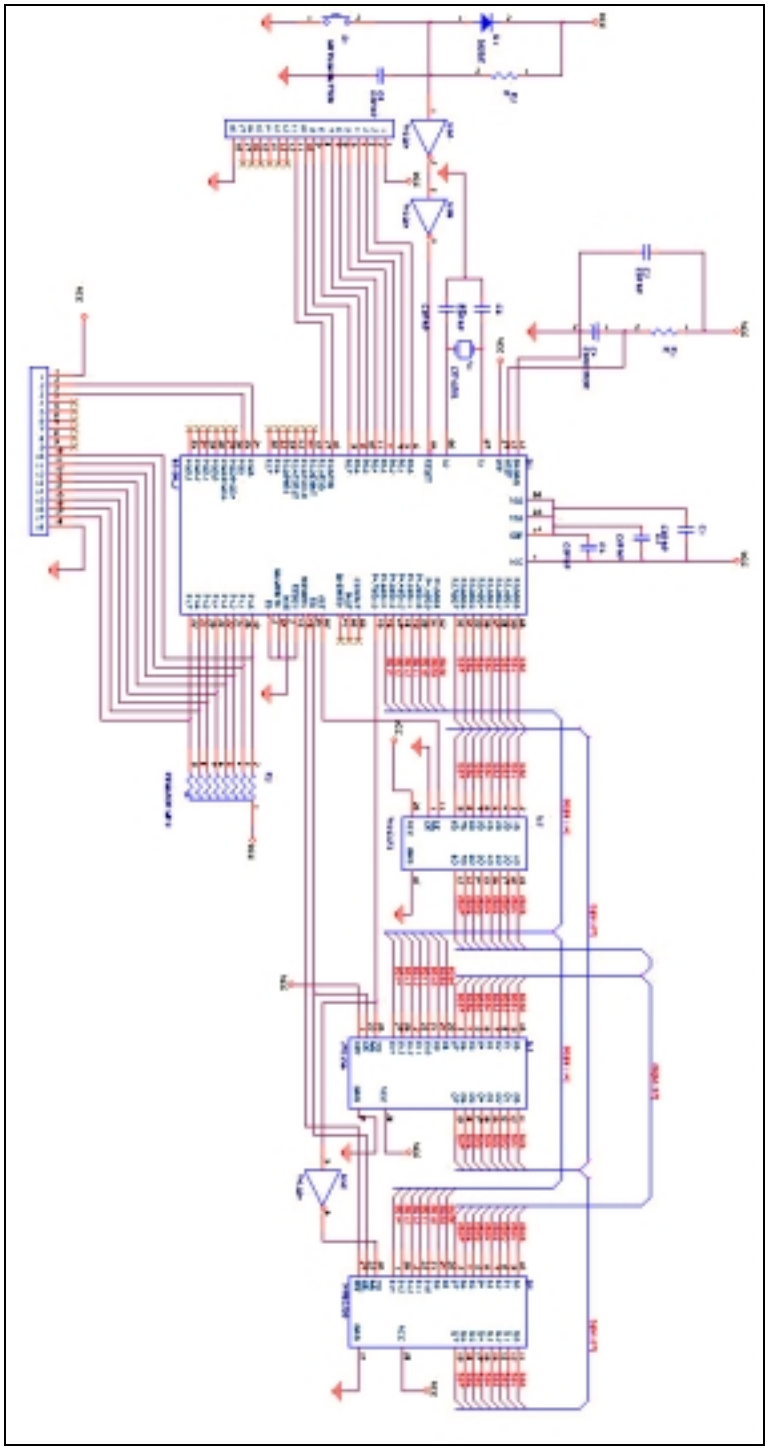
[25] S.-W. Hong, G.-J. Choi, J.-I. Bae, and D.-S. Ahn, "*A Study of Cooperative Algorithm in Multi Robots by Reinforcement Learning*", Proc. of the 2001 IEEE Int. Conf. on Control, Automation and Systems, October 17-21, 2001.

[26] RF. Communication Module Reference at "http://www.rfsolutions.co.uk/ rf_prods/ fm_bim.html"

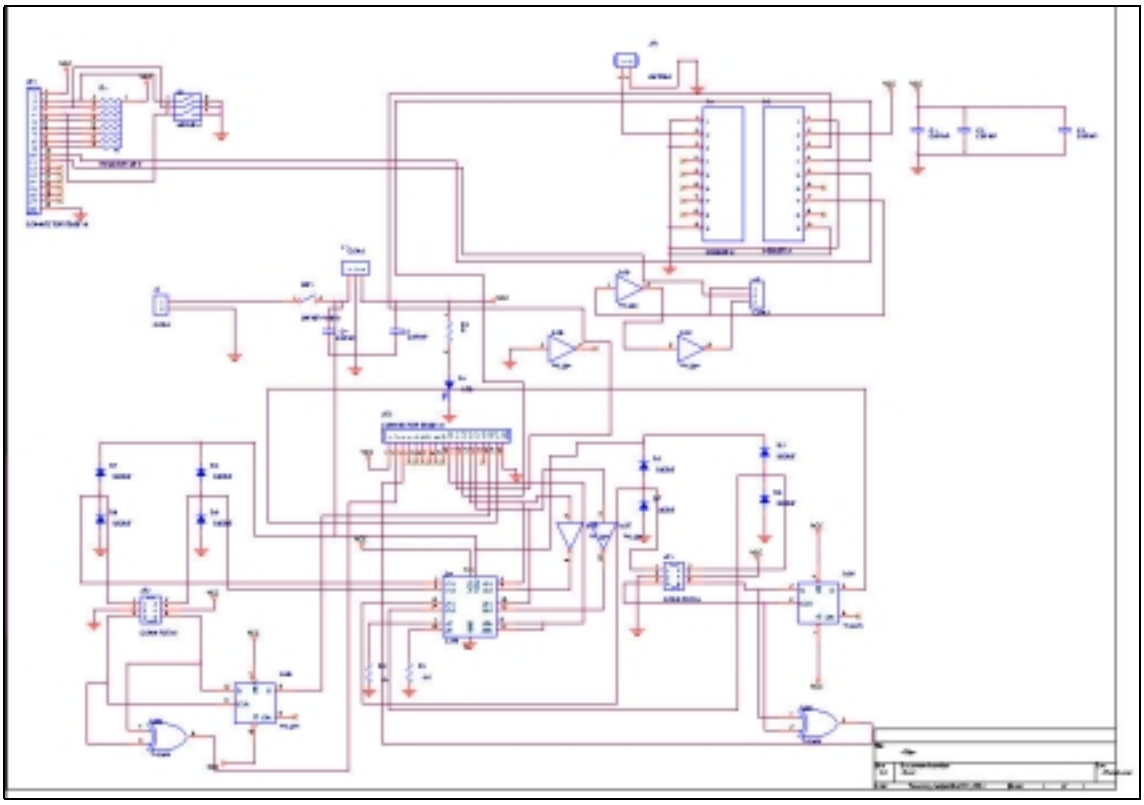
[27] Frame Grabber Reference at "http://www.matrox.com/imgprodm.htm#Me teor"

# 부 록

## A. 마이크로 컨트롤러 보드 회로도

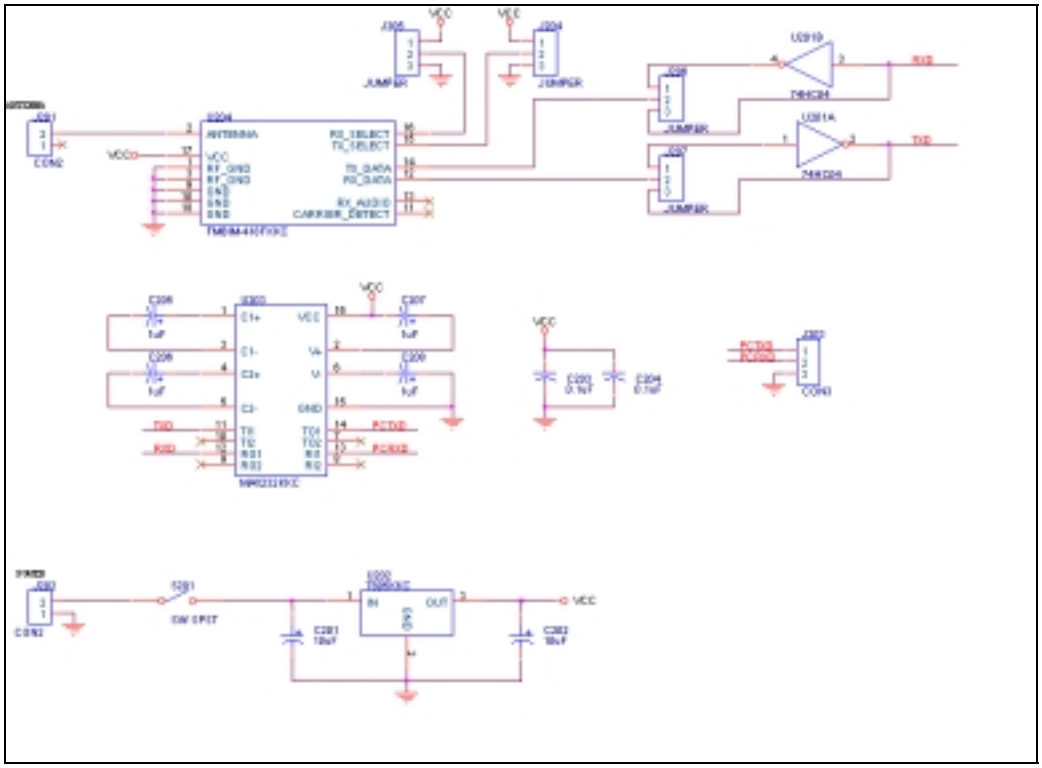


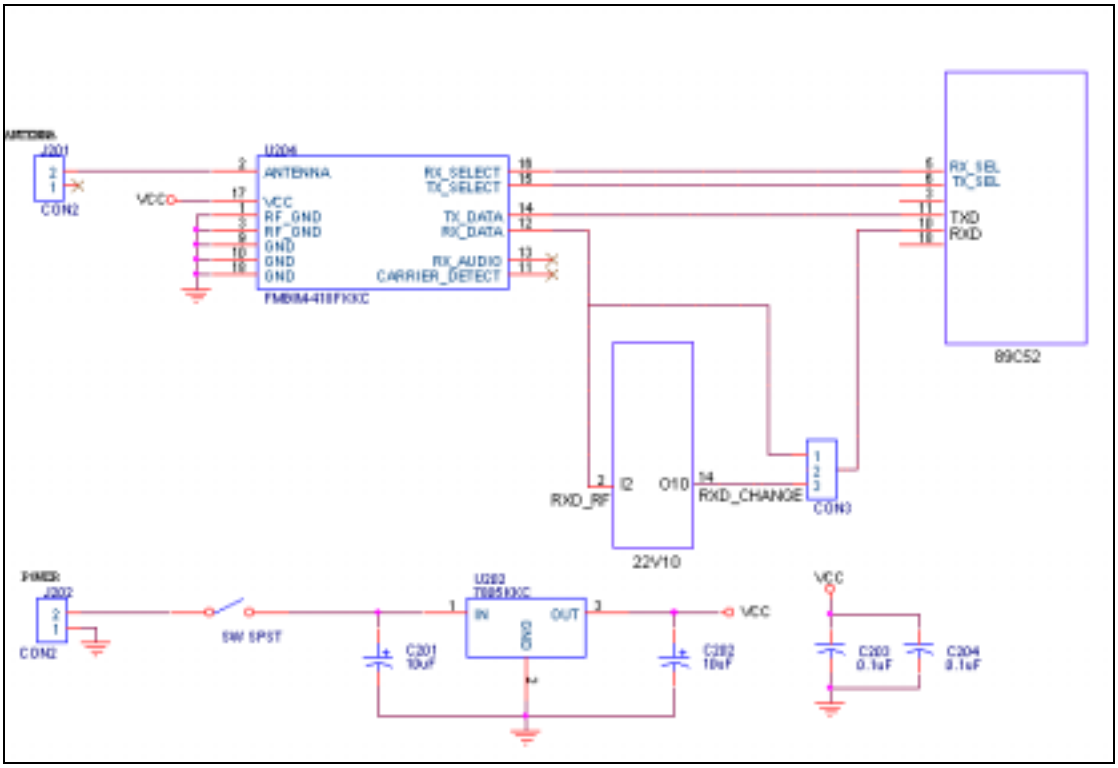
## B. 모터 구동부 회로도



## C. RF통신부 회로도

| 핀 15 (TX select) | 핀 16 (RX select) | 역할 |
|---|---|---|
| 1 | 1 | Power down(<1A) |
| 1 | 0 | Receiver enabled |
| 0 | 1 | Transmitter enabled |
| 0 | 0 | Self test loop back |

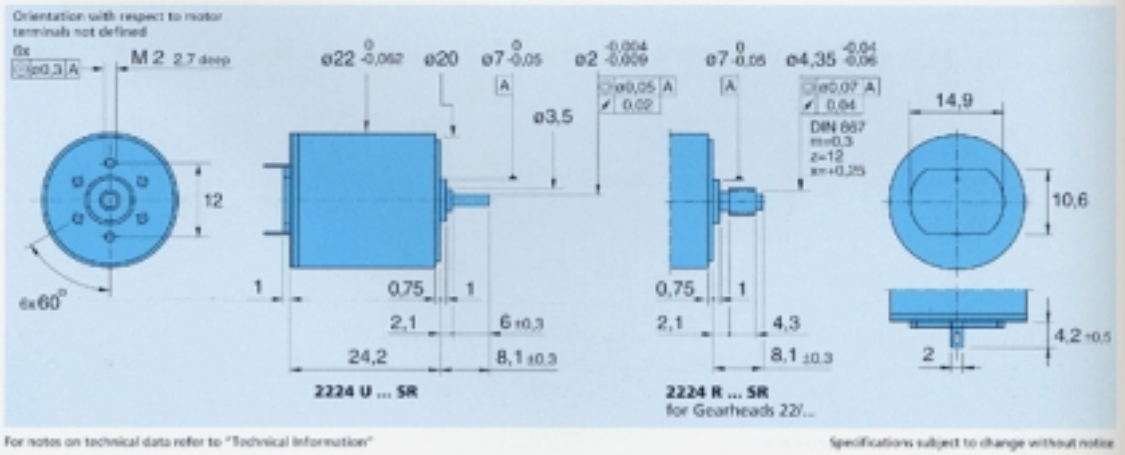



J201
CON2
U204
ANTENNA
VCC
RF_GND
RF_GND
GND
GND
GND
RX_SELECT
TX_SELECT
TX_DATA
RX_DATA
RX_AUDIO
CARRIER_DETECT
FMBIM-418FKKC
RX_SEL
TX_SEL
TXD
RXD
89C52
RXD_RF
I2
O10
RXD_CHANGE
CON3
22V10
J202
CON2
SW SPST
U202
IN
OUT
VCC
C201
10uF
C202
10uF
C203
0.1uF
C204
0.1uF

# D. DC 스위스 미니모터 2224 사양

## Series 2224 ... SR

| | | 2224 U | 003 SR | 006 SR | 012 SR | 018 SR | 024 SR | 036 SR | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Nominal voltage | $U_N$ | 3 | 6 | 12 | 18 | 24 | 36 | Volt |
| 2 | Terminal resistance | R | 0,56 | 1,94 | 8,71 | 17,50 | 36,30 | 91,40 | Ω |
| 3 | Output power | $P_{2\,max.}$ | 3,92 | 4,55 | 4,05 | 4,54 | 3,88 | 3,46 | W |
| 4 | Efficiency | $\eta_{max.}$ | 80 | 82 | 82 | 82 | 81 | 80 | % |
| 5 | No-load speed | $n_0$ | 8 100 | 8 200 | 7 800 | 8 100 | 7 800 | 7 800 | rpm |
| 6 | No-load current (with shaft ø 2,0 mm) | $I_0$ | 0,066 | 0,029 | 0,014 | 0,010 | 0,007 | 0,005 | A |
| 7 | Stall torque | $M_H$ | 18,5 | 21,2 | 19,8 | 21,4 | 19,0 | 16,9 | mNm |
| 8 | Friction torque | $M_R$ | 0,23 | 0,2 | 0,2 | 0,21 | 0,2 | 0,22 | mNm |
| 9 | Speed constant | $k_n$ | 2 730 | 1 380 | 657 | 454 | 328 | 219 | rpm/V |
| 10 | Back-EMF constant | $k_E$ | 0,366 | 0,725 | 1,520 | 2,200 | 3,040 | 4,560 | mV/rpm |
| 11 | Torque constant | $k_M$ | 3,49 | 6,92 | 14,50 | 21,00 | 29,10 | 43,50 | mNm/A |
| 12 | Current constant | $k_I$ | 0,286 | 0,144 | 0,069 | 0,048 | 0,034 | 0,023 | A/mNm |
| 13 | Slope of n-M curve | $\Delta n/\Delta M$ | 438 | 387 | 394 | 379 | 411 | 462 | rpm/mNm |
| 14 | Rotor inductance | L | 11 | 45 | 200 | 450 | 800 | 1 800 | µH |
| 15 | Mechanical time constant | $\tau_m$ | 11 | 11 | 11 | 11 | 11 | 11 | ms |
| 16 | Rotor inertia | J | 2,4 | 2,7 | 2,7 | 2,8 | 2,6 | 2,3 | gcm² |
| 17 | Angular acceleration | $\alpha_{max}$ | 77 | 78 | 74 | 77 | 74 | 74 | $\cdot 10^3$rad/s² |
| 18 | Thermal resistance | $R_{th1}$ / $R_{th2}$ | 5 / 20 | | | | | | K/W |
| 19 | Thermal time constant | $\tau_{w1}$ / $\tau_{w2}$ | 6,8 / 440 | | | | | | s |
| 20 | Operating temperature range: | | | | | | | | |
| | – motor | | – 30 ... + 85 (optional – 55 ... + 125) | | | | | | °C |
| | – rotor, max. permissible | | + 125 | | | | | | °C |
| 21 | Shaft bearings | | sintered bronze sleeves | | ball bearings | | ball bearings, preloaded | | |
| 22 | Shaft load max.: | | (standard) | | (optional) | | (optional) | | |
| | – with shaft diameter | | 2,0 | | 2,0 | | 2,0 | | mm |
| | – radial at 3000 rpm (3 mm from bearing) | | 1,5 | | 8 | | 8 | | N |
| | – axial at 3000 rpm | | 0,2 | | 0,8 | | 0,8 | | N |
| | – axial at standstill | | 20 | | 10 | | 10 | | N |
| 23 | Shaft play: | | | | | | | | |
| | – radial | ≤ | 0,03 | | 0,015 | | 0,015 | | mm |
| | – axial | ≤ | 0,2 | | 0,2 | | 0 | | mm |
| 24 | Housing material | | steel, black coated | | | | | | |
| 25 | Weight | | 46 | | | | | | g |
| 26 | Direction of rotation | | clockwise, viewed from the front face | | | | | | |
| **Recommended values** | | | | | | | | | |
| 27 | Speed up to | $n_{e\,max.}$ | 8 000 | 8 000 | 8 000 | 8 000 | 8 000 | 8 000 | rpm |
| 28 | Torque up to | $M_{e\,max.}$ | 5 | 5 | 5 | 5 | 5 | 5 | mNm |
| 29 | Current up to (thermal limits) | $I_{e\,max.}$ | 2,200 | 1,200 | 0,570 | 0,400 | 0,280 | 0,180 | A |



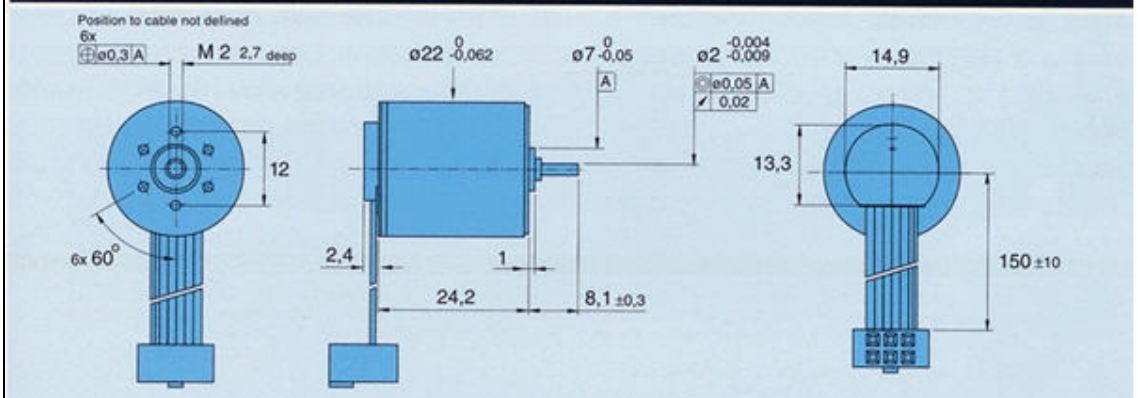

For notes on technical data refer to "Technical Information"

Specifications subject to change without notice

## Series IE2 – 512

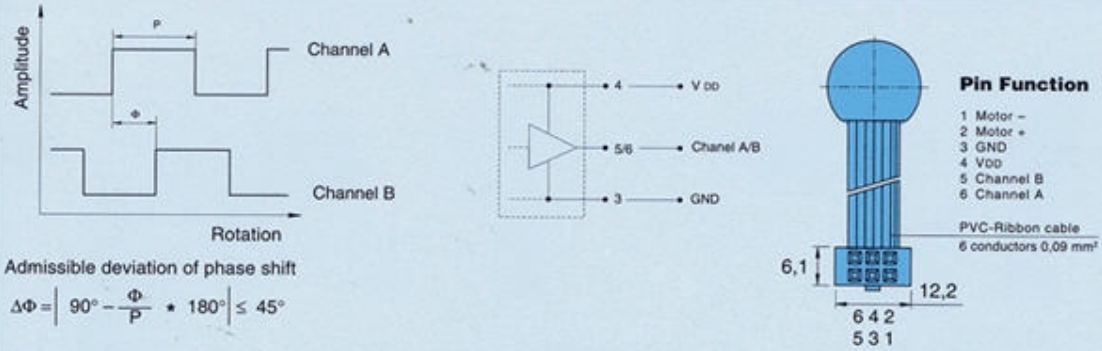
| | | IE2 – 64 | IE2 – 128 | IE2 – 256 | IE2 – 512 | |
|---|---|---|---|---|---|---|
| Lines per revolution | N | 64 | 128 | 256 | 512 | |
| Signal output, square wave | | 2 | | | | channels |
| Supply voltage | $V_{DD}$ | + 5 V + 5% | | | | V DC |
| Current consumption, typical ($V_{DD}$ = 5 V DC) | $I_{DD}$ | typ. 6, max. 12 | | | | mA |
| Output current, max. [1] | $I_{OUT}$ | 5 | | | | mA |
| Pulse width | P | 180 ± 45 | | | | °e |
| Phase shift, channel A to B | Φ | 90 ± 45 | | | | °e |
| Signal rise/fall time, max. ($C_{LOAD}$ = 50 pF) | tr/tf | 0,1 / 0,1 | | | | µs |
| Frequency range [2], up to | f | 20 | 40 | 80 | 160 | kHz |
| Inertia of code disc | J | 0,09 | | | | gcm² |
| Operating temperature range | | – 25 ... + 85 | | | | °C |

[1] $V_{DD}$ = 5 V DC: Low logic level < 0,5 V, high logic level > 4,5 V: CMOS and TTL compatible
[2] Velocity (rpm) = f (Hz) x 60/N




**DC-Micromotor 2224 U ... SR with integral Encoder IE2 – 64 ... 512**

### Output signals / Circuit diagram / Connector information




Admissible deviation of phase shift

$$\Delta\Phi = \left| 90° - \frac{\Phi}{P} * 180° \right| \leq 45°$$

**Output signals**
when direction of rotation as seen from the shaft end is clockwise

**Output circuit**
Note: Motor terminal resistance increases by approx. 0,4 Ω.

**Connector**
DIN-41651
grid 2,54 mm

# 감사의 글

학문의 뜻을 품고 실헐실에 몸을 담은지 엇그제 같은데 벌써 2년이라는 시간이 지나 그 결실을 보게 되었습니다. 처음에 무작정 시작하면 무엇이든 이루어 질 것이라 믿고 이리 저리 도전하였던 지난 시간은 많은 보람도 있었지만 또한 후회와 부끄러움으로 가득한 시간이기도 합니다. 먼저 지난 2년 동안 많은 것을 가르쳐 주시고 무지의 저를 일깨워 주신 안두성 교수님께 먼저 감사의 말씀을 드리고 싶습니다. 마지막까지 좋은 논문이 완성될 수 있도록 도와 주신 백운경 교수님, 권순재 교수님께 고개 숙여 감사드립니다. 실험실에 큰형님이신 배효정 형님, 힘들 때마다 형님처럼 도와 주신 규종이 형, 처음 시작하는 저에게 많은 도움을 주셨던 상후형, 성재형 모두에게 감사의 인사를 드립니다. 나의 동기이자 친한 친구이기도 한 광수형, 경수 때문에 실험실 생활을 더욱 재미있고 활력있게 했던 것 같습니다. 모두에게 감사드립니다. 지난 몇 달간 나의 연구를 도와 가며 밤낮을 공생했던 종철이, 동훈이 에게도 고맙다는 말을 전합니다.

마지막으로 나의 사랑하는 가족에게 이 논문을 바칩니다.