

工學碩士 學位論文

인터넷을 이용한 모빌로봇
제어에 관한 연구

指導教授 安斗星

이 論文을 工學碩士 學位論文으로 提出함

釜慶大學校 大學院

機械工學科

朴珖秀

目 次

Abstract	4
1. 서 론	6
2. 시스템 개발 환경	8
2.1 인터넷 기반의 원격제어 시스템	8
2.2 모빌로봇 IP	8
2.3 무선 통신 장비	10
2.4 원격 관리 클라이언트 프로그램	11
2.5 데이터 전송	12
2.5.1 프로토콜	12
2.5.2 TCP/IP 모델	13
2.6 인터넷	18
3. Web Based Mobile Robots Control System	23
3.1 시스템 개요	23
3.2 Java언어	24
3.2.1 Java의 장점	27
3.3. Graphic User Interface (GUI)	31
3.3.1 Login Module	32
3.3.2 Virtual Representation Module	32
3.3.3. Video Feedback Module	35
3.3.4 Chat Module	35

3.4 Networking	37
3.4.1 Socket connect	38
3.4.2 Java Applet	38
3.4.3 Internet Transmission Latency	39
3.5 Mobile Robots Control	40
3.5.1 Robot Soccer Mobile Robot	40
3.3.2 Micro Controller	43
3.5.3 모터부	44
3.5.4 통신부	46
3.5.5. Vision System	48
4. 실험 및 결과	50
4. 1 Fuzzy Logic Based Controller를 사용한 모빌로봇의 이동	50
4. 2 두 대의 모빌로봇을 이용한 목표물 이동	60
5. 결론 및 향후 연구 과제	66
참고문헌	68

Interacting Mobile Robots for Tele- Operation System Using the Internet

Kwang-Soo Park

Department of Mechanical Engineering, Graduate School

Pukyong National University

Abstract

Today robot system is applied to wide areas, such as automation, space reconnaissance and a military satellite. In particular, artificial intelligence theory such as fuzzy, neural network and genetic algorithm made it possible to give self-control to robots to carry out tasks for itself. In addition, many researches are in progress not only in robot areas for special purposes, but in daily-life areas such as medical assistance or gardening. Though robots hold their own self-control

abilities to some extent, however, we need to observe their condition and surroundings around them. Furthermore, when a robot manager is in a remote place, it needs to recognize information through communication backgrounds like the Internet network. The development of the Internet enables a variety of multimedia service including voice and video materials. The Internet-based interactive meeting on the computer screen and remote control system now available, are also noteworthy. With the help of Internet-based programming language, robot systems are getting more widely applied to the Internet.

This paper discusses the interacting mobile robots for tele-operation system using the internet. The main components of this paper are user interface, networking, simulation, and robot control. The standard graphic user interface (GUI) is implemented using Java Programming language. The web browser is used to integrate the virtual environment and the standard GUI (Java applet) in a single user interface. Users can access a dedicated WWW server and download the user interface.

Java application has been developed to communicate and control the robot using a RF communication. In addition to user's direct control, a fuzzy logic control is applied. We propose the use of FLC (fuzzy logic controllers) that incorporate expert knowledge in term of linguistic rules.

1. 서 론

오늘날 로봇시스템은 공장 자동화, 우주탐사, 군사 등 광범위한 분야에서 응용되고 있다. 특히 퍼지, 신경회로망, 유전자 알고리즘 등의 인공지능 이론은 로봇에게 자율성을 부여하여 로봇 혼자서도 주어진 임무를 수행 할 수 있는 단계에 이르렀다. 비단 특수 목적의 로봇 분야뿐만 아니라 청소, 의료보조, 정원관리 등 일상 생활에서 응용될 수 있는 연구도 진행되고 있다. 그러나, 로봇이 어느 정도의 자율성을 확보하고 있더라도 로봇의 상태와 주변환경을 관찰할 필요가 있다. 더구나 로봇 관리자가 원격지에 있을 경우 네트워크 등 통신환경을 구축하여 정보를 인식할 필요성이 있다. 인터넷의 발전은 문자에서 벗어나 음성, 영상 등의 멀티미디어 서비스를 제공하고 있다. 현재 양방향 화상회의, 원격 감독 시스템 등 인터넷 기반의 응용 분야도 주목 할 만 하다. 또한 인터넷 기반의 프로그래밍 언어로 인해 인터넷에서의 응용분야의 범위는 더욱 확대되어 가는 추세이다.

본 논문에서는 웹 브라우저를 사용하여 클라이언트에서 서버측과 연결된 이동로봇을 컨트롤 한다. 인터넷을 사용하지 않은 기존의 로봇 컨트롤 시스템으로는 전용의 컨트롤 설비가 필요하게 되고 거리와 위치에 제한을 받게 된다. 하지만 인터넷을 이용하여 모빌로봇을 제어하게 되면 거리와 위치에 상관없이 웹 브라우저가 있는 어느 곳에서도 모빌로봇을 제어할 수 있다는 장점이 있다.

본 논문의 주안점은 웹 기반의 환경에서 이동로봇을 보다 정확하게

제어하는 것이다. 기존의 웹 기반 로봇제어와 다른점은 소형 모빌로봇을 제어 한다는 점과, 비전데이터를 이용 한다는 점, Java Applet으로 컨트롤 판넬을 GUI상에 설계 했다는 점, Fuzzy 컨트롤러를 사용하여 Gain값을 조절하여 제어 했다는 점, VRML의 사용으로 로봇의 제어 상태를 미리 확인해 볼 수 있다는 점, 두 대의 로봇으로 하나의 목표물을 이동시켰다는 점 등이 되겠다.

본 논문에서는 두 가지 실험을 하였다. 첫번째는 로봇 한대가 목표위치까지 장애물을 피하여 보다 빠른 속도로 이동하는 방법으로서 로봇에 부착된 로봇 컨트롤러는 Gain값이 고정된 PID제어가 아닌 Gain값 조절이 가능한 FUZZY제어 알고리즘을 사용하여 보다 빠르게 사용자가 원하는 로봇의 위치와 방향으로 로봇이 이동해 가는 방법을 생각해 보았다. 두 번째는 로봇 두 대를 사용하여 목표위치까지 볼을 이동시키는 실험을 하였다.

본 실험에 사용된 이동로봇은 국내에서 활성화 되고 있는 로봇축구용 모빌로봇을 사용하였고, CPU는 80196을 사용하였다. 로봇과의 통신은 RF모듈을 사용하였고, 로봇의 위치는 비전 시스템을 사용하여 확인하였다.

본 논문 2장에서는 시스템의 개발환경에 대하여 살펴보고, 3장에서는 실제 적용된 웹 컨트롤 시스템의 구조와 특징에 대하여 살펴보고, 4장에서는 실제 로봇을 제어하는 실험을 하여보았고, 마지막으로 5장에서는 결론과 향후 연구과제에 관해 기술 하였다.

2. 시스템 개발 환경

2.1 인터넷 기반의 원격제어 시스템

본 연구에 있어서 인터넷 기반의 원격 제어 시스템의 구조는 크게 4부분으로 나눌 수가 있다. 먼저 최종 제어 대상인 로봇과 그 로봇을 무선관리 하는 웹 서버, 또한 로봇의 움직임 감시를 위해 서버와 통신하는 클라이언트, 마지막으로 데이터 전송 수단인 인터넷이다.

최종 제어 대상은 모빌 로봇이 되고 로봇과 무선으로 통신하기 위해 인터넷에 유선으로 연결되어 있는 웹 서버가 하나 있다. 그리고 사용자는 인터넷을 이용하여 PC프로그램인 클라이언트에서 웹 서버로 접속을 시도하여 제어 감시 할 수 있는 것이다.

즉, 원격지에 있는 사용자는 인터넷에서 프로그램을 실행하여 실제의 제어 대상과 통신하기 위해 중앙에 미들 서버를 거쳐야 한다는 뜻이다. 유선으로 제어시스템을 구현한다면 통신 선이 설치된 환경에 제약을 받을 뿐만 아니라 전원 선과 통신 선이 동시에 연결 되어야 하기 때문에 시스템 최종 구현에 있어서 복잡성을 초래하게 된다.

2.2 모빌로봇 IP

본 논문에서의 관리대상 시스템은 바로 로봇이다. 이 로봇 내부에는 고유의 IP주소를 갖는 Socket서버로 존재하며 이 IP는 웹 서버에 의해서 영향을

받는다.

즉, 사용자가 접속하는 인터넷 상에 존재하는 IP는 웹서버 IP이다. 이런 식으로 시스템을 구성하게 되면 이론상 로봇에는 한 개의 IP주소가 부여되어야 한다. 그러나 실제 시스템에 있어서는 무선데이터를 직접 받는 부분에 IP주소가 한 개가 있고, 모빌로봇 제어 부분에 또 한 개의 IP 주소가 있다.

모빌로봇은 프로그램 실행 시 로봇 전체를 초기화 하며 소켓을 형성하고 연결이 오기를 기다리고 있다. 모빌로봇은 또한 명령에 따른 데이터를 분석 실행하며 자신이 처한 상황이나 내부 데이터를 사용자에게 보내주는 역할을 수행한다.

인터넷을 사용하여 모빌로봇과 서버가 데이터를 주고 받으려면 TCP/IP전송을 사용한다. IP Address를 식별하여 데이터를 전송하게 된다. 인터넷 상에서 물리적인 네트워크 주소와 일치하는 개념으로 부여된 32비트의 주소가 IP 주소이다. IP 주소를 이용하면 네트워크상의 유일한 호스트를 식별하는 것뿐만 아니라, 호스트가 있는 네트워크를 식별할 수 있다. IP 주소는 클래스로 나뉘어 있으며 하나의 네트워크에서 모든 호스트는 동일한 prefix를 공유한다.

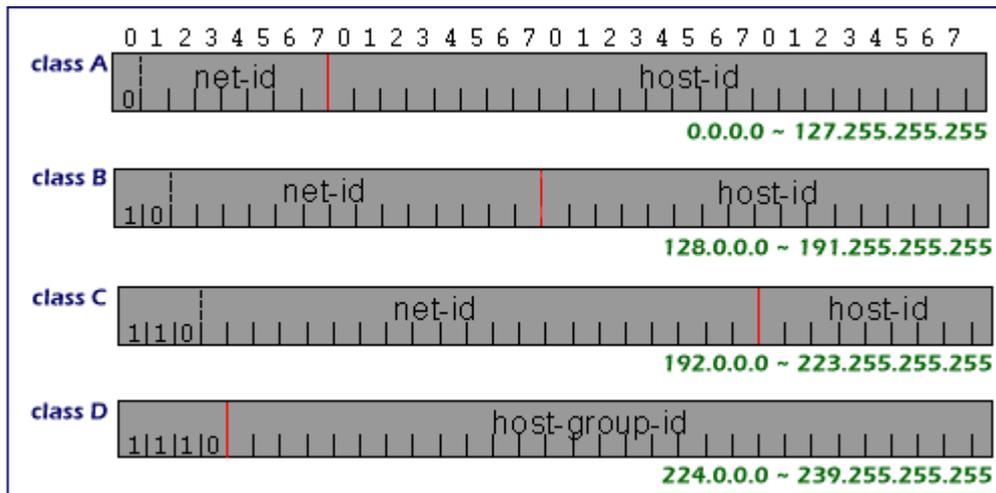


Fig. 1 IP address

클래스 A는 7 비트의 netid와 24 비트의 hostid로 나뉘어 있으므로 하나의 네트워크에 2^{16} 개보다 많은 호스트가 존재할 때 사용할 수 있다. 클래스 B는 $2^8 \sim 2^{16}$ 개 사이의 중간 크기 네트워크에서 사용할 수 있으며, 클래스 C는 2^8 개 호스트 이하의 네트워크에서 사용할 수 있다. 그러나 상대적으로 클래스 A와 B는 할당할 수 있는 네트워크 수가 적으므로 그 주소가 고갈되어 최근에는 클래스 B 크기의 네트워크에 클래스 C를 여러 개 할당해주는 방법을 취하고 있다. 또한 현재 IP (IPv4)의 32 비트 주소 체계로는 전세계의 증가하는 호스트에 주소를 할당하기 어렵기 때문에, 128 비트의 주소체계를 쓰는 IPv6의 연구가 활발히 진행 중이다.

2.3 무선 통신 장비

무선 통신 장비는 사용자가 원격지에서 연결하여 데이터를 보내면 TCP/IP 데이터를 RS232의 시리얼 데이터로 변환해주고 또한 반대로 로봇 쪽에서

시리얼 데이터를 받으면 사용자 쪽으로 TCP/IP 데이터로 변환 시켜주는 역할을 한다. 본 장비는 사용자가 접속 중에 있을 때, 다른 곳에서 또 다른 접속 시도가 있을 경우 기존의 연결을 해지하고 최후의 연결 시도에 연결하게 된다. 이것은 소프트웨어적으로 조절 할 수 있는 문제인데 다중 사용자일 경우 우선순위를 정해 접속 가능하게 할 수 있다.

2.4 원격 관리 클라이언트 프로그램

사용자는 클라이언트 프로그램을 이용하여 로봇에 연결을 할 수 있다. 클라이언트 프로그램은 Java Applet을 사용하여 만들었으며 사용자가 컨트롤 시스템에 접속하기 위한 연결기능과 로봇에 명령을 내리기 위한 기능, 또한 로봇에서 전송되어 오는 데이터를 표시할 수 있는 모니터링 기능을 내장하고 있다.

클라이언트 프로그램은 사용자의 요구에 맞춰 보정 되어야 하며 다른 시스템에 적용하게 되면 그에 맞게 다시 수정을 하여야 한다. 본 논문에서는 로봇의 기능에 맞추어 프로그램을 작성하였으며, 그 기능으로는 클라이언트측에서 Applet 컨트롤 패널 상에 목표 위치를 지정하면 그곳으로 로봇이 이동하게 되는 기능과, 수동 로봇 조작으로서 로봇동작에 있어서 전진, 좌회전, 우회전, 후진 기능을 갖추었고 로봇의 움직임에 따른 속도의 정보를 받아 표시하는 기능과 로봇이 이동하게 될 곳까지의 이동정보를 이미지로서 보여주는 기능이 있다.

2.5 데이터 전송

컴퓨터가 널리 보급되고 사용자의 폭이 넓어지면서 흩어져 있는 컴퓨터 간의 자원을 공유하기 위해 발전된 통신 기능은 데이터 전송 뿐만 아니라 프린터와 스캐너, CD-ROM과 같은 입, 출력 장치 그리고 각종 소프트웨어까지 공유 할 수 있게 되었다. 예전에는 단일 컴퓨터에서 혼자만 즐겨왔던 게임도 이제는 다중 사용자가 동시에 참여하는 네트워크 게임이 주류를 이루고 있다. 즉, 특수한 영역으로만 생각했던 네트워크 프로그래밍이 이제는 컴퓨팅 환경의 필수로 보편화 된 것이다. 이를 반영하듯 요즘 발표되는 대부분의 운영체제는 네트워크 기능을 기본적으로 내장하고 있다.

Windows는 소규모 네트워크에서 사용되는 NetBEUI(NetBios Extended User Interface), TCP/IP, 노벨의 NetWare환경에서 사용되는 IPX/SPX 호환 프로토콜이 내장되어 있다. 이들 프로토콜에 인터페이스 할 수 있는 윈도우 소켓과 NetBIOS(Network Basic Input Output System)을 제공한다.

2.5.1 프로토콜(Protocol)

프로토콜이란 서로 다른 기종의 컴퓨터나 장치 사이에서의 데이터 교환을 위해 사전에 결정해야 할 여러 물리적, 소프트웨어적인 조건들을 표준화 시킨 규칙 또는 규약의 집합이다. 이러한 규약에는 상대방의 요청 및 응답, 통신의 연결 및 해제, 메시지의 블록화 형식, 오류 발생시의 처리 방법, 각종 코드 변환 등이 포함된다. 프로토콜은 그 자체만으로는 어떤

소프트웨어나 하드웨어가 아니고 단지 약속 일 뿐이다. 국제 표준화 기구 (ISO : International Standards Organization) 모델이 있는데 이 실험에서는 현재 인터넷에서 기본적인 네트워크로 사용되는 TCP/IP 모델을 기준으로 삼았다.

2.5.2 TCP/IP 모델

인터넷에서 가장 널리 사용되는 TCP/IP는 단 두 대의 컴퓨터를 연결하는데도 사용된다. TCP/IP(Transmission Control Protocol / Internet Protocol)는 1969년에 미 국방성의 첨단 연구 계획 사업부(Advanced Research Projects Agency)에서 ARPANET이라 불리는 자원 공유 실험용으로 개발 되었다. 패킷 교환 네트워크를 이용해서 고속 통신 접속을 제공하는 ARPANET(현 DARPA NET)은 현재 인터넷 발전의 중심이 된다. TCP/IP는 단순히 TCP프로토콜과 IP 프로토콜에 국한되는 용어가 아니라, 다른 프로토콜(ARP, RARP, ICMP, UDP등)들을 포함하는 서비스를 포함하는 말이다. TCP/IP는 UNIX를 비롯해서 Windows시스템에서 기본적으로 제공되며, 가장 널리 사용되는 프로토콜이고 그만큼 다양한 서비스를 가지고 있다. TCP/IP는 2개의 계층으로 이루어진 프로그램이다. 상위계층인 TCP (Transmission Control Protocol)는 메시지나 파일들을 좀더 작은 패킷으로 나누어 인터넷을 통해 전송하는 일과, 수신된 패킷들을 원래의 메시지로 재조립하는 일을 담당한다. 하위계층, 즉 IP (Internet Protocol)는 각 패킷의 주소부분을 처리함으로써, 패킷들이 목적지에 정확하게 도달할 수 있게 한다.

네트워크 상의 각 게이트웨이는 메시지를 어느 곳으로 전달해야할지를 알기 위해, 메시지의 주소를 확인한다. 한 메시지가 여러 개의 패킷으로 나뉘어진 경우 각 패킷들은 서로 다른 경로를 통해 전달될 수 있으며, 그것들은 최종 목적지에서 재조립된다.

TCP/IP는 통신하는데 있어 클라이언트/서버 모델을 사용하는데, 컴퓨터 사용자(클라이언트)의 요구에 대응하여, 네트워크 상의 다른 컴퓨터(서버)가 웹 페이지를 보내는 식의 서비스를 제공한다. TCP/IP는 본래 점대점(點對點) 통신을 하는데, 이는 각 통신이 네트워크 상의 한점(또는 호스트 컴퓨터)으로부터 시작되어, 다른 점 또는 호스트 컴퓨터로 전달된다는 것을 의미한다. TCP/IP와 TCP/IP를 이용하는 상위계층의 응용프로그램들은 모두 "커넥션리스 (connectionless)"라고 불리는데, 이는 각 클라이언트의 요구가 이전에 했던 어떠한 요구와도 무관한 새로운 요구로 간주된다는 것을 의미한다 (일상적인 전화통화가 통화시간 내내 지속적으로 연결되어 있어야 하는 것과는 다르다). 커넥션리스는 네트워크를 독점하지 않으므로, 모든 사람들이 그 경로를 끊임없이 공동으로 사용할 수 있게 한다 (사실 TCP 계층 그 자체는 어떤 한 메시지가 관계되어 있는 한 커넥션리스가 아니라는데 유의해야 한다. TCP 접속은 어떤 한 메시지에 속하는 모든 패킷들이 수신될 때까지 계속 유지된다).

많은 인터넷 사용자들이 TCP/IP를 이용하는 상위계층 응용프로토콜에 대해서는 잘 알고 있다. 이러한 상위계층 프로토콜에는 웹서비스에 사용되는 HTTP를 비롯하여, 멀리 떨어져 있는 원격지의 컴퓨터에 로그인할 수 있게

해주는 Telnet (Telnet), 그리고 파일전송에 사용되는 FTP (File Transfer Protocol)와 메일 전송에 사용되는 SMTP (Simple Mail Transfer Protocol) 등이 있다. 이러한 프로토콜들은 종종 TCP/IP와 함께 패키지로 일괄 판매된다.

PC 사용자들은 보통 인터넷에 접속하기 위해 SLIP (Serial Line Internet Protocol)이나 PPP (Point-to-Point Protocol) 프로토콜을 사용한다. 이러한 프로토콜들은 다이얼업 전화접속을 통해 접속서비스사업자의 모뎀으로 보내질 수 있도록 IP 패킷들을 캡슐화한다.

TCP/IP와 관련이 있는 프로토콜로 UDP (User Datagram Protocol)가 있는데, 이것은 특별한 목적을 위해 TCP 대신에 사용되는 것이다. 라우팅 정보를 교환하기 위해 네트워크 호스트 컴퓨터에 의해 사용되는 프로토콜에는 ICMP (Internet Control Message Protocol), IGP (Interior Gate TCP/IP way Protocol), EGP (Exterior Gateway Protocol), 그리고 BGP (Border Gateway Protocol) 등이 있다. 이 외에도 통신 프로토콜에는 XNS(Xerox Networking System), SNA(IBM's System Network Architecture), UUCP(Unix to Unix Copy)등이 있다.

통신 기능은 일련의 계층 집단으로 분할되며, 각 계층은 다른 시스템과 통신하려는 데에 필요한 관련된 기능을 수행한다. 각 계층은 그 기능들의 세부 내용을 은폐하고 보다 원시적인 기능을 수행하기 위하여 바로 아래에 있는 계층에 의존한다.

Fig. 2는 OSI(Open System Interconnection)모델과 TCP/IP 모델을 비교한

그림이다. 각 계층별로 하는 기능을 간략히 설명하면 다음과 같다.

물리계층(Physical Layer)

IEEE가 기존에 정의한 표준을 따르고 있으며 OSI 모델의 물리 계층과 데이터 링크 계층에 해당한다.

인터넷 계층(Internet Layer)

데이터그램(IP 프로토콜에서 다루는 패킷 데이터)을 정의하고 라우팅 하는 역할을 수행한다. 데이터그램은 보낸 주소(Source address), 받을 주소(destination address), 데이터, 그 외 제어 필드를 가지고 있다. OSI 모델의 네트워크 계층에 해당한다.

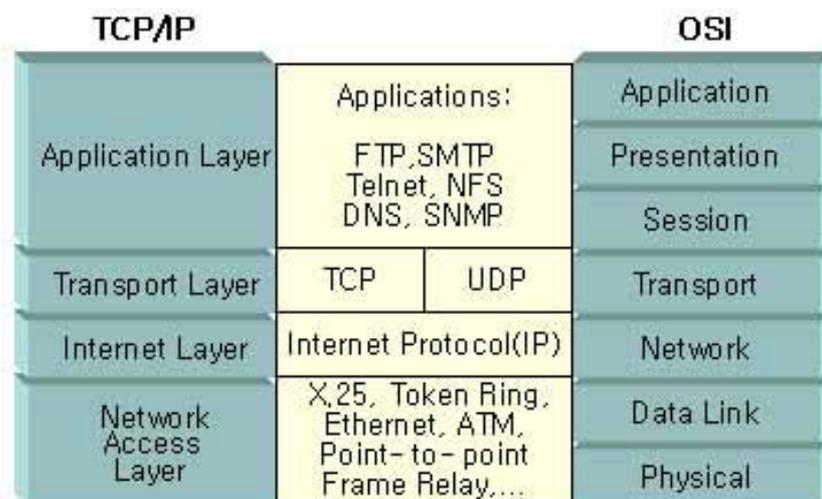


Fig. 2 OSI 모델과 TCP/IP 모델

트랜스포트 계층(Transport Layer)

도착해야 할 컴퓨터로 데이터를 전송하는 역할을 한다.

어플리케이션 계층(Application Layer)

FTP, Telnet 프로그램등이 여기에 속하며, OSI 모델의 세션, 프리젠테이션, 어플리케이션 계층에 해당한다.

Fig. 3은 본 연구에서 로봇과 호스트 사이에 데이터가 전송되는 과정을 도식화 한 것이다.

윗 계층에서 아래 계층으로 내려올 때 마다 각 계층에서 필요한 헤더가 실제 데이터에 결합된다. 이렇게 하여 물리 계층까지 내려간 데이터는 물리적 전송로를 따라 수신측 시스템으로 이동하게 된다. 이 과정에서 데이터는 많은 라우터나 브리지를 경유하여 수신 시스템에 도착한다.

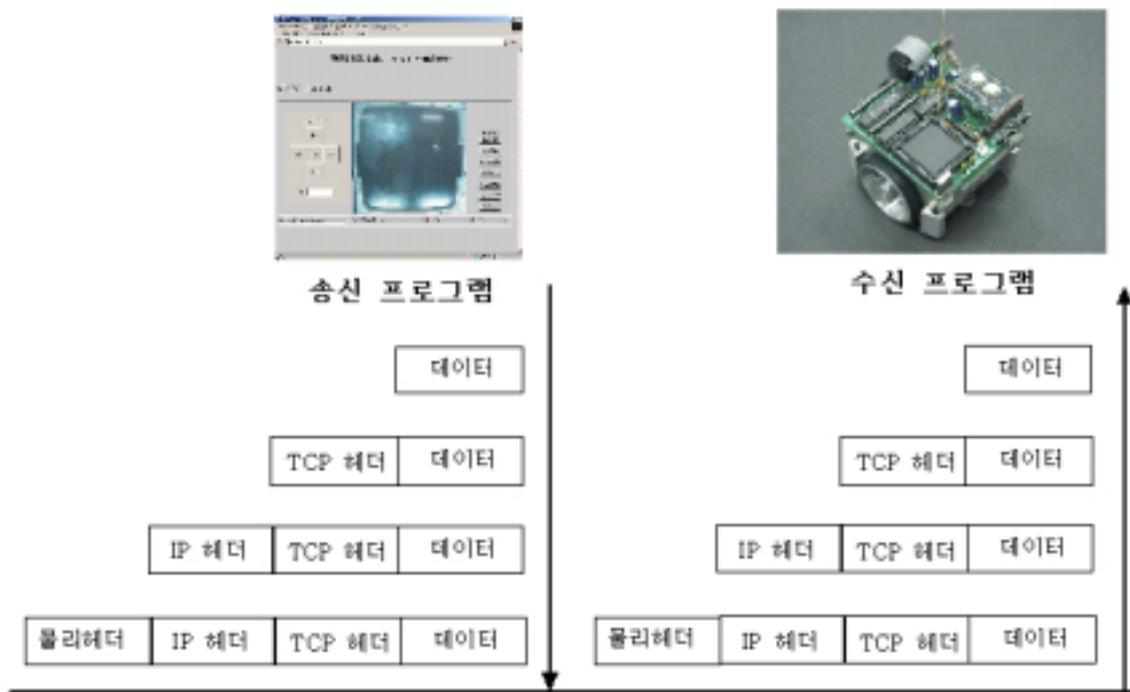


Fig. 3 TCP/IP 모델에서의 데이터 전송

이렇게 도착한 데이터는 송신측과는 반대로 송신측에서 붙인 헤더를 분리하고 분석하는 작업을 거치면서 최종적으로 데이터를 얻게 된다. 일반적인 네트워크 통신은 이와 같이 이루어진다. 여기서 실제 데이터에 결합되는 TCP 헤더와 IP 헤더를 간단히 도식화 하면 Fig. 4와 같다.

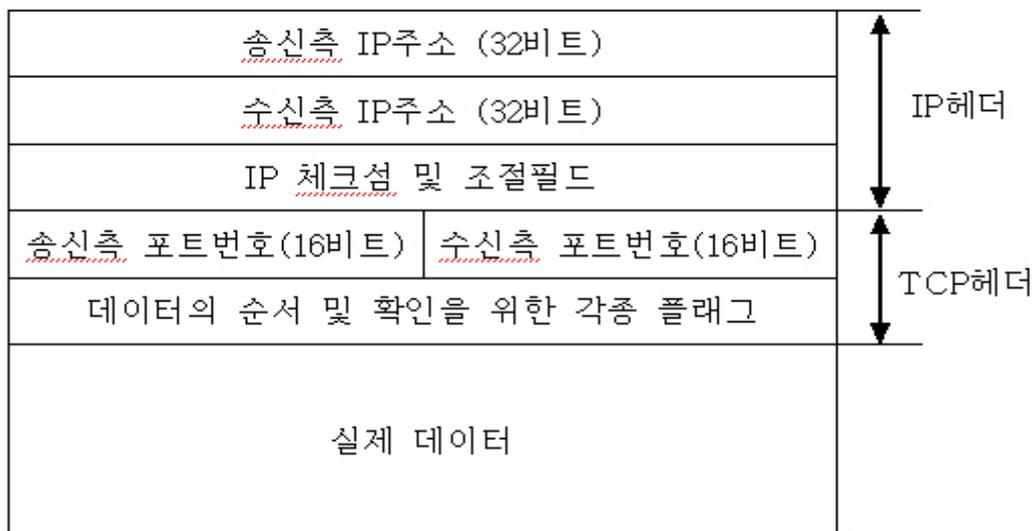


Fig. 4 TCP/IP 전송 Packet

2.6 인터넷

이젠 인터넷의 시대라 한다. 장차 모든 제어기는 인터넷으로 통제가 가능해질 것이다. 인터넷의 발전과정과 그리고 인터넷의 장점에 관해 이야기하고자 한다. 인터넷은 전세계 100여국에서 1천만대에 달하는 컴퓨터들이 하나로 연결되어 있는 세계 최대의 정보통신 네트워크이다. 간접적으로 연결되어 있는 국가를 포함하면 모두 178개국이며 전자우편으로 의사소통을 할 수 있는 나라는 154개국에 달한다. 현재 파악된 인터넷 사용 인구는 세계적으로 1억 정도로 추산된다. 소위 세계화(Globalization)의 실체는

인터넷을 통하여 실현되는 가상공간(Cyberspace)이다. 인터넷은 지리적으로 흩어져 있는 컴퓨터 세계에 대응하는 가상세계이며 하나의 거대한 지구촌(Global village)을 이룬다. 인터넷 망 이외에도 연구용 망인 BITNET, UUCP, FidoNet 등이 있는데 점차 인터넷과 연결되는 추세에 있다. 국가간의 망을 연결하는 백본 회선(Backbone Line)의 비용문제로 인하여 아직도 인터넷을 사용할 수 없는 곳이 있는데 다른 네트워크를 통하여 인터넷과 연결되기도 한다. 인터넷 이외의 네트워크와 연결되는 망을 아우터 넷(Outernet)이라 한다. 한국의 인터넷망은 공공 망과 상업 망으로 구분되며 현재 500개 이상의 기관에서 5만대 가량의 호스트 컴퓨터가 연결되어 있다. 국내에서 인터넷이 상업적으로 개방된 것은 1994년 말부터인 데 연 10% 이상의 성장속도로 이용이 확대되고 있다. 인터넷 프로토콜 전세계의 수많은 컴퓨터가 하나의 네트워크로 연결되는 것은 모두 동일한 프로토콜(Protocol)을 사용하기 때문이다. 인터넷은 다음과 같이 3단계의 과정을 통하여 발전하였다.

미국에서 군사 목적의 전산망 발전(60년대말~70년대) :

인터넷은 원래 군사적인 목적에 의해서 만들어졌다. 1969년 미 국방성이 대형 컴퓨터 이용을 촉진시키고 군사 기술 유출 위험을 분산시키기 위해, 관련 연구를 수행하고 있던 주요 대학의 컴퓨터를 연결한 알파넷(ARPANET: Advanced Research Projects Agency NET)을 구축하면서 시작되었다. 학술,연구 목적으로 용도 확대(80년대):본격적인 인터넷 시대가 열리게 된 것은 미국의 국립과학재단(NSF: National Science

Foundation)이 알파넷을 흡수하여 전국적인 대단위의 학술적 데이터베이스망을 구축하게 되면서부터 서로 다른 종류의 컴퓨터 네트워크를 상호 연결하여 각종 정보를 공유할 수 있는 네트워크 기술이 개발되어 연구자들 간에 이용이 확산되었다.

상업적 활용의 시작(90년대): 학술적 목적으로 소수의 학자들에 의해서만 사용되던 인터넷은 1990년부터 인터넷에 월드 와이드 웹(WWW: World Wide Web)의 등장으로 사용이 용이해지고 문자, 화상, 음성 데이터 등의 전송이 가능해지면서 인터넷의 대중화 및 상업적 활용이 본격화되어 새로운 국면을 맞게 되었다. 인터넷을 일반 대중도 손쉽게 사용할 수 있게 만드는 넷스케이프(Netscape) 또는 익스플로러(Explorer)와 같은 소프트웨어의 비약적인 발전과 함께 인터넷을 사용하는 일반 대중의 수가 점차 늘어나면서 수많은 기업체들이 상업적인 도구로써 인터넷을 적극 활용하고 있다. 2000년에 접어들면서 모바일 인터넷이 널리 확산되기 시작하였다. 개인 휴대 통신인 PDA나 모바일 Phone을 사용하여 인터넷을 사용하고 있고, 그 동안 유선으로 연결하였던 네트워크의 구조가 무선랜(Wireless Lan)으로 변해가는 추세이다. 근래에는 대학 캠퍼스 내에서도 무선랜을 활용한 원격강의가 행해지고 있다. 아래에는 인터넷의 발전과정을 기술하여 보았다.

<인터넷의 발전>

1836 전신(telegraph)의 발명----->텔레커뮤니케이션의 신호탄

1858~1866대서양 횡단 케이블 설치----->텔레커뮤니케이션을 위한 허브
탄생

1876그레이엄 벨이 전화 발명 ----->인터넷 연결을 위한 백본

1957스푸트니크위성 발사 ----->글로벌 커뮤니케이션시대 향한 첫걸음

1962~1968패킷 스위칭 네트워크 개발 ----->정보보호에 대한 인식

1969 UCLA,스탠퍼드 연구소----->인터넷의 탄생

1972전자메일 개발----->전자메일이 가능해짐

1973런던대-노르웨이 왕립레이더 연구소----->세계의 인터넷으로
네트워크로 연결

1974TCP(Transmission Control Program)----->인터넷 통신 초석마련

1977전자메일 실용화----->키보드로 대화하는 시대 열림

1979유즈넷(USENET)----->인터넷을 통한 토론문화 정착

1982TCP/IP(Transmission Control Protocol/Internet Protocol)개발

1984도메인명 시스템(DNS)----->인터넷 주소 기억하기 쉬워짐

1988 Internet Relay Chat(IRC) 개발

1991 미네소타대가 고퍼 개발----->인터넷 대중문화의 출발점

1992 모자이크 등장 ----->인터넷이 급격히 팽창

1993 네스케이프 개발 ----->진정한 WWW혁명의 시발점

1995 도메인명 유료화 ----->인터넷 상업화

1996 MS 익스플로러 개발

1997 웹 사이트 100만 호스트 1950만

1998 인터넷 인구 1 억 명

2000 인터넷 인구 3 억 명

2001 인터넷 인구 10억명

3. Web Based Mobile Robots Control System

3.1 시스템 개요

전체 시스템 구성은 Fig. 5와 같다. 먼저 모빌로봇과 모빌로봇의 이미지 처리 및 이동명령을 생성하기 위한 컨트롤 시스템, 웹 브라우저를 탑재한 클라이언트 PC, 그리고 웹 서버로서 구성된다. 컨트롤 시스템은 이미지를 통해 이동로봇 및 목표지점의 위치와 방향각을 계산하여 이동로봇의 이동명령을 생성한다. 또한 이미지 처리를 통해 얻어진 로봇의 위치를 웹 서버를 통해 클라이언트로 송신한다. 웹 서버는 컨트롤 시스템으로부터 이동로봇의 위치를 전송 받아 로봇의 이미지를 클라이언트에 나타낸다. 컨트롤 시스템은 이동로봇에 RF통신으로 데이터를 전송한다.

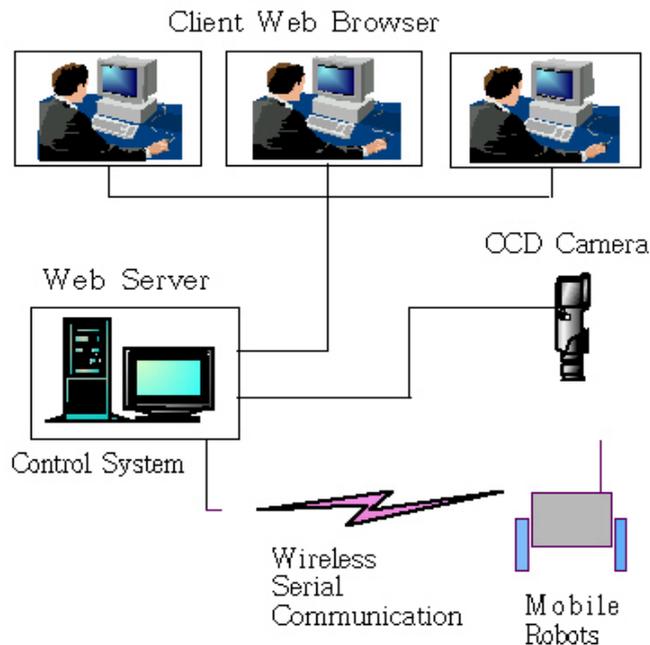


Fig. 5 시스템 구성도

3.2 Java 언어

본 논문의 모든 웹 Application은 Java언어를 사용하여 구성하였다. Java언어에 관하여 간략하게 소개하고자 한다. 자바(Java)를 이야기할 때 크게 두 가지로 나누어 이야기 할 수 있다. 먼저, 기계어, 어셈블리어(Assembly), 포트란(FORTRAN), 코볼(COBOL), 파스칼(PASCAL), 또는 C 등과 같이 프로그래밍을 하기 위해 사용하는 자바 언어가 있고, 다른 하나는 자바 언어를 이용하여 프로그래밍 하기 위해 사용할 수 있는 자바 API(Application Programming Interface)와 자바 프로그램을 실행시켜 주기 위한 자바 가상머신(Java Virtual Machine) 등을 가리키는 자바 플랫폼(Platform)이 있다. 다시 말해서, 자바 언어는 Visual C++ 와 비유될 수 있고, 자바 플랫폼은 윈도우 95/98/NT 및 윈도우 95/98/NT API와 비유될 수 있다.

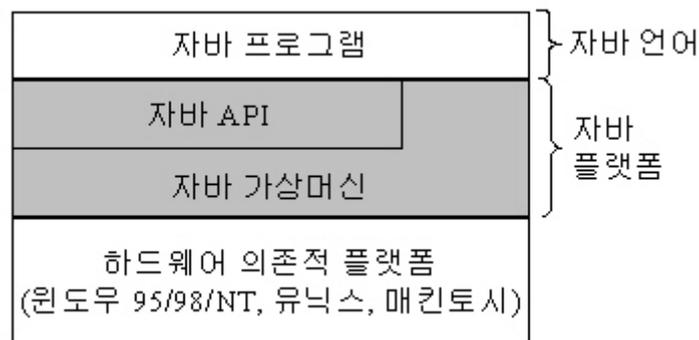


Fig. 6 Java 의 계층적 모델

자바 언어(Java Language)를 이용하여 작성한 자바 프로그램(Java

Program)은 자바 컴파일러(Java Compiler)를 이용하여 자바 바이트코드(Java Byte code)로 컴파일 되고, 이 자바 바이트코드는 자바 가상머신에 의해 해석되어 실행되는데, 이때 자바 가상 머신은 자바 바이트코드에 대한 해석기 즉 인터프리터(interpreter)로 동작하게 된다. 이렇게 자바 프로그램은 컴파일 방식 및 인터프리터 방식이 모두 적용된다.

자바 바이트코드는 자바 가상머신에서 실행되는 기계어라고 생각하면 된다. 그리고, 모든 자바 인터프리터는 자바 가상머신을 구현해 놓은 것으로, 자바 가상머신과 자바 인터프리터를 같은 것으로 생각할 수 있다. 이러한 자바 가상머신은 JDK(Java Development Kit)에 포함되어 있을 수도 있고, 자바 호환 웹 브라우저 내에 내장되어 있을 수도 있다. 또는, 자바 칩과 같이 하드웨어에 직접 구현될 수도 있다. 자바 바이트코드는 “write once, run anywhere”라는 말을 가능하게 해 준다. 다시 말해서, 자바 언어를 이용하여 작성한 자바 프로그램을 각 플랫폼(윈도우 95/98/NT, 리눅스, 유닉스, 매킨토시 등)에 맞게 제공되는 자바 컴파일러를 통해서 바이트코드로 컴파일 할 수 있다. 그리고, 이 바이트코드는 자바 가상머신이 있는 어떤 곳에서도 실행될 수 있다.

플랫폼이란 프로그램이 실행되는 하드웨어 또는 운영체제와 같은 소프트웨어적인 환경을 말한다. 대부분의 플랫폼은 하드웨어와 운영체제를 함께 일컬어 말한다. 그러나 자바 플랫폼은 하드웨어와 무관하게 동작하는 오직 소프트웨어적인 플랫폼이란 점에서 다른 플랫폼과 다르다. 이를 위해 자바 플랫폼은 두 가지의 구성요소를 갖는다.

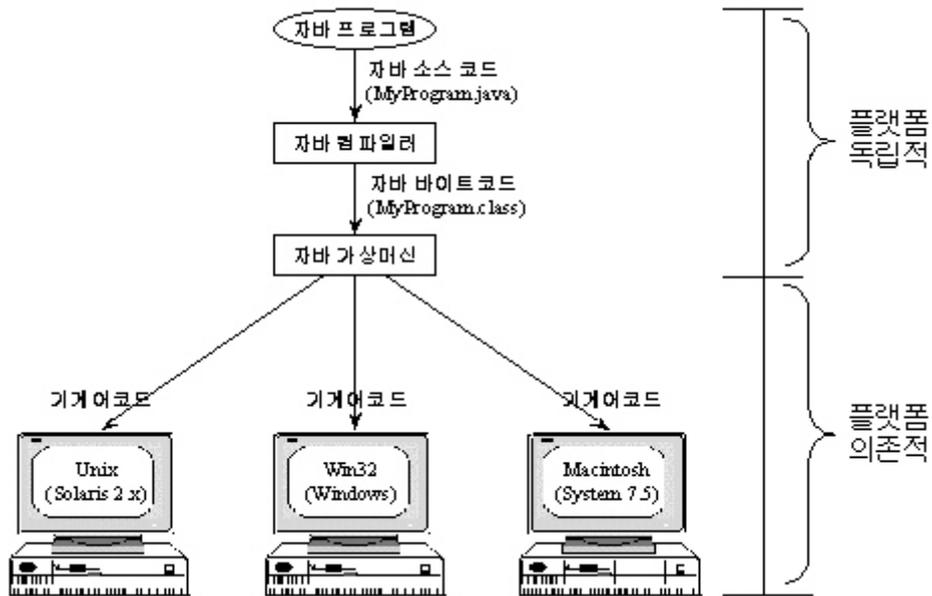


Fig. 7 플랫폼 독립적인 Java

* 자바 가상머신(Java Virtual Machine; Java VM):

자바 가상머신은 자바 플랫폼의 기반을 이루며, 다양한 하드웨어 기반 플랫폼에 포팅(porting) 된다. 다시 말해서, 자바 가상머신은 윈도우 95/98/NT, 유닉스, 또는 매킨토시 등과 같은 기존의 운영체제 또는 인터넷 익스플로러와 넷스케이프 등과 같은 웹 브라우저 등, 여러 가지 플랫폼에 설치되어 사용될 수 있으며, 사용자는 자바 바이트코드로 컴파일된 자바 프로그램을 실행시키기 위해서 이 자바 가상머신을 이용하면 된다.

* 자바 API(Java Application Programming Interface):

자바 API는 윈도우 API와 같이 운영체제에서 제공해 주는 라이브러리와 같은 것이다. 다시 말해서, 자바 프로그램을 개발하기 위해 사용할 수 있는 라이브러리 또는 클래스들이라 할 수 있다. 이러한 자바 API는 서로 관련된

클래스들을 묶어서 패키지 단위로 제공되고 있다

3.2.1 Java의 장점

*간단하다(Simple) :

자바의 주된 특징은 기존의 C/C++ 언어의 문법을 기본적으로 따르고, C/C++ 언어가 갖는 전처리기, 포인터, 포인터 연산, 다중 상속, 연산자 중첩(overloading) 등 복잡하고 이해하기 난해한 특성들을 제거함으로써 기존의 프로그램 개발자들이 쉽고 간단하게 프로그램을 개발할 수 있도록 한다. 자바는 쓰레기 수집(garbage collection) 기능을 포함시켰을 때 희생을 감수해야 했다. 다시 말해서 프로그래머에게 메모리 관리라는 짐을 덜어 주었지만, 반대로 자바 실행 시간 시스템의 복잡성은 늘어났다. 자바의 쓰레기 수집 기능은 객체 지향 프로그래밍의 부담을 줄여 주고, 고질적인 소스 코드 버그를 줄이는 데 크게 기여했다.

*객체 지향적이다(Object-Oriented):

자바는 C++와는 달리 처음부터 객체지향 개념을 기반으로 하여 설계되었고, 객체지향 언어가 제공해 주어야 하는 추상화(Abstraction), 상속(Inheritance), 그리고 다형성(Polymorphism) 등과 같은 특성들을 모두 완벽하게 제공해 주고 있다. 또한, 자바의 이러한 객체지향적 특성은 분산 환경, 클라이언트/서버 기반 시스템이 갖는 요구사항도 만족시켜 줄 수 있다.

*견고하다.(Robust) : Exception Handling - 높은 신뢰성(Reliability)을 갖는다:

애플리케이션이 튼튼할수록 신뢰도는 커진다. 자바는 컴파일 시에 에러 검사를 철저히 하고, 실행 시에 발생할 수 있는 에러에 대해서도 실행 시에 철저히 검사를 수행함으로써 신뢰도가 높은 프로그램을 작성할 수 있도록 해 준다. 또한, C/C++ 프로그램 개발자들을 가장 혼란스럽게 하고, 프로그램의 치명적인 오류를 발생시킬 수 있는 포인터 및 포인터 연산을 자바에서는 사용하지 않게 함으로써, 포인터를 사용함으로써 프로그래머가 범할 수 있는 오류를 없앴다는 것이다.

*보안성(Security)을 갖는다:

자바는 분산환경에서 작동하도록 설계 되었다. 그러나, 자바는 자바 언어와 자바 런타임 시스템 내에 보안 기능이 내재되어 있기 때문에 보안성이 있는 프로그램을 개발할 수 있도록 해 준다. 새로운 객체의 메모리 할당이 new 연산자를 통해 명시적으로 이루어 지기 때문에 안전하다. 또한 메소드 호출은 프로시저를 실행 할 수 있는 유일한 방법이다. 이 기능은 객체 지향 패러다임의 일부이다. new 연산자는 ClassLoader 라는 시스템 수준의 클래스에 의존한다. 이것은 불러 오는 클래스에 입각한 보안 규칙을 적용한다. 그 결과로 ClassLoader 클래스 형은 인스턴스화한 모든 클래스에 허용될 기능을 조정하는 역할을 한다. 또한 , 자바 애플리케이션은 시스템 힙(heap), 스택 (stacks) 또는 메모리에 액세스 할 수 없다. 이러한 특성은

자바 프로그램이 네트워크 환경에서 바이러스 등과 같은 프로그램이 파일 시스템을 파괴하려는 것을 막을 수 있도록 해 준다.

*아키텍처 중립적(Architecture-neutral)이고 이식성(Portable)이 높다:

자바는 서로 다른 이종(Heterogeneous)의 네트워크 환경에서 분산 되어 실행될 수 있도록 설계되었다. 이와 같은 환경에서는 응용 프로그램들이 다양한 하드웨어 아키텍처 위에서 실행될 수 있어야만 한다. 이를 위해 자바 컴파일러는 이종의 하드웨어 및 소프트웨어 플랫폼에서 효율적으로 코드를 전송하기 위해 설계된 아키텍처 중립적인 중간 코드인 바이트코드를 생성한다. 이는 동일한 자바 프로그램의 자바 바이트코드가 자바 가상머신이 설치되어 있는 어떤 플랫폼에서도 실행될 수 있도록 하는 것이다. 또한, 자바는 기본 언어 정의를 엄격하게 함으로써 효율적인 이식성을 제공해 주고 있다. 예를 들어, int 형과 같은 기본 데이터형의 크기를 플랫폼과 무관하게 일정하게 하고, 연산자의 기능을 확실하게 규정하고 있다. C 언어를 이용하여 int 형을 선언할 때, 도스에서는 16비트, 윈도우 95/98/NT 등 32비트 운영 체제 환경에서는 32비트, 유닉스에서는 32비트 등 그 플랫폼에 따라 크기가 다르지만, 자바에서는 플랫폼에 상관없이 32비트로 고정되도록 하였다. 이는 자바 프로그램이 실행되는 환경이 자바 가상머신으로 동일하기 때문이다.

*높은 수행성능(High-performance)을 제공한다:

자바에서는 인터프리터가 런타임 환경을 검사할 필요 없이 실행될 수 있도록 구성하였기 때문에 뛰어난 성능을 제공해 준다. 쓰레기 수집기(garbage collector) 즉 메모리 관리자는 자동으로 낮은 우선순위의 백그라운드 스레드로 실행되어 메모리가 필요할 때에만 동작하도록 함으로써, 자바 가상머신에게 무리를 주지 않으면서 보다 나은 수행 성능을 제공할 수 있도록 해 준다. 또한, 방대한 양의 계산을 수행하는 프로그램은 계산이 많은 부분을 본래의 플랫폼에 해당하는 기계어 코드로 재작성하여 자바 프로그램과 인터페이스 할 수 있도록 하였다.

*인터프리터(Interpreter) 방식이다:

자바 언어로 작성된 자바 프로그램을 중간언어 형태인 자바 바이트코드로 컴파일하고, 이렇게 생성된 자바 바이트코드를 자바 인터프리터가 해석함으로써, 자바 인터프리터와 런타임 시스템이 이식(porting)된 모든 플랫폼에서 자바 바이트코드를 직접 실행할 수 있다.

*다중 스레드(Multi-thread)를 지원한다:

자바의 다중 스레드 기능은 동시에 많은 스레드를 실행시킬 수 있는 프로그램을 만들 수 있도록 해 준다. 자바는 동기화 메소드들을 기본적으로 키워드로 제공함으로써, 자바 언어 수준에서 다중 스레드를 지원해 준다. 자바 API에는 스레드를 지원해 주기 위한 Thread 클래스가 있으며, 자바 런타임 시스템에서는 모니터와 조건 잠금 함수를 제공해 준다.

위와 같은 Java만의 장점으로 인해 인터넷상에서 Application을 개발함에 있어서 Java로서 프로그래밍을 하게 되면 보다 쉽고, 견고하며, 보안성과 수행성능이 높은 컨트롤을 할 수 있다.

3.3 Graphic User Interface (GUI)

사용자 인터페이스는 4개의 독립적인 모듈로 이루어져 있다. server-side program과 client-side applet을 각각 포함한다. 4개의 모듈은 chat module, Video feedback Module, Virtual Representation Module, Login Module로 구성되어있다.



Fig. 8 User interface

3.3.1 Login Module

동시에 여러 명의 사용자가 모빌로봇을 컨트롤 하려하면, 각각의 사용자가 각각의 제어를 하게 될 것이다. 그렇게 되면 모빌로봇에 심각한 문제가 발생하게 된다. 그것을 방지하기위해 Login Module을 두어 Login을 한 단 한명의 사용자만 이동로봇을 컨트롤 할 수 있게 했다. Login을 통한 접속 방법을 사용하면, 로봇 사용자가 접속한 시간과 로봇을 작동시킨 시간 등의 이력을 데이터화 하여 남길 수 있다는 장점이 있다.



Fig. 9 Login module

3.3.2 Virtual Representation Module

로봇을 컨트롤 할 때, 사용자는 실제 로봇의 환경을 잘 모르기 때문에 로봇이 움직일 수 있는 범위를 벗어난 제어를 하게 될 수도 있다. 그러한 것을 미연에 방지하기 위해 사용자가 원하는 위치로 로봇이 정확히

움직이는가 하는 것을 로봇에 전달 될 이동명령(데이터) 만으로 확인해 볼 수 있게 하는 기능이다. 로봇을 움직이는 명령을 data적인 수치를 사용하여, GUI상에 Representation 해서 보여진다. 이것은 VRML을 사용하였다.

VRML 은 WWW(World Wide Web) 의 3 차원 표준 데이터 포맷으로 텍스트 형식의 데이터를 브라우저를 통해 3 차원 장면으로 사용자들에게 표현할 수 있으며, 개방된 플랫폼과 독립적인 명세서(specification)를 제공한다.

HTML(HyperText Markup Language)로 대표되는 웹(Web)기술은 인터넷의 대중화에 크게 기여를 했지만 현실공간을 사실감 있게 표현할 수 없다는 기술적인 단점을 가지고 있었다. 이에 반해 현실생활을 생생히 표현할 수 있는 3D 기술 언어인 VRML(Virtual Reality Markup Language)은 3 차원적 공간개념이 추가된 스크립트 언어로써 가상세계를 기술해 준다. 이는 가상현실의 한 부분을 나타낼 수 있으며, 이러한 가상공간에서 자신의 분신을 아바타(Avatar)라 한다. 본 실험에서는 Virtual Representation Module에서 VRML로서 모빌로봇으로 표현되는 3D이미지를 아바타라고 할 수 있다.

VRML2.0 버전에서는 HTML 문서와의 결합뿐만 아니라 기존의 실시간 멀티미디어 전송 기술 및 자바와 같은 새로운 스크립트를 수용하며, 실시간 스트리밍 포맷인 리얼오디오(RealAudio), VDOLive, 쇼크웨이브(Shockwave)등도 수용될 수 있다. 또한 자바(JAVA)와의 결합으로 인해 이벤트(Event) 처리를 비롯하여 3 차원 개념을 통한 환경안에서

다양한 경험치들을 클라이언트들에게 제공할 수 있다.

이러한 자바 언어와 VRML 간의 인터페이스(Interface)를 EAI(External Authoring Interface)라 하며, EAI 는 자바와 VRML 을 연결 시켜 VRML 의 특정한 노드(Node)에 접근하여 제어하거나 장면 그래프(Scene Graph)의 계층 구조를 변경시킬 수가 있다.

클라이언트 GUI를 통해 모빌로봇이 움직이도록 명령을 내리면 그 명령이 바르게 로봇에게 전달 될 것인지 아닌지를 이 모듈로부터 알 수 있다. GUI를 통해 좌표값을 명령하면 명령대로 움직일 것인지의 여부를 VRML을 사용하여 미리 움직여지는 아바타를 통해 확인해 볼 수 있다. 이 기능은 GUI에서 ON/OFF버튼을 두어 충돌할 위험이 있는 경우나 또는 미로통과등의 보다 복잡한 구역을 이동로봇이 이동하게 될 때 한정하여 사용할 수 있는 기능이다.

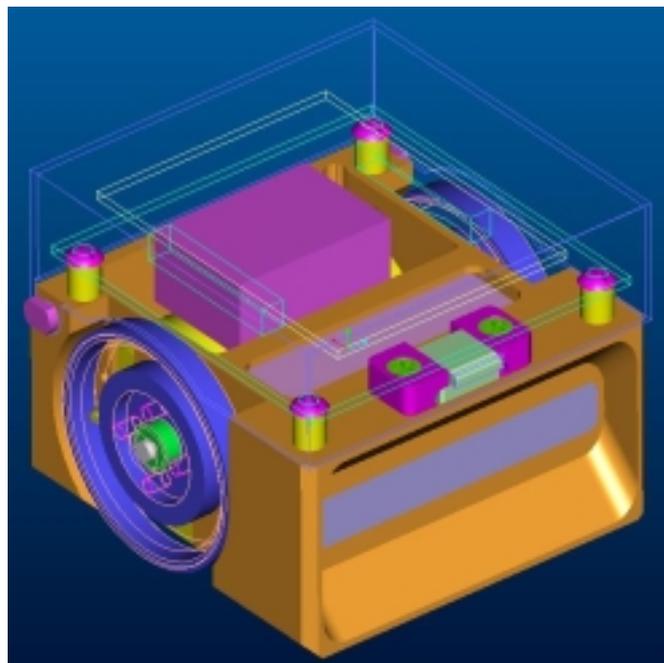


Fig. 10 VRML 을 이용한 모빌로봇 모델링

3.3.3 Video feedback Module

실제 이동로봇이 움직인 화면을 클라이언트의 GUI로 전송해주는 기능을 한다. feedback된 화면을 Java server program으로부터 소켓전송을 통해 Java applet으로 전송한다. 이때의 이미지는 GIF나 JPEG 포맷이다. 해상도는 200x150의 크기로 초당 10~15프레임이 전송된다. 본 논문에서의 전송방식은 유니캐스트 방식을 사용했다.

유니캐스트는 양방향과 점대점 연결을 구성하는 컴퓨터의 망을 말한다. 오늘날 대부분의 망은 이러한 방식으로 동작한다. 사용자가 파일을 요구하고, 서버는 특정 클라이언트에게 파일을 전송한다. 네트워크를 통하여 멀티미디어를 스트리밍 할 때 유니캐스트의 장점은 클라이언트 컴퓨터는 멀티미디어 스트림을 제공하는 컴퓨터와 통신을 할 수 있다는 점이다. 예를 들면 서버는 유니캐스트 비디오 스트림을 클라이언트에게 제공하고 클라이언트는 마이크로소프트 미디어 플레이어에 있는 VCR 제어장치를 이용하여 서버에게 스트림을 중지하거나 스트림에 있는 표시기로 전진 또는 후진을 하도록 요청할 수 있다. 유니캐스트의 단점은 서버에 연결된 각각의 클라이언트가 별도의 스트림을 받음으로써 네트워크 대역폭을 많이 사용한다는 것이다.

3.3.4 Chat module

이 모듈은 사용자가 문자의 전송을 통해 서버측과 교신할 수 있는

채팅창이며, 이동로봇의 움직임을 서버측 관리자에게 전달 할 수 있는
 기능이다. 이곳에는 창을 두개를 두어 한 개의 창에서는 채팅을 이용할 수
 있게 했고 또한 윈도우 녹음기와의 연동으로 클라이언트측에서 서버측으로
 음성을 녹음해서 전달하는 기능까지 포함시켰다. 서버측의 관리자와
 클라이언트측의 사용자가 보다 완벽한 커뮤니케이션을 이룰 수 있다.

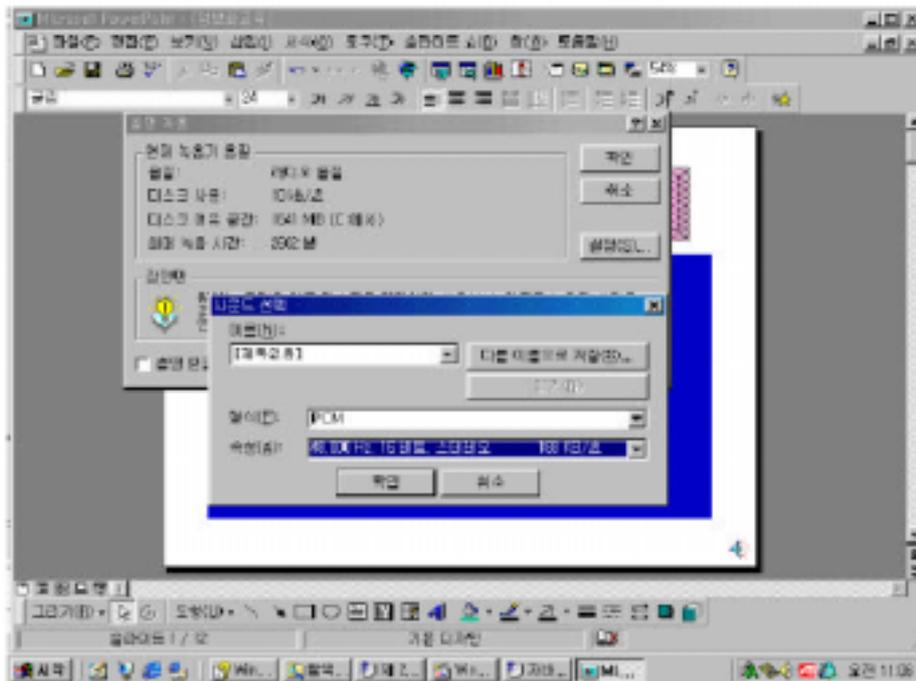


Fig. 11 Chat module에 음성녹음

3.4 Networking

시스템의 통신은 소켓을 생성하여 네트워크를 형성함으로써 이루어진다. 각각의 시스템은 소켓 접속을 통해 네트워킹되고 클라이언트의 사용자 인터페이스는 웹서버로부터 애플릿을 전송 받아 로봇을 제어한다. 시스템간의 통신은 두 가지로 나누어지는데 클라이언트에서 웹서버를 통한 컨트롤 시스템과의 통신과 컨트롤 시스템으로부터 클라이언트와의 통신이다. 클라이언트 사용자 인터페이스는 사용자로부터 로봇의 목표위치를 입력받아서 컨트롤시스템으로 전송하고, 로봇의 이미지와 로봇의 환경을 프레임을 통해 출력한다. 컨트롤시스템은 이미지 프로세싱을 통해 얻어진 로봇의 현재위치를 클라이언트로 전송하고 현재 목표와 이동목표위치로부터 이동명령을 생성하여 로봇에게 무선으로 전송한다.

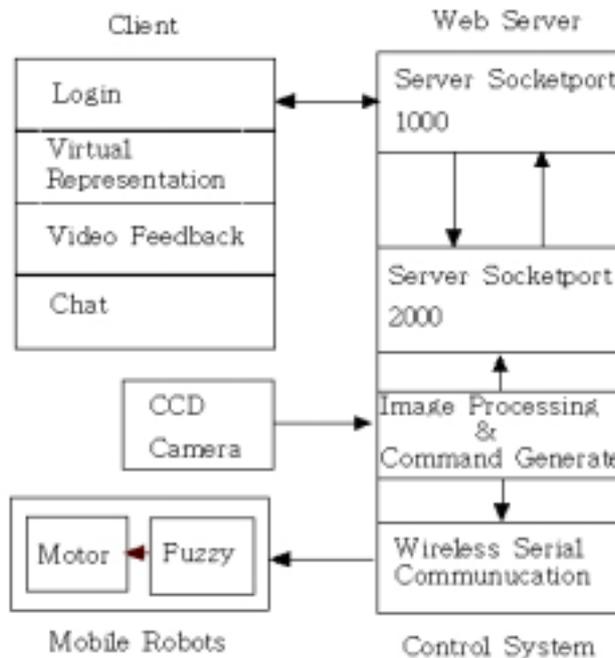


Fig. 12 Networking architecture

3.4.1 Socket connect

소켓 인터페이스는 TCP/IP 프로토콜 기반의 네트워크 프로그래밍을 위해서 미국 버클리 대학에서 개발된 API이다. 먼저 서버 프로그램에서는 클라이언트로부터의 접속을 대기하기 위한 소켓을 생성하고 대기 상태로 들어가게 된다. 서버 프로그램에서 이와 같이 대기하고 있던 소켓에 클라이언트의 접속 요청이 감지되면, 새로운 소켓을 하나 생성하여 접속한 클라이언트와의 통신을 책임지게 된다. 그리고 대기하고 있던 소켓은 또 다른 클라이언트의 접속을 기다리게 되는 것이다. 그러다가 또 다른 요청이 감지되면, 역시 새로운 소켓을 하나 더 생성하여 새로이 접속 요청을 한 클라이언트와의 통신을 수행하게 하는 것이다.

본 논문에서는 클라이언트 측에서 LOGIN을 통해 웹 서버에 접속을 하게 되면 웹 서버에서는 클라이언트의 접속 요청이 감지되면 소켓 1000번을 생성하여 연결하게 된다. 접속된 서버소켓 Port1000번은 다시 Image processing과 Command Generate와의 접속을 위해서 소켓Port 2000번을 생성하여 연결하게 된다. 컨트롤 시스템은 이렇게 접속된 웹 서버와의 통신을 하게 되고 클라이언트 측에서는 서버와 연결된 컨트롤 시스템과 데이터를 교환 하게 된다.

3.4.2 Java Applet

본 논문의 User Interface는 Java Applet을 통해서 구현하였다.

<APPLET>~</APPLET> 태그를 이용하여 HTML 페이지 내에 포함되어, 자바 호환 웹 브라우저에 의해서 실행되도록 작성된 자바 프로그램이다. 다시 말해서, 홈페이지 내에 삽입되어 자바 호환 웹 브라우저에 의해 실행되도록 규약에 맞추어 작성된 자바 프로그램을 말하는 것이다. Fig. 13은 자바 애플릿의 실행 과정을 나타낸 것이다.

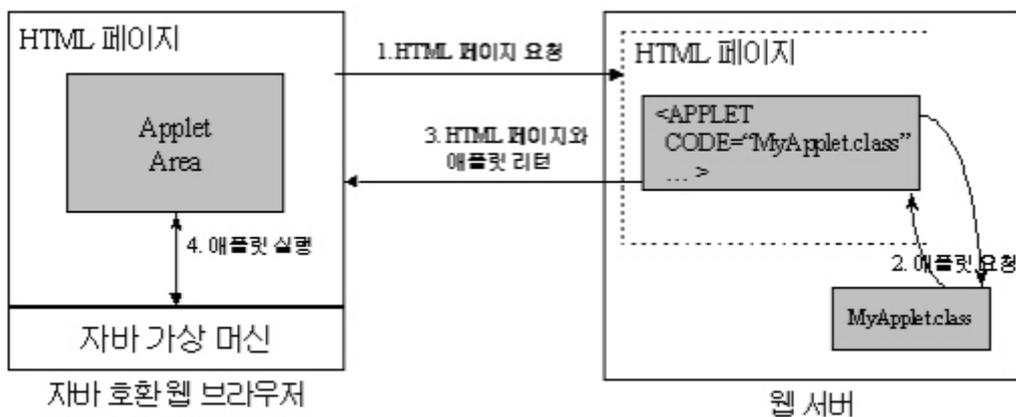


Fig. 13 Java Applet의 실행도

3.4.1 Internet Transmission Latency

인터넷으로 시스템을 제어하려 할 때, 가장 큰 문제점은 인터넷 상에서의 시간지연 문제이다. 정확한 데이터 값을 실시간 적으로 교환하여 작동되는 하드웨어와의 연결에서 인터넷의 시간지연은 크나큰 오류를 일으켜 모빌로봇을 위험에 처하게 할 수 있다.

이를 해결 할 수 있는 첫번째 방법으로는 모빌로봇에 Autonomous Control 을 주어 모빌로봇이 인터넷의 회선이 끊어졌을 때에도 자율적으로 위험을 피해가는 것이다. 하지만 이 또한 일시적인 인터넷의 중단 시 자율적인

행동을 할 수 있다는 것을 의미한다. 즉, 인터넷의 회선이 두절되면 모빌로봇을 명령을 받아 들일 수 없게 되고 그것은 바로 컨트롤의 불능을 의미한다. 다시 말해 정확한 시간지연에 관한 해결책이라고는 하기 어렵다. 두번째 방법으로는 인터넷회선의 속도를 높이는 것이다. TCP/IP 접속의 장점은 가장 대중적이며 패킷의 교환에 있어서 하나를 주고 확인하고 다시 하나를 받는 과 같은 방법으로 데이터가 전송되기 때문에 안정적이라는 것이다. 하지만 그에 반해 단점으로는 데이터 전송속도가 늦어진다는 단점이 있다. UDP를 사용하게 되면 보다 빠른 데이터 전송을 기대할 수 있다.

UDP 서비스에서 데이터 전송은 데이터그램이라는 상호 연관성 없는 데이터 단위들의 독립적 전송을 통해 이루어진다.

3.5 Mobile Robots Control

3.5.1 Robot Soccer Mobile Robot

로봇 축구 시스템을 이루는 요소 중에 하나가 로봇 시스템이다. Fig. 14는 로봇 시스템의 주요 구성부를 나타낸 것으로, 로봇 시스템은 크게 마이크로컨트롤러, 모터 구동부, 통신부, 전원부로 이루어져있다.

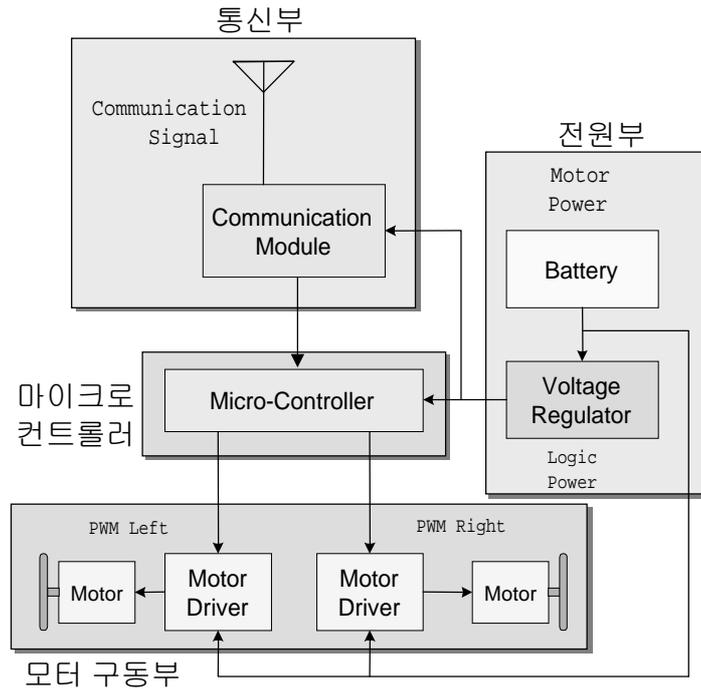


Fig. 14 Robot System의 주요 구성부

완성된 로봇은 Fig. 15와 같다. 로봇은 크게 회로기판, 충전지, 모터, 몸체(바퀴포함)로 구성된다. 로봇의 회로기판은 2개로 분리되어 있으며 밑에 기판에는 주로 파워 소자들 (전원 레귤레이터, 모터구동회로 등)과 RF모듈이 있고 위에 기판에는 마이크로 컨트롤러가 있다

마이크로 컨트롤러는 Intel 80C196을 사용한다. 충전지는 8.4V 이고 로봇의 모터 전원과 레귤레이터(LM2575)를 지난 5V는 로직 회로의 전원으로 쓰인다. 모터는 8 : 1의 기어 비를 갖는 내접기어를 사용하고 있다. 엔코더는 모터에 포함되어 있는 형태이고 한바퀴에 512 Pulse 의 해상도를 갖는데 이를 마이크로 컨트롤러에서 Pulse를 받게 된다.

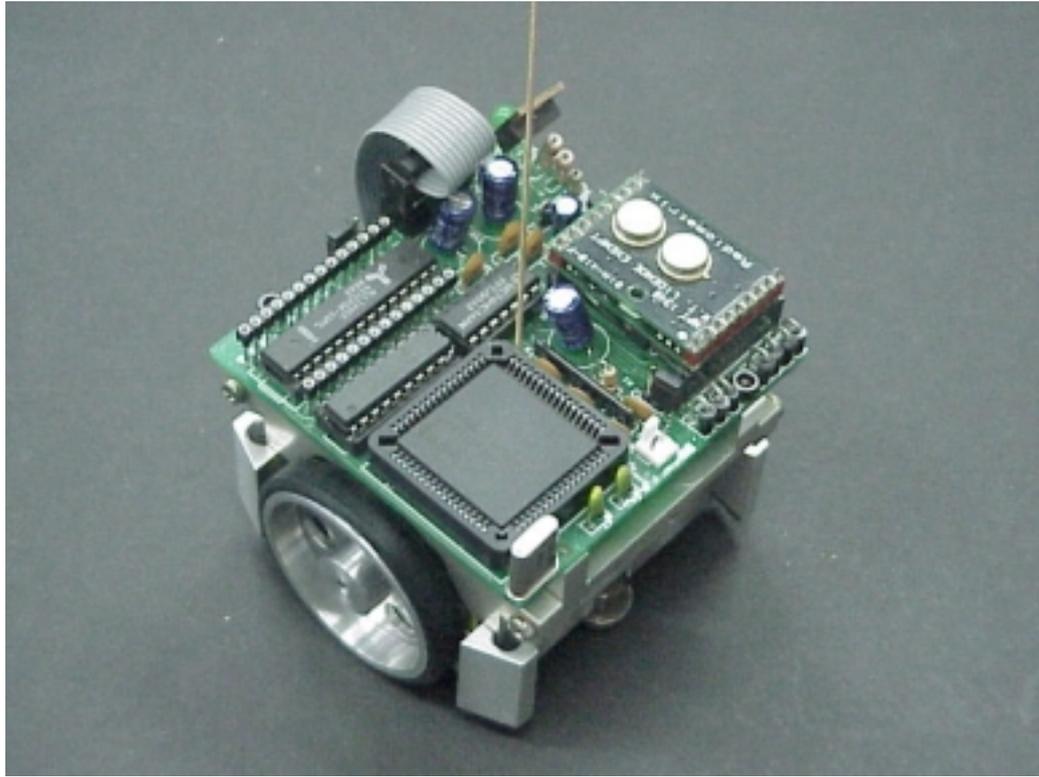


Fig. 15 완성된 모빌로봇의 외형

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

전체 시스템의 사양은 Fig. 16에서 기술되어 있는 바와 같다.

구성요소	제품명	사양
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
---------	--------------	------------

Fig. 16 The list of hardware

3.5.2 Micro Controller

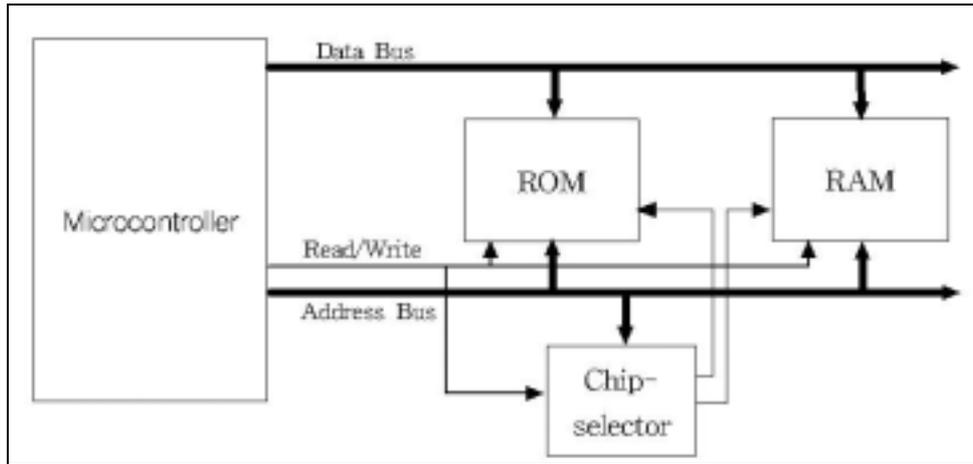


Fig. 17 The structure of microcontroller

보드를 설계하기 앞서 먼저 고려해야 할 사항은 바로 마이크로 컨트롤러의 선정이다. 어떤 마이크로컨트롤러를 선택하는가에 따라 성능, 기능, 크기, 용도, 구조 등이 크게 달라지기 때문이다. 마이크로컨트롤러의 기본 구조는 Fig 17에서 나타내었다. 원하는 시스템 사양의 속도, 처리 능력, 기능성, 호환성, 편의성 등이 선정 기준을 결정하게 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 핀을 이용하여 모터를 제어하며 직렬통신을 할 수 있는 80C196KC를 선택하였

다.

80C196KC의 경우 외부 메모리와 래치, 칩선택 전용 칩을 모두 사용했기 때문에 공간 활용면에서 매우 비효율적이라 할 수 있다. 이를 해결할 방안으로 현재는 PIC계열로 대체하는 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.

3.5.3 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 18은 펄스 변조에 따른 효과를 보여주고 있다.

Fig 19는 H-브릿지 회로로 하나의 모터를 정역전 제어하기 위해 4개의 트랜지스터가 연결되어 있으며 정역전시키는 원리는 Fig 20, Fig 21에서 보여주고 있다.

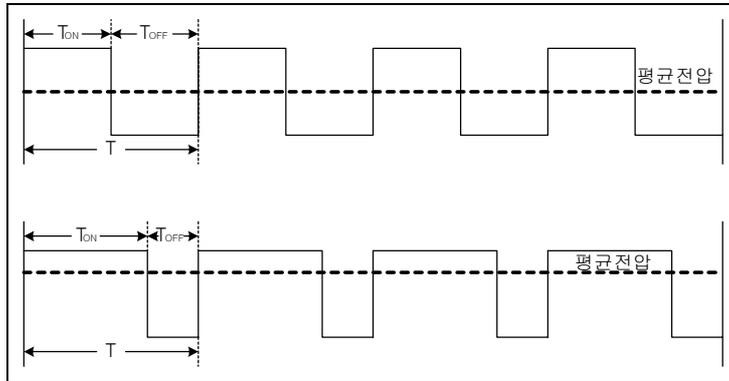


Fig 18. PWM method

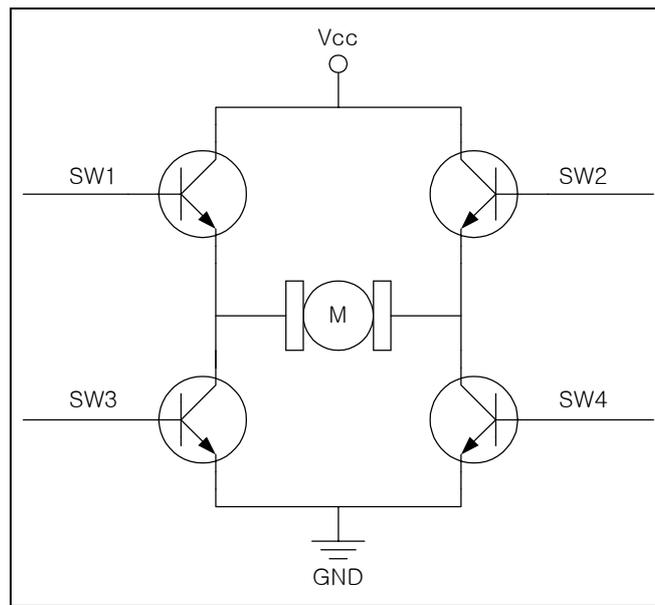


Fig 19. H-bridge circuit

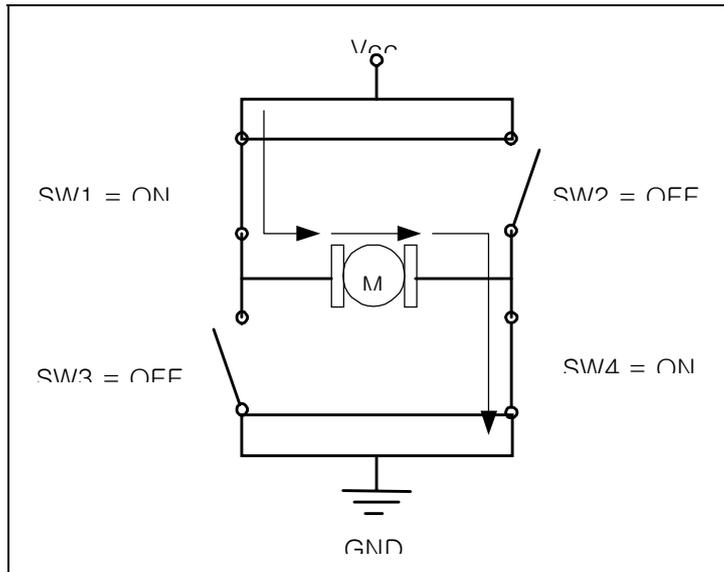


Fig 20. The forward direction

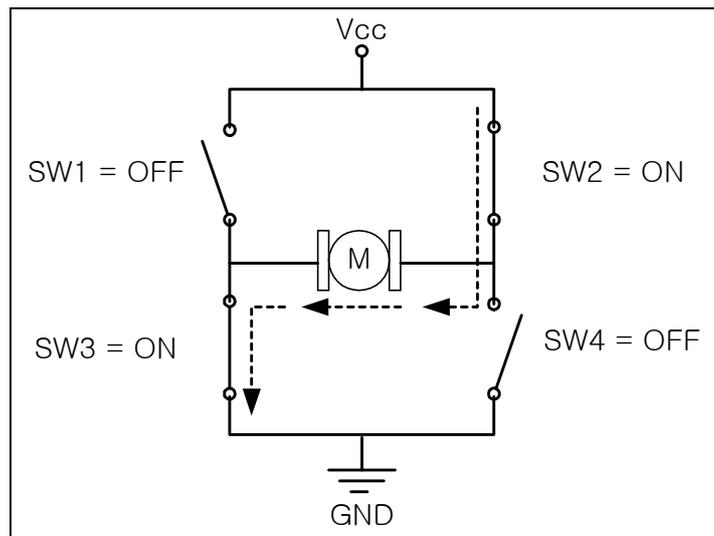


Fig 21. The backward direction

3.5.4 통신부

통신부는 상용 모듈을 이용해서 통신을 하였다.

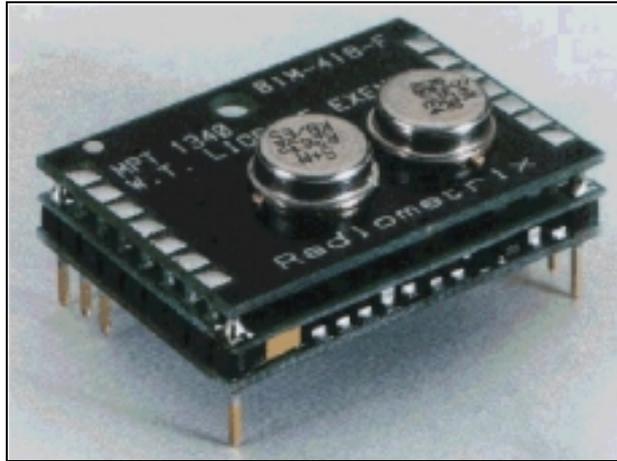


Fig 22. RF module

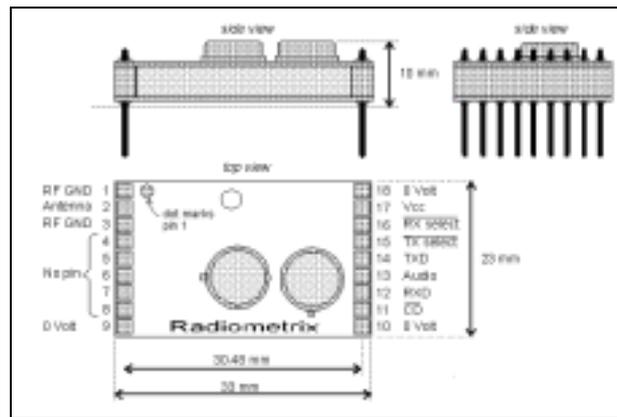


Fig 23. 모듈의 핀 순서

모듈은 RF solutions의 RF-418, 433로서 많이 알려진 제품으로 40Kbps 까지 전송이 가능하다. 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.

3.5.5 Vision System

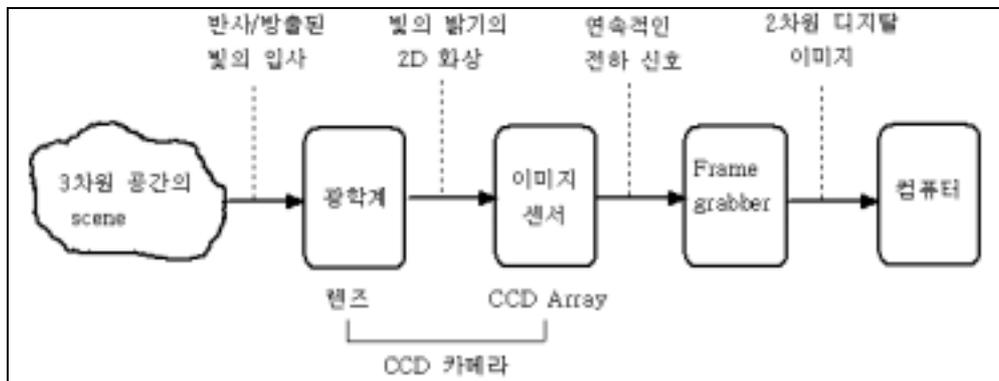


Fig 24. Vision System

Fig 24는 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.

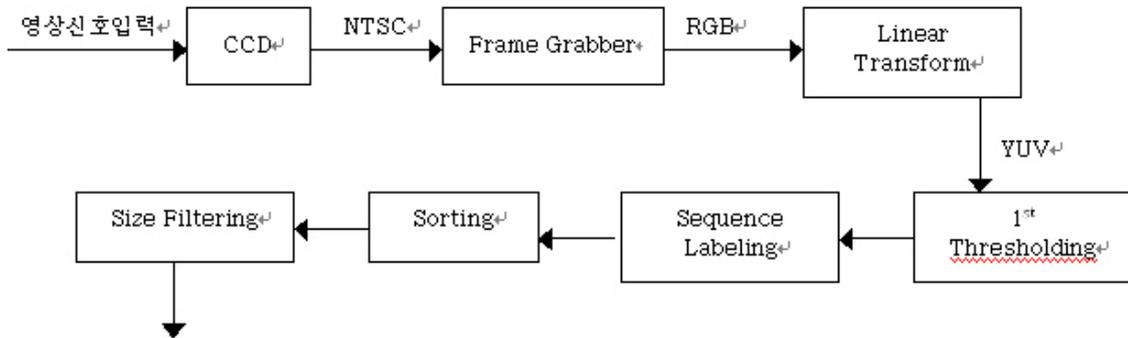


Fig 25. Vision algorithm

이것을 다시 알고리즘으로 표현한다면 Fig 25와 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 로봇을 찾아서 위치 및 각도 정보를 알아 낸다.

4. 실험 및 결과

본 논문에서는 두 가지 실험을 하였다. 첫번째는 로봇 한대가 목표위치까지 장애물을 피하여 보다 빠른 속도로 이동하는 방법으로서 로봇에 부착된 로봇 컨트롤러는 PID제어가 아닌 FUZZY제어 알고리즘을 사용하여 보다 빠르게 사용자가 원하는 로봇의 위치와 방향으로 로봇이 이동해 가는 방법을 생각해 보았다. 두 번째는 로봇 두 대를 사용하여 목표위치까지 볼을 이동시키는 실험을 하였다.

4. 1 Fuzzy Logic Based Controller를 사용한 모빌로봇의 이동

컨트롤 시스템은 로봇의 이미지 처리부와 소켓통신 그리고 이동로봇에게 명령을 송신하기 위한 RF 통신부로 구성된다. 로봇의 이동명령 방식은 사용자가 클라이언트에서 이동하고자 하는 좌표값을 입력하면 입력된 좌표가 컨트롤시스템으로 전송되고 컨트롤시스템은 전송받은 좌표를 바탕으로 로봇의 이동명령을 생성한다. 클라이언트의 인터페이스 구현을 위해서 현재 인터넷 언어로 널리 사용되는 JAVA를 사용했고, 컨트롤시스템은 C++ 언어로 구현했다.

모빌로봇의 이미지는 Image Grabber로서 캡처 된다. 이 이미지 데이터는 동축케이블을 통해 컨트롤시스템으로 연결된다. 이렇게 해서 모빌로봇의 위치를 알게 되고 분석 할 수 있게 된다. 타겟 포지션 명령을 사용자가

지정하게 되면 모빌로봇의 현재 위치 좌표에 기반 하여 새로운 좌표를 생성하게 된다. 이렇게 새로 생성된 타겟 포지션으로의 이동명령은 RF Transmitter를 통해 전달된다. 이 명령을 받은 로봇은 움직임을 시작하게 된다.

컨트롤시스템으로부터 2개의 Input명령이 모빌로봇으로 전달되게 된다. 하나는 현재 포지션과 타겟 포지션(X_c) 사이의 거리이고, 다른 하나는 수평축으로부터 시계방향으로의 Head Angle(θ_c) 값 차이가 된다.

Fig. 26은 모빌로봇의 Kinematics Model을 나타낸다.

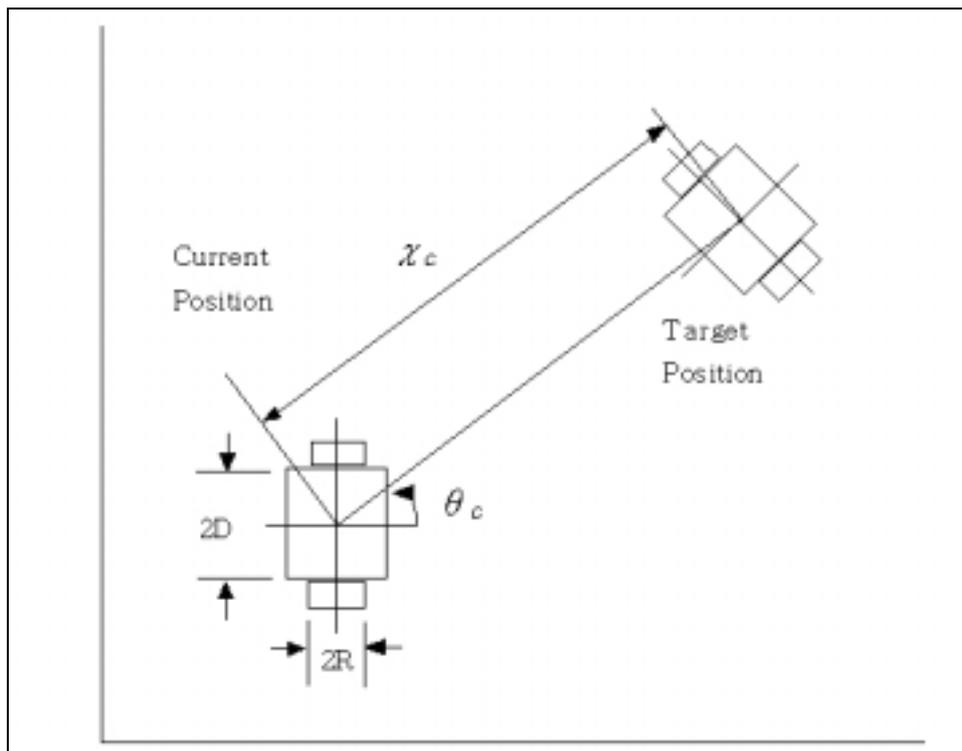


Fig. 26 Kinematics model of the mobile robots

모빌로봇의 기구학과 동역학은 아래의 두 가지 식으로부터 얻을 수 있다. 먼저 기구학 모델은 다음과 같다.

$$\begin{bmatrix} x_c \\ \theta_c \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ -R/D & R/D \end{bmatrix} \begin{bmatrix} \theta_L \\ \theta_R \end{bmatrix}$$

여기서, θ_L 과 θ_R 은 모빌로봇의 양쪽 바퀴의 각각의 각도 변위이다.

동역학 모델은 다음과 같다.

$$\theta_c(s) = \frac{R}{D} \frac{K_w}{s(\tau s + 1)} U_{diff}(s)$$

$$U_{diff}(s) = U_R(s) - U_L(s)$$

$$x_c(s) = \frac{R}{2} \frac{K_w}{s(\tau s + 1)} U_{comm}(s)$$

$$U_{comm}(s) = U_R(s) + U_L(s)$$

여기서 K_w 는 Static Gain값이고, τ 는 모터 시스템의 일정한 시간 U_L 과 U_R 은 왼쪽바퀴와 오른쪽 바퀴의 각각의 모터가 요구하는 Input Voltage이다. 로봇의 현재위치 X_c 와 Y_c 는 CCD 카메라로부터 얻은 컬러화상을 이미지 프로세싱을 통해 구한다.



Fig. 27 Color vision camera

로봇은 로봇의 위치와 자세를 계산하기 위해 두가지의 폐치색을 가진다. 로봇의 위치는 주 폐치색을 찾아서 계산하게 되고, 부폐치는 로봇의 자세를 계산하기 위해 사용된다. 폐치의 중심은 면적계산법을 통해 중심위치를 구할 수 있다. 로봇의 위치와 방위각을 계산하기 위해서 전체 이미지를 통한 면적중심법으로 이동로봇의 위치를 찾아내고, 특정영역 검색을 통해 이동로봇의 위치와 방향각을 계산한다. 계산된 이동로봇의 위치와 방향각을 로봇에게 전달하면 로봇내부의 Fuzzy Logic Based Control System으로 Gain값을 조절하여 보다 빠르고 정확하게 목표지점으로 로봇이 이동한다.

이동 명령 신호(X_c)와 (θ_c)는 모빌로봇 내부에 있는 RF Receiver로부터 받게 된다. 명령신호와 Actual Position 사이의 에러 (e_x , e_θ)는 점점더 각각의 신호를 바꾸게 된다. 이 에러는 모빌로봇 내부에 있는 Fuzzy Controller에게 전달된다. 퍼지 컨트롤러의 Out Put의 Common-mode Voltage U_{comm} 은 병진 운동하는 모션을 통제한다. 그리고 Differential-mode Voltage U_{diff} 는 회전운동을 통제한다. 이러한 두 가지의 신호는 각각 모터의 Voltage (U_L U_R) 로 변환된다.

Fuzzy Logic Controlled 모빌로봇의 Block Diagram을 나타낸 것이 아래의 Fig. 28이다.

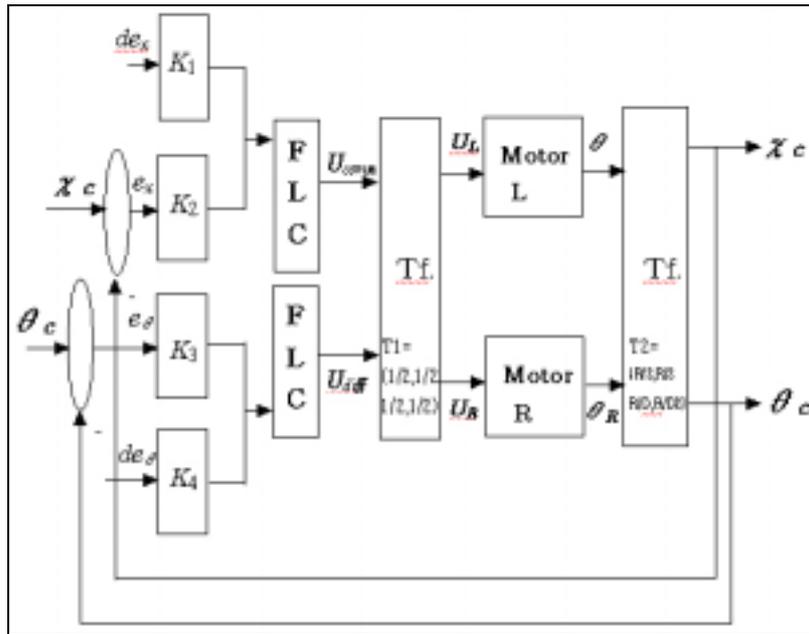


Fig. 28 Block diagram of the fuzzy logic controlled mobile robots position control system



Fig. 29 Robot soccer ground

컨트롤러는 전형적인 퍼지 컨트롤러를 사용했다. 기본적으로 퍼지

컨트롤러는 4가지의 Input (dex , ex , $de\theta$, $e\theta$)을 갖게 되고 2가지의 Output ($Ucomm$, $Udiff$)을 갖게 된다. 이렇게 각각 들어온 4가지의 Input은 게인값 K1에서 K4로 값을 변환 할 수 있다. 이곳을 통과한 값은 FLC(Fuzzy Logic Controller)를 통과하게 되고 Output으로 2개의 $Ucomm$, $Udiff$ 를 발생한다.

실험은 로봇축구 경기장 위에서 목표위치를 명령했을 때 전형적인 PID 컨트롤러만을 사용했을 때와 퍼지 컨트롤러를 사용했을 때 헤딩앵글의 변화 속도와 목표위치까지 어떠한 로봇이 먼저 도달하는가에 관한 실험을 하였다.

실험은 가로 150Cm 세로 120Cm의 로봇 축구 경기장을 주변환경으로 설정하였다. 웹서버의 운영체제는 Windows 2000 server이다. 이미지 처리를 위해 CCD카메라, 640x480픽셀의 컬러프레임 그라버를 사용하였다. 로봇의 이동명령 생성은 로봇의 목표위치가 주어지면 로봇의 위치, 방향각, 목표지점과의 방향각 오차를 생성한다.



Fig. 30 Robot control system

실험결과 모빌로봇에게 이동명령을 내렸을 때 보다 Active한 상황에서 Gain값 변화가 가능한 퍼지 컨트롤러를 가진 로봇이 Gain값의 조절로서 Gain값의 변화가 고정되어있는 PID 컨트롤러를 가진 로봇에 비해 헤딩앵글은 빠른 속도로 변화하게 되고 지시한 목표점까지 매우 민첩하게 이동함을 알 수 있었다.

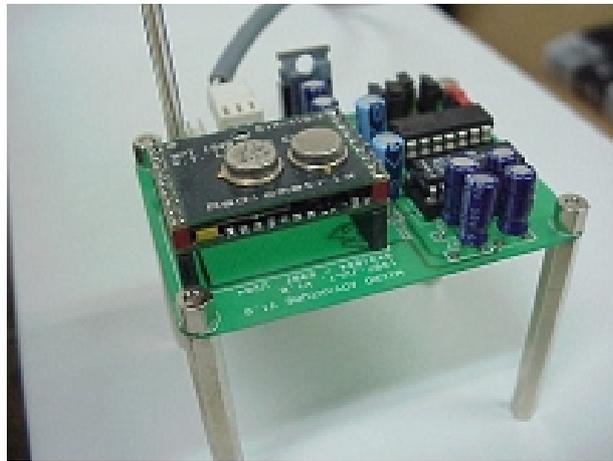


Fig. 31 RF transmitter

실험결과를 나타낸 것이 Fig.32-35 이다.

아래의 그래프는 퍼지 컨트롤러를 사용했을 때의 시간에 대한 헤딩앵글의 변화를 나타낸 것이다.

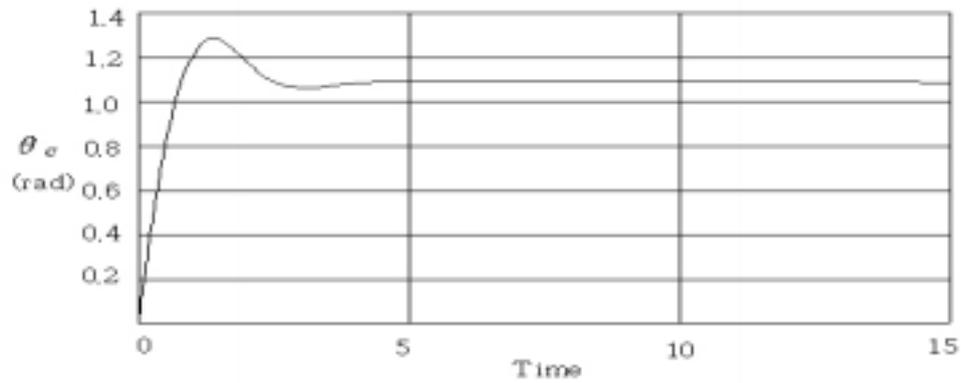


Fig. 32 Response on using the fuzzy logic controller : Heading angle

아래의 그래프는 퍼지 컨트롤러를 사용했을 때의 시간에 대한 이동거리의 변화를 나타낸 것이다.

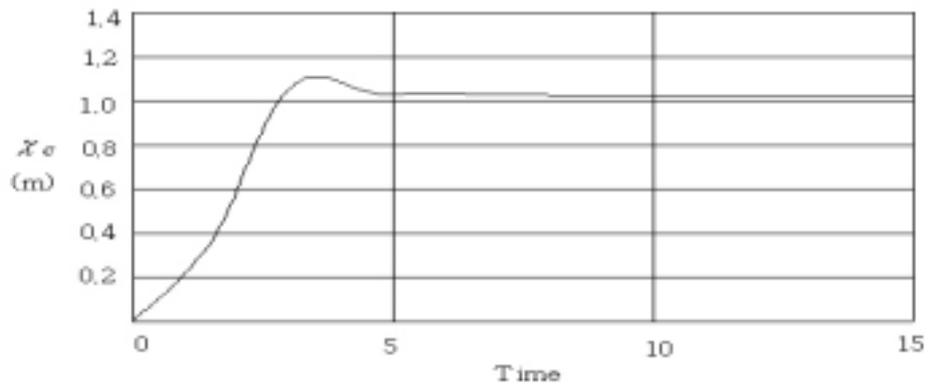


Fig. 33 Response on using the fuzzy logic controller : Distance

아래의 그래프는 PID 컨트롤러를 사용했을 때의 시간에 대한 헤딩앵글의 변화를 나타낸 것이다.

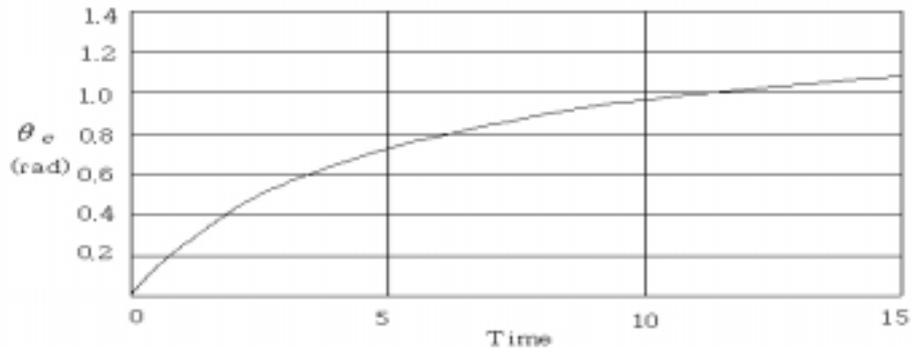


Fig. 34 Response on using the PID logic controller : Heading angle

아래의 그래프는 PID 컨트롤러를 사용했을 때의 시간에 대한 이동거리의 변화를 나타낸 것이다.

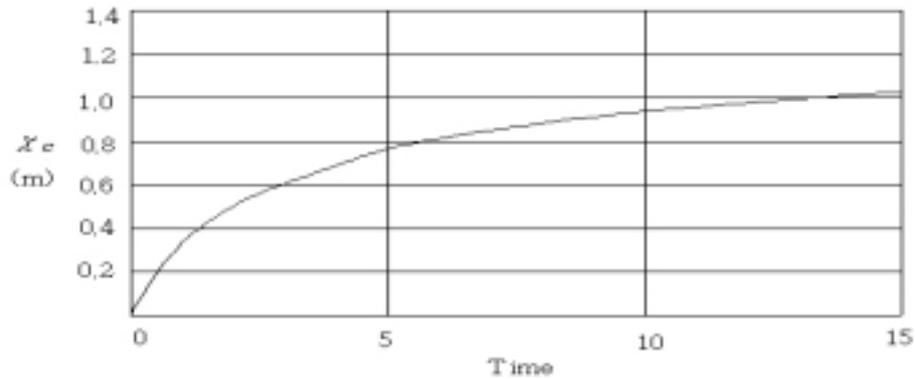


Fig. 35 Response on using the PID logic controller : Distance

그래프에서 알 수 있는 것처럼 퍼지 컨트롤러를 사용하여 Gain값을 변화시켜 컨트롤 했을 때가 Gain값이 고정된 PID 컨트롤러에 비해 모빌로봇이 목표지점으로 더 빠른 시간 안에 이동하는 것을 알 수 있다. Fig. 36은 실제로 로봇이 이동하는 모습을 보여준다.

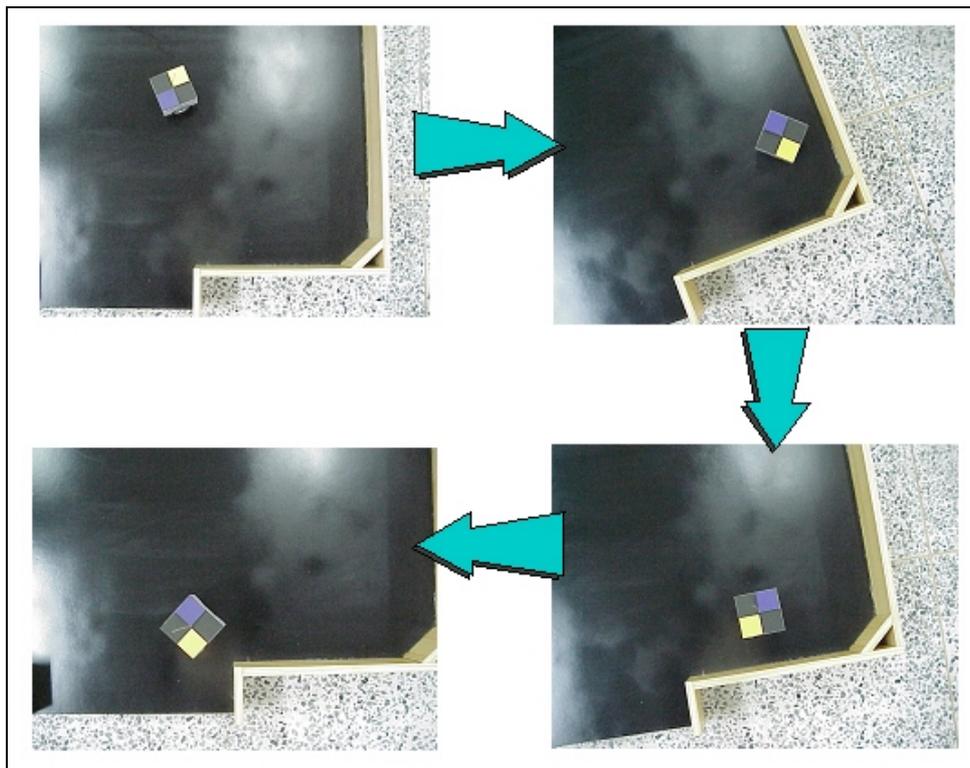


Fig. 36 Movement the mobile robot

4.2 두 대의 로봇을 이용한 목표물 이동

이번엔 로봇 두 대를 사용하여 실험해 보았다. 한대 만을 사용했을 때에 비해 인터넷으로 컨트롤 할 때 어떠한 어려움이 발생 하는가에 관한 것을 알아보기 위해 실험하였다. 또한 두 대의 멀티 에이전트 로봇을 공을 잡는 하나의 알고리즘으로 정확한 위치로 로봇이 목표물을 이동 시키는 것을 구현해 내는데 그 착안점을 두었다.

Fig. 37은 두 대의 로봇 컨트롤을 위해 새롭게 구성된 사용자 인터페이스를 나타낸다. 오른쪽 흰 바탕은 시뮬레이션 모듈 부분이고 그 아래는 실제 화상이 전송되어 들어오는 화면이다.

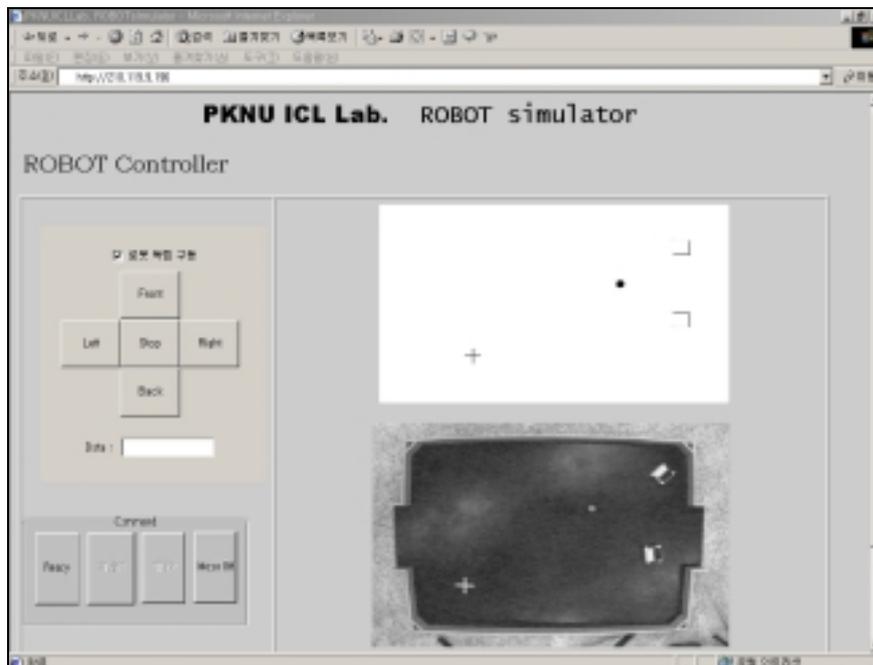


Fig. 37 2대의 이동로봇 제어를 위한 GUI

두 대의 로봇이 하나의 물체를 잡기 위해선 먼저 로봇의 위치를 알아야 한다. 그렇게 하기 위해 비전 시스템에서는 로봇의 꼭지점을 찾아내어

로봇이 목표물에 도달하여 목표물을 두 대의 로봇이 잡을 때의 각도와 이동할 때의 포메이션을 정확하게 유지하는데 도움을 주었다. Fig. 38은 비전시스템이 실제로 포착한 화상을 나타낸다.

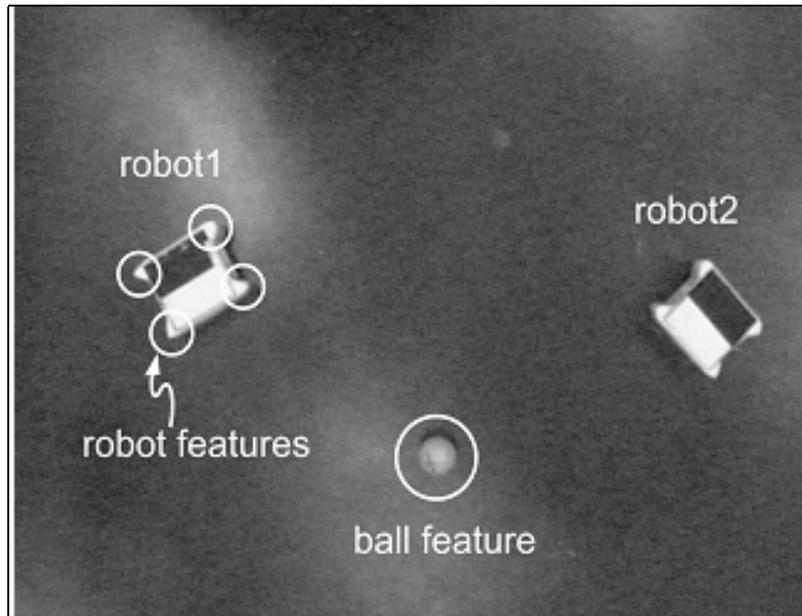


Fig. 38 Image feature point setting

로봇이 물체를 골 포지션까지 정확하게 이동시키기 위해선 먼저 물체를 정확하게 잡아야 한다. 공을 잡을 수 있는 가장 좋은 방법으로 45도의 각도를 가진 두 대의 로봇이 접근하여 공을 잡으면 가장 잘 잡을 수 있다는 것을 알게 되었다. Fig. 39에서 나타낸 것처럼 공까지의 거리가 일정해지면 로봇 두 대는 그림처럼 45도의 각도를 가지고 목표물인 공에 접근하게 된다. 로봇은 또한 방향과 로봇의 ID를 알게 해주는 패치가 달린 외부 케이스를 목표물을 잡기 편하도록 디자인 하였다. 그래서 네 귀퉁이가 뿔족하게 되어있다. 그렇게 디자인하였기 때문에 목표물인 공을 잡기에 훨씬 수월했으며 목표 지점까지 공을 이동시키는 동안에도 공이 두 대의

로봇사이에서 벗어나지 않을 수 있었다.

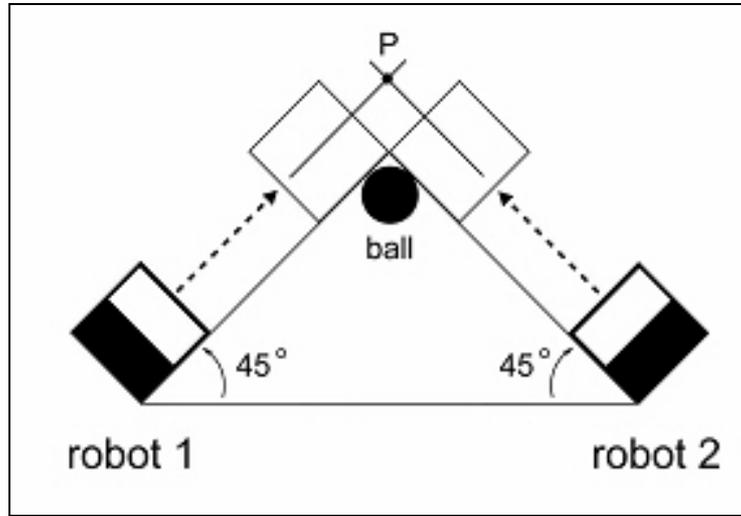


Fig. 39 The position of robots for caching the ball

Fig. 40은 로봇이 공을 잡고 난 뒤 골 포지션까지 이동하기 위해서 골 포지션과의 각도를 두 대의 로봇이 맞추어 가는 것을 나타낸 것이다. 수평과의 각도 α 에 골 포지션까지의 각도 θ 를 구하여 방향을 틀어 골 포지션까지 직진으로 목표물을 옮겨 가게 된다.

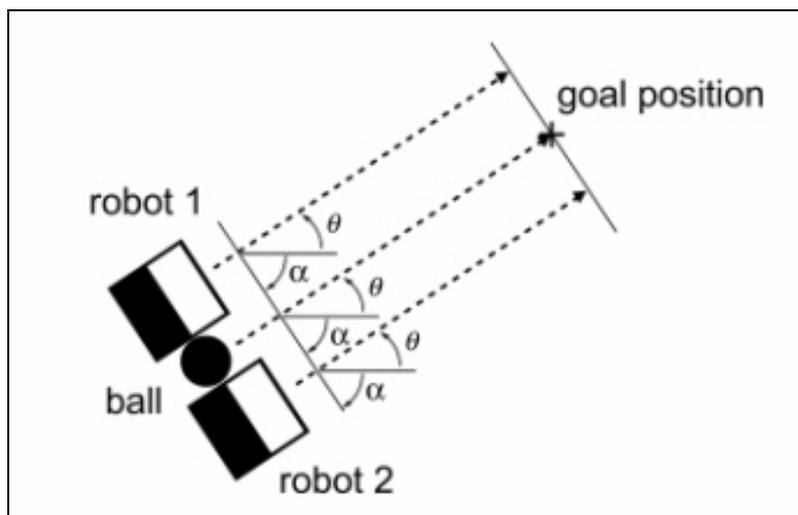


Fig. 40 The position of robots for maintaining the formation

로봇이 이동하기 전에 정확하게 목표물까지 이동하는가를 미리 알아보기 위해 클라이언트측의 Applet창으로 사용자가 입력한 이동 목표 값에 대한 시뮬레이션을 GUI를 통해 보여주게 된다. 그렇게 함으로써 로봇이 위험에 처하는 것을 미리 알아 볼 수 있게 하였다. 그렇게 하여 나타난 시뮬레이션 결과가 Fig. 41과 같다. 시뮬레이션에서는 사용자가 입력한 목표위치까지의 이동 값이 로봇이 목표물을 목표위치에 이동시키는데 아무 이상이 없음을 알 수 있다.

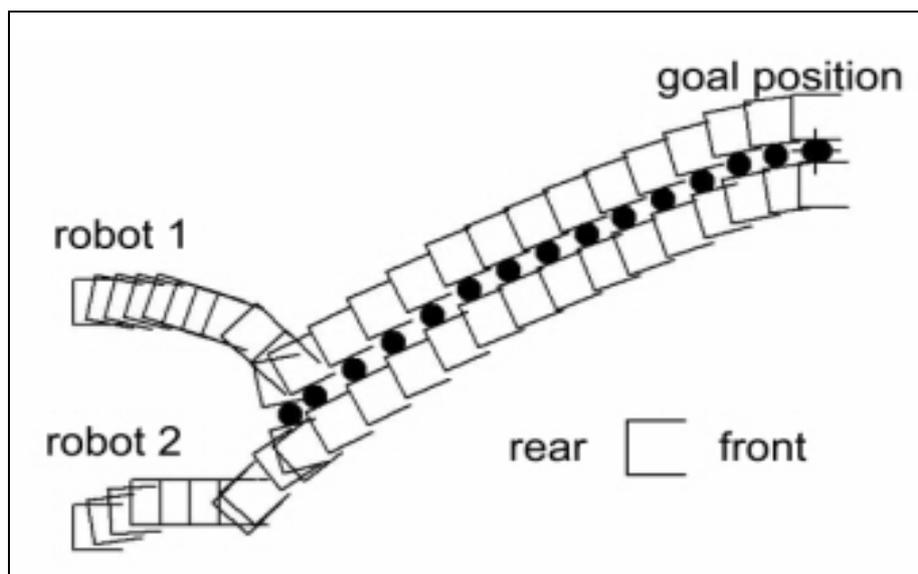


Fig. 41 The result of simulation

그렇게 하여 나타난 결과가 Fig. 42-45까지 이다. Fig. 42의 1은 사용자가 골 포지션을 클릭했을 때 로봇의 초기위치를 나타낸다. 2는 공을 잡기위해 45도의 각도로 두 대의 로봇이 위치한 것을 나타낸다.

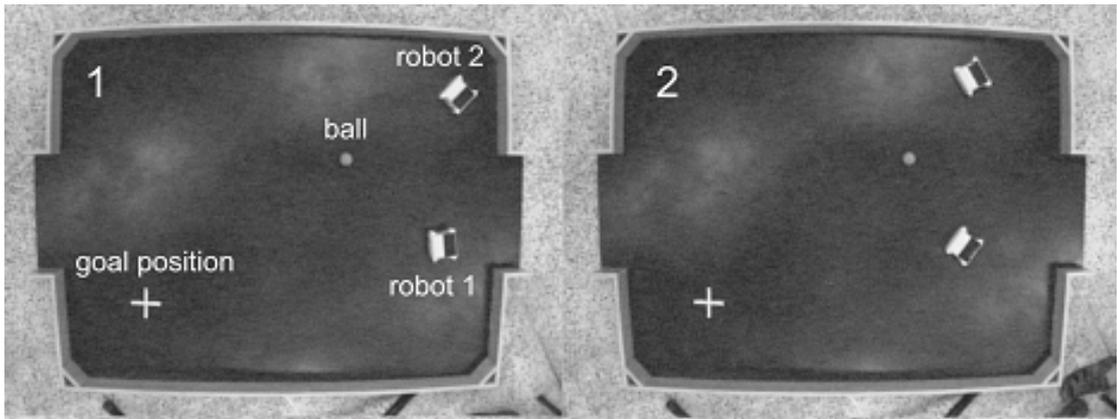


Fig. 42 Movement the multi robot

Fig. 43은 공을 잡기위해 45도의 각도로 접근한 로봇이 공을 잡은 뒤 목표위치까지 이동하기 위해 목표위치와의 각도를 회전하는 모습을 나타낸다.

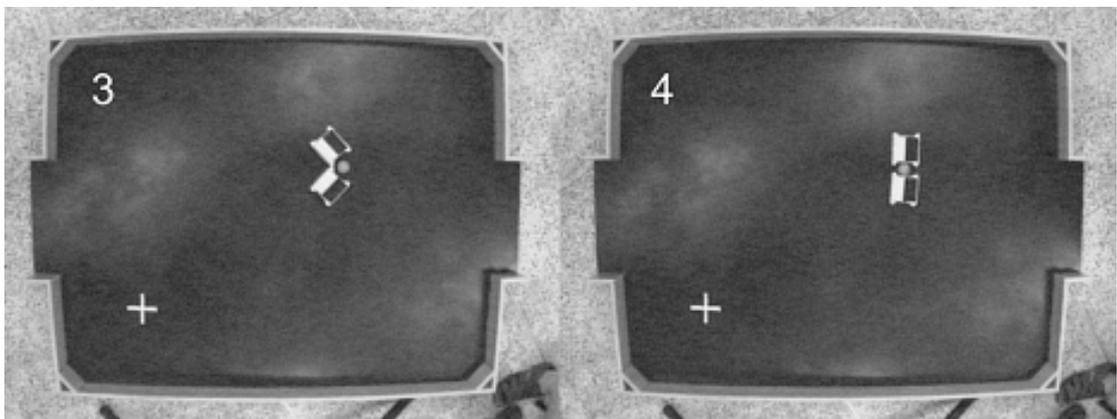


Fig. 43 Cached ball

Fig. 44는 두 대의 로봇이 포메이션을 유지하며 골포지션으로 이동중인 모습을 나타낸다.

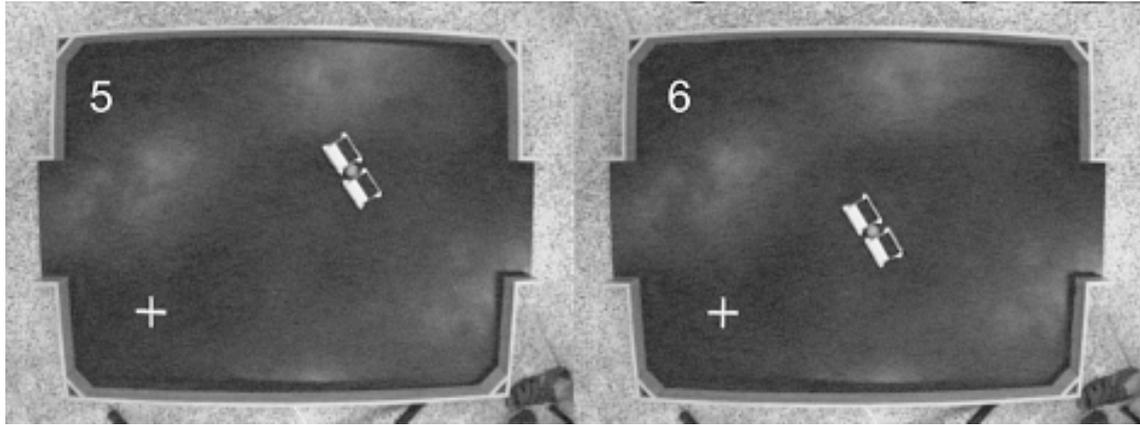


Fig. 44 Move to goal position

Fig. 45는 골 포지션에 정확하게 도착한 두 대의 로봇을 나타낸다.

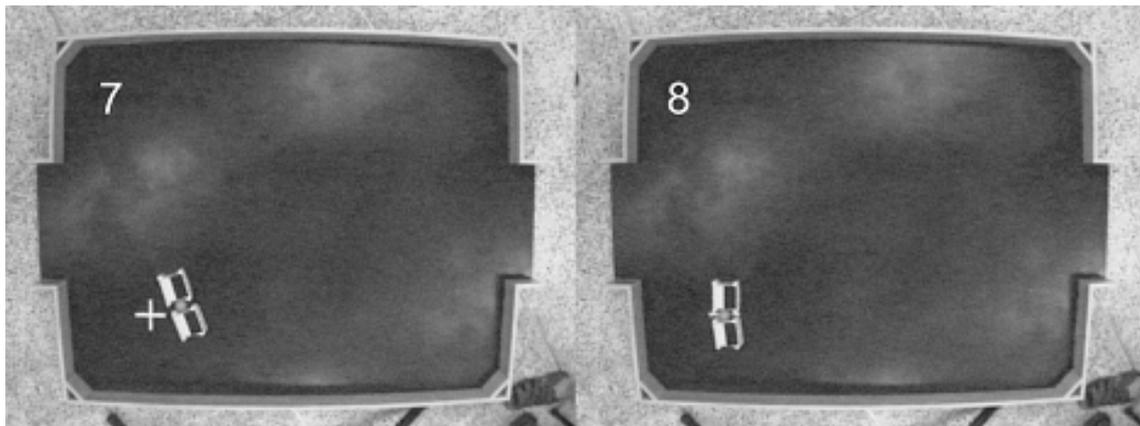


Fig. 45 To arrive the goal

5. 결론 및 향후 연구 과제

인터넷은 가장 대중화된 통신수단으로 발전하였다. 인터넷을 이용하여 컨트롤 하게 되면 가장 간편하게 제어 할 수 있다는 장점이 있다. 본 논문에서는 인터넷을 이용하여 모빌로봇을 컨트롤 하였다. 인터넷을 이용한 기존의 모빌로봇 컨트롤 방법은 아주 단순한 기능밖에 수행하지 못하였다. 본 논문에서는 로봇을 움직이게 하는 데이터 값을 이용하여 미리 로봇의 움직임을 확인 한 후 실제 로봇을 움직이는 방법을 사용하여 로봇이 위험에 처할 수 있는 상황을 줄일 수 있었다. 또한 Login module을 사용하여 한 명의 사용자만으로 컨트롤에 방해가 없게 하였다. 또한 채팅 모듈을 사용하여 원격지에서 사용자가 먼 곳에 떨어져 있는 로봇 주변의 관리자에게 문자정보와 음성정보를 전 할 수 있는 기능 또한 사용 하였다. 이러한 기능들의 사용으로 로봇과 사용자가 보다 편리하고 안전하게 로봇을 컨트롤 할 수 있었다.

본 논문에서는 두 가지 실험을 하여 보았다. 첫번째는 로봇을 보다 빠르게 이동 시키기 위해 퍼지 컨트롤러를 사용하여 이동로봇을 제어하여 보았고, 두 번째는 하나의 목표물을 이동시키기 위해 두 대의 로봇이 협조하여 정확하게 목표위치까지 목표물을 이동시키는 실험을 하였다. 이 두 가지 실험 모두 인터넷 기반 하에서 성공적으로 작동함을 확인 할 수 있었다.

실제 외부 세계를 컨트롤 하기 위해 여러 가지 다양한 기술들이 요구 되고 있다. 인터넷의 장점 만큼이나 다양하게 많은 문제점 또한 가지고 있다.

인터넷은 시간지연이라는 약점을 가지고 있다. 그렇기 때문에 실시간 제어에는 한계가 있다. 그것을 극복하기 위해선 보다 빠른 회선의 개발과 프로토콜의 연구, 웹 캐쉬 등의 기법을 이용한 메모리의 최적화 등의 연구가 보다 활성화 되어야 할 것이다.

앞으로의 인터넷은 개개인이 인터넷을 지니고 다니는 Mobile Communication Service로 발전해 나갈 것이다. 웹 브라우저를 탑재한 시계나 PDA, Mobile Phone으로 공장의 기계를 가동시키고 제어하는 시대가 곧 도래 할 것이라 예상되고 또한 사용자가 보다 편리하게 웹 기반 제어를 할 수 있도록 보다 많은 연구가 필요하리라 생각된다.

참 고 문 헌

[1]. S. Hyati and R. Volpe, "The Rocky 7 Rover : A Mars Sciencecraft Prototype", Proc. of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, New Mexico, pp. 2458~2464, April, 1997.

[2]. <http://cwis.usc.edu/dept/garden>

[3]. T. M. Chen and R. C. Luo, "Remote Supervisory Control of Autonomous Mobile Robot Via World Wide Web", ISIE'97-Guimaraes, Portugal, pp. 60~64, 1997.

[4]. Ren C. Luo, Wei Zen Lee, Jyh Hwa Chou, and Hou Tin Leong, "Tele-Control of Rapid Prototyping Machine Via Internet for Automated Tele-Manufacturing", Proc. of the 1999 IEEE International Conference on Robotics and Automation, pp. 2203~2208, 1999.

[5]. T. Fitric. "live Remote Control of a Robot via The Internet", IEEE Robotics & Automation Magazine, pp. 7~8, September 1999.

[6]. Denis Gracanin, Maja Matijasevic, Nikos C. Tsourveloudis, Kimon P. Valavanis, "Virtual Reality Testbed for Mobile Robots" ISIE'99-Bled, Slovenia,

1999

[7]. Roland Siegwart, Patrick Saucy, "Interacting Mobile Robots on the Web"
ICRA'99, Detroit, Mi, USA, 1999

[8]. R. C. Gonzalez and R. E. Wood "Digital Image Processing", Addison
Wesley, 1992.

[9]. G. Campion, G. Bastin and B. D'Andrea Novel, "Structural Properties and
Classification of Kinematic and Dynamics Models of Wheeled Mobile Robots",
IEEE Trans Robotics and Automation, vol. 12, no. 1, pp. 47-62, Feb. 1996.

[10]. J. Santos-Victor, "Vision-Based Remote Control of Cellular Robots,"
Robotics and Autonomous System, Vol. 23, pp. 221-234, 1998.

[11]. K. Ogata, "Discrete-Time Control System. Englewood Cliffs, NJ:
Prentice-Hall, 1995.

[12]. Cha, J.H., Lee, S.G., Jeoun, H.Y. : "Internet Based Remote Control
System Using Virtual Reality. : 한국 CAD/CAM 학회 논문집, pp. 88-94, 2000. 3

[13]. Kong, S.H., "인터넷을 통한 생산설비의 원격제어", FA저널, pp. 90-96, 2001