

## 인터넷을 통한 로봇의 원격 제어

이동주\*/양태규\*\*

### 요 약

본 논문에서는 웹 브라우저를 이용하여 로봇을 원격 제어한다. 4대의 독립된 PC를 사용하여 각각의 기능을 분리하고 상호 연동하는 구조로 PC를 연결하였다. 조작자가 시간과 장소에 제약받지 않고 웹 브라우저를 이용하여 제어값을 입력하면 제어값은 웹 서버를 거쳐 데이터베이스에 저장된다. 로봇과 연결된 PC는 데이터베이스에서 1ms 간격으로 제어값을 입력받아 제어값의 변화가 있을 경우만 제어값의 변화량으로 로봇이 동작한다. 스텝모터를 이용하여 간단한 로봇을 제작하고 이를 웹 브라우저를 이용하여 실제 동작시킨다. 조작자가 원하는 제어값이 최종 수신단의 로봇에 제대로 전달되어 동작하는지 확인하고 웹 브라우저를 이용한 로봇제어의 가능성을 확인하였다.

## 1. 서론

원격제어는 크게 유선과 무선의 연결매체를 이용하여 이루어진다. 유선이나 무선망을 구축할 경우 많은 비용이 필요하나 인터넷을 이용한 원격제어의 경우 전 세계로 연결된 인터넷망을 이용하기 때문에 망 구성에 따른 추가 설치비가 필요하지 않다. 원격제어의 경우 원격의 로봇을 직접 조작하거나 로봇의 상태를 알 수 없기 때문에 조작자가 원격지에서 로봇의 상태를 알 수 있도록 하는 기술이나 실제 작업환경과 유사한 환경을 제공하는 기술이 필요하다.

이 부분에서는 Berkeley의 Paul Debevec이 영상 기반으로 성공적으로 수행한 예가 있으며 CMU의 Kanade와 메릴랜드 대학의 Aloimonos가 각기 다른 방식의 3차원 복원 기술을 사용하

여 수행한 바 있다. 인터넷을 이용한 원격제어는 서버와 클라이언트의 개념을 도입하여 조작자는 클라이언트로 설정하고 제어대상체를 서버로 하는 구조를 가진다.

현재 연구되고 있는 분야는 각종 MPU, MCU를 이용한 Micro Processor 접근방법, Embedded real time system, PDA와 같은 RTOS방법 그리고 웹 브라우저를 이용한 WTTS 등 여러 분야가 있다<sup>[1][2]</sup>. 실례로 천체 망원경 원격제어(Univ. of Bradford)<sup>[3]</sup>, SCARA 로봇 원격 제어(univ. of southern california)<sup>[4]</sup>, 6축 로봇 원격제어(Univ. of Western Australia)<sup>[5]</sup>, Eric Paulos의 원격 전 시관 관람 시스템(Univ. of berkely)<sup>[6]</sup> 등이 있다.

인터넷 기술은 소프트웨어적인 부분만이 아니라 하드웨어적인 부분에서도 많은 발전을 이루고 있다. 시간지연 문제, 데이터 손실, 다중 사용자로 인한 보안의 문제, 원활한 통신을 위한 최저 통신 속도 보장 및 향상, 새로운 형태의 미디어의 수용 등 여러 가지 개선을 통하여 미래

\* (주)무비스 하드웨어 개발팀

\*\* 목원대학교 공과대학 전자정보통신공학과

의 인터넷은 유선과 무선의 역할을 충분히 수행할 것이다<sup>[7][8]</sup>.

본 논문에서는 웹 브라우저를 이용하여 로봇을 구동시키고 로봇 동작의 정확성을 확인하여 시간과 장소에 제약받지 않는 제어 가능한 로봇 시스템의 모델을 제시한다. 웹 브라우저를 이용하여 원격 로봇에게 제어값을 보내기 위해서는 웹서버를 거쳐 데이터베이스 서버에 저장된다. 이렇게 저장된 제어값은 로봇과 직접 연결된 PC의 컨트롤 프로그램에 의해서 1ms 간격으로 데이터베이스 서버의 제어값을 입력받아서 로봇 제어 명령을 전달하게 된다. 웹서버는 시간과 장소에 구애받지 않고 웹 브라우저만 있다면 로봇을 원하는 상태로 조작 가능하다. 웹 브라우저를 이용하여 로봇을 제어할 경우 그 정확성을 측정하기 위해 로봇을 동작시켰다. 또한 소켓 프로그램을 이용하여 원격 로봇을 제어한 결과를 비교하여 웹 브라우저를 이용하여 로봇을 원격으로 제어해 그 결과를 비교하여 동작의 정확성을 측정한다.

## II. 원격제어의 개요

인터넷의 발달과 비전 시스템의 발전으로 낮은 비용으로 원격 시스템을 만들 수 있다. 소켓 프로그램은 TCP/IP를 이용한다. TCP/IP는 기종이 서로 다른 컴퓨터 시스템을 서로 연결해

이터를 전송하기 위한 통신 프로토콜이다. 유닉스(UNIX) 운영체제 내에 채용되었으며 인터넷에도 사용되었고, 유닉스와 인터넷 사용이 늘어나면서 TCP/IP는 네트워크상에서 데이터를 전송하는 표준이 되었다. TCP/IP는 OSI 모델을 직접 따르지 않는다. 표 1은 TCP/IP계층을 보여주고 있다.

소켓 프로그램은 클라이언트/서버 구조로 동작하는데 시스템의 자원에 접근할 때 두 종류의 구별되는 개체가 관여한다는 사실을 반영한다.

첫 번째 개체는 클라이언트로 시스템의 자원에 접근하고자 요청을 보내는 쪽이다. 두 번째 개체는 서버로 클라이언트의 요청이 들어올 때만 작동한다. 서버는 자원들을 직접 다룰 수 있으며 그 결과를 클라이언트에게 보낸다. 서버는 전적으로 클라이언트의 요청을 기다리는 컴퓨터이고 요청이 들어오면 즉시 반응한다. 소켓을 이용한 인터넷 원격 제어의 경우 클라이언트와 서버와의 데이터 송수신으로 원격 제어 가능하다.

원격조작 시스템은 제어 기법에 따라 단방향 제어와 양방향 제어로 분류할 수 있다. 단방향 제어는 작업할 때 서버의 작업환경에서 발생하는 환경이 작업자에게 전달되지 않으나, 양방향 제어는 작업 환경이 작업자에게 전달되어 작업성을 높일 수 있다. 단방향 제어의 경우 작업자는 시각정보만을 이용하여 비교적 간단한 작업의 수행만이 가능하다. 그러나 양방향 제어시스템에서는 작업 환경이 작업자에게 전달되어 이에 따라 작업자는 기존 운동을 수정하여 보다

〈표 1〉 TCP/IP 계층 구조

Application	FTP RFC959	SMTP RFC821	TELNET RTC859	SNMP RFC1098
Transport	TCP : RFC793		UDP : RFC 768	
Internet	ARP : RFC826	IP : RFC791	ICMP : RFC792	
Network Access	Physical Network			
Physical				

복잡한 작업의 수행이 가능하다. 그러나 양방향 시스템의 경우 거리가 멀 경우 시간지연에 의해 전체적으로 시스템이 불안정하게 동작할 수 있다. 이러한 단점을 극복하기 위해 슬레이브에 작업장의 가상환경을 제공하여 문제를 해결하는 기법을 사용하고 있고 또는 작업자의 제어값을 논리적인 알고리즘을 이용하여 학습하여 원격 제어하는 방법을 사용한다. 다른 제어 기법으로는 소켓 프로그램을 사용하지 않고 CGI나 JAVA를 사용하여 웹 브라우저를 이용하여 원격 제어하는 방법이 있다.

Common Gateway Interface(CGI)는 외부 프로그램과 웹서버간의 연결 역할을 하기 위한 규약으로 데이터의 흐름은 클라이언트에서 서버로 TCP/IP 연결을 통해 데이터를 요청하면 서버가 CGI프로그램에게 실행을 하도록 요청하고 CGI 프로그램에서 서버로 출력이 전달된 후 적절한 MIME헤더를 갖고 서버가 클라이언트로 응답한 후, 연결이 해제된다.

또한, JAVA는 C와 C++에서 포인터와 다중 상속 같은 일부 기능들을 제외하고 보안과 멀티스레드와 같은 새로운 기능들을 추가함으로써 만들어진 객체지향 프로그래밍 언어다. 자바는 멀티 플랫폼으로 유닉스나 PC의 윈도우OS, 매킨토시 등 어떠한 환경에서도 동작할 수 있고, 기존 OS용 응용프로그램들이 인터넷을 통해 자바용으로 재사용 될 경우에는 기존 OS 패러다임과는 매우 다른 독자적인 OS가 된다. JAVA는 사용상의 편의성과 이식성이 높고, 스마트카드에서 슈퍼컴퓨터까지 기종에 관계없이 폭넓게 활용할 수 있는 장점이 있다. 웹 브라우저를 이용하면 소켓 프로그램과 같이 전용 구동 프로그램이 필요 없이 인터넷이 연결된 곳 어디서든 대상체를 원격 제어할 수 있다는 장점이 있으나 웹 서버를 경유하기 때문에 시간지연이 많이 발

생하는 단점이 있다.

### III. 웹 브라우저를 통한 원격 로봇제어

#### 3.1 웹 브라우저를 이용한 데이터 송수신 구조

웹 브라우저에서 동작하는 로봇을 구성하기 위해 4대의 PC를 사용하였다. 표 2는 원격으로 각각 떨어져 있는 4대의 PC의 역할과 사용된 프로그램을 나열한 것이다.

웹 브라우저(PC1)를 통하여 입력된 로봇 제어값은 웹서버(PC2)를 경유하여 데이터베이스 서버(PC3)에 저장된다. 이렇게 저장된 데이터는 다시 웹서버(PC2)를 거쳐 웹 브라우저(PC1)에 표시된다. 로봇과 연결된 컨트롤 PC(PC4)에서는 1ms 간격으로 데이터베이스 서버(PC3)에 접속하여 제어값의 변화가 있을 경우 제어값을 받아 들여 로봇을 제어한다. 로봇은 컨트롤 PC(PC4)에서 입력받은 제어값으로 동작하게 되고 로봇 끝단에 위치한 카메라를 통하여 로봇의 움직임을 컨트롤 PC(PC4)의 웹서버에서 다시 임의의 웹 브라우저(PC1)에 동영상을 제공한다.

이러한 연결구조에서 웹 브라우저와 웹서버는 ASP를 통해 양방향 정보 제공의 기법을 사용하였고 웹 서버와 데이터베이스 서버, 그리고 데이터베이스 서버와 로봇 컨트롤 PC와의 연결에서 ODBC를 사용하였다. 그림 1은 전체적인 시스템 블록선도를 보여주고 있다.

〈표 2〉 PC 사용 목적 및 사용 프로그램

컴퓨터 기능	PC1	PC2	PC3	PC4
용도	Web Browser (Client)	Web Server	DB Server	Robot Control (Server)
OS	win95,98,NT	Win98	Win2000	win98
Software	Explorer5.0	IIS Server	MS-Sql 7.0	Robot Control 전용 프로그램 (Visaul Basic)
IP		203.230.150.119	203.230.150.134	203.230.150.133

### 3.2 조작자 웹브라우저 및 로봇 구 동 프로그램

그림 2는 인터넷 상에서 조작자가 웹브라우저를 이용하여 웹서버에 초기 접속 모습으로 조작자의 인증을 거친 후 로봇을 컨트롤할 수 있는 초기 인증 과정이다. 초기 인증과정을 거치면 그림 3과 같이 로봇을 제어하기 위한 데이터 입력 과정을 거친다. 웹브라우저에서 전송되는 컨트롤 데이터는 웹서버에서 ODBC로 연결된 데이터베이스 서버에 저장된다. X는 수평방향의 각도이고 Y는 수직방향의 각도이다. 수평방향과 수직방향의 입력값은 정수값으로 Null 값을 허용하지 않는다. 실제 로봇의 움직임은 수평방향의 컨트롤 데이터를 먼저 처리하고 처리가 끝나면 수직방향의 컨트롤 데이터를 처리하게 된다. 입력된 데이터가 웹서버를 거쳐 데이터베이스 서버에 저장된 후 제어값이 그림 4와 같이 조작자 웹브라우저에 표현된다. 조작자는 양방향 정보 제공 언어인 ASP를 통하여 데이터베이스에 저장된 제어값을 확인할 수 있고 이를 통하여 원하는 제어값이 데이터 베이스에 저장됨을 확인한다. 또한 로봇의 끝단에 위치한 카메라에를 통해 조작자가 원하는 위치의 동영상을 볼 수 있으며 이를 통해 로봇의 상태를 인지하여 로봇

조작의 움직임을 컨트롤 데이터와 비교 할 수 있다. 그림 5는 로봇 컨트롤 프로그램으로 Visual Basic을 이용하여 제작하였다. 데이터베이스 서버에 1ms 간격으로 접속한 후 데이터의 변화가 있을 경우 컨트롤 프로그램에 수평과, 수직방향의 제어값이 표시되고 한 스텝당 해상도가 1.8도 이기 때문에 입력받은 제어 값을 1.8로 나누어 로봇을 제어하기 위한 펄스의 수를 결정하였다. 수평방향은 정회전만 가능하고 수직 방향의 입력값이 바로 직전의 입력값과 비교하여 클 경우는 정회전, 작을 경우는 역회전으로 결정하고 동작전의 제어값과 비교하여 오차값 만큼 펄스를 발생시켜 로봇을 구동시킨다. 또한 입력값에 따라 모터의 회전 방향을 표시하여 로봇의 방향이 어디를 향하고 있는지 알수 있게 하였으며 로봇의 동작속도는 스텝모터의 펄스 폭 간격을 80ms로 설정하였다.

### 3.3 로봇 구성 및 컨트롤러

로봇은 조작자가 원하는 곳을 관측할 수 있도록 구성하였고 동작 모터는 스텝모터를 사용하였다. 로봇과 연결된 컴퓨터에서 모터까지 연결된 구조는 그림 6과 같다. 제어 값이 데이터베이스 서버에서 컨트롤 프로그램으로 값이 1ms 간격으로 입력되고 값의 변화가 있을 경우 로봇

이 구동하게 된다. 원격의 조작자가 보내온 제어값을 패러럴 포트를 통하여 로봇에게 값이 전달되어 로봇이 동작한다.

먼저 수평방향의 제어값은 COM1 포트의 0x04, 0x05, 0x06, 0x07 번지로 보내어져 컨트롤러2에 의해 수평방향의 스텝모터를 동작시키고, 수평방향의 모터 회전이 끝나면 수직 방향의 데이터값이 COM1 포트의 0x40, 0x50, 0x60, 0x70 번지로 보내어져 컨트롤러 1에 의해 수직방향의 스텝모터를 동작시킨다.

사진 1은 제작한 로봇의 외부 모습으로 두개의 스텝모터와 한 개의 디지털 카메라를 사용하여 제작하였다. 제작한 로봇은 수평방향으로 360도 회전이 가능하며 수직으로는 180도 회전이 가능하다. 수직방향의 회전 방향은 데이터베이스에 저장된 제어값의 변화량으로 결정된다. 수평방향의 스텝모터는 360도 회전 가능하게 제작하였고, 수직방향의 스텝모터는 조작자의 제어값에 따라 정회전, 역회전 180도 회전 가능하도록 제작하였다. 수직방향 스텝모터의 끝단에는 디지털 카메라를 부착하여 조작자가 원하는 방향을 볼 수 있도록 하였고 컨트롤 컴퓨터의 웹서버를 거쳐 조작자의 웹브라우저에 조작자가 원하는 방향의 영상을 보낸다.

#### IV. 실험 및 결과 고찰

실험에서는 웹브라우저를 이용하여 원격 로봇 제어를 했을 경우와 웹브라우저를 사용하고 웹프로그램으로 원격제어를 했을 경우의 결과를 비교하여 웹브라우저를 이용한 원격 로봇 제어의 동작의 정확성을 확인하고 웹브라우저를 이용한 원격 로봇 제어의 가능성을 증명한다.

웹브라우저를 이용한 원격 로봇 제어를 위해 임의의 제어값을 웹브라우저에서 입력하여 원격 로봇을 동작시키고 로봇이 원하는 제어값으로 동작하도록 실험했다. 데이터베이스 서버의 초기값을 수평방향과 수직방향을 모두 0으로 조정하고 실험하였다. 먼저 임의의 장소의 웹브라우저에서 웹서버에 접속한 후 수평방향과 수직방향의 제어값을 입력하여 로봇이 정확히 동작하는지 측정하였다. 웹브라우저를 이용한 로봇 제어의 정확성을 판별하기 위해 동일한 제어값을 가지고 직접 로봇을 구동하였고 소켓프로그램을 제작하여 웹에서 로봇을 구동시켰다. 또한, 인터넷에서 발생하는 시간지연을 측정하기 위해 ICMP 프로토콜을 이용하여 핑테스트를 실시하여 거리에 따른 대략적인 지연시간을 측정하였다. 로봇의 동작 속도는 80ms의 펄스 간격으로 정하고 제어값이 로봇에 전달되면 수평방향으로 동작하고 동작이 끝나면 수직방향의 동작을 하도록 설정하였다.

그림 7은 웹브라우저에서 조작자가 수평방향과 수직방향의 제어값을 입력하였을 때 데이터베이스 서버인 MS-SQL서버에 제어값이 저장된다. 조작자가 원격 로봇이 동작하도록 제어값을 입력했을 경우 제어값들이 차례로 저장된다. 이렇게 저장된 컨트롤 데이터들은 다시 조작자의 웹브라우저에 표시되어 조작자가 로봇의 상태가 어떠한지 파악할 수 있다. 이렇게 저장된 컨트롤 데이터들은 로봇과 연결된 컨트롤 프로그램에서는 데이터베이스 서버에 접속하여 제어값만큼 수평방향의 스텝모터가 회전하고 회전이 끝나면 다시 수직방향의 스텝모터가 회전한 후 다시 데이터베이스 서버에 1 ms 간격으로 다시 접속하여 제어값의 변화를 체크한다. 웹브라우저에서 제어값을 입력했을 때 로봇의 끝단에 위치한 카메라를 통해 로봇의 움직임을 확인할 수

있다.

그림 8과 그림 9는 로봇의 끝단에 위치한 카메라를 통해 조작자가 원하는 방향의 영상으로 로봇이 움직이기 전의 모습과 움직인 후의 모습이다. 그림 8은 수평방향으로 0 수직방향으로 0으로 그 값을 초기화했을 경우 로봇 끝단에 위치한 카메라로 보여지는 초기 영상이다. 그림 9는 수평방향으로 200 수직방향으로 100의 값을 보냈을 경우 로봇 끝단에 위치한 카메라로 보여지는 영상으로 조작자가 원하는 방향의 영상이다.

그림 10과 컨트롤 프로그램으로 입력된 제어값에 의해 그 값들이 어떻게 처리되는지 보여준다. 그림 10은 수평방향 200 수직방향 100의 제어값이 입력된 후의 모습으로 수평방향의 경우는 정회전만을, 수직방향의 경우는 입력값에 따라 정회전과 역회전이 결정되고 로봇에 전달될 펄스값이 정해진다.

그림 11은 로봇의 움직임을 로봇 끝단의 디지털 카메라를 통해 웹브라우저에 동영상을 제공하고 조작자가 로봇의 상태를 알 수 있게 했다. 웹브라우저에 동영상을 보내기 위해서 소프트웨어 Inetcam을 사용하였다. 그림 12는 Inetcam에서 제공되는 것으로 조작자의 웹브라우저에 동영상의 정보를 표시해준다. 조작자가 웹브라우저에서 입력한 제어값에 의해 원격 로봇을 동작시켰다.

웹브라우저를 이용한 로봇의 움직임이 정확하게 동작하였는지 확인하기 위해 동일한 제어값을 가지고 컨트롤 PC에서 직접 로봇을 제어했다. 또한 웹브라우저를 사용하지 않고 웹에서 원격제어를 하기 위해 소켓 프로그램을 제작하여 로봇을 동작시켜 동작 특성을 비교하였다.

또한, 오실로스코프를 이용하여 로봇을 직접 제어했을 경우, 소켓 프로그램을 이용하여 웹상에서 제어했을 경우, 웹브라우저를 이용하여

제어했을 경우 동일한 펄스 신호를 얻었다. 실험하는 과정에서 지연요소로 인하여 시간지연이 발생하였다. 웹브라우저를 사용하였을 경우 지연이 가장 많이 발생하였다.

그림 13은 소켓 프로그램을 이용한 원격 로봇 제어의 블록선도이고 그림 14는 웹브라우저를 이용한 원격 로봇 제어의 블록선도이다. 그림 13은 소켓 프로그램으로 로봇을 구동할 경우 인터넷에서 최단 거리로 연결된다. 그러나 웹브라우저를 이용하여 로봇을 제어하기 위하여 본 실험에서 제안한 데이터 송수신 구조에서 로봇을 구동할 경우는 그림 14와 같이 최단 거리로 연결되는 구조가 아니라 여러 곳을 경유하는 형태를 가진다.

실험에서 소켓 프로그램을 이용하여 로봇이 동작하기까지의 시간은 클라이언트에서 로봇과 연결된 서버까지의 핑테스트 값인 RTT(Round Trip Time)의 1/2의 시간이 걸린다. 하지만 웹브라우저를 이용하면 로봇을 동작시키기까지 RTT는 각 4대의 PC간의 거리 즉 각 PC에서 다음 PC까지의 핑테스트 시간의 1/2을 더한 값과 PC의 동작 상태를 더한 값이 시간지연이다. 결과적으로 소켓 프로그램보다 이 논문에서 제안한 PC 간의 연결 구조는 시간지연이 크게 나타났다. 이러한 시간 지연은 데이터의 크기에 의해서도 발생하지만 거리에 따른 시간지연이 크게 작용한다.

본 실험에서는 4byte의 작은 데이터를 가지고 실험했기 때문에 데이터 크기에 의한 지연은 실제 로봇이 구동되기까지의 과정에서는 무시할 정도로 작았고 거리에 따른 지연요소가 크게 작용하였다. 로봇끝단에 위치한 카메라에서 보내어지는 동영상이 데이터의 크기가 크기 때문에 거리에 따른 지연보다는 데이터의 크기에 따른 지연요소가 크게 작용하였다.

그림 15, 그림 16, 그림 17에서 측정된 값들은 10km, 200km 2000km이상 거리의 수신단에 64 bit의 데이터를 보내서 되돌아온 시간을 측정 한 값이다.

실험을 통하여 거리가 멀수록 지연시간이 많이 발생한다는 사실을 확인했다. 그러나 지연시간이 일정하지 않고 시간별로 또는 수시로 발생하여 시간의 비 선형성이 나타났다. 그림을 통하여 인터넷 시간지연은 작은 데이터 전송에 있어서 거리에 따라 시간지연이 크게 발생함을 알 수 있다.

## V. 결론

4대의 PC가 상호 연동하는 형태의 데이터 연결 구조를 사용하였기에 시간 지연이 발생했으나 조작자의 제어값은 최종 수신단인 로봇에 정확히 전달되었고 로봇의 움직임도 이상 없이 동작하였다. 실제로 4대의 PC는 근거리에 위치해 있지만 랜을 통하여 인터넷으로 연결되어 제안된 데이터 송수신 구조를 통하여 인터넷으로 데이터를 주고받았다. 동영상을 제외한 모든 데이

<표 3> 거리에 따른 시간지연

	거리(km)	평균 지연 시간(ms)	손실률(%)	리우터 수
충남대	10	115	10	7
서울대	200	132	18	9
옥스퍼드대	2000 over	642	8	23

본 실험에서는 4byte의 크기로 실험을 실시하였기 때문에 웹 브라우저를 이용하지 않고 소켓 프로그램을 사용하였을 경우, 표 3과 같이 10Km 내외는 115ms, 200Km 내외는 132ms, 2000Km 내외는 642ms의 시간 지연이 발생함을 알 수 있었다. 웹 브라우저를 이용하여 실험을 하였을 경우의 시간지연은 소켓 프로그램을 이용하여 실험을 했을 경우보다 시간지연이 더 발생했고 로봇 끝단에서 보내지는 동영상의 경우는 데이터의 값이 크기 때문에 더 많은 지연이 발생했다. 그러나 시간 지연이 발생하였음에도 불구하고 실험, 소켓 프로그램 그리고 웹 브라우저에서 로봇을 동작시킨 결과가 일치하였다.

터들은 텍스트 형태의 4byte 데이터가 송수신되고 있으므로 큰 시간지연은 발생하지 않았다.

웹 브라우저를 이용한 실제 로봇의 움직임을 측정 한 결과 로봇 동작의 정확성을 검증하였고 시간 지연에 많은 영향을 미치지 않는 로봇 제어의 경우 매우 안정적으로 동작하는 모델임을 보였다. 웹 브라우저를 이용한 원격제어의 경우 원격제어를 하지 않거나 소켓 프로그램을 이용한 원격제어의 경우보다 시간지연이 많이 걸리지만 언제 어디서나 특정한 프로그램 필요 없이 웹 브라우저만 있다면 조작자가 원하는 동작을 원격으로 조작 할 수 있다는 점에서 그 중요함과 응용은 더욱 증대될 것이다

## 참고문헌

- [1] McCanne, S., and Jacobson, V. 1995(San Francisco, CA), VIC: A Flexible Framework for Packet Video. ACM Multimedia. New York: ACM, pp.1-12
- [2] Robert J. Anderson, Mark W. Spong, "Bilateral Control of Teleoperators with Time Delay", IEEE Transaction on Automatic Control, Vol.34, No.5, pp.494-501, may 1989
- [3] <http://www.brad.a.uk/acad/indtech>
- [4] <http://www.usc.edu/dept/raiders>
- [5] <http://telerobot.mechumwa.edu.ac.kr>
- [6] <http://vive.cs.berkeley.edu/capek/>
- [7] W. R. Ferrell, "Remote Manipulation with Transmission Delay", IEEE Trans Human Factors in Electronics HFE-6, no.1 1965
- [8] Bolot, J. 1993(Ithaca,NY). End-to-End Packet Delay and Loss Behavior in the Internet. In SIGCOMM '93. New York: ACM, pp.289-298.



## Remote Control of a Robot Through the Internet

Dong-Ju, Lee\*/Tae-Kyu, Yang\*\*

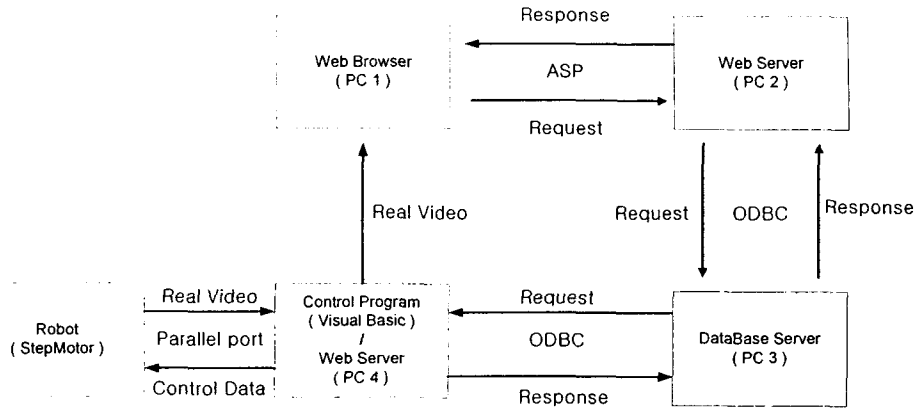
### Abstract

In this paper, We control remote robot using web browser. This system used independent four PC and each PC is used to operator PC, database server, web server and robot control PC. Remote control can escape the operator from constraint the time and workspace. If operator inputting control data in web browse then the control data is saved the data base server. The robot control PC is inputted control data from database at access intervals of 1ms. Also, if the control data is changed then robot is operated. It made simple robot using step-motor and the robot is operated by web browser. Finally, Robot system is exactly operated using control data and certificate of possibility the remote robot control using web browser.

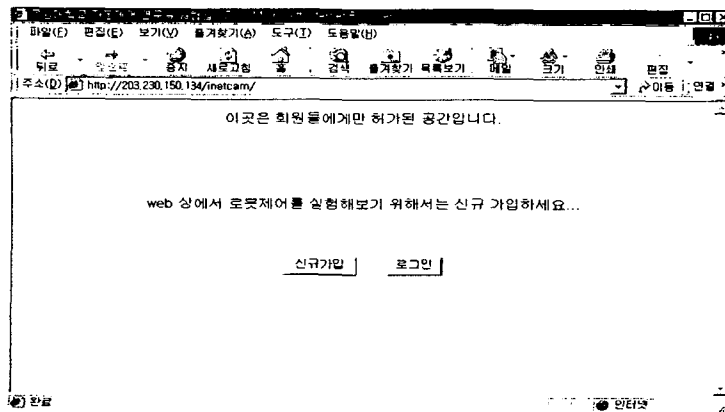
---

\* Movies Co. Led.

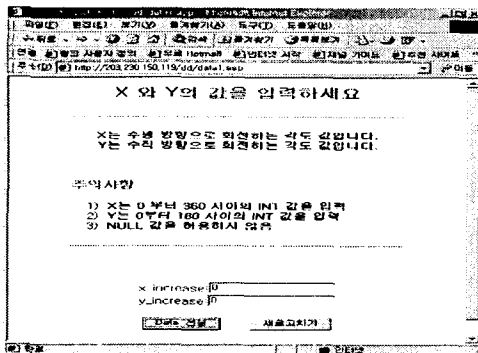
\*\* Dept. of Electronics and Information Communication Eng., Mokwon University.



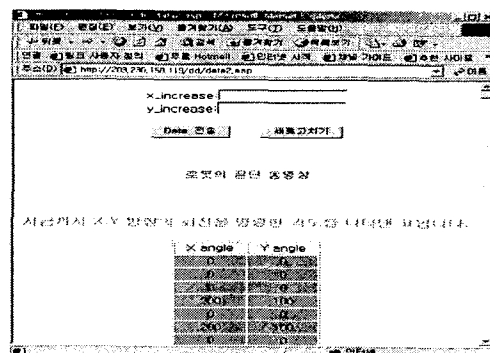
(그림 1) 데이터 연결 블록선도



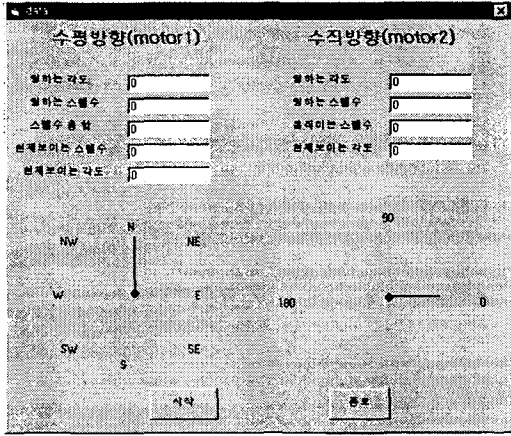
(그림 2). 웹브라우저 login



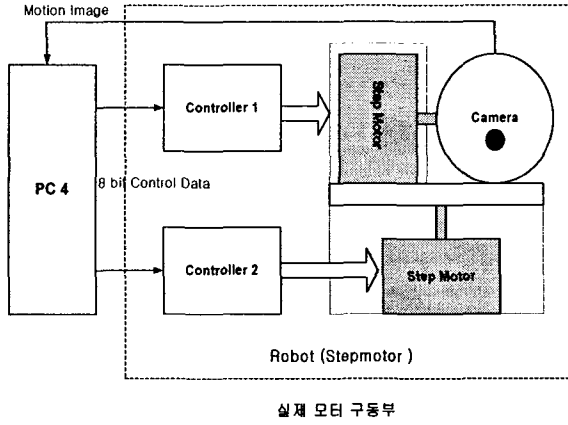
(그림 3) 컨트롤 데이터 입력



(그림 4) 데이터베이스 접속 후 웹브라우저



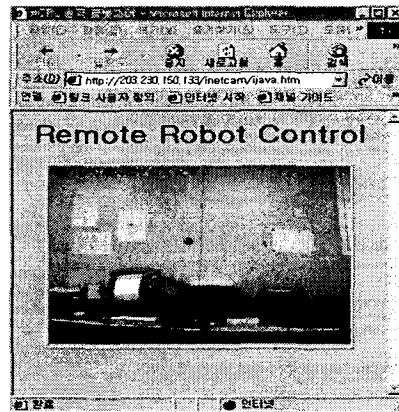
(그림 5) 컨트롤 프로그램



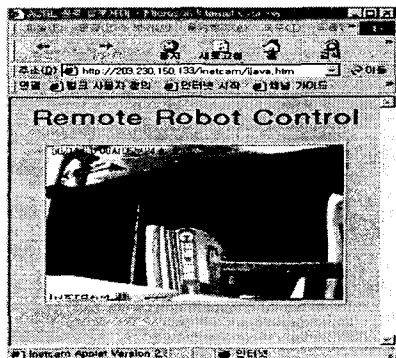
(그림 6) 하드웨어 블록선도

y_increase	x_angle	y_angle	Sort descending
100	0	0	0
0	0	0	0
100	0	0	0
320	160	0	0
100	180	0	0
50	0	0	0
45	120	0	0
100	0	0	0
62	50	0	0
100	30	0	0
0	0	0	0
200	100		

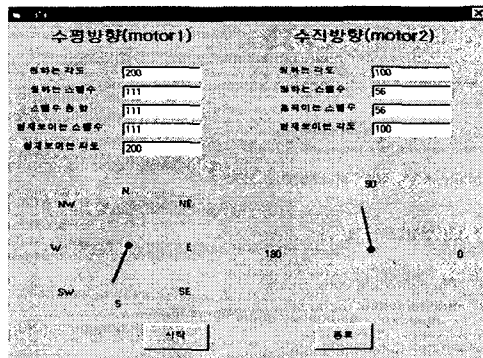
(그림 7) MS-SQL 서버



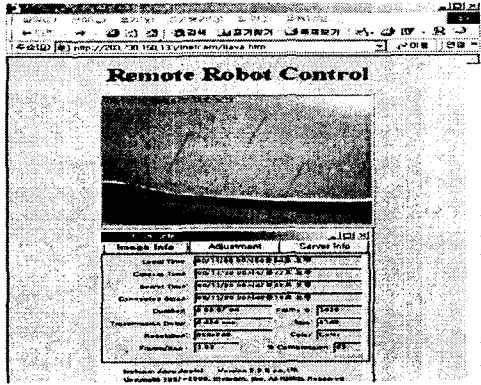
(그림 8) 동작전 동영상



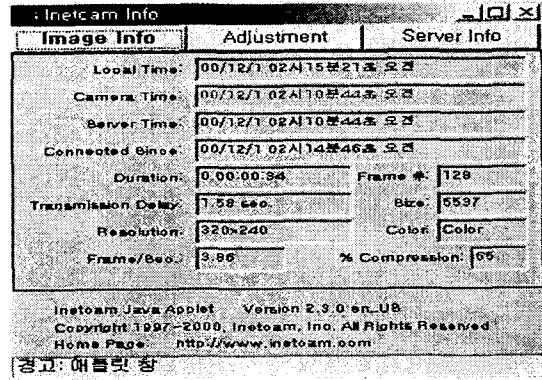
(그림 9) 동작후의 동영상



(그림 10) 컨트롤 프로그램



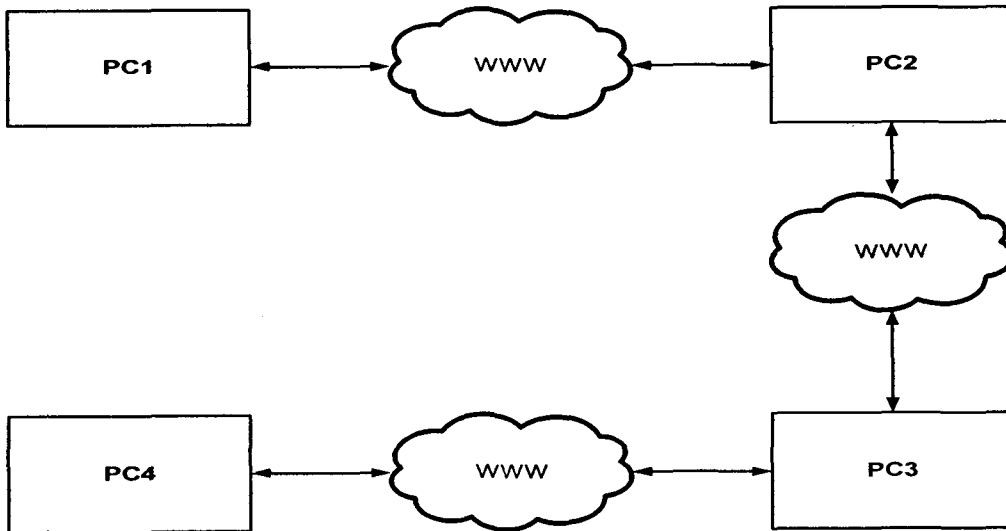
(그림 11) 로봇의 동영상



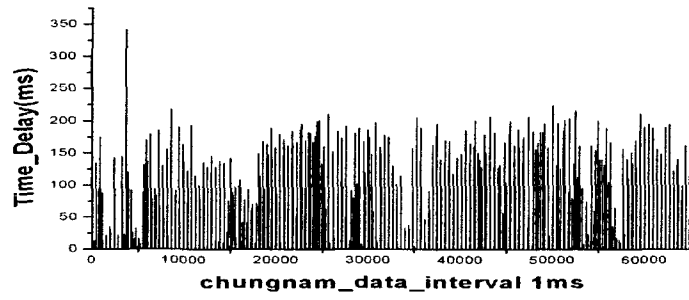
(그림 12) 조작자와 로봇간의 시간 지연



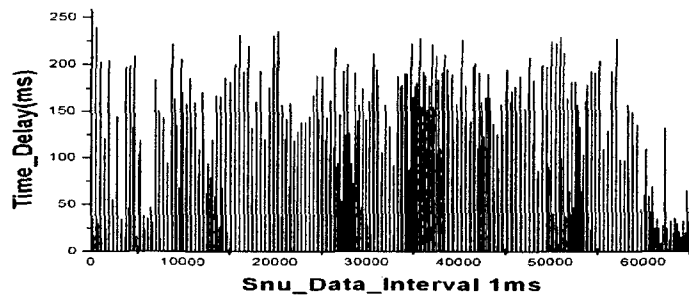
(그림 13) 소켓 프로그램 블록선도



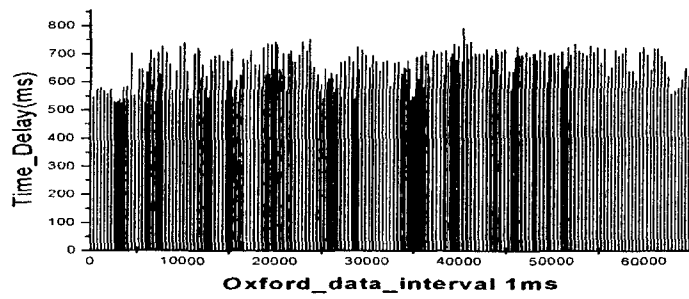
(그림 14) 제안된 블록 선도



(그림 15) 충남대학교 핑테스트



(그림 16) 서울대학교 핑테스트



(그림 17) 옥스퍼드대 핑테스트



(사진 1) 제작 로봇