

직접명령 방식 인터넷 주행로봇 시스템의 거리 오차 보상

Distance Error Compensation of Direct Control Type Internet-based Mobile Robot System

이 강 희, 김 수 현, 곽 윤 근*
(Kang-Hee Lee, Soo-Hyun Kim, and Yoon-Keun Kwak)

Abstract : This research is concerned with the development of an Internet-based robot system, which is insensitive to the unpredictable internet time delay. For this purpose, a simple mobile robot system that moves in response to the user's direct control on the internet has been developed. The time delay in data transmission is an important problem for the construction of this kind of system. Therefore, the PPS (Position Prediction Simulator) is suggested and implemented to compensate for the time delay problem of the internet. The simulation and experimental results show that the distance error can be reduced using the developed PPS.

Keywords : internet-based robot, direct control, PPS, time delay, VRML, EAI

1. 서론

통신 기술의 발전과 통신망의 확장으로, 일반인들도 어디서나 쉽게 인터넷에 접속할 수 있는 환경이 구축되고 있다. 국내의 경우, 2002년 6월 기준으로 1Mbps급 이상의 초고속 인터넷 가입자 수가 전체 가구의 64%에 해당하는 921만 가구에 이르며, 인터넷 이용자 수 또한 전체 인구의 58%인 2,565만 명이 집계되고 있다[1]. 인터넷의 급속한 발전과 함께 기존의 웹 사이트를 통한 정보 검색이나 전자메일 교환, 파일 전송 외에도 인터넷 자원을 이용한 새로운 응용 방안들이 등장하고 있다. 이미 가진 기기들을 하나로 연결하는 홈네트워크링이나 원격 방법 시스템 등과 같이 기존의 오프라인 장비들을 인터넷을 통하여 통합시키려는 노력들이 실용화되고 있으며, 이러한 응용 방안들 중의 하나로서 인터넷을 통하여 원격에 위치한 로봇을 조종하는 인터넷 로봇(Internet-based robot) 분야에 대한 연구 또한 활발히 이루어지고 있다.

기존의 원격조종(teleoperation) 로봇 시스템에서는 전문가들이 전용 통신회선과 전용 인터페이스를 이용하여 근거리에서 위치한 로봇을 제어해 왔던 것에 비해, 인터넷 로봇은 인터넷이라는 공용의 통신망을 이용하며 비전문가들도 단말기나 운영 체제의 종류와 상관없이 어디서나 쉽게 접속할 수 있는 환경의 제공을 목적으로 한다. 이러한 인터넷 로봇은 통신망의 경제성, 시스템의 유연성 및 개방성이라는 장점과 함께 데이터 전송 시의 불확실성 및 시간 지연, 보안 문제와 같은 단점을 동시에 가지고 있다. 따라서 기존의 원격조종 로봇이 우주, 해저, 원자력 발전소, 재난지역 등의 탐사와 같은 전문적인 목적으로 사용되었던 것에 반해, 인터넷 로봇은 아직까지는 완구, 오락, 방법, 안내 등과 같이 대중을 상대로 한 분야에 제한되고 있으나 그 활용분야가 점차 넓어지고 있는 추세이다.

최초의 인터넷 로봇은 1994년 머큐리 프로젝트를 통해 개발된 원격 발굴 시스템이었다[2]. 같은 해에 호주에서는 6축 로봇을 움직여 장난감 블록으로 구조물을 쌓는 시스템이 개발되었다[3]. 인터넷 주행로봇 시스템에 대한 연구도 이루어졌는데, Xavier 로봇은 원격에서 지정한 건물내의 특정 위치로 자동 주행하여 임무를 수행하도록 개발되었으며[4], Rhino 로봇은 원격의 사용자가 지정한 위치로 이동하여 전시물의 감상을 도와주는 박물관 안내 로봇으로 개발되었다[5]. 초기의 인터넷 로봇은 원격에서 보내주는 실제 영상 정보를 바탕으로 사용자가 과제를 입력하면 이에 따른 로봇의 이동 명령들이 결정된 후, 순차적으로 실행되는 간접명령(indirect control) 방식이었다. 간접명령을 통하여 주행로봇을 조종할 경우엔 경로 계획, 충돌 회피 등은 로봇의 자율성만으로 해결해야 한다는 단점이 있다.

로봇의 세부적인 이동을 사용자가 조종하는 직접명령(direct control) 방식을 이용한 인터넷 로봇으로는, 소형 주행 로봇의 속도와 방향을 조종하여 미로를 찾아 가는 시스템[6]과 영상 위의 목적 지점을 지정하여 로봇을 주행 시키는 방식의 미술관 안내 로봇 시스템[7]이 개발되었다. 이 시스템들은 사용자에게 로봇의 위치나 상태 정보를 제공하는 원격지의 실제 영상이 인터넷을 통하여 전달될 때 시간 지연되는 문제 때문에 제한된 주행만이 가능하다는 단점을 가지고 있었다.

불확실성을 가진 인터넷을 통한 데이터 전송시의 시간 지연을 보상하기 위한 방안으로 먼저 로봇과 주변 환경에 대하여 특정 시간 후의 위치를 예측하고 이를 원격의 사용자에게 제공하는 연구가 이루어졌는데, 경사면을 굴러 내려오는 공을 받는 로봇 팔을 원격에서 조종하기 위해 이전 영상으로부터 공의 도착 위치를 미리 계산하여 예측하는 시스템[8]과, 로봇 팔로 트랙을 돌고 있는 물체를 잡기 위하여 움직이는 물체의 예측 위치를 보여주는 시스템[9]이 개발되었으며, 주행 로봇의 위치 정보 손실 및 지연에 대해서 보상하도록 위치를 예측하여 3차원으로 보여주는 시스템[10]도 개발되었다. 이 시스템들은 로봇의 이동 경로 오차는 줄일 수 있었으나,

* 책임저자(Corresponding Author)

논문접수 : 2003. 4. 10., 채택확정 : 2003. 11. 30

이강희 : 한국과학기술원 기계공학과 대학원(cbill@kaist.ac.kr)

김수현, 곽윤근 : 한국과학기술원 기계공학과

(kimsh@kaist.ac.kr/kyk@kaist.ac.kr)

※ 본 연구는 2003년 BK21사업의 지원으로 수행되었음.

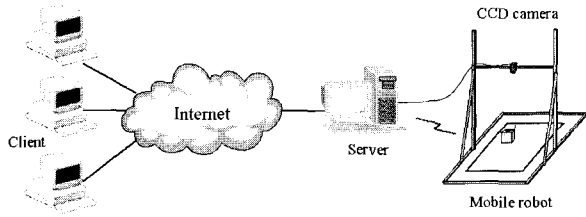


그림 1. 인터넷 로봇 시스템 구성도.
Fig. 1. Configuration of the Internet-based robot system.

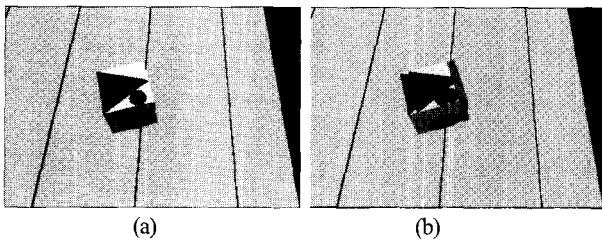


그림 2. 위치 예측 시뮬레이터. (a) 3차원 가상 화면, (b) 예측 위치를 표시한 영상.
Fig. 2. PPS. (a) 3D virtual scene, (b) Position prediction scene.

명령어가 로봇에 도착했을 시점의 로봇 위치에 대한 정보를 사용자가 알 수 없기 때문에 특정 지점에서의 정지와 같은 순간적인 직접명령은 불가능 하였다.

국내에서도 인터넷 로봇에 관한 연구가 활발히 이루어지고 있다. 웹 기반의 3차원 가상 시뮬레이터를 이용한 연구로서 로봇 시뮬레이터[11]와 실시간 모니터링 시스템[12]이 개발 되었으며, 인터넷을 통하여 원격에서 운영하는 이동로봇 시스템에 대한 연구[13]가 이루어진 바 있다.

본 연구의 목적은 로봇 정보, 이동 명령 같은 데이터의 전송 시 인터넷에서 발생하는 시간 지연 문제에 강한 인터넷 로봇 시스템의 개발이다. 시간 지연을 보상하기 위한 새로운 방안으로 제안된 위치 예측 시뮬레이터(Position Prediction Simulator, PPS)를 인터넷 로봇 시스템에 적용하여, 원격에 위치한 사용자가 직접명령 방식으로 주행로봇을 조종할 때 정밀한 위치 제어가 가능함을 시뮬레이션 및 로봇 주행실험을 통하여 검증하고자 한다.

II. 위치 예측 시뮬레이터

개발된 인터넷 로봇 시스템은 그림 1과 같이 로봇, 서버, 클라이언트부로 구성되며 원격지의 서버와 클라이언트는 인터넷을 통하여 서로 연결된다. 인터넷은 특정 회사나 개인이 소유한 네트워크가 아니며, 현재 쓰이고 있는 IP 주소를 이용한 통신 방식은 서비스 품질을 보장하지 않는 방식이기 때문에 데이터 전송 시의 시간 지연이 인터넷 로봇 연구에서는 가장 큰 문제점으로 알려져 있다.

본 연구에서는 시간 지연을 감안한 로봇의 위치를 추정하여 표시하는 위치 예측 시뮬레이터를 개발하여 이 문제점을 해결하고자 하였다. 개발한 위치 예측 시뮬레이터는 3차원 가상 화면부와 위치 예측부로 구성되며, 사용자에게 보여지는 화면은 그림 2와 같다. 그림 2(a)는 원격지의 주행로봇을

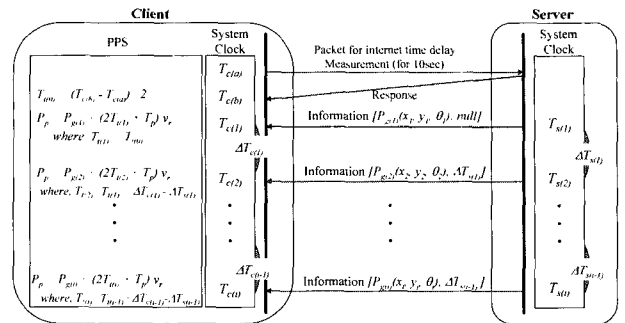


그림 3. 단계별 위치 예측 방법.
Fig. 3. Schematic diagram of position prediction.

표시하는 3차원 가상 화면이며, 그림 2(b)는 위치 예측부에서 사용자의 명령이 로봇 측에 도달했을 때를 시점으로 로봇의 위치를 추정하여 3차원 가상 화면 위에 반투명으로 첨가하여 표시한 모습이다.

로봇의 예측 위치는 인터넷에서의 전송 지연과 시스템 자체에서 소요되는 처리 시간을 고려하여 다음 식으로부터 계산하였다.

$$P_p = P_{g(i)} + (2T_{i(i)} + T_p)v_r \tag{1}$$

$$T_{i(i)} = T_{i(i-1)} + \Delta T_{c(i-1)} - \Delta T_{s(i-1)} \tag{2}$$

P_p 는 예측된 로봇의 위치이며, $P_{g(i)}$ 는 서버에서 주어진 i 번째의 로봇 위치이다. P_p 는 서버와 클라이언트 간의 왕복 인터넷 전송 시간 $2T_{i(i)}$ 와 기 측정된 시스템 처리시간 T_p 및 주행 로봇의 속도 v_r 로부터 계산된다. $T_{i(i)}$ 는 (2)에서와 같이 이전 단계의 시간 지연 $T_{i(i-1)}$ 과 클라이언트 및 서버에서의 i 번째와 $(i-1)$ 번째의 시스템 시간 차이 값인 $\Delta T_{c(i-1)}$, $\Delta T_{s(i-1)}$ 로부터 계산 된다. 예측한 위치를 계산하기 위한 서버와 클라이언트에서의 흐름도는 그림 3과 같다.

초기의 인터넷 전송시간 $T_{i(0)}$ 는 클라이언트에서 전송한 특정 문자열의 왕복 시간으로부터 측정한다. 이와 같은 실제의 측정이 반복될 경우엔 인터넷의 과부하로 작용할 수 있으므로 측정은 10sec에 한번만 수행하고 그 사이의 전송시간 변화는 예측하였다. 이 때 서버와 클라이언트 간의 시간 동기를 맞추는 것이 어렵기 때문에 각 시스템에서의 시간 차이를 비교하였고, 이를 위하여 서버에서 클라이언트로 로봇의 위치 정보와 함께 서버 시스템에서의 각 단계별 시간 차이 값을 같이 보내주었다.

III. 인터넷 로봇 시스템

1. 하드웨어

이미 선행연구를 통하여, 인터넷을 통하여 접속한 사용자가 주어진 OX 퀴즈의 정답 영역으로 원격에 위치한 로봇을 이동 시키는 인터넷 로봇 시스템을 개발한 바 있다[14]. 본 연구에서는 OX 퀴즈 로봇 시스템의 하드웨어를 일부만 교체하여 다시 사용하였다. 주행 로봇은 그림 4(a)의 상용 축구 로봇을 사용 하였으며, CCD 카메라는 그림 4(b)와 같이 Sony사의 프로그래시브 방식 흑백 카메라인 XC-55를 사용하였다.

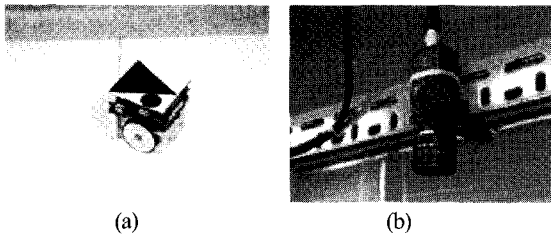


그림 4. 하드웨어 구성. (a) 주행로봇, (b) CCD 카메라.
Fig. 4. Hardware environment. (a) Mobile robot, (b) CCD camera.

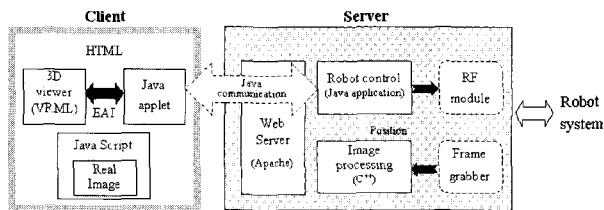


그림 5. 소프트웨어 구성도.
Fig. 5. Configuration of software.

카메라는 초점거리 8mm 렌즈를 장착하였으며, 별도의 팬-틸트(pan-tilt) 조정 기능 없이 경기장 중앙 상부에 고정 설치하였다.

사용된 서버는 1.8GHz 중앙처리장치, 512Mbytes 메모리, Windows 2000 운영 시스템을 탑재한 개인용 컴퓨터이며 랜 카드를 이용하여 인터넷과 연결하였다. 장착된 프레임 그래픽 보드는 Matrox사의 Meteor-II 다채널 컬러 포획 방식으로 CCD 카메라에서 받은 영상 신호를 처리 한다. 시리얼 포트에는 로봇과의 데이터 전송을 위한 무선 주파수(radio frequency) 송수신 장치를 연결하였다. 주행 경기장은 640x480mm 규격으로 제작하였으며, 편리한 위치 측정을 위하여 상면에 모눈종이를 부착하였다.

2. 소프트웨어

시스템의 서버와 클라이언트에서 구동 되는 각각의 소프트웨어는 그림 5와 같이 구성하였다.

서버에서 구동 되는 프로그램은 웹 서버, 로봇 제어, 영상 처리의 3개 부분으로 구성된다. 웹 서버는 아파치(Apache)로 구축하였다. 로봇 제어 프로그램은 자바 어플리케이션(Java application)으로 구현하였으며 클라이언트 측에서 구동 되는 자바 애플릿(applet)과의 통신을 통해 사용자가 입력한 로봇의 이동 및 정지 명령을 전송 받아 지정된 형식으로 변환시켜 로봇부로 전달하는 동시에 영상 처리부에서 계산된 로봇의 위치 정보를 클라이언트 측에 전송하는 작업을 수행한다.

클라이언트부의 사용자 인터페이스는 그림 6과 같이 좌측의 실제 영상부와 우측의 로봇 조종부 및 3차원 가상 화면부로 구성된다. 실제 영상은 1초에 한번씩 서버에 저장된 영상을 불러와 보여준다. 자바 애플릿으로 구현된 로봇 조종부는 사용자의 입력에 따른 로봇 이동 및 정지 명령을 서버 측의 자바 어플리케이션 프로그램에 전달하는 동시에 서버 측으로부터 받은 로봇의 위치 정보를 전송 받아서 3차원 가상 화면부에 넘겨주는 역할도 담당한다. 3차원 가상 화면부는 서버 측으로부터 별도의 프로그램을 받을 필요가 없이 웹 페이지에

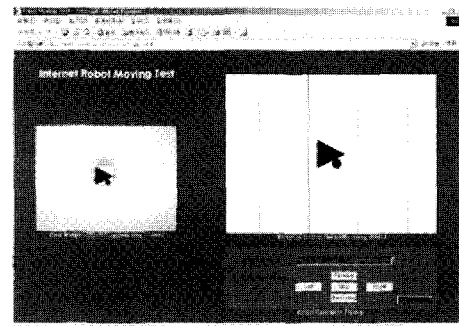


그림 6. 로봇 조종용 사용자 인터페이스.
Fig. 6. User interface of the Internet robot.

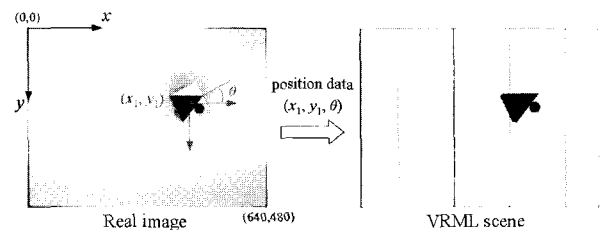


그림 7. 영상처리에서 구한 로봇 위치 정보를 이용한 3차원 가상 화면.
Fig. 7. Relation of robot position between real image and 3D virtual scene.

서 표현이 가능한 VRML(Virtual Reality Modeling Language)로 구현하였다. 3차원 가상 화면을 이용할 경우, 로봇과 주변 환경의 모습은 미리 정의된 데이터에 의해 표현되며, 로봇이 이동하는 경우에는 변화된 위치 정보만을 전송 받으므로 데이터 전송시간의 감소로 빠른 정보 전달이 가능하다. 또한 축소, 확대, 이동, 회전과 같은 다양한 시점 조절이 가능하다는 장점이 있다. 본 연구에서는 로봇 조종부를 구성한 자바와 VRML간의 통신을 위하여 EAI(External Authoring Interface) 방식을 사용하였다.

3. 영상 처리부

C++ 언어로 구현한 영상 처리부는 미리 저장되어 있는 로봇 샘플과의 비교 작업을 통하여 로봇의 현재 위치를 구한다. 위치 정보는 그림 7과 같이 x, y축 좌표 및 회전 각도 θ 로 구성되며, 포획한 영상이 640x480픽셀 크기이므로 1픽셀당 1mm의 해상도를 갖는다. 또한 영상을 JPEG(Joint Photographic Experts Group) 형식으로 압축 변환시켜 저장하여, 클라이언트 부에서 실제 영상을 보여줄 때 사용한다.

고정된 카메라를 이용하여 획득한 영상에서 로봇의 위치를 인식하는 경우엔 그림 8(a)에서와 같이 로봇 자체의 높이로 인한 오차가 발생한다. 실제로, 그림 8(b)에서 구한 로봇 위치와 그림 8(c)에서 구한 로봇 상면과 같은 모양의 종이를 놓고 구한 실제 위치는 x, y 값 모두 차이가 있음을 보인다. 이 오차는 로봇이 카메라가 설치된 중앙에서 멀리 떨어질수록 커진다.

로봇의 위치 오차를 줄이기 위하여 그림 8(a)에서의 기하학적 관계를 이용하여 다음과 같은 간단한 위치 보정 식을 구하였다.

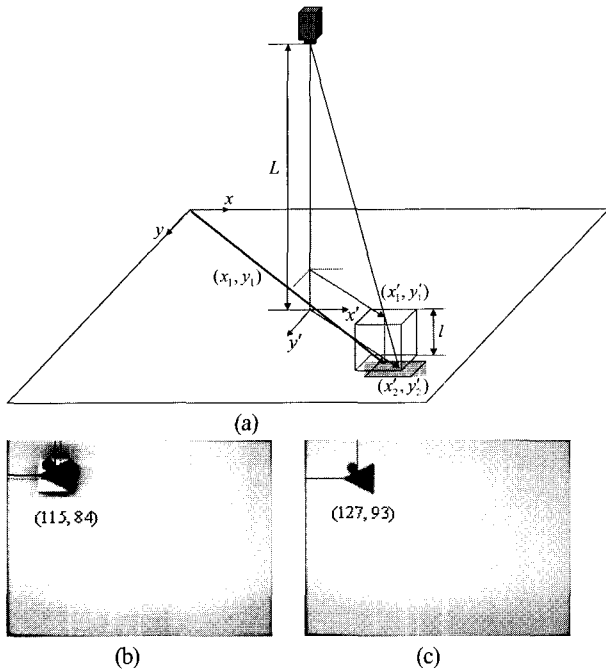


그림 8. 로봇 위치 보정. (a) 좌표계, (b) 로봇 영상, (c) 실제 로봇 위치.

Fig. 8. Calibration of robot position. (a) The coordinate in the calibration, (b) Robot image, (c) Image of real position.

표 1. 로봇 위치 보정 결과.

Table 1. Camera and robot position calibration result.

Axis	Mean value of error (mm)		Maximum error (mm)	
	Measure d	Estimate d	Measure d	Estimate d
x	10.4	1.0	17.5	2.8
y	8.9	2.1	15.5	4.4

$$x_1 = x'_1 + 320 = \frac{(L-l)}{L} x'_2 + 320 \quad (3)$$

$$y_1 = y'_1 + 240 = \frac{(L-l)}{L} y'_2 + 240 \quad (4)$$

여기에서 x_1, y_1 은 x, y 좌표계에서의 구하고자 하는 실제 로봇 위치이며, 카메라 설치 위치를 원점으로 하는 x', y' 좌표계에서 x'_1, y'_1 는 영상 처리를 통해 구한 로봇 위치, x'_2, y'_2 는 보정 후의 로봇 위치이다. L 은 주행 경기장에서 카메라까지의 높이이고, l 은 로봇 자체의 높이이며 본 시스템에서는 각각 1,085mm와 76mm이다. 로봇 위치의 오차 보정이 정확히 이루어졌는지 알아보기 위하여 경기장의 16개 지점에서 영상 처리로 구한 로봇의 위치와 보정 후의 위치에 대하여 실제 위치와 오차를 비교한 결과, 표 1에서와 같이 평균 오차와 최대 오차 값이 감소하였다.

IV. 로봇 주행 시뮬레이션

1. 주행 시뮬레이션 조건

본 연구에서는 직접명령, 간접명령, 위치 예측 시뮬레이터를 이용한 직접명령의 3가지 방식에 대하여 시뮬레이션을

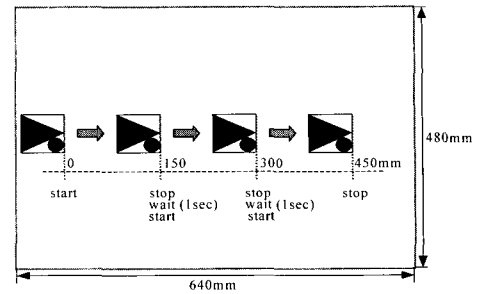


그림 9. 로봇 주행 시뮬레이션 조건.

Fig. 9. Condition of robot moving simulation

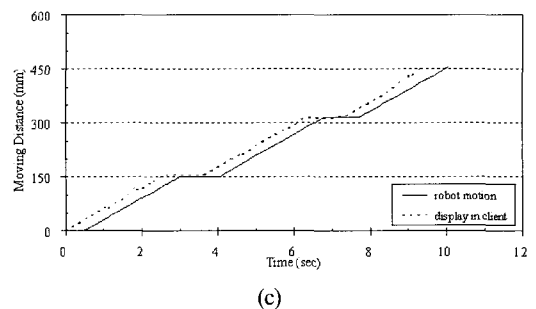
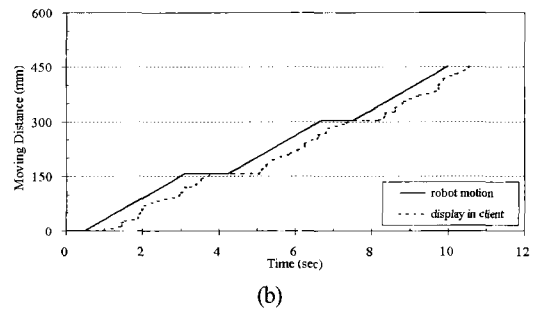
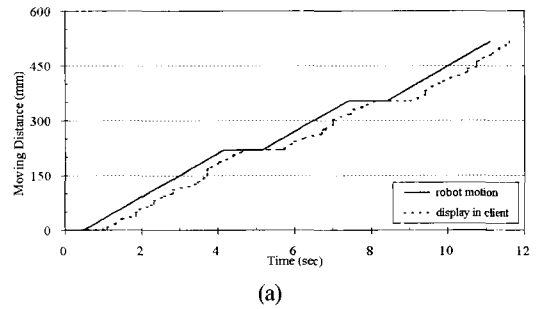


그림 10. 시뮬레이션 결과. (a) 직접명령, (b) 간접명령, (c) 직접명령 + 위치 예측 시뮬레이터.

Fig. 10. Simulation results. (a) Direct control, (b) Indirect control, (c) Direct control + PPS.

통한 주행로봇의 위치 조종성을 비교해 보았다. 직접명령은 진행, 정지, 방향전환과 같이 하위레벨의 작동을 지시하는 명령이며, 간접명령은 특정 위치로의 주행, 임무의 수행 같은 상위레벨의 과제를 지시하는 명령이다. 간접명령의 경우엔 사용자의 명령이 미리 정해진 이동 계획 등에 따라 단계별로 세분화 된 후, 순차적으로 로봇측에 전달 된다.

시뮬레이션을 위한 로봇의 주행 조건은 그림 9와 같이 로봇이 출발하여 150mm 진행 후 정지하는 것을 3회 반복하도록

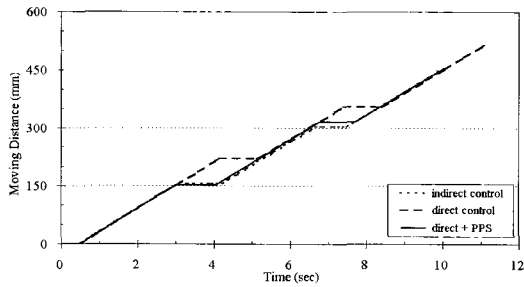


그림 11. 각 명령방식 별 로봇 위치 비교.
Fig. 11. Robot motion in simulation result.

하였다. 로봇의 속도는 60mm/sec로 가정하였으며 감속 및 가속 구간은 고려하지 않았다. 인터넷의 전송 시간 지연은 100msec에 한번씩 변하는 최대 1sec 이하의 임의(random) 값을 발생시켰으며, 시스템의 처리 시간은 100msec의 고정 값을 사용하였다. 3가지 방식 모두 원격지의 로봇 정보는 실제 영상이 아닌 3차원 가상 화면에서 얻는 것으로 가정하였다.

2. 시뮬레이션 결과

로봇 주행 조건에 따라 실시한 시뮬레이션 결과는 그림 10과 같다. 실선은 로봇의 주행 위치를 나타내며, 점선은 클라이언트부에서 사용자에게 보여지는 위치를 나타낸다. 위치 예측 시뮬레이터를 적용한 직접명령의 경우, 사용자는 실제 로봇의 위치보다 앞서서 진행하는 로봇의 위치를 보게 된다. 직접명령을 이용한 주행에서는 그림 10(a)과 같이 로봇이 예정 위치보다 평균 63.4mm 더 진행하여 정지하였다. 로봇이 정지할 위치를 미리 정해 놓은 상태에서, 주기적으로 명령을 보내게 한 간접명령의 결과는 그림 10(b)와 같으며, 이 경우 로봇의 정지 거리 오차는 평균 3.6mm로 매우 작다. 위치 예측 시뮬레이터를 적용한 직접명령의 결과는 그림 10(c)와 같으며, 로봇의 정지 거리 오차는 평균 7.7mm이다.

각 방식에서 로봇의 위치만을 비교한 결과는 그림 11과 같다. 발생시킨 인터넷 전송시간 지연 값에 따라 조금씩 차이가 있으나, 전반적으로 직접명령의 경우 시간 지연에 따른 영향이 크고, 간접명령 방식과 위치 예측 시뮬레이터를 적용한 직접명령의 경우엔 영향이 적었다. 이와 같이 제한된 위치 예측 시뮬레이터를 직접명령에 접목할 경우 간접명령만큼의 시간 지연에 대한 강인성을 가지면서, 직접명령 방식의 장점인 장애물 발생시의 대처 등을 직접 해결할 수 있는 특성을 보임을 알 수 있다.

V. 로봇 주행 실험

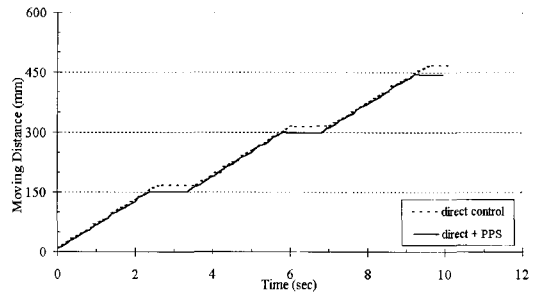
1. 시스템 처리 시간 측정

로봇의 위치 예측을 위해서는 서버와 클라이언트 간의 인터넷 전송 시간과 함께 시스템의 처리 시간을 정확히 알아야 한다. 본 연구에서는 영상처리와 같은 소프트웨어 소요 시간 뿐 아니라, 로봇과 서버간의 무선 주파수 통신 시간, 로봇의 반응 시간 및 로봇 정지시의 미끄러짐 등을 모두 포함하여 시스템 처리 시간을 정의하였다. 이의 측정을 위하여 클라이언트와 서버가 동일한 컴퓨터에서 구동되는 인터넷 트래픽이 없는 환경에서의 로봇 정지 실험을 실시한 후, 이때의 정지 거리 오차를 측정하였다. 실험 결과는 표 2와 같다.

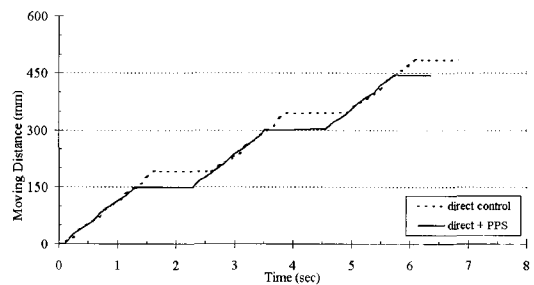
표 2. 시스템 처리 시간.

Table 2. System process time.

Robot speed (mm/sec)	Distance error (mm)		Process time (msec)
	without PPS	with PPS	
60	7.2	1.1	120
120	16.9	2.1	141



(a)



(b)

그림 12. 로봇 주행 실험 결과 (a) 로봇 속도 : 60mm/sec, (b) 로봇 속도 : 120mm/sec.

Fig. 12. Experimental results. (a) Robot speed : 60mm/sec, (b) Robot speed : 120mm/sec.

먼저 위치 예측 시뮬레이터를 적용하기 않은 상태에서 로봇의 주행 속도를 60, 120mm/sec로 변화시키면서 15회 실시한 정지 실험에서의 평균 거리 오차를 구한 후, 이 거리 오차를 로봇의 속도로 나누어 각 경우에 대하여 120, 141msec의 시스템 처리 시간을 구하였다. 측정된 시스템 처리 시간의 유용성을 확인하기 위하여 이를 위치 예측 시뮬레이터에 적용하여 같은 조건에서 로봇 주행 실험을 실시한 결과, 정지 거리 오차는 1~2mm로 감소하였다.

2. 고정된 전송시간 지연에서의 로봇 주행

인터넷에서의 고정된 시간 지연 효과를 얻기 위하여, 로봇 서버 프로그램에 100msec의 시간지연부를 첨가하였다. 로봇의 주행 조건은 그림 9의 시뮬레이션 조건과 동일하게 150, 300, 450mm 지점에서 정지 후, 1초간 쉬었다가 출발하게 하였으며, 클라이언트 프로그램에서 자동으로 판단하여 정지 및 출발 명령을 보내도록 하였다. 로봇의 주행 결과는 그림 12와 같다. 그림 12(a)는 로봇의 속도가 60mm/sec, 그림 12(b)는 120mm/sec일 때의 결과이다. 이 결과는 서버 프로그램에서 각 시간에 따른 로봇의 위치를 데이터 파일로 저장하여 구하였다. 그림에서 실선은 위치 예측 시뮬레이터를 사용한 결과이며, 점선은 사용하지 않은 결과로서, 두 조건에서 모두,

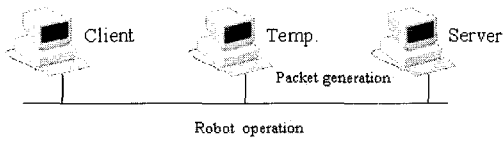


그림 13. 로봇 조종 실험 조건.

Fig. 13. Robot operation test condition.

표 3. 주행 실험 거리 오차 결과.

Table 3. Experimental results in traffic generation condition.

Traffic Generation (%)	Distance error (mm)	
	Direct control	Direct +PPS
0	16.2	5.4
25	24.6	9.6
50	24.2	10.2
75	32.0	10.8

위치 예측 시뮬레이터를 사용할 경우 원하는 지점에 정확히 로봇을 정지 시킬 수 있음을 보인다.

3. 변화하는 전송시간 지연에서의 로봇 주행

실제의 인터넷 환경처럼 트래픽 부하를 바꿔가면서 실험을 하기 위하여 그림 13과 같이 클라이언트, 서버, 임시 컴퓨터를 동일 랜(LAN)에 연결한 후, 서버 측에서 스니퍼(Sniffer pro 4.6) 프로그램을 이용하여 임시 컴퓨터로 인위적인 트래픽을 발생시켰다.

먼저 임시 컴퓨터에서 서버 측으로 1,472 bytes 크기의 핑(ping) 명령을 보낸 후 이 패킷(packet)을 서버측의 스니퍼에서 포획하고 이를 복사하여 지속적으로 보내는 방법으로 트래픽을 인위적으로 발생시켰다. 이 때 발생하는 패킷의 간격을 조절하는 방법으로 임시 컴퓨터와 서버간의 회선점유율을 변화시켰다.

위치 예측 시뮬레이터의 효율성 여부를 알아보기 위하여, 트래픽을 변화시켜 가면서 60mm/s로 주행하는 로봇을 사용자가 정지 명령을 입력하여 지정된 선에 맞추어 정지 시키는 실험을 실시하였다. 트래픽을 발생시켜 서버와 임시 컴퓨터 간의 회선 점유율을 0, 25, 50, 75%로 변화시켜 가면서 위치 예측 시뮬레이터를 적용하기 전과 후에 로봇이 정지한 위치의 거리 오차 값을 측정 한 결과는 표 3과 같다.

각 값들은 5회 측정한 거리 오차의 절대값을 평균한 결과이다. 위치 예측 시뮬레이터를 적용하지 않은 경우엔 회선 점유율을 증가시킴에 따라 거리 오차도 크게 증가함을 알 수 있다. 위치 예측 시뮬레이터를 적용한 경우에는, 트래픽이 증가해도 거리 오차의 증가는 적었다. 실험 결과처럼 위치 예측 시뮬레이터를 적용할 경우 인터넷 트래픽이 증가하여도 주행로봇의 조종 오차를 줄이는 효과가 있음을 알 수 있다.

VI. 결론

본 연구에서는 로봇의 실제 위치를 예상하여 사용자에게 보여 주는 위치 예측 시뮬레이터(PPS)를 개발하여 인터넷에서의 데이터 전송 시에 발생하는 시간 지연 문제를 보상하였

다. 시뮬레이션 결과, 위치 예측 시뮬레이터를 적용한 직접명령의 경우, 간접명령 수준의 정밀한 로봇 조종이 가능하면서 장애물 발생시의 긴급 정지와 같은 직접명령의 장점을 가지고 있음을 알 수 있었다. 또한 실제 로봇의 주행 및 정지 실험을 통한 거리 오차 측정 결과, 개발된 위치 예측 시뮬레이터를 사용한 인터넷 로봇 시스템은 인터넷 트래픽의 변화에 강한 특성을 보임을 증명하였다.

향후 연구과제로는 개발된 위치 예측 시뮬레이터로 보상이 가능한 범위를 찾고 그 이상의 인터넷 전송시간 지연에 대응하기 위한 로봇의 모션 예측 기능이 추가되어야 하며, 가변 속도를 가진 로봇의 주행에 대한 요소가 위치 예측 시뮬레이터부에 반영되어야 한다.

참고문헌

- [1] Korea Network Information Center, "A survey on the number of Internet users and Internet behavior," <http://www.nic.or.kr/>, July, 2002.
- [2] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter and J. Wiegley, "Desktop teleoperation via the World Wide Web," *In Proceedings of IEEE Int. Conf. on Robotics and Automation*, pp. 654-659, 1995.
- [3] K. Taylor, B. Dalton and J. Trevelyan, "Web-based telerobotics," *Robotica*, vol. 17, pp. 49-57, 1999.
- [4] R. Simmons, "Xavier: An autonomous mobile robot on the web," *In Proceedings of IEEE/RSJ IROS Workshop on Robots on the Web*, pp. 43-48, 1998.
- [5] D. Schulz, W. Burgard, D. Fox, S. Thrun and A. B. Cremers, "Web interfaces for mobile robots in public places," *IEEE Robotics & Automation Magazine*, vol. 7, no. 1, pp. 48-56, March, 2000.
- [6] P. Saucy and F. Mondada, "KhepOnTheWeb: one year of access to a mobile robot on the Internet," *In Proceedings of IEEE/RSJ IROS Workshop on Robots on the Web*, pp. 23-30, 1998.
- [7] S. Maeyama, S. Yuta and A. Harada, "Experiment on a remote appreciation robot in an art museum," *In Proceedings of IEEE/RSJ Intl. Conf. IROS*, pp. 1008-1013, 2000.
- [8] K. Tanaka, E. Nakagawa, M. Ito, N. Mizuno, T. Yamada, E. Shimizu and K. Kagayama, "An Internet-based tele-robot environment for a time critical task," *Systems, Man, and Cybernetics, IEEE SMC '99 Conference*, vol. 5, pp. 1106-1110, 1999.
- [9] J. Kikuchi, K. Takeo, and K. Kosuge, "Teleoperation system via computer network for dynamic environment," *In Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3534-3539, 1998.
- [10] D. Schulz, W. Burgard, and A. B. Cremers, "Predictive simulation of autonomous robots for teleoperation systems using the world wide web," *In Proceedings of IEEE/RSJ IROS Workshop on Robots on the Web*, pp. 31-36, 1998.
- [11] 홍순혁, 이상현, 전재욱, 윤지섭, "웹 기반 로봇 시뮬레이터," 제어 · 자동화 · 시스템공학 논문지, vol. 7, no. 3, pp. 255-262, March, 2001.
- [12] 박홍성, 정명순, 김봉선, "웹 기반 실시간 모니터링 시스템의 구조," 제어 · 자동화 · 시스템공학 논문지, vol. 7, no. 7, pp. 632-639, July, 2001.

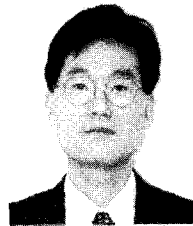
[13] 박태현, 강근택, 이원창, “인터넷을 이용한 이동로봇의 원격 운용 시스템,” 제어 · 자동화 · 시스템공학 논문지, vol. 8, no. 3, pp. 270-274, March, 2002.

[14] 이강희, 이연백, 김수현, 곽윤근, “위치 예측 시뮬레이터를 이용한 인터넷 로봇 시스템의 거리 오차 보상,” 한국정밀공학회지, vol. 20, no. 5, May, 2003.



이 강 희

1969년 4월 11일생. 1993년 KAIST 정밀공학과 (공학사). 1995년 KAIST 자동화 및설계공학과 (공학석사). 1995년~현재 KAIST 기계공학과 박사과정 재학중. 관심분야는 인터넷로봇.



김 수 현

1956년 8월 5일생. 1978년 서울대학교 기계공학과 (공학사). 1991년 Imperial College, Univ. of London 기계공학과 (공학박사). 1991년~현재 KAIST 기계공학과 교수. 관심분야는 나노/바이오 시스템, 측정 광학



곽 윤 근

1944년 6월 26일생. 1971년 서울대학교 기계공학과 (공학사). 1978년 University of Texas, Austin, 기계공학과 (공학박사). 1985년~현재 KAIST 기계공학과 교수. 관심분야는 감성로봇, 협로주행 로봇.