

# 가변 지속시간을 갖는 이벤트 기반 원격제어

박준영\*, 박장현\*\*

## Event Based Tele-Operation with Variable Holding Time

Joonyoung Park\*, Jahng-Hyon Park\*\*

### ABSTRACT

Necessity of the tele-operation has been increased in many fields. Since the Internet is inexpensive and available all over the world, it is a strong candidate for the transmission media of the tele-operation. However, the Internet has random time delays that may cause instability in the system especially if the tele-operation is bilateral. In the past few years many attempts have been made to overcome the random time delay. So far, they are still insufficient in terms of performance. The 'Variable holding time' is introduced to improve the performance of the 'Event based tele-operation' which controls a system with a non-time action reference. By holding each event for proper time, the system can quickly respond and be stabilized. The proper holding time should be selected based on the characteristics of the task that the system performs. The factors that reflect those characteristics are investigated. The fuzzy logic is employed to obtain the proper holding time for each event while the tele-operation system is in operation. The experimental results presented in this paper verify effectiveness of the proposed method.

**Key Words** : Tele-Operation(원격제어), Event Based Control(이벤트 기반 제어), Bilateral Control(양방향 제어), Internet(인터넷), Time Delay(시간지연), Robot(로봇), Fuzzy Logic(퍼지논리)

### 기호설명

$h$  = Holding time for each event  
 $s$  = Event : non-time action reference

### 1. 서론

인터넷의 활용범위는 점점 확장되어, 이를 원격지 로봇 시스템의 제어에 활용하려는 시도가 활발히 진행되고 있다. 인터넷은 이미 세계 대부분의 지역에 유·무선으로 구축되어 있으므로 이를

이용하면 저렴한 비용으로 어느 곳이나 원격제어 시스템을 구축할 수 있는 것이다(Fig. 1). 반면에 인터넷에는 불규칙한 시간 지연(Irregular time delay)이 존재한다는 단점도 있다. 이는 데이터 패킷(Packet)이 지나는 경로와 이 경로를 공유하는 다른 데이터와의 처리방식에 의해 필연적으로 일어나는 현상이며, 예측 불가능한 특성이 있다. 이러한 시간 지연은 실시간 양방향 제어 시스템(Real time bilateral control system)의 안정성에 치명적인 악영향을 줄 수 있다.

이러한 시간지연의 영향을 피하기 위해 기존

\*\*\* 2002년 3월 29일 접수  
\* 한양대학교 대학원 기계설계학과  
\*\* 한양대학교 기계설계학과

의 원격제어 연구들은 비실시간 제어(Non real time control)를 하거나, 단방향의 제어 방식(Unilateral control)을 취하거나, 또는 시간지연의 상한 폭(Bounded communication delays)을 두는 등의 가정하에 제어를 행한 사례가 대부분이었다. 최근 들어 인터넷상의 실시간 양방향 제어 시스템이 몇 가지 소개되고 있지만 이들 역시 개선의 여지가 남아있다. A. Bemporad 는 예측제어기(Predictive controller)를 이용해 원격지에 위치한 슬레이브(Slave) 시스템의 상태를 예측하여 제어하는 방법을 제시하였다<sup>(1)</sup>. 하지만 이 방법은 제어 대상의 정확한 모델링이 필요하다는 단점이 있다. 대부분의 로봇 시스템은 정확한 모델링이 쉽지 않을 뿐더러 로봇 시스템이 알려지지 않은 외부환경에서 작업할 경우에는 적용이 불가능 하다. G. Niemeyer 와 J.-J. E. Slotine 은 파동 변수 필터(Wave variable filter)를 이용하여 안정성을 확보하였다<sup>(2)</sup>. 그러나 이것은, 수동성(Passivity) 조건을 만족시키기 위해 작업자의 제어 입력과 피드백(Feedback) 정보를 재구성 하는 방식으로, 작업자에게 정확한 정보전달이 어려웠다. N. Xi 와 T. J. Tarn 은 시간에 독립적인 제어 기준(Non-time action reference)으로 제어하는 이벤트 기반 제어법을 제안하였다<sup>(3-5)</sup>. 이로서 불규칙 시간 지연의 크기에 상관없는 안정적인 제어를 실현 하였으나, 각 이벤트 사이의 시간 지연이 응답속도를 저하 시키는 단점을 가지고 있다.

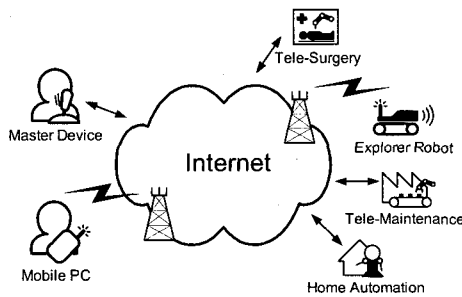


Fig. 1 Tele-Operations via Internet

본 논문에서는 기존의 이벤트 기반제어기에 가변 지속시간(Variable holding time)이란 개념을 도입함으로써 그 성능을 높이고자 한다. 이것은 현재의 작업형태에 따라, 시스템의 응답성과 안정성을 동시에 만족시키도록 적합한 시간 동안 하나의 이벤트를 지속시키는 방법이다. 적합한 지속시간

은 작업의 형태를 규정할 수 있는 여러 인자들로 부터 퍼지논리를 이용하여 실시간으로 구할 수 있다. 이와 더불어 피드백 데이터에 시각적인 위치 정보뿐만 아니라, 힘 정보를 포함시켜 작업자의 조작능력을 한층 높일 것이다<sup>(6)</sup>. 이러한 방법으로 어떠한 시간 지연에도 안정성을 유지하며 높은 응답성을 갖는 양방향 제어 시스템을 완성하고 이를 실제 로봇에 적용하여 그 타당성을 검증할 것이다.

## 2. 이벤트 기반 제어

### 2.1 시간에 독립적인 제어기준

대부분의 제어 방법들은 시간을 기준으로 제어가 행해진다. 이 방법을 통하면 사용자의 제어 입력과 시스템의 피드백 정보(Feedback data)는 쉽게 동기화(Synchronization)될 수 있다. 하지만 제어 루프상에 불규칙 시간 지연이 존재한다면 시간을 기준으로 한 제어방식은 그 동기화가 끊어지게 된다. Fig. 2 에서 보는 바와 같이 양방향 시간지연  $T_s$ ,  $T_r$  이 있을 경우, 사용자에게 피드백 되는 정보,  $Y(s - T_s - T_r)$  와 사용자가 새로 생성하는 제어 입력,  $R(t)$  은 같은 제어기준 상에 있는 정보가 아니다. 그러므로 사용자가 과거의 시스템 상태를 참조하여 새로운 제어 입력을 시스템에 가함으로써, 시스템은 불안정(Unstable)해질 수 있는 것이다.

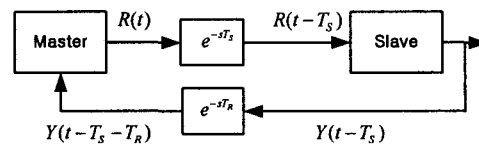


Fig. 2 The time based control through Internet

그러나 시간에 독립적인 제어 기준으로 시스템을 제어한다면 불규칙 시간 지연의 영향에 관계 없이 시스템의 동기화를 유지할 수 있다. Fig. 3 은 시스템의 이동거리, 절대 위치, 새로운 제어 입력의 도착여부 등과 같은, 시간에 독립적인 제어기준인 이벤트(Event, s)로 제어되는 시스템을 보여준다. 사용자는 원격시스템으로부터 피드백 정보,  $Y(s)$  를 받은 이후에야 새로운 제어입력,  $R(s)$  를 생성한다. 한편, 정지되어 있던 원격시스템은 제어 입력,  $R(s)$  의 도착 후 이를 시스템에 가하고 새로운 이벤트와 피드백정보,  $Y(s+1)$  를 생성한다.

이렇게 마스터 시스템과 슬레이브 시스템이 순차적으로 동작하기 때문에 교환되는 정보간 동기화가 유지되는 것이다.

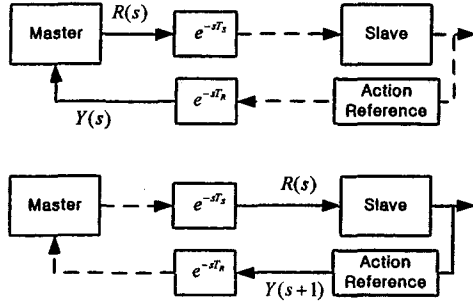


Fig. 3 The event based control through Internet

### 2.2 이벤트 기반 제어의 문제점

이벤트 기반 제어는 그 특성상 각 이벤트 사이에 불규칙 시간 지연이 존재 하게 된다. 이것은 시스템의 안정성 확보 과정에서 나온 필연적 결과로 그 시간 동안 시스템이 다음 이벤트를 기다리며 멈추게 된다. 이러한 이유로 Fig. 4 에서 보는 바와 같이, 사용자가 지속적으로 같은 제어 정보

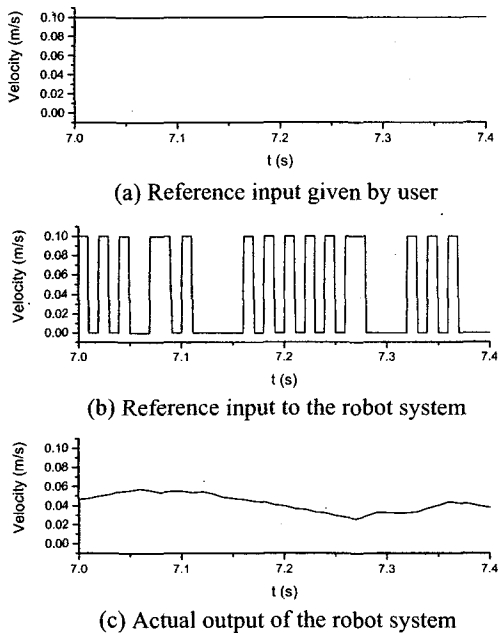


Fig. 4 Time delays between two events cause low performance

를 보낸다고 하더라도 실제 시스템은 각 이벤트 사이의 시간 지연이 반영된 정보를 받게 된다. 그리고 그 결과 시스템은 사용자의 의도에 크게 못 미치는 결과를 보이게 된다.

### 3. 가변 이벤트 지속시간

#### 3.1 이벤트 지속시간

앞에서 살펴본 바와 같이 이벤트 기반 제어에서는 각 이벤트 사이에 불규칙한 시간지연이 존재한다. 만약 한 이벤트가 끝난 시점, 즉, 시스템의 상태가 피드백 된 이후에도 이전 이벤트의 제어 입력을 일정시간 유지 시켜 준다면 이벤트 사이의 공백이 매워지고 연속적인 제어가 가능하다. 본 논문에선 이렇게 하나의 이벤트를 일정시간 유지시키는 시간을 이벤트 지속시간(Holding time,  $h$ ) 이라고 정의하겠다. 이 지속시간을 앞장의 예와 같이 비슷한 제어 입력이 들어올 경우나 시간지연이 사용자의 반응속도에 비해 작은 경우 등에 적용하면 빠른 응답 속도와 시스템의 채터링(Chattering) 방지효과를 얻을 수 있다. 그러나 Fig. 5 에서 보는 바와 같이, 지속시간이 적용되면 피드백 되는 시스템의 상태와 실제 시스템의 상태 사이는 그 지속시간만큼 격차가 발생하게 된다. 그 결과 정교한 작업이 어렵고 상이한 제어 입력에는 빠르게 대처하지 못하며 더 나아가서 시스템이 불안정해 질 수 있다. 그러므로 시스템의 안정성을 유지하면서 각 이벤트 사이의 공백을 최소화 시켜주는 적절한 이벤트 지속시간을 얻는 것이 중요하다.

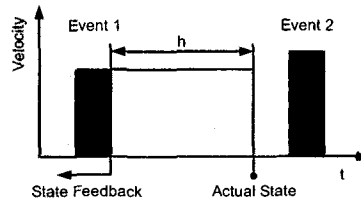


Fig. 5 Large holding time can cause instability

기존의 이벤트 기반제어에서는 시스템과 작업의 특성에 따라 실험적으로 얻어진 지속시간을 적용하였다. 그러나 로봇 시스템은 상이한 특성을 갖는 다양한 작업을 수행하기 때문에 기 결정된 하나의 이벤트 지속시간으로는 전 작업영역에서

만족할 만한 결과를 얻기 힘들다. 그래서 본 논문에서는 작업도중 작업상황에 적합한 이벤트 지속시간을 실시간으로 구하여 적용시키는 시스템을 개발하게 되었다.

**3.2 이벤트 지속시간의 결정인자**

자동적으로 적합한 이벤트 지속시간을 얻기 위해서는 먼저 이의 결정에 필요한 인자(Factor)들을 생각해 보아야 한다. 이러한 인자들로 작업자의 의도, 제어 입력의 유사성, 주변 환경에 의한 요인을 들 수 있다. Table 1 에는 각 인자들의 특성이 요약되어 있다.

Table 1 Factors to decide holding time

h Factor	Very Small (VS)	Small (S)	Medium (M)	Large (L)	Very Large (VL)
User's Intention	Very Careful (VC)	Careful (C)	Normal (N)	Rapid (R)	Very Rapid (VR)
Ref. Input	Very Different (VDIF)	Different (DIF)	Similar (SIM)	Very Similar (VSIM)	Same (SAM)
Sensor	Very Small (VS)	Small (S)	Medium (M)	Large (L)	Very Large (VL)

(1) 작업자의 의도 : 작업자의 의도란 작업자가 최근에 피드백 받은 데이터를 토대로 앞으로 어떠한 작업을 수행할지를 원격시스템에 알려주는 것이다. 로봇을 이용하여 정밀한 위치제어나 힘 제어를 원한다면 작은 이벤트 지속시간을 적용하여야 할 것이다. 이렇게 함으로서 시스템은 느린 응답성을 갖지만 제어 데이터간의 견고한 동기화가 형성되기 때문에 정밀한 제어를 할 수 있다. 반면에 주변에 장애물이 없는 상황에서 빠른 응답을 원하는 작업을 원할 경우엔 이벤트 지속시간을 늘려서 각 이벤트가 연속적으로 일어나도록 하여야 할 것이다.

(2) 제어 입력의 유사성 : 시스템에 도달하는 제어 입력이 여러 이벤트에 거쳐 일정한 값이 들어

온다면 다음에도 비슷한 값이 들어올 가능성이 높다. 이러한 경우엔 이벤트 지속시간을 늘려서 시스템에 계속 일정한 입력이 들어가도록 할 수 있고 이로서 빠른 속도로 시스템을 제어할 수 있다. 반면에 상이한 제어입력이 들어올 경우에는 이벤트 지속시간을 줄여서 새로운 제어 입력에 대응해야 할 것이다.

(3) 주변 환경 : 로봇 시스템은 센서를 통해 주변환경을 감지하고 자신이 처한 상태를 판단할 수 있다. 가까운 곳에 장애물이 감지된 경우나 외부에서 힘이 가해지는 경우엔 조심스럽게 제어해야 한다. 조금이라도 동기화가 어긋나면 시스템에 손상을 줄 수 있기 때문에, 각 이벤트의 결과를 얻은 이후에 시스템에 새로운 제어 입력을 주어야 한다. 그러므로 이러한 때에는 이벤트 지속시간을 충분히 줄여야 할 것이다.

**4. 시스템 설명**

가변 지속시간을 갖는 이벤트 기반 원격제어 시스템을 Fig. 6 과 같이 구성하였다. 먼저 사용자(Master)는 원격 시스템(Slave)으로부터 피드백 정보인 로봇의 위치  $P(s)$ 와 가해지는 힘  $F(s)$ 를 받게 된다. 이를 근거로 사용자는 새로운 이벤트에 쓰일 제어 입력  $V_r(s)$ 와 원하는 작업 특성  $U(s)$ 를 보내게 된다. 인터넷을 통해 새로운 제어 입력을 받은 원격 시스템은 센서 값  $S(s)$ , 작업자의 의도  $U(s)$ , 속도유사성 등의 정보를 취합해 해당 이벤트에 적합한 지속시간  $h(s)$ 를 구한다. 그리고 이벤트 매니저는 새로운 입력이 들어오는 순간의 시스템 상태  $P(s)$ ,  $S(s)$ 를 사용자에게 피드백 시키고, 앞서 구한 이벤트 지속시간 동안 제어 입력을 유지시키게 된다. 그리고 이벤트 매니저에서 최종적으로 구성된 제어 입력  $V_r(t)$ 를 따라 로컬 컨트롤러(Local controller)가 로봇을 제어하게 된다. Fig.

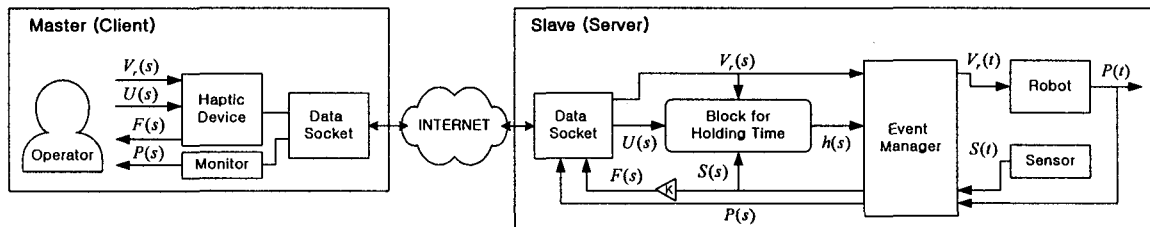
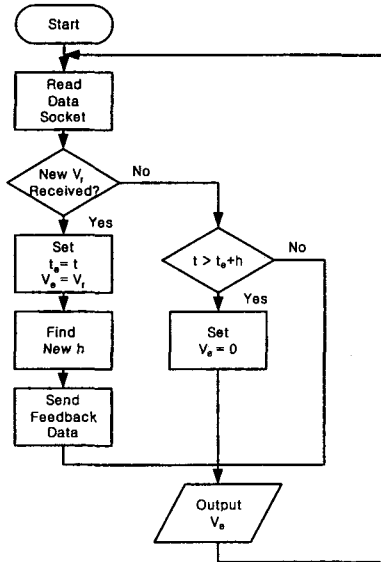


Fig. 6 Block diagram of event based tele-operation system with variable holding time

7 에는 이벤트 매니저가 수행하는 알고리즘을 플로우 차트로 나타내었다.



$V_r$  : Reference data from the Master  
 $t_e$  : Time on which the event started  
 $t$  : Current Time  
 $V_e$  : Reference Input  
 $h$  : Holding Time

Fig. 7 Flow chart of the event manager

#### 4.1 퍼지 논리를 이용한 지속시간 결정

앞서 설명한 인자들로부터 적절한 이벤트 지속시간을 얻는, Fig. 8 과 같은 시스템을 구성하였다. 이 시스템은 센서와 작업자로부터 정보를 얻어 퍼지 논리(Fuzzy logic)를 이용하여 해당 이벤트에 적합한 지속시간을 구한다. 퍼지 논리에 사용된 소속함수(Membership function)는 Fig. 9 에 보여지는 것과 같이 모두 삼각형 함수를 이용하였다. 또 퍼지규칙(Fuzzy rule)은 Fig. 10 에 규칙표면(Rule surface)로 표현되었다.

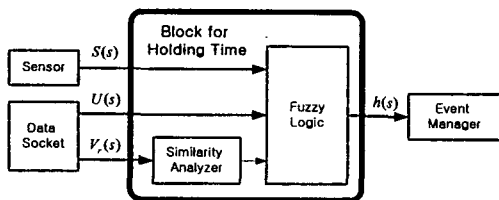


Fig. 8 System Block for deciding the holding time

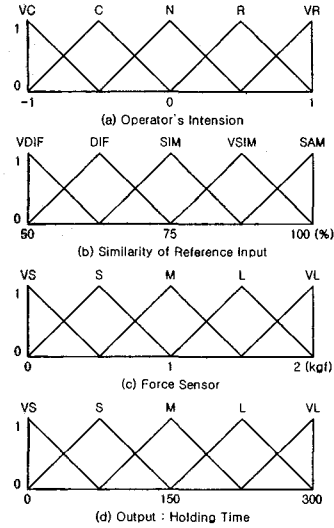


Fig. 9 Fuzzy membership functions

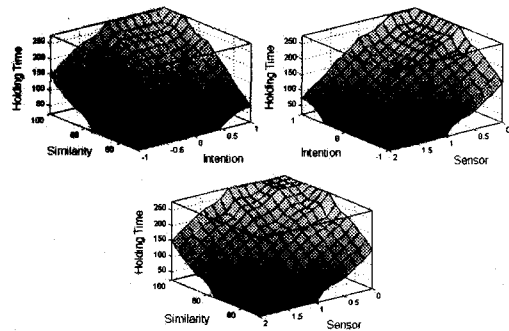


Fig. 10 Fuzzy rule surfaces

#### 4.2 로봇 시스템

제어 대상으로는 Fig. 11 과 같은 3 절 로봇 팔을 선정하였다. 이 로봇의 기구학적 관계는 다음의 식 (1)과 (2)로 표현된다.

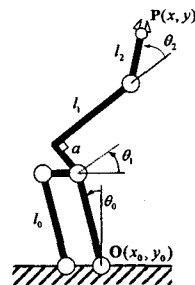


Fig. 11 Robot arm

$$P = \begin{bmatrix} x_0 - l_0 \sin \theta_0 + l_1 \cos \theta_1 \\ -a \sin \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y_0 + l_0 \cos \theta_0 + l_1 \sin \theta_1 \\ +a \cos \theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (1)$$

$$dP = J d\theta \quad (2)$$

where

$$J = \begin{bmatrix} -l_0 \cos \theta_0 & \begin{pmatrix} -l_1 \sin \theta_1 - a \cos \theta_1 \\ -l_2 \sin(\theta_1 + \theta_2) \end{pmatrix} & -l_2 \sin(\theta_1 + \theta_2) \\ -l_0 \sin \theta_0 & \begin{pmatrix} l_1 \cos \theta_1 - a \sin \theta_1 \\ +l_2 \cos(\theta_1 + \theta_2) \end{pmatrix} & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

또, 이를 제어 하기 위하여 Fig. 12 와 같은 제어 시스템을 구성하였다. 사용자는 로봇 끝점(End effector)의  $x, y$  방향의 속도만을 전달하므로 로봇은 여유 자유도(Redundant degrees of freedom)를 가진다. 그러므로 식(3)에 나타낸 것과 같은 자코비안 행렬(Jacobian matrix)의 Pseudo-inverse,  $J^\#$  을 통해 로봇 관절이 최소한의 움직임을 갖게 하는 각 관절 각속도의 최적해를 구하게 된다<sup>(7)</sup>. 그리고 이를 통해 얻은 관절의 속도를 추종하도록, 기존의 PI 제어기를 통해 로봇을 제어한다.

$$J^\# = J^T [J J^T]^{-1} \quad (3)$$

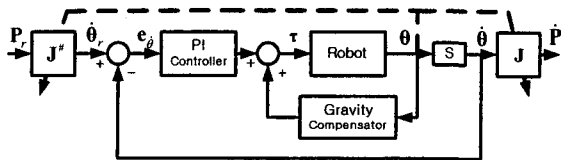


Fig. 12 Local controller for the robot arm

### 5. 실험

앞서 설명한 시스템을 Fig. 13 에 나타낸 것과 같이 직접 구성하여 실험을 행하였다. 제어 대상물은 6 축 로봇(FARAMAN As1-i )이다. 이 로봇의 6 개의 회전 조인트(Revolute joint) 중 3 개를 사용하여 한 평면상에서 동작하는 3 관절 로봇 팔을 구성하였다. 그리고 이 로봇의 제어를 위해 다축 제어용 보드(Multi Motion Controller)를 사용하였다.

또한 이 로봇의 끝에는 6 축 포스/토크 센서(6 Axis Force / Torque sensor)를 장착하여 외부에서 가해지는 힘을 측정하였다. 사용자는 포스 피드백 조이스틱(Force feedback joystick)을 통하여, 제어 입력인 로봇 끝점의 속도를 보내주고 로봇에 가해지는 힘을 피드백 받는다.

실험은 인터넷을 통하여 서울의 한양대학교와 광주과학기술원간에 진행되었다. 실험도중 이 두 지점간의 평균 왕복 시간 지연은 291msec 였다. 제한된 시스템이 여러 종류의 작업에서 높은 응답속도와 정확성, 안정성을 동시에 확보한다는 것을 검증하기 위해 세가지 종류의 작업을 선정하여 실험을 행하였다. 이 실험의 비교대상은 기존의 이벤트 제어 방식 즉, 이벤트 지속시간이 이미 결정된 시스템이다. 지속시간이 30msec 로 안정적인 제어에 적합하도록 설정된 시스템과 200msec 로 응답성이 높게 설정된 시스템 등 두 가지를 선정하였다.

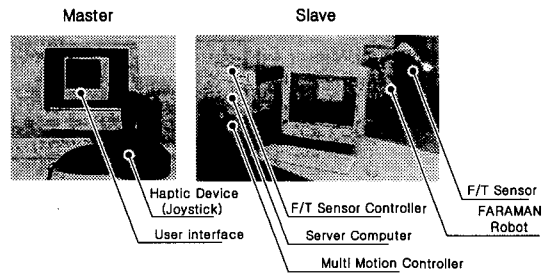


Fig. 13 System configuration for the experiments

#### 5.1 실험 결과 1

먼저 주변에 장애물이 없는, 혹은 로봇의 속도에 비해 매우 멀리 떨어져 있는 환경에서의 작업을 수행하였다. 이러한 상황에서 로봇이 한 영역에서 다른 영역으로 이동할 경우에는 정확성보다는 빠른 작업 속도가 중요시 된다. 현재 사용자는 지속적으로 1m/s 의 속도를 시스템에 가하고 있다. 그러나 Fig. 14 는 이벤트 지속시간이 30msec 으로 제한되어 있는 경우로, 각 이벤트 사이의 시간지연으로 인해서 사용자의 의도와는 상이한 결과를 보여준다. 반면에 이벤트 지속시간이 200msec 로 상대적으로 큰 값이 설정된 경우인 그림 Fig. 15 과 제한된 방법에 의하여 이벤트 지속시간을 결정하는 경우인 그림 Fig. 16 에서는 좋은 결과를 볼 수 있다.

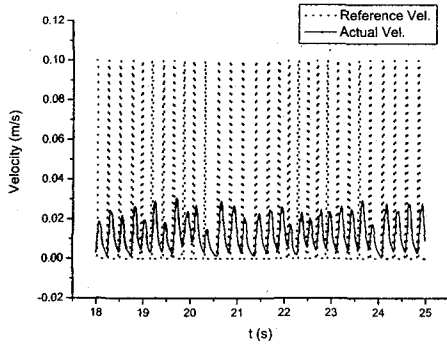


Fig. 14 Reference & actual velocity (h=30msec)

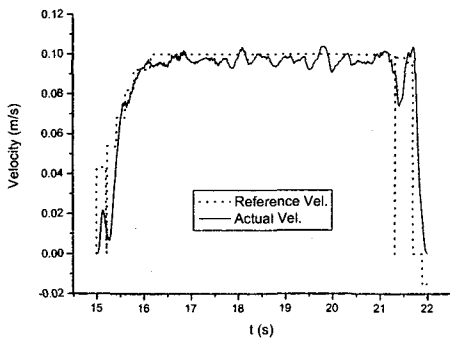


Fig. 15 Reference & actual velocity (h=200msec)

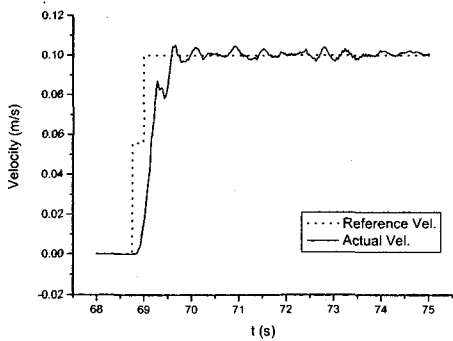


Fig. 16 Reference & actual velocity (Variable h)

## 5.2 실험 결과 2

다음으로 정확한 목표 위치로 이동시키는 실험을 행하였다. 정교한 작업을 위해 작업대상물에 접근하는 경우로 빠른 응답보다는 정확성, 안정성이 중요시 되는 작업이다. 이벤트 지속시간이 30msec 로 설정된 경우(Fig. 17)에는 사용자가 로봇의 상태를 피드백 받은 이후에 새로운 제어 입력을 보내기 때문에 속도는 다소 느리지만 목표지점

에 정확히 도달하는 것을 볼 수 있다. 반면 이벤트 지속시간이 200msec 로 설정된 경우(Fig. 18)엔 목표지점으로 접근하는 속도는 빠르지만 동기화가 어긋나 있기 때문에 정확한 제어가 어렵다는 것을 보여준다. 마지막으로 제안된 방법의 경우(Fig. 19)엔 목표지점으로 접근할수록 이벤트 지속시간을 감소시키기 때문에 처음엔 빠른 속도로 접근하다가 목표지점 근처로 갈수록 정확히 제어되는 것을 볼 수 있다.

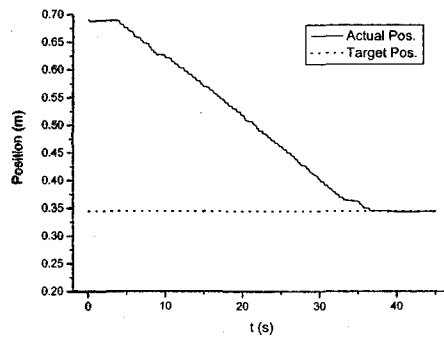


Fig. 17 Target & actual position (h=30msec)

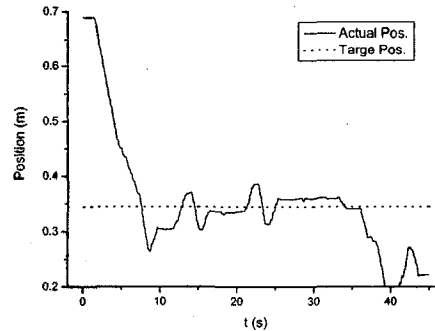


Fig. 18 Target & actual position (h=200msec)

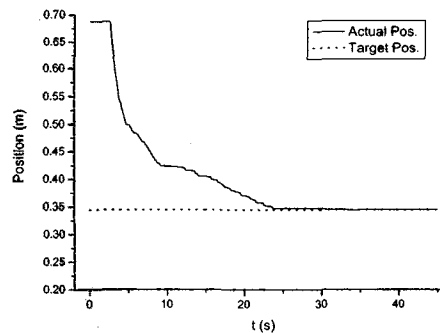


Fig. 19 Target & actual position (Variable h)

### 5.3 실험 결과 3

마지막으로 외부환경과 접촉하여 힘 제어를 행하였다. 이러한 작업을 수행할 경우엔 사용자와 로봇의 동기화가 조금만 어긋나도 과도한 힘이 가해져서 로봇이 손상을 입을 위험성이 있다. 그러므로 안정성에 초점을 맞춘 제어가 행해져야 한다. 이벤트 지속시간이 30msec로 설정된 경우(Fig. 20)와 제안된 방법에 의해 이벤트 지속시간을 충분히 줄여준 경우(Fig. 22)가 200msec로 고정된 경우(Fig. 21)보다 오차가 적다는 것을 알 수 있다.

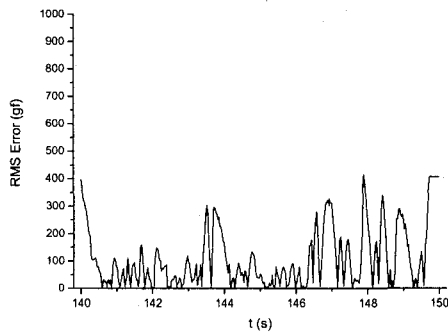


Fig. 20 Force control error (h=30msec)

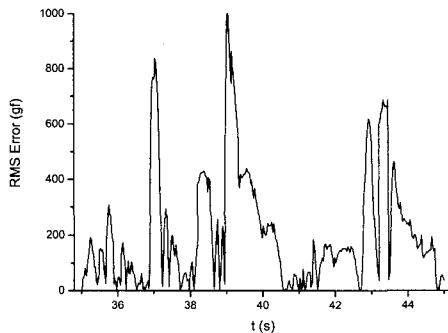


Fig. 21 Force control error (h=200msec)

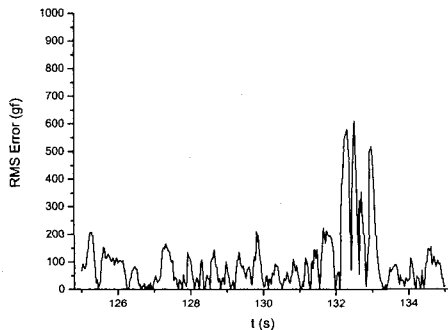


Fig. 22 Force control error (Variable h)

### 6. 결론

인터넷을 통해서 속도와 힘을 제어할 수 있는 양방향 원격 제어 시스템을 개발하였다. 이 시스템에는 인터넷 상에 존재하는 불규칙한 시간 지연을 극복하기 위해 이벤트 기반 제어법이 적용되었다. 더 나아가서 이 시스템의 응답속도의 향상을 위해서 가변 지속 시간을 제안하였다. 이것은 작업의 특성에 따라서 이벤트를 일정시간 지속시키는 방법으로 시스템이 안정성과 높은 응답성 모두를 충족시킨다. 이 가변 지속시간은 사용자와 센서로부터 정보를 얻어 퍼지논리를 이용하여 얻을 수 있었다. 그리고 실제 실험을 통해 제안된 방법이 여러 종류의 작업에서 우수한 결과를 보여준다는 것을 확인할 수 있었다.

### 참고문헌

1. Bemporad, A. "Predictive Control of Teleoperated Constrained Systems with Unbounded Communication Delays," Proceedings of the 37th IEEE Conf. on Decision & Control, pp. 2133-2138, 1998.
2. Niemeyer, G. and Slotine, J.-J. E., "Towards Force-Reflecting Teleoperation Over the Internet," Proceedings of the 1998 IEEE Int. Conf. on Robotics & Auto., pp. 1909-1915, 1998.
3. Xi, N. and Tarn, T. J., "Action Synchronization and Control of Internet Based Telerobotic Systems," Proceedings of the 1999 IEEE Int. Conf. on Robotics & Auto., pp. 219-224, 1999.
4. Xi, N. and Tarn, T. J., "Stability analysis of non-time referenced Internet-based telerobotic systems," Robotics and Autonomous Systems, Vol. 32, pp. 173-178, 2000.
5. Elhajj, I., Xi, N. and Liu, Y., "Real-Time Control of Internet Based Teleoperation with Force Reflection," Proceedings of the 2000 IEEE Int. Conf. on Robotics & Auto., pp. 3284-3289, 2000.
6. Sheridan, T. B., "Telerobotics, Automation, and Human Supervisory Control," The MIT Press, 1992.
7. Asada, H. and Slotine, J.-J. E., "Robot Analysis and Control," John Wiley A Sons, pp. 51-70, 1986.