

힘 반영 원격 제어 시스템의 모델링

° 최홍석*, 차덕순*, 정완균*, 염영일*, 정규원**
* 포항공과대학 기계공학과, ** 산업과학기술연구소

A Model for Force Reflected Teleoperating System

° H.S. Choi*, D.S. Cha*, W.K. Chung*, Y. Youm*, K.W. Chung**
* Dept. of Mech. Eng., POSTECH, ** RIST

Abstract : Teleoperation is the extension of a person's sensing and manipulation capability to a remote location based on sufficient on-line information about the task and environment. It is known that kinesthetic feedback from slave to master increases the realism with which the operator interacts with the environment.

This paper presents a model for teleoperator control system which includes human dynamics and kinesthetic feedback. This proposed model is simulated and verified by experiments using the PUMA 560 slave robot and developed master arm.

1 서론

원격 제어 시스템은 조작자와 작업 환경이 공간적으로 떨어져 있으므로 조작자에게 현장의 충분한 정보가 제공되도록 해야한다. 자유 공간 상에서의 작업은 시각 정보 만으로도 충분히 컨트롤이 가능하지만 작업 물체와의 접촉이 요구되는 작업에서는 접촉 힘에 대한 정보가 조작자에게 전달되어야 원만하게 작업을 수행할 수 있다.

조작자에게 반영되는 힘은 현장감을 살릴 수 있다는 측면에서는 필수적인 정보이지만 조작자의 피로, 원격 제어 시스템의 유연성과 작업 능력의 저하를 초래할 수 있으며, 심지어는 시스템을 불안정하게 하는 원인이 될 수도 있다. 이러한 힘 반영 시스템을 잘 디자인하고, 컨트롤하기 위해서는 적당한 prototyping과 실험이 요구된다. 따라서, 힘 반영 원격 제어 시스템에 대한 적당한 모델링이 가능하다면, 컨트롤 시스템 디자인과 이에 따른 실험의 가이드로써 활용할 수 있을 것이다. 본 논문에서는 조작자의 dynamics를 고려한, 힘 반영 원격 제어 시스템 전체를 모델링하고, 시뮬레이션과 실험을 통해 이 모델의 타당성을 조사하였다.

2 하드웨어 구성

2.1 Master Arm의 구성

원격 제어 시스템에서 master arm은 인간의 명령을 slave robot에 정확히 전달시키는 도구로써 뿐만 아니라, 작업 현실감을 조작자에게 반영시키는 도구로써 중요한 부분을 차지하고 있다. 1992년 포항공대 로봇 실험실에서 제작한 6 자유도 master arm은 단순

위치 제어만 가능하도록 설계되어졌으므로[6], 이를 개조하여 1축 힘 반영이 가능하도록 하였다[7]. 힘 반영을 위한 master arm의 actuator로는 torque-controlled DC servo motor를 사용하였다. 그리고, 위치 측정용 센서는 backlash로 인한 오차를 없애기 위해 직접 master arm의 1축에 부착하였다. DC servo motor는 일반적으로 고속 저출력에 적합한 actuator이기 때문에 실제 힘 반영 제어에서는 시스템의 보호를 위해 최고 출력에 제한을 두어, master arm에 반영할 수 있는 작업 공간 내에서의 최대 힘의 크기를 20 N으로 한정하였다.

2.2 Slave Robot의 구성

Slave robot으로는 산업용으로 많이 쓰이고 있는 PUMA 560을 이용하였다. PUMA의 손목에 걸리는 힘을 측정하기 위해 손목에 부착된 force/torque sensor는 Lord사에서 제작한 Model 75/250을 사용하였다. 최대 33 N의 힘과 28.2 Nm의 torque를 측정할 수 있으며, 측정 주파수는 최대 100 Hz 즉 10 ms마다 한번씩 측정할 수 있다. FT sensor의 정밀도는 약 0.261 N이다. 실제로는 sensor자체의 하중과 부하의 관성의 영향으로 인해 정확한 접촉력을 측정할 수는 없었다.

2.3 실시간 제어기의 구성

실험에 사용된 원격 제어 시스템을 실시간으로 제어하기 위해 Motorola사에서 개발된 VMEbus 시스템을 사용했다. 주 CPU인 MVME 147SA board를 이용하여 system management, program 개발, program download, 사용자와의 interface 등을 담당하게 하고, target CPU로 MVME 147 board를 사용하여 download된 program이 실시간으로 수행되도록 하였다. Digital input/output board는 AVME 9470을, analog output board로는 AVME 9210을 사용했다.

각각의 독립적인 target CPU board들은 shared memory를 통해 서로 data를 공유하여 병렬 처리됨으로써 실시간 제어가 가능하도록 하였다. 실시간 제어기의 각 구성 요소의 역할 및 기능은 그림 1에 나타나 있다. 3개의 target CPU는 30 ms마다 data를 교환하도록 운영 시간이 일치되어 있다. 실제적인 sampling frequency는 slave robot의 position 및 master arm의 torque control을 담당하는 target CPU의 frequency인 500 Hz로 볼 수 있다.

2.4 원격 제어 알고리즘

원격 제어 시스템의 제어 기법으로는 좌표계가 고정되어 조작자의 방향 판단에 유리한 forward & inverse kinematics를 이용하였다. 제어 시스템의 기본적인 제어선도는 그림 2에 나타나 있다. 제어 순서를 살펴보면 먼저 master arm의 각 관절의 각도를 계산하고, forward kinematics를 이용하여 계산한다. 그 계산된 위치에 slave robot과의 작업 공간 비율을 맞추기 위해 position scale factor를 곱한다. 증폭된 위치를 PUMA의 inverse kinematics를 풀어 slave robot의 각 관절의 원하는 각도를 구한다. 이 값을 15 등분으로 interpolation하여 500 Hz의 sampling frequency로 위치 제어를 한다. Master arm에의 힘 반영을 위해 slave robot의 손목에 부착된 FT sensor로부터 100 Hz마다 값을 읽어 이 값을 힘과 모멘트 성분으로 분해한다. 여기에 시스템을 안정화시키기 위해 force scale factor를 곱하고, master arm의 Jacobian을 이용해 statics를 계산하여 master arm에 힘 반영하였다.

실제적으로 힘 반영 제어 실험은 1 축만으로 제한하였으므로 1 축의 기준 좌표의 원점으로부터 master arm의 joystick까지의 직선 거리를 구하고, 1 축에 수평으로 직각인 접촉력을 곱한 모멘트를 기준으로 master arm의 1 축 힘 반영 제어를 하였다. 그림 3은 전체 원격 제어 시스템의 layout이다.

3 모델링

모델링의 간략화를 위해 그림 4와 같은 단순한 1 축 시스템을 기준으로 하고, 제어 기법만은 실제의 시스템과 같게 하였다.

3.1 Generalized Impedance Control

원격 제어 시스템에서 조작자는 명령을 발생시키는 가장 중요한 구성 요소이다. 그러나, 조작자의 명령은 많은 오차를 포함하고 있으므로, man-machine interaction의 개념에 필요한 모델이 필요하다. Generalized impedance control(GIC)은 이러한 오차를 포함하고 있는 명령을 수행할 수 있는 제어 기법으로, 위치 제어와 힘 제어를 서로 분리시키지 않은 통일된 것으로 식 (1)와 같이 표시된다[5].

$$M_d(\ddot{X} - \ddot{X}_d) + B_d(\dot{X} - \dot{X}_d) + K_d(X - X_d) = B_f(\dot{F}_e + \dot{F}_c) + K_f(F_e + F_c) \quad (1)$$

여기서, M_d, B_d, K_d, B_f, K_f 는 generalized impedance의 parameters이고, F_e, F_c 는 각각 desired contact force와 measured interaction force, $X_d, \dot{X}_d, \ddot{X}_d$ 는 각각 원하는 위치, 속도, 가속도이다. 즉 GIC는 M_d, B_d, K_d, B_f, K_f 의 2 종류의 parameter를 이용하여 위치 제어와 힘 제어사이의 절충을 시도한 것이다.

3.2 인간 팔의 모델링

조작자가 master arm에 위치 명령을 주기 위해 팔의 근육에 힘을 가할 때, 그 힘의 입력을 목표 지점까지의 위치 오차와 접촉 지점에서의 힘의 오차 2 가지로 나누어 고려한다. 그리고 인간이 발생시키는 힘을 위치 오차에 의해 발생시키는 힘과 힘의 오차에 의해 발생하는 힘으로 나누어 생각한다.

먼저 위치 오차를 e_p , 힘의 오차를 e_f 라 하자. F_h 를 human actuating force, $B_{hv}, K_{hv}, B_{hf}, K_{hf}$ 를 위치와 힘에 대한 impedance parameter라 할 때, $F_h = F_{hv} + F_{hf}, F_{hv} =$

$B_{hv}\dot{e}_p + K_{hv}e_p, F_{hf} = B_{hf}\dot{e}_f + K_{hf}e_f$ 가 된다. F_{ch} 를 인간 팔에 작용하는 interaction force, M_h 를 human arm physical mass, X_m 을 master arm의 위치라 할 때, 인간 팔의 운동 방정식은 아래와 같다.

$$F_h - F_{ch} = M_h\ddot{X}_m. \quad (2)$$

3.3 Master Arm의 모델링

Master arm의 운동 방정식은 아래와 같다.

$$F_h - F_r = M_m\ddot{X}_m + B_m\dot{X}_m. \quad (3)$$

여기서, F_r 은 reflecting force, M_m, B_m 은 master arm의 mass와 damping coefficient이다. 식 (2)을 식 (3)에 대입하면, master arm과 조작자의 팔을 하나의 시스템으로 하여 식 (4)과 같이 나타낼 수 있다.

$$F_h - F_r = (M_h + M_m)\ddot{X}_m + B_m\dot{X}_m. \quad (4)$$

Slave robot에의 위치 명령(X_{ds})는 master arm의 위치에 position scale factor(K_{ps})를 곱해야 하므로 아래와 같다.

$$X_{ds} = K_{ps}X_m. \quad (5)$$

3.4 Slave Robot의 모델링

Slave robot의 dynamic model을 그림 5와 같이 생각할 때, 운동 방정식들은 아래와 같다.

$$X_e = X_{ds} - X_s \quad (6)$$

$$F_s = K_{sp}X_e - K_{sp}d\dot{X}_s \quad (7)$$

$$F_s + K_wX_w = (M_{s1} + M_{s2})\ddot{X}_s + B_s\dot{X}_s \quad (8)$$

$$-K_w - F_a = M_{s2}\ddot{X}_w \quad (9)$$

$$M_s = M_{s1} + M_{s2} \quad (10)$$

$$F_a = K_{ev}(X_s - X_{ev}) \quad (11)$$

$$F_r = K_fF_a. \quad (12)$$

여기서, M_{s1} 은 effective mass of slave robot, M_{s2} 은 effective mass of FT sensor, X_e 은 command position error, X_{ev} 은 position of environment, X_w 은 deflection of FT sensor, B_s 은 damping coefficient of slave robot, K_w 은 impedance of FT sensor, K_{ev} 은 inpedance of environment, K_f 은 force scale factor, K_{sp} 은 position feedback gain of slave robot, $K_{sp}d$ 은 velocity feedback gain of slave robot, F_s 은 command force to slave robot, F_a 은 measured contact force 이다.

3.5 전체 시스템의 모델링

지금까지 조작자의 팔, master arm, slave robot으로 구분하여 부분별로 모델링한 것을 그림 6에 전체적으로 나타내었다. 그림에 나타난 전달 함수들은 다음과 같이 정의한다. $G_{hv} = B_{hv}s + K_{hv}$, $G_{hf} = B_{hf}s + K_{hf}$, $Z_h = M_h s^2$, $Z_m = M_m s^2 + B_m s$, $Z_e = K_{ev}(1 - X_{ev}/X_s)$. 조작자를 기준으로 볼 때, slave robot의 위치 제어는 단순히 master robot의 위치 명령을 추종하는 open loop control이고, master arm의 힘 반영 제어도 작업 물체와의 접촉력을 단순히 조작자에게 반영시키는 open loop control이

표 1. 시뮬레이션에 사용된 parameters

parameter group	parameter	value
Human arm	$B_{hf} = 0.4$	$K_{hf} = 0.3$
	$B_{hv} = 1.5$	$K_{hv} = 0.3$
	$M_h = 1.0$	
Master arm	$B_m = 0.05$	$M_m = 0.8$
	$K_{ps} = 1.25$	
Slave robot	$K_w = 25000$	$K_{ev} = 45000$
	$K_{spd} = 25.0$	$K_{spp} = 87.0$
	$M_{s1} = 2.5$	$M_{s2} = 0.1$
	$B_s = 0.11$	

다. 그러나 위의 두가지 제어가 조작자를 중심으로 서로 연결되므로 조작자를 포함하는 전체 시스템은 closed loop control이 된다. 또한 전체적으로 위치 제어에 의해 작동되는 시스템이므로 작업 물체와의 접촉힘을 직접 제어하는 것은 거의 불가능하고, 단지 무리한 접촉힘이 slave robot과 작업 물체에 발생하지 않도록 조작자에게 접촉힘에 대한 정보를 전달하는 기능만이 가능하다는 것을 알 수 있다.

4 실험 및 시뮬레이션 방법

4.1 시뮬레이션

작업 물체는 slave robot의 초기 위치로부터 5 cm 떨어진 곳에 있다고 가정하였고, 원하는 접촉력 F_d 는 5 N으로 정했다. 인간의 팔의 흔들림을 고려하여 크기 1 cm, 주파수 10 Hz의 외란을 위치 오차에 첨가했다. 조작자의 위치 판단 오차를 고려하여 정확한 작업 물체와의 거리보다 1.125 배 먼 거리를 위치 입력 X_d 로 하였다. 또한 인간의 감각의 부정확함을 고려하여 크기 0.5 N, 주파수 50 Hz의 외란을 힘 오차에 첨가하였다. 실제의 시스템에서 발생하는 중력의 영향, actuator의 saturation, sampling 및 quantization 등에 따른 비선형성과 기구학 계산에 따른 시간 지연, 인간의 계속적인 작업에 의한 학습 효과 등은 무시되었다.

시뮬레이션에 사용된 parameter들의 값이 표 1에 나와 있다. 이 중 $M_h, B_m, M_m, K_w, K_{ev}, M_{s1}, M_{s2}, B_s$ 는 실험적으로 얻어진 slave robot의 특정 position response와 force response가 simulation 결과와 비슷하게 되도록 값을 임의로 조정한 값들이다. 미분 방정식은 4th order Runge-Kutta 수치 적분 알고리즘을 이용하였고, integral interval은 sampling time과 같은 2 ms로 해서 풀었다.

4.2 실험

힘 반영 원격 제어의 실험은 slave robot의 end effector에 알루미늄 막대를 고정시키고 나서, 작업대의 두께 4 mm의 철판에서 안정된 접촉을 유지시키는 것을 목적으로 하고, force scale factor (K_f)의 값을 변화시켜 가면서 시스템의 응답 특성을 조사하였다.

5 결과 및 토의

Force scale factor (K_f)의 크기를 변화시켜가면서 simulation과 실험 결과를 비교해 나타난 그래프들이 그림 7 - 그림 12에 나타나 있다. K_f 의 변화에 따른 시스템의 특성을 비교한 것은 현재와 같은 단순한 kinesthetic force reflecting system에서는 K_f 의 크기가 시스템의 안정성에 큰 영향을 미치지 때문이다[7].

K_f 가 0.2 일 때는 비교적 안정된 접촉 상태를 유지할 수 있으나 K_f 가 0.3 일 때 oscillation이 나타나기 시작하고, 0.4 일 때

는 시스템이 불안정해져서 제대로 접촉을 유지하지 못함을 알 수 있다. K_f 가 증가함에 따라 접촉력이 증가하고, $K_f = 0.4$ 일 때 제대로 접촉을 유지하지 못하는 것은 시뮬레이션과 실험의 전체적인 경향이 대체로 일치하고 있음을 보여준다. 그래프에서 나타나듯 master arm의 overshoot이 실험 값에 비하여 시뮬레이션 값이 상당히 크고, 그에 따라서 peak contact force 값도 시뮬레이션의 값이 상당히 더 크다. 이것은 실험의 경우는 조작자가 시각 정보를 위치 오차의 계산에 이용할 뿐만 아니라, 일종의 근접 센서로서도 활용하기 때문이다. 그러므로 좀더 정확한 human model을 만들 필요가 있고, 동시에 시각 정보가 불충분한 환경에서는 근접 센서의 활용도 고려되어야 할 것이다.

위의 실험 결과를 통해 조작자에게 반영되는 힘은 현장감을 전달시키는 정보인 동시에 조작자의 위치 명령을 교란시키는 하나의 외란으로도 작용할 수 있다는 사실을 알 수 있다. 이것은 반영되는 힘의 크기가 커질수록 접촉력도 같이 커지는 사실을 통해서도 확인할 수 있다. 즉 반영 힘이 커지면 조작자도 반사적으로 큰 힘을 내고, 이 힘들이 서로 상승 작용을 하여 시스템이 불안정해 지는 것을 알 수 있다. 그러므로 kinesthetic force reflecting system에서는 적절한 K_f 의 선정이 중요하다는 것을 알 수 있다.

6 결론

본 논문에서는 PUMA 560 로봇을 slave로 하는 1 축 힘 반영 원격 제어 시스템을 구축하고, 조작자를 포함한 전체 콘트롤 시스템의 모델링을 재안하였다. 그리고, 시뮬레이션과 실험을 통해 이 모델링의 타당성을 조사하였다. 모델을 이용한 시뮬레이션과 실험 결과를 비교해 볼 때 전체적인 경향은 비슷하게 나타나나 human model의 불완전성으로 인해 약간의 오차를 드러냈다. 인간의 판단 능력 또는 학습 능력 등을 고려한 human model이 필요함을 알 수 있다.

인간 팔의 dynamics를 고려한 이번 원격 제어 시스템의 모델은 master와 slave 간의 dynamic characteristic이나 control 알고리즘 설계의 가이드로써 활용할 수 있다. 원격 제어 시스템의 advanced control 형태인 시간 지연, shared control 등이 포함된 model에 대해서도 더 연구가 필요하다.

참고 문헌

- [1] A. K. Bejczy and M. Handlykken, "Experimental Results with a Six Degree of Freedom Force Reflecting Hand Controller", *Int. Proc. 17th. Annual Conf. on Manual Control*, pp. 465-477, 1981.
- [2] B. Hannaford and R. Anderson, "Experiment and Simulation Studies of Hard Contact in Force Reflecting Teleoperation", *Proc. 1988 IEEE int. Conf. on Robotics and Automation*, pp. 584-589, 1988.
- [3] K. P. Chin and T. B. Sheridan, "The Effect of Force Feedback on Teleoperation", *MIT Man-Machine Systems Lab*, pp. 504-511, 1990.
- [4] D. A. McAfee and P. Fiorini, "Hand Controller Design Requirement and Performance Issues in Telerobotics", *Proc. 1991 IEEE int. Conf. on Robotics and Automation*, pp. 186-192, 1991.
- [5] S. K. Lee and H. S. Lee, "An Advanced Teleoperator Control System : Design and Evaluation", *Proc.*

- [6] 유 승렬, "원격 제어를 위한 Master Arm에 대한 연구", 석사 학위 논문, 포항공과대학, 1992.
- [7] 차 덕순, "힘 반영 원격 제어 시스템의 안정성에 관한 연구", 석사 학위 논문, 포항공과대학, 1993.

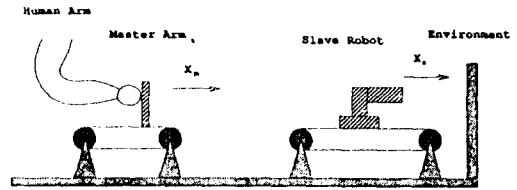


그림 4. 1 자유도 원격 제어 시스템

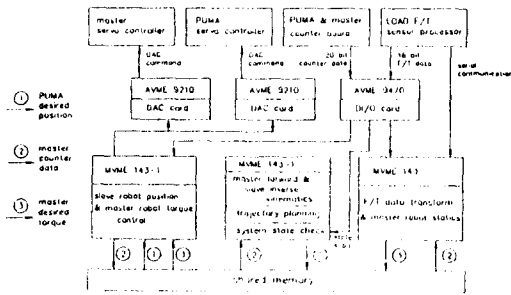


그림 1. 실시간 제어기의 구성

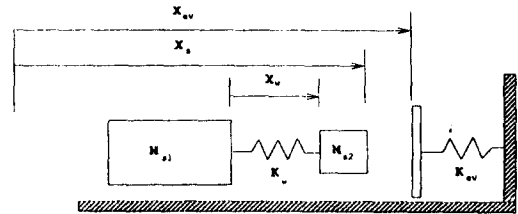


그림 5. Slave robot의 dynamic model

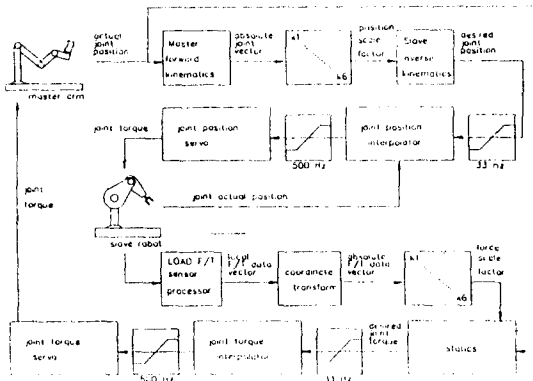


그림 2. 원격 제어 시스템의 control algorithm

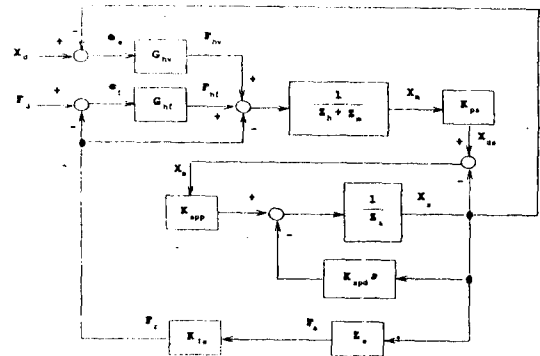


그림 6. 원격 제어 시스템의 전체적인 모델

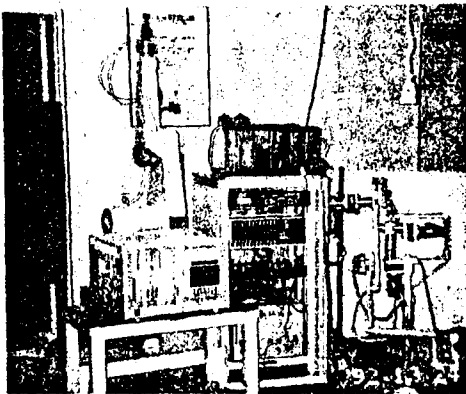
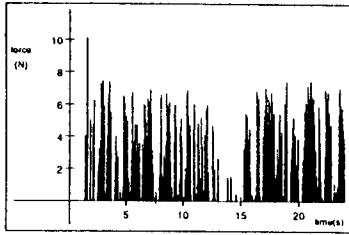
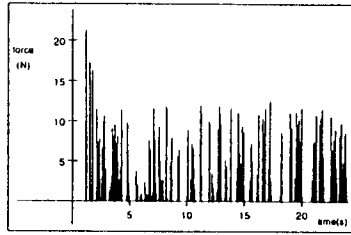


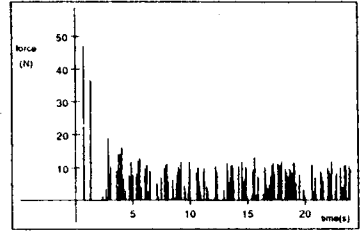
그림 3. 원격 제어 시스템의 layout



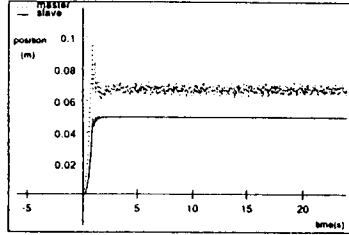
(a) 작업 물체와의 접촉 힘



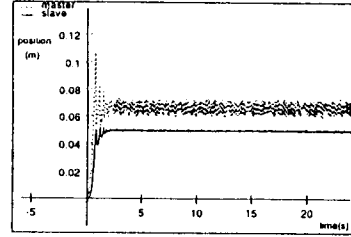
(a) 작업 물체와의 접촉 힘



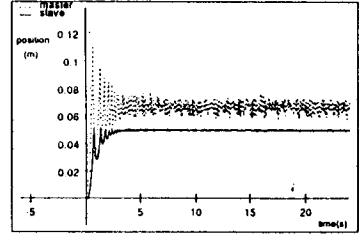
(a) 작업 물체와의 접촉 힘



(b) 원격 제어 로봇의 위치



(b) 원격 제어 로봇의 위치

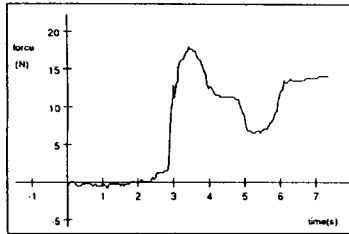


(b) 원격 제어 로봇의 위치

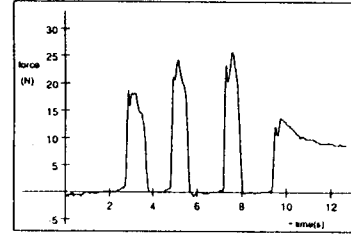
그림 7. Simulation when $K_{fs} = 0.2$

그림 9. Simulation when $K_{fs} = 0.3$

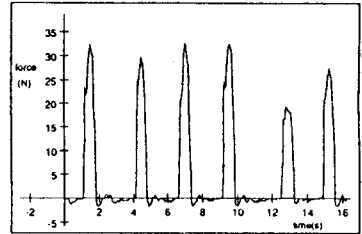
그림 11. Simulation when $K_{fs} = 0.4$



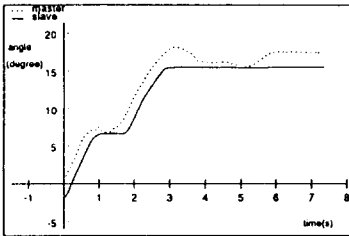
(a) 작업 물체와의 접촉 힘



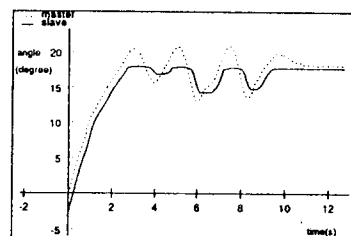
(a) 작업 물체와의 접촉 힘



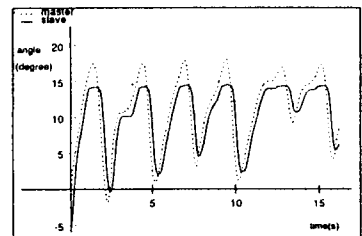
(a) 작업 물체와의 접촉 힘



(b) 원격 제어 로봇의 위치



(b) 원격 제어 로봇의 위치



(b) 원격 제어 로봇의 위치

그림 8. Experiment when $K_{fs} = 0.2$

그림 10. Experiment when $K_{fs} = 0.3$

그림 12. Experiment when $K_{fs} = 0.4$