

힘 제어 알고리즘을 이용한 응용 S/W 패키지의 개발
(The Development of Application S/W Packages using Force Control Algorithm)

○ 정 제 옥 고 명 삼 이 범 희
(J.O.Jeong M.S.Ko B.H.Lee)

서울대학교 공과대학 제어계측공학과
로보틱스 및 지능시스템 연구실

For the robot manipulator in performing precision tasks, it is indispensable that the robot utilize the various sensors for intelligence. In this paper, the hybrid position/force control method is implemented with a force/torque sensor, two personal computers, and a PUMA 560 manipulator. Two application S/W packages for edge following and peg-in-hole tasks are developed by the proposed force control algorithm. The related experimental results are then presented and discussed.

1. 서론

현대 산업사회, 특히 공장 자동화분야에 있어서의 로봇의 역할은 날이 갈수록 커지고 있다. 생산 자동화에 의한 생산성의 놀라운 향상뿐만 아니라 인간이 하기에는 위험이 따르는 일이 나 높은 정밀도가 요구되는 일, 지루한 작업들을 로봇이 대신함으로써 인해 더욱 정밀하고 지능을 갖춘 로봇의 개발이 요구되고 있다.

그러나, 현재까지 공장에서 사용되는 로봇의 대부분은 용접이나 칠 작업, 또는 물건을 집어 옮기는 단순한 작업에만 사용되고 있는 실정이다. 그 가장 큰 이유는 로봇이 주의의 상황 판단 및 거기에 대처할 능력을 갖추지 못했기 때문이다. 이러한 로봇은 좀더 복잡한 작업들, 예를 들면 윤곽 추적이나 폐인홀 작업 등은 오차로 인하여 제대로 수행할 수 없다.

이러한 로봇에 지능을 부여하기 위해서는 여러가지 센서를 이용한 감지장치를 지닌 로봇의 개발이 필수적이다. 많이 사용되는 센서 시스템으로는 비전 시스템, 원거리 센서 시스템, 접촉 센서, 힘-토크 센서 등이 있는데, 여기서는 힘-토크 센서를 사용하여 힘 제어 알고리즘을 로봇에 구현하고 그에 따른 여러가지 작업을 실험해본다.

힘-토크 센서를 이용한 응용 프로그램의 개발은 PUMA 560 매니퓰레이터에 IBM PC-AT, 386 PC를 이용하여 이루어진다. IBM PC-AT와 PUMA 로봇과의 연결은 그 동안의 연구[7]를 통하여 이루어져 있고, 거기에는 직선운동과 월드 모델링, kinematics 식등이 구현되어 있다. 사용되는 힘-토크 센서는 산업용 로봇에 사용되는 Lord사의 6축 출력력을 낼수 있는 센서로서 3방향의 힘과 3방향의 토크를 알아 낼 수 있다. 이 센서 시스템의 출력을 IBM-PC에서 받아들일 수 있도록 인터페이스 보드를 제작하였으며 이 인터페이스 보드를 386 PC에 연결하여 힘 제어 알고리즘을

위한 센서 값을 사용할 수 있게 한다.

힘 제어 프로그램은 386 PC에서 구현되며 386 PC는 IBM PC-AT와 common memory를 통하여 필요한 정보를 교환하게 된다. 응용 프로그램은 IBM PC-AT상에서 이미 완성되어 있는 여러 동작 알고리즘을 분석하고 힘 제어 프로그램에 맞게 고친다음, 그것을 바탕으로 새로이 작성하여 실험한다.

(1) 힘 제어 알고리즘

로봇 힘 제어는 작업 목적과 경로 계획(trajectory generation), 힘과 위치 피드백(force and position feedback), 환경에 따른 경로의 수정등을 모두 포함하고 있다. 힘 제어에 관하여 많은 연구가 이루어져 왔는데 그것을 로봇이 어떤 방법으로 감지한 힘을 이용하여 위치 명령을 수정하는가 또는 어떻게 로봇이 행할 작업과 요구되는 힘 제어 방법(force control strategy)을 연관시키느냐에 따라 분류해 보면[12] logic branching feedback방법과 continuous feedback방법으로 나눌 수 있다.

Logic branching방법은 접촉과 같은 discrete event에서 쓰여지는 방법으로 이것을 컴퓨터 알고리즘으로 보면 다음과 같이 나타낼 수 있다.

MOVE ALONG X-AXIS UNTIL FORCE EXCEEDS F
즉 로봇이 움직이는 도중 감지된 힘이 힘 명령보다 크면 멈추게 되는 방법으로 간단한 물체 검출과 작업전에 물체와의 접촉을 위하여 사용한다.

Continuous feedback방법은 6축 힘-토크 센서를 이용하여 continuous motion도중에 로봇 팔과 환경사이의 계속적인 접촉을 유지하도록 하는 방법으로 이 방법을 사용하면 로봇으로 하여

움 빼인홀이나 윤곽추적, 조립작업과 같이 로봇이 물체와 접촉해야만 하는 작업, 즉 움직임의 제한(motion constraint)이 있는 작업을 효과적으로 수행시킬 수 있다.

현재까지 연구된 힘 제어 시스템은 그림 1.1에 보인 구조로 대부분 설명될 수 있다[12]. 즉 로봇이 계획된 위치로 명령을 받고 있을 때 힘 되먹임 원칙(force feedback strategy)에 의하여 위치나 속도가 수정된다. 만일 환경과의 접촉이 일어나면 힘 센서를 통한 힘 되먹임(force feedback)이 생기게 되고 힘 되먹임 원칙(force feedback strategy)에 의하여 현재 위치나 경로를 수정하게 된다.

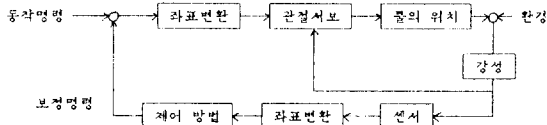


그림 1.1 힘 제어 시스템의 구조

위 구조에서 여러가지 변형이 생기게 되는데 가장 널리 알려진 것들로 stiffness control(Salisbury[14]), damping control(Paul, Shimano[11]), impedance control(Hogan[15]), hybrid position/force control(Raibert, Craig[2]) 등이 있다. 이 중에서도 hybrid position/force control 방식은 힘 제어속과 위치 제어속을 분리하여 명령을 줄 수 있어 여러가지 응용 범위가 넓기 때문에 본 연구에서도 이 방식을 이용하여 힘 제어 알고리즘을 구성한다.

2. 본론

(1) 전체 시스템의 개요

그림 2.1에 힘 제어를 위한 전체 시스템의 개요를 나타내었다.

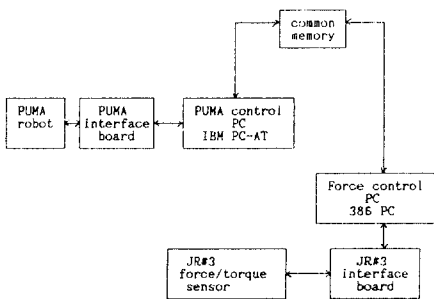


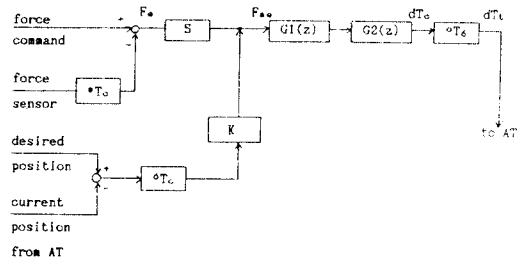
그림 2.1 전체 시스템의 개요

전체 시스템은 IBM PC-AT와 386 PC, 그리고 common memory 보드와 Lord 힘/토크 센서 인터페이스 보드 등으로 이루어져 있다. IBM PC-AT는 전체적인 동작을 담당하고 있으며 힘 제어가 필요할 때에는 386PC에 명령을 내려 386PC로 하여금 힘 제어를 실행시키고, 힘 제어의 결과를 다시

받아 PUMA로봇의 움직임을 조정한다. 이때 IBM PC-AT와 386PC는 힘 제어 알고리즘을 위하여 많은 양의 데이터 교환을 필요로 하므로 이를 위하여 common memory 보드를 사용하여 처리하고 있다. Lord 힘/토크 센서 인터페이스 보드에서는 Lord 힘/토크 센서 전자 박스의 병렬 포트의 값을 읽어 386PC에서 읽을 수 있도록 하고 있다.

(2) 힘 제어 알고리즘

힘 제어 알고리즘의 개요가 그림 2.2에 나와 있다. 본 연구에서 사용한 힘 제어 알고리즘은 복합 위치/힘 제어(hybrid position/force control) 방식으로 Maple과 Becker가 제시한 방식[1]의 변형이라고 할 수 있다. 이 방법은 사용자가 임의로 보정 운동이 행해지는 좌표계(compliant frame)를 지정해 줄 수 있고 힘 명령을 보정 좌표계에서 줄 수 있으며 PUMA 로봇의 위치 제어를 거의 개조 시키지 않고 쓸 수 있다는 점에서 PC상에서 힘 제어를 실현시키기에 가장 유리한 구조라 할 수 있다.



T_c : transform from tool frame to compliant frame
 T_t : transform from compliant frame to tool frame
 T_s : transform from sensor frame to compliant frame
 S : selection matrix
 K : stiffness matrix
 $G1, G2$: S/W compensator

그림 2.2 힘 제어 알고리즘

먼저 IBM PC-AT로부터 4x4 행렬로 이루어져 있는 현재 위치(X_p)와 목적 위치(X_d)를 읽는다. 이때의 4x4행렬은 월드 좌표계에서의 변환 행렬이고 6x1 벡터는 x, y, z 좌표에서의 이동값(translation term)과 변환각(rotation angle)을 나타내고 있다. 이 위치 값들을 식 2.1에 의해 보정 좌표계로 변환시킨 다음 위치 오차(E)를 식 2.2에 의하여 계산한다.

$$X_{dc} = X_d * T_c$$

$$X_{pe} = X_p * T_c \quad (2.1)$$

$$X_e = (X_{pe})^{-1} * X_{dc} \quad (2.2)$$

계산된 오차는 4x4 행렬이나, 감지된 힘과 명령 힘은 6x1 벡터이므로 4x4 행렬을 6x1 벡터로 바꿔줘야 한다. 6x1 벡터로 변환된 위치 오차 X_e 는 강성(stiffness) 행렬을 거쳐 예상 힘으로 계산된다. 이 강성값은 사용자가 임의로 정해줄 수 있는 값으로 예를 들면 오차가 10cm일 때

stiffness가 0.1N/cm 이라면 예상 힘은 $F = kx$ 라는 식에 따라 1N이 된다. 로봇트는 예상 힘을 감하는 방향으로 제어되기 때문에 만일 강성이 0이라면 로봇트는 최대한의 유연성(compliance)을 갖고 작업하게 된다. 즉 강성과 유연성을 서로 반대되는 작용을 하게 되며, 강성이 크면 불순속 로봇트의 움직임은 예정된 경로를 따라가려 하므로 deburring, grinding등의 작업에 적합하며, 강성이 작을수록 로봇트는 보정 동작을 민감하게 하므로 찍인홀이나 윤곽 추적등에 적합하다.

한편 외부 루프에서는 명령 힘과 감지된 힘의 오차를 계산하여야 하는데 이때 감지된 힘(F_s)은 센서 좌표계에서의 값이므로 이 값을 보정 좌표계로 변환시켜 주게 된다. 이것은

$$F_s = F_s * T_s \quad (2.3)$$

에 의하여 계산된다. 힘 오차 F_o 는

$$F_o = F_d - F_s \quad (2.4)$$

로 계산되어 선택(selection)행렬을 통과함으로써 특정 축만 힘 제어를 할 수 있게 된다. 선택 행렬은 0과 1값만을 가지는 행렬로 힘 제어를 하는 축은 1, 위치 제어를 하는 축은 0이 된다.

예상 힘과 힘 오차를 더하여 실제 힘 오차(actual force error, F_{ao})를 계산하고 그 값을 제어기(compensator)로 들어가게 된다.

$$F_{ao} = F_o * S + K * X_o \quad (2.5)$$

제어기는 $(z-b)/(z-a)$ 로 표시되는 software 제어기이다. 이때 제어기는 힘 제어 루프의 차수를 결정하게 되는데 제어기의 이득(gain)이 크면 로봇트는 작은 힘에도 민감한 반응을 나타내게 된다.

사용할 제어기의 식은 $z/(z-1)$ 로 적분기(integrator)로써 사용되고 있으며, 이 제어기가 위치를 적분함에 따라 로봇트는 가해진 힘에 비례하는 속도로 움직이게 된다. 제어기를 이중으로 쓰게 되면 로봇트의 위치가 아닌 속도를 적분함에 따라 로봇트는 가해진 힘에 비례하는 가속도로 움직이게 된다. 이 구조는 사람이 손으로 로봇트를 교시(teach)하는등의 빠른 응답을 필요로 할때에 유용하다.

Compensator를 거쳐 계산된 힘 오차는 6x1 벡터이므로 이 값을 다시 4x4행렬(A_{tc})로 바꿔주므로 식 2.6에 의하여 robot의 물 좌표계로 변환시킨다음 이 값을 IBM PC-AT에 위치 보정 명령(A_{t6})으로 보내 주게 된다.

$$A_{tc} = T_s * A_{tc} * T_s^{-1} \quad (2.6)$$

(4) 힘 제어 알고리즘의 응용 및 실험 결과

힘 제어 응용 프로그램을 위하여 가장 기본적으로 제공되어야 하는 것이 앞의 1.1장에서 얘기한 continuous feedback방법과 logic branching방법의 실험을 위한 move - until과

move - comply이다.

로봇트가 움직이는 도중 물체에 부딪히거나 물체를 접촉해내면 멈추어야 하는데, 이때 쓰여지는 것이 move - until이다. 이 함수는 사용자가 정한 힘 제어축과 힘 명령을 받아 움직이다가 감지된 힘이 힘 명령보다 커지면 멈추게 된다.

로봇트가 물체와 접촉을 한 이후에 여러가지 작업을 하기 위해서는 힘 센서의 계속적인 피백임(continuous force feedback)을 받으면서 이에 대한 보정 동작, 즉 원래의 계산된 경로를 따라 움직이면서 접촉에 의한 힘에 따라 경로변경을 하게 되는데, 이것을 위한 함수가 move - comply이다. 현재 작성된 move-comply는 사용자가 정한 힘 제어 축과 힘 명령 및 보정 좌표계의 위치를 받아 미리 계산된 경로를 따라 움직이면서 보정좌표계의 힘 제어축의 힘을 힘 명령에 따라 조정하게 된다.

위의 명령들을 실험하기 위한 구성을 그림 2.3에 나타내었다.

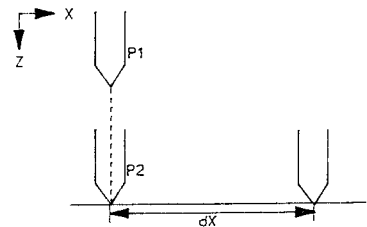
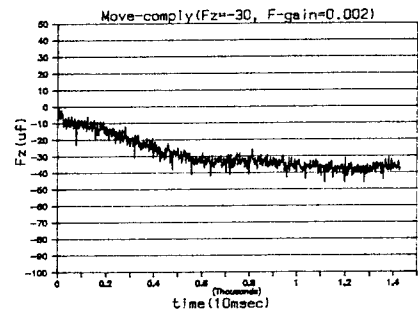
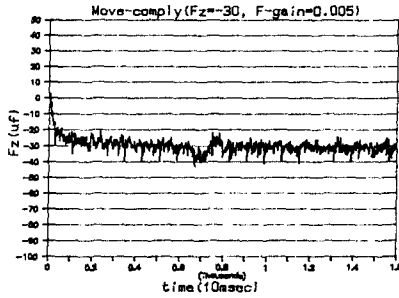


그림 2.3 실험 구성

먼저 Robot을 물체 표면의 준비위치(P_1)에 이동시켜 놓고 move - until을 Z축에 대하여 수행한다. 그 다음 robot의 현재 위치(P_2)에서 X축으로 dx만큼의 위치까지 경로를 계산하고 힘 제어축을 Z축으로, 힘 명령을 -30uf(unit force)로 정한 다음 move - comply를 수행하였다. 그에 따른 실험 결과를 아래 그림에 보았다. 이때 그림의 X축은 단위 1이 힘 제어 루프를 한번 실행하는 시간으로 실제로는 약 10msec정도이며 Y축은 감지된 힘의 양으로 단위 1uf당 약 0.03Kg-f이다.





위의 실험 결과에서 보면 Move - comply를 실행 도중 robot가 물체와 접촉하면서 움직이는 도중 물체 표면에 힘 명령에 따른 일정한 힘을 가하는 것을 볼 수 있다. 특히 힘 제어 루프 gain이 커짐에 따라 더욱 빨리 힘 명령을 따라가는 것을 볼 수 있다.

(4) 윤곽 추적

로보트를 이용한 임의의 물체의 윤곽 추적 문제는 Starr[3], M.H.Choi[6], S.H.Lee[14] 등의 연구가 있어 왔다. 그러나 M.H.Choi의 연구를 제외한 윤곽 추적 연구는 완전한 물체의 형상을 알아내지는 못하였다. M.H.Choi는 힘 제어 알고리즘의 구현에 있어서 VAL언어를 이용하였기 때문에 힘 제어 알고리즘의 실시간 처리가 불가능하였다.

윤곽 추적의 목적은 임의의 물체의 형태를 알아내고, 또 추적도중 물체에 일정한 힘을 가하는 것이 목적이다. 이를 위하여 386PC의 힘 제어 알고리즘을 약간 고쳐 윤곽 추적 알고리즘으로 이용한다.

윤곽 추적시에는 추적도중 도구(tool)의 방향(orientation)이 일정하게 유지될수도 있고 물체 표면에 일정한 방향을 유지하기 위해 도구(tool)를 돌릴수도 있다. 본 논문에서는 전자의 방법을 사용하였으며 이때 물체 표면에 일정한 힘을 가하기 위해 가상의 보정 좌표계(compliant frame)를 지정하여 보정 좌표계가 물체 표면의 변화에 따라 방향이 바뀌도록 하고 있다. 이 좌표계에서 물체 표면에 접한 방향으로 위치 제어를 하고, 직각 방향으로 힘 제어를 함으로써, 물체 표면에 가해지는 힘과 추적 속도를 일정하게 유지할수 있다. 다음 그림에 윤곽 추적의 알고리즘을 나타내었다.

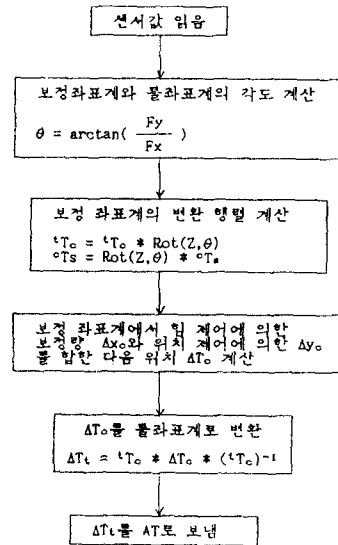
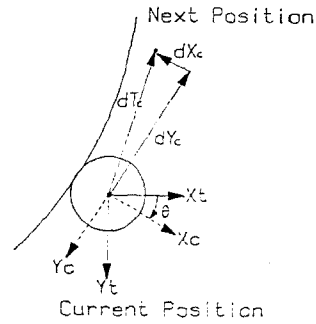


그림 2.6 윤곽 추적 알고리즘

위의 실험결과를 아래에 보았다. 추적 물체의 모양은 그림 2.7과 같고 시작점(P1)에서 시작하여 물체의 표면을 따라가다가 다시 시작위치에 오게 되면 추적을 끝낸다. 그림 2.8은 물체 추적도중에 보정 좌표계에서의 Fx를 나타내고 있다. 이때의 힘 명령은 30uf로 추적 도중의 힘 응답은 30±10 uf로 실제 오차가 30gf-cm를 넘지 않음을 알 수 있다.

그림 2.9는 물체 추적도중의 보정 좌표계와 물 좌표계사이의 각도를 나타내고 있으며, 쓰여진 번호는 그림의 같은 번호의 위치에서의 값이다. 이를 보면 물체 표면의 변화에 따라 보정 좌표계가 적절히 변화되어 가는 것을 볼 수 있다.

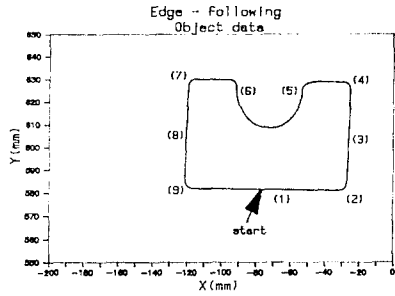


그림 2.7 추적 물체의 윤곽

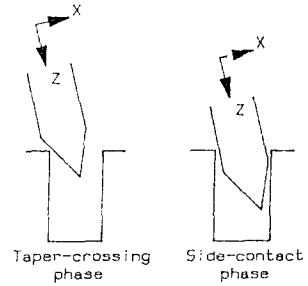


그림 2.10 백인홀 작업 구성

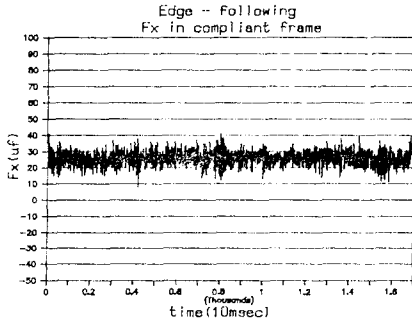


그림 2.8 보정 좌표계의 Fx

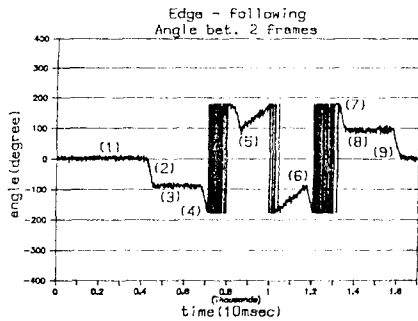


그림 2.9 보정 좌표계의 변환각

(5) 백인홀

백인홀 작업은 힘 제어의 응용으로 가장 많이 이용되는 예라 할 수 있겠다. 로봇의 위치 제어만을 이용하여 백인홀을 하는 경우에는 로봇 자체의 오차와 위치 계산 오차등으로 인하여 올바르게 작업을 할 수 없다. 이 오차를 힘 제어를 이용하여 로봇의 동작을 직접히 보정함으로써 백인홀 작업을 끝낼수 있다. 그림 2.10 에는 본 논문에서 실험한 백인홀 작업의 구성을 나타내었는데, 백은 물이 상륜게 좁아있으며(round tapered) 백과홀의 직경은 약 1mm의 차이를 갖고 있다.

보정 좌표계는 백의 물에 설정하고 백의 나아가는 방향은 Z축으로 한다. 이때 위치 제어 축은 Z축이 되고 힘 제어 축은 X축을 제외한 모든 축으로 한다. 백인홀 작업시에는 힘에 의한 위치 제어뿐만 아니라 토크에 의한 각의 보정이 이루어져야 하지만, 각 오차는 위치 오차보다 매우 작기 때문에 힘 제어 루프에서의 토크 이득은 힘 이득의 약 0.01배이하로 주어야 한다. 백을 홀의 위에 접근시킨다음 삽입을 시작하면 taper-crossing상태가 시작된다. 이때 중요한 것은 백이 홀의 중심으로 나아가는 것이지만 힘 제어 알고리즘은 토크제어를 하고 있기 때문에 각 오차가 없어도 토크에 의한 각 보정을 할 수 있다. 그러나 taper-crossing상태에서 side-contact상태로 넘어가게 되면 taper-crossing상태에서 생겼던 각 오차를 없앨 수 있게 된다. 그림 2.11는 백인홀 작업의 실험 결과이다.

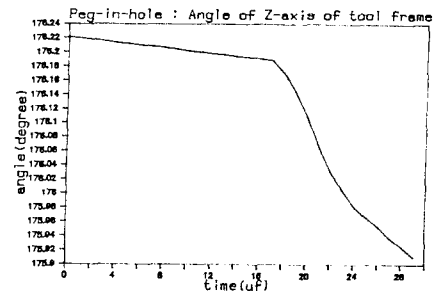
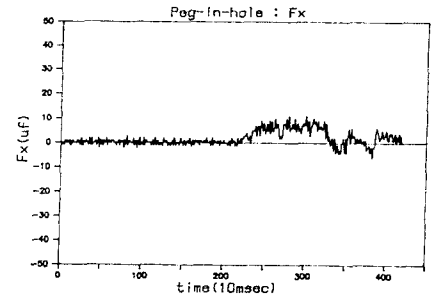


그림 2.11 백인홀 실험 결과

실험은 X축방향으로 약 3mm정도의 위치 오차와 5도정도의 각 오차를 두고 하였다. 첫번째 그림은 Fx의 변화를 나타내고 있는데, 백이 홀에

접촉하면서 생기는 힘을 위치 보정을 함으로써 상쇄해나가는 것을 볼 수 있다. 두번째 그림은 물체의 Z축과 월드 좌표계의 Z축사이의 각도를 나타내고 있으며, 꺾이 줄이 삽입되면서 각 오차를 보정해나가는 것을 볼 수 있다.

3. 결 론

현재 산업용 로봇은 그 응용에 있어서 미미 프로그램 되어 있거나 사람의 가르침에 의한 매우 단순한 작업들만 처리하고 있다. 그것은 센서를 이용한 로봇의 환경에 대한 반응을 할수 있는 능력이 주어지지 않았기 때문이며 이는 로봇의 다양한 응용을 불가능하게 하고 있다.

본 연구에서는 힘/토크 센서를 사용한 힘 제어 알고리즘을 IBM PC와 PUMA로봇에 구현하고 힘 제어 응용의 기본적인 프로그램을 작성 및 실험하였다. 이 프로그램들을 조합하면 여러가지 다양한 응용을 실현할 수 있을 것으로 보이나 실제로는 작업마다 그 구성이 틀리므로 실행할 작업에 따라 프로그램을 고쳐주어야 한다는 단점이 있다. 그러나 이 연구의 결과를 이용하여 여러 작업에 힘 제어를 쉽게 이용할 수 있도록 응용 프로그램들을 작성하면 많은 작업에 힘 제어를 이용하여 보다 정밀한 작업을 할 수 있게 될 것이다.

결론적으로 본 연구에서 다른 힘-토크 센서를 이용한 힘 제어 실현과 그 응용 프로그램의 작성 및 실험은 로봇의 위치오차와 작업 환경의 부정확성을 로봇으로 하여금 자체적으로 극복할 수 있도록 함으로써 물체의 검출 및 그 밖의 정밀을 요하는 복잡한 작업들을 수행할 수 있게 될 것이다.

4. 참고 문헌

[1] James A. Maples & Joseph J. Becker, "Experiments In Force Control Of Robotic Manipulator," IEEE Int. Conf. on Robotics and Automation.

[2] Raibert & Craig, "Hybrid Position/Force Control Of Manipulators," Tran. ASME, Vol 102, June 1981

[3] Gregory P. Starr, "Edge Following With A PUMA 560 Manipulator Using VAL-II," 1986 IEEE Int. Conf. on Robotics and Automation.

[4] Jin S. Lee, "A Shared Position/Force Control Methodology For Teleoperation," RCA corporation, Advanced Technology Laboratories.

[5] H. Van Brussel & J. Simons, "Automatic Assembly By Active Feedback Accomodation,"

[6] M. H. Choi, "A Study On The Two Dimensional Object Contour Tracking With

A Sensor Equipped Manipulator", M.S.Thesis, Dept of Control & Instrumentation, Seoul Nat'l Univ., 1988.

[7] 이 원식, "컨베이어 벨트를 포함하는 로봇작업 시스템의 구성에 관한 연구", 서울대학교 제어계측과 석사 학위 논문, 1989.

[8] H.van Brussel, J. Simons, & J. Deschutter, "An Intelligent Force Controlled Robot," Annals of the CIRP, Vol.31, Jan. 1982.

[9] B. E. Shimano, C. C. Geschke, C. H. Spalding III, R. Goldman, D. W. Scarborough, "AIM: a task level control system for assembly," Proc. 8th Int. Conf. Assembly Automation, March 1987.

[10] Lozano-Perez T., "Robot Programming," Proc. IEEE, Vol. 17, July 1983.

[11] R. P. Paul & B. E. Shimano, "Compliance & Control," Proc of Joint Automatic Control Conf., pp.694-699, 1976.

[12] D.E.Whitney, "Force Feedback Of Manipulator Fine Motions", Joint Automatic Control Conference, San Francisco, 1976.

[13] J. De Schutter, "Improved Force Control Laws For Advanced Tracking Applications," IEEE Int. Conf. on Robotics and Automation, 1988.

[14] J. K. Salisbury, "Active Stiffness Control Of A Manipulator In Cartesian Coordinates", Proc. 19th IEEE Conf. on Decision and Contr, Dec 1980.

[15] N. Hogan, "Control Of Mechanical Impedance Of Prosthetic Arms", Proc. JACC, 1980.

[16] R. P. Paul, "Manipulator Cartesian Path Control", IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-9, No.11, Nov 1979.

[17] C. Reboulet, A. Robert, H. Poilve, & A. Gallet, "Hybrid Position Force Control - Application to Assembly", 15th ISIR.

[18] H. Asada & K. Ogawa, "On The Dynamic Analysis Of A Manipulator And Its End Effector Interacting With The Environment", IEEE Int. Conf. on Robotics and Automation, 1987.

[19] S. Lee & J. M. Lee, "The Control Of Surface Contact and Slide Using Wrist Force/Torque Sensor", Int. Symposium on Robotics and Automation, 1987.

[20] H. Kazerooni & J. I. Baecker, "Hybrid Force/Position Control In Robotic Deburring", Conf. on Applied Motion Control, 1987.