한국공작기계학회 2000년도 춘계학술대회 논문집 pp.679~684

병렬기구 공작기계의 프로그램 개발

Development of a CNC Machine using a Parallel Mechanism

박근우(창원대 대학원), 김태성(창원대 대학원), 최병오(KIMM), 이민기(창원대 공대)
Kun-Woo Park(Graduate School, Changwon National Univ.),
Tae-Sung Kim(Graduate School, Changwon National Univ.),
Buyng-Oh Choi(KIMM),
Min-Ki Lee(Changwon National Univ.)

Abstract

This paper presents the development of system and program for a Parallel-Typed CNC Machine. The system consists of parallel manipulator, PC (Personal Computer), DMC (DSP Motion Controller), and machining tools. In order to control the manipulator, the program, which is implemented in "C/C++" language, involves inverse/direct kinematics, velocity mapping, Jacobian and etc. A controller computes the kinematic formulation in real-time and generates the motion by the DMC. A monitor, which has access to program and sensory information, displays the status of manipulator.

Key Words : Parallel-Typed CNC Machine(공작기계), DMC (운동제어 콘트롤러), Linear Actuator (직선 운동 액추에이터), Velocity Mapping (속도변환), Trajectory Control (경로제어)

1. 서론

기계 가공 작업을 위한 병렬기구형 공작기계가 설계·제작되었다[1]. 본 논문에서는 병렬기구형 공작기계의 시스템을 구성하고 구동 프로그램을 개발한다. 기계 가공작업은 일반적으로 NC 공작기계에 의해 행해지는데 이들에 대한 시스템 구성과 언어 개발에 대해서는 많은 연구가 진행되고있다. 그러나 이러한 공작기계는 대부분 가공물에 대해 평면적인 작업을 수행하므로 직선과 곡선을 복합적으로 가공하기에는 거의 불가능하다. 이를 해결하기 위해서 극히 일부의 기계 가공작업에 로봇이 적용되고[2,3] 연구도 미비한 실정이다. 위의 경우는 다 관절 로봇을 연삭에 적용한 사례로 가공시 변형를 줄이고 안정성을 높이기 위한 제어 시스템 개발에 주안점을 두고 있다.

다관절 로보트는 낮은 강성으로 큰 부하를 낼 수 없어 적용분야가 한정되어 있다. 요즘 강성이 큰 병렬기구형 로봇을 여러 분야에 적용하고 있으나[4,5] 구동방식이 다 관절과 다르므로 새로운 개념의 시스템 구성과 언어 개발에 대한 연구가 진행되어야 한다.

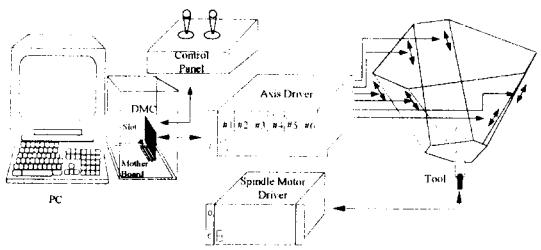
지금까지의 로봇 구동 프로그램은 다 관절 로봇에 대한 것이었고 대부분 전용 컨트롤러에 전용 언어로 되어 있어 자체가 가지고 있는 센서들에 의해서만 정보를 받고 외부 센서를 이용할 경우 시스템 구성에 많은 어려움이 있다. 또한 프로그램이 가지고 있는 기능 이외의 기능을 사용자가 개발하는데 한계가 많다. 고도화된 수학적 모델로 머니퓰레이터의 세부적인 운동을 사용자가 정해 주도록 한 기호적인 머니퓰레이터 언어인 AL [6]과 WAVE [7]등이 개발되었으나 사용자가 운동에 대한 상세한 사항을 알아야 했다. 이점을 보완한 것으로 IBM에서

AUTOPASS [8]를 개발했는데 로봇의 운동보다는 대상물의 조립 기능에 초점을 맞춘 인공지능을 가진 언어로 대형 컴퓨터에서 운영되었다. 또한, 서보 제어 과정에서 센서를 통해 궤환되는 정보를 사용자가 직접 지정할 수 있도록 함으로써 로봇의 프로그램과 제어를 편리하게 할 수 있는 시스템 등 [9]이 제안되었다.

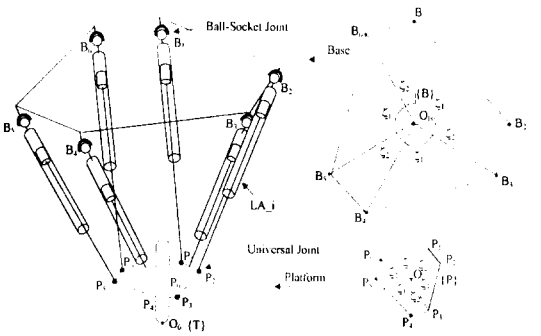
본 연구에서는 외부 센서들과 원활한 통신과 하드웨어 확장을 위해 PC와 DSP Motion Controller (DMC)로 구성한다. "C/C++"-언어를 사용함으로써 다양한 함수를 개발하고, 모든 프로그램은 사용자의 편의를 위해서 그래픽 사용자 인터페이스 방식으로 실행한다.

2. 병렬기구형 공작기계 시스템 구성

병렬기구형 공작기계 시스템은 Fig.1과 같이 병렬기구 머니퓰레이터와 이를 제어하는 6개의 Axis Drivers, 6축 동시제어 DMC, Control Pannel 및 PC 등으로 구성되어 있다.



(a)



(b)

Fig. 1 (a) A system for the Parallel-Typed CNC Machine, (b) a Parallel Manipulator

병렬기구의 구조는 Fig. 1(b)와 같이 semi-regular planar hexagon의 형태인 베이스(Base)와 플랫폼(Platform) 사이에 6개의 리니어 액추에이터가 설치되어 있다. 설치 위치는 베이스와 플랫폼 각각에 ζ_1=90°, ζ_2=30°로 설치되어 있다. 리니어 액추에이터의 베이스 부 연결은 볼-소켓(ball-socket) 조인트로, 플랫폼 부는 유니버설 조인트로 각각 연결되어 있다. 리니어 액추에이터의 길이를 늘였다 줄였다함으로써 위치와 자세를 동시에 제어한다.

PC는 사용자가 개발한 모든 수학적 연산을 실시하며 키보드에 의해서 머니퓰레이터를 교시한다. 연산결과 및 키보드 명령은 Data Bus를 통해서 DMC에 제공하며 DMC는 Axis Driver, 입출력 I/O, 엔코더 등과 연결되어 머니퓰레이터를 구동한다. 구동 명령은 DMC에 의해 해석 가능한 "C"-언어 함수로 위치, 속도, 궤적 등을 제어하는 함수, 디지털 및 아날로그 센서의 입출력을 처리하는 함수, 시스템을 관리하는 함수 등으로 되어있다. 따라서 사용자가 이들 함수를 이용하여 머니퓰레이터 프로그램을 개발할 수 있다.

각 Axis Driver는 DMC에서 +/- 10V사이의 아날로그 신호를 받아 모터를 구동할 수 있는 전원을 공급한다. 액추에이터는 DC 서보 모터에 의해서 구동되는데 리니어 액추에이터인 경우, 풀리로 1/2로 감속하고, lead가 5mm인 Ball Screw에 의해서 직선운동을 생성한다. 구동 모터는 최고 속도 80m/min, 출력 6.0 KW를 만족하도록 결정했다. 전체 시스템의 관리와 입/출력 처리를 위해 Control Panel과 PLC 등이 추가된다.

3. 병렬기구 머니퓰레이터의 구동 프로그램

모니터 명령에서 공작기계의 위치와 방위를 나타내는 데이터는 아래와 같이 선언된다.

P_i = {{X, Y, Z}, {Yaw, Pitch, Roll}}

여기서 P_i는 데이터의 변수이고, P_T={X, Y, Z}는 툴 끝단의 위치, R_T = {Yaw, Pitch, Roll}은

툴에 부여된 직각좌표의 X, Y, Z축이 기준 좌표축과 이루는 각도이다. 이러한 데이타 선언은 파일에 기록되는데 역방향 기구학 해석을 이용하여 리니어 액추에이터(LA_i, i=1,2,...,6)의 길이로 환산한다. 병렬기구의 역방향 기구학 해석을 위해 Base 및 Platform의 중심점 O_B와 O_P에 직각 좌표 {B}와 {P}를 각각 부여하고, Spindle 끝단, O_6에 Tool 좌표 {T}를 부여한다.

툴의 위치 ($\overrightarrow{O_BO_6}$)와 방위(R_T)가 주어질 때, 폐 루프 $O_BO_6P_iB_i$ 의 관계에 의해서 점 B_i ($i=1,\cdots,6$)에서 P_i ($i=1,\cdots,6$)까지의 벡터 $\overrightarrow{B_iP_i}$ 는

$$\overrightarrow{B_iP_i} = \overrightarrow{O_BO_6} + R_T\ {}^T\overrightarrow{O_6P_i} - \overrightarrow{O_BB_i} \qquad (1)$$

이다. 여기서 R_T 는 좌표 {B}에대한 Tool좌표 {T}의 방위를 나타내는 회전행렬로 첨자는 벡터와 회전행렬을 나타내는 좌표이다. 따라서 리니어 액추에이터의 길이는

$$\| \overrightarrow{B_iP_i} \| = \| \overrightarrow{O_BO_6} + R_T\ \overrightarrow{{}^TO_6P_i} - \overrightarrow{O_BB_i} \| \qquad (2)$$

가 된다.

기구학 해석은 리니어 액추에이터의 길이뿐만 아니라 조인트의 각도가 최대, 최소 한계 값의 범위 안에 있는지를 판단하고 만족할 때만 정상적인 해석을 끝낸다.

기구학 해석에 의한 길이 및 각도를 모터의 펄스로 환산하고 궤적생성 함수에 축, 가속도, 속도, 위치 등의 변수를 지정한다. 생성된 데이터는 DMC에서 모터를 구동하는데 필요한 data frame으로 변환되고 DMC 외부 메모리에 적재된다. DMC는 PC에서 Motion Start 명령이 실행되면 모터를 구동시킨다. 이때 DMC는 내부 loop를 통해서 연속 운동의 마지막 위치임을 알리는 명령까지 계속한다.

속도 제어를 위해서는 툴의 속도가 주어질 때 각 리니어 액추에이터의 속도를 구해야 한다. 툴의 속도는 자코비안 행렬에 의해 리니어 액추에이터의 속도로 아래와 같이 변환된다.

$$\begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \\ \vdots \\ \dot{l}_6 \end{bmatrix} = J \begin{bmatrix} V_T \\ W_T \end{bmatrix} \qquad (3)$$

여기서 $[\ \dot{l}_1\ \ \dot{l}_2\ \cdots\ \dot{l}_6\ ]^T$는 리니어 액추에이터의 속도, $[\ V_T\ \ W_T\ ]^T$는 툴의 속도이고 J는 자코비안 행렬이다. 자코비안 행렬은 screw 원리[10]에 의해서

$$J = \begin{bmatrix} \overrightarrow{U_{1_x}} & \overrightarrow{U_{2_x}} & \cdots & \overrightarrow{U_{6_x}} \\ \overrightarrow{U_{1_y}} & \overrightarrow{U_{2_y}} & \cdots & \overrightarrow{U_{6_y}} \\ \overrightarrow{U_{1_z}} & \overrightarrow{U_{2_z}} & \cdots & \overrightarrow{U_{6_z}} \\ (\overrightarrow{O_6P_1}\times\overrightarrow{U_1})_x & (\overrightarrow{O_6P_2}\times\overrightarrow{U_2})_x & \cdots & (\overrightarrow{O_6P_6}\times\overrightarrow{U_6})_x \\ (\overrightarrow{O_6P_1}\times\overrightarrow{U_1})_y & (\overrightarrow{O_6P_2}\times\overrightarrow{U_2})_y & \cdots & (\overrightarrow{O_6P_6}\times\overrightarrow{U_6})_y \\ (\overrightarrow{O_6P_1}\times\overrightarrow{U_1})_z & (\overrightarrow{O_6P_2}\times\overrightarrow{U_2})_z & \cdots & (\overrightarrow{O_6P_6}\times\overrightarrow{U_6})_z \end{bmatrix}^T \qquad (4)$$

이다. 위에서 유도된 자코비안으로 식(3)과 같이 툴 끝단의 속도를 리니어 액추에이터의 속도로 변환시켜 서보 모타가 생성한다.

속도 변환을 위해 리니어 액추에이터의 방향을 알아야 한다. 그러나 속도만 제어하므로 머니퓰레이터의 현 위치/방위에 대한 정보는 알 수 없다. 이 정보를 얻기 위해서는 각 축의 위치에 따른 병렬기구의 위치/방위를 구하는 순방향 해석이 프로그램에 포함되어야 한다.

순방향 기구학 해석[11]은 각 리니어 액추에이터의 길이가 주어질 때 끝단의 위치와 자세를 구하는 것으로 병렬기구의 경우 비 선형 연립 방정식으로 다중 근을 가진다[12]. 이것은 기하학적 구속조건(Geometric Constraint, GC)에 의해서 결정되는데 식(2)에 의해서 아래와 같이 6개의 구속식을 얻을 수 있다.

$$\| \overrightarrow{O_BO_6} + R_T\ \overrightarrow{{}^TO_6P_i} - \overrightarrow{O_BB_i} - \overrightarrow{B_iP_i} \| = 0 \qquad (i=1,2,..,6) \qquad (5)$$

여기서 각각 $\overrightarrow{O_BO_6}$는 x, y, z의 함수, R_T은 플랫폼의 자세를 나타내는 yaw, pitch, roll의 함수

이고, $\overrightarrow{O_6P_i}$, $\overrightarrow{O_BB_i}$는 설계 고정 값, $\|\overrightarrow{B_iP_i}\|$는 리니어 액추에이터의 길이이다. 구속식의 근은 뉴턴의 연산법을 적용하여 구한다.

구속식 (5)를

$$F_1=[f_1,f_2,f_3,f_4,f_5,f_6]^T$$

로 정의하고,

$$X=[x,y,z,\ yaw,pitch,roll]^T$$

로 정의할 때

$$X^{k+1}=X^k-\left[\frac{\partial F(X^k)}{\partial X}\right]^{-1}F(X^k) \qquad (6)$$

여기서 $[\partial F(X^k)/\partial X]^{-1}$ 은 역 자코비안 행렬로 많은 계산을 요한다. 그러나 초기값 $X^0=[x^0,y^0,z^0,yaw^0,pitch^0,roll^0]^T$ 를 이전 단계의 위치·자세 값으로 사용함으로써 이들이 근에 가까우므로 $[\partial F(X^k)/\partial X]^{-1}$ 은 수렴시간을 줄일 수 있다. 위 식에 의해서 근의 오차가 값 10^{-3}이내가 되었을 때 계산을 완료되면 리니어 액추에이터의 길이에 따른 툴의 위치와 방위를 구해진다.

4. 그래픽 사용자 인터페이스 프로그램

병렬기구형 공작기계는 복합형상을 가공하기 때문에 기존의 공작기계 전용 제어기로는 제어하기 힘들다. 따라서 PC-Based 제어기를 채택하여 위치·속도 제어 프로그램 및 사용자 인터페이스 프로그램 등을 GUI(Graphic User Interface)방식으로 개발하였다. 특히 기존의 공작기계 프로그램은 각각의 G코드의 기능을 암기하여 프로그램을 하여야 한다. 본 연구에서는 이러한 것을 해결하기 위해서 Fig. 2와 같이 HMI(Human Machine Interface)방식으로 개발되었다. 이 방식은 모든 명령어들의 다이얼로그형으로 되어 있어 처음 공작기계를 접하는 사용자가 학습하기 쉽다.

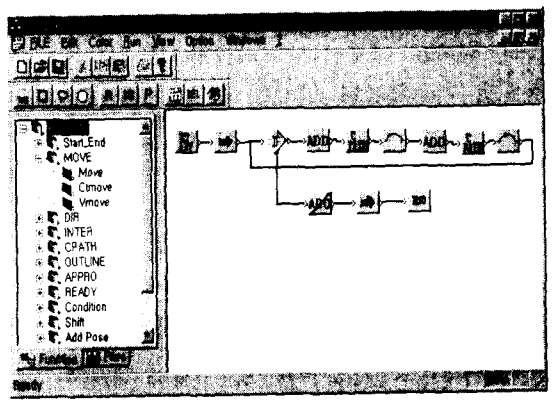


Fig. 2 Graphic User Interface Program

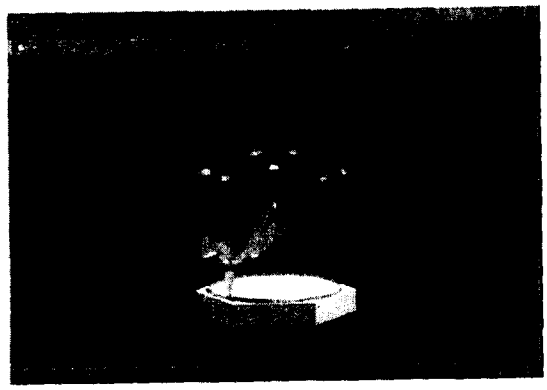


Fig. 3 3D Graphic Simulation Program

프로그램은 Fig. 2와 같이 일반적인 윈도우 응용프로그램과 비슷한 구성을 가진다. 가공 프로그램의 작성방법은 왼쪽 윈도우창에 나타나는 명령들을 선택한 하고 마우스, 키보드 등의 입력장치로 오른쪽 에디터 윈도우창에 배열한다. 배열된 명령들은 그림에서와 같이 아이콘의 형태로 나타나는데 이들을 흐름도를 그리듯 서로 연결하여 작업순서를 결정한다. 명령의 종류에는 G코드에서 제공하는 직선가공, 원호가공, 조건 등의 모든 명령에 구 가공 등의 복합형상의 가공을 위한 명령이 추가 되었다. 가공프로그램은 파일로 저장되고, 읽기 명령으로 재 사용이 가능하다.

또, 개발된 병렬기구형 공작기계는 일반공작기계와는 달리 가공작업 경로가 작업공간 내에 있는지를 직감적으로 알 수가 없다. 따라서 실제 가공작업을 수행하기 전에 Fig. 3과 같이 개발된 그래픽 시뮬레이션 프로그램을 이용하여 작업이

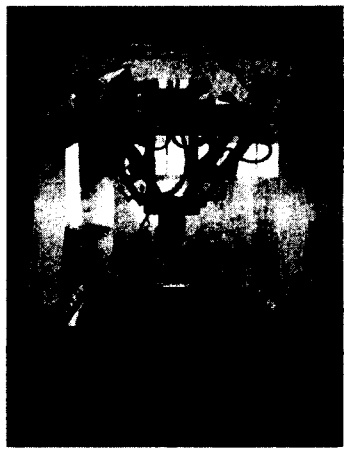
가능한지를 판단한다.

5. 연구결과

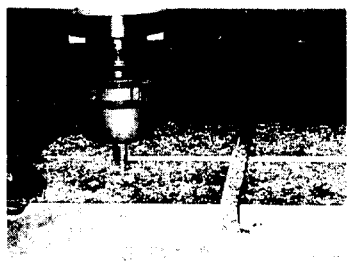
Fig. 3은 실제 제작된 병렬기구형 공작기계와 가공작업의 사진이다. 이 공작기계에 개발된 사용자 인터페이스 프로그램을 이용하여 대표적인 3차원 형상인 반구를 가공하였다.

반구를 가공하기 위한 작업순서는 다음과 같다.

1) 사용자 인터페이스 프로그램을 이용하여 구 가공 프로그램을 만든다.
2) 3차원 그래픽 시뮬레이션 프로그램을 이용하여 가공 프로그램이 작업공간 내에서 경로를 생성하는지를 확인한다.
3) 작업공간 내에서 작업이 가능하면 사용자 인터페이스 프로그램으로 가공 프로그램을 실행하여
4) 가공작업을 수행한다.

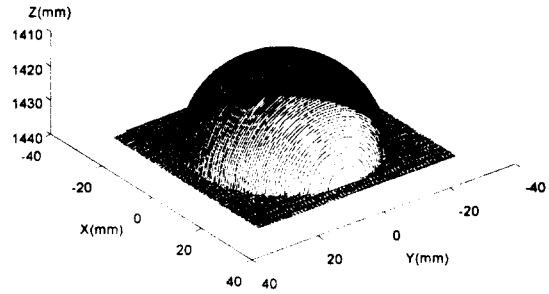


(a)



(b)

Fig. 3 (a) The constructed Parallel-Type CNC machine, (b) machining work



(a)

(b)

Fig. 4 (a) Data of spherical shape, (b) the manufactured work piece

가공을 편리하게 하기 위해서 구 표면의 경로를 자동으로 생성하는 함수를 개발하였다. Fig. 4(a)는 이 함수를 이용하여 생성된 구 가공 경로를 3차원 그래픽으로 모델링한 것이다. Fig. 4(b)는 가공한 구 가공물의 사진이다. 이 가공은 구의 지름을 60mm, 가공깊이는 2mm, 측면 방향의 이송거리는 1mm, 툴의 이송속도는 120 mm/min로 설정하여 가공하였으며 가공툴은 지름 12mm 볼-엔드밀을 사용하였다.

6. 결론

본 논문은 기계 가공기의 유현성과 작업성을 향상시킬 수 있는 병렬 메커니즘을 이용한 공작기계의 프로그램 개발을 수행하였다.

사용자 인터페이스 프로그램을 개발하기 위해서 병렬 메커니즘의 기구학해석을 수행하였고

이를 이용하여 위치 제어 및 속도 제어 프로그램을 개발하였다. 또한 3차원 복합형상 가공의 편리를 위하여 다양한 형상의 가공 함수를 개발하였고, 대표적인 3차원 형상인 구를 가공해 프로그램의 편리성을 보였다.

개발된 프로그램은 기존의 문자코드 입력방식보다 도식화 형태의 입력방식으로 실제 적용시 빠른 가공 프로그램 작성이 가능하다. 향후 다양한 형상의 가공 함수 개발과 기존의 G코드 프로그램 자동 변환함수 등을 개발할 것이다.

참고문헌

1. 최병오, 이 민기, 외 15명, "6축 복합형상 가공기 개발" 공업기반기술개발사업 최종 보고서, 산업자원부, 2000. 1.

2. Whitney, D. E., and Tung, E. D., "Robot Grinding and Finishing of Cast Iron Stamping Dies," ASME J. of Dynamic System, Measurement, and Control, vol. 114, no. 1, pp. 132-140, 1992.

3. Her, M. G. and Kazerooni, "Automated Robotic Deburring of Parts Using Compliance Control," ASME J. of Dynamic System, Measurement, and Control, vol. 113, pp. 60-66, 1991.

4. F. Pierrot, A. Foumier, and P. Dauchez, "Toward a fully parallel 6-DOF robot for high-speed applications," J. of Rob. Autom. pp. 15-22, 1992.

5. G. L. Long and C. L. Collins, "A pantograph linkage parallel platform master hand controller for force reflection," Proc. IEEE Int Conf. Rob. Autom., Nice, pp. 390-395, 1992.

6. R. Finkel, et al., "An Overview of AL, a Programming Language for Automation," Proceedings of 4th International Joint Conference of Artificial Intelligence, PP. 758-765, 1975.

7. R. P. Paul, "WAVE: A Model-Based Language for Manipulator Control", The Industrial Robot, vol. 4, pp. 10-17, 1977.

8. L. Lieberman and M. Wesley, "AUTOPASS: An Automatic Programming System for Computer-Controlled Mechanical Assembly," IBM Journal of Research Development, vol.21, no. 4. pp. 321-333, 1977.

9. C. C. Geschke, " A System for Programming and Controlling Sensor-Based Robot Manipulators," IEEE transactions on Pattern Analsys and Machine Intelligence, vol. PAMI-5, no. 1, pp. 1-7, 1983.

10. K. H. HUNT, *Kinematic geometry of mechanisms*, Oxford Univ. Press, 1978

11. M. Raghavan, "The Stewart platform of general geometry has 40 configurations," in *Advances in Design Automation*, G. A. Gabriele. Ed., ASME Press, New York 1991. pp. 397-402.

12. E. F., Ficher, "A Stewart Platform-Based Manipulator: General Theory and Practical Construction", Int. J. of Robotics Research, vol. 5, pp.157-182, 1986.