

### 직선 궤적 계획을 위한 로봇 제어에 관한 연구

◦길진수      홍석교  
아주대학교 제어공학과

#### A Study on the Robot Control for Straight Line Trajectory Planning

◦Jin Su Gil      Suk Kyo Hong  
DEPT. OF CONTROL ENG. AJOU UNIV.

#### Abstract

In this paper, the algorithm of Resolved Motion Rate Control(RMRC) is applied to the robot manipulator to implement a desired straight trajectory in the cartesian space. PI controller is also used to control the velocity and position which are produced by RMRC algorithm.

And Bounded Deviation Method is used to determine the intermediate knot points which satisfy a given tolerance limit, between the straight line segment.

#### 1. 서론

산업 현장에서 고도의 비선형 시스템으로 이루어진 로봇을 이용하여 생산성의 증대와 비용 절감 효과를 이루기 위해서 제품 생산 라인에서의 로봇은 보다 향상된 동작 속도를 필요로 한다. 이것은 직감적으로 액츄에이터의 크기를 증가시켜 해결할 수도 있으나 무게, 비용, 전력소모가 커지기 때문에 적당한 해결책이 아니다. 그것의 적당하고 효율적인 해결책의 하나로서, 주어진 작업 공간내의 임의의 두 점 사이를 로봇 매니퓰레이터가 수행할 수 있는 경로는 무한히 많지만, 이들 경로중 많은 잇점을 가지고 있는 직선 경로를 이용하여 로봇의 동작 시간을 가능한 한 최소화 하는 것이다. 그러나 많은 다관절체로 이루어진 로봇을 직선으로 움직이게 하는 것은 쉽지 않은 일로 간주되고 있다. 이미 많은 사람들이 그것에 관련된 연구를 하였는데, Paul[1]이 두번의 회전과 두번의 병진을 이용하여 직선 경로를 계획 하였고, Taylor[2]는 회전 동작을 표현하기 위해 키퍼니온 방법을 개발 하였다. Lee[3]는 부드러운과 토오크 제약 조건을 둘다 만족 하는 주어진 직선 경로상에 궤적 설정점을 정확하게 결정하기 위한 이산시간 궤적계획(discrete time trajectory planning)방법을 개발 하였다. 또 Whitney[4][5] 와 Paul [1]은 각각 inverse jacobian 과 inverse kinematics을 이용한 차동 직선 운동(differential straight line motion)방식을 제시 하였다. 본 논문에서는 직각 좌표 공간상에서 원하는 궤적을 실행하기 위하여 D.E. Whitney의 Resolved Motion Rate Control (RMRC)알고리즘을 적용하였고, RMRC에서 생성된 위치와 속

도 제어를 위해서 PI제어기를 사용하였다. 또한 주어진 허용오차를 만족하면서 임의의 두점간의 직선을 이루는 중간 경로점(intermediate knot points)을 구하기 위하여 Taylor[2]의 Bounded Deviation Method를 이용하였다. 실험 대상 로봇으로는 Rhino-XR3 로봇을 사용하였고, XR3 로봇 전용 콘트롤러 대신에 PC-AT와 인터페이스된 제어기를 만들어 5축중 3축만을 사용하여 제시된 알고리즘을 구현하면서 여러가지 도형의 궤적을 그려 보았다.

#### 2. Resolved Motion Rate Control(RMRC)

직각 좌표 공간에서 원하는 궤적을 수행 하기 위해서는 직각 좌표공간과 조인트 공간사이의 관계식을 먼저 알아야 하는데, 조인트 공간상에서의 속도와 직교 공간상에서의 속도는 다음의 (2)식으로 나타낼 수 있다. 이 식을 이용하면 직교 공간상의 속도에 대한 조인트 공간상의 속도를 구할 수 있다.

$$\dot{x}(t) = f(q) \tag{1}$$

$$\dot{x}(t) = J(q) \dot{q}(t) \tag{2}$$

여기서  $f(q)$ 는 전향 기구학(forward kinematics)을 나타내는 벡터이고  $J(q)=[f(q)/q]$ 는 매니퓰레이터의 Jacobian 행렬을 나타낸다. 6축인 로봇의 경우 각 조인트축의 속도를 나타내는 속도벡터는 다음의 (3)식으로 나타낸다.

$$\dot{q} = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3 \ \dot{q}_4 \ \dot{q}_5 \ \dot{q}_6]^T \tag{3}$$

여기서 T는 transpose를 나타낸다.

식 (2)으로 부터 자코비안 행렬의 inverse를 구하면 주어진 직각 좌표 공간의 선속도에 대응하는 각 조인트축의 각 속도를 구할 수 있으며 다음식으로 나타 내어진다.

$$\dot{q} = J^{-1}(q) \dot{x} \tag{4}$$

여기서  $J^{-1}(q)$ 는 pseudoinverse matrix로

$$J^{-1}(q) = [J^T(q)J(q)]^{-1}J^T(q)$$

로 나타낸다.

만약 조인트의 수가 6이라면 jacobian matrix는 정방 행렬이 되어  $J^{-1}(q) = J^{-1}(q)$ 이 되며 식(4)는 결과적으로

$$\dot{q} = J^{-1}(q) \dot{x} \tag{5}$$

(5)식과 같은 식이 된다.

본 실험에서는 3축만을 사용하여 실험을 하였고 때문에 엔드 이펙터의 방향은 고려치 않았다. 그러므로 자코비안 행렬을 구하기 위해서는 위치만을 고려한 자코비안 행렬이 필요하다.

관절 속도 벡터  $\dot{q}$ 와 직교 좌표상의 속도 벡터  $\dot{x}$ 는 다음과 같은 식으로 나타낼 수 있다.

$$\dot{q} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T \quad (6)$$

그리고  $\dot{x} = [\dot{X} \quad \dot{Y} \quad \dot{Z}]^T \quad (7)$

기구학 방정식을 이용하여 직교좌표 공간과 관절 좌표 공간사이의 관계를 구하면 다음과 같이 된다.

$$\dot{x} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} f_1(q_1, q_2, q_3) \\ f_2(q_1, q_2, q_3) \\ f_3(q_1, q_2, q_3) \end{bmatrix} \dot{q} \quad (8)$$

$\dot{x}$ 의 각 component는 (8)식을 미분하여 얻어진다.

$$\begin{aligned} \dot{X} &= \frac{\partial f_1}{\partial q_1} \dot{q}_1 + \frac{\partial f_1}{\partial q_2} \dot{q}_2 + \frac{\partial f_1}{\partial q_3} \dot{q}_3 \\ \dot{Y} &= \frac{\partial f_2}{\partial q_1} \dot{q}_1 + \frac{\partial f_2}{\partial q_2} \dot{q}_2 + \frac{\partial f_2}{\partial q_3} \dot{q}_3 \\ \dot{Z} &= \frac{\partial f_3}{\partial q_1} \dot{q}_1 + \frac{\partial f_3}{\partial q_2} \dot{q}_2 + \frac{\partial f_3}{\partial q_3} \dot{q}_3 \end{aligned} \quad (9)$$

혹은 행렬 형태로 나타내면

$$\dot{x} = J(q) \dot{q}$$

와 같고 직교 공간상의 속도 벡터에 대한 조인트 공간상의 속도 벡터는

$$\dot{q} = J^{-1}(q) \dot{x} = J^{-1}(q) \dot{x} \text{ 와 같다.}$$

여기서 Jacobian matrix는 다음과 같이 정의 된다.

$$J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\ \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \end{bmatrix} \quad (10)$$

### 3. 직선 궤적을 이루는 경로점의 설정

본 논문에서는 주어진 허용오차 내에서 근사화된 직선운동을 위한 경로점을 Taylor의 Bounded Deviation Method [6]를 이용하여 결정하였는데 여기서 사용한 알고리즘은 Taylor의 방법을 개선한 것[7][8]으로 직선을 이루는 직교 공간상의 임의의 두 점간의 knot points를 결정하는 방법은 다음과 같다.

Step 3.1 : 먼저  $\epsilon > 0$ 로 표시되는 허용 오차를 입력한다.

Step 3.2 : 역 기구학 방정식을 이용하여 직교 공간상에 주어진 초기 위치( $w^0$ )와 최종 위치( $w^1$ )에 해당하는 조인트 벡터( $q^0$  와  $q^1$ )를 구한다.

Step 3.3 : 조인트 공간상의 중간점을 구한다

$$q^m = \frac{q^0 + q^1}{2} \quad (11)$$

Step 3.4 : 절차 3.3 에서 구한  $q^m$ 을 기구학 방정식을 이용하여 그  $q^m$ 에 해당하는 직교 공간상의 좌표를 구한다.

$$w^m = f(q^m) \quad (12)$$

Step 3.5 : 주어진 초기위치와 최종위치의 정확한 직교 공간상의 중간점을 구한다.

$$w^M = \frac{w^0 + w^1}{2} \quad (13)$$

Step 3.6 : 만약에  $\|w^M - w^m\| \leq \epsilon$  을 만족하면 Step 3.8로 점프한다. 그렇지 않다면  $w^M$ 을 새로운 경로점(knot point)으로 설정한다.

Step 3.7 : 새로운 두 세그먼트( $w^0, w^M$ )와 ( $w^M, w^1$ )를 Step3.1부터 Step 3.6까지 다시 적용시켜 주어진 허용오차를 만족할 때까지 계속 실행한다.

Step 3.8 : End of Algorithm

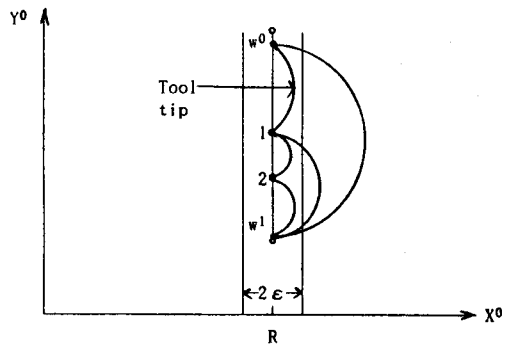


FIG. 1 Interpolated joint space approximation to straight line motion

### 4. 실험 및 실험 결과

Rhino XR3-robot의 기구학 방정식은 (14)식과 같고 (10)식에서 정의한 방법대로 Jacobian matrix를 구하면 (15)식과 같다.

$$\begin{aligned} f_1 &= p_x = c_1(a_2c_2 + a_3c_23) \\ f_2 &= p_y = s_1(a_2c_2 + a_3c_23) \\ f_3 &= p_z = -a_2s_2 - a_3s_3 \end{aligned} \quad (14)$$

여기서  $c_1 = \cos \theta_1$   
 $s_1 = \sin \theta_1$   
 $c_{23} = \cos(\theta_2 + \theta_3)$   
 $s_{23} = \sin(\theta_2 + \theta_3)$  이다.

$$J(q) = \begin{bmatrix} -s_1(a_2c_2 + a_3c_23) & -c_1(a_2s_2 + a_3s_23) & -a_3c_1s_23 \\ c_1(a_2c_2 + a_3c_23) & -s_1(a_2c_2 + a_3c_23) & -a_3s_1s_23 \\ 0 & -a_2c_2 & -a_3c_3 \end{bmatrix} \quad (15)$$

(2)식에 의해 조인트 축의 각속도에 대한 엔드 이펙터의 선속도는 (16)식과 같이 된다.

$$\dot{x} = \begin{bmatrix} -s_1(a_2c_2 + a_3c_23) & -c_1(a_2s_2 + a_3s_23) & -a_3c_1s_23 \\ c_1(a_2c_2 + a_3c_23) & -s_1(a_2c_2 + a_3c_23) & -a_3s_1s_23 \\ 0 & -a_2c_2 & -a_3c_3 \end{bmatrix} \dot{q} \quad (16)$$

그리고 (5)식에 의해 엔드 이펙터의 선속도에 대한 조인트 축의 각속도를 구하면 다음식과 같이 된다.

$$\dot{q} = \begin{bmatrix} -s_1(a_2c_2 + a_3c_23) & -c_1(a_2s_2 + a_3s_23) & -a_3c_1s_23 \\ c_1(a_2c_2 + a_3c_23) & -s_1(a_2c_2 + a_3c_23) & -a_3s_1s_23 \\ 0 & -a_2c_2 & -a_3c_3 \end{bmatrix}^{-1} \dot{x}$$

먼저 역 기구학 방정식(inverse kinematics)을 이용하여 그 라인 세그먼트를 따라 결정된 중간 knot points에 해당하는 joint의 관절 각도를 구한다. XR3로봇에 RMRC제어 알고리즘을 적용하여 생성되는 위치와 속도로 X,Y좌표에 궤적을 로봇이 직접 그리게 하고, 로봇이 지나간 경로에 대한 기구학 식을 계산하여 PC상에서 원하는 궤적과 실제의 궤적을 그래픽 처리하여 비교하였다. 먼저 직선을 그려 보았고 그 다음에 삼각형을, 그리고 사각형과 임의의 사각형을 그려보았다. 편의상 모든 궤적의 출발점은 (X,Y)좌표가 (400,0)[mm]이 되도록 하였다. Bounded error는 0.1(cm)에서 0.01(cm)의 범위로 바꾸어 가면서 실험을 하였고, 각 도형의 좌표 및 결과는 각 그림 6 에서 그림 13 까지 나와 있다. 주 프로그램의 순서도가 그림 2, 그림 4, 그림 5에 나와 있고, 그림 3는 적용 알고리즘의 블록도이다.

5. 결론

본 논문에서는 Whitney의 Resolved Motion Rate Control(RMRC) 알고리즘과 PI제어기를 이용하여 궤적 연구의 가장 기본이 되는 직선으로 이루어지는 임의의 도형을 그려 보았다. 실험 결과 그림에서 볼 수 있듯이 Bounded error가 작아지면 작아 질수록 궤적의 오차는 줄어들었으나 계산량이 증가하여 궤적을 그리는 실행 시간이 늘어났으며, 그런대로 만족할 만한 결과를 얻을 수 있었고 알고리즘의 타당성을 확인할 수 있었다. 그러나 로봇의 동력학을 고려치 않고 실험을 행하였기 때문에 부하 변동에 따른 결과는 얻을 수가 없었다. 앞으로 로봇의 동력학을 고려하여 본 알고리즘을 적용 한다면 좀더 규모가 큰 로봇에도 이용 가능하리라 본다.

참고 문헌

[1] Paul, R.C. 1979. "Manipulator Cartesian path control." IEEE Trans. Sys. Man Cyber., vol SMC-9.  
 [2] R.H.Taylor. "Planning and execution of straight line manipulator trajectories." IBM J. Res Develop .. vol. 23, pp. 424-436, July 1979.  
 [3] K.S.Fu, R.C.Gonzalez, and C.S.G.Lee. "ROBOTICS: Control, Sensing Vision, and Intelligence." McGraw-Hill Book Company, 1988.  
 [4] D.E. Whitney, "The mathematics of coordinate control of prosthetic arms and manipulators." ASME Trans J. Dyn. Measurement Contr., 94(4):303-309, 1972  
 [5] Whitney, D.E. 1969(June). "Resolved motion rate control of manipulators and human prostheses." IEEE Trans. Man Mach. Sys., vol. MMS-10, pp. 47-53  
 [6] Luh, J.Y.S., C.S., and Chang, P.R. 1983(Dec.). "Formulation and optimization of cubic polynomial joint trajectories for industrial robots." IEEE Trans. Automatic Control. AC-28(12)  
 [7] Y.H.Chang and T.T.Lee and C.H.Liu, "On-line cartesian path trajectory planning for robot manipulator." IEEE Int. Conf. on Robotics and automation., vol. 1, pp. 62-67, 1988.  
 [8] ROBERT J. SCHILLING, "Fundamentals and Robotics, Analysis and Control", Prentice-Hall International, Inc., 1990.  
 [9] Rolph H. Castain and Richard P. Paul. "An On-line Dynamic Trajectory Generator." Int. J. Robotics Research, 3, 68-72, 1984.  
 [10] S. Chand and Keith L. Doty. "On-line Polynomial Trajectories for robot manipulator." Int. J. Robotics Research, vol 4, No 2, 1985.

[11] H.Zhang, G.Trott, R.P.Paul. "Minimum delay PID control of interpolated joint trajectories of robot manipulator." IEEE Trans. Electr. vol. 37, No. 5, October, 1990.  
 [12] Yoram Koren, "Robotics For Engineers." McGraw-Hill, Inc., 1985.

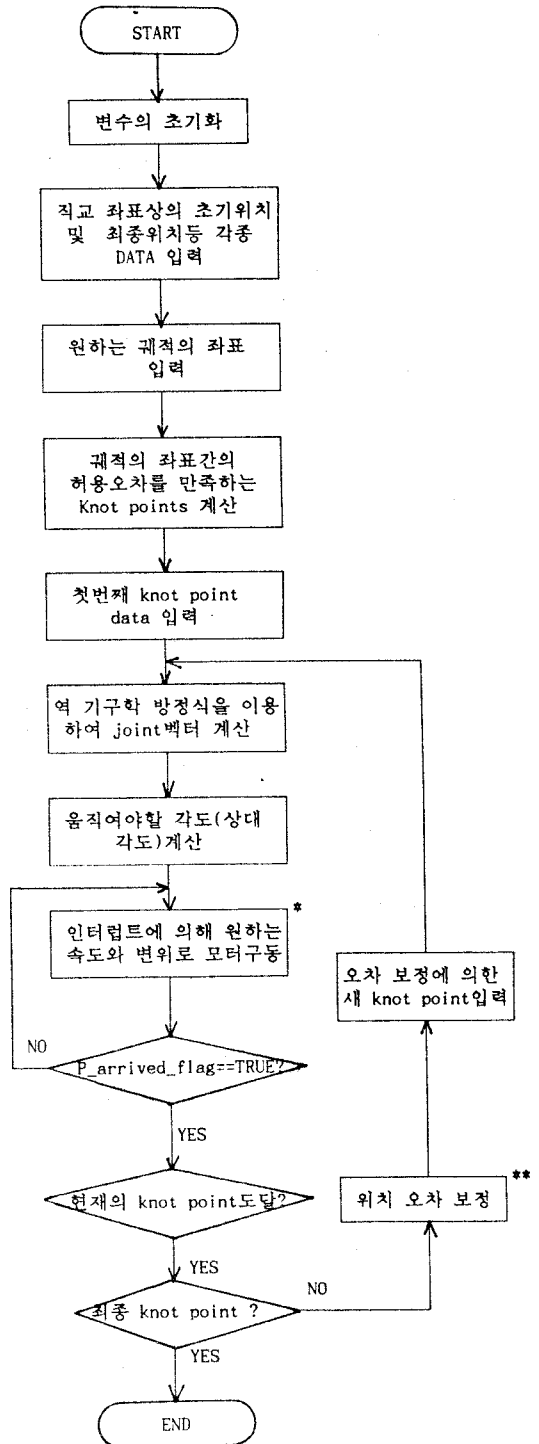


그림 2 주 프로그램 순서도  
 FIG. 2 Flow chart of main program

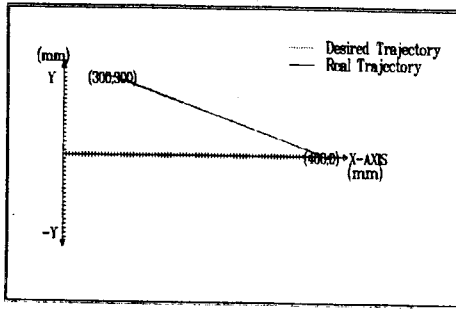


그림 6 직선 궤적의 결과(Bounded error=0.01)  
FIG. 6 Result of straight trajectory  
(Bounded error=0.01)

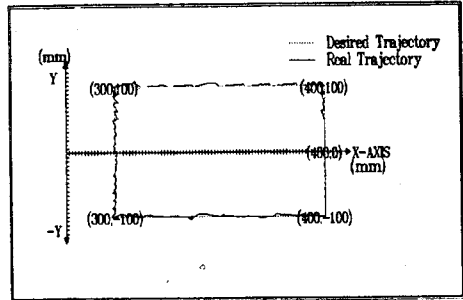


그림 10 사각형 궤적의 결과(Bounded error=0.05)  
FIG. 10 Result of rectangle trajectory  
(Bounded error=0.05)

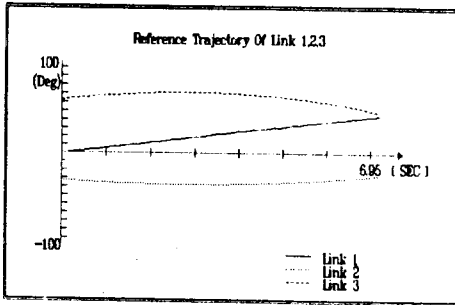


그림 7 그림 6.의 링크 1,2,3의 기준 궤적  
FIG. 7 Reference trajectory of link 1,2,3

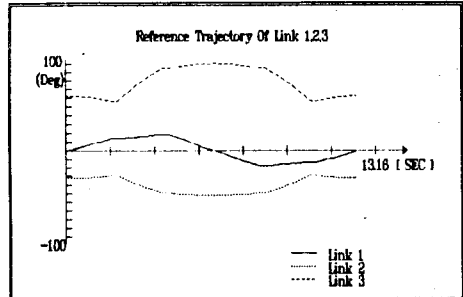


그림 11 그림 10의 링크 1,2,3의 기준 궤적  
FIG. 11 Reference trajectory of link 1,2,3

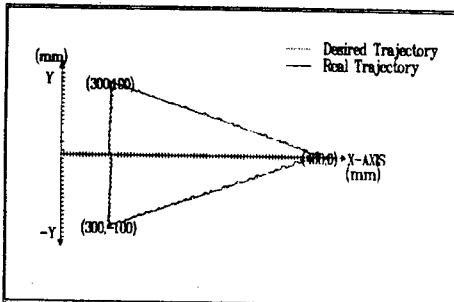


그림 8 삼각형 궤적의 결과(Bounded error=0.01)  
FIG. 8 Result of triangle trajectory  
(Bounded error=0.01)

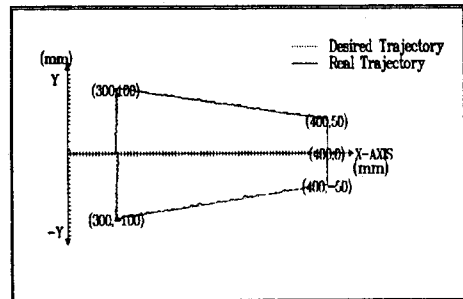


그림 12 사각형 궤적의 결과(Bounded error=0.01)  
FIG. 12 Result of tetragon trajectory  
(Bounded error=0.01)

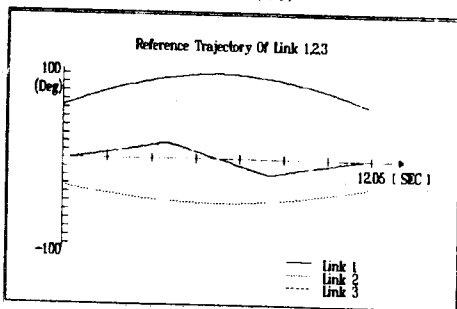


그림 9 그림 8.의 링크 1,2,3의 기준 궤적  
FIG. 9 Reference trajectory of link 1,2,3

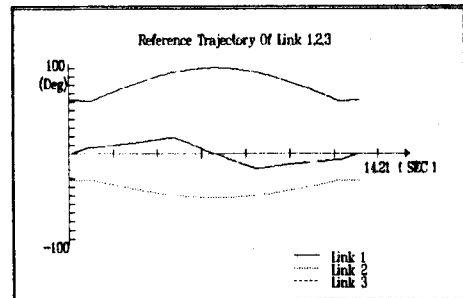


그림 13 그림 12.의 링크 1,2,3의 기준 궤적  
FIG. 13 Reference trajectory of link 1,2,3