

<論 文>

로봇의 기구 및 동역학적 기능평가를 위한 시뮬레이션 프로그램 개발

김 재 근* · 오 흥 국**

(1984년 9월 4일 접수)

Simulation Study of Kinematics and Dynamics of Robots

Jae Kun Kim and Hung Kuk Oh

Abstract

This study is to develop a program, with which the motion of the system can be simulated in view of kinematics and dynamics, and then evaluate functional capabilities of many robots in order to select the most suitable one for any given working conditions.

According to the theory of Denavit & Hartenberg, rectangular coordinate is set at each joint in the robot of six joints, as this one is the major concern, and transformation matrices are derived between any two coordinates.

The necessary displacement of each joint for executing a given work is obtained from the position and posture transformation and the instantaneous velocity of the joint is got from given velocity distribution of the path.

The necessary torque of each joint is finally calculated by Newton-Euler Method.

This simulation program is applied to a robot with six rotational joints,

Rotational angle and torque to time which has a given motion trajectory are figured and also torque to r.p.m.

1. 서 론

본 논문은 로봇의 동작 기능을 사전에 컴퓨터에 의해 검토할 수 있고, 로봇의 제작에도 도움을 줄 수 있는 프로그램 개발을 목적으로 한다.

*대우조선공업(주) 기술연구소

**정회원, 아주대학교 공과대학 기계공학과

본 논문에서 수행한 프로그램은 크게

(1) 로봇의 변위 계산

(2) 로봇의 속도 계산

(3) 로봇의 토크 계산

으로 나눌 수 있다.

변위 계산을 하기 위하여 각 관절에 Denavit & Hartenberg⁽¹⁾의 이론을 적용하였고, Paul, R.P.⁽²⁾가 적교 좌표계에서 경로를 제어하기 위하여 전개한 방식을

응용하였다.

속도 제한은 원하는 속도 분포⁽³⁾에 따라 매 시간마다 구하였다.

본 연구를 진행하는데 있어서 가장 중요한 부분은 토크 계산이라 할 수 있다.

토크 계산이 중요한 이유는

- (1) 로봇 제어에의 응용
- (2) 산업 로봇의 최적 설계와 자동화로 볼 수 있다⁽⁴⁾.

따라서, 그동안 많은 연구자들이 Lagrange 식에 기조를 두고 연구하였으나^(5,6,7,8,9), 계산 시간이 많이 소요되고, 실시간(real time)내에서 구하는데 어려움이 있었다.

그 후 Raibert 와 Horn⁽¹⁰⁾이 디지털 컴퓨터에 의해 자동적으로 계산을 간단히 할 수 있었지만, 표(table)로 구성하는 방법이어서 많은 기억 용량이 필요한 약점이 있었다.

또 하나의 가능한 연구는 Newton Euler 방법^(11,12)에 기초를 둔 것이다.

이 방법은 연속적으로 링크(link)에 적용할 수 있는 순환식(recursive)이므로 Lagrange 식과 같은 복잡성을 피할 수 있다. 그러나, 모든 관절에 대한 입력 힘과 토크가 자체 좌표계에 대한 것이어야 되는 반면, 이 방법은 기준좌표계에 대해 구해지는 단점이 있다.

1980년에 이르러 Luh, Walker 와 Paul⁽¹³⁾이 Newton Euler Formulation 을 링크자체 좌표계에 대해 계산할 수 있는 방법을 고안하였다. 이 논문의 연구결과를 보면 Fortran 을 사용하여 PDP 11/45에서 계산할 때 standford manipulator 의 모든 joint 의 힘과 Torque 를 계산하는데 Lagrange 식은 7.9초가 걸렸는데 Newton-Euler 방법으로는 2.5ms 가 걸렸다.

본 논문은 Luh, Walker 와 Paul 이 제안한 Newton Euler 방법에 따라 토크를 계산하였고, 임의로 로봇을 선택하여 시뮬레이션 프로그램을 적용하였다.

2. 운동학적 해석

2.1. 링크 기구에 대한 좌표 변환

Manipulator 는 상태 운동을 할 수 있는 관절에 의해 연결된, 여러개의 링크로 구성되어 있다.

Manipulator 의 각 링크간의 관계를 해석하기 위하여 Denavit & Hartenberg⁽¹⁾의 이론이 많이 사용되고 있다.

링크 $i-1, i, i+1$ 이 관절 $i, i+1$ 에 의해 연결되어

있을때(Fig. 1), 링크 i 에 고정된 직교좌표계는 다음과 같이 정의된다.

- (1) 링크 $i+1$ 의 회전축을 z_i 축이라 한다.
- (2) 링크 i 의 원점 θ_i 는 z_{i-1} 축과 z_i 축 사이의 공통 수선과 z_i 축과 만나는 점이다.
- (3) 공통 수선을 연장하여 x_i 축이라 한다.
- (4) 오른손 좌표계를 따르도록 y_i 축을 설정한다.

이 이론에 따라 각 링크에 설정된 직교 좌표계 사이의 관계를 나타내는 변환 매트릭스를 구하기 위하여, 관절 매개 변수 $a_i, \alpha_i, s_i, \theta_i$ 를 다음과 같이 정의한다.

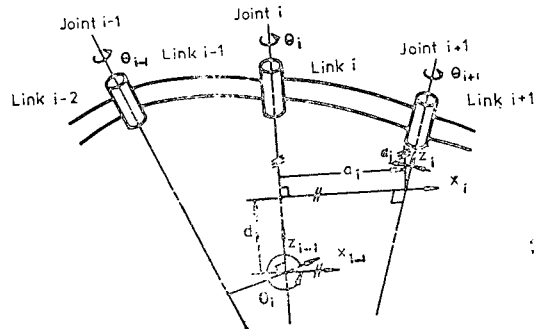


Fig. 1 Link parameters $\theta_i, d, a,$ and α for rotational joints

a_i ; z_{i-1} 축과 z_i 축 사이의 거리

α_i ; x_i 축에 관한 z_{i-1} 축과 z_i 축 사이의 각

s_i ; 원점 θ_{i-1} 로부터 x_{i-1} 축과 x_i 축이 만나는 점까지의 거리

θ_i ; z_{i-1} 축에 관한 x_i 축과 x_{i-1} 축 사이의 각

이 정의에 의해 링크 $i-1$ 의 좌표계와 링크 i 의 좌표계 사이의 관계는 2개의 회전 매트릭스와 2개의 병진 매트릭스로 나타낼 수 있다.

이 변환 매트릭스를 A 매트릭스라 부르고, 링크 i 에 대해 A_i 로 표시한다.

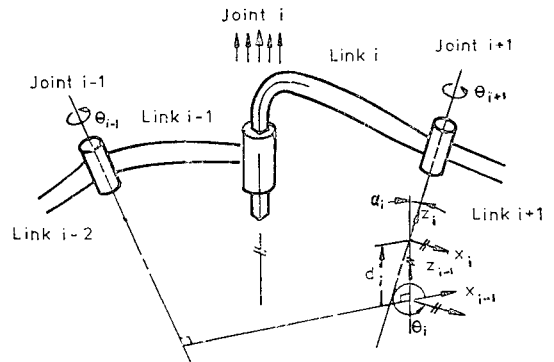


Fig. 2 Link parameters $\theta, d, a,$ and α for a sliding joint

$$A_i = \text{Rot}(z_{i-1}, \theta_i) \text{ Trans}(0, 0, s_i) \text{ Trans}(a_i, 0, 0) \text{ Rot}(x_i, \alpha_i)$$

$$= \begin{bmatrix} -\cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

여기서, Rot 와 Trans 는 각각 회전과 병진 매트릭스를 의미한다.

만약 Fig. 2 와 같이 미끄럼 관절로 되어 있으면, A 매트릭스는 다음과 같다.

$$A_i = \begin{bmatrix} -\cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & 0 \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

미끄럼 관절인 경우 $\theta_i = \theta_{i-1}$, $\alpha_i = \alpha_{i-1}$ 이 된다.

n 개의 링크를 가진 manipulator 의 처음 좌표계와 마지막 좌표계와의 관계를 0T_n 로 표시하고, A 매트릭스를 이용하여 나타내면

$${}^0T_n = A_1 \cdot A_2 \cdot A_3 \cdot \dots \cdot A_n \quad (3)$$

으로 표시할 수 있으며, 0T_n 은 T_n 으로 표시하기로 한다.

기존 좌표계로부터 공구(혹은 손)의 위치 자세(position & posture) P 라 하면, 이때 위치 자세 P 는

$$P = Z \cdot T_n \cdot E \quad (4)$$

로 표시될 수 있다.

여기서, Z 는 기존 좌표계와 manipulator 의 처음 좌표계와의 관계를 나타내는 매트릭스이고, E 는 manipulator 이 마지막 좌표계와 공구(tool)와의 관계를 나타내는 매트릭스이다.

2.2. 일반적인 회전 변환식 유도

임의의 특정한 축에 대한 일반적인 회전 매트릭스를 구하기 위하여, 특정축 k 가 임의의 좌표계 C 에서 z 축의 단위 벡터라 가정한다.

$$C = \begin{bmatrix} n_x & O_x & a_x \\ n_y & O_y & a_y \\ n_z & O_z & a_z \end{bmatrix} \quad (5)$$

$$k = a_x i + a_y j + a_z k \quad (6)$$

기존 좌표계에 대한 좌표계 T 는, 좌표계 C 에 대하여 좌표계 X 로 나타낼 수 있다.

$$T = CX \quad (7)$$

위의 가정에 의해, k 축에 관해 좌표계 T 를 회전시킨 것은, 좌표계 C 의 z 축에 대해 X 를 회전시킨 것과 같다.

$$\text{Rot}(k, \theta) \cdot T = C \cdot \text{Rot}(c_z, \theta) \cdot X \quad (8)$$

$$\text{Rot}(k, \theta) \cdot T = C \cdot \text{Rot}(c_z, \theta) \cdot C^{-1} \cdot T \quad (9)$$

여기서, c_z 는 좌표계 C 의 z 축이다.

따라서,

$$\text{Rot}(k, \theta) = C \cdot \text{Rot}(c_z, \theta) \cdot C^{-1} \quad (10)$$

이고, 이를 전개하여 간단히 하면 다음과 같다.

$$\text{Rot}(k, \theta) = \begin{bmatrix} k_x k_x \text{Vers}\theta + \cos\theta & k_x k_y \text{Vers}\theta - k_z \sin\theta & k_x k_z \text{Vers}\theta + k_y \sin\theta \\ k_x k_y \text{Vers}\theta + k_y \sin\theta & k_y k_y \text{Vers}\theta + \cos\theta & k_y k_z \text{Vers}\theta - k_x \sin\theta \\ k_x k_z \text{Vers}\theta - k_y \sin\theta & k_y k_z \text{Vers}\theta + k_x \sin\theta & k_z k_z \text{Vers}\theta + \cos\theta \end{bmatrix} \quad (11)$$

여기서, $k_x = a_x$, $k_y = a_y$, $k_z = a_z$, $\text{Vers}\theta = 1 - \cos\theta$ 이다.

2.3. 각 관절의 변위 계산

Manipulator 는 원하는 지점을 나타내는 위치와 공구의 방향을 나타내는 자세로 나눌 수 있다.

식 (1) 의 좌측 3x3 매트릭스가 자세를 나타내고, 우측의 3x1 매트릭스가 위치를 나타낸다. [Ref. 14 참조]

Manipulator 는 대부분 6 관절로 이루어져 있으며, 이중 3 관절은 위치를 결정하고, 나머지 3 관절이 자세를 결정한다.

본 논문에서는 manipulator 의 원하는 위치자세가 주어지면, 위치와 자세로 나누어 각 관절의 변위를 구하였다.

6 관절중 3 관절이 위치를 결정하고, 나머지 3 관절이 자세를 결정하는 manipulator 의 위치에 대한 변위 계산은 기하학적 형상에 따라 계산이 달라지지만, 쉽게 해결할 수 있으므로 생략하고, 자세에 대한 변위 계산만을 고려하기로 한다.

Manipulator 의 자세는 공구가 사물에 접근하는 벡터(approaching) a, 잡는 방향을 결정하는 지향 벡

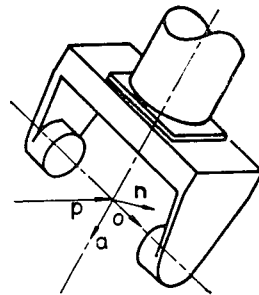


Fig. 3 n, o, a and p vectors

터 (orientation vector) \mathbf{o} 와 오른손 좌표계를 따르도록 결정하는 법선 벡터 (normal vector) \mathbf{n} 을 사용하여 표시할 수 있다.

자세 P_{01} 에서 자세 P_{02} 로 이동 시킬때 자세 P_{02} 는
$$P_{02} = P_{01} \cdot D \quad (12)$$

로 나타낼 수 있다.

여기서, D 는 자세 P_{01} 에서 자세 P_{02} 로 이동시킬때 요구되는 드라이브 변환 매트릭스 (drive transformation matrix) 이다.

식 (12) 를 D 에 대해서 풀면 다음과 같다.

$$D = P^{-1}_{01} \cdot P_{02}$$

$$= \begin{bmatrix} n_{x1} & n_{y1} & n_{z1} \\ o_{x1} & o_{y1} & o_{z1} \\ a_{x1} & a_{y1} & a_{z1} \end{bmatrix} \begin{bmatrix} n_{x2} & o_{x2} & a_{x2} \\ n_{y2} & o_{y2} & a_{y2} \\ n_{z2} & o_{z2} & a_{z2} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{n}_1 \cdot \mathbf{n}_2 & \mathbf{n}_1 \cdot \mathbf{o}_2 & \mathbf{n}_1 \cdot \mathbf{a}_2 \\ \mathbf{o}_1 \cdot \mathbf{n}_2 & \mathbf{o}_1 \cdot \mathbf{o}_2 & \mathbf{o}_1 \cdot \mathbf{a}_2 \\ \mathbf{a}_1 \cdot \mathbf{n}_2 & \mathbf{a}_1 \cdot \mathbf{o}_2 & \mathbf{a}_1 \cdot \mathbf{a}_2 \end{bmatrix} \quad (13)$$

또한 드라이브 변환 매트릭스 D 는 2 개의 회전매트릭스를 이용하여 나타낼 수 있다⁽²⁾.

첫번째 회전 매트릭스는 P_{01} 에서의 접근 벡터 \mathbf{a}_1 을 P_{02} 에서의 접근 \mathbf{a}_2 벡터로 이동시키는 것이다.

이 회전은 자세 P_{01} 을 z 축에 관하여 각 η 만큼 회전시켜 얻어진 y 축을 구하고, 이 y 축을 k 축으로 하여 각 ξ 만큼 회전시킨 것이다(식 11 참조).

이 회전을 R_a 라 하면 다음과 같이 구해진다.

$$R_a = \begin{bmatrix} S_\eta^2 V_\xi + C_\xi & -S_\xi C_\eta V_\xi & C_\eta S_\xi \\ -S_\eta C_\xi V_\xi & C_\eta^2 V_\xi + C_\xi & S_\eta S_\xi \\ -C_\eta S_\xi & -S_\eta S_\xi & C_\xi \end{bmatrix} \quad (14)$$

여기서, $S_\eta = \sin \eta$, $C_\eta = \cos \eta$, $V_\xi = 1 - \cos \xi$ 이고, 앞으로는 모두 이렇게 약해서 쓰기로 한다.

두번째 회전 매트릭스는 자세 P_{01} 에서의 지향 벡터 \mathbf{o}_1 으로부터 자세 P_{02} 에서의 지향 벡터 \mathbf{o}_2 로 회전시키는 것이다.

이 회전을 R_o 로 표시하고, 단축히 z 축에 관해 각 ζ 만큼 회전시킨 것과 같다.

따라서, 드라이브 변환 매트릭스 D 는 다음과 같이 전개된다.

$$D = R_a \cdot R_o$$

$$= \begin{bmatrix} C_\xi (S_\eta^2 V_\xi + C_\xi) + S_\xi S_\eta V_\xi \\ C_\xi (-S_\eta C_\eta V_\xi) + S_\xi (C_\eta^2 V_\xi + C_\xi) \\ -C_\xi C_\eta S_\xi - S_\xi S_\eta S_\xi \\ -S_\xi (S_\eta^2 V_\xi + C_\xi) - C_\xi S_\eta C_\eta V_\xi & C_\eta S_\xi \\ S_\xi S_\eta C_\eta V_\xi + C_\xi (C_\eta^2 V_\xi + C_\xi) & S_\eta S_\xi \\ S_\xi C_\eta S_\xi - C_\xi S_\eta S_\xi & C_\xi \end{bmatrix} \quad (15)$$

식 (13) 과 식 (15) 의 원소 (1, 3), (2, 3), (3, 3) 으로 부터

$$C_\eta S_\xi = \mathbf{n}_1 \cdot \mathbf{a}_2 \quad (16)$$

$$S_\eta S_\xi = \mathbf{o}_1 \cdot \mathbf{a}_2 \quad (17)$$

$$C_\xi = \mathbf{a}_1 \cdot \mathbf{a}_2 \quad (18)$$

가 된다.

식 (16) 과 (17) 로부터 각 η 를 구하면

$$\tan \eta = \frac{\mathbf{o}_1 \cdot \mathbf{a}_2}{\mathbf{n}_1 \cdot \mathbf{a}_2} \quad -\pi \leq \eta < \pi$$

이고, 각 ξ 는 $\eta = 90^\circ$ 혹은 $\eta = -90^\circ$ 일 경우와 $\eta \neq 90^\circ$ 혹은 $\eta \neq -90^\circ$ 일 경우로 나누어 구할 수 있다.

(1) $\eta = 90^\circ$ 혹은 $\eta = -90^\circ$ 일 경우 식 (17) 과 식 (18)

로 부터

$$\tan \xi = \frac{\mathbf{o}_1 \cdot \mathbf{a}_2}{(\mathbf{a}_1 \cdot \mathbf{a}_2) S_\eta} \quad -\frac{\pi}{2} \leq \xi \leq \frac{\pi}{2} \quad (19)$$

(2) $\eta \neq 90^\circ$ 혹은 $\eta \neq -90^\circ$ 일 경우 식 (16) 과 식 (18)

로 부터

$$\tan \xi = \frac{\mathbf{n}_1 \cdot \mathbf{a}_2}{(\mathbf{a}_1 \cdot \mathbf{a}_2) \cdot C_\eta} \quad -\frac{\pi}{2} \leq \xi \leq \frac{\pi}{2} \quad (20)$$

식 (15) 를 R_o 에 대해 풀면

$$R_o = R_a^{-1} \cdot D$$

이고, 이를 전개하면 다음과 같다.

$$\begin{bmatrix} C_\xi & -S_\xi & 0 \\ S_\xi & C_\xi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_\eta^2 V_\xi + C_\xi & -S_\eta C_\eta V_\xi & -C_\eta S_\xi \\ -C_\eta C_\eta V_\xi & C_\eta^2 V_\xi + C_\xi & -S_\eta S_\xi \\ C_\eta S_\xi & S_\eta S_\xi & C_\xi \end{bmatrix} \quad (21)$$

식 (21) 의 원소 (2, 1) 과 식 (2) 로부터 다음을 구할 수 있다.

$$S_\xi = -S_\eta C_\eta V_\xi (\mathbf{n}_1 \cdot \mathbf{n}_2) + (C_\eta^2 V_\xi + C_\xi) (\mathbf{o}_1 \cdot \mathbf{n}_2) - S_\eta S_\xi (\mathbf{a}_1 \cdot \mathbf{n}_2) \quad (22)$$

$$C_\xi = -S_\eta C_\eta V_\xi (\mathbf{n}_1 \cdot \mathbf{o}_2) + (C_\eta^2 V_\xi + C_\xi) (\mathbf{o}_1 \cdot \mathbf{o}_2) - S_\eta S_\xi (\mathbf{n}_1 \cdot \mathbf{a}_2) \quad (23)$$

식 (22) 와 식 (23) 으로부터 각 ζ 를 구하면

$$\tan \zeta = \frac{S_\xi}{C_\xi} \quad -\pi \leq \zeta < \pi \quad (24)$$

이 된다.

3. 속도 및 동역학적 해석

속도는 주어진 작업 조건을 수행하기 위한 속도 분포를 미리 정하고, 이 분포에 의해 각 관절의 속도를 구한다⁽¹⁴⁾.

이 속도에 따라 Luh, Walker 와 Paul⁽¹³⁾ 이 전개한 방식을 이용하여 각 관절의 토크를 계산한다.

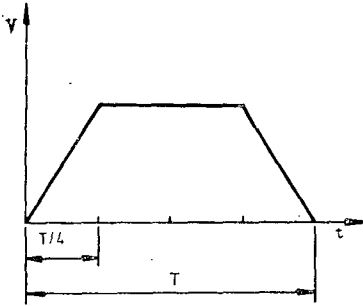


Fig. 4 Distribution of velocity

4. 시뮬레이션을 위한 프로그램 구성

본 논문에서는 다음과 같이 프로그램을 구성하였다.

- (1) 기본 데이터를 읽는다.
 기본 데이터로는 다음과 같은 것들을 들 수 있다.
 - (가) 로봇의 관절 수
 - (나) 각 관절의 매개 변수 ; $\alpha_i, \theta_i, s_i, a_i$
 - (다) 각 관절의 무게 중심 좌표
 - (라) 각 관절의 관성 모우멘트
 - (마) 변환 매트릭스 Z, E
 - (바) 운동 범위의 한계 및 토크의 한계
- (2) 로봇의 초기 위치 자세 P_1 을 읽는다.
- (3) 이동시키기 원하는 위치 자세 P_2 를 읽는다.
- (4) 식 (4)에 의해 변환 매트릭스 T_6 를 구한다.
- (5) T_6 로부터 관절의 회전각 ϕ_1, ϕ_2, ϕ_3 를 구한다.
- (6) A_3^6 를 구한다.

$$A_3^6 = (A_2^3)^T \cdot (A_1^2)^T \cdot (A_0^1)^T \cdot A_0^6$$

여기서, A_0^6 는 T_6 에서 자세만을 취한 것이다.

- (7) A_3^6 로부터 관절의 회전각 ϕ_4, ϕ_5, ϕ_6 (혹은 η, ξ, ζ)를 구한다. (2.3절)
- (8) 원하는 속도 분포에 따라 시간 t 에서의 속도와 가속도를 구한다.
- (9) 각 관절의 토크를 구한다.
- (10) 주어진 시간 T 에 달하였으면 11로 가고, 그렇지 않으면 시간을 $t + \Delta t$ 로 하여 8로 가서 계속하라.
- (11) 계속 수행할 작업이 있으면 12로 가고, 그렇지 않으면 멈추다.
- (12) A_3^6 와 현재의 회전각을 기억하고, 3으로 가서 계속하라.

5. 시뮬레이션 실행 및 결과

본 논문에서는 임의의 6관절 로봇을 선택하여 FORTRAN 프로그램에 의하여 시뮬레이션을 수행하였다. (Burroughs B1800/B1700)

본 논문에서 적용한 시뮬레이션 작업은 로봇의 초기 위치 자세를 A 로 하여, 1초동안 위치 자세 B 로 가서 물건을 잡은 후, 1초동안 위치 자세 C 로 가서 물건을 놓고, 다시 1초동안 위치 자세 A 로 돌아가는 것이다.

시간에 따른 각 관절의 회전각을 Fig. 5에 보였고, 이 결과에 의해 로봇이 움직일 수 있는 범위인가 아닌가를 파악할 수 있다.

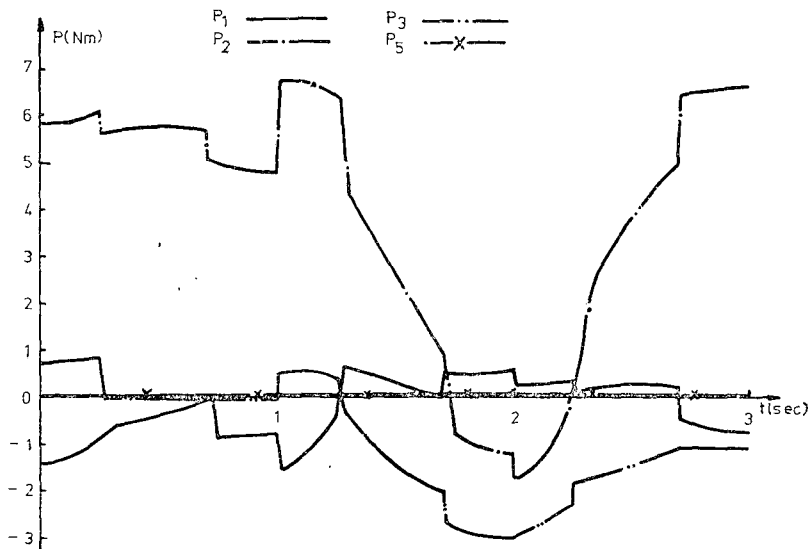


Fig. 5 Time functions of the generalized coordinates

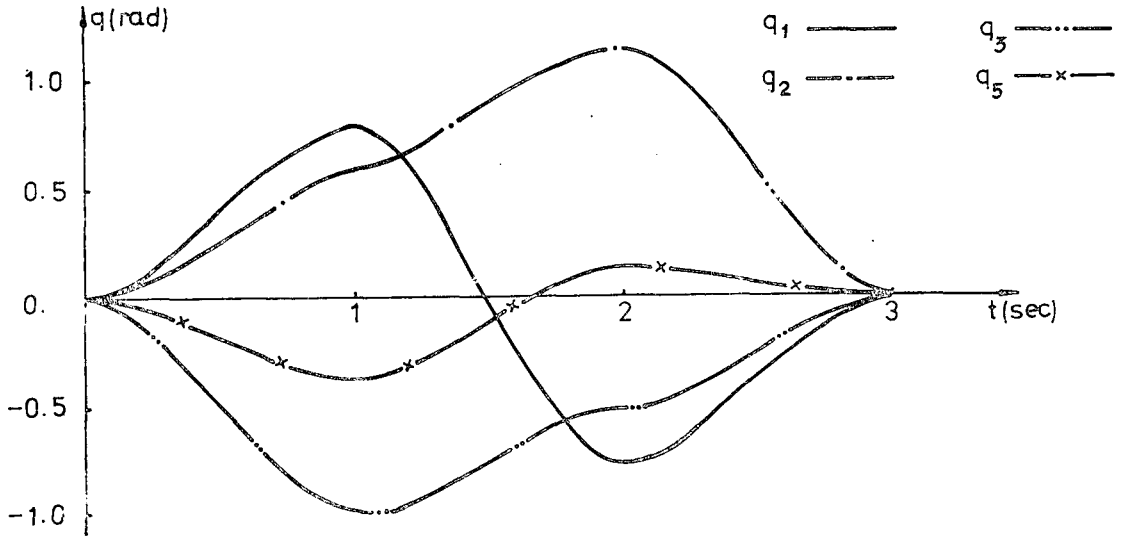


Fig. 6 Time-function of the drives in mechanism joints

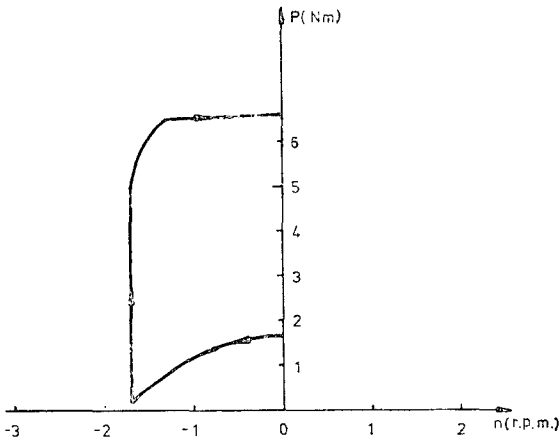


Fig. 7 Dependence torque-r.p.m. for joint 2

Fig. 6에는 시간에 따른 각 관절의 구동 토크를 나타내었고, 이 결과에 의하여 로봇이 구동할 수 있는 토크를 낼 수 있는 범위인가, 아닌지도 알 수 있다.

Fig. 6에서 볼 때 시간에 따른 구동 토크의 변화가 가장 큰 관절 2를 택하여, 회전수에 따른 구동 토크의 변화를 Fig. 7에 나타내었다.

이 그림을 모우터의 제원과 비교하면, 로봇의 제작시 모우터의 선정에도 이용할 수 있을 것이다.

6. 결 론

6 관절 모두가 회전 관절인 로봇에 주어진 작업을

적용하여, 시뮬레이션을 수행한 결과 다음과 같은 결론을 얻었다.

- (1) 움직여야 할 회전각 및 시간에 따른 회전각의 변화를 알 수 있으므로, 과연 로봇이 움직일 수 있는 범위인가 아닌가를 파악할 수 있었다.
- (2) 시간에 따라 각 관절에 걸리는 구동 토크가 계산되므로 로봇이 이것을 낼 수 있는 범위인지 아닌지 알 수 있었다.
- (3) 회전수에 따른 토크가 구해지므로 로봇 제작시 모우터의 선택도 용이하였다.

후 기

본 연구는 아주대학교의 연구비에 의하여 수행되었으며 이에 사의를 표합니다.

References

- (1) Denavit J. and Hartenberg R.S., "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices", ASME J. Applied Mechanics 22, pp. 215~221, 1955
- (2) Paul R.P., "Manipulator Cartesian Path Control", IEEE Trans. on Systems, man and Cybernetics, Vol. SMC-9, 11, pp. 702~711, 1979
- (3) コンピュータ・アシステッド・ロボントシステム・

- エンジニアリングのシステムデザイン 調査報告書, JIRA, pp.141~219, 1981
- (4) Vukobratovic M. and Potkonjak V., Scientific Fundamentals of Robotics (1. Dynamics of Manipulation Robots), Springer Verlag, New York, 1982
- (5) Vulobratovic M., "Synthesis of Artificial Motion," J. of Mechanism and Machine Theory, Vol. 13, 1, 1978
- (6) Vukobratovic M. and Potkonjak V., "Contribution to Automatic Forming of Active chain Models via Lagrangian Form", J. Applied Mechanics, 1, 1979
- (7) Vukobratovic M., and Potkojak V., "Two Methods for Computer Forming of Dynamic Equations of Active Mechanisms", J. Mechanism and Machine theory, Vol. 14, 3, 1979
- (8) Hollerbach J.M., "A Recursive Lagrangian Formulation of Manipulator Dynamic and Comparative Study of Dynamics Formulation Complexity", IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-10, 11, pp.730~736, 1980
- (9) Kahn M.E. and Roth, B., "The Near Minimum-Time Control of Open-Loop Articulated Kinematic Chains", J. Dynamic Systems, Measurement, Control, 93, pp.164~172, 1971
- (10) Raibert M.H. and Horn B.K.P., "Manipulator Control Using the Configuration Space Method", Industrial Robot 5, 2, pp.67~73, 1978
- (11) Ho J.Y.L., "Direct Path Method for Flexible Multi-body Space-Craft Dynamics", J. Spacecraft and Rockets (AIAA), Vol. 14, 2, pp.102~110, 1977
- (12) Aristova M.V., Ignatiev M.B., and Prokhorov V.M., "Algorithmic System for Robot's Motion Simulation", Proceedings of Symposium on Theory and Practice of Robots and Manipulators, Warsaw, Poland, 1976
- (13) Luh J.Y.S., Walker M.W. and Paul R.P.C., "On-Line Computational Scheme for Mechanical Manipulators", J. Dynamic Systems, Measurement, and Control", Vol. 102, pp.69~76, 1980
- (14) Paul, "Robot Manipulators", The MIT Press, 1982