

여유 자유도를 갖는 로봇의 기구학적 제어방법 연구를 위한

시뮬레이션 프로그램의 개발

이 검 수\*, 서 일 홍\*\*, 서 병 실\*, 김 경 기\*\*

한양대학교 전자통신공학과\* , 전자공학과\*\*

Development of Softwares for the Kinematic Control of Redundant Manipulators

Lee Kyoung-Soo , Suh Il-Hong , Suh Byung-Suhl and Kim Kyoung-Ki

Electronic Communication Dept. and Electronics Dept. ,Hanyang University

Abstract

Some of computational algorithms for the kinematic control of redundant manipulators are investigated and analyzed by extensive computer simulations.

I. 서론

공장 자동화의 일환으로 여러가지 제조 분야에서 산업용 로봇이 사용되는 경우가 많아짐에 따라, 최근에는 산업용 로봇의 그 기능의 다양성과 유동성에 대한 요구가 증가하고 있다. 이러한 요구를 충족시키기 위하여 로봇 매니퓰레이터에 더욱 많은 자유도를 주고 있는데, 이와같이 어떠한 과제수행에 필요한 자유도 이상의 자유도를 가진 로봇을 그 과제에 대하여 여유 자유도를 가지고 있다고 말한다. [2] 여유 자유도를 갖는 매니퓰레이터는 특히 작업 라인이 제약을 받는 복잡한 제조 작업에 유용하다. 이외에도 작업 영역내에서 장애물을 피하고, 자유도를 상실하게 되는 특이상태(Singular State)를 피하는 등 여러가지 면에서 여유 자유도가 있는 로봇보다 다양한 능력을 가지고 있다. [2, 3, 4] 따라서 이러한 여유 자유도를 갖는 매니퓰레이터를 제어, 실행시키기 위한 효율적인 알고리즘의 개발이 더욱 중요하게 되었다.

본 논문에서는 위에서 기술한 여유 자유도를 갖는 로봇의 제어를 위한 컴퓨터 소프트웨어 알고리즘에 관하여 논하고자 한다.

II. 여유 자유도를 갖는 로봇의 기구학적 방정식 및 그 일반해

매니퓰레이터의 위치와 자세를 관절의 각도값으로 표시하면, n 차원 end effector 의 원하는 위치와 자세를 나타내는 벡터 r 및 n 차원 관절각 벡터 θ 에 대하여,

$$r = f(\theta) \quad (1)$$

로 나타낼 수 있다. 식 1)을 시간에 관하여 양변을 각각 미분하면 속도 관계식

$$\dot{r} = J(\theta) \dot{\theta} \quad (2)$$

이 된다. 식 2)에서  $J(\theta)$ 는  $\frac{\partial f}{\partial \theta}$ 로서  $m \times n$  차원 자코비안 행렬(Jacobian Matrix) 이라고 하며, 비선형 관계식의 식 1)을 선형화시키는 계수가 된다. 여기서 문제는 주어진  $\dot{r}$  에 대하여  $\dot{\theta}$  를 찾는 것으로서, 식 2)의 해는 일반적으로

$$\dot{\theta} = J^+(\theta) \cdot \dot{r} + (I - J^+(\theta) \cdot J(\theta)) \cdot z(\theta) \quad (3)$$

로 주어진다. 따라서 식 3)을 풀기 위해서는 J 및 J의 가상 역변환인  $J^+$ 를 관절각  $\theta$ 가 변할 때마다 구해야만 한다. 또한 식 3)을 컴퓨터를 이용하여 구하고자 할때, 근사화를 하여야 하는바, 근사식이 계산이 빠르고, 오차가 적도록 하여야 한다.

이제 위에서 기술한 바와같이, J 및  $J^+$ 를 컴퓨터를 이용하여 수치 반복적으로 구하는 방법 및 식 3)을 가장 효율적으로 풀기 위한 방법들을 조사하여, 이를 컴퓨터 시뮬레이션을 통하여 비교, 검토하고자 한다.

III. 자코비안 행렬 및 가상 역변환의 수치 반복적 계산 방법

3-1. 자코비안 행렬 계산 방법

식 3)을 풀기 위하여 필요한 자코비안 행렬은 [1]에서와같이 closed form으로 구하는 방법은 많은 관절을 가지는 redundant manipulator에서는 매우 지루하고, 오류를 범하기 쉬운 과정을 필요로 한다. 또한 많은 연산 과정을 필요로 하므로 계산상의 효율성도 떨어지게 된다. 따라서 반복적이고 효율적인 계산에 의해 자코비안 행렬을 구하는 방법에 대하여 많은 연구가 이루어 졌다. [5] [6] 본 논문에서는 [5]에서 제시한 방법을 사용하여 자코비안 행렬을 구하고자 한다.

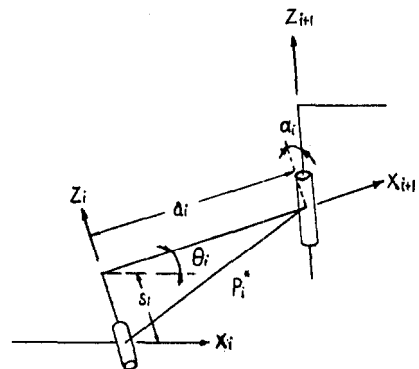


FIG.1 Definition of Link parameters

그림 1.의 일반적 형상이다하여 로드리게스 방정식을 적용하여 좌표계간의 관계식을 구하면 다음과 같다.

$$\begin{aligned} Z_i &= Z_{ii} \cos \alpha_i + Y_{ii} \sin \alpha_i \\ X_i &= X_{ii} \cos \theta_i + Y_{ii} \sin \theta_i \cos \alpha_i + Z_{ii} \sin \theta_i \sin \alpha_i \\ Y_i &= X_{ii} \sin \theta_i + Y_{ii} \cos \theta_i \cos \alpha_i + Z_{ii} \cos \theta_i \sin \alpha_i \end{aligned} \quad (4)$$

$$P_i^* = X_{ii} a_i + Y_{ii} s_i \sin \alpha_i + Z_{ii} c_i \cos \alpha_i \quad (5)$$

$$J_i^* = [J_{i1}^* \dots J_{in}^*] \quad (5)$$

$P_i^*$ 는  $i$ 번째 좌표계 원점에서  $i+1$ 번째 좌표계 원점까지의 거리 벡터이며,  $J_i^*$ 는  $i$ 번째 좌표계 원점에서  $j$ 번째 좌표계 원점까지의 위치 벡터이다. 그리고  $a_i$ 는 링크의 길이를 뜻하고,  $\theta_i$ 는 좌표계간의 차감각을 의미한다.  $\theta_i$ 는 관절의 회전각으로서  $X_{ii}$  축과  $X_i$  축사이의 각도이고,  $\alpha_i$ 는 좌표계간의 비틀림각도로서,  $Z_{ii}$  축과  $Z_i$  축사이의 각도이다. 여기서  $j$ 는 end effector를 뜻한다.

3)식과 4)식을 이용하여 [9]에서와같이  $i$ 번째 관절에 의한 end effector 자코비안 행렬의  $i$ 번째 열(column)을 구하면 다음과 같다.

$$\begin{aligned} \text{회전 관절일 경우:} \\ (J_j^*) &= [J_{j1}^* \dots J_{jn}^* \mid Z_j^*]^T \\ \text{미끄럼 관절일 경우:} \\ (J_j^*) &= [J_{j1}^* \dots J_{jn}^* \mid 0]^T \end{aligned} \quad (6)$$

$J_j^*$ 는  $i$ 번째 좌표계를 end effector에 관하여 표시한 것이다.

### 3-2. 가상 역행렬 계산 방법

여유 자유도를 갖는 경우의 자코비안 역행렬인, pseudoinverse matrix도 일반적인 경우에 적용하기 위하여 수치 반복적인 방법을 사용하고자한다. 이를 위하여 다음과 같은 알고리즘을 사용하였다. [7]

행렬  $J$ 를 각각의 열  $J_k$ 로 분해한다.

$$J_{nk} = [J_{k1} \ J_{k2} \ \dots \ J_{kn}]$$

그러면  $J_{nk}$ 의 가상 역행렬 행렬  $J_k^*$ 는

$$J_k^* = \begin{bmatrix} J_{k1}^* & \dots & J_{kn}^* \\ & & d_k \end{bmatrix} \quad k=1, 2, \dots, n$$

으로 표시된다.

$$\begin{aligned} \text{여기서, } d_k &= J_{k1}^* J_{k1} \quad ; \quad (k \times 1) \text{ 차원} \\ c_k &= J_k - J_{k1}^* d_k \quad ; \quad (1 \times n) \text{ 차원} \\ b_k &= c_k^T + (1 - c_k^T c_k)^{-1} (1 + c_k^T d_k)^{-1} d_k^* J_{k1}^* \end{aligned}$$

이다.

## IV. 자코비안 방정식의 수치 해법

본 절에서는 식 (3)에 주어진 자코비안 방정식의 해를 수치적으로 구하는 6 가지 방법을 오차 및 계산 속도의 관점에서 비교 분석하고자한다. 이를 위하여, 먼저 기준 좌표계에서 마니풀레이터의 end effector의 원하는 자세 및 위치가 다음과같이 주어진다.

$$r(1), r(2), r(3), \dots, r(n) \quad (7)$$

한편 경로 오차  $e(k)$ 는 수치 해법에 의한 해  $(k)$ 를 식(1)에 대입하여 얻은 마니풀레이터의 end effector 위치 및 자세의 차이로 정의한다.

$$e(k) = r(k) - f(S(k)) \quad (8)$$

### 4-1. 방법의 기술

#### 방법 1.

$$\begin{aligned} \theta(k+1) &= \theta(k) + J^*(\theta(k)) \cdot \Delta r(k) \\ &\quad - J^*(\theta(k)) \cdot G(r(k) - f(\theta(k))) + Re(\theta(k)) \end{aligned}$$

여기서  $r(k)$ 는  $r(k+1) - r(k)$  이고  $G$ 는 귀환 이득이며,  $Re(k)$ 는

$$(I - J^*(\theta(k)) \cdot G(\theta(k))) \cdot Z(\theta(k)) \quad \text{이다.}$$

#### 방법 2.

앞의 방법에서는  $J$  값을 매번 계속하여 구해야 하므로 매우 많은 계산 시간과 양을 필요로 하게된다. 따라서 본 방법에서는 매번 반복하지 않고 간헐적으로  $J$ 를 구하고자한다. 이때,  $J$ 의 계산 구간은 19)에서는 일정한 간격으로 제한하고있다. 한편 계산 구간을 과거의 오차 크기에 따라 조정하는 방법을 이용할 수 있을것으로 사료된다. 본 논문에서는 전자의 방법을 방법 2-1, 후자의 방법을 방법 2-2라고 하기로한다.

$$\begin{aligned} \theta(k+1) &= \theta(k) + H(k) \cdot \Delta r(k) \\ &\quad + H(k) \cdot G(r(k) - f(\theta(k))) + Re(\theta(k)) \quad (9) \end{aligned}$$

$$H(k) = \begin{cases} J(\theta(1)) & \text{for } k=1, 2, 3, \dots, m(2)-1 \\ J(\theta(m(2))) & \text{for } k=m(2), m(2)+1, \dots, m(3)-1 \\ J(\theta(m(3))) & \text{for } k=m(3), m(3)+1, \dots, m(4)-1 \\ \vdots & \vdots \\ \vdots & \vdots \end{cases}$$

식 9)에서 방법 2-1의 경우  $m(i)$ 는 정수이며 방법 2-2의 경우는 식 (10)과 같다.

$$\begin{aligned} m(i) &= m(i) - 1 \quad \text{if } \rho \leq \epsilon \quad (10) \\ m(i) &= \frac{1}{\sqrt{\sum_{k=1}^{m(i)} e^2(k)}} \quad \text{if } \rho > \epsilon \end{aligned}$$

위에서 기술한 방법 1. 및 2. 외에도 식 1)의 비선형 방정식을 Newton 방법에 의하여 풀고자 하는 데서 여러가지 방법을 찾을 수 있다. 구체적으로 (i+1) 번째의 관절 각도  $\theta(k+1)$ 을 구하기 위하여

$$r(k+1) = f(\theta(k+1)) \quad (11)$$

을 뉴턴 방법으로 풀어보면 다음과 같다.

$$\begin{aligned} h^{(i)}(k+1) &= h^{(i)}(k+1) + J^*(h^{(i)}(k+1)) \{r(k+1) \\ &\quad - f(h^{(i)}(k+1))\} + Re(h^{(i)}(k+1)) \\ &\quad i=0, 1, 2, \dots, n \\ s(k+1) &= h^{(i)}(k+1) \end{aligned}$$

여기서  $S(k)$ 는  $r(k) = f(\theta)$ 의 해이다.

#### 방법 3. (Newton's method with $n=0$ )

$$\begin{aligned} h^{(i)}(k+1) &= s(k) + J^*(s(k)) \{r(k+1) - f(s(k))\} \\ &\quad + Re(s(k)) \\ s(k+1) &= h^{(i)}(k+1) \end{aligned}$$

이 방법은  $G=-I$  일때의 방법 1.과 일치한다.

#### 방법 4. (Newton's method with $n=1$ )

$$\begin{aligned} h^{(i)}(k+1) &= s(k) + J^*(s(k)) \{r(k+1) - f(s(k))\} \\ &\quad + Re(s(k)) \\ h^{(i)}(k+1) &= h^{(i)}(k+1) + J^*(h^{(i)}(k+1)) \{r(k+1) \\ &\quad - f(h^{(i)}(k+1))\} + Re(h^{(i)}(k+1)) \\ s(k+1) &= h^{(i)}(k+1) \end{aligned}$$

#### 방법 5. (Quasi-Newton's method 1.)

$$\begin{aligned} h^{(i)}(k+1) &= s(k) + J^*(s(k)) \cdot \Delta r(k) + Re(s(k)) \\ h^{(i)}(k+1) &= h^{(i)}(k+1) + J^*(h^{(i)}(k+1)) \{r(k+1) \\ &\quad - f(h^{(i)}(k+1))\} + Re(h^{(i)}(k+1)) \\ s(k+1) &= h^{(i)}(k+1) \end{aligned}$$

여기서는  $J^*$ 를 중복하여 사용하므로 방법 4.에 비해 계산 속도가 빠르다.

방법 6. (Quasi-Newton's method 5.)  

$$h'(k+1) = h(k) + J^*(h'(k)) \cdot \Delta p(k) + R_6(h'(k))$$

$$h''(k+1) = h'(k+1) + J^*(h''(k+1)) \cdot \Delta p(k+1)$$

$$s(k+1) = h''(k+1)$$

여기서는 앞의 방법들과 비교할때 전 단계에서 사용한 자코비안 행렬  $J^*(h'(k))$  를 사용하므로 계산 시간이 줄어든다.

#### 4-2. 사용러이슨 결과 및 검토

이상의 방법들을 비교 검토하기 위하여 그림 2)에 보는 바와같이 3개의 회전 관절을 가지는 미니멀러이더 적용을 시켜 평면상에서 원하는 궤적을 추적하는 사용러이슨을 수행하고 자한다. 2 차원에서 위치를 제어하기 위해서는 2 개의 자유도만으로 가능하기 때문에, 그림 2)의 3 자유도 로봇의 경우에는 1개의 자유도가 여유 자유도로 활용된다. 여유 자유도는 manipulability 를 극대화하도록 이용하였다.

$$z(\theta) = \partial h(\theta) / \partial \theta, \quad W(\theta) = \sqrt{\det J^*(\theta) J^T(\theta)}$$

$W(\theta)$  는 measure of manipulability 이다.

한편  $h(k)$  는 (0.4460, 0.0915) 와 (0.4360, -0.0005) 을 잇는 직선으로 주었으며, 계산 구간 수인 N 은 100 및 25 로 선정하였다. 그림 3)은 3 자유도 로봇이 원하는 경로를 추적할 때의 posture 를 보여준다. 여기서는 경로 오차가 잘 나타나지 않고있다. 표 1)에서는 6가지 방법을 사용한 경우들에 대하여 평균 경로 오차량을 나타내었으며 그림 3)에서 7가지는 방법 1) 및 2)에 대하여 N 을 100 및 25 로 선정하고 이득을 바꾼 경우에 대한 평균 오차량을 나타내었다. 방법 3) 부터 6)까지는 지면상 그림으로 나타내지는 않았으나 오차량의 정도에 비하여 많은 계산량을 필요로 한다. 따라서 정밀도와 계산속도를 고려할때, 자코비안 행렬 가상의 변환의 계산을 일정한 간격을 두어 간헐적으로 구하여 사용하는 방법 2) 가 6가지 방법중 최선으로 생각할 수 있다. 특히 방법 2-2는 [9]에서 제시하지 않은 것으로서, 오차량에 따라 계산구간 수를 조절할 수 있으므로 경우에 따라서는 계산량을 상당히 줄일 수가있다. 여기서 주목할 것은 상기 결과가 [9]에서 제시한 Non-redundant 로봇에 대한 결과와 매우 일치 한다는 것이다. 그렇지만 off-line 제어시에는 계산 속도는 문제가 되지 않으므로, 여러번 반복적으로 구하는 방법 4) 등 적절한 방법을 선택하여 사용할 수 가있다.

#### V. 결론

본 논문에서는 여유 자유도를 갖는 로봇의 기구학적 제어를 위하여 필요한 수치 알고리즘을 검토, 분석하였다. 특히, 자코비안 행렬 및 가상의 변환을 수치 반복적으로 구하는 방법을 실험하였으며, 이를 자코비안 행렬 방정식의 일반해를 구하는 6 가지 방법 의 비교 연구에 적용하였다.

또한 비교연구결과, 계산 속도 및 경로 오차를 기준으로 자코비안을 간헐적으로 구하는 방법이 다른 방법보다 유효하였다. 그렇지만, 본 논문 결과가 일반적 이기 위해서는 모든 가능한 경로에 대하여 수학적으로 오차량의 bound 를 구하여 비교하여야 할 것이다. 이는 어려운 문제로 사료된다.

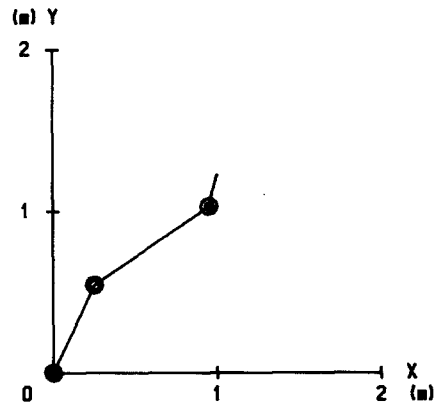


Fig.2 3-link manipulator

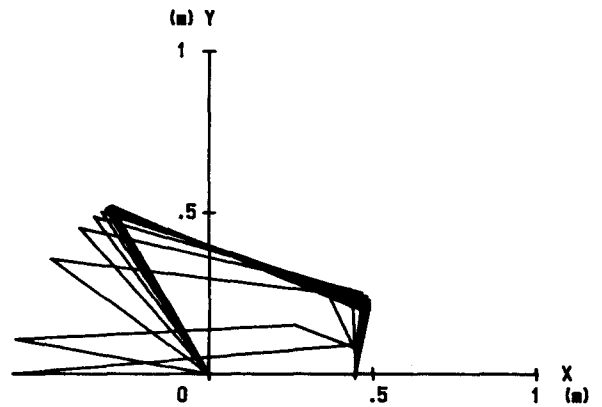


Fig.3 Simulation results

Table1. Average Errors

	Method1	Method2	Method4	Method5	Method6
AE1	2.61E-5	1.72E-4	3.23E-5	8.30E-5	2.27E-5
AE2	2.65E-5	2.20E-4	3.58E-5	1.12E-5	3.48E-5

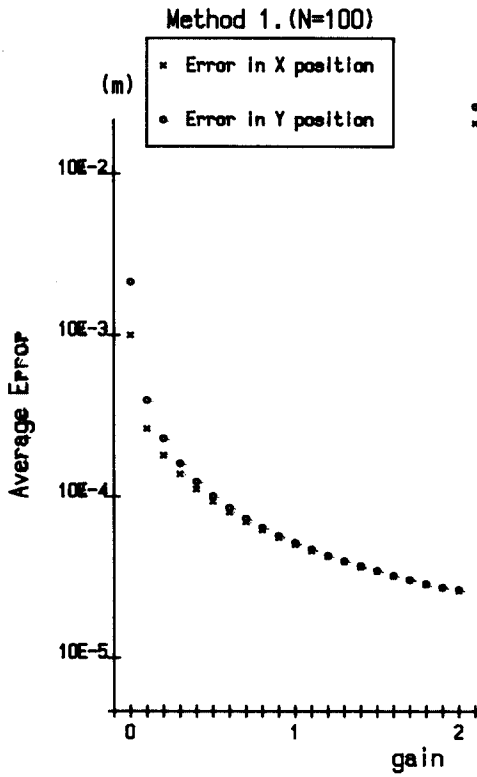


Fig.4 Average Error and Gain

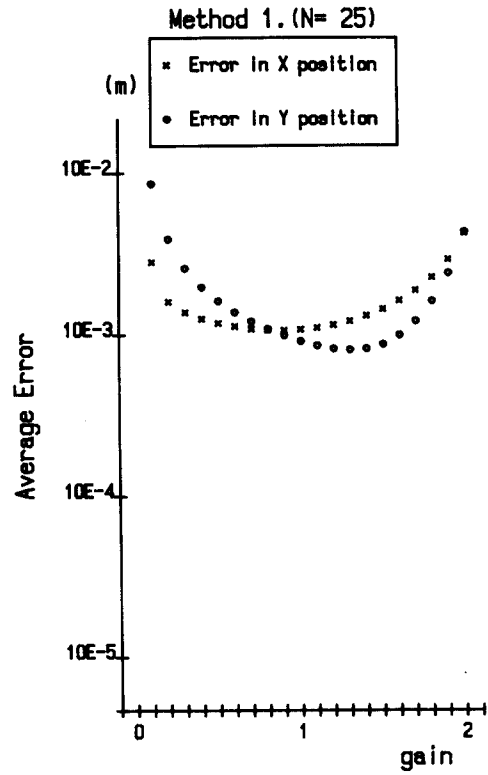


Fig.5 Average Error and Gain

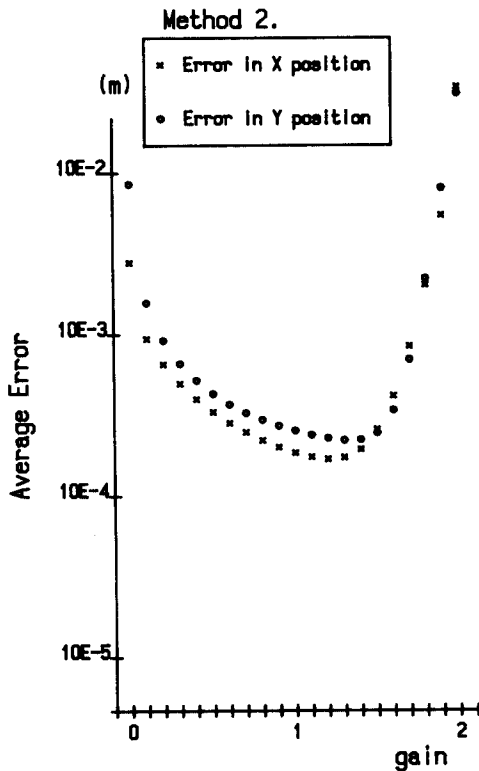


Fig.6 Average Error and Gain  
( N=100,  $m(i+1)-m(i)=4$  )

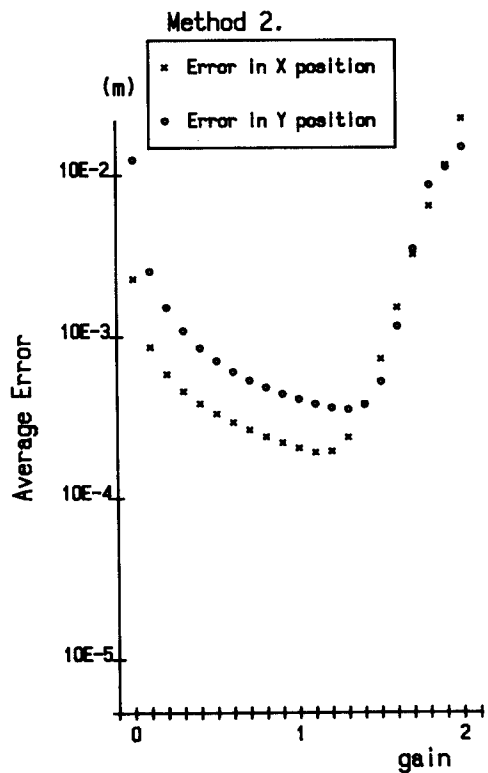


Fig.7 Average Error and Gain  
( N=100,  $m(i+1)-m(i)=6$  )

참고 문헌

1. R. P. Paul, Robot Manipulators: Modeling, Programming and Control, The MIT Press, Cambridge, Mass., 1981.
2. T. Yoshikawa, "Analysis and Control of Robot Manipulators with Geometrical", The First International Symposium of Robotics Research, pp. 704-711, 1984.
3. H. Nakafusa, "Analysis and Control of Articulated Robot Arms with Redundancy", Proc. 8th IFAC World Congress, XIV, pp. 79-82, Aug., 1982.
4. A. A. Mariajewski and S. A. Orin, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments", The International Journal of Robotics Research, Vol. 4, No. 3, pp. 107-117, Fall, 1985.
5. L. T. Wang and P. Rovati, "Recursive Computation of Kinematic and Dynamic Equations for Mechanical Manipulators", IEEE Journal of Electric and Automation, Vol. RA-1, No. 3, pp. 112-121, 1985, 9.
6. D. E. Orin and W. W. Schrader, "Efficient Jacobian Determination for Robot Manipulators", The First International Symposium of Robotics Research, pp. 724-734, 1984.
7. S. Barnett, Matrices in Control Theory with Applications to Linear Programming, pp. 120-136, Van Nostrand Reinhold Company, London, 1971.
8. D. E. Whitney, "Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", Trans. ASME Ser. B, 94, pp. 303-309, 1972.
9. J. Furusho and S. Orinshi, "An Efficient Approach for Solving the Inverse Kinematics of Manipulators", 15th ISIR, Vol. 2, pp. 1051-1059, 1985.