

매니퓰레이터 로봇의 역운동기하학 해석 및 시뮬레이션

I. 서론

대부분 매니퓰레이터 로봇은 작업 수행 시 구동기의 명령이 실행되는 관절 공간(joint space)의 좌표계 시스템에 의존한다. 그렇기 때문에 작업 공간에서의 작업 계획(task space)은 기준 각도 등의 변수를 계산하기 전에 액추에이터 공간에서의 적절한 좌표 변환이 필요하다^[1]. 본 논문에서는 매니퓰레이터 로봇의 작업 공간에서의 이동 궤적을 관절 공간에서의 궤적으로 변환하는 방법에 대해 다룬다. 대부분의 매니퓰레이터 로봇은 작업도구가 로봇의 가장 끝 지점에 위치한다. 이를 엔드이펙터(end-effector)라고 한다. 엔드이펙터는 로봇의 관절운동에 의해 위치이동과 회전이동이 결정된다. 이러한 로봇이 작업을 하도록 설정하기 위해서는, 엔드이펙터가 원하는 위치로 정확히 움직여야 한다. 이를 위해서 로봇을 제어하기 위해서는 운동기하학(kinematics)

엔드이펙터는 로봇의 가장 끝 지점으로, 주로 작업도구가 장착되는 부분이다. 엔드이펙터의 정확한 운동을 위해 로봇을 제어하기 위해서는 운동기하학으로부터 계산해야 한다. 운동기하학은 정운동기하학과 역운동기하학으로 나뉜다.

으로부터 계산을 해야 한다. 운동기하학은 정운동기하학(forward kinematics)과 역운동기하학(inverse kinematics)으로 나뉘는데, 정운동기하학은 로봇의 각 관절의 운동에 따라 엔드이펙터가 어떻게 움직이게 되는지 계산하는 방법이고, 역운동기하학에서는 엔드이펙터가 원하는 위치로 움직이기 위한 관절 각도의 변화를 계산하는 방법이다. 매니퓰레이터 로봇의 운동을 제어하거나 작업 계획을 하기 위해서 역운동기하학은 필수적이라고 할 수 있다.

본 논문에서는 매니퓰레이터 로봇의 역운동기하학 해석 및 시뮬레이



이진영
명지대학교 전기공학과



이범주
명지대학교 전기공학과



션을 위해 먼저 로봇의 운동기하학 모델을 결정하고, 정 운동기하학을 풀이한 후 역운동기하학을 해석한다. 마지막으로 MATLAB을 이용하여 실험로봇의 시뮬레이션을 하고 결과를 도출한다.

II. 역운동기하학 식 유도 및 해석

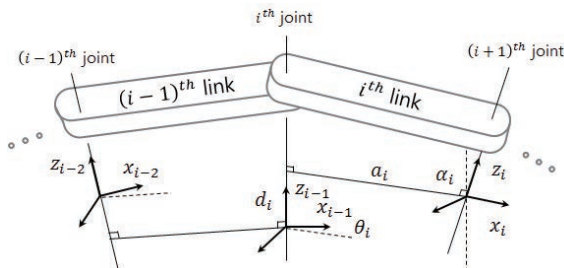
역운동기하학을 푸는 대표적인 방법은 자코비안 행렬을 이용하는 방법^[2]과 기하학을 이용한 닫힌 형태(closed form)로 구하는 방법이 있다^[3-5]. 관절각을 닫힌 형태로 구하는 방법은 반복 과정이 있는 역 운동기하학보다 계산이 빠르게 된다는 장점이 있으나, 이 방법은 로봇의 구조가 바뀔 때 마다 운동기하학 식을 유도해야 하는 단점이 있다. 본 논문에서는 자코비안을 이용하여 역운동기하학 기법에 대해 다룬다.

로봇 모델을 해석하기 위한 순서로, 각 관절의 좌표계를 설정한 후 정운동기하학을 풀이한다. 정운동기하학 방정식에서 자코비안 행렬을 도출하여 로봇 제어를 위한 정보를 추출하는 것이 목적이다.

1. 매니플레이터 로봇의 운동기하학 모델

매니플레이터 로봇의 시뮬레이션을 위해 가장 먼저 해야 할 것은 관절 좌표계를 설정하는 것이다. 본 논문에서는 D-H 표기법(Denavit-Hartenberg notation)을 이용하여 로봇의 각 링크마다 좌표계를 부착하고, 로봇의 초기 위치를 설정하였다.

엔드이펙터의 위치와 회전은 각 관절의 변화에 의해 영향을 받으므로 <그림 1>과 같이 표현하였다. D-H 표기법은 각 좌표계의 z축을 각 관절의 운동축에 설정한다. 좌표계의 중심은 z축 위에 임의로 설정하며, x축과 y축



<그림 1> D-H 표기법에 따른 좌표계 설정

<표 1> 매니플레이터 로봇의 D-H 파라미터

| 링크(i) | q_i | d_i | l_i | a_i |
|-------|----------------------|-------|-------|-------|
| 1 | $q_1 + q_{1,offset}$ | d_1 | l_1 | a_1 |
| 2 | $q_2 + q_{2,offset}$ | d_2 | l_2 | a_2 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| n | $q_n + q_{n,offset}$ | d_n | l_n | a_n |

은 오른손 법칙을 이용하여 결정한다. 이 후의 좌표계는 D-H 파라미터를 결정하며 반복적인 방법을 이용하여 결정하게 된다. 다음은 D-H 표기법을 기반으로 D-H 파라미터(Denavit-Hartenberg parameter)를 도출한 것이다.

D-H 파라미터의 각 변수 q , d , l , a 는 <그림 1>의 각 부분에 결정된 것과 같다. q 는 이전 링크의 x축과 다음 링크의 x축 사이의 각도이며, d 는 이전 좌표계의 원점과 다음 좌표계의 x축의 연장선과 이전 좌표계의 z축이 만나는 점과의 거리이고, l 은 이전 좌표계의 z축과 다음 좌표계의 z축 사이의 최단 직선의 길이이다. 이 직선이 다음 좌표계의 x축을 결정한다. 마지막으로, a 는 이전 좌표계의 z축과 다음 좌표계의 z축 사이의 각도이다.

2. 정운동기하학(Forward kinematics) 풀이

역운동기하학 해석에서는 정운동기하학이 선행되어야 한다. 이는 로봇이 목표위치로 가기 위해 움직이는 동안 엔드이펙터의 위치와 회전이 계산에 따라 항상 변하기 때문이다. 정운동기하학의 풀이는 II.1절에서 설정한 D-H 표기법을 이용하여 동차 변환 행렬(homogeneous transformation matrix)을 도출하면 간단히 계산할 수 있다^[5,6]. 동차 변환행렬은 강체의 운동을 기술할 수 있는 행렬로, 다음과 같은 형태이다.

$$A = \begin{bmatrix} R d \\ 0 \ 1 \end{bmatrix} \quad \text{식 1}$$

식 1에서, R 은 3×3 크기의 회전행렬이며 d 는 3×1 크기의 위치벡터, 0 은 1×3 크기의 영행렬이다. 따라서 동차 변환행렬은 4×4 크기의 정사각행렬로 정의된다. 식 2



는 <표 1>의 D-H 파라미터와 식 1을 이용하여 i 번째 링크에서 $i+1$ 번째 링크로의 동차변환행렬이다.

$$A_i = Rot(z, q_i) Trans(z, d_i) Trans(x, l_i) Rot(x, a_i) \quad \text{식 2}$$

식 2에서, $Rot(a, b)$ 는 회전 변환을 나타내는 동차변환행렬을 나타내며 a 는 회전축, b 는 회전각도를 의미한다. $Trans(a, b)$ 는 이동을 나타내는 동차변환행렬이고, d 는 이동 거리를 표현한다. A_i 를 식으로 표현하면 다음과 같다.

$$A_i = \begin{bmatrix} c\theta_i - s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i - c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{식 3}$$

식 3에서, $c\theta_i/s\theta_i$ 는 $\cos\theta_i/\sin\theta_i$ 를, $c\alpha_i/s\alpha_i$ 는 $\cos\alpha_i/\sin\alpha_i$ 를 의미한다. 식 2와 식 3을 활용하여 기준 좌표계에서 엔드 이펙터의 위치를 계산할 수 있다. 즉, 기준좌표계에서 n 번 좌표계의 좌표변환을 나타내는 식이다. 각 링크별 D-H파라미터에서 추출한 동차변환행렬을 기준부터 n 번까지 순서대로 곱하여 계산할 수 있으며, 식 4와 같다.

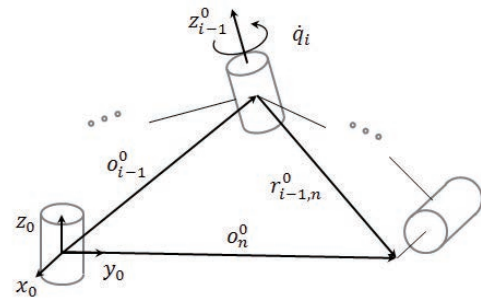
$$T_n^0 = A_1 A_2 \cdots A_n \quad \text{식 4}$$

3. 자코비안(Jacobian) 행렬 도출

정운동기하학 방정식은 직교 공간(cartesian space)에서 표현한 엔드이펙터의 자세와 관절 좌표 사이의 공간 함수로 정의된다^[1,6]. 따라서, 엔드이펙터의 속도와 관절 속도와의 관계는 이 함수의 도함수로 나타내며 이를 자코비안이라고 한다. 자코비안은 로봇의 운동을 제어하는 중요한 정보 중 하나이다. 이를 수식으로 표현하면 식 5와 같다.

$$\begin{bmatrix} v_{0,n}^0 \\ \omega_{0,n}^0 \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad \text{식 5}$$

식 5에 의거하여 엔드이펙터는 관절들에 의해 운동을 하게 된다. <그림 2>는 회전관절의 운동에 따른 엔드이펙터의 운동을 설명한다. 여기서, z 는 해당 좌표계의 z 축 단위벡터이며, o_n^0 은 기준 좌표계의 원점에서부터 n 번 좌표



<그림 2> 회전관절의 운동에 따른 엔드이펙터의 운동

계의 원점까지의 거리이다.

$$\Delta R = \begin{bmatrix} a_1^2 + c_\delta(1-a_1^2) & (1-c_\delta)a_1a_2 - s_\delta a_3 & (1-c_\delta)a_3a_1 - s_\delta a_2 \\ (1-c_\delta)a_2a_1 - s_\delta a_3 & a_1^2 + c_\delta(1-a_1^2) & (1-c_\delta)a_2a_3 - s_\delta a_1 \\ (1-c_\delta)a_3a_1 - s_\delta a_2 & (1-c_\delta)a_3a_2 - s_\delta a_1 & a_1^2 + c_\delta(1-a_1^2) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{식 9}$$

<그림 2>로부터 자코비안 행렬은 다음과 같은 구해진다.

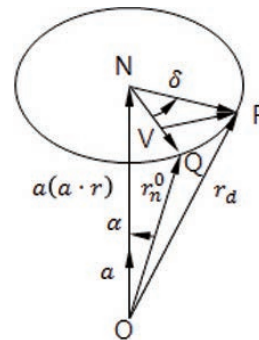
$$J = \begin{bmatrix} z_0^0 \times (o_n^0 - o_0^0) & z_1^0 \times (o_n^0 - o_1^0) & \cdots & z_n^0 \times (o_n^0 - o_{n-1}^0) \\ z_0^0 & z_1^0 & \cdots & z_{n-1}^0 \end{bmatrix}$$

4. 역운동기하학(Inverse kinematics) 풀이

정운동기하학을 풀면 목표위치에서 현재위치의 차이를 구할 수 있다.

$$\begin{aligned} \Delta p &= p_d^0 - p_n^0 \\ \Delta R &= R_n^0 R_d^0 \end{aligned} \quad \text{식 6}$$

위 수식에서, p_d^0 는 목표위치를 의미하며, R_d^0 는 목표회전을 의미한다. 이 기준좌표계에서 n 번 좌표계까지의 회전행렬로 나타나는 회전변환을 하나의 회전축과 해당 회전각도 만큼의 변화로 변환하여 나타낼 수 있다.



<그림 3> 기하학적 해석을 위한 예시 그림



$$r_d = \Delta R r_n^0 \quad \text{식 7}$$

r_d 은 기하학적인 해석 방법을 이용하여 해석할 수 있다. <그림 2>에서 \vec{ON} 은 좌표계의 z축방향 벡터이고, \vec{NV} 는 계산을 위해 \vec{ON} 벡터에서 r_n^0 벡터로 수직으로 표현한 벡터이다. \vec{VP} 벡터 또한 계산을 위해 \vec{NQ} 벡터 위에서 r_d 벡터의 끝점 P 로 표현한 벡터이다.

$$r_d = \vec{ON} + \vec{NV} + \vec{VP} \\ = (aa^T + \cos\delta(I - aa^T) + \sin\delta\hat{a})r_n^0 \quad \text{식 8}$$

식 7과 식 8로부터 회전행렬은 식 9와 같이 구할 수 있다. 여기서, 회전각도 δ 와 회전 축 벡터 a^n 은 ΔR 의 대각선방향의 원소들로부터 다음과 같이 계산되어진다.

$$\delta = \cos^{-1}\left(\frac{1}{2}(r_{11} + r_{22} + r_{33} - 1)\right) \\ a^n = \frac{1}{2\sin\delta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad \text{식 10}$$

a^n 은 n 번째 관절에 설정한 축을 기준으로 계산된 벡터이므로 이를 기준 좌표계에서 계산된 값을 나타낼 수 있도록 변환해야 한다.

$$a^0 = R_n^0 a^n \quad \text{식 11}$$

회전행렬을 회전축과 각도로 변환 후의 목표위치에서 현재위치의 차이를 구하기 위해 오차 벡터를 정의하면 다음과 같이 표현할 수 있다.

$$\Delta x = \begin{bmatrix} \Delta p \\ a^0 \delta \end{bmatrix} \quad \text{식 12}$$

목표 위치까지 도달하기 위한 최적의 각도를 계산하기 위해 비용함수를 설정하고 오차를 구할 수 있다.

$$e = W_x \Delta x \quad \text{식 13}$$

자코비안을 이용한 방법은 반복적인 계산이 필요하다. 그러므로 에러가 수렴하기 위한 가중치 또한 반복적으로 계산해야 한다. 에러가 수렴하기 위한 가중치는 Samuel R.B weighting 기법을 사용하여 구할 수 있다^[7]. 이것을 α 라고 하면 식 14와 같이 표현할 수 있다.

Algorithm 1 Inverse kinematics

```

/* initialize variables */
p_d^0 ← [x_d y_d z_d]^T;
R_d^0 ← Rot(φ_roll, φ_pitch, φ_yaw);
/* start iteration */
while (k < N) or (|e| < ε) do
/* current posture */
(R_n^0, p_n^0) = T(q_k);
/* posture error */
Δp = p_d^0 - p_n^0
ΔR ← R_n^0 R_d^0; // R_d^0 = R_n^0 ΔR
(a^n, δ) ← f(ΔR); // calculate axis-angle rep.
a^0 ← R_n^0 a^n; // change the coordinate sys.

Δx ← [Δp; a^0 δ];
e ← W_x Δx;
/* update */
α ← e · (JJ^T e) / (JJ^T e) · (JJ^T e); // Samuel R. B weighting
q_{k+1} ← q_k + α J^T e;
end while

```

<그림 4> 역운동기하학 알고리즘

$$\alpha = \frac{e \cdot (JJ^T e)}{(JJ^T e) \cdot (JJ^T e)} \quad \text{식 14}$$

현재의 각도에 가중치를 적용한 에러를 더하여 최적화된 다음 이동 각도를 구할 수 있다. 수식은 다음과 같이 표현한다.

$$q_{k+1} = q_k + \alpha J^T e \quad \text{식 15}$$

위의 과정을 에러가 사용자가 설정한 반복회수만큼 반복해서 계산하여 에러가 기준치 이내로 수렴하면 역운동기하학의 계산을 종료할 수 있다. <그림 4>는 앞선 내용들을 이용한 알고리즘이다.

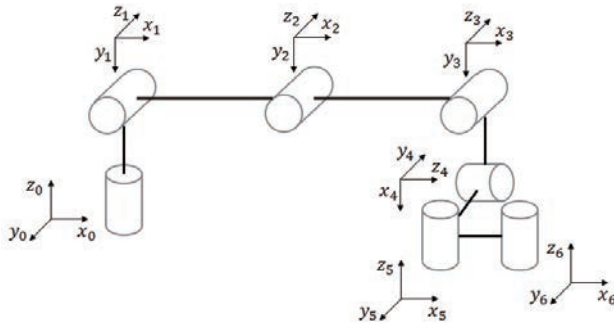
III. 실험 결과

II장의 과정을 이용하여 MATLAB으로 매니플레이터 로봇의 엔드이펙터가 목표위치까지 이동하는 과정을 시뮬레이션하였다. 시뮬레이션의 결과는 엔드이펙터가 목표위치까지 도달하기 위한 각 관절의 각도변화와 목표위치에서 현재위치의 차이를 나타내는 에러벡터를 확인하였고, 에러가 수렴하는 것을 norm을 이용하여 그래프로



〈표 2〉 로봇의 초기 관절각도 및 이동목표 관절각도

| 구분 | Link 1 | Link 2 | Link 3 | Link 4 | Link 5 | Link 6 |
|-------|--------|--------|--------|--------|--------|--------|
| 초기 각도 | 90° | 0° | 0° | 90° | -90° | 90° |
| 목표 각도 | 100° | 10° | 10° | 100° | -80° | 100° |



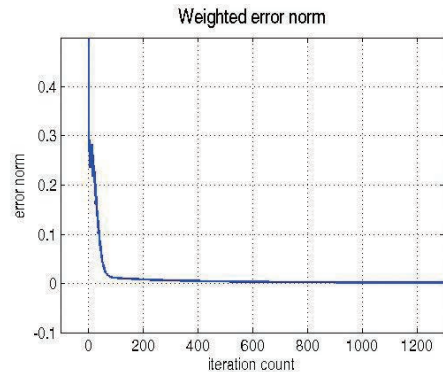
〈그림 5〉 실험 로봇의 초기 형상

확인할 수 있었다.

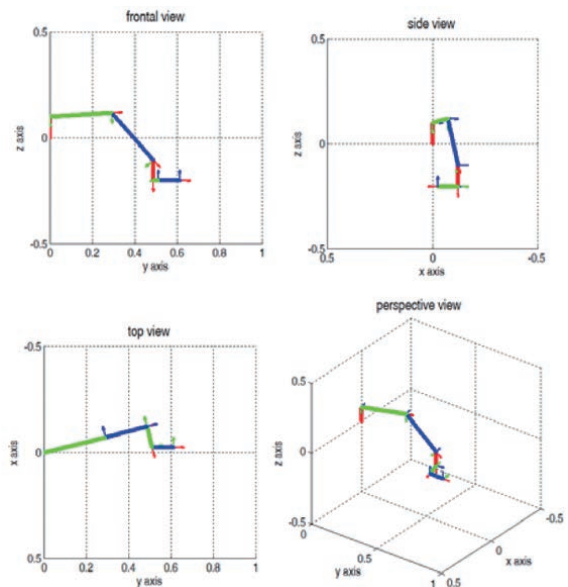
〈그림 5〉은 시뮬레이션을 하기 위한 로봇의 초기 형상을 나타내고 있다. 자코비안을 이용한 역운동기하학 풀이는 로봇의 초기 형상이 자코비안의 특이점에 있으면 풀이가 불가능한 경우가 되기 때문에 초기 형상을 지정해 주어야 한다. 다음은 로봇의 초기 관절각도와 엔드이펙터가 목표한 위치에 도달했을 때의 관절각도를 정리한 표이다.

시뮬레이션 과정은 목표 관절각도를 정운동기하학에 대입하여 동차변환행렬을 구하는 것으로 시작한다. 그 후 매니플레이터 로봇이 초기상태에서 설정한 목표로 도달할 수 있도록 계산하기 위해 〈그림 4〉의 알고리즘을 이용하면서, 반복할 때 마다 그 당시의 이동한 관절 각도를 업데이트하면서 계산을 한다. 이 과정을 거치면서 목표에서 현재의 에러값이 수렴할 때 까지 반복한다. 에러값이 원하는 범위 이내로 수렴한 것이 1301번째 이므로 시뮬레이션을 종료하였다. 에러 범위를 설정하기 위해 기준 범위를 $1e-6$ 으로 설정하였다. 에러 수렴 판별은 에러행렬을 제공하여 크기를 구하여 판별하였다. 이 값을 기반으로 에러가 수렴하는 과정을 확인하기 위해 그래프를 도시하였다.

〈그림 6〉의 결과를 통해 에러의 수렴을 확인할 수 있다. 에러가 초반에는 빠르게 수렴하고, 후반에 느리게 수렴하게 되는데, 이는 〈그림 4〉의 알고리즘이 Newton-



〈그림 6〉 에러 수렴 판별 그래프



〈그림 7〉 6축 로봇 시뮬레이션

Raphson방법을 기반으로 한 방법이기 때문이다. 〈그림 7〉은 시뮬레이션의 결과로 도시된 6축 매니플레이터 로봇의 형상을 각각 다른 시점에서 바라본 것이다.

VI. 결론

본 논문에서는 매니플레이터 로봇의 역운동기하학 해석을 위하여, 엔드이펙터의 선속도 및 각속도와 각 관절각도의 관계에 의한 자코비안 행렬을 이용하는 방법을 소개하였다. 이 방법은 관절각을 닫힌 형태로 구하는 방법에 비해 연산속도는 다소 느리지만, 로봇이 달라질 때 마



다 수식을 새롭게 유도할 필요가 없다는 장점을 가지고 있다. 6축 매니퓰레이터 로봇의 모델에 기반하여 시뮬레이션을 하였으며 목표 자세의 오차가 허용범위 이내로 수렴하는 것을 확인하였다.

참고 문헌

- [1] Mark W. Spong et al. "Robot Modeling and Control", JOHN WILEY & SONS, INC., 2006
- [2] Aaron D'Souza et al. "Learning Inverse Kinematics", In proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct.29~Nov.03, 2001
- [3] 이종경, 이지영, 성영휘, "4지 로봇의 역기구학 해", 정보 및 제어 학술대회 논문집, pp.136~137, 2014
- [4] 이종규, 이상룡, 이춘영, 양승한 "2개의 자유도를 가진 병렬 매니퓰레이터의 기구학 해석", 한국정밀공학회지, pp.64~71, 2012
- [5] 배영걸, 정 슐, "바닥작업이 가능한 양팔 서비스 로봇의 기구학 설계, 제작 및 제어", 전자공학회 논문지, pp.203~211, 2013
- [6] Frank L. Lewis et al. "Robot Manipulator Control Theory and Practice", Marcel Dekker, Inc., 2004
- [7] Samuel R. Buss "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares method", IEEE Journal of Robotics and Automation, 2009



이진영

- 2015년 2월 명지대학교 전기공학 학사
- 2015년 3월~현재 명지대학교 전기공학 석사과정

〈관심분야〉
로봇제어, Motion planning, 휴머노이드 로봇



이범주

- 2004년 3월~2008년 8월 한국과학기술원 전기전자공학 박사
- 2002년 3월~2004년 2월 한국과학기술원 전기전자공학 석사
- 1998년 3월~2002년 2월 연세대학교 전기전자공학/기계공학 학사

〈관심분야〉
휴머노이드 로봇 보행 기술 연구, 감속기 연구, 로봇 메커니즘 설계 및 개발, 로봇 제어알고리즘 연구 및 개발