

기구학적 최적 제어를 사용한 자유 부유 매니플레이터의 여유도 해결

Redundancy Resolution for Free-Floating Manipulators Using Kinematic Optimal Control Approach

김 용 민*, 김 병 국
(Yong-Min Kim¹ and Byung-Kook Kim¹)

¹Korea Advanced Institute of Science and Technology

Abstract: An efficient sequential computation algorithm of kinematic optimal control is suggested for redundancy resolution of free-floating manipulators. Utilization of minimum principle usually requires involved and tedious procedure of differentiation of Hamiltonian. Due to the constraints of momentum conservation, it is not easy to get exact differential equations of boundary value problem for even relatively simple free-floating manipulator models. To overcome this difficulty, we developed an effective sequential algorithm for the computation of terms appeared in the differential equations. The usefulness of suggested approach is verified by simulation of a planar 3-joints free-floating manipulator.

Keywords: redundancy resolution, free-floating manipulator, optimal control

I. 서론

매니플레이터(manipulator)의 엔드 이펙터(end-effector)의 운동 궤적이 주어졌을 때, 관절 공간(joint space)의 자유도가 이 운동을 실현하는데 필요한 자유도보다 큰 경우, 우리는 그러한 매니플레이터는 기구학적 여유도(kinematic redundancy)를 갖는다고 말한다. 이 때, 주어진 엔드 이펙터의 궤적을 실현할 수 있는 매니플레이터의 관절들의 속도 값은 일반적으로 무한히 많은 조합이 존재하게 되는데, 이 중 하나로 관절의 운동을 결정해주는 것을 여유도 해결(redundancy resolution)이라 하며, 그 방법에 대해서는 이제까지 다양한 연구가 수행된 바 있다[1,2].

이러한 여유도 해결 방법은 크게 즉각적인 정보만을 이용하는 국소 최적화(local optimization) 방법[2-4]과 전체 시간의 정보를 이용하는 전역 최적화(global optimization) 방법[5,6]으로 나눌 수 있다. 전자는 대개 실시간 피드백 제어의 형태로 구현되며, 기본적으로 자코비언(Jacobian)의 의사 역행렬(pseudo-inverse)을 사용하게 된다. 이 때 자코비언의 영공간에는 목적에 따라 특정한 동작을 취하게 만드는 성분을 투영시킬 수 있는데 Yoshikawa [3]는 조작도(manipulability)를 최적화하는 성분을, Hollerbach [2]와 Kazeroonian [4]은 토크를 최소화하는 성분을 투영하는 제어 법칙을 각기 다른 관점에서 유도하여 제시한 바 있다. 이와 같은 국소 최적화 방법은 계산량이 적기 때문에 실시간 제어의 형태로 구현이 가능하다는 것이 장점이지만 긴 엔드 이펙터 궤적에 대해 적용할 경우에는

관절의 속도가 급증하는 불안정한 상태로 빠지는 문제가 있다[7].

이에 비해 후자의 전역 최적화 방법은 많은 계산이 필요하기 때문에 실시간 제어가 요구되는 분야에는 적합하지 않다. 그러나 이 방법은 긴 엔드 이펙터 궤적에 대해서도 구동기에 부담을 적게 주는 궤적을 얻을 수 있으므로 정해진 동작을 반복하는 일이 잦은 산업용 로봇이나 우주 공간에서의 임무처럼 사전에 충분한 시뮬레이션을 통해 동작의 적합성을 검증하는 일이 중요한 응용 분야에서 장점을 갖는다. Nakamura [5]는 Pontryagin의 최소 원리(minimum principle)를 적용하여 매니플레이터의 여유도 해결 문제를 경계치 문제(boundary value problem)으로 변환한 뒤 이를 푸는 접근 방법을 제시하였고, Kim [6]은 Nakamura [5]의 접근 방식을 최적 제어 이론의 형식에 맞추어 재정리한 결과를 보여주었다.

이상의 연구에서는 매니플레이터의 몸체가 고정된 경우를 다루었지만 인공위성에 장착된 매니플레이터처럼 몸체가 고정되어 있고 공간 상에서 떠다니는 경우에는 팔 관절의 움직임에 따라 몸체도 같이 움직이게 되므로 이에 대한 고려가 필요하다. 특별히 몸체의 운동을 구동기로 제어하지 않는 경우를 자유 부유 매니플레이터(free-floating manipulator)라 부르는데 이러한 자유 부유 매니플레이터의 여유도 해결 문제는 몸체가 고정된 경우에 비해 많은 연구가 이루어지지 않았다. Umetani [8]는 일반화된 자코비언(generalized Jacobian)을 사용하면 몸체가 고정된 매니플레이터에서 사용되는 의사 역행렬 방법이 자유 부유 매니플레이터에도 통용 가능함을 보여주었고, Yoshida [9]는 일반화된 자코비언을 사용하면 Hollerbach [2]가 제시한 관절 토크를 국소 최적화하는 방법을 우주 매니플레이터에도 적용시키는 것이 가능함을 보여주었다. 앞서 언급한 Nakamura [5]와 Kim [6]의 연구에서 사용된 것과 유사한 전역 최적화 기법이 자유 부유 매니플레이터에

* 책임저자(Corresponding Author)

논문접수: 2011. 1. 6., 수정: 2011. 3. 22., 채택확정: 2011. 5. 6.

김용민, 김병국: 한국과학기술원

(ymkim@rtcl.kaist.ac.kr/bkkim@kaist.ac.kr)

※ 본 연구는 (고려대학교 로봇 자율 주행기술 전문인력양성센터를 통한) 지식경제부/ 한국산업기술진흥원 융복합형로봇전문인력양성 사업의 지원으로 수행되었음.

적용된 사례로는 Agrawal [10]의 연구가 있다. 한편 Huang [11]은 경계치 문제를 푸는 대신 유전 알고리즘(genetic algorithm)을 사용하여 자유 부유 매니플레이터 관절 운동의 전역 최적해를 탐색하는 방법을 소개한 바 있다. 그러나, 여기서 다루어진 문제는 주어진 시작, 종료 조건과 구동기 제약 조건을 만족하면서 저크(jerk)를 최소화하는 관절 운동을 생성하는 것으로, 본 논문에서 다루고자 하는 여유도 해결 문제와는 차이가 있다.

본 논문에서는 전역 최적화 방법을 자유 부유 매니플레이터 모델에 적용할 때 경계치 문제를 구성하는 미분방정식을 정확하게 얻기 위한 알고리즘을 제시하고자 하였다. 전역 최적화 방법을 다룬 위의 연구 사례들[5,6,10]에서 나타나는 수식의 전개 과정은 모두 다르지만, 일련의 유도 과정의 결과로써 경계치 문제가 형성되고 이를 풀어서 주어진 엔드 이펙터 궤적을 만족하는 관절 궤적을 얻게 된다는 점은 동일하다. 그런데 모델이 복잡한 경우에는 경계치 문제를 구성하는 각각의 미분 방정식을 명확하게 얻기가 어렵다. 유한 차분 근사법(finite difference approximation)을 사용하여 근사적으로 이식들을 구할 수도 있지만 오차가 누적되어 잘못된 해를 얻거나, 수치 적분을 수행하는 과정에서 계산 값이 발산할 위험이 있으므로 정확한 식을 구할 필요가 있다. Nakamura [5]와 Agrawal [10]의 연구에는 이 문제에 대한 언급이 없고, Kim [6]은 자코비언의 의사역행렬을 미분하는 어려움을 회피하기 위해 원래의 경계치 문제를 보다 쉽게 계산이 되는 형태로 변환하는 접근 방식을 취하였다. 그러나 자유 부유 매니플레이터의 경우에는 일반화된 자코비언이 사용되는데 이것이 이미 운동량 보존 제약 조건이 반영되어 복잡한 형태로 되어 있기 때문에 Kim [6]이 사용한 방식으로도 이 문제를 해결할 수는 없다. 본 논문에서는 자코비언과 각운동량 보존 제약 조건식의 계산 과정을 재귀적인 형태로 계산하는 알고리즘을 제시함으로써 이를 해결하였다.

본 논문의 구성은 다음과 같다. II 장에서는 간결한 수식 표현을 위해 Featherstone [12]에 의해 제안된 공간 벡터 대수(spatial vector algebra)에 대한 간략한 설명과 함께 자유 부유 매니플레이터의 문제에서 나타나는 운동량 보존 제약 조건에 관해 다루었다. III 장에서는 최적 제어 이론의 틀에 맞추어 자유 부유 매니플레이터의 여유도 해결 문제를 정의하고, IV 장에서는 III 장에서 여유도 해결 문제를 경계치 문제로 변환하는 과정에서 등장하는 여러 복잡한 항의 값을 체계적으로 계산하기 위한 알고리즘을 설명하였다. V 장에서는 평면 3관절 자유 부유 매니플레이터 모형에 대해 III, IV 장에서 논의한 방법을 적용한 시뮬레이션 결과를 보였다. 마지막으로 VI 장에서는 결론을 내리고 연구의 내용을 요약하였다.

II. 공간 벡터 대수를 이용한 운동량 보존 제약 조건과 자코비언 계산

공간 벡터 대수(Spatial vector algebra)는 강체로 구성된 다관절 물체계의 운동을 기술하는 표현 방식으로 Featherstone [12]에 의해 제안되었으며, 잘 알려진 Lie 대수를 이용한 표현 방식[13]과 마찬가지로 병진 운동 성분과 회전 운동 성분을 하나의 식에 통합하여 쓰기 때문에 간결하게 식을 서술할 수 있는 것이 장점이다. 그러나 Lie 대수를 이용한 표현 방식은

행렬이나 벡터를 아래 첨자로 넣는 관행 때문에 많은 항이 연관되면 수식이 읽기 어려워지는 문제가 있다. 공간 벡터 대수 표현법은 상세한 내용에서는 Lie 대수를 사용하는 방법과 거의 유사하지만, 좀 더 읽기 쉽기 때문에 본 논문에서는 이를 이용하여 수식을 기술하였다.

i 번째 물체에 부착된 좌표계에 대해 공간 벡터 대수에서는 이 좌표계의 속도를 아래와 같이 회전 성분과 선형 성분을 하나의 벡터로 모아서 표기한다.

$$v_i = [\omega_i^T \quad v_i^T]^T \tag{1}$$

여기서 ω_i 는 회전 운동 성분, v_i 는 병진 운동 성분이다. i 번째 관절의 속도가 \dot{q}_i 이라면 물체의 몸체 좌표계 상에서 표현된 이 물체의 속도는 $S_i \dot{q}_i$ 이 된다. 여기서 S_i 는 관절의 축 방향에 따라 결정되는 벡터로 회전 축이 z 축 방향인 회전 관절의 경우 $S_i = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$, 미끄럼관절(prismatic joint)의 경우 $S_i = [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$ 이 된다. 또한 속도 벡터 v_i 로부터 아래와 같은 확장된 외적(cross product) 행렬을 정의할 수 있다.

$$[v_i \times] = \begin{bmatrix} [\omega_i^\otimes] & 0 \\ [v_i^\otimes] & [\omega_i^\otimes] \end{bmatrix} \tag{2}$$

여기서 $[\omega_i^\otimes]$, $[v_i^\otimes]$ 은 각각 ω_i 와 v_i 로부터 만들어진 스큐 행렬(skew matrix)을 의미한다. 한편, i -번째 좌표에서 j -번째 좌표로의 좌표변환은 아래와 같이 표현된다.

$${}^j X_i = \begin{bmatrix} {}^i R_i & 0 \\ -{}^j R_i [r_{i,j}^\otimes] & {}^j R_i \end{bmatrix} \tag{3}$$

여기서 ${}^i R_i$ 은 i -번째 좌표에서 j -번째 좌표로의 회전행렬, $r_{i,j}$ 은 두 좌표계 원점 사이의 벡터, $[r_{i,j}^\otimes]$ 은 이 벡터로부터 만들어진 스큐 행렬(skew matrix)을 의미한다. 이상의 공간 벡터 대수 표현을 이용하면 전체 시스템의 운동량 총합을 아래와 같은 식으로 표현할 수 있다.

$$\sum_{i=1}^N \sum_{j=1}^i {}^i X_0^T H_i^i X_j S_j \dot{q}_j = \begin{bmatrix} L \\ P \end{bmatrix} \tag{4}$$

여기서 N 은 전체 시스템의 자유도 또는 링크의 개수이다. 편의상 위의 식은 그림 1과 같이 매니플레이터의 팔이 하나

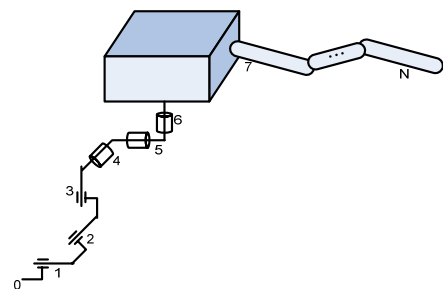


그림 1. 하나의 팔을 갖는 자유 부유 매니플레이터 모형.
Fig. 1. Single-armed free-floating manipulator model.

만 장착되어 있고, 좌표계의 번호는 관성 좌표계를 0번으로 하되 매니플레이터의 바깥 링크 쪽으로 나감에 따라 번호가 증가되도록 할당하는 것으로 가정하였다. 따라서 (4) 식은 관성 좌표계에서 표현된 운동량의 총합이 된다.

(4) 식에서 \mathbf{P} 는 선형운동량의 총합, \mathbf{L} 은 각운동량의 총합으로, 외부로부터 작용하는 힘이나 토크가 없을 때 이 값은 일정하게 보존된다. 이를 운동량 보존 제약 조건(momentum conservation constraint)이라 하며[8], 특별히 초기에 시스템이 정지 상태에 있었을 경우 이 값은 모두 0이 된다. 이후의 논의에서는 편의상 $\mathbf{L} = \mathbf{P} = \mathbf{0}$ 인 경우로 간주하고 식을 전개하였다. \mathbf{H}_i 는 i 번째 물체의 질량 m_i 와 관성모멘트 \mathbf{I}_i 로부터 유도되는 행렬로 아래와 같이 표현된다.

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{I}_i + m_i [\mathbf{c}_i^\otimes] [\mathbf{c}_i^\otimes]^T & m_i [\mathbf{c}_i^\otimes] \\ m_i [\mathbf{c}_i^\otimes]^T & m_i \mathbf{1} \end{bmatrix} \quad (5)$$

여기서 관성모멘트 \mathbf{I}_i 는 i 번째 물체의 무게중심에서 계산된 값이며, \mathbf{c}_i 는 i 번째 좌표계에서 이 무게중심까지의 벡터이다. 또한 $\mathbf{1}$ 은 항등행렬(identity matrix)을 의미한다. 한편, (4) 식의 우변의 형태와 관련해서는 아래와 같은 성질이 성립함이 알려져 있다[12].

$$\sum_{i=1}^N \sum_{j=1}^i (\dots) = \sum_{j=1}^N \sum_{i=j}^N (\dots) \quad (6)$$

(6)의 성질을 (4)에 적용하고 인덱스 i 와 j 를 맞바꾸어 다시 쓰면, 운동량 보존 제약 조건은 아래와 같은 꼴로 정리하여 나타낼 수 있다.

$$\sum_{i=1}^N \mathbf{D}_i \dot{\mathbf{q}}_i = \begin{bmatrix} \mathbf{L} \\ \mathbf{P} \end{bmatrix} \quad (7)$$

여기서, $\mathbf{D}_i = \sum_{j=i}^N {}^j \mathbf{X}_0^T \mathbf{H}_j {}^i \mathbf{X}_i$ 는 아래의 알고리즘1과 같이 재귀적인 방식으로 계산이 가능하다.

Algorithm 1: Computation of \mathbf{D}_i

Step1: Initialization:

$$\mathbf{A}_0 = \mathbf{1} \quad (8)$$

Step 2: Computation of \mathbf{A}_i (1st loop)

For $i=1$ to N

$$\mathbf{A}_i = \mathbf{A}_{i-1} {}^i \mathbf{X}_{i-1}^T \quad (9)$$

End For

Step 3: Computation of \mathbf{D}_i (2nd loop)

For $i=N$ To 1

$$\mathbf{B}_i = \begin{cases} \mathbf{H}_i & \text{if } i = N \\ \mathbf{H}_i + {}^{i+1} \mathbf{X}_i^T \mathbf{B}_{i+1} {}^i \mathbf{X}_i & \text{otherwise} \end{cases} \quad (10)$$

$$\mathbf{D}_i = \mathbf{A}_i \mathbf{B}_i \mathbf{S}_i \quad (11)$$

End For

자유 부유 매니플레이터에 알고리즘1을 적용할 때의 유의

점은 매니플레이터 몸체의 이동을 기술하기 위해 도입된 링크는 무게가 없는 가상의 링크이므로 이들의 질량과 관성 모멘트를 0으로 설정해야 하는 점이다. 예를 들어 그림 1과 같은 모델에서는 1~3번 링크가 무게가 없는 가상의 미끄럼 관절(prismatic joint)이 되고, 4~5번 링크는 무게가 없는 가상의 회전 관절이 되며, 6번 회전 관절 링크의 질량과 관성 모멘트가 위성 몸체의 원래 값이 된다[12]. 한편, 기준 좌표계로부터 엔드 이펙터 좌표계까지의 좌표변환 ${}^E \mathbf{X}_0$ 는

$${}^E \mathbf{X}_0 = \begin{bmatrix} {}^E \mathbf{R}_0 & \mathbf{0} \\ -{}^E \mathbf{R}_0 [\mathbf{r}_E^\otimes] & {}^E \mathbf{R}_0 \end{bmatrix} \quad (12)$$

으로 표현되며[12], 이것을 시간에 대해 미분한 값은,

$$\begin{aligned} {}^E \dot{\mathbf{X}}_0 &= \begin{bmatrix} {}^E \dot{\mathbf{R}}_0 & \mathbf{0} \\ -{}^E \dot{\mathbf{R}}_0 [\mathbf{r}_E^\otimes] - {}^E \mathbf{R}_0 [\mathbf{v}_E^\otimes] & {}^E \dot{\mathbf{R}}_0 \end{bmatrix} \\ &= {}^E \mathbf{X}_N \frac{d}{dt} ({}^N \mathbf{X}_{N-1} \dots {}^1 \mathbf{X}_0) \end{aligned} \quad (13)$$

이다. (13)의 우변에서 괄호로 묶여있는 부분을 전개하면 다음과 같이 나타낼 수 있다.

$$\begin{aligned} &\frac{d}{dt} ({}^N \mathbf{X}_{N-1} \dots {}^1 \mathbf{X}_0) \\ &= {}^N \dot{\mathbf{X}}_{N-1} {}^N \mathbf{X}_0 + \sum_{i=2}^{N-1} ({}^N \mathbf{X}_i {}^i \dot{\mathbf{X}}_{i-1} {}^{i-1} \mathbf{X}_0) + {}^N \mathbf{X}_1 {}^1 \dot{\mathbf{X}}_0 \end{aligned} \quad (14)$$

자코비언의 i 번째 열은 엔드 이펙터 속도에서 i 번째 관절의 속도가 미치는 영향을 의미하므로 (14)식에서 $\dot{q}_i = 1$, $\dot{q}_j = 0$ ($j \neq i$)로 두고 ${}^E \dot{\mathbf{X}}_0$ 을 계산한 다음, 이로부터 ${}^E \dot{\mathbf{R}}_0$, \mathbf{v}_E 의 성분을 추출하면 자코비언의 i 번째 열을 구성할 수 있다. 여기서 ${}^i \dot{\mathbf{X}}_{i-1} = -[\mathbf{S}_i \times] {}^i \mathbf{X}_{i-1} \dot{q}_i$ 인 성질이 성립[12]하므로 이를 이용하여 (14)을 다시 쓰면,

$$\begin{aligned} &\frac{d}{dt} ({}^N \mathbf{X}_{N-1} \dots {}^1 \mathbf{X}_0) \\ &= -[\mathbf{S}_N \times] {}^N \mathbf{X}_0 \dot{q}_N - \sum_{i=1}^{N-1} ({}^N \mathbf{X}_i [\mathbf{S}_i \times] {}^i \mathbf{X}_0 \dot{q}_i) \end{aligned} \quad (15)$$

따라서 자코비언의 i 번째 열 J_i 는 (15)의 식에 $\dot{q}_i = 1$, $\dot{q}_j = 0$ ($j \neq i$)을 대입한 결과에서 ${}^E \dot{\mathbf{R}}_0$, \mathbf{v}_E 의 성분을 추출하여 구할 수 있다. 아래의 알고리즘2는 이러한 과정을 정리한 것이다.

Algorithm 2: End-Effector Jacobian Computation

Step1: Initialization

$$\mathbf{U}_0 = \mathbf{1} \quad (16)$$

Step 2: Compute $\mathbf{U}_i, \mathbf{U}_E$ (1st loop)

For $i=1$ to N

$$\mathbf{U}_i = {}^i \mathbf{X}_{i-1} \mathbf{U}_{i-1} \quad (17)$$

End For

$$\mathbf{U}_E = {}^E \mathbf{X}_N \mathbf{U}_N \quad (18)$$

Step 4: Extract ${}^E \mathbf{R}_0$, $[\mathbf{r}_E^\circ]$ from \mathbf{U}_E

Step 5: Compute $J_i(2^{\text{nd}} \text{ loop})$

For $i=N$ to 1

$$\mathbf{C}_i = \begin{cases} {}^E \mathbf{X}_N & \text{if } i = N \\ \mathbf{C}_{i+1} {}^{i+1} \mathbf{X}_i & \text{otherwise} \end{cases} \quad (19)$$

$$\begin{aligned} \mathbf{Y}_i &= -\mathbf{C}_i [\mathbf{S}_i \times] \mathbf{U}_i \\ &= \begin{bmatrix} {}^E \dot{\mathbf{R}}_{0,i} & \mathbf{0} \\ -{}^E \dot{\mathbf{R}}_{0,i} [\mathbf{r}_E^\circ] - {}^E \mathbf{R}_0 [\mathbf{v}_{E,i}^\circ] & {}^E \dot{\mathbf{R}}_{0,i} \end{bmatrix} \end{aligned} \quad (20)$$

Extract ${}^E \dot{\mathbf{R}}_{0,i}$, $\mathbf{v}_{E,i}^\circ$ from \mathbf{Y}_i

$$[\omega_i^\circ] = {}^E \dot{\mathbf{R}}_{0,i} {}^E \mathbf{R}_0^T \quad (21)$$

$$[\mathbf{v}_i^\circ] = \mathbf{v}_{E,i}^\circ \quad (22)$$

$$\mathbf{J}_i = [\omega_i^T \quad \mathbf{v}_i^T]^T \quad (23)$$

End For

Step 6: Set end-effector Jacobian $\mathbf{J} = [\mathbf{J}_1 \quad \dots \quad \mathbf{J}_N]$

여기서 (21)은 좌표계의 회전행렬과 회전속도 사이에 일반적으로 성립하는 성질로 일반적인 강체의 동역학을 다룬 문헌에서 확인할 수 있다[14].

III. 자유 부유 매니플레이터의 기구학적 최적제어

본 절에서는 자유 부유 매니플레이터의 여유도 해결 문제를 최적 제어 이론의 틀에 맞추어 서술하고, 이로부터 경계치 문제를 유도하는 과정을 서술하였다. 서술의 편의상 위성 몸체의 위치와 방향은 \mathbf{x}_1 , 매니플레이터 관절의 회전각은 \mathbf{x}_2 로 상태 벡터를 정의하였다. 그러면 운동량 보존 관계식은 아래와 같이 쓸 수 있다.

$$\mathbf{E}_1 \dot{\mathbf{x}}_1 + \mathbf{E}_2 \dot{\mathbf{x}}_2 = \mathbf{0} \quad (24)$$

여기서 $\mathbf{E}_1 = [\mathbf{D}_1 \quad \dots \quad \mathbf{D}_n]$, $\mathbf{E}_2 = [\mathbf{D}_{n+1} \quad \dots \quad \mathbf{D}_N]$ 이고, n 은 위성 몸체의 운동을 조정하는 마지막 관절의 번호로 그림 1과 같은 구성의 매니플레이터에서는 $n=6$ 이 된다. 한편, 엔드 이펙터의 속도 제약 조건은 아래와 같이 기술할 수 있다.

$$\mathbf{v}_E = \mathbf{F}_1 \dot{\mathbf{x}}_1 + \mathbf{F}_2 \dot{\mathbf{x}}_2 = \mathbf{G} \dot{\mathbf{x}}_2 = (\mathbf{F}_2 - \mathbf{F}_1 \mathbf{E}_1^{-1} \mathbf{E}_2) \dot{\mathbf{x}}_2 \quad (25)$$

여기서 $\mathbf{F}_1 = [\mathbf{J}_1 \quad \dots \quad \mathbf{J}_n]$, $\mathbf{F}_2 = [\mathbf{J}_{n+1} \quad \dots \quad \mathbf{J}_N]$ 이고 \mathbf{v}_E 는 엔드 이펙터의 속도이다. (25)의 $\mathbf{G} = \mathbf{F}_2 - \mathbf{F}_1 \mathbf{E}_1^{-1} \mathbf{E}_2$ 은 일반화된 자코비언(generalized Jacobian)로 몸체가 고정된 매니플레이터에서의 자코비언과 동일한 역할을 수행하게 된다[8]. 이제 엔드 이펙터의 속도 \mathbf{v}_E 가 주어졌을 때, 이를 추종하는 자유 부유 매니플레이터의 상태 방정식은 아래와 같이 정리된다.

$$\begin{aligned} \dot{\mathbf{x}}_1 &= -\mathbf{E}_1^{-1} \mathbf{E}_2 (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}) \\ \dot{\mathbf{x}}_2 &= \mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi} \end{aligned} \quad (26)$$

여기서 $\mathbf{\Gamma} = \mathbf{1} - \mathbf{G}^\# \mathbf{G}$, $\mathbf{\Phi} = \mathbf{G}^\# \mathbf{v}_E = \mathbf{G}^\# \mathbf{v}_E$ 이다. 여기서 $\mathbf{G}^\# = \mathbf{G}^T (\mathbf{G} \mathbf{G}^T)^{-1}$ 이며, \mathbf{u} 는 \mathbf{G} 의 영공간(null space)으로 투영되는

임의의 벡터로, (26)의 시스템에서는 입력의 역할을 하게 된다. 이제 이상의 정의를 토대로 아래와 같이 최적 제어 문제를 기술할 수 있다.

$$\begin{aligned} \text{Minimize } f &= \int_0^{t_f} \frac{1}{2} \dot{\mathbf{x}}_2^T \dot{\mathbf{x}}_2 dt \\ \text{subject to } \dot{\mathbf{x}}_1 &= -\mathbf{E}_1^{-1} \mathbf{E}_2 (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}), \quad \dot{\mathbf{x}}_2 = \mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi} \end{aligned} \quad (27)$$

(27)의 문제로부터 해밀토니언(Hamiltonian)은 아래 (28)과 같이 정의할 수 있다.

$$\begin{aligned} H &= \frac{1}{2} (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi})^T (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}) \\ &\quad - \lambda_1^T \{ \mathbf{E}_1^{-1} \mathbf{E}_2 (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}) \} + \lambda_2^T (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}) \end{aligned} \quad (28)$$

여기서 $\mathbf{\Gamma}^T = \mathbf{\Gamma}$, $\mathbf{\Gamma}^2 = \mathbf{\Gamma}$, $\mathbf{\Phi}^T \mathbf{\Gamma} = \mathbf{0}$ 인 성질을 이용하면 위 식은 아래와 같이 정리된다.

$$H = \frac{1}{2} \mathbf{u}^T \mathbf{\Gamma} \mathbf{u} + \frac{1}{2} \mathbf{\Phi}^T \mathbf{\Phi} + (\lambda_2 - \mathbf{E}_2^T \mathbf{E}_1^{-T} \lambda_1)^T (\mathbf{\Gamma} \mathbf{u} + \mathbf{\Phi}) \quad (29)$$

최소 원리(Minimum Principle)로부터 (29)를 \mathbf{u} 로 미분한 값은 0이 되어야 한다[15].

$$\frac{\partial H}{\partial \mathbf{u}} = \mathbf{\Gamma} (\mathbf{u} + \lambda_2 - \mathbf{E}_2^T \mathbf{E}_1^{-T} \lambda_1) = \mathbf{0} \quad (30)$$

(30)로부터 최적 제어 입력 \mathbf{u}_* 은 아래와 같이 정리된다.

$$\mathbf{u}_* = \mathbf{E}_2^T \mathbf{E}_1^{-T} \lambda_1 - \lambda_2 \quad (31)$$

(31)를 (29)에 $\mathbf{u} = \mathbf{u}_*$ 으로 대입하여 정리하면 다음과 같은 결과를 얻는다.

$$H_* = -\frac{1}{2} \mathbf{u}_*^T \mathbf{\Gamma} \mathbf{u}_* + \frac{1}{2} \mathbf{\Phi}^T \mathbf{\Phi} - \mathbf{u}_*^T \mathbf{\Phi} \quad (32)$$

이제 $\dot{\lambda}_1 = -\partial H_* / \partial \mathbf{x}_1$, $\dot{\lambda}_2 = -\partial H_* / \partial \mathbf{x}_2$ 임을 이용하여 보조 상태 방정식을 구해보면 아래와 같이 정리된다.

$$\dot{\lambda}_{i,j} = (\mathbf{\Gamma} \mathbf{u}_* + \mathbf{\Phi})^T \frac{\partial \mathbf{u}_*}{\partial x_{i,j}} + \frac{1}{2} \mathbf{u}_*^T \frac{\partial \mathbf{\Gamma}}{\partial x_{i,j}} \mathbf{u}_* + (\mathbf{u}_* - \mathbf{\Phi})^T \frac{\partial \mathbf{\Phi}}{\partial x_{i,j}} \quad (33)$$

여기서 $x_{i,j}$ 와 $\lambda_{i,j}$ 는 각각 벡터 \mathbf{x}_i 와 λ_i 의 j 번째 원소를 의미한다. 이상의 결과를 종합하면 (27)의 최적 제어 문제는 아래와 같은 일련의 미분 방정식에 의해 기술되는 경계치 문제가 된다.

$$\begin{aligned} \dot{\mathbf{x}}_1 &= -\mathbf{E}_1^{-1} \mathbf{E}_2 (\mathbf{\Gamma} \mathbf{u}_* + \mathbf{\Phi}) \\ \dot{\mathbf{x}}_2 &= \mathbf{\Gamma} \mathbf{u}_* + \mathbf{\Phi} \\ \dot{\lambda}_{i,j} &= (\mathbf{\Gamma} \mathbf{u}_* + \mathbf{\Phi})^T \frac{\partial \mathbf{u}_*}{\partial x_{i,j}} + \frac{1}{2} \mathbf{u}_*^T \frac{\partial \mathbf{\Gamma}}{\partial x_{i,j}} \mathbf{u}_* + (\mathbf{u}_* - \mathbf{\Phi})^T \frac{\partial \mathbf{\Phi}}{\partial x_{i,j}} \\ \mathbf{u}_* &= \mathbf{E}_2^T \mathbf{E}_1^{-T} \lambda_1 - \lambda_2 \end{aligned} \quad (34)$$

이 때, 경계 조건은. $t=0$ 에서는 초기 조건이 주어지고, $t=t_f$ 에서는 주어진 엔드 이펙터 제약 조건만을 만족하기만 하면 되므로 아래와 같이 주어지게 된다[15].

$$\begin{aligned} \mathbf{x}_1(0) &= \mathbf{x}_{1,0} \\ \mathbf{x}_2(0) &= \mathbf{x}_{2,0} \\ (\mathbf{1} - \mathbf{J}^\# \mathbf{J}) \left[\boldsymbol{\lambda}_1(t_f)^T \quad \boldsymbol{\lambda}_2(t_f)^T \right]^T &= \mathbf{0} \end{aligned} \quad (35)$$

IV. 경계치 문제의 미분방정식을 구하기 위한 계산과정

식 (33)의 미분방정식을 적분하여 $\boldsymbol{\lambda}_i(t)$ 의 궤적을 구하기 위해서는 $\frac{\partial \mathbf{u}_*}{\partial x_{i,j}}, \frac{\partial \Phi}{\partial x_{i,j}}, \frac{\partial \Gamma}{\partial x_{i,j}}, \frac{\partial \mathbf{G}}{\partial x_{i,j}}$ ($i=1,2$)의 값을 정확히 구할 필요가 있다. 이 식은 아래와 같이 보다 기본적인 항들의 합과 곱으로 전개하여 나타낼 수 있다.

$$\frac{\partial \mathbf{u}_*}{\partial x_{i,j}} = \left(-\mathbf{E}_1^{-1} \frac{\partial \mathbf{E}_1}{\partial x_{i,j}} \mathbf{E}_1^{-1} \mathbf{E}_2 + \mathbf{E}_1^{-1} \frac{\partial \mathbf{E}_2}{\partial x_{i,j}} \right)^T \boldsymbol{\lambda}_1 \quad (36)$$

$$\begin{aligned} \frac{\partial \Phi}{\partial x_{i,j}} &= \left[\frac{\partial \mathbf{G}^T}{\partial x_{i,j}} (\mathbf{G} \mathbf{G}^T)^{-1} - \right. \\ &\quad \left. \mathbf{G}^\# \left(\frac{\partial \mathbf{G}}{\partial x_{i,j}} \mathbf{G}^T + \mathbf{G} \frac{\partial \mathbf{G}^T}{\partial x_{i,j}} \right) (\mathbf{G} \mathbf{G}^T)^{-1} \right] \mathbf{v}_E \end{aligned} \quad (37)$$

$$\begin{aligned} \frac{\partial \Gamma}{\partial x_{i,j}} &= -\frac{\partial \mathbf{G}^T}{\partial x_{i,j}} (\mathbf{G}^\#)^T \\ &\quad + \mathbf{G}^\# \left\{ \left(\frac{\partial \mathbf{G}}{\partial x_{i,j}} \mathbf{G}^T + \mathbf{G} \frac{\partial \mathbf{G}^T}{\partial x_{i,j}} \right) (\mathbf{G}^\#)^T - \frac{\partial \mathbf{G}}{\partial x_{i,j}} \right\} \end{aligned} \quad (38)$$

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial x_{i,j}} &= \frac{\partial \mathbf{F}_2}{\partial x_{i,j}} - \frac{\partial \mathbf{F}_1}{\partial x_{i,j}} \mathbf{E}_1^{-1} \mathbf{E}_2 \\ &\quad + \mathbf{F}_2 \mathbf{E}_1^{-1} \frac{\partial \mathbf{E}_1}{\partial x_{i,j}} \mathbf{E}_1^{-1} \mathbf{E}_2 - \mathbf{F}_2 \mathbf{E}_1^{-1} \frac{\partial \mathbf{E}_2}{\partial x_{i,j}} \end{aligned} \quad (39)$$

여기서 (36)~(39)의 수식을 전개하는 데에는 $\frac{\partial(\mathbf{X}^{-1})}{\partial \mathbf{y}} = -\mathbf{X}^{-1} \frac{\partial \mathbf{X}}{\partial \mathbf{y}} \mathbf{X}^{-1}$ 인 성질이 사용되었다. 이제 (36)~(39)의 값을 계산하려면 $\frac{\partial \mathbf{E}_1}{\partial x_{i,j}}, \frac{\partial \mathbf{E}_2}{\partial x_{i,j}}, \frac{\partial \mathbf{F}_1}{\partial x_{i,j}}, \frac{\partial \mathbf{F}_2}{\partial x_{i,j}}$ 의 값을 구할 필요가 있고, 이를 위해서는 $\frac{\partial \mathbf{D}_i}{\partial q_k}, \frac{\partial \mathbf{J}}{\partial q_k}$ 의 값을 알아야 한다. 이 값은 아래의 알고리즘 3, 4를 이용해서 구할 수 있다. 알고리즘 3과 4는 알고리즘 1, 2의 각 행을 q_k 로 편미분하여 유도 되었으며, 편이상 벡터 또는 행렬 \mathbf{X} 에 대한 편미분 $\frac{\partial \mathbf{X}}{\partial q_k}, \frac{\partial \dot{\mathbf{X}}}{\partial q_k}$ 을 각각 $\tilde{\mathbf{X}}, \dot{\tilde{\mathbf{X}}}$ 으로 축약하여 표기하였다. 여기서 알고리즘 3은 알고리즘 1에서 계산된 결과를 사용하며, 알고리즘 4는 알고리즘 2의 계산 결과를 이용하게 된다.

Algorithm 3: Computation of $\frac{\partial \mathbf{D}_i}{\partial q_k}$

Step1: Initialization

$$\tilde{\mathbf{A}}_0 = \mathbf{0} \quad (40)$$

Step 2: Computation of $\tilde{\mathbf{A}}_i$ (1^{st} loop)

For $i=1$ to N

$${}^i \tilde{\mathbf{X}}_{i-1} = \begin{cases} -[\mathbf{S}_i \times] {}^i \mathbf{X}_{i-1} & \text{if } i=k \\ 0 & \text{otherwise} \end{cases} \quad (41)$$

$$\tilde{\mathbf{A}}_i = \tilde{\mathbf{A}}_{i-1} {}^i \mathbf{X}_{i-1}^T + \mathbf{A}_{i-1} {}^i \tilde{\mathbf{X}}_{i-1}^T \quad (42)$$

End For

Step 3: Computation of $\frac{\partial \mathbf{D}_i}{\partial q_k}$

For $i=N$ to 1

$$\tilde{\mathbf{B}}_i = \begin{cases} 0 & \text{if } i=N \\ {}^{i+1} \tilde{\mathbf{X}}_i^T \mathbf{B}_{i+1} {}^{i+1} \mathbf{X}_i + \\ {}^{i+1} \mathbf{X}_i^T \tilde{\mathbf{B}}_{i+1} {}^{i+1} \mathbf{X}_i + & \text{otherwise} \\ {}^{i+1} \mathbf{X}_i^T \mathbf{B}_{i+1} {}^{i+1} \tilde{\mathbf{X}}_i \end{cases} \quad (43)$$

$$\frac{\partial \mathbf{D}_i}{\partial q_k} = (\tilde{\mathbf{A}}_i \mathbf{B}_i + \mathbf{A}_i \tilde{\mathbf{B}}_i) \mathbf{S}_i \quad (44)$$

End For

Algorithm 4: Computation of $\frac{\partial \mathbf{J}}{\partial q_k}$

Step 1: Initialization

$$\tilde{\mathbf{U}}_0 = \mathbf{0} \quad (45)$$

Step 2: Computation of $\tilde{\mathbf{U}}_i, \tilde{\mathbf{U}}_E$ (1^{st} loop)

For $i=1$ to N

$${}^i \tilde{\mathbf{X}}_{i-1} = \begin{cases} -[\mathbf{S}_i \times] {}^i \mathbf{X}_{i-1} & \text{if } i=k \\ 0 & \text{otherwise} \end{cases} \quad (46)$$

$$\tilde{\mathbf{U}}_i = {}^i \tilde{\mathbf{X}}_{i-1} \mathbf{U}_{i-1} + {}^i \mathbf{X}_{i-1} \tilde{\mathbf{U}}_{i-1} \quad (47)$$

End For

$$\tilde{\mathbf{U}}_E = {}^E \mathbf{X}_N \tilde{\mathbf{U}}_N \quad (48)$$

Extract ${}^E \tilde{\mathbf{R}}_0$ and $[\tilde{\mathbf{r}}_E^\circ]$ from $\tilde{\mathbf{U}}_E$

Step 3: Computation of Derivative of End-Effector Jacobian

For $i=N$ to 1

$$\tilde{\mathbf{C}}_i = \begin{cases} 0 & i=N \\ \tilde{\mathbf{C}}_{i+1} {}^{i+1} \mathbf{X}_i + \mathbf{C}_{i+1} {}^{i+1} \tilde{\mathbf{X}}_i & \text{otherwise} \end{cases} \quad (49)$$

$$\begin{aligned} \tilde{\mathbf{Y}}_i &= -\tilde{\mathbf{C}}_i [\mathbf{S}_i \times] \mathbf{U}_i - \mathbf{C}_i [\mathbf{S}_i \times] \tilde{\mathbf{U}}_i \\ &= \begin{bmatrix} {}^E \hat{\mathbf{R}}_{0,i} & \mathbf{0} \\ \boldsymbol{\zeta}_i & {}^E \hat{\mathbf{R}}_{0,i} \end{bmatrix} \end{aligned} \quad (50)$$

Extract ${}^E \hat{\mathbf{R}}_{0,i}, [\tilde{\mathbf{v}}_{E,i}^\circ]$ from $\tilde{\mathbf{Y}}_i$

$$[\tilde{\omega}_i^\circ] = {}^E \hat{\mathbf{R}}_{0,i} {}^E \mathbf{R}_0^T + {}^E \hat{\mathbf{R}}_{0,i} {}^E \tilde{\mathbf{R}}_0^T \quad (51)$$

$$[\tilde{\mathbf{v}}_i^\circ] = [\tilde{\mathbf{v}}_{E,i}^\circ] \quad (52)$$

$$\tilde{\mathbf{J}}_i = [\tilde{\omega}_i^T \quad \tilde{\mathbf{v}}_i^T]^T \quad (53)$$

End For

Step 4: Set derivative of end-effector Jacobian $\frac{\partial \mathbf{J}}{\partial q_k} = [\tilde{J}_1 \dots \tilde{J}_N]$

(50)에서 ζ_i 는 아래와 같다.

$$\zeta_i = -{}^E \hat{\mathbf{R}}_{0,i} [\mathbf{r}_E^{\otimes}] - {}^E \hat{\mathbf{R}}_{0,i} [\dot{\mathbf{r}}_E^{\otimes}] - {}^E \hat{\mathbf{R}}_0 [\mathbf{v}_{E,i}^{\otimes}] - {}^E \mathbf{R}_0 [\dot{\mathbf{v}}_{E,i}^{\otimes}] \quad (54)$$

이상을 종합하여 다음과 같이 기구학적 최적 제어를 사용한 자유 부유 매니퓰레이터의 여유도 해결을 위한 전체적인 알고리즘을 구성할 수 있다. 먼저 알고리즘1을 수행하여 행렬 \mathbf{D} 을 계산하고, 알고리즘2를 수행하여 자코비언 행렬 \mathbf{J} 를 계산한다. 다음으로 알고리즘1의 계산 결과들을 이용하여 $k=1$ 에서 $k=N$ 까지 알고리즘3을 반복 수행한다. 이 결과로 $\partial \mathbf{D}_i / \partial q_k$ ($k=1, \dots, N$)을 얻게 된다. 마찬가지로 알고리즘2의 계산 결과들을 이용하여 $k=1$ 에서 $k=N$ 까지 알고리즘4를 반복한다. 이 결과로 $\partial \mathbf{J} / \partial q_k$ ($k=1, \dots, N$)을 얻게 된다. 이제 $\partial \mathbf{D}_i / \partial q_k$ 와 $\partial \mathbf{J} / \partial q_k$ 의 값을 알고 있으므로 이를 이용하여

$\frac{\partial \mathbf{E}_1}{\partial x_{i,j}} = \begin{bmatrix} \frac{\partial \mathbf{D}_1}{\partial x_{i,j}} & \dots & \frac{\partial \mathbf{D}_n}{\partial x_{i,j}} \end{bmatrix}$, $\frac{\partial \mathbf{E}_2}{\partial x_{i,j}} = \begin{bmatrix} \frac{\partial \mathbf{D}_{n+1}}{\partial x_{i,j}} & \dots & \frac{\partial \mathbf{D}_N}{\partial x_{i,j}} \end{bmatrix}$, $\frac{\partial \mathbf{F}_1}{\partial x_{i,j}} = \begin{bmatrix} \frac{\partial J_1}{\partial x_{i,j}} & \dots & \frac{\partial J_n}{\partial x_{i,j}} \end{bmatrix}$, $\frac{\partial \mathbf{F}_2}{\partial x_{i,j}} = \begin{bmatrix} \frac{\partial J_{n+1}}{\partial x_{i,j}} & \dots & \frac{\partial J_N}{\partial x_{i,j}} \end{bmatrix}$ 을 구성할 수 있다. 여기서 $x_{i,j}$ 는 벡터 \mathbf{x}_i 의 j 번째 원소이며, $\mathbf{x}_1 = [q_1 \dots q_n]^T$, $\mathbf{x}_2 = [q_{n+1} \dots q_N]^T$ 이다.

$\partial \mathbf{E}_1 / \partial x_{i,j}$, $\partial \mathbf{E}_2 / \partial x_{i,j}$, $\partial \mathbf{F}_1 / \partial x_{i,j}$, $\partial \mathbf{F}_2 / \partial x_{i,j}$ 를 (36)~(39)에 대입하면 $\partial \mathbf{u}_i / \partial x_{i,j}$, $\partial \mathbf{\Phi} / \partial x_{i,j}$, $\partial \mathbf{\Gamma} / \partial x_{i,j}$, $\partial \mathbf{G} / \partial x_{i,j}$ 의 값을 계산할 수 있고, 이 결과를 (34)에 대입하면 경계치 문제를 구성하는 모든 미분방정식을 명확히 얻게 된다. 이제(34), (35)로 정의되는 경계치 문제를 푸는 궤적을 구하면 자유 부유 매니퓰레이터의 여유도 해결 문제에 대한 해를 얻을 수 있다. 이러한 경계치 문제를 풀기 위한 수단으로는 기존에 연구되어 있는 다양한 방법들을 동원할 수 있는데, 본 논문에서는 다음 절의 시뮬레이션 결과를 얻기 위해 simple shooting method [16]를 사용하였다.

V. 시뮬레이션

이 장에는 앞에서 설명한 전체적인 알고리즘을 평면 3관절 자유 부유 매니퓰레이터 모델에 적용한 시뮬레이션 결과를 수록하였다. 시뮬레이션에 사용된 모델에서 몸체의 질량은 100 kg, 회전 관성은 66.67 kgm², 매니퓰레이터 링크의 질량은 2 kg, 회전 관성은 0.7083 kgm²으로 설정하였다. 시뮬레이션의 초기 조건은 그림 2와 같이 몸체의 위치와 방향각은 각각 (0, 0)과 0°로, 매니퓰레이터의 관절 각은 (45°, 60°, -60°)으로 설정하였으며 엔드 이펙터는 (-1, -1) m/sec의 일정한 속도로 이동하도록 하였으며, 엔드 이펙터의 방향에 대해서는 제약 조건을 주지 않았다.

비교를 위해 그림 3~7에는 단순한 의사 역행렬 피드백을 사용한 결과를, 그림 8~12에는 본 논문에서 제시한 방식을 사용하여 얻은 결과를 수록하였다. 그림 2의 초기 조건에서

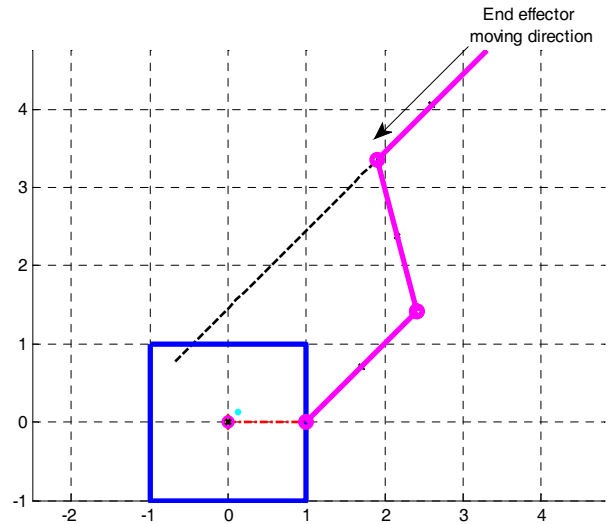


그림 2. 초기 자세.

Fig. 2. Initial configuration.

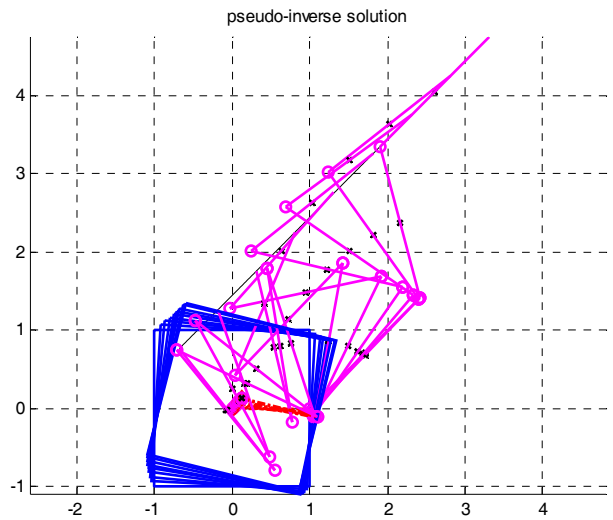


그림 3. 단순한 의사 역행렬 피드백 적용시 궤적.

Fig. 3. Motion trajectory with simple pseudo-inverse feedback.

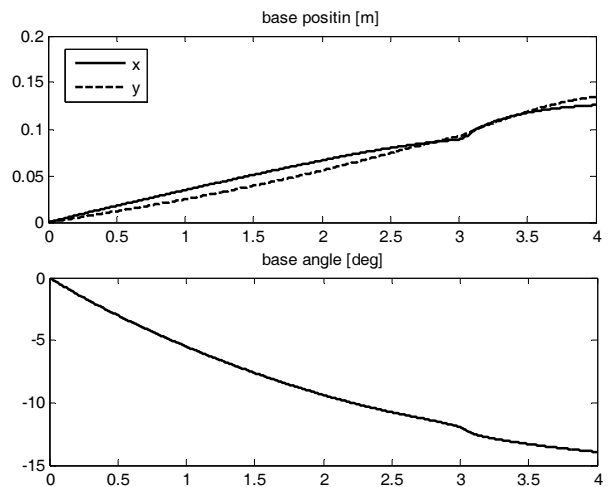


그림 4. 몸체의 위치, 방향(의사역행렬 피드백).

Fig. 4. Base position and angle with pseudo-inverse feedback.

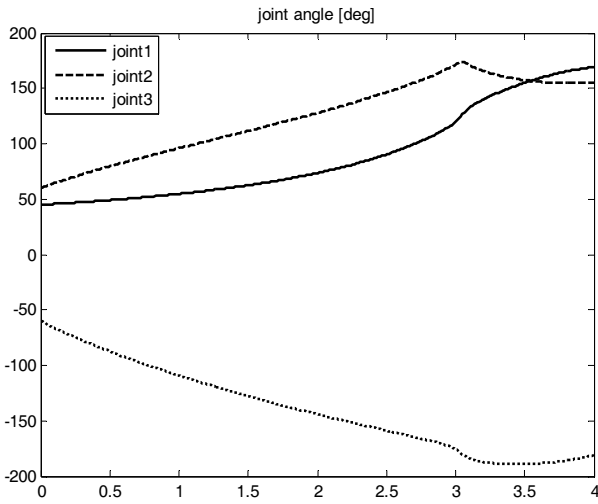


그림 5. 관절의 회전각(의사역행렬 피드백).
Fig. 5. Trajectory of joint angle with pseudo-inverse feedback.

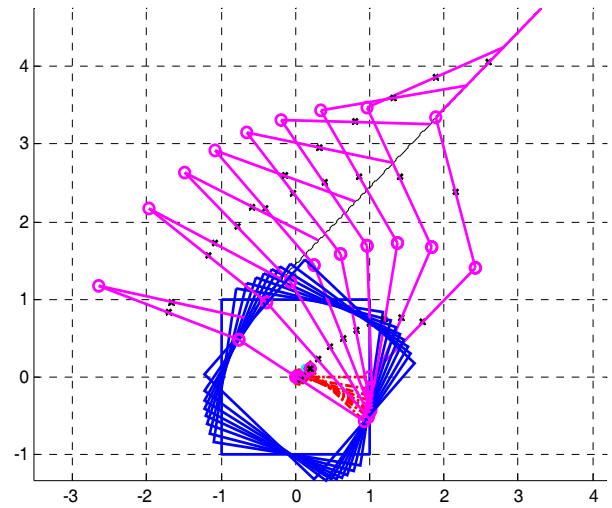


그림 8. 최적 제어 문제를 풀어 얻은 궤적.
Fig. 8. Motion trajectory with optimal control.

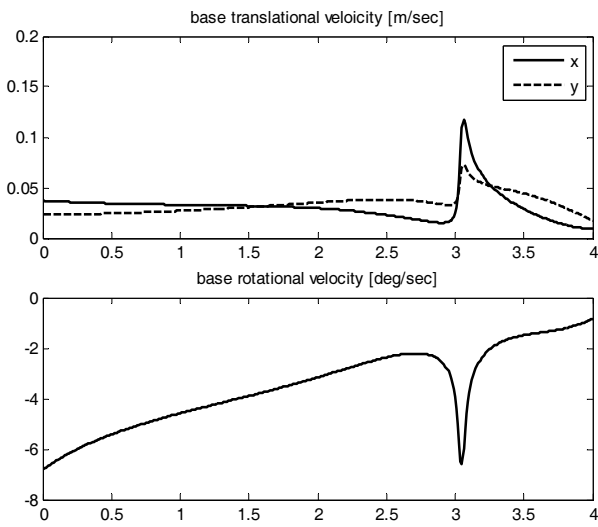


그림 6. 물체의 속도(의사역행렬 피드백).
Fig. 6. Trajectory of base velocity with pseudo-inverse feedback.

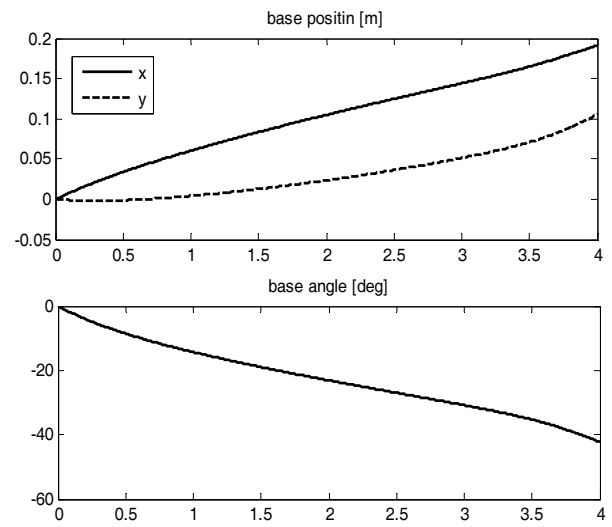


그림 9. 물체의 위치와 방향(최적제어).
Fig. 9. Trajectory of base position and angle with optimal control.

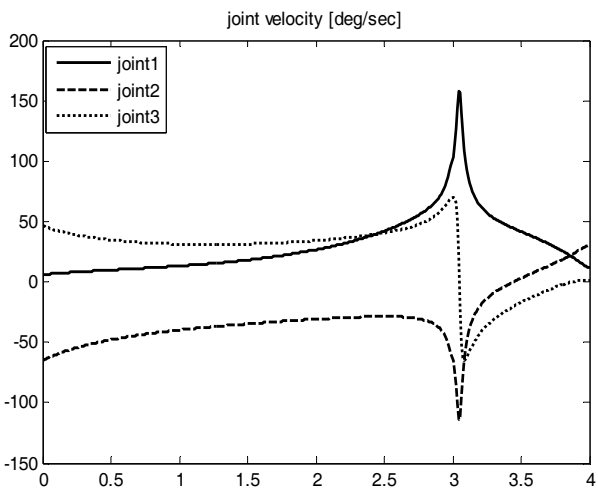


그림 7. 관절의 회전속도(의사역행렬 피드백).
Fig. 7. Trajectory of joint velocity with pseudo-inverse feedback.

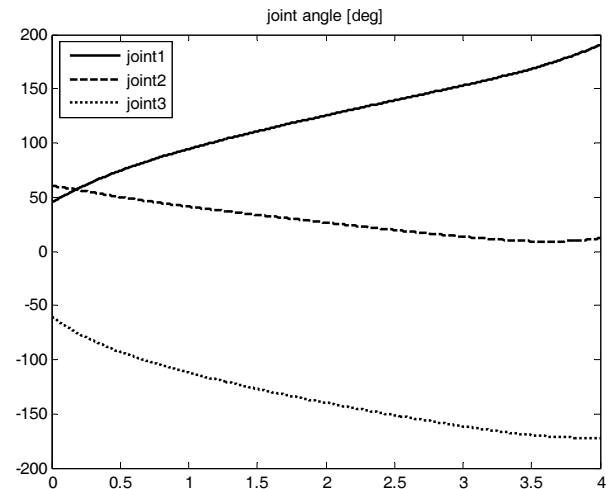


그림 10. 관절의 회전각(최적제어).
Fig. 10. Trajectory of joint angle with optimal control.

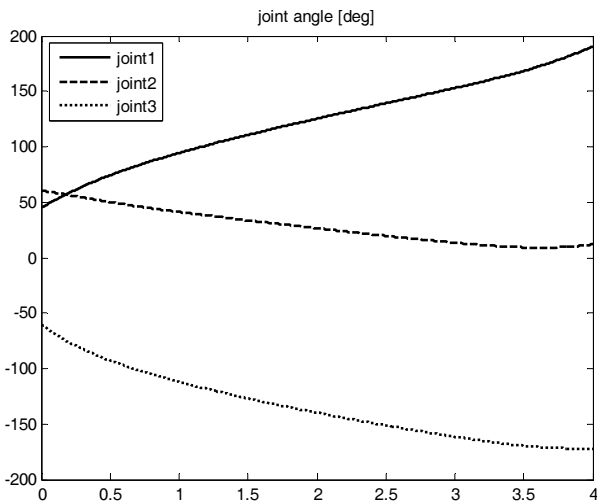


그림 11. 몸체의 속도(최적제어).
Fig. 11. Trajectory of base velocity with optimal control.

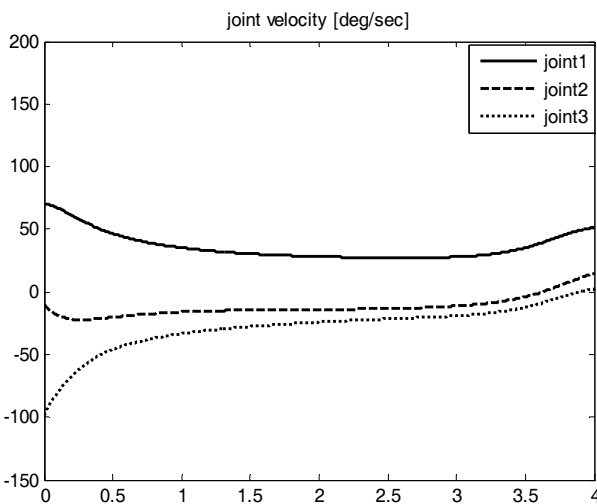


그림 12. 관절의 회전속도(최적제어).
Fig. 12. Trajectory of joint velocity with optimal control.

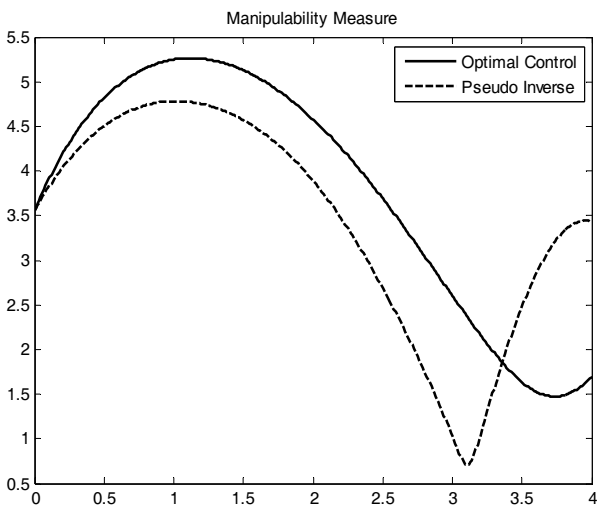


그림 13. 조작도 지수.
Fig. 13. Manipulability measure.

표 1. 관절 속도의 피크 값 (deg/sec).

Table 1. Peak joint velocity (deg/sec).

Joint	Pseudo-Inverse	Suggested Algorithm
1	157.8240	70.4688
2	69.7189	-22.0748
3	-113.5345	-98.1823

표 2. 관절 속도의 피크 순간의 관절 가속도 (deg/sec²).

Table 2. Joint acceleration at the instant of peak joint velocity (deg/sec²).

Joint	Pseudo-Inverse	Suggested Algorithm
1	-154757.4	-811.6214
2	-14153.14	136.2558
3	148085.1	-296.0277

단순한 의사 역행렬을 사용하는 피드백 제어를 적용하면 그림 3과 같이 매니플레이터 자세가 기구학적 특이점으로 유도되어 그 결과, 관절의 회전 속도와 몸체의 속도가 그림 6, 7과 같이 급격히 튀게 된다. 이러한 속도 궤적은 관절의 급격한 가속을 요구하므로 구동기에 무리를 주게 된다. 반면, 최적 제어 문제를 풀어 얻은 그림 8~12의 결과에서는 기구학적 특이점을 회피하게 되어 부드러운 속도 궤적이 얻어지는 것을 볼 수 있다.

그림 13은 두 궤적에 대해 $\sqrt{\det(\mathbf{GG}^T)}$ 으로 정의되는 조작도 지수를 표시한 것으로 이 값이 0에 근접할수록 기구학적 특이점에 가까워지는 것임을 의미한다. 의사 역행렬을 사용하는 경우에는 3초 부근에서 매니플레이터의 자세가 기구학적 특이점 부근을 지나게 되면서 조작도 지수 값이 급격히 떨어졌다가 다시 증가하는 현상이 나타나지만 본 논문에서 제안한 방식으로 얻은 결과에서는 이러한 급격한 변화가 나타나지 않는 것을 볼 수 있다.

표 1, 2는 두 경우에서 구동기에 가해지는 부담을 비교하기 위해 관절 속도와 가속도의 피크 값을 정리한 것이다. 이를 보면 단순 의사 역행렬을 사용하는 국소적 방법을 적용한 경우에는 기구학적 특이점을 지나게 되면서 관절에 매우 큰 부담이 가해지게 되는데 비해 본 논문에서 제시한 방법을 사용하여 얻은 궤적에서는 구동기에 가해지는 부담이 훨씬 적음을 알 수 있다.

VI. 결론

본 논문에서는 자유 부유 매니플레이터의 여유도 해결 문제를 기구학적 최적 제어 문제로 변환하여 푸는 체계적인 알고리즘을 설명하였다. 통상적인 최적 제어 이론에 따라 여유도 해결 문제를 기술하면 일련의 미분방정식으로 구성된 경계치 문제를 얻게 되는데, 관절의 개수가 많은 매니플레이터에서는 이 미분방정식의 각 항을 명확한 표현식으로 나타내기가 어려워진다. 관절의 개수가 적고 몸체가 고정되어 있는 매니플레이터 모델의 경우에는 수식 처리를 도와주는 Maple이나 Mathematica와 같은 소프트웨어를 사용하는 것도 유용한 해결책이 될 수 있다. 이러한 소프트웨어는 경계치 문제를 구성하는 미분방정식의 표현식을 symbolic math processing을 통해 직접 구하게 된다. 그러나 저자의 경험으로는 이러

한 방법은 모델의 복잡도가 증가할수록 소프트웨어의 symbolic math processing에 소요되는 시간 또한 급격하게 증가할뿐더러, 그 결과로 얻어지는 식이 너무 길어져서 시뮬레이션 코드에 적용하기가 곤란해지는 문제가 있었다. 더욱이 자유 부유 매니플레이터의 경우에는 비교적 단순한 모델이라 하더라도 각운동량 보존 제약 조건이 더해지기 때문에 symbolic math processing이 비약적으로 복잡해지게 된다. 이 논문에서는 이러한 어려움을 해결하기 위해 순차적인 계산을 통해 경계치 문제를 구성하는 미분방정식을 정확히 구하는 알고리즘을 제시하였다. 본 논문에서 제시한 방식은 수학적 표현식을 직접 유도하는 것이 아니라 편미분으로 표현되는 항의 값만을 재귀적인 계산 알고리즘을 통해 구하게 되므로 많은 관절로 구성된 복잡한 매니플레이터 시스템에 대해서도 적용될 수 있는 것이 장점이다.

제시된 알고리즘의 유용성을 검증하기 위해 3-관절 평면 자유 부유 매니플레이터 모형에 대해 시뮬레이션을 수행하였으며 그 결과, 의사 역행렬을 사용한 단순한 피드백 적용 시 기구학적 특이점이 발생하는 조건에서도 제안한 방법을 적용하면 구동기에 부담을 적게 주는 부드러운 관절 궤적이 얻어지는 것을 확인할 수 있었다. 이와 같이 최적 제어 문제를 풀어서 동작을 계획하는 방식은 해를 생성하는데 많은 계산이 필요하므로 실시간 제어에는 부적합하지만 시뮬레이션에서 보인 것과 같이 간단한 피드백 제어만으로는 문제가 발생하는 상황에서 이를 보완하는 수단으로써 유용하게 쓰일 수 있을 것이다. 본 논문에서는 기구학 수준에서 로봇의 운동을 논하였으나 추후의 연구에서는 동역학을 포함하는 수준으로 문제를 확장하여 논의해보고자 한다.

참고문헌

- [1] B. Siciliano, "Kinematic control of redundant robot manipulators: a tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201-212, 1990.
- [2] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 308-316, Aug. 1987.
- [3] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanism," *Proc. of International Conference on Robotics and Automation*, pp. 1004-1009, 1985.
- [4] K. Kazerounian and A. Nedungadi, "Redundancy resolution of serial manipulators based on robot dynamics," *Mechanism and Machine Theory*, vol. 23, no. 4, pp. 295-303, 1988.
- [5] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 32-42, 1987.
- [6] S. Kim, K. Park, and J. Lee, "Redundancy resolution of robot manipulators using optimal kinematic control," *Proc. of*

International Conference on Robotics and Automation, pp. 683-688, 1994.

- [7] K. Suh and J. M. Hollerbach, "Local versus global torque optimization of redundant manipulators," *Proc. of International Conference on Robotics and Automation*, pp. 619-624, 1987.
- [8] Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized Jacobian matrix," *IEEE Transactions on Robotics and Automation*, pp. 303-314, 1989.
- [9] K. Yoshida, B. Kurazume, and Y. Umetani, "Torque optimization control in space robots with a redundant arm," *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pp. 1647-1652, 1991.
- [10] O. P. Agrawal and Y. Xu, "On the global optimum path planning for redundant space manipulators," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 24, no. 9, pp. 1306-1316, Sep. 1994.
- [11] P. Huang, Y. Xu, and B. Liang, "Global minimum-jerk trajectory planning of space manipulator," *International Journal of Control, Automation, and Systems*, vol. 4, no. 4, pp. 405-413, Aug. 2006.
- [12] R. Featherstone, *Rigid Body Dynamics Algorithm*, Springer, 2008.
- [13] F. C. Park, J. E. Bobrow, and S. R. Ploen, "A lie group formulation of robot dynamics," *The International Journal of Robotics Research*, pp. 279-287, 1995.
- [14] R. Murray, Z. Li, and S. Sastry, *A mathematical introduction to robotic manipulation*, CRC Press, 1994.
- [15] A. Bryson and Y. Ho, *Applied Optimal Control*, Taylor & Francis, 1975.
- [16] J. Z. Ben-Asher, *Optimal control theory with aerospace applications*, The American Institute of Aeronautics and Astronautics, 2010.



김 용 민

2004년 한국과학기술원 전자공학과 졸업. 2006년 한국과학기술원 전기 및 전자공학과 석사. 2006년~현재 동 대학원 전기 및 전자공학과 박사과정 재학중. 관심분야는 우주 매니플레이터의 동작 계획, 최적 제어.



김 병 국

1975년 서울대학교 전자공학과 졸업. 1975년 KAIST 전기 및 전자 공학과 석사. 1981년 동 대학원 박사. 1981년~1986년 우진계기(주) 연구실장. 1982년~1984년 University of Michigan 방문연구. 1986년~현재 KAIST 전기 및 전자공학과 교수. 관심분야는 실시간 시스템, 로보틱스, 임베디드 제어.