

여유자유도를 갖는 로봇 매니퓰레이터의 제어방법에 관한 조사 연구

서인홍\*  
한양대\*

전우남\*  
한국과학기술원

A survey on the control methods for the redundant robot manipulators

Il Hong Suh  
Hanyang Univ.

Zeungnam Bien  
KAIST

Abstract-This survey paper presents a review of control methods for redundant robot manipulators. Use of redundant degree of freedoms by local and global optimization techniques are described in terms of the Jacobian matrix equation for the redundant robot manipulators. Relevant problems for further use of redundant robot manipulators are then discussed.

1. 서론

본 논문에서는 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도는 로봇 매니퓰레이터의 관절 각도나 위치, 속도, 가속도 등을 나타내며, 이를 제어하기 위해서는 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산한다. 본 논문에서는 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법을 소개한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

2. 자코비안 행렬(Jacobian matrix) 방정식

2.1. 자코비안 행렬 방정식

여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

$$\dot{X}_e = f(\theta) \quad (1)$$

식(1)에서  $f(\theta)$ 는 연속시간 미분가능한 3차원 벡터이며,  $X_e$ 는 3차원 벡터이다. 따라서, 식(1)을 미분하면 식(2)를 얻는다.

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

여기서  $J(\theta)$ 는 3차원 공간의 3차원 행렬이며,  $\theta$ 는 3차원 벡터이다.  $J(\theta)$ 는 Jacobian matrix를 나타내며, 이를 이용하여 위치, 속도, 가속도 등을 계산한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

식(2)의 자코비안 행렬 방정식을 이용하여 위치, 속도, 가속도 등을 계산한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

$$Ay = b \quad (3)$$

식(3)에서  $y$ 는 3차원 벡터이며,  $b$ 는 3차원 상수 벡터이다.  $A$ 는  $n \times n$  행렬이다. 여기서 (1)은 위치, (2)는 속도, (3)는 가속도 방정식을 나타낸다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

이와 같이 식(2)에서  $\dot{\theta}$ 를 구하기 위해서는 식(3)을 풀어야 한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

2.2 자코비안 행렬 방정식의 일반화

여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다. 또한, 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법에 대한 조사를 소개한다. 여유자유도를 갖는 로봇 매니퓰레이터의 제어 방법은 크게 두 가지로 나눌 수 있다. 하나는 지역 최적화 방법이고, 다른 하나는 전역 최적화 방법이다. 지역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 전역 최적화 방법은 Jacobian matrix를 이용하여 위치, 속도, 가속도 등을 계산하는 방법이다. 본 논문에서는 지역 최적화 방법과 전역 최적화 방법을 소개한다.

$$\dot{\theta} = J^+(\theta)\dot{X}_e + (I_n - J^+(\theta)J)\dot{\theta} \quad (4)$$

여기서  $J^+(\theta) = (J^T J)^{-1} J^T$ 이며,  $I_n$ 는  $n \times n$  단위 행렬이다.  $n$ 차원의 관절 공간(joint space)에 있는 임의의 행렬이다. 식(4)에서  $(I_n - J^+(\theta)J)\dot{\theta}$ 는  $n$ 차원 벡터  $z$ 를  $N(J)$ 로 사상(mapping)하는 부분 좌표이다.

한편 [25, 26]에서는 식(4)의 일반화에 필요한  $J^+$ 의 계산이 복잡하고 시간이 많이 걸린다는 단점을 지적하고, 이를 보완하기 위해 다음과 같이 식(5)를 제안한다.

$$\dot{\theta} = \begin{bmatrix} J \\ \lambda \end{bmatrix}^{-1} \begin{bmatrix} \dot{X}_e \\ 0 \end{bmatrix} + \hat{N} \dot{y} \quad (5)$$

(5)에서  $\hat{x}$ 은  $(n-m) \times n$  행렬로서 각각의 열 벡터 (row vector)  $N(J)$ 의 수직인 벡터로, 이들은  $J$ 의 수직인 벡터들의 기본 벡터들이다. 식(5)의 어떤 특성은 명조함 하겠는가?  $J$ 의 가역성(invertible)은  $N(J)$ 의 수직인 벡터들이  $n-m$ 개의 선형 독립한 벡터를 이루는 것을 의미한다.  $N(J)$ 의 수직인 벡터들이  $n-m$ 개의 선형 독립한 벡터를 이루는 것은  $J$ 의 가역성(invertible)을 의미한다.  $J$ 의 가역성(invertible)은  $N(J)$ 의 수직인 벡터들이  $n-m$ 개의 선형 독립한 벡터를 이루는 것을 의미한다.  $J$ 의 가역성(invertible)은  $N(J)$ 의 수직인 벡터들이  $n-m$ 개의 선형 독립한 벡터를 이루는 것을 의미한다.

이 특이성(Singularity)이란 로부트 자유타도 가운데 외부의 특이성(Singularity)이 존재하는 경우를 말한다. 이 때 로부트 자유타도 가운데 외부의 특이성(Singularity)이 존재하는 경우를 말한다. 이 때 로부트 자유타도 가운데 외부의 특이성(Singularity)이 존재하는 경우를 말한다.

3. 부분 최적화(Local optimization)을 통한 어유자유타도의 이용

어유자유타도를 효율적으로 이용 하는 한 방법으로 및 함수로 주어지는 부가적 목표성능 지수를 정하고, 이 성능 지수에 대한 최적화 문제를 푸는 방법이 있다. 구체적으로 먼저 다음 문제를 생각해 보기도 하자.

[문제 1]

식(2)로 주어지는 자코비안 행렬 방정식을 만족하면서, 동시에 주어지는 목표 성능 지수  $H(\theta, \hat{e})$ 을 최소로 하는  $\hat{e}$ 를 구하라.

여기서  $H(\theta, \hat{e})$ 는 제어 문제의 성능 지수(Performance Index)에 비할 수 있다.  $\theta$ 는 시스템 파라미터를 나타내며,  $\hat{e}$ 는 제어 입력이다. 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법을 찾는 것은, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 유사하다. 그러나 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법은, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 다르다.

3.1 일반해를 이용하는 방법

이 방법의 기본 개념은 식(2)의 자코비안 행렬 방정식을 만족시키는  $\hat{e}$ 의 일반해를 구한 후, 이 일반해 중에서 식(4)를 만족시키는  $\hat{e}$ 를 찾는 것이다. 식(4)를 만족시키는  $\hat{e}$ 는, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 유사하다. 그러나 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법은, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 다르다.

$$\dot{\hat{e}}^* = -k \partial H(\theta, \hat{e}) / \partial \hat{e}, \quad k; \text{ 임의의 상수} > 0 \quad (6)$$

입을 보였다. 또한 [1.3.4]에서는  $H(\theta, \hat{e})$  이

$$H(\theta, \hat{e}) = \| \hat{x}_2 - J_2(\theta) \hat{e} \|_2 \quad (7)$$

로 주어지는 형태를 다루었으며 식(4)를 식(7)에 대입하여  $H(\theta, \hat{e})$ 을 최소로 하는  $\hat{e}$ 를 다음과 같이 구하였다.

$$\hat{e}^* = [J_2(I - J^+ J)]^+ (\hat{x}_2 - J_2^+ \hat{x}_e) \quad (8)$$

식(7)의 성능 지수  $H(\theta, \hat{e})$ 은 부가적으로 주어지는  $\hat{x}_2$ 에 대하여  $\hat{e}$ 이

$$\hat{x}_2 = J_2(\theta) \hat{e} \quad (9)$$

으로 주어지는 포하나의 행렬 방정식을 가장 잘 만족하는  $\hat{e}$ 를 구하기 위한 성능 지수이다.

따라서 물리적인 의미가 있도록 하면서 부가적 목표를 식(9)와 같이 설정하는 것은 [그림 3]에서와 같은 문제를 해결하는 방법이다. 그러나 식(9)를 이용하여 식(1)을 만족시키도록 하는 방법은, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 다르다.

3.2 각도만의 함수로 주어지는 성능 지수의 종류 및 의미

성능 지수  $H(\theta, \hat{e})$ 가 각도만의 함수로 주어지는 경우,  $H(\theta, \hat{e})$ 를 표기하기도 한다. 앞의 식(6)을 이용하여 식(6)을 만족시키도록 하는 방법은, 식(2)와 식(4)를 이용하여 식(1)을 만족시키도록 하는 방법과 다르다.

$$\hat{H}(\theta) = \frac{1}{2} (\theta - \theta_r)^T P (\theta - \theta_r) \quad (10)$$

을 제안하였다. 식(10)에서  $\theta_r$ 은 장애물 회피를 가능케 하는 피니톨레타의 관측 상태를 의미한다.

둘째, 특이성 회피(Singularity avoidance)를 위한로서 [2.6.31.32]에서는 다음과 같은 조작성 지수(mainp-lability index)를 정의하여 사용하였다.

$$\hat{H}(\theta) = \sqrt{\det(J(\theta)J^T(\theta))} \quad (11)$$

$$|K(\theta)| = |\det J(\theta)| \quad (12)$$

가 됨을 주목하면, 식(11)은 식(12)의 일반화로서 생각할 수 있고, 결국 식(11)을 최대로 하도록 어유자유타도의 제어 되는 것임을 알 수 있다.

자코비안 행렬과 관련된 로부트의 성능 지표로서는 [5]에서 언급한 자코비안의 조건수(condition number)란 개념이 있으며 다음과 같이 정의된다.

$$\text{Cond}(J) = \sigma_{\max} / \sigma_{\min} \quad (13)$$

$\sigma_{\max}$ 와  $\sigma_{\min}$ 은  $J^T J$ 의 고유치(eigenvalue)들 중에서 최대값과 최소값을 의미하며 행렬  $J$ 의 특이성(singular value) [27]이라고 한다. [27]에서는 식(13)의 조건수란

$$\| \hat{x}_2 \| / \| \hat{x}_e \| \leq \text{Cond}(J) \| \hat{e} \| / \| \hat{e} \| \quad (14)$$

의 관계를 갖고 있음을 보였는 바, 식(14)의 의미는  $\hat{x}_e$ 의 오차는  $\hat{e}$ 의 오차에 비해 조건수를 곱한 것보다 작다는 것으로부터, 일반적으로 오차가 로부트 순부분 운동의 오차로부터 가능한 한 작게 유발시키기 위해서는 조건수를 1에 선접해 가야 한다는 것이다. 그렇지만 식(14)를  $\hat{x}_e$ 로 선접하는 것은  $\sigma_{\max}$  및  $\sigma_{\min}$ 이  $\theta$ 에 대해 어떻게 변하는지(explicit form)가 아니어서 사용하기가 매우 어렵다고 생각된다. 이 밖에도 조작성 지표를 이용한 경로 계획에 관한 연구도 발표되어 있다 [34].

셋째로 [11,12]등은 주어지는 경로를 로부트 순부분 운동의 목표점까지 도달하게 하는 데 있어, 이 경로를 기설정된 운동 규칙의 영향을 가지지 못하도록 하기 위한 성능 지수로

$$\hat{H}(\theta) = \sum_{i=1}^n (\theta_i - \theta_{i,c})^2 / (\theta_{i,c} - \theta_{i,\max}) \quad (15)$$

를 사용하였다. 여기서  $\theta_{i,c}$ 는  $i$ 번째 관절모타의 허용 가능한 각도 범위 가운데 위치이며,  $\theta_{i,\max}$ 는  $i$ 번째 관절모타의 허용 가능한 각도 범위의 상한이다. 식(15)가 필요하게 된 계기를 잘 설명하였다. 그러나 식(15)는 RMR(resolved motion rate control) 방법의 주어진 궤적을 따라가는 작업을 여러 번 반복하여 로부트 동작 수행도를 따라가도록 하였다. 초기 설정된 로부트 동작(configuration)이 달라지는 소위 "singularity motion"이 발생하지 않는다는 이유에서 [그림 4참조], 맨 처음 설정된 동작으로 돌아갈 수 있는 관절 각도 값들이 여러 번 반복될 수 있으므로, 식(15)와 같은 관절모타의 속도 제한 영역을 넘지 않도록 하기 위한 성능 지수가 필요한 것이다.

이 밖에도 식(10), 식(11), 식(15) 등과 같은  $\hat{H}(\theta)$ 를 상대적인 중요성을 감안하여 가중치를 준 후 합하여 사용하는 경우도 있으며 [29], 주어진 상황에 따라서 적절히 식(6)을 바꾸어 주는 경우도 있다. [3]

3.3 일반해를 이용하지 않는 방법

[13,19,38]등에서는 일반해를 이용하지 않고 [문제 1]을 해결하는 방법을 제시하였는 바, [13]에서는 성능 지수

$$\hat{H}(\theta, \hat{e}) = \frac{1}{2} \hat{e}^T M_1 \hat{e} + M_2 \hat{e} \quad (16)$$

과 같이 특정만 형태로 선정하여,  $\hat{H}(\theta, \hat{e})$ 을 최소로 하는 최적해를 다음과 같이 구하였다.

$$\hat{e} = G \hat{x}_2 - (I - GJ) M_1^{-1} M_2$$

$$G = M_1^{-1} J^T (J M_1^{-1} J^T)^{-1} \quad (17)$$

여기서, 식(16)의  $H(\theta, \hat{e})$ 은 로부트의 각 관절모타에서 소모되는 동력 에너지의 미하는 것으로서,  $M_1$ 과  $M_2$ 는 모타의 동적 특성을 고려하여 선택되는 가중행렬이다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  
 $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$\dot{x}_e = (x_d^{(k+1)} - x_a^{(k)}) / \Delta t \quad (18)$$

식 (18)에서  $x_d^{(k+1)}$ 은  $(k+1)$  번째의 원하는 경로상의 위치이며,  $x_a^{(k)}$ 은  $k$  번째 로봇트의 실제 위치이다.

(29)에서 제시한 문제 (1)을 해결하는 방법은 상기 기술한 방법과 유사하게, 초기 조건을 최소화하는 제어 입력을 찾는 것이다.

$$x_d^{(k+1)} - x_a^{(k)} = J(\theta) (\theta^{(k+1)} - \theta^{(k)}) \quad (19)$$

에 대하여,  $x_d^{(k)}$ 가  $x_a^{(k)}$ 이 되도록 하는  $\theta^{(k)}$  가운데서  $H(\theta, \dot{\theta})$ 의 최소를 찾는 것이 목적이다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$x_d^{(k+1)} - x_a^{(k)} = J(\theta_R^{(k)}, \theta_R^{(k+1)}) + J(\theta_A^{(k)}, \theta_A^{(k+1)}) \quad (20)$$

식 (20)에서  $\theta_R$ 은 각 차원 벡터로서 로봇트 운동부의 운동 변수를 나타내며,  $\theta_A$ 는 각 차원 벡터로서, 여유 자유도를 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 기술한 방법의 다음은 원하는 위치에 도달하기 위한 방법이다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

#### 4. 전체 최적화를 위한 여유 자유도의 제어

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$P.I. = \int_{t_0}^{t_f} H(\theta, \dot{\theta}) dt \quad (21)$$

를 최소화 하는  $z$ 를 구하라. 단, 초기조건은 식 (1)에서

$$x_e(t_0) = f(\alpha, t_0) \quad (22)$$

으로서 기구학적 관계식을 의미한다.

(문제 2)는 고정 초기시간 (fixed initial time), 고정 종말시간 (fixed final time), 고정 초기 위치 (fixed initial position), 고정 종말 위치 (fixed terminal position), 고정 초기 속도 (fixed initial velocity), 고정 종말 속도 (fixed terminal velocity)를 갖는 문제이다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$\hat{H}(\lambda, \theta, t, z) = H(\theta, \dot{\theta}) + \lambda^T (J^T x_e + (I - J^T J) z) \quad (23)$$

으로 주어지며, 최적의 경로의 경로  $\theta(t)$ 는

$$\dot{\theta} = (\partial \hat{H} / \partial \lambda)^T \quad (24)$$

$$\dot{\lambda} = -(\partial \hat{H} / \partial \theta)^T \quad (25)$$

로 주어지는 2차원 미분 방정식의 해로서, 이때  $z$ 는 임의의 값으로 주어지며,  $\lambda$ 는 임의의 값으로 주어진다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$\lambda(t_f) = 0 \quad (26)$$

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$(I - J^T) J(\theta(t_f)) \lambda(t_f) = 0 \quad (27)$$

을 만족해야 한다. 식 (27)에서  $(I - J^T)$ 의 랭크 (rank)가  $(n-m)$  이므로, 식 (27)의 조건이  $(n-m)$  이 된다. 한편, 제단점 상용기수 P.I.는

$$P.I. = \int_{t_0}^{t_f} \dot{\theta}^T \theta dt \quad (28)$$

$$P.I. = \int_{t_0}^{t_f} (\sqrt{\det J^T J}) + \dot{\theta}^T \theta dt \quad (29)$$

로서,  $\dot{\theta}^T \theta$ 은 최적 제어 문제에서 임의의 특이성 (singularity)을 피하고자 포함된 것이다.

#### 5. 검토 및 연구 과제

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

$$\dot{\theta} = J^T (\ddot{x}_e - \ddot{J} \dot{\theta}) + (I - J^T J) \dot{\theta} \quad (30)$$

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

이러한 경우, 초기 조건을 만족하는 해를 구하는 것은 미분방정식 (29.30)에서  $\dot{x} = f(x, u, t)$ 의 해를 구하는 것과 동치이다. 여기서  $x(0) = x_0$ 는 초기 조건을 나타내며,  $x(t_f) = x_f$ 는 종말 조건을 나타낸다. 이 경우, 최적 제어 문제를 해결하기 위해서는, 목적 함수를 최소화하는 제어 입력을 찾는 것이 필요하다. 이는 일반적으로 미분방정식 (29.30)을 만족하는 제어 입력을 찾는 문제로 귀결된다.

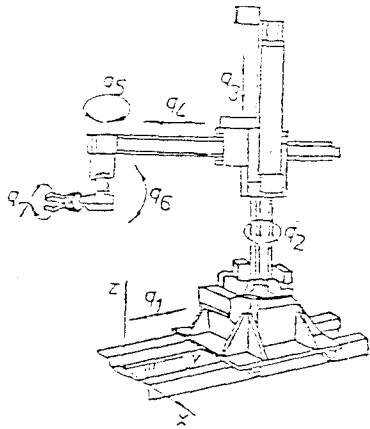


그림 1. 7 자유도 산업용 로봇의 한 예

Fig.1. An Industrial Robot with 7 d.o.f.

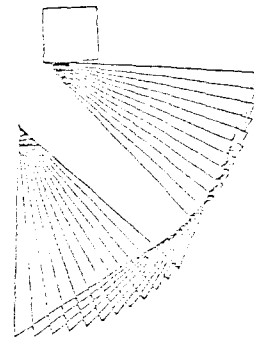


그림 4. 가장 역변환 제어 (Whitney 방법) 에 의한 반복 작업시의 초기자세

Fig.4. Initial Configuration during iterative operations under pseudo inverse control

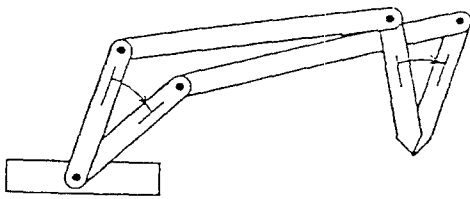


그림 2 링크들의 자기운동

Fig.2. Self-motion of the linkages

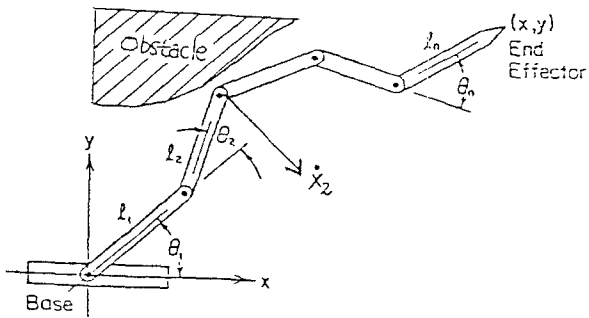


그림 3. 장애물 회피를 위한 여유자유도 제어

Fig.3. Control of redundancy for obstacle avoidance

참 고 문 헌

1. Y. Nakamura and H. Hanafusa, "Task priority based redundancy control of robot manipulators," Proceedings of 2nd Int. Symp. of Robotics Research, Kyoto, Japan, pp. 357-364, 1984.
2. T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," 24th IEEE Conference on Decision and Control, pp.1004-1009, 1985.
3. A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments" Int. J. of Robotics Research, Fall, pp.109-117, 1985.
4. H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," Preprints of the 8th IFAC Triennial World Congress, Kyoto Japan, pp. XIV-78-XIV-83, 1981.
5. C.A. Klein, "Use of redundancy in the design of robotic systems," 24th IEEE Conf. on Decision and Control, pp. 207-214, 1985.
6. T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," Robotics Research: The First Int. Symp., ed. M. Brady and R. Paul, Cambridge, Mass., MIT Press, pp.735-748.
7. M. M. Stanisic and G. B. Pennock, "A nondegenerate orientation solution of a fourjointed wrist," Proc. of 24th IEEE Conf. on Decision and Control, pp. 998-1003, 1985.
8. W. D. Fisher, "The Kinematic control of redundant manipulators," Ph.D dissertation, School of Elec. Eng., Purdue University, 1984.
9. M. S. Konstantinov et al., "Kinematic control of redundant manipulators," Proceedings of
10. M. S. Konstantinov et al., "A contribution to the inverse kinematic problem for industrial robots, Proceedings of 12th ISIR, pp. 459-467, 1982.
11. A. Liegeois, "Automatic supervisory control of the configuration and behavior of Multibody mechanisms," IEEE Trans. Syst. Man, Cybern., Vol. SMC-7, No. 12, pp. 868-871, 1977.
12. C.A. Klein, and C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," IEEE Trans. Syst. Man, Cybern., vol. SMC-13, No. 3, pp. 245-250, 1983.
13. M. Vukobratovic and M. Kircanski, "A dynamic approach to nominal trajectory synthesis for redundant manipulators," IEEE Trans. Syst. Man, Cybern., vol. SMC-14, No. 4, pp. 580-586, 1984.
14. J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," 24th IEEE Conference on Decision and Control, pp. 1016-1021, 1985.
15. J. P. Trevelyan, P.D. Kovesi, and M.C.H. Ong, "Motion control for a sheep shearing robot," Robotics Research: The First International Symposium, ed. M.Brady and R.Paul, Cambridge Mass, MIT Press, pp. 175-190, 1984.
16. J.M. Hollerbach, "Optimum kinematic design for a seven degree of freedom manipulator," Robotics Research: 2nd Int. Symp., ed. H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 216-222, 1984.
17. K. Kaneko, "Basic experiments on a hexapod walking machine (MELWALK-III) with an approximate straight-line link mechanism," Proc. of '85 ICAR, Tokyo, Japan, pp. 397-404, 1985.
18. M. Benati, P. Morasso and V. Tagliacozzo, "The inverse kinematic problem for anthropomorphic manipulator arms," ASME J. of Dyna. Syst. and Control, vol. 104, pp. 110-113, 1982.
19. Y. Umetani, Private Communication.
20. D.E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," ASME J. Dynamic Systems, Meas., Control, pp. 303-309, 1972.
21. L.T. Wang and B. Ravani, "Recursive computations of kinematic and dynamic equations for mechanical manipulators," IEEE J. of Robotics and Automation, vol. RA-1, No. 3, pp. 124-131, 1985.
22. D. E. Orin and W. W. Schrader, "Efficient computation of the Jacobian for robot manipulators," Int. J. Robotics Res., vol. 3, No. 4, pp. 66-75, 1984.
23. R.P. Paul, Robot Manipulators, MIT Press :Cambridge, Mass., 1981.
24. S. Barnett, Matrices in control theory, Van Nostrand Reinhold Company: London, 1971.
25. J.Y.S. Luh and Y.L. Gu, "Industrial robots with seven joints," 24th IEEE Conf. on Decision and control, pp/. 1010-1015, 1985.
26. J. Baillieul, "Kinematic programming alternatives for redundant manipulators," 24th IEEE Conf. on Decision and Control, pp. 722-728, 1985.
27. V.C. Klema and A-J. Laub, "The singular value decomposition: its computation and some applications," IEEE Trans. Automat. Contr., vol. AC-25, No. 2, Apr. 1980.
28. J.K. Salisbury and J.J. Craig, "Articulated hands: Force control and kinematic issues," International Journal of Robotics Research, vol. 1, No. 1, pp. 4-17, 1982.
29. A.A. Goldenberg, B. Benhabib and R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," IEEE J. of Robotics and Automation, vol. RA-1, No. 1, pp. 14-20, 1985.
30. A.A. Goldenberg and D.L. Lawrence, "A Generalized solution to the inverse kinematics of robotic manipulators," ASME J. of Dynamic Syst., Meas., Contr., vol. 107, pp. 103-106, 1985.
31. T. Yoshikawa, "Dynamic manipulability of robot manipulators," J. of Robotic Systems, vol. 2, pp. 113-124, 1985.
32. T. Yoshikawa, "Manipulability of robot mechanisms," Preprints of the 2nd Int. Symp. of Robotics Research, Kyoto, Japan, pp. 91-98, 1984.
33. H. Asama and H. Yoshikawa, "Development of a metamorphic manipulator with 9 degrees of freedom," ibid. pp. 415-422, Tokyo, Japan, 1985.
34. M. Unkiyama, K. Shimizu, and K. Hakemori, "Performance evaluation of manipulators using the Jacobian and its application to trajectory planning," Preprints of the 2nd Int. Symp. of Robotics Research, Kyoto, Japan, pp. 99-106, 1984.
35. J. Furusho and S. Onishi, "An efficient approach for solving the inverse kinematics of manipulators," Proceedings of 15th ISIR, pp.1051-1058, Tokyo, Japan, 1985.
36. J.Y.S. Luh, M.W. Walker and R.P.C. Paul, "Resolved-acceleration control of mechanical manipulators," IEEE Trans. Automat. Contr. vol. AC-25, No. 3, pp. 195-200, 1980.
37. C. Wu and R.P. Paul, "Resolved motion force control of robot manipulators," IEEE Trans. Syst. Man and Cybern., vol. SMC-12, No.3, pp.266-275, 1982.
38. M. Athans and P.L. Falb, Optimal control: An introduction to the theory and its application, McGraw-Hill, New York, 1966.