

작업 전이 알고리즘 기반 로봇 동작 제한 극복 프레임워크

A Unified Framework for Overcoming Motion Constraints of Robots Using Task Transition Algorithm

장근우¹ · 김상현¹ · 박수한¹ · 박재흥[†]

Keunwoo Jang¹, Sanghyun Kim¹, Suhan Park¹, Jaeheung Park[†]

Abstract: This paper proposes a unified framework that overcomes four motion constraints including joint limit, kinematic singularity, algorithmic singularity and obstacles. The proposed framework is based on our previous works which can insert or remove tasks continuously using activation parameters and be applied to avoid joint limit and singularity. Additionally, we develop a method for avoiding obstacles and combine it into the framework to consider four motion constraints simultaneously. The performance of the proposed framework was demonstrated by simulation tests with considering four motion constraints. Results of the simulations verified the framework's effectiveness near joint limit, kinematic singularity, algorithmic singularity and obstacles. We also analyzed sensitivity of our algorithm near singularity when using closed loop inverse kinematics depending on magnitude of gain matrix.

Keywords: Task Transition, Joint Limit Avoidance, Singularity Avoidance, Obstacle Avoidance

1. 서 론

직교 좌표계(Cartesian Space)에서 정의된 로봇의 말단 장치의 위치와 방위로 움직이기 위해서는 로봇의 관절 위치 값을 구해야한다. 이를 역기구학(Inverse Kinematics)라고 부르고, 역기구학의 해는 일반적으로 해석적인 해(Analytic Solution)와 수치적인 해(Numerical Solution)로 나눌 수 있다^[1].

해석적인 해는 정기구학(Forward Kinematics)에서 구한 삼각함수를 포함한 비선형식을 기반으로 각 관절 위치 해를 직

접 구한다^[2-4]. 수치적인 해는 로봇의 말단 장치 속도 벡터와 로봇의 관절 속도 벡터 사이의 관계를 선형화시킨 자코비안(Jacobian)의 역행렬을 이용하여 구한다^[5-8]. 해석적인 해의 경우, 보통 역삼각함수를 통해 관절 위치 해를 얻는데 해의 조합에 따라 로봇의 자세가 다양해진다. 또한 7자유도 이상의 로봇은 해를 무한히 구할 수 있어 이를 고려하여 해를 구해야 한다. 이에 비해 수치적인 해는 자코비안의 의사역행렬(Pseudo-inverse Matrix)을 통해 상대적으로 쉽게 해를 구한다. 또한, 자코비안의 영공간을 활용하여 복수의 작업을 보다 쉽게 설계하며, 로봇의 기구학적 구조에 의존적이지 않아 다양한 형태의 로봇에서 수치적인 해를 구할 수 있다.

하지만 자코비안을 이용해 해를 구하기 전에 말단 장치의 위치와 방위가 작업 공간(Workspace) 내에 있는지 확인해야 한다. 이 때, 작업 공간은 관절 한계, 특이점, 장애물과 같은 제한 조건들이 반영되어야 한다. 이러한 제한 조건들은 작업을 시작하기 전에 반영하여 말단 장치의 경로를 계획(Path Planning)하기도 하고, 작업을 하는 도중에 반영하기도 한다. 그러나 경로를 계획하는 방법은 계산 시간이 오래 걸려 로봇이 작업하면서 발생할 수 있는 외란이나 환경 변화에 대해 실시간으로 반영하는 것이 어렵다.

Received : Feb. 6. 2018; Revised : Apr. 13. 2018; Accepted : Apr. 30. 2018

※ This work was mainly supported by Industrial strategic technology development program (No. 10060121) funded by the Ministry of Trade, Industry & Energy (MI, Korea). And this work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055798) and by Industrial strategic technology development program (No. 10077538) funded by the Ministry of Trade, Industry & Energy (MI, Korea).

1. Graduate School of Convergence Science and Technology, Seoul National University (jkw0701@snu.ac.kr, ggory15@snu.ac.kr, psh117@snu.ac.kr)

† Corresponding author: Graduate School of Convergence Science and Technology, Seoul National University, Gwanggyo-ro, Yeongtong-gu, Suwon, Korea. Advanced Institutes of Convergence Technology, Gwanggyo-ro, Yeongtong-gu, Suwon, Korea (park73@snu.ac.kr)

이러한 단점 때문에 작업을 하는 도중에 제한조건을 반영하는 연구가 활발히 진행되어 왔다. H. Zghal et al.^[9]은 각 관절 위치 값과 각 관절의 상한치, 하한치로 이루어져 있는 성능 기준(Performance Criterion)을 만들었다. 이에 대한 구배(Gradient)를 자코비안의 영공간(Null-space)에 대입한 관절 속도 해를 통해 관절 한계를 회피하였다. Y. Nakamura and H. Hanafusa^[10]은 자코비안의 특이값(Singular Value)에 조정 항을 더하였다. 이는 특이값이 영으로 수렴하더라도 자코비안의 의사역행렬이 항상 존재하게 한다. 이를 통해 특이점을 회피하였다. B. Dariush et al.^[11]은 로봇과 장애물 사이의 거리에 따른 전위 함수를 도입하고 이에 대한 구배값을 이용해 가중행렬을 구성하였다. 가중행렬을 반영한 자코비안의 의사역행렬을 통해 해를 구하여 장애물을 회피하였다. 이외에도 많은 연구에서 수치적인 해를 구하는 과정에서 제한 조건을 고려하여 회피하였다^[12-20].

앞서 소개한 연구들을 포함한 여러 연구들에서는 두 가지 한계점이 존재한다. 첫째로 제한 조건들을 회피하는 작업을 더 우선으로 수행해야 할 상황에 대한 고려가 없었다^[9,11,17]. 이 연구들은 모두 주 작업 자코비안의 영공간 내에서 회피 작업을 설계하였기 때문에 제한 조건에 대한 회피를 보장할 수 없었다. 둘째로 작업을 연속적으로 삽입 및 제거하는 것에 대한 고려가 없었다^[10,13,14]. 즉, 회피 작업을 삽입 및 삭제할 때 생기는 수치적 해의 불연속성을 고려하지 않았다.

Lee et al.^[21]은 위의 두 가지 한계점을 극복하고자 수치적 역기구학 기반 작업 전이(Task Transition) 알고리즘을 제안하였고, 이를 이용하여 관절 한계를 회피하였다. 또한 Kim and Park^[22]은 이에 대한 후속 연구로 자코비안의 특이값 분해(Singular Value Decomposition)로 얻은 특이값을 통해 특이점을 판별하고 작업 전이 알고리즘을 적용하여 특이점을 회피하였다.

본 논문에서는 작업 전이 알고리즘을 기반으로 위의 두 선행 연구^[21,22]에서 고려하지 않은 장애물을 회피하는 알고리즘을 제안한다. 이 알고리즘은 장애물과 가까워지면 장애물과 가장 가까운 로봇 링크 위의 점을 계산하고 그 점이 장애물에서 멀어지도록 하는 작업을 생성한다. 이를 작업 전이 알고리즘에 적용하여 장애물을 회피한다. 추가적으로, 본 논문에서는 앞의 두 연구와 장애물 회피 알고리즘을 통합한 프레임워크를 제시한다. 프레임워크는 각 제한 조건에 우선순위를 두어 총 4개의 계층적 구조로 이루어져있고, 관절 한계, 특이점, 그리고 장애물을 동시에 고려하면서 극복할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 작업 전이 알고리즘을 소개하고 이를 이용한 장애물 회피 알고리즘을 설명한다. 또한, 앞서 소개한 두 연구^[21,22]의 관절 한계 회피 방법, 특이점 회피 방법과 통합한 프레임워크를 정리하면서 2장을 마

무리한다. 3장에서는 7자유도 로봇 매니플레이터 MITSU BISHI PA-10^[23]을 시뮬레이션 환경에서 프레임워크를 검증한 결과 및 분석을 담고 있으며, 4장에서 본 논문을 결론짓는다.

2. 통합 프레임워크

이 장에서는 선행 연구로써 작업의 삽입 및 제거를 연속적으로 하여 해의 불연속성을 없앤 작업 전이 알고리즘^[21]을 정리한다. 또한, 이에 대한 후속연구로써 작업 전이 알고리즘을 이용한 장애물 회피 알고리즘을 제안한다. 마지막으로, 관절 한계, 특이점, 장애물을 모두 고려할 수 있는 프레임워크로 확장시킨다.

2.1 작업 전이 알고리즘

두 작업($\dot{x}_1^d \in \mathbb{R}^{m_1}$, $\dot{x}_2^d \in \mathbb{R}^{m_2}$)과 이에 해당하는 자코비안($J_1 \in \mathbb{R}^{m_1 \times n}$, $J_2 \in \mathbb{R}^{m_2 \times n}$ ($m_1 + m_2 \leq n$))으로 작업과 로봇 관절 속도($\dot{q} \in \mathbb{R}^n$)와의 관계를 식 (1)로 나타낼 수 있다. 여기서 1번 작업만 수행하는 관절 속도 해는 식 (2)이다.

$$\dot{x}_1^d = J_1 \dot{q}, \quad \dot{x}_2^d = J_2 \dot{q} \quad (1)$$

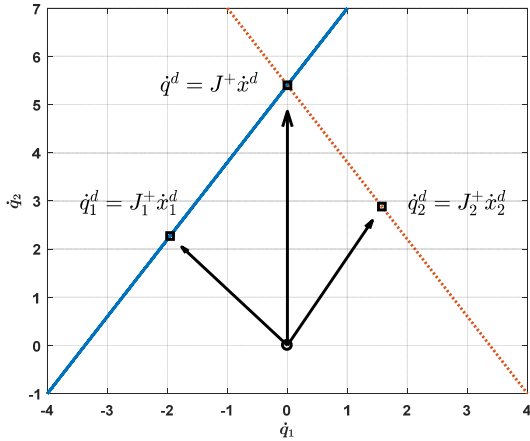
$$\dot{q} = J_1^+ \dot{x}_1^d \quad (2)$$

$\dot{q}^d \in \mathbb{R}^n$ 은 전체 관절 속도 해, $J_1^+ \in \mathbb{R}^{n \times m_1}$ 은 J_1 의 의사역행렬을 의미한다. 2번 작업을 삽입하여 동시에 수행한다고 하면, 해는 식 (3)와 같이 구한다. 작업을 삽입하면서 해는 식 (2)에서 식 (3)으로 바뀌고 이 순간에 해의 불연속성이 발생한다. 그 이유는 의사역행렬이 \dot{q}^d 의 2-노름(2-norm)을 최소화시키기 때문이다. 이를 [Fig. 1]에 m_1, m_2, n 이 각각 1, 1, 2일 때에 대해 나타내었다. [Fig. 1]에서 파란색 실선과 주황색 점선은 각각 \dot{x}_1^d, \dot{x}_2^d 에 대한 해를 나타낸다. \dot{q}_1, \dot{q}_2 은 로봇의 첫 번째, 두 번째 관절 속도이다. 한 작업만 수행하던 중에 작업을 추가하여 해를 구하면 두 선의 교점이 해가 된다.

$$\dot{q}^d = J^+ \dot{x}^d \quad (3)$$

$$J = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}, \quad \dot{x}^d = \begin{bmatrix} \dot{x}_1^d \\ \dot{x}_2^d \end{bmatrix}$$

이처럼 작업의 삽입 및 제거 중에 발생하는 해의 불연속성을 고려한 프레임워크가 작업 전이 알고리즘이다. 작업 전이



[Fig. 1] Discontinuity of solution due to an abrupt insertion of a task

알고리즘은 0에서 1사이의 실수 값을 갖는 활성화 지표 (Activation Parameter)라는 개념을 도입해 각 작업에 부여한다. 이 값을 0에서 1로 연속적으로 변화시켜서 작업을 삽입, 1에서 0으로 연속적으로 변화시켜 작업을 제거한다. 1번 작업 (\dot{x}_1^d)의 우선순위가 2번 작업(\dot{x}_2^d)보다 높으면, 일반적인 해는 식 (4)로 얻는다.

$$\begin{aligned} \dot{q}^d &= J_1^+ \dot{x}_1^d + N_1 \dot{q}_{21} \\ N_1 &= I - J_1^+ J_1 \\ \dot{q}_{21} &= (J_2 N_1)^+ (\dot{x}_2^d - J_2 J_1^+ \dot{x}_1^d) \end{aligned} \quad (4)$$

$N_1 \in R^{n \times n}$ 은 J_1 의 영공간, $I \in R^{n \times n}$ 은 단위행렬, $\dot{q}_{21} \in R^n$ 은 J_1 의 영공간에서 수행되는 2번 작업의 관측 속도 해를 의미한다. 여기에 1번 작업과 2번 작업에 활성화 지표 h_1, h_2 을 반영한 식은 다음과 같다.

$$\begin{aligned} \dot{q}^d &= J_1^+ \dot{x}_1^i + (J_2 N_1)^+ (\dot{x}_2^i - J_2 J_1^+ \dot{x}_1^i) \\ \dot{x}_1^i &= h_1 \dot{x}_1^d + (1 - h_1) J_1 J_2^+ h_2 \dot{x}_2^d \\ \dot{x}_2^i &= h_2 \dot{x}_2^d + (1 - h_2) J_2 J_1^+ h_1 \dot{x}_1^d \end{aligned} \quad (5)$$

식 (5)을 통해 우선순위를 1번 작업에 두면서 h_1, h_2 을 이용하여 1번 작업과 2번 작업의 삽입 및 제거를 연속적으로 할 수 있다.

$$\begin{aligned} \dot{q}^d &= J_1^+ \dot{x}_1^i + \sum_{n=2}^p \dot{q}_n^i \\ \dot{q}_n^i &= N_{[n-1]}^+ J_n^+ (\dot{x}_n^i - J_n \sum_{k=1}^{n-1} \dot{q}_k^i) \quad (n \geq 2) \end{aligned} \quad (6)$$

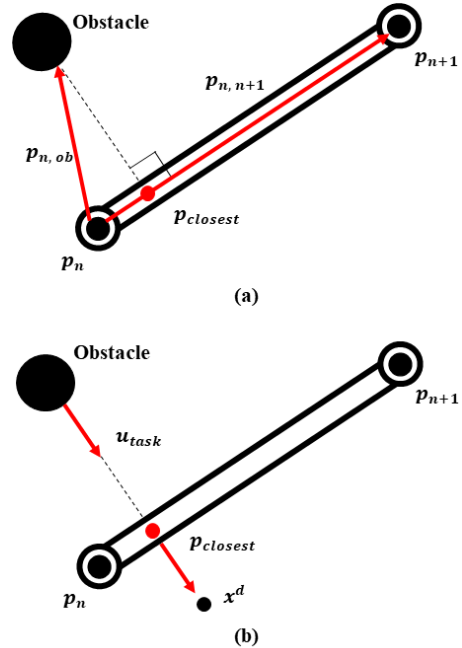
$$\begin{aligned} N_{[k]} &= \prod_{s=1}^k N_{s|s-1} \quad (k \geq 2, N_{[1]} = N_1) \\ \dot{x}_n^i &= h_n \dot{x}_n^d + (1 - h_n) J_n \dot{q}_{[n]}^d \\ \dot{q}_{[n]}^d &= J_1^+ \dot{x}_1^i + \sum_{r=2}^{n-1} \dot{q}_r^i + \sum_{r=n+1}^p \dot{q}_r^i \end{aligned} \quad (7)$$

비슷한 방법으로 p 개의 작업에 대해서도 확장하여 식(6), (7)에 정리하였다. 우선순위는 1번 작업이 가장 높으며 순서대로 낮아지며 p 번 작업이 가장 낮다.

2.2 장애물 회피 알고리즘

로봇이 장애물을 회피하기 위해서는 로봇이 장애물과 가까워지고 있는지에 대한 판별과 이를 반영한 움직임이 필요하다. 판별은 로봇과 장애물 사이의 한계 거리를 설정하여 한다. 판별 후, 움직임을 만들기 위해 장애물에서 로봇을 가리키는 위치 벡터를 구하고, 이를 이용해 장애물과 가장 가까운 로봇 링크 위의 점을 기준으로 속도 성분을 만든다. 결론적으로, 로봇과 장애물사이의 거리에 따른 활성화 지표를 설정하여 작업을 삽입하여 장애물을 회피한다.

판별을 위해 먼저 장애물과 가장 가까운 로봇 링크 위의 점을 계산한다. [Fig. 2(a)]와 같이 n 번째 링크와 장애물이 있다



[Fig. 2] (a) Illustration of position vectors for robot and obstacle (b) Illustration of desired position vector for avoiding an obstacle

고 가정한다. 원점에서 n 번째 관절까지의 위치 벡터 $p_n \in R^3$, 원점에서 $n+1$ 번째 관절까지의 위치 벡터 $p_{n+1} \in R^3$, n 번째 관절에서 장애물을 가리키는 벡터 $p_{n,ob} = p_{ob} - p_n \in R^3$ 을 정의한다. p_{ob} 은 원점에서 장애물을 가리키는 위치 벡터이고, 마지막으로 n 번째 관절에서 $n+1$ 번째 관절을 가리키는 위치 벡터 $p_{n,n+1} \in R^3$ 을 정의한다. 위에 정의한 위치 벡터들의 내적 비율 α 을 식 (8)로 정한다.

$$\alpha = \frac{p_{n,n+1} \cdot p_{n,ob}}{p_{n,n+1} \cdot p_{n,n+1}} \quad (8)$$

α 의 범위에 따라 n 번째 링크 위에서 장애물과 가장 가까운 위치 벡터 $p_{n,dolest} \in R^3$ 을 구할 수 있다. 이를 식 (9)에 나타내었다.

$$p_{n,dolest} = \begin{cases} p_n & \text{if } \alpha < 0 \\ p_{n+1} & \text{if } \alpha > 1 \\ p_n + \alpha p_{n,n+1} & \text{else} \end{cases} \quad (9)$$

식 (9)을 이용해 모든 링크에 대해 위치 벡터들을 구하고 p_{ob} 와의 거리를 비교한다. 그 중에서 장애물과 가장 가까운 위치 벡터 $p_{dolest} \in R^3$ 와 p_{ob} 와의 거리를 이용해 장애물과 가까워지고 있는지 판별한다.

다음으로, 움직임을 만들어낼 작업과 자코비안을 만든다. 이를 위해 장애물에서 멀어지는 방향의 단위 벡터 u_{task} 와 장애물을 회피하기 위한 목표 위치 벡터 x^d 을 식 (10), (11)과 [Fig. 2(b)]에 그림으로 나타내었다.

$$u_{task} = \frac{p_{dolest} - p_{ob}}{\|p_{dolest} - p_{ob}\|_2} \quad (10)$$

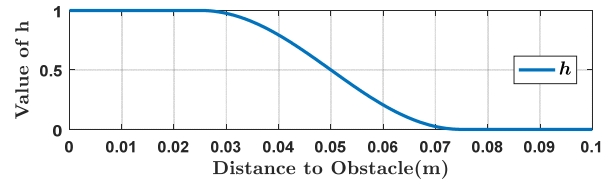
$$x^d = p_{dolest} + k_p u_{task} \quad (11)$$

위의 정의한 목표 위치는 3자유도이지만 1자유도만 사용하여 장애물을 회피하기 위해 u_{task} 의 전치를 이용하여 (12), (13)과 같이 목표 작업과 자코비안을 회전시킨다.

$$\dot{x}_{task} = u_{task}^T (x^d - p_{dolest}) \in R \quad (12)$$

$$J_{task} = u_{task}^T J_{dolest} \in R^{1 \times n} \quad (13)$$

k_p 은 이득 상수이고, J_{dolest} 은 로봇의 원점부터 p_{dolest} 까지의 자코비안 행렬이다.



[Fig. 3] Value of activation parameter depending on distance between robot and obstacle

마지막으로 로봇과 장애물 사이의 거리에 따라 활성화 지표를 정하고, 작업 전이 알고리즘을 적용하여 작업을 삽입 및 삭제하여 장애물을 회피한다. 로봇과 장애물 사이의 거리에 따른 활성화 지표는 식 (14)와 [Fig. 3]에 나타내었다.

$$h = \begin{cases} f(\rho) & \text{if } \rho < \beta, \quad \rho \geq \beta - \gamma \\ 1.0 & \text{if } \rho < \beta - \gamma \\ 0.0 & \text{else} \end{cases} \quad (14)$$

$$f(\rho) = 0.5 + 0.5 \sin\left(\frac{\pi}{\gamma}(\rho - \beta) - \frac{\pi}{2}\right)$$

$\rho = \|p_{ob} - p_{dolest}\|_2$ 은 로봇과 장애물 사이의 거리, β, γ 은 각각 한계 거리와 버퍼(Buffer) 영역의 폭을 나타낸다. [Fig. 3]은 β 은 0.075 m, γ 은 0.05 m일 때의 활성화 지표를 나타내었다. 따라서 로봇과 장애물 사이의 거리가 0.075 m부터 활성화 지표가 0부터 증가하기 시작하면서 작업이 삽입된다.

2.3 통합 프레임워크

통합 프레임워크는 관절 한계, 특이점, 장애물에 대한 회피 작업을 하나의 프레임워크에서 적용하도록 정리한다. 먼저 우선순위에 따라 작업의 번호를 부여한다. 우선순위가 제일 높은 1번 작업은 관절 한계 회피 작업, 2번 작업은 장애물 회피 작업, 3번 작업은 자코비안의 특이값에 영향을 받지 않는 작업이다. 4번 작업은 특이값에 영향을 받는 작업이다. 4번 작업은 특이값에 따라 활성화 지표를 이용하여 연속적으로 제거하면 특이점 회피 작업을 수행할 수 있다^[22]. 식 (6), (7)을 기반으로 통합 프레임워크를 식 (15-17)과 같이 구성할 수 있다.

$$\dot{q}^d = \dot{q}_1^i + \dot{q}_2^i + \dot{q}_3^i + \dot{q}_4^i \quad (15)$$

$$\begin{aligned} \dot{q}_1^i &= J_1^+ \dot{x}_1^i \\ \dot{q}_2^i &= (J_2 N_1)^+ (\dot{x}_2^i - J_2 \dot{q}_1^i) \\ \dot{q}_3^i &= (J_3 N_1 N_{21})^+ (\dot{x}_3^i - J_3 (\dot{q}_1^i + \dot{q}_2^i)) \\ \dot{q}_4^i &= (J_4 N_1 N_{21} N_{3|21})^+ (\dot{x}_4^i - J_4 (\dot{q}_1^i + \dot{q}_2^i + \dot{q}_3^i)) \end{aligned} \quad (16)$$

$$\begin{aligned}
 N_{21} &= I - (J_2 N_1)^+ (J_2 N_1) \\
 N_{321} &= I - (J_3 N_1 N_{21})^+ (J_3 N_1 N_{21}) \\
 \dot{x}_1^i &= h_1 \dot{x}_1^d + (1 - h_1) J_1 \dot{q}_{[1]}^d \\
 \dot{x}_2^i &= h_2 \dot{x}_2^d + (1 - h_2) J_2 \dot{q}_{[2]}^d \\
 \dot{x}_3^i &= h_3 \dot{x}_3^d + (1 - h_3) J_3 \dot{q}_{[3]}^d \\
 \dot{x}_4^i &= h_4 \dot{x}_4^d + (1 - h_4) J_4 \dot{q}_{[4]}^d
 \end{aligned} \tag{17}$$

$\dot{q}^d \in R^n$ 는 목표 관절 속도 해, $\dot{x}_i^i \in R^m$ 는 m 개 관절에 대한 관절 한계 회피 작업, $J_1 \in R^{m \times n}$, h_1 는 그에 대한 자코비안과 활성화 지표, $\dot{x}_2^i \in R^1$ 는 장애물 회피 작업, $J_2 \in R^{1 \times n}$, h_2 는 그에 대한 자코비안과 활성화 지표, $\dot{x}_3^i \in R^{p-r}$ 는 특이값에 영향을 받지 않는 작업, $J_3 \in R^{(p-r) \times n}$, h_3 는 그에 대한 자코비안과 활성화 지표이다. $\dot{x}_4^i \in R^r$ ($r \geq 1$)는 특이점 회피 작업, $J_4 \in R^{r \times n}$, h_4 는 그에 대한 자코비안과 활성화 지표이다. \dot{x}_3^i, \dot{x}_4^i 는 목표 경로 추종 작업 $\dot{x}^d \in R^p$ 가 특이값 분해된 자코비안의 직교 방향 벡터들에 의해 회전된 것이고, J_3, J_4 는 목표 경로 추종 작업의 자코비안 $J \in R^{p \times n}$ 이 특이값 분해된 자코비안의 직교 방향 벡터들에 의해 회전된 것이다. p 는 목표 경로 추종 작업의 차원, r 는 특이값이 0에 가까운 방향 벡터들의 수, $I \in R^{n \times n}$ 은 단위 행렬이다.

3. 실험

이 장에서는 2장에서 제안한 장애물 회피 알고리즘과 통합 프레임워크를 시뮬레이션 환경에서 검증한다. 3.1절에서 실험 환경을 설명하고 3.2절에서 3가지 제한 조건에 대한 각각의 회피 작업과 통합 프레임워크를 실험한 결과를 정리한다. 또한, 페루프 역기구학의 되먹임 이득 행렬에 대한 민감도 분석을 추가적으로 진행하였는데, 이를 이 장의 마지막 절에 정리하였다.

3.1 실험 환경

통합 프레임워크를 구현하기 위해 [Fig. 4]와 같이 7자유도 로봇 매니퓰레이터 MITSUBISHI PA-10^[23]을 사용하였다. [Fig. 4]에 각 링크, 각 관절 축 위치와 방향을 나타내었다. 이 로봇을 가상현실에서 제어하기 위해 로봇 시뮬레이터 V-REP^[24]을 사용하였고, 물리엔진으로는 Vortex를 선택하였다. 시뮬레이션을 수행한 컴퓨터의 사양은 Intel i7 4.0GHz, RAM 16GB이다. 로봇의 제어주기는 200 Hz이고, 로봇의 제어

입력은 로봇의 관절 위치 값이다.

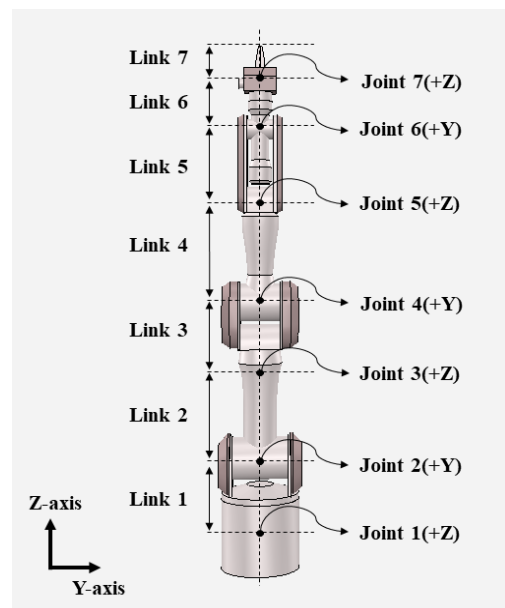
목표 경로를 추종하는 작업에 대해서는 관절 속도 해를 수치 적분하여 생긴 추종 오차를 없애기 위해 식 (18)과 같이 페루프 역기구학(Closed Loop Inverse Kinematics)을 이용하여 관절 위치 값을 입력하였다. 이외의 모든 회피 작업에 대해서는 단순히 관절 속도 해를 수치 적분하여 관절 위치 값을 입력하였다.

$$\begin{aligned}
 \dot{q}^d &= J^+ (\dot{x}^d + Ke) \\
 q^d &= q^{curr} + \dot{q}^d \Delta t
 \end{aligned} \tag{18}$$

$\dot{q}^d \in R^7$ 은 관절 속도 해, $J^+ \in R^{7 \times s}$ 은 해당 작업에 대한 자코비안 의사 역행렬, $K \in R^{s \times s}$ 은 되먹임 이득(Gain) 행렬, $\dot{x}^d \in R^s$ 은 목표 속도, $e = x^d - x \in R^s$ 은 위치 추종 오차, s 은 작업의 차원, $q^d \in R^7$, $q^{curr} \in R^7$ 은 각각 제어 입력, 현재 관절 위치 값, Δt 은 제어 주기이다.

3.2 실험 결과

실험 결과는 제시한 통합 프레임워크를 기반으로 각 회피 작업을 수행하여 검증하였다. 통합 프레임워크를 검증하기 전에, 실험에 사용된 (15-17)의 차원을 식 (19)과 (20)에 정리한다. 목표 경로 추종 작업은 직교 좌표계 위치와 Y-축 방향의 방위로 정하였고, 활성화 지표를 적용할 때에는 (18)에서와 같이 목표 속도향과 위치 추종 오차향의 합을 전체 작업으로 정의한다. 관절 한계 회피의 경우 4번째 관절만 고려하였고, 장애



[Fig. 4] 7-DOF manipulator MITSUBISHI PA-10

물 회피의 경우도 하나의 장애물만 고려하였다.

$$\dot{q}^d \in \mathbb{R}^7, \dot{x}^d \in \mathbb{R}^4, J \in \mathbb{R}^{4 \times 7}, K = 400I \in \mathbb{R}^{4 \times 4} \quad (19)$$

$$\begin{aligned} \dot{x}_1^d &\in \mathbb{R}, J_1 \in \mathbb{R}^{1 \times 7} \\ \dot{x}_2^d &\in \mathbb{R}, J_2 \in \mathbb{R}^{1 \times 7} \\ U_n^T \dot{x}_d &= \dot{x}_3^d \in \mathbb{R}^3, U_n^T J = J_3 \in \mathbb{R}^{3 \times 7} \\ U_s^T \dot{x}_d &= \dot{x}_4^d \in \mathbb{R}, U_s^T J = J_4 \in \mathbb{R}^{1 \times 7} \end{aligned} \quad (20)$$

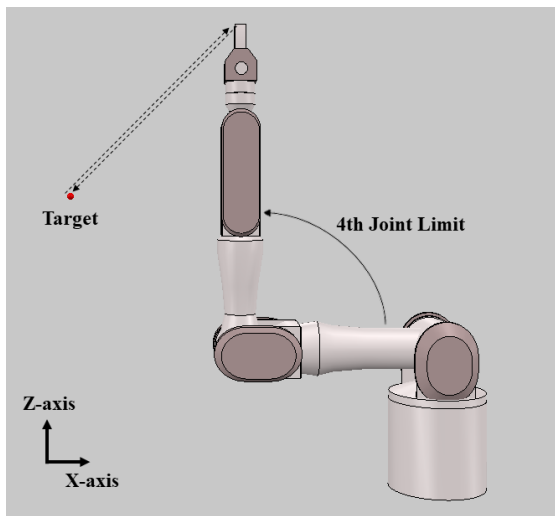
식 (20)의 $U_n \in \mathbb{R}^{4 \times 3}, U_s \in \mathbb{R}^{4 \times 1}$ 은 각각 J 를 특이값 분해 하여 얻은 직교 방향 벡터들 중 특이값이 큰 방향 벡터들과 특이값이 0에 가까운 방향 벡터이다.

3.2.1 관절 한계 회피

Lee et al.^[21]은 관절 위치의 상한치(Upper Limit), 하한치(Lower Limit)에 근처에서 버퍼 영역을 정하였다. 로봇의 관절 위치가 버퍼 영역 안으로 들어가면 버퍼 영역 밖으로 움직이도록 하는 작업을 (21)과 같이 설계하였다. 버퍼 영역 내 관절 위치 값에 따라 활성화 지표를 연속적으로 변화시켜 관절 한계를 회피하였다. 이 실험에서는 [Fig. 5]와 같이 4번째 관절에 대해서 관절 한계 회피 작업을 수행하였다.

$$\begin{aligned} \dot{x}_1^d &= k_4 (\tilde{q}_{up} - q_4) \\ \dot{x}_1^d &= k_4 (\tilde{q}_{lo} - q_4) \end{aligned} \quad (21)$$

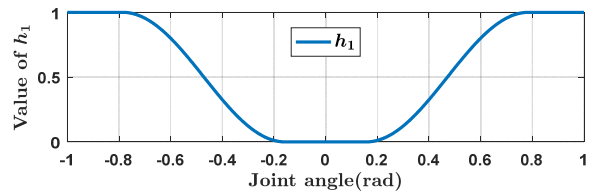
$$\begin{aligned} \tilde{q}_{up} &= \bar{q}_4 - \lambda \\ \tilde{q}_{lo} &= q_4 + \lambda \end{aligned} \quad (22)$$



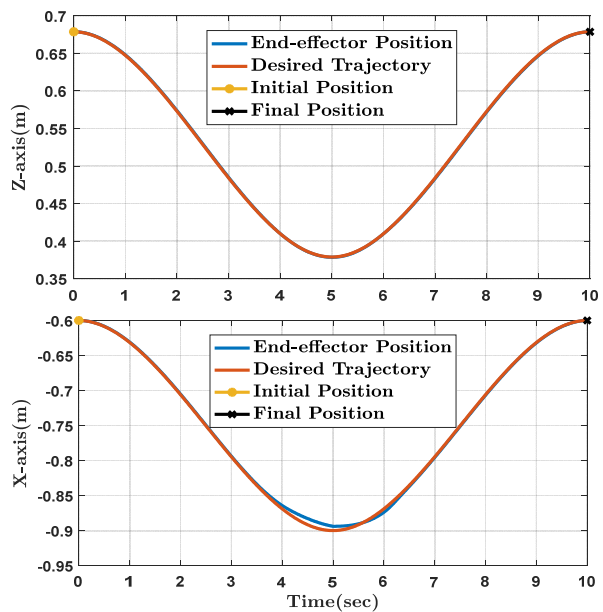
[Fig. 5] Desired trajectory for experiment of joint limit avoidance

\dot{x}_1^d 은 4번째 관절에 대한 관절 상한치, 하한치 회피 작업, k_4 는 이득 상수, $\tilde{q}_{up}, \tilde{q}_{lo}$ 는 4번째 관절 버퍼영역의 상한, 하한 경계 값, q_4 은 현재 4번째 관절 위치 값을 의미한다. (22)의 λ 은 버퍼 영역의 폭, \bar{q}_4, q_4 은 4번째 관절 위치 값의 상한치, 하한치이다. 해당 작업에 대한 활성화 지표는 관절 위치 값에 따라 달라지며 (23), (24)에 나타내었다. 0에서 1사이는 $f(q_4), g(q_4)$ 로 (24)에 나타내었다. 관절의 상한치와 하한치는 각각 $\frac{\pi}{4} \text{ rad}, -\frac{\pi}{4} \text{ rad}$ 이고 버퍼영역의 폭은 $\frac{\pi}{5} \text{ rad}$ 인데 이에 대한 활성화 지표를 [Fig. 6]에 나타내었다.

$$h_1 = \begin{cases} 1.0, & \text{if } q_4 \geq \bar{q}_4 \\ f(q_4), & \text{if } \tilde{q}_{up} < q_4 < \bar{q}_n \\ 0.0, & \text{if } \tilde{q}_{lo} \leq q_4 \leq \tilde{q}_{up} \\ g(q_4), & \text{if } q_4 < q_4 < \tilde{q}_{lo} \\ 1.0, & \text{if } q_4 \leq q_4 \end{cases} \quad (23)$$



[Fig. 6] Value of activation parameter depending on joint angle



[Fig. 7] Desired trajectory and end-effector position over time

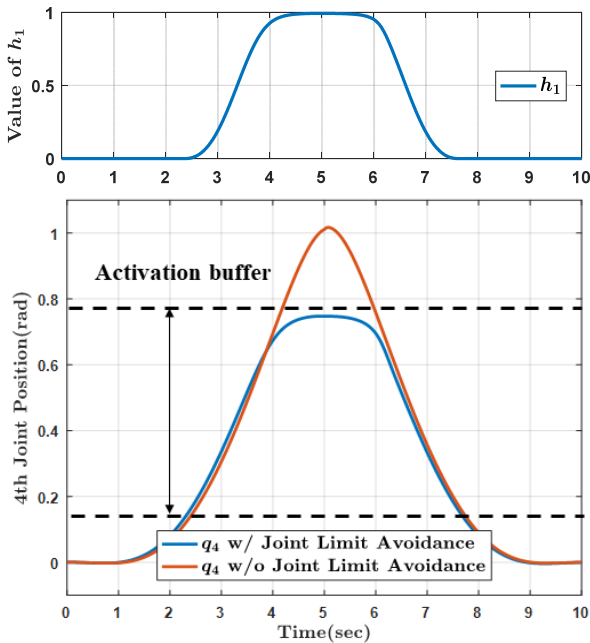
$$\begin{aligned}
 f(q_4) &= 0.5 + 0.5 \sin\left(\frac{\pi}{\lambda}(q_4 - \tilde{q}_{up}) - \frac{\pi}{2}\right) \\
 g(q_4) &= 0.5 + 0.5 \sin\left(\frac{\pi}{\lambda}(q_4 - \tilde{q}_{lo}) - \frac{\pi}{2}\right)
 \end{aligned}
 \tag{24}$$

말단 장치의 목표 경로 x^d 는 [Fig. 5]에 나와 있듯이 시작 위치에서 4번째 관절 한계 영역 내에 있는 빨간색 점을 거쳐 시작 위치를 돌아오는 경로이다. 빨간색 점은 말단 장치에서 X-축으로 -0.3 m, Z-축으로 -0.3 m에 위치한다. 경로는 3차 스플라인(Cubic-spline)함수를 이용하여 각각 5 초씩 총 10 초 동안 움직이도록 생성하였다. 방위는 Y-축 방향의 방위만 제어하였고, 2번 작업의 활성화 지표 h_2 은 항상 0, 3번과 4번 작업의 활성화 지표 h_3, h_4 은 항상 1로 두었다. 활성화 지표는 [Fig. 6]에 따라 적용하였다.

[Fig. 7]은 시간에 대한 목표 경로와 말단 장치의 위치를 나타내었다. 3 초 근처부터 버퍼 영역에 진입하면서 로봇이 목표 경로를 정확히 추종하지 못하였고 7 초 이후 버퍼 영역을 벗어나면서 목표 경로를 다시 추종하기 시작했다. [Fig. 8]에서 관절 값이 버퍼 영역에 진입하면서부터 활성화 지표가 증가하였고, 상한치 아래를 유지하였다. 또한, 동일한 경로에 대해서 관절 한계 회피 작업을 수행한 관절 값과 수행하지 않은 관절 값의 차이를 통해 관절 한계 회피 작업을 검증하였다.

3.2.2 특이점 회피

특이점 회피 실험은 자코비안의 특이값에 따라 작업 전이

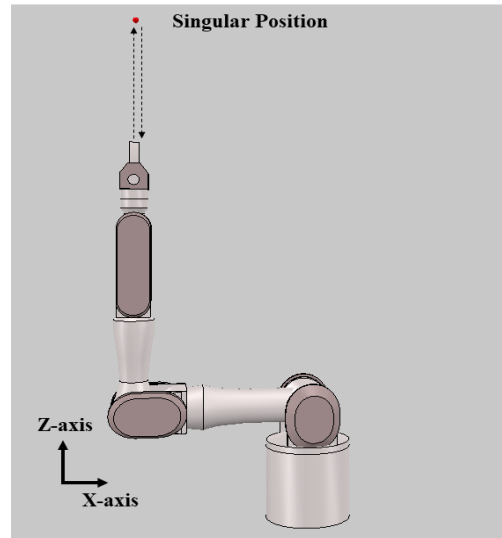


[Fig. 8] Value of activation parameter and 4th joint position with and without joint limit avoidance task over time

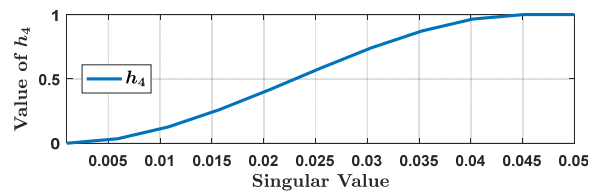
알고리즘을 적용하였다. 목표 경로 x^d 는 [Fig. 9]에 나와 있듯이, 처음 위치에서 빨간색 점을 거쳐 다시 처음 위치로 돌아오는 경로이다. 빨간색 점은 말단 장치에서 Z-축으로 0.25 m에 위치한다. 경로는 3차 스플라인 함수를 이용해 각각 5 초씩, 총 10 초 동안 움직이도록 생성하였다. 방위는 Y-축 방향의 방위만 유지하도록 제어하였고, 1번과 2번 작업의 활성화 지표 h_1, h_2 은 항상 0으로 두었다. 빨간색 점은 로봇의 말단 장치가 도달하지 못하는 점이다. 따라서 빨간색 점 근처는 특이점 영역이 되고 이 영역 주위에서 로봇의 목표 경로 추종 능력과 해의 발산 여부를 통해 특이점 회피 작업을 검증하였다.

작업과 자코비안을 특이값 분해된 방향 벡터들을 이용해 분해하고, 자코비안의 제일 작은 특이값을 기준으로 특이점을 판별하였다. 자코비안의 특이값이 0에 가까워지면 h_4 의 값을 0으로 수렴시켜 4번 작업을 제거하여 특이점을 회피하였다. 3번 작업은 특이값에 영향을 받지 않기 때문에 h_3 의 값을 항상 1로 두어 3번 작업을 수행하였다. 특이값에 따른 각 작업의 활성화 지표는 특이값이 0.001부터 0.05사이에서 결정되는데 이를 [Fig. 10]에 나타내었다.

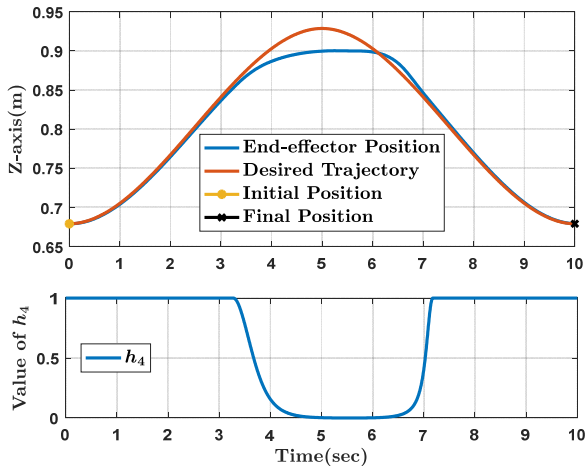
[Fig. 11]에서 말단 장치 위치는 Z-축으로 최대로 움직여 경



[Fig. 9] Desired trajectory for experiment of singularity avoidance



[Fig. 10] Value of activation parameter of task depending on singular value



[Fig. 11] Desired trajectory, end-effector position and value of activation parameter over time

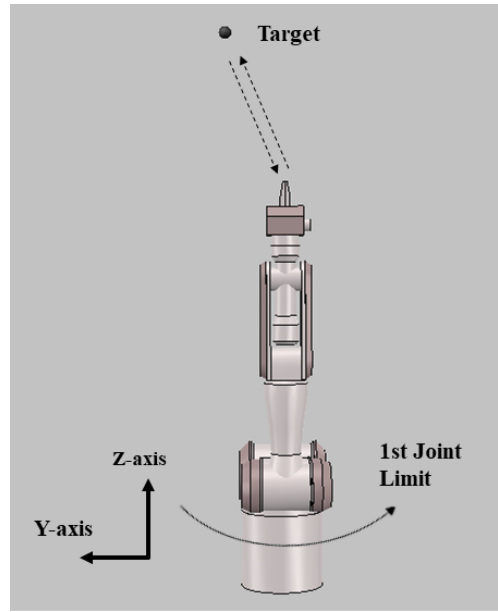
로를 추종하였다. 활성화 지표는 말단 장치가 최대 높이까지 도달하기 전에 0이 되었다가, 돌아오면서 1로 되었다. 이를 통해 특이점 근처에서 특이점 회피 작업을 수행하여 해가 발산하지 않고 로봇이 안정적으로 움직인 것을 검증하였다.

3.2.3 알고리즘 특이점 회피

여러 작업의 우선순위가 존재하는 경우, 본 논문에서 실험한 기구학적 특이점 이외에도 알고리즘 특이점이 발생할 수 있다. 알고리즘 특이점은 우선순위가 높은 작업 자코비안 행의 범위 공간과 우선순위가 낮은 작업 자코비안 행의 범위 공간이 겹치면서 발생한다. 이는 식 (4)에서 $J_2 N_1$ 의 특이값이 0으로 수렴하면서 발생한다. 본 논문에서 적용한 특이점 회피 알고리즘으로 알고리즘 특이점을 회피할 수 있는데 이에 대한 실험 환경과 결과는 다음과 같다.

우선순위가 높은 1번 작업은 1번 관절 한계 회피 작업으로 설정하였고, 2번 작업은 목표 경로 추종 작업으로 설정하였다. 여기서 $J_2 N_1$ 을 특이값 분해하여 특이값이 0으로 수렴하면 특이 방향으로의 작업을 제거하여 알고리즘 특이점을 회피하도록 하였다.

실험 환경은 3.1절과 동일하며, 목표 경로 x^d 는 [Fig. 12]에 나타내었는데 처음 위치에서 검은색 점을 거쳐 다시 처음 위치로 돌아오는 경로이다. 검은색 점은 말단 장치에서 Y-축으로 0.1 m, Z-축으로 0.25 m에 위치한다. 경로는 3차 스플라인 함수를 이용해 각각 5 초씩, 총 10 초 동안 움직이도록 생성하였다. 방위는 Y-축 방향의 방위만 유지하도록 제어하였고, 1번 관절의 상한치와 하한치는 각각 $\frac{\pi}{6} rad$, $-\frac{\pi}{6} rad$ 이고 버퍼영역의 폭은 $\frac{\pi}{6} rad$, k_1 은 0.5로 하였다. 이에 대한 목표 작업과 자코비안을 다음과 같이 정리하였다.



[Fig. 12] Desired trajectory for experiment of algorithmic singularity

$$\dot{x}_1^d \in R, J_1 \in R^{1 \times 7} \tag{25}$$

$$J_{task} = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \in R^{4 \times 7}, J_{task} N_1 = [U_n, U_s] S V^T$$

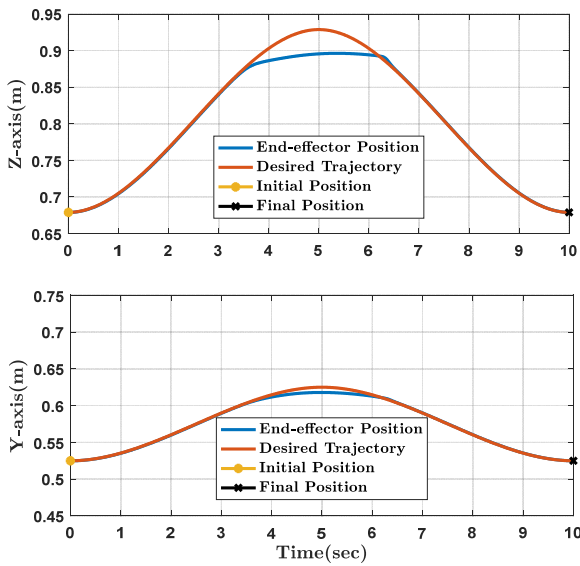
$$\dot{x}_{task}^d = \dot{x}^d + K(x^d - x) \in R^4 \tag{26}$$

$$\dot{x}_2^d = U_n^T \dot{x}_{task}^d \in R^3, J_2 = U_n^T J_{task} N_1 \in R^{3 \times 7}$$

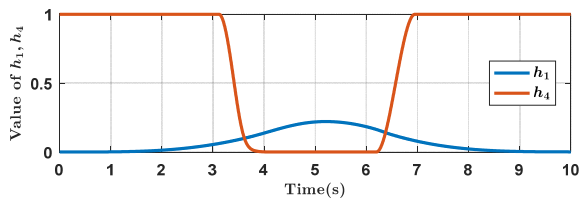
$$\dot{x}_3^d = U_s^T \dot{x}_{task}^d \in R, J_3 = U_s^T J_{task} N_1 \in R^{1 \times 7}$$

식 (26)의 $U_n \in R^{4 \times 3}$, $U_s \in R^{4 \times 1}$ 은 각각 $J_2 N_1$ 를 특이값 분해하여 얻은 직교 방향 벡터들 중 특이값이 큰 방향 벡터들과 특이값이 0에 가까운 방향 벡터이다.

실험 결과는 [Fig. 13-14]에 나타내었다. [Fig. 13]은 시간에 따른 목표 경로와 말단 장치 위치를 나타내었는데, 목표점이 알고리즘 특이점 근처이면서 동시에 1번 관절 한계 버퍼영역에 진입하는 영역이므로 목표 경로를 정확히 추종하지 못하였다. [Fig. 14]은 시간에 따른 활성화 지표를 나타내었는데 목표점에 도달하면서 1번 관절의 버퍼영역에 진입하여 h_1 이 증가하였고, 알고리즘 특이점에 도달하면서 h_4 가 0으로 감소하여 특이점을 회피하였다. 이를 통해 본 논문에서 적용한 특이점 회피 알고리즘으로 자코비안의 범위 공간이 겹치면서 발생하는 알고리즘 특이점을 회피하였다.



[Fig. 13] Desired trajectory and end-effector position over time

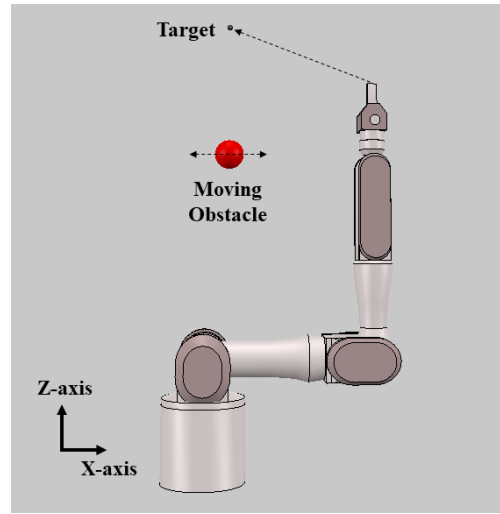


[Fig. 14] Value of activation parameters of tasks for algorithmic singularity

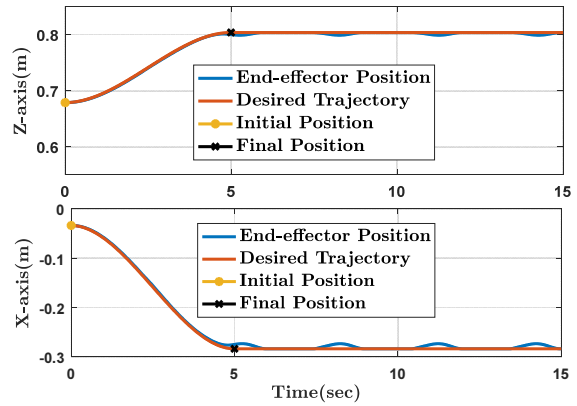
3.2.4 장애물 회피

장애물 회피 실험은 움직이는 장애물에 대하여 작업 전이 알고리즘을 적용하여 검증하였다. 목표 경로 x^d 는 [Fig. 15]에 표현하였는데, 말단 장치를 기준으로 X-축으로 -0.25 m, Z-축으로 0.125 m에 위치한 검은색 점에 도달하는 경로이다. 장애물은 빨간색 구이며 반지름이 0.05 m이고, 말단 장치를 기준으로 X-축으로 -0.25 m, Z-축으로 -0.125 m에 위치하고 있다. 장애물의 X-축 위치는 진폭이 0.03 m이고 주기가 3 초인 사인파 형태로 변화도록 설정하였다. 경로는 3차 스플라인 함수를 이용하여 5 초 동안 이동하도록 생성하였다. 방위는 Y-축 방향의 방위만 유지하도록 제어하였고, 1번 작업의 활성화 지표 h_1 은 항상 0, 3번과 4번 작업의 활성화 지표 h_3, h_4 은 항상 1로 두었다. 장애물 회피 작업의 활성화 지표는 [Fig. 3]과 같이 설정하여 로봇과 장애물 사이의 거리에 따라 증감한다. (11)의 k_p 은 3.0을 입력하였다.

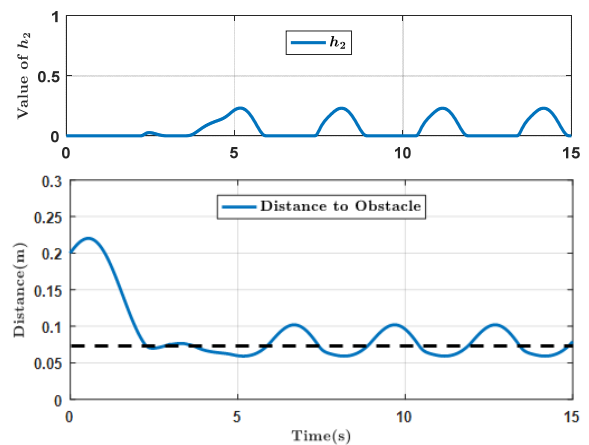
장애물과 가까워지면 활성화 지표가 증가하면서 장애물에서 멀어지는 방향의 작업이 삽입되어 장애물을 회피한다. 이에 대한 그래프를 [Fig. 16]과 [Fig. 17]에 나타내었고, 시간에



[Fig. 15] Desired trajectory for experiment of obstacle avoidance

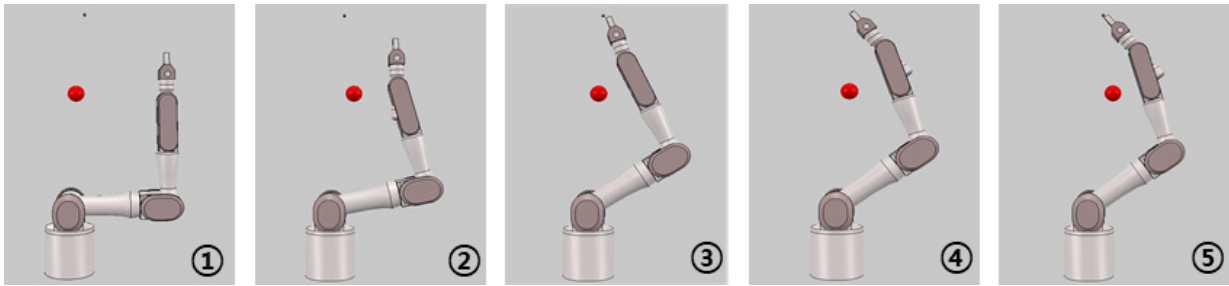


[Fig. 16] Desired trajectory and end-effector position over time



[Fig. 17] Value of activation parameter and distance between robot and obstacle along time

따른 로봇의 사진을 [Fig. 18]에 나타내었다. [Fig. 16]은 시간에 대한 목표 경로와 말단 장치 위치를 나타내었다. 5 초 근처

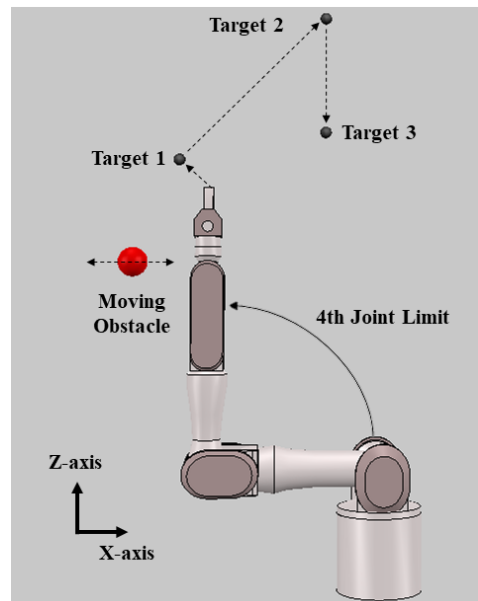


[Fig. 18] Snapshots of simulated manipulator with obstacle avoidance. From left to right, the shots of robot avoiding obstacle were taken at 0 s, 2 s, 4 s, 6 s and 8 s during experiment

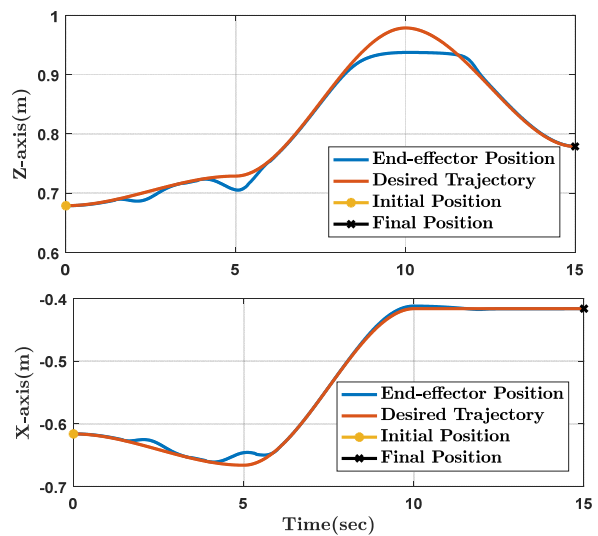
를 시작으로 장애물이 가까워지는 주기인 3 초마다 로봇이 목표 경로를 정확히 추종하지 못하였다. 이는 장애물 회피 작업의 우선순위가 목표 경로 추종 작업보다 높기 때문이다. [Fig. 17]는 시간에 대한 2번 작업의 활성화 지표, 그리고 로봇과 장애물 사이의 거리를 나타내었는데 장애물이 가까워짐에 따라 2번 작업의 활성화 지표는 증가하였고 장애물이 멀어지면서 감소하였다.

3.2.5 통합 프레임워크

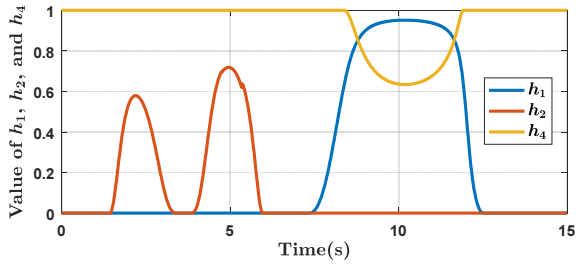
통합 프레임워크는 세 가지 제한 조건을 동시에 고려하면서 회피하는 과정을 통해 검증하였다. 목표 경로 x^d 은 [Fig. 19]과 같이 총 세 가지 목표점을 거치는 경로이다. 먼저 말단 장치를 기준으로 X-축으로 -0.05 m, Z-축으로 0.05 m에 위치한 목표점 1을 거치는데 경로 근처에는 움직이는 장애물이 존재하여 회피하여야 한다. 장애물은 반지름이 0.05 m인 빨간색 구이며, 말단 장치를 기준으로 X-축으로 -0.125 m, Z-축으로 -0.125 m에 위치하고 있다. 장애물의 X-축 위치는 앞의 실험과 동일하게 진폭이 0.03 m이고 주기가 3 초인 사인과 형태로 변화하도록 설정하였다. 다음으로 목표점 1을 기준으로 X-축으로 0.25 m, Z-축으로 0.25 m에 위치한 목표점 2에 도달하는 경로인데 목표점 2 근처에서는 4번째 관절의 한계 영역과 특이점 영역이기 때문에 이를 동시에 회피하여야 한다. 마지막 경로는 목표점 2를 기준으로 Z-축으로 -0.2 m에 위치한 목표점 3에 도달하는 것이다. 경로는 3차 스플라인 함수를 이용하여 각각의 목표점에 도달하도록 5 초씩 총 15 초 동안 이동하도록 생성하였다. 방위는 Y-축 방향의 방위만 제어하도록 하였다. 4 번째 관절의 상한치와 하한치는 각각 $\frac{\pi}{3} \text{ rad}$, $-\frac{\pi}{3} \text{ rad}$ 이고 비회영역의 폭은 $\frac{\pi}{6} \text{ rad}$, k_4 은 0.5로 하였다. 3번 작업의 활성화 지표 h_3 은 항상 1로 두었다. (11)의 k_p 은 3.0을 입력하였고, 장애물 회피 작업과 특이점 회피 작업에 대한 활성화 지표는 [Fig. 3], [Fig. 10]에 따라 적용하였다.



[Fig. 19] Desired trajectory for experiment of unified framework



[Fig. 20] Desired trajectory and end-effector position over time



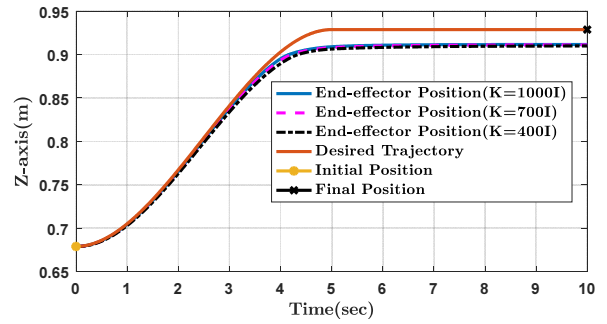
[Fig. 21] Value of activation parameters of tasks for unified framework

실험 결과를 [Fig. 20]과 [Fig. 21]에 정리하였다. [Fig. 20]은 시간에 따른 목표 경로와 말단 장치의 위치를 나타내는데, 3 초, 5 초 근처에서 장애물과 가까워져서 목표 경로를 정확히 추종하지 못하였고, 8 초 근처에서부터 12 초 근처까지 관절 한계 베퍼 영역과 특이점에 도달하여 목표 경로를 정확히 추종하지 못하였다. 나머지 영역에서는 목표 경로를 정확히 추종하였다. [Fig. 21]은 시간에 따른 각 작업의 활성화 지표를 나타내었다. 3 초, 5 초 근처에서 장애물과 가까워져서 h_2 가 증가하였다가 회피하여 다시 감소하였다. 8 초 근처에서부터 12 초 근처까지 h_1 은 증가하였다가 다시 0으로 감소하였고, h_4 은 감소하였다가 다시 1로 증가하였다. 이처럼 통합 프레임워크를 통해 세 가지 제한 조건을 동시에 고려하면서 회피할 수 있었고, 관절 속도 해의 불연속성 없이 작업을 수행할 수 있었다.

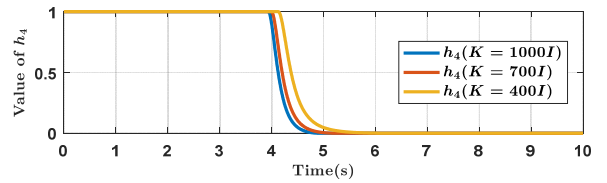
3.2.6 페루프 역기구학의 되먹임 이득 행렬 값에 따른 특이점 회피 알고리즘의 민감도 분석

식 (18)의 페루프 역기구학을 적용하여 로봇을 제어할 때, 되먹임 이득 행렬 K 의 대각 성분 값을 크게 설정하면, 목표 경로를 추종하는 해의 정확성을 높일 수 있다. 하지만 특이점 근처에서는 해가 발산하려는 경향이 더 커진다. 이에 대해 K 의 대각 성분 값에 따른 본 논문에서 적용한 특이점 회피 알고리즘의 민감도에 대해 분석해보았다. 실험 환경은 3.1절에서 설명한 환경과 동일하며, 목표 위치는 [Fig. 9]에서 설정한 위치와 동일하다. 경로는 3차 스플라인 함수를 이용해 5 초 동안 움직이도록 생성하였고, 5 초 이후에는 경로 생성을 정지하였다. 방위는 Y-축 방향의 방위만 유지하도록 제어하였다. 특이값에 따른 활성화 지표는 [Fig. 10]과 동일하게 설정하였다. K 은 $400I$, $700I$, $1000I$ 인 경우에 대해서 실험하였고 이에 대한 결과를 [Fig. 22-24]에 나타내었다.

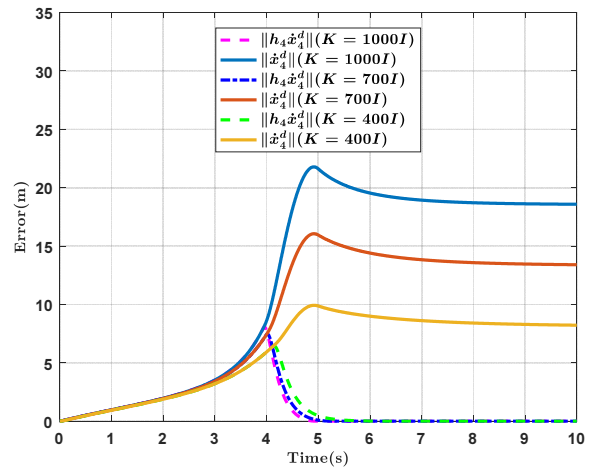
[Fig. 22]은 시간에 따른 목표 경로와 말단 장치 위치를 나타내었는데 K 의 대각 성분 값에 따라 두 경로의 차이가 거의 없었고, 특이점 회피 작업으로 인해 목표 경로를 모두 추종하지 못하였다. [Fig. 23]은 시간에 따른 특이점 회피 작업인 4번 작업의 활성화 지표를 나타내었는데 K 가 $1000I$ 인 경우가 제일



[Fig. 22] Desired trajectory and end-effector position over time depending on magnitude of feedback gain matrix



[Fig. 23] Value of activation parameter over time depending on magnitude of feedback gain matrix



[Fig. 24] Error norm of singularity avoidance task over time depending on magnitude of feedback gain matrix

빠른 속도로 활성화 지표가 0에 수렴하였다. 마지막으로, [Fig. 24]은 시간에 따른 목표 속도향과 위치 오차항을 합한 벡터의 2-노름, 그리고 여기에 4번 작업의 활성화 지표를 곱한 2-노름을 나타내었다. K 가 $1000I$ 인 경우가 K 가 $400I$, $700I$ 인 경우보다 오차가 더 빠르고 많이 증가하지만 활성화 지표가 그만큼 빠르게 0으로 수렴하므로 특이점 근처에서 발산하지 않고 특이점을 회피할 수 있는 것을 확인할 수 있다. 결론적으로, 본 논문에서 적용한 특이점 회피 알고리즘은 되먹임 이득 행렬 크기에 영향을 받지 않고 안정적으로 특이점을 회피하였다.

4. 결 론

본 논문에서는 작업의 삽입 및 제거를 연속적으로 할 수 있는 작업 전이 알고리즘을 이용하여 장애물을 회피하는 알고리즘을 제안하였다. 또한 관절 한계, 특이점, 장애물을 포함한 세 가지 제한 조건을 통합적으로 예측, 회피하는 하나의 프레임워크를 제시하였다. 제시한 프레임워크를 검증하기 위해 7자유도 매니퓰레이터를 이용해 시뮬레이션 환경에서 실험하였다. 실험은 제한 조건을 각각 회피하는 작업과 동시에 고려하여 회피하는 작업을 수행하여 결과를 정리하였다.

본 연구의 후속 연구로는 다음과 같다. 본 연구에서는 수치적 역기구학을 기반으로 하는 통합 프레임워크를 제시하였지만 이를 역동역학을 기반으로 하는 통합 프레임워크로 확장할 것이다. 또한 이를 실제 로봇에 적용하여 구현할 것이다.

References

- [1] R. M. Murray, Z. Li, and S. S. Sastry, "A mathematical introduction to robotic manipulation," *CRC Press*, 1994.
- [2] H. H. An, W. I. Clement, and B. Reed, "Analytic inverse kinematic solution with self-motion constraint for the 7-dof restore robot arm," *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Besacon, France, pp. 1325-1330, 2014.
- [3] Z. Cui, H. Pan, D. Qian, Y. Peng, and Z. Han, "A novel inverse kinematics solution for a 7-dof humanoid manipulator," *2012 IEEE International Conference on Mechatronics and Automation*, Chengdu, China, pp. 2230-2234, 2012.
- [4] Y.-L. Kim and J.-B. Song, "Analytical inverse kinematics algorithm for a 7-dof anthropomorphic robot arm using intuitive elbow direction," *Journal of Korea Robotics Society*, vol. 6, no. 1, pp. 27-33, Jan., 2011.
- [5] J.-H. Lee, J.-Y. Kim, J.-H. Lee, D.-H. Kim, H.-K. Lim and S.-H. Ryu, "Inverse kinematics solution and optimal motion planning for industrial robots with redundancy," *Journal of Korea Robotics Society*, vol. 7, no. 1, pp. 35-44, Jan., 2012.
- [6] C.-G. Kang, "Solution space of inverse differential kinematics," *Journal of Korea Robotics Society*, vol. 10, no. 4, pp. 230-244, Dec., 2015.
- [7] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93-101, Jan., 1986.
- [8] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47-53, Jun., 1969.
- [9] H. Zghal, R. V. Dubey and J. A. Euler, "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy," *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, vol. 2, pp. 1006-1011, 1990.
- [10] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *IEEE/ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163-171, Jul., 1986.
- [11] B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, "Constrained closed loop inverse kinematics," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, pp. 2499-2506, 2010.
- [12] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398-410, Jun., 1997.
- [13] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotic Research*, vol. 4, no. 3, pp. 109-117, Sep., 1985.
- [14] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286-292, Apr., 1995.
- [15] J. Kim, G. Marani, W. K. Chung, and J. Yuh, "Task reconstruction method for real-time singularity avoidance for robotics manipulators," *Advanced Robotics*, vol. 20, no. 4, pp. 453-481, Apr., 2006.
- [16] G. Schreiber and G. Hirzinger, "Singularity consistent inverse kinematics by enhancing the jacobian transpose," *Advances in Robot Kinematics: Analysis and Control*, Lenarčič J., Husty M.L. eds, Springer, Dordrecht, 1998, pp. 475-482.
- [17] F. Chaumette and E. Marchand, "A redundancy-based iterative scheme for avoiding joint limits: application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 719-730, Oct., 2001.
- [18] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-time collision avoidance for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 448-457, Jun., 1995.
- [19] K.-K. Lee and M. Buss, "Obstacle avoidance for redundant robots using Jacobian transpose method," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, pp. 3509-3514, 2007.
- [20] S. R. Buss and J.-S. Kim, "Selectively damped least squared for inverse kinematics," *Journal of Graphics Tools*, vol. 10, pp. 37-49, Jan., 2011.
- [21] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics and Automation*, vol. 28, no. 6, pp. 1260-1277, Dec., 2012.
- [22] S. Kim and J. Park, "Singularity avoidance algorithms for controlling robot manipulator: a comparative study," *Journal of Korea Robotics Society*, vol. 12, no. 1, pp. 42-54, Mar., 2017.
- [23] K. Oonishi, N. Oonishi, and K. Shimoyama, "Producing and the latest development programs of the portable general purpose intelligent arm 'Mitsubishi PA-10'," *Advanced Robotics*, vol. 15, no. 3, pp. 333-337, Apr., 2001.

- [24] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp. 1321-1326, Nov., 2013.



장근우

2016 고려대학교 기계공학부(공학사)
2016~현재 서울대학교 융합과학부 석사,
박사 통합과정

관심분야: Mobile Manipulator, Motion Planning



김상현

2012 서울대학교 기계항공공학부(공학사)
2012~현재 서울대학교 융합과학부 석사, 박사
통합과정

관심분야: Robot Hand, Dual Arm Manipulator



박수한

2017 광운대학교 로봇학부(공학사)
2017~현재 서울대학교 융합과학부 석사, 박사
통합과정

관심분야: Mobile Manipulator, Planning



박재흥

1995 서울대학교 항공우주공학과(공학사)
1997 동 대학원 항공우주공학과(공학석사)
2006 Stanford University Aero/Astro
(공학박사)
2009~현재 서울대학교 융합과학기술대학원
교수

관심분야: Robot-environment Interaction, Multi Contact Control,
Whole Body Control