

작업 전이 알고리즘을 이용한 매니플레이터의 자가 충돌 방지 알고리즘

장근우¹, 김상현¹, 박수한¹, 이상엽¹, 박재홍^{1,2}

¹서울대학교, ²차세대융합기술원

Self-collision avoidance algorithm for robot manipulator using continuous task transition algorithm

Jang Keunwoo¹, Kim Sanghyun¹, Park Suhan¹, Lee Sang Yup¹, Park Jaeheung^{1,2}

¹Seoul National University, ²Advanced Institute of Convergence Technology

e-mail: jkw0701, ggory15, psh117, sangyup8378, park73@snu.ac.kr

요 약

본 논문에서는 매니플레이터의 내부 링크 간 자가 충돌을 방지하는 알고리즘을 제안한다. 알고리즘은 해의 불연속성 없이 작업의 삽입이나 삭제가 가능한 작업 전이 알고리즘을 기반으로 한다. 또한, 알고리즘은 매니플레이터의 내부 링크 간 거리 값에 따라 일정 거리 이하부터 서로 멀어지는 방향의 작업을 삽입하여 충돌을 회피한다. 알고리즘의 검증은 7자유도 매니플레이터를 통해 이루어졌다. 또한, 실험 영상은 <https://youtu.be/FH2wKkQfClc> 을 통해 확인할 수 있다.

1. 서론

매니플레이터가 작업을 수행하기 위해서는 목표 경로를 생성해야 한다. 목표 경로를 생성할 때에는 시작점과 목표점 사이의 경로에서 발생할 수 있는 제한 조건들을 반영해야 한다. 작업을 시작하기 전에 제한 조건을 반영하는 경로 계획(Path planning) 방법은 작업 중에 발생할 수 있는 외란이나 환경 변화에 대처하기 힘들다는 단점이 있다. 이를 위해 작업 중에 제한 조건을 회피하는 연구가 활발히 진행되어 왔다[1][2]. Han and Park[1]은 작업 공간 제어 프레임워크 기반 작업 전이 알고리즘을 제안하고, 이를 이용해 특이점과 관절 한계를 회피하였다. 제 1 저자의 다른 논문[2]에서는 이를 확장하여 장애물까지 고려하여 회피하였다. 본 논문에서는 제 1 저자의 다른 논문[2]에서 제안한 장애물 회피 알고리즘을 여러 개의 링크 쌍의 충돌을 고려해야 하는 자가 충돌 방지 알고리즘으로 확장한다. 이 알고리즘은 링크 간의 거리가 일정 이하가 되면 서로 멀어지는 방향의 작업을 삽입하여 링크 간 충돌을 회피한다.

본 논문의 2장에서는 작업 전이 알고리즘과 자가 충돌 방지 알고리즘을 소개하고, 3장에서는 7자유도 매니플레이터로 알고리즘을 검증한 결과를 정리한다. 4장에서는 본 논문을 결론짓는다.

2. 본론

2.1 작업 전이 알고리즘

작업 전이 알고리즘[1]은 해의 불연속성이 없이

작업을 삽입하거나 삭제할 수 있는 알고리즘이다. 0에서 1사이의 실수 값을 갖는 활성화 지표(Activation parameter)라는 개념을 도입하여, 각 작업에 부여한다. 작업을 삽입하는 경우, 활성화 지표를 0에서 1로 연속적으로 변화시킨다. 작업을 삭제하는 경우에는 활성화 지표를 1에서 0으로 변화시킨다. 선행 연구[1]에서 제안하는 식은 다음과 같다.

$$\Gamma = J_1^T A_1 \tilde{f}_1^i + N_1^T J_2^T A_{21} \tilde{f}_2^i \quad (1)$$

$$N_1^T = I - J_1^T \bar{J}_1^T, \quad \bar{J}_1^T = A_1 J_1 A^{-1}$$

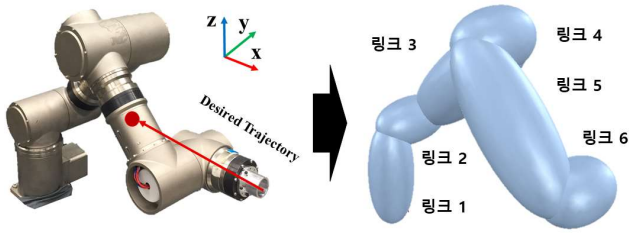
$$\tilde{f}_n^i = h_n \tilde{f}_n + (1 - h_n) J_n A^{-1} \Gamma_{[n]}^* \quad (2)$$

식 (1)과 (2)은 1번 작업의 우선순위가 2번 작업보다 높을 경우, 전체 토크 입력 Γ 을 구한 식이다. h_n 은 n 번 작업의 활성화 지표이고, $\Gamma_{[n]}^*$ 은 n 번 작업을 제외한 토크 입력 값이다.

2.2 자가 충돌 방지 알고리즘

알고리즘은 크게 두 단계로 구성되어 있다. 링크 간의 충돌을 예측하는 단계와 충돌이 예상되면 회피 작업을 제일 높은 우선순위로 삽입하는 단계이다.

링크 간의 충돌을 예측하기 위해 충돌 가능성이 있는 링크 쌍을 추려낸다. [그림 1]은 7자유도 매니플레이터의 각 링크를 타원체로 모델링한 그림을 나타내었다. 충돌 가능성이 있는 링크 쌍은 총 4개로, 링크 1과 5, 링크 2와 5, 링크 1과 6, 링크 2와 6이다. 다음으로, Flexible Collision Library(FCL)[3]을 이용하여 링크 간 거리 값과



[그림 1] 7자유도 매니퓰레이터 모델링과 목표 경로 가장 가까운 두 점을 계산한다.

회피 작업을 삽입하기 위해, 각 링크 쌍의 회피 가속도와 자코비안을 식 (3)과 (4)을 통해 정한다.

$$\ddot{x}_i = k_p \left(1 - \frac{d_i - d_{lo}}{d_{up} - d_{lo}}\right) \frac{x_{a,i} - x_{b,i}}{\|x_{a,i} - x_{b,i}\|} \in R^3 (i=1, \dots, 4) \quad (3)$$

\ddot{x}_i 은 i 번째 링크 쌍에 대한 회피 가속도이고, d_i, d_{lo}, d_{up} 은 링크 간 거리, 거리에 대한 하한치, 상한치이다. $x_{a,i}, x_{b,i}$ 은 링크 쌍의 가장 가까운 두 점을 가리키는 위치 벡터이다.

$$\tilde{f}_1 = \begin{pmatrix} p_{12}^T (\ddot{x}_1 + \ddot{x}_2) \\ p_{34}^T (\ddot{x}_3 + \ddot{x}_4) \end{pmatrix} \in R^2, \quad J_1 = \begin{pmatrix} p_{12}^T J_{12} \\ p_{34}^T J_{34} \end{pmatrix} \in R^{2 \times 7} \quad (4)$$

\ddot{x}_1, \ddot{x}_2 은 링크 1과 2가 링크 5를 밀어내는 회피 가속도이고 \ddot{x}_3, \ddot{x}_4 은 각각 링크 6을 밀어내는 가속도이다. p_{12}, p_{34} 은 회피 가속도 합의 단위 벡터이고, J_{12}, J_{34} 은 각각 회피 가속도의 크기에 대해 $(x_{a,1}, x_{a,2})$ 와 $(x_{a,3}, x_{a,4})$ 의 가중 평균점까지의 자코비안이다.

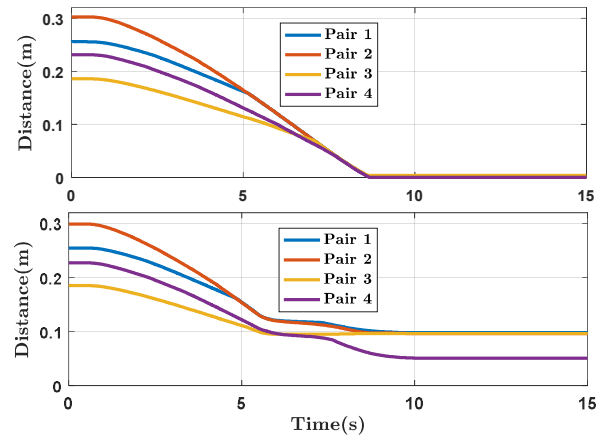
마지막으로, 링크 간 거리가 0.1m이하부터 회피 작업의 활성화 지표를 0에서부터 1까지 선형적으로 증가하도록 정하였다.

3. 실험 결과

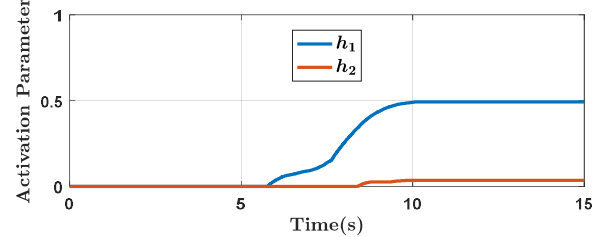
알고리즘은 [그림 1]의 7자유도 매니퓰레이터의 이용하여 검증하였다. 로봇의 제어 입력은 토크 값이고, 제어 주기는 2000Hz 이다.

1번 작업은 자가 충돌 회피 작업이고, 2번 작업은 말단 장치가 목표 경로를 추종하는 작업이다. 식 (3)의 k_p, d_{lo}, d_{up} 은 각각 10, 0.0m, 0.1m로 두었다. 목표 경로 추종 작업은 비례-미분 제어기로 설계하였고, 활성화 지표는 항상 1로 두었다. 목표 위치는 [그림 1]와 같이 말단 장치를 기준으로 X축으로 -0.35m에 위치한다. 경로는 3차 스플라인 함수를 이용하여 10초 동안 이동하도록 생성하였다.

실험결과는 [그림 2]와 [그림 3]에 나타내었다. [그림 2]에서는 회피 작업이 없이 목표 경로만 추종하면 자가 충돌이 일어나지만 회피 작업을 삽입하는 경우에는 자가 충돌이 일어나지 않는 것을 볼 수 있다. [그림 3]은 링크 간 거리가 0.1m이하부터 활성화 지표 2개가 증가하는 것을 확인할 수 있다.



[그림 2] 시간에 따른 링크 간 거리 (위 : 회피 작업 없음, 아래 : 회피 작업 삽입)



[그림 3] 시간에 따른 활성화 지표

4. 결론

본 논문에서는 작업 전이 알고리즘을 이용하여 매니퓰레이터의 자가 충돌을 방지하는 알고리즘을 제안하였다. FCL을 이용하여 로봇 링크 간의 가장 가까운 점과 거리를 계산한 후, 거리에 따라 회피 작업을 삽입하여 충돌을 방지하였다. 본 논문의 후속 연구로는 자가 충돌 방지의 적용 범위를 양팔 모바일 매니퓰레이터로 확장하고 이에 따라 알고리즘도 확장하여 적용하는 것이다.

후기

이 연구는 2018년도 산업통상자원부 및 산업기술 평가관리원(KEIT) 연구비 지원에 의한 연구임 ('10077538')

참고문헌

- [1] H. Han and J. Park, "Robot control near singularity and joint limit using a continuous task transition algorithm", Intl. J. Adv. Robot. Syst., 2013.
- [2] K. Jang, S. Kim, S. Park and J. Park, "A unified framework for overcoming motion constraints of robots using task transition algorithm," J. Korea. Robot. Soc., 2018.
- [3] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," IEEE Intl. Conf. Robot. Autom., 2012.