

Configuration Space 접근법을 이용한 여유 자유도 로봇의 자기 충돌 회피

문재성(포항공과대학교 기계공학과), 정완균* (포항공대 기계공학과), 염영일(포항공대 기계공학과)

Self-Collision Avoidance using Configuration Space Approach for Redundant Manipulators

J. S. Moon(Mech. Eng. POSTECH), W. K. Chung(Mech. Eng. POSTECH), Y. Youm(Mech. Eng. POSTECH)

ABSTRACT

There are two steps to solve the self-collision avoidance problems for redundant manipulators. First, all links are regarded as cylinders, and then the collisions should be checked among all pairs of the links. Between two cylinders, we can get the collision information derived from the concept of configuration space obstacle in real time. Therefore, it is possible to detect the links where collisions are likely in real time by setting the risk radius which is larger than actual radius. Second, the configuration control points (CCP) should be placed at the ends of the detected links. A cost function is the sum of the distances between the CCPs. To maximize the cost function means the links go far away each other without self-collisions.

Key Words : Configuration space approach, Algebraic collision avoidance conditions (대수적 충돌 회피 조건), redundant manipulators (여유 자유도 로봇), Configuration control point (형상 통제점)

1. 서론

로봇이 행하는 작업은 작업공간(Task Space) 상에서 이루어진다. 그리고 로봇의 관절들로 이루어진 공간을 관절공간(Joint Space)이라고 한다. q 는 로봇의 관절값을 나타내는 n 차의 관절공간 상의 벡터이고 p 는 말단 장치(End-effector)의 위치를 나타내는 m 차의 작업공간 상의 벡터이다. 두 벡터 사이에는 $f(q)=p$ 의 관계가 성립된다. 작업공간과 관절공간의 차수가 같으면 자코비안을 이용한 속도 수준에서의 역기구학을 풀어서 단 하나의 q 벡터를 얻을 수 있지만 관절공간의 차수가 더 크면 무수히 많은 해가 존재하게 된다. 이 때 두 공간의 차수의 차이 $r=n-m$ 을 여유 자유도라고 한다. 여유자유도가 있는 로봇은 영운동(null motion)의 추가적인 작업을 이용하여 다양한 구속 조건이 있는 경우에도 역기구학의 해를 구할 수 있다. 본 논문에서는 구속조건으로써 자기 충돌 회피 알고리즘을 제안한다.

자기 충돌(Self-Collision)은 로봇의 링크들이 서로 부딪히는 것이다. 충돌이 발생하면 로봇의 손상은 물론 제어에도 어려움을 겪게 된다. Sezgin *et*

al. [1]은 형상 통제점(Configuration Control Point, CCP)을 설정하여 작업을 수행하는 중에 일어나는 자기 충돌을 회피할 수 있는 알고리즘을 제안하였다. 이 방법은 비용함수(cost function)를 증가시킴으로써 로봇의 링크들을 서로서로 가능한 한 멀리 떨어뜨린다. 비용함수는 링크에 설정된 여러 개의 CCP 사이의 유클리디안 거리의 합으로 이루어진 최대거리기준(Maximum Distance Criterion, MXDC)을 이용한다. 그런데 CCP를 위치시키는 방법에 따라서 충돌 회피 결과가 달라진다. 가장 좋은 위치는 충돌이 일어날 가능성이 있는 링크에 위치시키는 것이다.[1]

이 논문에서는 충돌 회피를 위한 일반적인 방법을 제시하였다. 우선 Configuration Space(C-space) 장애물[5]의 개념에 의해 유도된 대수적 충돌 회피 조건으로 충돌에 관한 정보를 얻는다.[2],[3] 충돌 위험이 있는 링크를 찾으면 그 링크의 쌍에만 MXDC를 적용하고 경사투영법[4]을 이용하여 두 링크가 멀어지도록 하는 역기구학 해를 얻는다. 모의 실험은 MATLAB을 이용하여 프로그래밍하였고 일반적인 n 자유도 로봇의 충돌 회피를 보여준다.

2. 충돌 탐지(Collision Detection)

충돌 탐지를 위하여 Suh and Kim[2]은 로봇의 링크를 모두 실린더로 간주하고 대수적 충돌 회피 조건을 구하였고, Kwon *et al.* [3]은 이를 n 자유도 로봇에 적용하였다.

3 차원 공간상에 있는 두 링크의 위치는 서로 평행인 경우와 그렇지 않은 경우로 구분된다. 전자의 경우 충돌 여부는 다음과 같이 알 수 있다.

두 링크의 반경방향 거리가 각각의 링크반경의 합보다 작거나 같고 두 링크 중심의 길이방향 거리가 각각의 링크길이의 합의 1/2 보다 작거나 같으면 두 링크는 충돌이 일어난 경우이다.

후자의 경우는 각각의 링크의 중심축을 포함하는 서로 평행인 두 평면이 존재하게 된다.[3] 이때 두 평면 사이의 거리가 각각의 링크반경의 합보다 크면 충돌이 없고 작거나 같으면 다음 경우를 살펴 보아야 한다.

이 경우는 두 링크의 위치 관계에 따라 충돌 유무가 달라진다. 기준이 되는 하나의 평면을 정한 후 다른 평면에 있는 링크를 기준 평면으로 수직 투영함으로써 3 차원 문제를 2 차원 평면상에 있는 두 개의 링크 문제로 전환한다. 링크 b 에 기준점(reference vertex, rv)을 설정하면 링크 a 의 외곽선을 따라 기준점을 이동하면서 Fig. 1 에서 점선으로 표시되는 C-space 장애물을 얻을 수 있다. 이제 기준점이 그 안에 있게 되면 충돌이 일어난 경우이다.

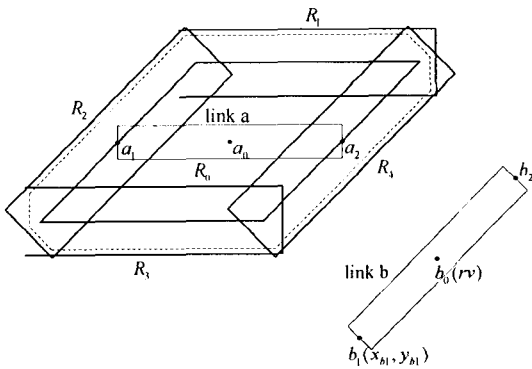


Fig. 1 C-space obstacles and five sub-regions

대수적 충돌 회피 조건식을 얻기 위하여 C-space 장애물을 $R_1 \sim R_5$ 인 4 개의 직사각형 영역과 R_0 인 가운데 영역으로 다섯 등분한다. 두 링크 사이의 각도에 따라서 C-space 장애물의 크기가 변하게 되므로 직사각형 영역은 각도가 0 에서 90 인 경우를 모두 포함하는 최대 크기로 정한다.

직사각형 영역의 중심을 global 좌표의 원점으로

이동시킨 후 한 변의 길이가 2 인 정사각형으로 변환 시키면 정사각형 내부 영역은 $|x| + |y| \leq 2$ 로 표현된다. n 값이 클수록 정사각형에 근접한다. 동일한 변환을 기준점에도 적용하여 얻은 x^* , y^* 를 위의 부등식에 대입하여 식이 만족되면 충돌이 일어난 경우이다.

R_1 과 R_5 영역에서 충돌이 발생하는 조건은 식 (1), (2)와 같다. 링크 a 의 중심 a_0 가 global 좌표의 원점이 되고 a_1 이 x 축 위에 있도록 미리 변환해 주면 조건식이 보다 간단해진다.

$$\{x_{b1}(r_a + r_b)\}^n + \{y_{b1}(x_{a2} + r_b)\}^n \leq 2\{(x_{a2} + r_b)(r_a + r_b)\}^n \quad (1)$$

$$\{x_{b2}(r_a + r_b)\}^n + \{y_{b2}(x_{a2} + r_b)\}^n \leq 2\{(x_{a2} + r_b)(r_a + r_b)\}^n \quad (2)$$

R_2 와 R_4 영역에서 충돌이 발생하는 조건은 식 (7), (8)과 같다. r_a , r_b 는 링크의 반경이고 l_a , l_b 는 링크의 길이이다.

$$x'_{b2} = x_{b0} - 2r_b(y_{b2} - y_{b0})/l_b \quad (3)$$

$$y'_{b2} = y_{b0} + 2r_b(x_{b2} - x_{b0})/l_b \quad (4)$$

$$\tilde{b}_2 = b_0 + (l_b + 2r_a)(b_2 - b_0)/l_b \quad (5)$$

$$\tilde{b}'_2 = b_0 + (r_a + r_b)(b'_2 - b_0)/r_b \quad (6)$$

$$\begin{aligned} & \{(x_{b0} - x_{a1})(\tilde{y}'_{b2} - y_{b0}) + (x_{b0} - \tilde{x}'_{b2})(y_{b0} - y_{a1})\}^n \\ & + \{(x_{b0} - x_{a1})(y_{b0} - \tilde{y}_{b0}) + (\tilde{x}_{b2} - x_{b0})(y_{b0} - y_{a1})\}^n \\ & \leq 2\{(\tilde{x}_{b2} - x_{b0})(\tilde{y}'_{b2} - y_{b0}) - (\tilde{x}'_{b2} - x_{b0})(\tilde{y}_{b2} - y_{b0})\}^n \end{aligned} \quad (7)$$

$$\begin{aligned} & \{(x_{b0} - x_{a2})(\tilde{y}'_{b2} - y_{b0}) + (x_{b0} - \tilde{x}'_{b2})(y_{b0} - y_{a2})\}^n \\ & + \{(x_{b0} - x_{a2})(y_{b0} - \tilde{y}_{b2}) + (\tilde{x}_{b2} - x_{b0})(y_{b0} - y_{a2})\}^n \\ & \leq 2\{(\tilde{x}_{b2} - x_{b0})(\tilde{y}'_{b2} - y_{b0}) - (\tilde{x}'_{b2} - x_{b0})(\tilde{y}_{b2} - y_{b0})\}^n \end{aligned} \quad (8)$$

마지막으로 가운데 영역, R_0 에서는 두 링크의 중심축으로 충돌 여부를 확인하면 된다. 중심축의 양 끝점을 각각 m_1 , m_2 라고 하면 그 위의 임의의 한 점은 λ 의 함수로 다음과 같이 표현된다.

$$e(\lambda) = m_1 + \frac{1+\lambda}{2}(m_2 - m_1), \text{ where } |\lambda| \leq 1. \quad (9)$$

이렇게 표현된 두 링크의 중심축의 교점은 $e_a(\lambda_a) = e_b(\lambda_b)$ 를 만족하는 λ 값을 구해서 찾을 수 있다. 각각의 λ 가 $|\lambda_a| \leq 1$ 그리고 $|\lambda_b| \leq 1$ 을 만족 시키면 충돌이 일어난 경우이다. 이 조건은 직사각형 영역에서와 같이 $|\lambda_a|^n + |\lambda_b|^n \leq 2$ 로 표현할 수 있다. R_0 영역에서 충돌이 발생하는 조건은 식 (10) 과 같다.

$$\begin{aligned} & 2^{n-1}\{x_{b0}(y_{b1} - y_{b2}) + y_{b0}(x_{b2} - x_{b1})\}^n \\ & + 2^{n-1}\{y_{b0}(x_{a2} - x_{a1})\}^n \leq \{(x_{a2} - x_{a1})(y_{b1} - y_{b2})\}^n \end{aligned} \quad (10)$$

3. 충돌 회피(Collision Avoidance)

형상 통제점(Configuration Control Point, CCP)이 로봇의 특정한 지점에 설정이 되면 그 위치는 관절 값(joint variables)의 함수가 된다. 그러므로 여러 CCP 사이의 거리합인 MXDC 또한 관절값의 함수가 되어 이를 이용한 충돌 회피가 가능하다.[1] 이와 같은 방법은 링크 l 개로 이루어진 로봇이 자기 충돌을 회피하도록 하기 위하여 최소한 l 개의 CCP를 요구한다. 즉, 각 링크에 최소한 하나의 통제점이 있어야 한다. 그러나 링크는 질점(particle)이 아닌 크기를 가진 강체이고 또한 어느 링크들 사이에서 충돌이 일어날지 모르기 때문에 하나의 링크에는 여러 개의 CCP가 필요하다. 이러한 CCP의 증가는 계산량의 증가로 이어져 바람직하지가 않다. 그리고 모든 링크에 통제점이 설정되어 그 거리를 멀어지게 하여 충돌을 회피하는 방법은 불필요한 움직임이 포함되어 관절 경로가 지나치게 우회되는 결과를 야기한다. 합리적이면서도 일반적으로 적용될 수 있는 방법은 충돌이 일어날 가능성이 높은 링크에만 CCP가 설정되어 그 두 링크를 서로 멀어지게 하는 비용함수를 얻는 것이다. 그래서 일반적인 n 자유도 로봇의 충돌 회피 알고리즘에는 충돌 탐지가 병행 되어야 한다.

3.1 충돌 회피를 위한 비용함수

앞 장의 충돌 탐지 알고리즘을 적용할 때 실제 링크 반경을 이용하면 두 링크 사이의 충돌을 탐지할 수 있다. 그런데 충돌을 회피하는 시점은 충돌이 탐지되기 이전에 이루어져야 한다. 이를 위하여 각각의 링크에 실제 링크 반경보다 큰 위험 반경(risk radius)을 설정한다. 위험 반경을 이용한 충돌 탐지는 두 링크가 충돌이 일어날 가능성이 높다는 것을 의미한다. 이제 충돌 회피는 이 두 링크를 멀어지게 함으로써 구현할 수 있다. Fig. 2에서 위험 반경이 적용된 실린더는 점선으로 표시되었다.

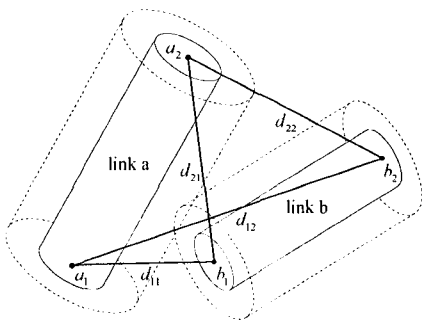


Fig. 2 Selected CCP between two links

위험 반경을 가진 실린더에서 충돌이 탐지되면 각각의 링크의 중심축의 양 끝점에 CCP를 설정한다. 식 (11)과 같은 MXDC를 구하면 MXDC의 증가는 두 링크를 서로 멀어지게 한다. 어느 한 시점에서 위험 반경으로 탐지된 링크의 쌍은 하나 이상일 수 있다. k 쌍의 링크가 탐지되었을 때 비용함수 h 는 식 (12)로 표현된다. 이제 경사투영법을 이용하면 비용함수 h 를 증가시키는 역기구학의 해를 구할 수 있다.

$$MXDC = \sum_{i=1}^k \sum_{j=1}^k d_{ij} \quad (11)$$

$$h = \sum_k MXDC_k \quad (12)$$

3.2 경사투영법

경사투영법(Gradient Projection Method, GPM)[4]은 최대화 혹은 최소화 하기를 원하는 비용함수 h 의 구배(gradient) ∇h 를 자코비안의 영공간으로 투영시키는 방법이다.

$$\dot{q} = J^+(q)\dot{p} + (I - J^+(q)J(q))\kappa\nabla h(q) \quad (13)$$

\dot{p} 을 0으로 두고 영운동(null motion)만을 생각해 보자. 게인(gain) κ 가 양의 값일 때 비용함수가 최대가 되는 방향으로 구배가 생기게 되고 역기구학의 해는 말단 장치의 위치가 고정된 채로 비용함수를 증가시키는 방향으로 관절 값을 생성한다. 결국 비용함수가 최대가 되어 구배가 0이 될 때 영운동이 멈추게 된다. 반대로 게인 κ 가 음의 값을 갖게 되면 비용함수를 최소화 하는 방향으로 영운동이 일어난다. 비용함수를 정하는 방법에 따라 영운동을 이용한 다양한 해의 특성을 얻을 수 있다.

4. 모의 실험

모의 실험은 MATLAB을 이용하여 일반적인 n 자유도 로봇의 충돌 회피를 구현하였다. 결과 분석을 위하여 13 자유도 7 링크 로봇을 구성하였다. Fig. 3은 로봇의 형상과 로봇에 부착된 D-H 좌표계이다. 링크의 길이는 {0} 좌표계에서 시작하여 차례로 [2, 1, 1, 1.5, 1, 1.5, 1.5]^T이다. 실제 링크 반경은 모두 0.1로 하였고 위험 반경은 모두 두 배인 0.2로 하였다. 모든 관절은 회전 관절이며 홈포지션은 [0, -90, 0, -90, 0, 90, 90, 90, -90, 90, 0, 90, 0]^T (deg)이다. 원하는 작업은 말단 장치가 초기위치 [0.5, 2.5, 1.5]^T에서 출발하여 나중위치 [-3.5, -3.5, 1.8]^T까지 이동하는 과정을 5차 다항식으로 계획하였다. 작업 시간은 3초이며 100Hz로 샘플링 하였다.

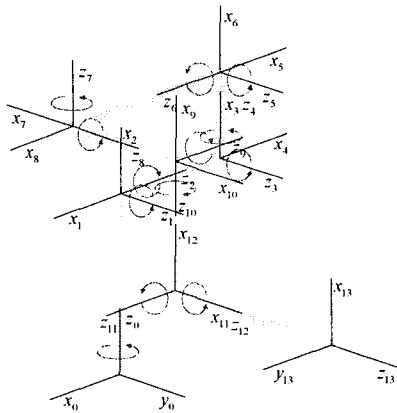


Fig. 3 13-DOF 7-link redundant manipulator

식 (13)의 계인 κ 가 0 이면 영운동 향이 제거되어 주어진 경로를 따라 가는 것 이외의 아무런 추가적인 작업도 할 수가 없다. κ 가 양의 값을 갖게 되면 비로소 비용함수 h 를 증가시키는 해를 얻게 되어 충돌 회피 알고리즘이 적용된다. 그러므로 모의 실험은 $\kappa=0$ 과 $\kappa=1$ 인 두 경우를 비교하였다.

충돌 탐지 정보는 Fig. 4 와 같이 실시간으로 Command Window 에 나타난다. 위험 반경에서 충돌이 탐지 되면 'Warning'이 출력되고 실제 반경에서는 'Collision Detection'이 출력된다. 이어서 탐지된 한 쌍의 링크에 대한 정보가 나타난다. $\kappa=0$ 인 경우 432 회의 'Warning'과 92 회의 'Collision Detection'이 출력된 반면 $\kappa=1$ 에서는 34 회의 'Warning'만 출력되었다. 전자의 경우는 'Warning'이 발생한 시점에서 아무런 대처를 하지 않아서 곧 이어 링크 간의 충돌이 발생하지만 후자의 경우 'Warning'이 발생하면 충돌 회피 알고리즘에 의해 해당 링크들을 서로 멀어지게 하므로 실제 링크 간의 충돌은 전체 작업시간 동안 한번도 일어나지 않는다. Fig. 5 는 두 계인값에 따른 로봇의 움직임이다. $\kappa=1$ 일 때, 충돌을 회피하기 위하여 링크들이 우회하는 것을 볼 수 있다.

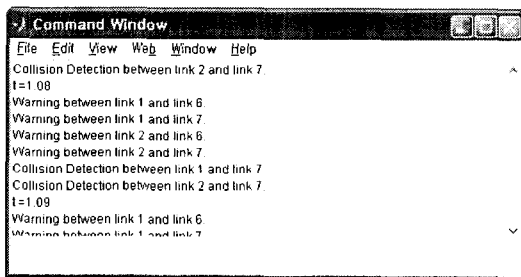


Fig. 4 Messages on Command window

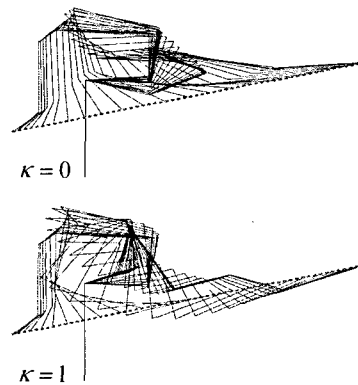


Fig. 5 Simulation results with different gains

5. 결론

자기 충돌 회피 알고리즘이 일반화 되기 위해서는 일반화된 방법으로 CCP 를 설정하여야 한다. 이를 위해 Configuration space 접근법을 이용한 충돌 탐지 알고리즘을 이용한다. 실제 링크 반경보다 큰 위험 반경을 설정하여 충돌을 탐지하면 충돌이 일어날 가능성이 높은 링크들을 찾을 수 있다. 그 링크들의 양 끝점에 CCP 를 설정하여 비용함수를 구성하고 경사투영법을 이용한 역기구학을 풀면 그 링크들을 서로 멀어지게 하는 해를 얻을 수 있다. 모의 실험은 알고리즘이 잘 적용됨을 보여준다.

참고문헌

1. U. Sezgin, L. D. Seneviratne, and S. Earles, "Collision avoidance in multiple-redundant manipulators," *International Journal of Robotics Research*, vol. 16, pp. 714-724, October 1997.
2. S. H. Suh and M. S. Kim, "A nonlinear programming approach to collision-avoidance trajectory planning of multiple robots," *Proc. Of the 1989 Korea Automatic Control Conference(KACC)*, pp. 635-642, 1989
3. S. J. Kwon, W. K. Chung, and Y. Youm, "Self-collision avoidance for n-link redundant manipulators" *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 937 -942, October 1991.
4. A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanism," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-7, no. 12, pp. 868-871, 1977.
5. Lozano-Perez, T., "Spatial Planning: A configuration Space Approach," *IEEE Trans. On Computers*, C-32(2), pp. 108-120, February 1983.