

다이나믹 프로그래밍에 의한 두 대의 로봇트를 위한 최소시간 경로계획

(A Near Minimum-Time Trajectory Planning for Two Robots Using Dynamic Programming Technique)

李志弘*, 吳英碩*

(Ji Hong Lee and Young Seok Oh)

要 約

공간상에 경로가 주어지 있는 두대의 로봇트를 위한 충돌이 없고 시간적으로 최소인 속도 및 조정계획을 다이나믹 프로그래밍 기법을 이용해 구하는 방법을 제안한다. 먼저, 최대 토크, 최대 허용 회전속도등의 각 로봇트의 동력원의 제약조건들을 일차적으로 각 로봇트의 길을 따라 움직인 거리의 함수의 최대속도, 허용가속도 범위등으로 변환시키고, 조정공간에서 충돌영역을 구한다. 그 다음, 속도가 포함되는 확장조정공간을 정의하고 여기에 표시되는 곡선에 대한 여러 제약조건으로 변환시킨다. 원래의 경로 계획 문제를 3차원 조정공간에서 연속인 조정-속도곡선을 구하는 문제로 변환시키고, 앞서 기술한 방법으로 구해진 제약조건들과 충돌회피 및 최소운동시간을 제약조건 및 목적함수로하여 다이나믹 프로그래밍 방법을 적용하여 원하는 해를 구하는 방법을 제안한다. 본 방법에 의한 예제도 기술한다.

Abstract

A numerical trajectory planning method for path-constrained trajectory planning is proposed which ensures collision-free and time-optimal motions for two robotic manipulators with limited actuator torques and velocities. For each robot, physical constraints of the robots such as limited torques or limited rotational velocities of the actuators are converted to the constraints on velocity and acceleration along the path, which is described by a scalar variable denoting the traveled distance from starting point. Collision region is determined on the coordination space according to the kinematic structures and the geometry of the paths of the robots. An Extended Coordination Space is then constructed; an element of the space determines the postures and the velocities of the robots, and all the constraints described before are transformed to some constraints on the behaviour of the coordination-velocity curves in the space. A dynamic programming technique is then provided with on the discretized Extended Coordination Space to derive a collision-free and time-optimal trajectory pair. Numerical example is included.

*正會員, 金烏工科大学 電子制御工學科
(Dept. of Cont. Eng., Kumoh Nat'l Inst. of Tech.)
接受日字: 1992年 5月 9日

I. 서 론

실제로 운영되고 있는 많은 로봇트 시스템은 전체작업

시간을 줄이기 위해 여러대의 로봇트가 작업공간을 공유하며 전체작업을 나누어 가능한 빨리 작업을 수행하도록 하는 형태를 취하고 있다. 이 경우 필수적으로 고려해야 하는 문제가 로봇트와 정지 장애물과의 충돌은 물론이고 로봇트끼리의 충돌에 관한 고려이다. 이러한 필요성으로 여러대의 로봇트를 위한 충돌 없는 경로계획은 로봇틱스 분야에서 오랫동안 다루어져 온 문제이나^[15] 로봇트 몸체의 기구학적 동력학적 구조가 복잡해서 여러가지 가정과 단순화가 문제해결에 도입되어 오곤했던 것이 사실이다. 더구나 최소시간 경로계획에 관해서는 한대의 로봇트의 경우에도 초기상태와 완료상태만이 주어진 일반적인 문제에 대해서도 아직 알려진 해석적인 방법이 없으며 여러가지 가정에 의한 단순화된 경우와^[7,11] 근사적인 경우에 대해서만^[14] 해결 방법들이 알려져 있을 뿐이다. 또한 이러한 어려움 때문에 여러대의 로봇트의 충돌회피운동과 최소 운동시간을 조직적으로 연결시켰던 연구결과도 아주 드물게 보고되고 있다.^[6,13]

본 논문에서 다루고자 하는 여러대의 로봇트를 위한 충돌회피경로 계획은 한대의 로봇트의 경로 계획에 기초를 두게 되므로 먼저 이 경우의 연구결과들에 대해 알아본다. 한대의 로봇트의 동력학을 고려한 최소시간 운동계획에 대해서 [7] 및 [8]에서 공간상의 길이 주어졌는 경우에 완벽한 해를 구하는 방법을 제시한 바 있는데, 이들 방법은 공통적으로 로봇트가 주어진 길을 따라 이동한 거리를 나타내는 변수로 기구학적, 동력학적 방정식을 표시하는 방법을 채택하고 있다. 그리고 나서 로봇트의 동력원의 토크 제약조건을 위한 변수의 위상공간에서의 속도, 가속도 제약조건으로 변환시킨 다음, 최대 가속도와 최소 가속도를 스윙칭하는 위상공간상의 위치들을 찾아 연결하여 최소시간을 갖는 속도계획을 구하도록 되어 있다. 최근들어 이 방법의 많은 문제점이 보완되어, 최대 또는 최소 가속도가 가정되지 않는 경우와 singlar 포인트(singular point) 또는 크리티컬 포인트(critical point)가 존재하는 경우도 해결이 가능하게 되었다.^[9-11]

그러나 두 대 이상의 로봇트의 경우에는 로봇트끼리의 충돌을 피해야만하는 새로운 제약조건이 추가되어 한대의 로봇트에 대해 개발된 여러 방법들을 바로 적용시킬 수가 없다. 그 이유는 충돌 감지와 경로계획이 서로 얽혀 있어 어느것을 먼저 수행시킬 수가 없다는 점이다. 이 문제에도 일반적인 경우 보다는^[1,2] 두대의 로봇트의 경우, 또 움직이는 공간상의 길이 주어졌는 경우에 한정되어 여러가지 해석적인 연구결과가 발표되고 있는 실정이다. 이 문제에서 가장 큰 어려움점은 어떻게 두 대의 로봇트의 충돌여부를 결정하는가하는 것인데, 이 경우에 편리하게 충돌여부를 판단할 수 있는 조정공간의 개념이 [5]에서 발표된 바 있다. 이 조정공간은 두 대의 로봇트가 각

각의 길위를 따라 이동한거리를 2차원 공간의 각각의 축으로 하고 시간이 배제된 그 공간 위에서 충돌영역을 그래픽적인 방법에 의해 구함으로써, 두 대의 로봇트의 경로 계획 문제를 2차원 조정공간에서 연속한 곡선을 찾는 문제로 단순화시킬 수 있도록 했다.

본 논문은 완전히 일반적인 문제의 해결에 대한 노력의 중간 단계로서, [13]에서 다른 조정공간에서의 하나의 충돌영역에 대해 해석적인 해를 제공한 방법을 일반적인 경우에도 확장적용하는 내용을 다루는데, 두 대의 로봇트가 동시작업하는 작업공간에서 각 로봇트의 공간상의 길이 주어졌는 경우에, 충돌을 피하며 전체 운동시간을 최소로하는 각 로봇트의 속도계획 문제에 대해 수치해석적 해결방법을 제시함을 목표로 한다.

구체적으로 먼저, 최대 토크, 최대 허용 회전속도 등의 각 로봇트의 동력원의 제약조건들을 일차적으로 각 로봇트의 길을 따라 움직인 거리의 함수로의 최대속도, 허용가속도 범위등으로 변환시키고 [5]의 조정공간에 속도를 포함시켜 확장조정공간을 정의한 뒤 각 로봇트의 위의 제약조건들을 확장조정공간에서 시간경로를 표시하는 연속곡선들의 존재영역, 기울기등에 대한 여러 제약조건으로 변환시킨다. 그 다음 조정공간에서 충돌영역을 확장된 조정 공간에서의 충돌 공간으로 확장한다. 그리하여, 원래의 경로 계획 문제를 3차원 확장조정공간에서 양끝점이 고정된 어떤 연속인 곡선을 구하는 문제로 변환시키고, 앞에서 구해진 제약조건들과 충돌회피 및 최소운동시간을 제약조건 및 목적함수로하여 [15]에서 한대의 로봇트의 경우에 적용한 바 있는 다이나믹 프로그래밍 방법을 적용하여 원하는 해를 구한다.

이어 2장에서는 문제를 정량적으로 기술하며, 확장된 조정공간과 그 내부의 여러사항에 대한 정의와 예를 기술하고, 간단한 형태로 변환된 문제와 그 해결을 위한 다이나믹 프로그래밍 방법에 대해 기술한다. 3장에서는 예제를 다루고 4장에서는 결론을 맺도록 하겠다.

II. 경로 계획

II장 1절에서는 문제를 설정하고, 2절에서는 확장된 조정공간을 정의하고 그 안에서의 조정-속도 곡선, 공간등을 기술하며 그리고 3절에서는 다이나믹 프로그래밍 방법을 적용할 수 있는 형태로 변형시키는 과정을 설명한다.

1. 문제 설정

기구학식과 동력학식이 다음으로 주어지는 두 대의 로봇트의 경우를 생각해 보자.

$$X^r = K(q^r), r=1, 2 \quad (1a)$$

$$u_i^r = \sum_k D_{ik}^r(q^r) \dot{q}_k^r + \sum_k \sum_m H_{ikm}^r(q^r) \dot{q}_k^r \dot{q}_m^r + G_i^r(q^r) \quad (1b)$$

$$r=1, 2, i=1, \dots, N^r$$

여기서 N^r 은 로봇트 r의 관절 수이고, X^r 및 q^r 은 로봇트 r의 팔끝의 위치와 자세 및 관절 변수(Joint Variable)이다. 그리고, 각관절의 토크와 최대허용 회전속도가 다음으로 제한되어 있다고 가정하다.

$$-u_{i, \max}^r \leq u_i^r \leq u_{i, \max}^r, r=1, 2, i=1, \dots, N^r \quad (2)$$

$$-\dot{q}_{i, \max}^r \leq \dot{q}_i^r \leq \dot{q}_{i, \max}^r, r=1, 2, i=1, \dots, N^r \quad (3)$$

각각의 로봇트가 자신의 길위를 따라 움직이면 각 로봇트의 관절변수가 다음의 식으로 표현될 수 있다.

$$q_i^r = f_i^r(s^r), r=1, 2, i=1, \dots, N^r, \quad (4)$$

$$0 \leq s^r \leq s_{\max}^r$$

또한 두 로봇트의 운동시간을 각각 T_f^1 및 T_f^2 라 할때 전체 운동시간은 다음으로 정의한다.

$$T_f = \max\{T_f^1, T_f^2\} \quad (5)$$

여기서 원래의 문제를 정량적으로 정리한다.

문제 1 :

(1)식으로 표현되는 두대의 로봇트가 (4)로 주어진 경로를 따라서 각각의 초기위치에서 목표위치로 움직일 때, (5)로 주어지는 시간을 최소로하며 충돌이 없으며 (2)-(3)의 제약조건을 만족시키는 두 쌍의 속도계획을 구하여라.

그러나 기구학적, 동력학적 특징의 복잡함으로 인해 이 문제에 일반적인 경우에 해를 제공할 수 있는 어떤 해석적인 방법도 아직 알려지지 않고 있다. 이러한 경로계획 문제가 off-line으로 구해도 무방하다는 점과 근래의 컴퓨터의 성능을 고려하여 수치해석적인 방법으로 주어진 문제의 해를 구하는 방법을 다음의 2-3 두절에서 생각해 보기로 한다. 이 문제의 해는 시간에 따른, 또는 각 로봇트의 경로에 따른 각 관절 각도, 각속도, 각가속도 및 토크를 출력시키므로 이 데이터들을 저장했다가 로봇트에 인가함으로써 로봇트를 동작시킬 수 있으며 사용된 로봇트의 동력학식이 근사적인 경우에도 기준값(nominal torque 또는 feedforward torque)으로 이용하고 보조제어기(예를 들어 PID 제어기)를 이용하여 오차를 극복하는 등의 방법을 이용할 수 있다.

2. 확장 조정 공간

속도와 충돌문제를 동시에 다루기 위해 [5]에서 제안되고 [13]에서 정리된 조정공간(Coordination Space)과 조정곡선 및 충돌영역을 먼저 재정의하고 이어 이들을 확장시켜 확장조정공간을 정의한다.

정의 1 : 조정공간(Coordination Space : CS)

s^1 과 s^2 을 각각 로봇트 1, 2가 각자의 길을 따라 이동한 거리라 하고 s_{\max}^1 과 s_{\max}^2 를 로봇트 1, 2의 길의 총운동거리라 할때, $0 \leq s^1 \leq s_{\max}^1, 0 \leq s^2 \leq s_{\max}^2$ 의 범위에서 (s^1, s^2)으로 표시되는 원소의 집합을 조정공간이라 한다.

정의 2 : 조정곡선(Coordination Curve : CC)

다음의 조건을 만족시키는 CS(Coordination Space)내의 곡선을 조정곡선이라 부른다.

1) (0, 0)를 출발하여 (s_{\max}^1, s_{\max}^2)에 이른다.

2) 이 곡선을 $s^2=C(s^1)$ 이라 하면 $s^2/ds^1 \geq 0$ 이 항상 만족된다.

정의 3 : 충돌 공간(Collision Region : CR)

CS 내부에서 두 로봇트가 충돌 상태에 해당되는 모든 점들의 집합을 충돌공간이라 한다.

이어 확장조정곡선과 조정속도곡선 및 확장된 충돌영역을 정의한다.

정의 4 : 확장조정공간(Extended Coordination Space, ECS)

s^1 과 s^2 을 각각 로봇트 1, 2가 각자의 길을 따라 이동한 거리라 하고 s_{\max}^1 과 s_{\max}^2 를 로봇트 1, 2의 길의 총운동거리라 하고 $v^1(s^1)$ 을 로봇트 1의 운동 속도라 할때, $0 \leq s^1 \leq s_{\max}^1, 0 \leq s^2 \leq s_{\max}^2, 0 \leq v^1 \leq v_{\max}^1$ 의 범위에서 (s^1, s^2, v^1)으로 표시되는 원소의 집합을 확장조정공간이라 한다.

정의 5 : 조정-속도 곡선(Coordination-Velocity Curve, CVC)

다음의 조건을 만족시키는 ECS내의 곡선을 조정-속도곡선이라 부른다.

1) (0, 0, $v^1(0)$)를 출발하여 ($s_{\max}^1, s_{\max}^2, v^1(s_{\max}^1)$)에 이른다.

2) CVC를 $s^1 \times s^2$ 평면(Coordination Space)에 투영(Projection)시켜 얻은 곡선을 $s^2=C(s^1)$ 이라 하면 $ds^2/ds^1 \geq 0$ 이 항상 만족된다.

이 조정곡선을 수식으로 표시하면 $v^1=V(s^1, s^2)$ 로 되고 이 3차원 곡선을 [5]에서의 조정공간(Coordination Space)로 투영시킨 2차원 곡선이 [5]에서 언급하는 조정곡선(Coordination Curve)이 된다.

정의 6 : 확장된 충돌 공간(Extended collision Region : ECR)

ECR 내부에서 두 로봇트가 충돌 상태에 해당되는 모든 점들의 집합을 확장된 충돌공간이라 한다.

여기서 주목할 것은 ECR은 v^1 에 관계가 없다는 것이며, 즉 s^1 과 s^2 에 의해서만 결정된다는 것이며, [5]에서 정의된 내용들과 비교하여 보면 CVC의 밀면이 [5]의 조정공간(Coordination Space)에 해당하고 ECR이 이면(조정공간)과 만나는 폐곡선이 [5]의 충돌영역(CR)에 해당함을 알 수 있다.

그림 1에 2차원에서 동작하는 두대의 로봇의 간단화된 형상을 표시하였으며, 이경우에 [5]에서 제시된 방법에 의해 구한 ECR을 $s^1 \times s^2$ 공간에 투영시킨 모양(조정공간에서의 CR)을 그림 2에 표시하였다. [5]에서는 로봇 한대를 어떤 위치에 고정시키고 나머지 한대를 가상으로 처음부터 끝까지 움직여 보면서 충돌영역을 찾고 다시 원래의 로봇의 위치를 다른 위치로 변화시켜 반복하고 하는 방법에 의해 그래픽적 방법으로 충돌영역을 구하였다. 이상 정의된 항목들을 이용하여 앞절에서 기술한 문제 1을 보다 간단한 문제로 변형하기로 한다. 즉, 가시화하기 힘든 두대의 로봇의 경로계획 문제를 3차원 확장조정공간에서 하나의 연속곡선을 찾는 문제로 변환시킨다. 이때 연속곡선은 두대의 로봇 각각이 각각의 길을 따라 이동할때의 속도와 관계되고 이 곡선의 기울기, 가속도 등은 각 로봇의 동력학적 제한조건(모터의 최대 노르크, 최대 허용 회전속도)과 관계가 있게 된다.

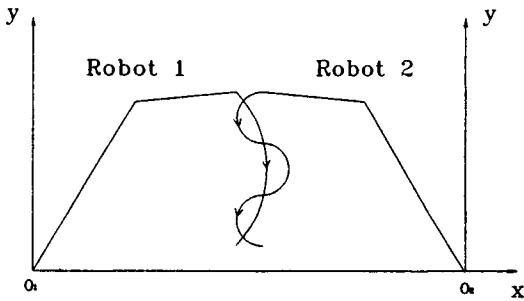


그림 1. 2차원에서의 간단한 로봇 시스템
Fig. 1. A simplified robot system in 2-D.

문제의 변환을 위해 각동력원의 제약조건을 확장공간에서 해 곡선의 제약조건으로 변환시키는 과정을 설명한다. 각각의 로봇에 [10][11][13]의 방법을 적용하면 모터의 제약조건 (2)-(3)은 다음의 식으로 변형된다.

$$a_{min}^r(s^r, v^r) \leq a^r(s^r, v^r) \leq a_{max}^r(s^r, v^r), \quad r=1,2 \quad (6)$$

$$0 \leq v^r(s^r) \leq v_{max}^r(s^r), \quad r=1,2 \quad (7)$$

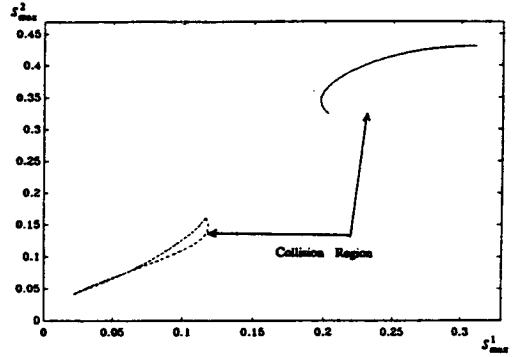


그림 2. 그림 1의 경우의 충돌영역
Fig. 2. Collision region corresponding to Fig. 1.

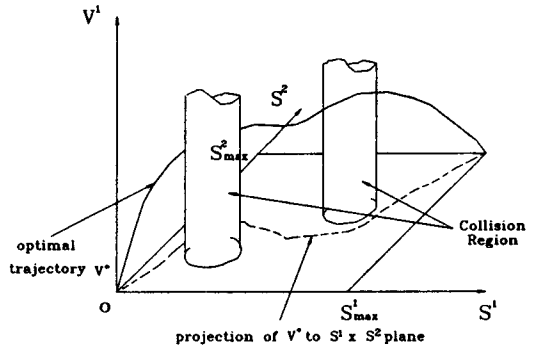


그림 3. 3차원 확장공간에서의 조정-속도 곡선
Fig. 3. Finding a curve in 3-D space.

여기서 s 는 (4)에서와 같이 각 로봇이 자신의 길위를 따라 움직인 거리이고, v 와 a 는 s 와 v 의 시간미분으로 얻어진 변수인 속도와 가속도이다. 즉, 각 로봇의 동력원의 제약조건이 $s \times v$ 공간에서의 움직임의 제약조건(기울기의 최대, 최소 또는 허용영역)으로 변환 되었다. 이제 로봇의 동력원의 제약조건이 어떻게 ECS의 CVC의 제약조건으로 바뀌게 되는지 살펴보자. 주어진 CVC의 $s^1 \times s^2$ 으로의 투영된 2차원 평면위의 곡선을(그림 3 참조) 다음식으로 표시한다.

$$s^2 = C(s^1) \quad (8a)$$

그러면

$$C_s \equiv \frac{ds^2}{ds^1} = \frac{ds^2/dt}{ds^1/dt} = \frac{v^2}{v^1} \quad (8b)$$

과 이를 이용하여

$$C_{ss} \equiv \frac{d}{ds^1} \left[\frac{v^2}{v^1} \right] = \frac{dv^2/ds^1 v^1 - v^2 (dv^1/ds^1)}{(v^1)^2}$$

$$= \frac{(dv^2/ds^1) (ds^1/dt) - v^2 (dv^1/dt) (dt/ds^1)}{(v^1)^2}$$

$$= \frac{a^2 - a^1 C_s}{(v^1)^2} \quad (9)$$

가 유도되는데 여기서,

$$a^r = \frac{dv^r}{dt} \quad (10)$$

로 표시하였다. C_s 는 (8a)의 기울기를 나타내고 C_{ss} 는 그 기울기의 변화율을 나타내는데 편의상 C_{ss} 를 변화율이라고 부르기로 한다. 여기서 (6)과 (9)를 합성하면 조정곡선에 대한 변화율의 제약조건이 유도된다.

$$C_{ss,min} \equiv \frac{a_{min}^2 - a_{max}^1 C_s}{(v^1)^2} \leq C_{ss} \leq$$

$$\frac{a_{max}^2 - a_{mic}^1 C_s}{(v^1)^2} \equiv C_{ss,max} \quad (11)$$

그리고 K 개의 충돌영역이 $s^1 \times s^2$ 평면과 만나는 영역의 내부(그림 3에서 두개의 기둥이 $s^1 \times s^2$ 평면과 만나는 폐곡선의 내부)를 다음으로 표시하면

$$CR_k = \{ (s^1, s^2) \mid \Gamma_k(s^1, s^2) \leq 0, k=1, 2, \dots, K \} \quad (12)$$

가 되어 문제 1은 보다 간단한 형태로 변환될 수 있다.

문제 2 :

문제 1에서 주어진 과제에 대해 ECS에서 (12)의 충돌 공간을 지나지 않으며 (6), (7)과 (11)의 조건을 만족시키며 (0, 0, $v^1(0)$)와 ($s_{max}^1, s_{max}^2, v^1(s_{max}^1)$)의 경계조건을 만족시키며 곡선상의 매 (s^1, s^2)에서 최대의 속도를 갖는 연속곡선, $v^1 = v^*(s^1, s^2)$ 을 구하라.

이러한 상황을 그림 3에 나타내었다. 즉, 원래의 두대의 로봇트의 충돌회피 최소시간 경로계획 문제(II-1의 문제 1)이 그림 3에서와 같이 주어진 두개의 시점(0, 0)에서 종점(s_{max}^1, s_{max}^2) 사이를 연결하는 연속 곡선을 찾되 기둥으로 표시된 충돌영역을 지나지 않고 곡선을 따른 기울기와 변화율이 제한된 상태에서 가능한 한 높이(높다는 것은 빠른 속도를 의미하고 이는 최소 시간을 의미^{[8])})를 매 위치에서 최고로 가지도록 하는 문제로 변형되었다.

3. 다이내믹 프로그래밍 방법에 의한 접근방법

II-2절의 문제 2에 대해 해석적인 방법에 의해 해(연

속곡선)를 구하는 방법은 아직 알려져있지 않다. 그러나 해를 구하더라도 메모리 문제 때문에 곡선 전부를 기억시켰다가 로봇에게 명령을 줄 수는 없는 문제이므로 여차피 디스크리타이즈시켜 저장할 수밖에 없다. 그러므로 여기서는 문제 2를 근사해를 구하는 문제로 변환시킨다. 이렇게 구해진 해는 저장되어 있다가 로봇트를 움직일 때 표준값으로 사용될 수 있다. 또한 로봇트의 동력학식이 정확히 주어지지 않은 경우라도 보조제어기(예를 들어 PID)의 보상값(Feedforward command)으로 사용될 수 있다. 문제 2의 해를 수치해석적 방법으로 구하기 위해 먼저 ECS를 그림 4처럼 디스크리타이즈시킨다. 즉, 전체 공간을 평행한 여러개의 평면으로 자르고 각 평면도 각 평면에서 일정한 간격으로 디스크리타이즈시킨다. 그러면 문제 2의 연속곡선을 찾는 문제는 각 평면에서 하나씩의 점을 택하고 그 점들사이를 직선으로 근사화한 작은 직선들의 연결을 구하는 문제로 근사화 된다. 이러한 근사 문제로 변환시키기 위해 이웃한 평면위에서 각각 한점씩이 주어졌을 때 계산되는 운동시간, 기울기 및 변화율등을 나타내는 식을 구해보자. 여기서 사용되고 있는 기호를 정리하면, k 번째 평면의 (i, j)위치의 실제 좌표는 ($s_i^1[k], s_j^2[k], v_i^1[k]$)로 한다. (i, j, k)는 그림 4에서 보듯이 k 번째 평면에서의 위치를 결정짓는 index로써 즉 열(row)에 해당하는 i 에 의해 s^1 과 s^2 가 결정되고 행(column)에 해당하는 j 에 의해 v^1 이 결정된다. 그러면 이 디스크리타이즈된 ECS에서 k 번째 평면의 (i, j)좌표와 $k+1$ 번째 평면의 (l, m)좌표와의 이동시간은 두점사이를 평균시간으로 이동한다는 단순화에 의해

$$\Delta t_{(i,j),(l,m)}[k] = \frac{2(s_l^1[k+1] - s_i^1[k])}{v_j^1[k] + v_m^1[k+1]} \quad (13)$$

로 근사화되고 로봇트 r 의 가속도는 두 점사이의 속도 변화가 (13)식의 시간동안에 일어난다는 단순화에 의해 다음으로 근사화된다.

$$a_{(i,j),(l,m)}^r[k] = \frac{(v_m^r[k+1])^2 - (v_j^r[k])^2}{2(s_l^r[k+1] - s_i^r[k])} \quad (14)$$

또한 로봇트 2의 속도는 (8b)의 ds^2/ds^1 을 직선으로 근사화 시키고 $C_s = v^2/v^1$ 이라는 (8b)의 결과를 이용하면 다음식으로 근사화되고

$$C_{s_{(i,j),(l,m)}}[k] = \frac{s_j^2[k+1] - s_i^2[k]}{s_l^1[k+1] - s_i^1[k]} \quad (15a)$$

$$v_j^1[k] = v_i^1[k] C_{s_{(i,j),(l,m)}}[k]$$

$$= v_i^1[k] \frac{s_j^2[k+1] - s_i^2[k]}{s_l^1[k+1] - s_i^1[k]} \quad (15b)$$

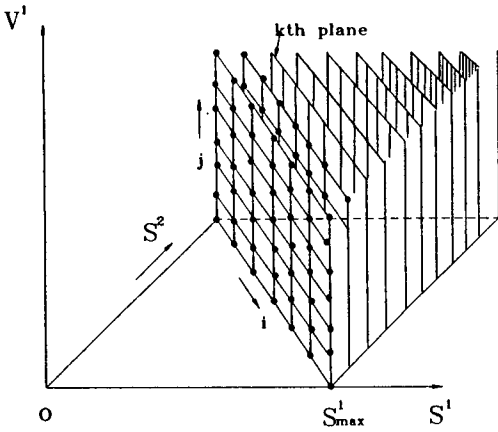


그림 4. 격자화된 확장조정공간
Fig. 4. Discretized $s^1 \times s^2 \times v^1$ space.

$k-1$ 번째 평면의 (p, q) 의 좌표와 k 번째 평면의 (i, j) 의 좌표와 $k+1$ 번째의 (l, m) 의 좌표와의 조정곡선의 변화율은 이어진 두 개의 직선 각각의 기울기와 전체이동거리에 의해 다음식으로 근사화한다.

$$C_{ss,p,q,l,m}[k] = \frac{2 \left[\frac{s_i^2[k+1] - s_i^2[k]}{s_i^1[k+1] - s_i^1[k]} - \frac{s_l^2[k] - s_l^2[k-1]}{s_l^1[k] - s_l^1[k-1]} \right]}{\|s_i^1[k+1] - s_l^1[k-1]\|} \quad (16)$$

이상으로 (6)-(9)로 주어진 연속곡선에 관한 여러 항목들을 디스크리타이즈시켜 다이나믹 프로그래밍에 의해 해를 구할 수 있는 형태로 근사화가 완료되었다. 그런데 이웃한 두 평면 또는 이웃한 세 평면위의 모든 점들이 서로 연결 가능한 것은 아니다. 왜냐하면 문제 2에서 언급한 바와 같이 한 평면 위의 어떤 점은 충돌 영역안에 있을 수도 있고, 이웃한 두 평면의 한점씩을 연결한 직선의 기울기는 허용한도를 넘을 수도 있으며, 이웃한 세 평면의 한점씩을 연결한 직선의 접합은 변화율의 한도를 넘을 수도 있기 때문이다. 여기서 해곡선의 일부로 소속될 수 있는 점들의 연결을 정의해 보자.

정의 7 : 한 평면 위에서의 어드미서빌리티

(17)을 만족하는 k 번째 평면의 (i, j) 좌표 $(s_i^1[k], s_j^2[k], v_i^1[k])$ 는 어드미서블(admissible)하다고 한다.

$$(s_i^1[k], s_j^2[k]) \in CR_p, p=1,2,\dots,P \quad (17)$$

여기서 CR_p 는 p 번째 충돌영역의 내부에 해당하는 CS상의 점들의 집합이다. 결국 (17)은 k 번째 평면의 (i, j) 위치가 충돌영역 밖에 있어야한다는 조건을 의미한다.

정의 8 : 이웃한 두평면 사이에서의 어드미서빌리티

(18)-(19)를 만족하는 k 번째 평면의 (i, j) 좌표 $(s_i^1[k], s_j^2[k], v_i^1[k])$ 와 $k+1$ 번째 평면의 (l, m) 좌표 $(s_l^1[k+1], s_m^2[k+1], v_l^1[k+1])$ 는 어드미서블(admissible)한 쌍이라 한다.

$$0 \leq v_j^1[k] \leq \min\{v_{max}^1[k], C_{s_i^1, v_l^1, m}^{-1}[k] v_{max}^2[k]\} \quad (18)$$

$$\begin{aligned} a_{min}^1(s_i^1[k], v_i^1[k]) &\leq a_{max}^1 a_{i,j,l,m}^1[k] \\ &\leq a_{max}^1(s_i^1[k], v_i^1[k]) \end{aligned} \quad (19)$$

(18)식은 로봇트 1, 2가 모두 허용속도 범위에서 속도를 가져야 한다는 조건을 의미하며 (19)식은 로봇트 1의 가속도가 허용범위내에 있어야 함을 의미한다.

정의 9 : 이웃한 세평면 사이에서의 어드미서빌리티

(20)식을 만족하는 k 번째 평면의 (i, j) 좌표 $(s_i^1[k], s_j^2[k], v_i^1[k])$ 와 $k+1$ 번째 평면의 (l, m) 좌표 $(s_l^1[k], s_m^2[k], v_l^1[k])$ 와 $k-1$ 번째 평면의 (p, q) 좌표 $(s_p^1[k-1], s_q^2[k-1], v_p^1[k-1])$ 는 어드미서블(admissible)한 트리플렛(triplet)라 한다.

$$\begin{aligned} C_{ss,min}(s^1, s^2, v^1, v^2) &\leq C_{ss,p,q,l,m}[k] \\ &\leq C_{ss,max}(s^1, s^2, v^1, v^2) \end{aligned} \quad (20)$$

이는 이웃한 세평면에서 뽑은 세점이 변화율의 제약조건을 만족시켜야함을 의미한다.

여기에서 문제 2의 근사해를 구하는 문제를 정리한다.

문제 3 :

전체 공간을 K 개의 평면으로 분할하고, 각 평면을 $I \times J$ 개의 점으로 디스크리타이즈시켰을 때, 첫번째 평면의 시점과 마지막 평면의 종점을 각 평면에서 정리 7-9에 의한 어드미서블한 하나씩의 점들을 연결하여 만든 곡선 중에서 최소 시간을 주는, 즉 $\sum_{k=0}^{K-1} \Delta t[k]$ 를 최소화하는 곡선을 구하라.

이 문제는 해탐색에 관한 문제로 다이나믹 프로그래밍을 적용시킬 수 있는 전형적인 형태이다. [12]에서 소개된 방법에 의해 $(s_i^1[k], s_j^2[k], v_i^1[k])$ 를 $x[k]$ 로 간단히 표시하고 $\phi(x[k], x[k+1])$ 을 k 번째 평면의 $x[k]$ 와 k 번째 평면의 $x[k+1]$ 사이의 운동시간이라하고 $J^*(x[k])$ 를 $x[k]$ 에서 최종 위치까지의 운동시간이라하면 다음의 최적정리를 얻는다.

$$J^*(x[k]) = \min_{x[k], \dot{x}[k]} \{ \phi(x[k], x[k+1]) + J^*(x[k+1]) \} \quad (21)$$

단, $D[k]$ 는 위에서 기술한바 k 번째 평면의 어드미서블한 좌표들의 집합이다. (21)식을 최종 위치(s_{max}^1, s_{max}^2 , $v^1(s_{max}^1)$)에서 초기위치($0, 0, v^1(0)$)로 마지막 평면에서 첫 평면으로 거꾸로 각평면에 대해 적용하며 각 평면을 탐색하여 첫평면에 이르게 되면 최적해가 구해지게 된다. 예제는 다음장에서 다루겠다.

III. 적용 예

그림 1에 보인 간단한 로봇 시스템과 [13]의 로봇2 동력학식과 기구학식으로 다음의 제약조건을 가한 경우를 생각한다.

$$-25Nm \leq u_1^* \leq 25Nm \quad (22)$$

$$-7Nm \leq u_2^* \leq 7Nm, \quad (23)$$

$$-2rad/sec \leq \dot{q}_1^* \leq 2rad/sec \quad (24a)$$

$$-2.5rad/sec \leq \dot{q}_2^* \leq 2.5rad/sec \quad (24b)$$

로봇 1의 총이동거리는 0.33m, 로봇 2 0.47m이며 반원들을 조합하여 구성하였는데 로봇 1은 중심(0.4, 0.2)반지름 0.15m, 초기위치(0.4, 0.35) 마지막 위치(0.4, 0.05)이고 로봇 2의 경우는 반지름이 각각 0.05m, 중심이 (0.45, 0.3)(0.45, 0.2)(0.45, 0.1)인 세 개의 반원을 조합하여 만들었다. 이경우에 [11]의 방법에 의해 두 로봇의 최대허용 속도가 그림 5 및 6으로 각각 구해졌

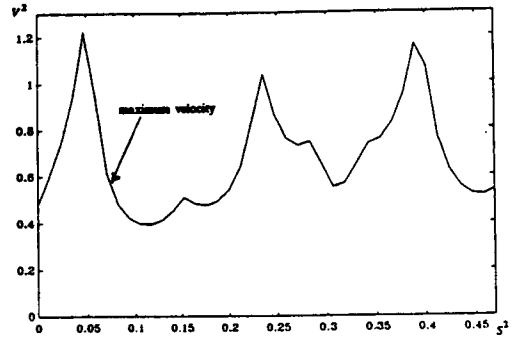


그림 6. 그림 1의 경우의 로봇 2의 최대허용 속도곡선
Fig. 6. Maximum velocity of the robot 2 for the path of Fig. 1.

다. 방법 자체의 제안에 의미를 두어 문제 3의 경우에 비교적 간단히 근사화시킨 경우를 시뮬레이션해 보았다. 평면의 갯수는 100개 각평면은 가로 100, 세로 100개의 격자(Grid)로 디스크리타이즈시켜 다이나믹 프로그래밍 방법을 적용시켰다. 여기에 II-3의 방법을 적용하여 얻은 3차원 곡선을 2차원 조정공간에 투영시켜 그림 7의 조정곡선을 얻었으며 그때의 두 로봇의 속도는 그림 8과 9로 구해졌으며 총 이동시간은 0.664 초로 계산되었다. 물론 격자의 간격을 더 세밀하게 함으로써 계산결과의 정밀도를 높일 수 있다는 점을 밝혀둔다. 그런데 그림 8과 9에서 보면 디스크리타이즈에 의한 근사화 등의 원인에 의해 각 로봇의 속도 변화가 심한 영역이 있음을 관찰할 수가 있다. 이러한 급작스런 변화는 바람직하지 않으므로 다음의 변경된 performance index를 생각한다.

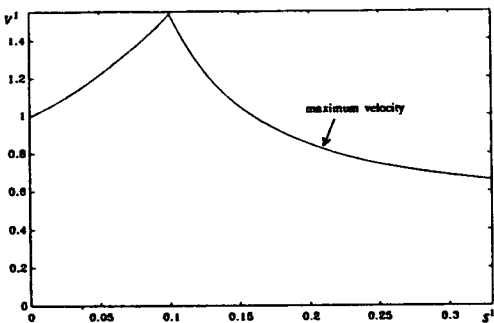


그림 5. 그림 1의 경우의 로봇 1의 최대허용 속도곡선
Fig. 5. Maximum velocity of the robot 1 for the path of Fig. 1.

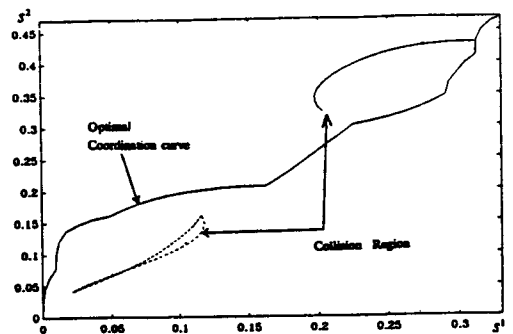


그림 7. 그림 1의 경우의 최적 조정곡선
Fig. 7. Optimal coordination curve.

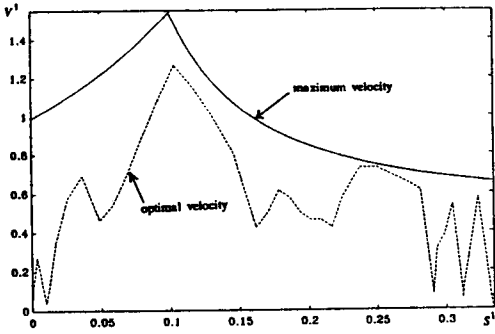


그림 8. 그림 1의 경우의 로봇 1의 최적 속도곡선
 Fig. 8. Optimal velocity of the robot 1 for the path of Fig. 1.

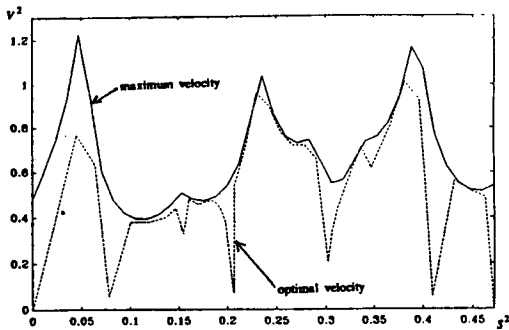


그림 9. 그림 1의 경우의 로봇 2의 최적 속도곡선
 Fig. 9. Optimal velocity of the robot 2 for the path of Fig. 1.

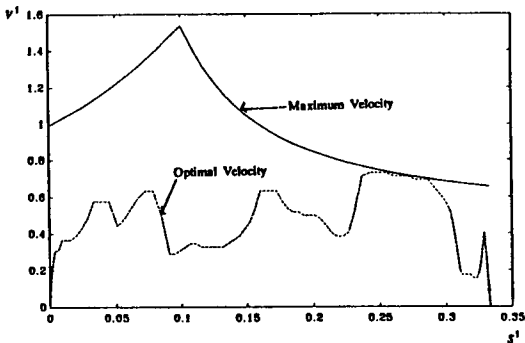


그림 10. 가중치를 가하여 구한 로봇 1의 최적 속도곡선
 Fig. 10. Optimal velocity of the robot 1 for Fig. 1 with weight.

$$J = \sum_{k=0}^{K-1} \left[w_1 \Delta t[k] + w_2 \left(\frac{v^1[k] - v^1[k+1]}{v^1[k] + v^1[k+1]} \right)^2 \right] \quad (25)$$

즉, 최소운동시간과 속도의 변화를 가중치로써 조합하여 두 개의 목표, 작은 운동시간과 작은 속도변화를 추구하는 해를 구한다는 목표이다. (25)의 둘째항의 분모는 속도가 큰 영역에서의 속도변화를 속도가 작은 영역에서의 속도변화를 속도가 작은 영역에서의 속도변화 보다 작은 벌칙을 주기위함이다. w_1 은 1.0 w_2 는 0.01로 했을 때의 결과가 그림 10, 11에 주어졌으며 어느정도 속도의 변화를 줄인 해를 구할 수 있음을 알 수 있다.

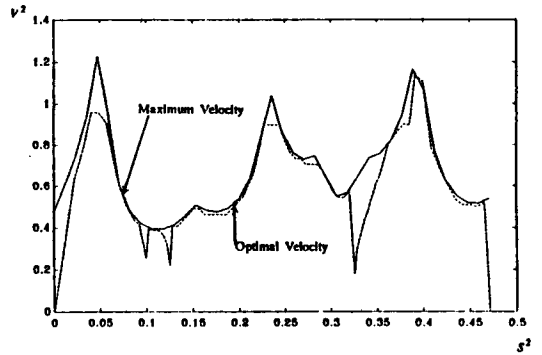


그림 11. 가중치를 가하여 구한 로봇 2의 최적 속도곡선
 Fig. 11. Optimal velocity of the robot 2 for Fig. 1 with weight.

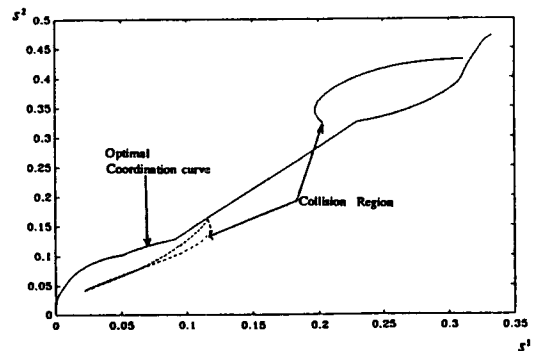


그림 12. 그림 10-11에 의한 조정곡선
 Fig. 12. Coordination curve corresponding to Fig. 10-11.

IV. 결 론

본 논문에서는 두 대의 로봇트가 공간상에 움직여야 할 길이 주어졌 있는 경우에 각 모터의 최대 토크와 허용 회전속도범위를 고려한 충돌없고 시간적으로 최적인 속도계획에 대해 다루었다.

각 모터에 대한 제약조건은 일차적으로 이동거리를 나타내는 변수의 함수꼴로 구해진 속도, 가속도의 제약조건으로 변환되고 이 제약조건은 다시 새롭게 정의된 확장조정공간에서의 조정-속도 곡선에 대한 제약조건으로 변환되었다. 그리하여 원래의 속도계획문제는 3차원 확장조정공간에서 양 끝점이 고정된 3차원 곡선을 구하는 문제로 간단화 되었다. 이 곡선을 찾기위해 확장조정공간은 디스크리타이즈되어 다이나믹 프로그래밍을 적용할 수 있는 형태로 근사화되고 각 제약조건들도 근사화 되었다.

제안된 방법은 예제를 통하여 적용 가능성을 보였다.

궁극적으로는 근사화되지 않은 해석적인 해결방법을 찾는 것이 바람직하므로 지속적인 이분야의 연구가 필요할 것으로 생각된다.

參 考 文 獻

- [1] E. Freund and H. Hoyer, "Real-time path-finding in multirobot systems including obstacle avoidanc," *Int. J. of Robotics Res.*, vol. 7, no. 1, pp. 42-70, Feb. 1988.
- [2] J. Lee and Z. Bien, "Collision-free trajectory control for multiple robots based on neural optimization network," *Robotica*, vol. 8, July-Sept. 1990.
- [3] C. Chang, M. J. Chung and Z. Bien, "Collision-free motion planning for two articulated robot arms using minimum distance functions," *Robotica*, vol. 8, Apr-Jun. 1990.
- [4] B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Trans. on SMC*, vol. SMC-17, no. 1, pp. 21-32, Jan/ Feb. 1987.
- [5] Y. Shin and Z. Bien, "Collision-free trajectory planning for two robot arms," *Robotica*, vol. 7, pp. 205-212, July-Sept. 1989.
- [6] Z. Bien and J. Lee, "A Minimum-time trajectory planning method for two robots," *IEEE Trans. on Robotics and Automat.*, vol. 8, no. 3, pp. 414-418, June. 1992.
- [7] J. E. Bobrow, S. Dubowski and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. of Robotics Res.*, vol. 4, no. 3, pp. 3-17, Fall. 1985.
- [8] K. G. Shin and N. D. McKay, "Minimum-time control robotic manipulators with geometric path constraints," *IEEE Trans. Automatic Control*, vol. AC-30, no. 6, pp. 531-541, Jun. 1985.
- [9] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. Robotics Automation*, vol. RA-3, no. 3, pp. 115-123, Apr. 1987.
- [10] J-J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Trans. Robotics Automation*, vol. 5, no. 1, pp. 118-124, Feb. 1989.
- [11] Z. Shiller and H-H. Lu, "Robust computation of path constrained time optimal motions," *Proc. of IEEE Int'l Conf. Robotics and Automation*, pp. 144-149, 1990.
- [12] D. E. Kirk, *Optimal Control Theory, An Introduction*, Prentice-Hall Inc., New Jersey, 1970.
- [13] 이지홍, 변중남, "공간상의 길이 주어진 두 대의 로봇트를 위한 최소시간 충돌회피 경로계획," 대한전자공학회는문지, 28권, B편, 5호, pp. 40-51, 1991년 5월
- [14] Z. Shiller and S. dubowsky, "Global time optimal motions of robotic manipulators in the presence of obstacles," *Proc. of IEEE Int'l Conf. Robotics and Automation*, pp. 370-375, 1988.
- [15] K. G. Shin and N. D. McKay, "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators," *IEEE Trans. on Automatic Control*. vol. AC31, no. 6, pp. 491-500, June, 1986.

 著 者 紹 介

李 志 弘 (正會員)

1983年 서울대학교 전자공학과 졸업(학사). 1985年 한국과학기술원 전기 및 전자공학과 졸업(석사). 1985年~1988年 현대 중공업 로봇 사업부 근무. 1991年 한국과학기술원 전기 및 전자 공학과 졸업(박사). 1991年 3월~1991年 6월 한국과학기술원 전기 및 전자공학과 연수 연구원. 1991年 7월~현재 금오공과대학 전자 제어공학과 조교수. 주관심분야는 지능제어, 로봇틱스, 뉴로퍼지 시스템 등임.

吳 英 碩 (正會員)

1982年 서울대학교 전자공학과 졸업(학사). 1984年 한국과학기술원 전기 및 전자공학과 졸업(석사). 1991年 한국과학기술원 전기 및 전자공학과 졸업(박사). 1989年 3월~현재 금오공과대학 전자제어공학과 조교수. 주관심분야는 로봇틱스, 컴퓨터 비전, 마이크로 프로세서 응용 등임.