

〈論 文〉

## 反復 學習概念을 利用한 두 臺의 로봇의 衝突回避 軌跡計劃

정낙영\* · 서일홍\*\* · 최동훈\*\*\*

(1990년 7월 30일 접수)

### Collision-Free Trajectory Planning for Dual Robot Arms Using Iterative Learning Concept

Nak Young Chong, Il Hong Suh and Donghoon Choi

**Key Words :** Collision-Free Trajectory Planning(충돌회피 궤적계획), Dual Robot Arms(두 대의 로봇팔), Iterative Learning Concept(반복 학습개념), Linear Programming Technique(선형계획법), Critical Distance(최단거리), Trajectory Error(궤적오차)

#### Abstract

A collision-free trajectory planning algorithm using an iterative learning concept is proposed for dual robot arms in a 3-D common workspace to accurately follow their specified paths with constant velocities. Specifically, a collision-free trajectory minimizing the trajectory error is obtained first by employing the linear programming technique. Then the total operating time is iteratively adjusted based on the maximum trajectory error of the previous iteration so that the collision-free trajectory has no deviation from the specified path and also that the operating time is near-minimal. To show the validity of the proposed algorithm, a numerical example is presented based on two planar robots.

#### 기 호 설 명

$d$	: 거리함수	$n$	: 각 로봇의 관절 수
$d_{cr}$	: 최단거리	$q$	: 관절좌표벡터
$d_{min}$	: 허용 최단거리	$R_p, R_o$	: 위치벡터
$\nabla d$	: 거리함수의 準勾配(subgradient)	$S, C$	: 동일 작업공간 내의 임의 순간에서의 小 공간
$D$	: 거리벡터	$T$	: 작업을 끝내는데 소요되는 총시간
$E_{max}$	: 궤적오차의 최대치	$T_{ei}$	: 최초오차 발생시각
$I$	: 성능지수	$T_{ef}$	: 최종오차 소멸시각
$J$	: 자코비안 행렬	$T_{et}, \Omega_j$	: 궤적오차가 존재했던 총 시간
$L$	: 종로봇 경로의 총길이	$x_1$	: 1번째 링크의 최근접점(critical point on the 1-th link)
$m$	: 로봇 침단부의 위치와 자세를 기술하는데 필요한 차원의 수	$\alpha$	: 스케일링 인자
		$\gamma_i, \omega_j$	: 가중치
		$\Delta t$	: 이산시간영역의 시간간격
		$\Delta q$	: 관절각의 증분

\*정회원, 한양대학교 대학원

\*\*한양대학교 전자공학과

\*\*\*정회원, 한양대학교 공과대학 기계설계공학과

$\Delta X$  : 직교좌표공간 내의 로봇 첨단부의 증분  
 $\varepsilon^*$  : 오차의 허용한계  
 $\lambda_1, \lambda_2$  : 스칼라 파라미터  
 $\tau$  : 반복 회수

## 1. 序 論

一般的으로 여러 개의 로봇이 동시에 運用되는 多로봇시스템(multirobot system)은 로봇들이 作業臺에 投入되는 形態에 따라 다음과 같은 3가지로 大別될 수 있다.

- (1) 孤立된 境遇(isolated case) : 모든 로봇들이 各各의 作業空間을 서로 侵害하지 않는 境遇
- (2) 弱結合된 境遇(loosely coupled case) : 各各의 로봇들이 同一作業空間內에서 獨立인 運動을 하는 境遇
- (3) 强結合된 境遇(tightly coupled case) : 로봇들이 주어진 同一物體를 잡고 처음의 位置에서 원하는 位置로 옮기는 境遇

孤立된 境遇나 强結合된 境遇의 各 로봇들은 서로 충분한 距離를 維持하면서 떨어져 있기 때문에 衝突危險性이 없다. 따라서 現在까지 提案되어 있는 軌跡計劃 技法만으로도 이에 대한 效果的인 制御는 充分히 해결될 수 있다. 그러나 弱結合된 境遇에 있어서는 各各의 로봇들의 運動을 效果的으로 調整해서 運用하지 않으면 모든 로봇들은 서로간의 衝突可能性 때문에 그들 作業時間의 대부분을 作業順序를 기다리는데 所要해버리고 말 것이다. 따라서 生産性的의 向上과 原價의 節減을 爲해서 各 로봇의 作業 休止時間(idle time)을 最小化하는 새로운 運用調整法에 의해 모든 로봇들을 同時に 作動시키는 것이 必要하게 되었다.

多로봇시스템을 調整하기 위한 衝突回避 運動計劃 形態의 經路計劃 方法에 關한 研究는 그리 많지 않은 實情이다<sup>(1-7)</sup>. 그 중에서 Freund와 Hoyer<sup>(1)</sup>는 衝突危險性을 感知하기 위해 假想的인 로봇을 假定하여 多로봇시스템의 온라인(on-line) 衝突回避 技法를 提案하였다. 그러나 그들의 方法은 단지 圓筒型(cylindrical type)의 로봇에만 制限的으로 使用할 수 있었다. Lee와 Lee<sup>(2)</sup>는 衝突地圖(collision map)와 時間計劃(time scheduling)의 概念을 考案하여 두 개의 로봇의 衝突回避 運動計劃을 實現시키는데 適用하였으나 로봇의 內側링크들의 衝突은 考慮하지 않고 손목부의 衝突만을 避하게 하

는데 그치고 있다. Yuh<sup>(6)</sup>의 論文에서는 한 로봇의 現在 位置로부터 最終位置까지의 作動空間을 衝突可能地域으로 定義하고 다른 로봇의 尖端部가 衝突可能地域으로 進入할 것이 豫想되면 그 로봇의 經路를 衝突可能地域의 外側境界를 따르도록 變更시킨다. 그러나 이 方法은 實質的으로 衝突危險이 存在하지 않을 수도 있는 安全地域에까지도 接近을 禁止시킴으로써 그 結果로 얻어지는 經路는 最上の 것이라고 保障할 수 없었다. Shin과 Bien<sup>(4)</sup>은 두 개의 平面 로봇의 指定된 經路들의 모든 衝突回避 調整을 可視化하기 위해 假障礙物(virtual obstacle)이 反映되는 코디네이션 차트(coordination chart)를 導入하였다. 그러나 그들의 方法은 假障礙物을 코디네이션 차트(coordination chart)로 寫像(mapping)시키는데 너무 많은 計算量을 要求하고 있으며 따라서 次元의 增大에 따르는 計算量의 拔擧한 增加로 非效率的인 境遇가 생긴다.

이에 본 研究에서는 主로 弱結合된 境遇의 多로봇시스템에 대한 效果的인 運動調整法을 開發하였다. 弱結合된 境遇의 多로봇시스템은 아크熔接, 噴射塗料, 金屬面의 디버링, 複雜한 組立作業, 檢査等 수많은 作業에 投入될 수 있다. 그와 같은 作業들은 大部分 連續經路制御(continuous path control) 方法을 要求하고 있는데 多로봇시스템인 境遇 作業臺 周圍의 靜止 혹은 移動 障礙物이나 諸로봇 相互간의 衝突危險性을 解決할 수 있는 效果的인 障礙物回避 能力을 追加的으로 要求하고 있다. 그리고 連續經路制御 形態의 거의 大部分의 모든 作業들은 로봇의 尖端部の 速度를 定速度로 維持해 주어야만 한다. 따라서 모든 로봇들을 指定된 經路의 離脫없이 可能的인 最大等速度로 運動시키기 위한 多로봇시스템의 衝突回避運動制御 알고리즘을 開發하게 되었다.

3次元 同一空間內에서 움직이는 두 개의 로봇을 等速度로 指定된 經路를 正確하게 따라갈 수 있게 하기 위한 軌跡計劃(trajjectory planning) 알고리즘이 反復 學習概念을 利用하여 提案되었다. 軌跡誤差(trajjectory error)를 每時間段階에서 원하는 目標位置와 尖端部の 位置 사이의 差벡터의 유클리디안 노름(Euclidean norm)이라고 定義하자. 具體的으로 線形計劃法(linear programming technique)을 利用하여 軌跡誤차를 最小化시키는 衝突回避 時經路가 優先 얻어진다<sup>(9)</sup>. 動作優先度(motion priority)를 줌에 따라 한 로봇은 主로봇(the master),

다른 한 로봇은 從로봇(the slave)으로 指定된다. 主로봇은 指定된 軌跡을 따라가게 되고 從로봇의 關節角의 増分들은 同一間隔의 每時間段階에서 關節角과 關節速度 制限式과 두 臺의 로봇間的 最短距離가 既設定된 許容距離보다 커야된다는 衝突回避條件을 滿足하면서 제 軌跡을 可能하면 近接하게 따라가게 하도록 決定되어진다. 그래서 從로봇의 最適 軌跡은 주어진 制限條件들 下에서 關節運動과 軌跡誤差의 加重合을 最小化함으로써 얻어진다. 그리고 總作動時間은 前反復段階에서 從로봇의 最大 軌跡誤차를 利用하여 反復的으로 調整되어진다. 그와같은 調整에 의해서 衝突回避 軌跡은 指定된 經路로부터 離脫하지 않게 되고 作動時間도 近似的으로 最小가 될 수 있다.

## 2. 本 論

### 2.1 問題敘述

이제 우리는 두 臺의 로봇의 尖端部를 既設定된 經路를 따라 초기狀態에서 최종狀態까지 움직이게 하는 一連의 最適關節座標벡터들을 決定하고자 한다. 이때 可能的 形狀  $q$ 는 關節角과 關節速度 制限範圍內에 있어야하고 두 로봇間的 距離의 項으로 表現되는 衝突回避條件을 滿足해야 한다.

각 로봇의 關節의 數를  $n$ , 로봇 尖端部の 位置와 姿勢를 記述하는데 必要한 次元의 數를  $m$ 이라 하고, 下添字  $m$ 과  $s$ 는 각각 主로봇과 從로봇을 나타낸다고 하자. 慣性座標係에 대한 두 로봇의 尖端部の 位置 및 姿勢  $X_m(t)$ 와  $X_s(t)$ 는 각각의 形狀  $q_m(t)$ 와  $q_s(t)$ 에 의해서 다음과 같이 唯一하게 決定된다.

$$X_m(t) = f_m[q_m(t)] \quad (1.a)$$

$$X_s(t) = f_s[q_s(t)] \quad (1.b)$$

윗 式에서  $f_m$ 과  $f_s: R^n \rightarrow R^m$ 는 非線形이고 連續이며 微分可能的 벡터函數이다. 方程式 (1.a)과 (1.b)를 풀기 위해 다음과 같은 線形化된 運動學的 모델을 利用한다.

$$\dot{X}_m = J_m(q_m)\dot{q}_m \quad (2.a)$$

$$\dot{X}_s = J_s(q_s)\dot{q}_s \quad (2.b)$$

여기서  $J = \partial f / \partial q \in R^{m \times n}$ 는 關節速度와 直角座標係內的 速度를 聯關시키는 자코비안 行列이 된다. 이제 問題는 다음과 같이 敘述될 수 있다.

(問題 1)  $T$ 를 주어진 일을 끝내는 데 필요한 未知의 時間이라고 하고  $X_m^D(t)$ 와  $X_s^D(t)$ ,  $0 \leq t \leq T$ 를 각각 主로봇과 從로봇을 위한 주어진 假軌跡(nominal trajectory)이라 하자.  $\|\cdot\|_n$ 을  $n$ 次元의 最大 노름(norm)이라 하고  $d[\cdot]$ 를 두 로봇間的 最短距離라고 하자. 또  $\epsilon_m^* > 0$ 와  $\epsilon_s^* > 0$ 를 각각 主로봇과 從로봇을 위한 誤差限界라고 하자. 이 때 主로봇의 對應되는 軌跡  $X_m(t)$ 가 식 (3.a)를 滿足하는 關節角의 時間函數  $q_m(t)$ 와 從로봇의 對應되는 軌跡  $X_s(t)$ 가 식 (3.b)를 滿足하는 關節角의 時間函數  $q_s(t)$ 를 衝突回避 條件인 (4)式下에서 決定하라.

$$P_1[X_m(t)] \triangleq \|X_m(t) - X_m^D(t)\|_n \leq \epsilon_m^*, \quad 0 \leq t \leq T \quad (3.a)$$

$$P_1[X_s(t)] \triangleq \|X_s(t) - X_s^D(t)\|_n \leq \epsilon_s^*, \quad 0 \leq t \leq T \quad (3.b)$$

$$d[X_m(t), X_s(t)] > 0. \quad (4)$$

이 問題는 두 臺의 로봇間的 衝突回避뿐만 아니라 經路(path)를 維持해야 하는 作業의 軌跡計劃方法을 包含하고 있다는 意味에서 새롭고 有用하다고 볼 수 있다. 既存의 衝突回避 軌跡計劃 方法들은 一般的으로 經路變更法(path modification)들을 使用하고 있기 때문에 식 (3.a)와 (3.b)에 대한 解를 줄 수 없었고<sup>(1,3)</sup> 速度變更法(velocity modification)을 利用하는 境遇도 定速度 要求條件을 滿足시키지 못했다<sup>(2,4)</sup>.

식 (3.a), (3.b)와 (4)를 同時에 다루기는 어렵기 때문에 여기서 우리는 主로봇이 指定된 軌跡을 正確하게 따라가고 從로봇은 主로봇과의 衝突을 避하면서 제 軌跡을 따라간다고 假定하자. 이 動作優先度는 從로봇을 本來의 指定된 軌跡으로부터 離脫시키게 된다. 이 離脫을 防止하기 위해서 時間스케일링概念을 導入한다.  $t_m$ 과  $t_s$ 를 主로봇과 從로봇의 새로운 時間스케일이라고 하고  $\alpha_m$ 과  $\alpha_s$ 를 각각 主로봇과 從로봇을 위한 스케일링係數라고 하자. 이제 問題 1은 다음과 같이 變形될 수 있다.

(問題 2) 衝突回數 條件 (6)下에서 主로봇의 軌跡  $X_m(t_m)$ 이 식 (5.a)를 滿足하게 하는 最小  $\alpha_m$ 과  $q_m(t_m)$ 을 求하고 同時에 從로봇의 軌跡  $X_s(t_s)$ 가 식 (5.b)를 滿足하게 하는 最小  $\alpha_s$ 와  $q_s(t_s)$ 를 決定하라.

$$P_2[X_m(t_m)] \triangleq \|X_m(t_m) - X_m^D(t_m)\|_n \leq \epsilon_m^*,$$

$$0 \leq t_m \leq \alpha_m T \quad (5.a)$$

$$P_2[X_S(t_s)] \triangleq \|X_S(t_s) - X_S^p(t_s)\|_n \leq \epsilon_m^*$$

$$0 \leq t_s \leq \alpha_s T \quad (5.b)$$

$$d[X_M(t_m), X_S(t_s)] > 0 \quad (6)$$

문제 2에서  $\alpha_s$ 와關節角의軌跡  $q_s(t_s)$ 는  $X_S(t_s)$ 가 (5.b)와 (6)을滿足하도록決定되어야 하는 반면  $\alpha_m$ 과  $t_m$ 은 이미 우리가 主로봇은設定된軌跡을正確하게따라가도록假定했기 때문에 각각 1.0과  $t$ 로固定시킬 수 있다. 그러나, 不幸하게도 既存의 어떠한方法<sup>(1-9)</sup>도 이와 같은問題에適用될 수 없었다. 그래서 이問題를 풀기위하여 反復學習概念을導入하게 되었다.

## 2.2 反復學習概念에 基礎한 解의 接近方法

문제 2를 풀기 위해  $\tau$ 를 反復回數라 하고  $\omega_j$ 와  $\gamma_i$ 를 各各加重值라 하자. 連續時間領域의問題를  $\Delta t_s$ 의時間問題를 갖는 離散時間領域의問題로置換하기 위해  $\Delta X_S$ 와  $\Delta q_s$ 를 各各 直交座標空間內에서의從로봇의增分과 식 (2.b)를滿足하는對應하는關節角의增分이라고 하자. 이增분이 매우 작다는假定下에서  $\Delta X_S \approx J_S \Delta q_s$ 라는關係式이成立될 수 있다. 그러면從로봇의最適關節形狀은 다음과 같은性能指數  $I(\Delta q_s)^r$ 를 每時間段階에서關節角과關節速度制限式과衝突回避條件式下에서最小化하는  $\Delta q_s$ 를 찾음으로써決定될 수 있다.

$$I(\Delta q_s)^r = \max_{j=1, \dots, m} \{ \omega_j | (\Delta X_S - J_S \Delta q_s)_j | \} + \sum_{i=1}^n \gamma_i | \Delta q_{s_i} | \quad (7)$$

그러나 식 (7)을最小化하는最適解는 식 (5.b)를滿足하지 않을 수도 있다. 그래서總作動時間이前反復段階에서의從로봇의最大軌跡誤差에 의해서變更된다. 식 (7)의最小化는 식 (5.b)를滿足할 때까지反復된다.

보다具體的으로 다음과 같이定義하자.

$$E_{\max} \triangleq \sup P_2(X_S), \quad (8)$$

$$\Omega_j \triangleq \{ t_s | P_2[X_S(t_s)] > \epsilon_s^* \} \quad (9)$$

$$T_{er} \triangleq \max \{ t_s | P_2[X_S(t_s)] > \epsilon_s^* \}, \quad (10)$$

$$T_{ei} \triangleq \min \{ t_s | P_2[X_S(t_s)] > \epsilon_s^* \}, \quad (11)$$

$$\text{and } T_{et} = T_{er} - T_{ei}. \quad (12)$$

만일  $\Omega_j$ 가 서로 素인 集合들의 集合集이라면  $T_{er}$ 와  $T_{ei}$ 는 各各 獨立的으로 얻어지고  $T_{er}$ 는 그들의差를 더함으로써決定될 수 있다. 또 最短距離  $d_{cr}$

을 每 反復段階에서의 두 臺의 로봇間的 最小距離라고 하자. 그러면  $\alpha_s$ 는 다음과 같이 주어질 수 있다.

$$\alpha_s = 1 + \frac{E_{\max} - (d_{cr} - d_{\min})}{L} \frac{T_{er}}{T_{ei}} \quad (13)$$

上記 解는 最大誤差가 每反復단계에서 最小化되므로軌跡誤差가 反復的으로 줄어들 것으로期待된다. 그래서 最大軌跡誤差  $E_{\max}$ 는  $T_{er}$ 와  $T_{ei}$ 를 스케일係數로 갖는學習과라미터로選定될 수 있다. 여기서  $T_{er}$ 와  $T_{ei}$ 는 最大誤差가 同一할 境遇에도軌跡誤差의 形態에 큰影響을 미치기 때문에 最大誤差와 함께考慮된다.

距離餘裕를 最短距離  $d_{cr}$  既設定된 許容最短距離  $d_{\min}$ 과의差로定義하자. 이距離餘裕는作動時間을最小化함으로써<sup>(10-13)</sup>生産性的增加를保證하기 위해從로봇이 너무 뒤떨어지는 것을防止할 수 있다. 그래서從로봇이軌跡誤差 없이 主로봇과 너무 많은餘裕를 두고 움직일 때 最短距離는從로봇을加速시키는因子로使用될 수 있고上記因子들의組合으로從로봇의尖端部の速度를調整할 수 있다.

알고리즘

從로봇의 經路의 總길이를  $L$ 이라고 하고 終結試驗을 위한 餘裕限界를  $\epsilon_s^*$ 라고 하자.

{initialize}

GET  $T, d_{\min}, L, \epsilon_s^*$

{iterate}

Do While  $E_{\max} \geq \epsilon_s^*$

Begin

關節角과 關節速度 制限範圍와 衝突回避 條件下에서 식 (7)을最小化한다.

{get  $E_{\max}, T_{er}, T_{ei}, d_{cr}$ }

{scale}  $T^r = \alpha_s T^{r-1}$

{update}  $T$

End

이 때 만일  $\epsilon_s^*$ 를 比較的 크게 잡는다면 이 알고리즘은 단지 點對點制御 (point-to-point control) 시스템에만 局限될 수 밖에 없지만  $\epsilon_s^*$ 를 매우 작게選擇하면 指定된 經路를 正確하게 따라가 주어야하는 作業에도 效果的으로 適用될 수 있다. 그러므로 이 알고리즘은  $\epsilon_s^*$ 를 適切히 選擇함에 따라 大部分

의 産業用 作業에 利用될 수 있다. 우리는 從로봇의 尖端部의 速度를 最大軌跡誤差와 前 反復段階에서의 最短距離의 函數로 變更시키므로 反復이 進行됨에 따라 誤差는 漸次 減少하고 從로봇 尖端部의 速度는 最適値에 接近하게 된다. 結局 經路離脫이 完全히 없어지고 두 臺의 로봇의 作動時間이 最大한 줄어들 수 있는 等速度의 準最適 軌跡을 찾을 수 있게 된다. 참고문헌 (14)에서는 每 反復段階에서의 制御信號를 發生하기 위하여 逆시스템모델과 함께 시스템의 파라미터 評價者가 利用되는 線形週期 制御 시스템類에 反復 學習制御方法이 提案되었다. 衝突回避를 위한 數學的 條件이 다음 節에 記術되어 있다.

2.3 두 로봇의 衝突回避

(1) 衝突回避 條件

$S^k$ 와  $C^k$ 를 時間  $k$ 라는 瞬間의 同一 作業空間內에서 各各 主로봇과 從로봇에 의해서 占有되는 小空間이라고 하자. 그리고

$d(x, C^k) \triangleq \min_{y \in C^k} \|x-y\|$ 이고  $d(S^k, C^k) \triangleq \min_{x \in S^k} d(x, C^k)$ 라고 하자. 여기서  $d(x, C^k)$ 는 從로봇 링크상의 한 點  $x$ 와 主로봇 링크상의 모든 點들간의 最短距離를 意味한다<sup>(6)</sup>.  $d(S^k, C^k)$ 는 두 로봇간의 最短距離를 나타낸다. 그때 두 로봇은 다음과 같은 條件을 滿足하면 衝突을 回避하게 된다.

$$d(S^k, C^k) \geq d_{\min}, \text{ for all } k. \tag{14}$$

距離函數  $d(x, C)$ 의 凸性(convexity)가 두 로봇간의 衝突回避 條件을 附與하는데 利用된다. 具體的으로 距離函數는 다음의 不等式을 滿足하게 된다.

$$d(z, C) \geq d(x, C) + \langle \nabla d, z-x \rangle \text{ for any } z \in R^n \tag{15}$$

여기서  $\langle x, y \rangle \triangleq \sum_{i=1}^n x_i y_i$ 이고  $\nabla d$ 벡터는 點  $x$ 에서의 凸性 函數  $d(x, C)$ 의 準勾配(subgradient)로 定義된다. 軌跡計劃의  $k$ 번째 段階에서 從로봇의 링크  $l$ 과 主로봇의 한 段階 앞선 形狀에 의해서 占有되는 小空間  $C$ 와의 距離를  $d(x_l[k], C[k+1])$ 라고 假定하면,  $x_l[k]$ 는 小空間  $C[k+1]$ 에 가장 近接한 從로봇의 링크상의 한 點을 나타낸다. 主로봇의 現在 位值 代身 한 段階 앞선 位置가 使用된 것은  $k$ 번째 段階 동안의 主로봇의 運動을 考慮해야 하

기 때문이다.  $k$ 번째 段階에서의 從로봇 關節角의 増分  $\Delta q_s[k]$ 는 最近接點의 變位  $\Delta x_l[k]$ 가 다음 不等式을 滿足하도록 求해진다.

$$d(x_l[k], C[k+1]) + \langle \nabla d_l[k], \Delta x_l[k] \rangle \geq d_{\min} \tag{16}$$

上記 式에서  $\nabla d_l[k]$ 는  $x_l[k]$ 에서의 距離函數  $d(x, C[k+1])$ 의 準勾配이다. 増分이 매우 작다고 假定하면  $\Delta x_l[k]$ 은  $J_l[k] \Delta q_s[k]$ 로 近似化될 수 있는데,  $J_l[k]$ 는 最近接點  $x_l[k]$ 의 자코비안 行列이다. 그래서 不等式 (16)은 다음과 같이 다시 쓰여질 수 있다.

$$-\langle J_l[k] \nabla d_l[k], \Delta q_s[k] \rangle \leq d(x_l[k], C[k+1]) - d_{\min}, \quad l=1, \dots, n \tag{17}$$

그러므로 두 臺의 로봇간의 衝突回避 條件은  $\Delta q_s[k]$ 의 線形函數로 表示될 수 있다. 每 最適化段階  $k$ 에서 距離  $d(x_l, C[k+1])$ 와 準勾配들, 그리고 從로봇의 링크상의 最近接點에 對한 자코비안 行列들  $J_l[k]$ 이 求해져야 한다.

(2) 最近接點과 最短距離의 決定

두 로봇의 모든 링크들이 線分으로 表示된다고 假定하자. 그와 같은 假定은 各 링크들의 中心線을 擇하고 두께가 既定된 許容距離에 包含된다면 大部分의 産業用 로봇에 適用될 수 있다.

우리는 먼저 最近接點의 位置와 두 로봇의 링크간의 最短距離를 決定하고자 한다(Fig. 1). 두 링크간의 모든 點들을 나타내기 위해서 스칼라 파라미터  $\lambda_1 \in [0, 1]$ 과  $\lambda_2 \in [0, 1]$ 를 導入하였다. 그러면 두 링크상의 任意的 點  $P$ 와  $Q$ 의 位置벡터는  $\lambda_1$ 과  $\lambda_2$ 를 利用하여 다음과 같이 表現된다.

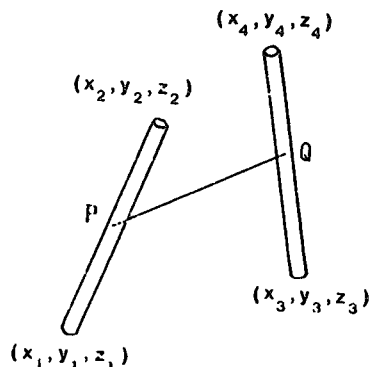


Fig. 1 Distance between the points on two links

$$R_P = [x_1 \ y_1 \ z_1]^T + \lambda_1 [x_2 - x_1 \ y_2 - y_1 \ z_2 - z_1]^T \quad (18)$$

$$R_Q = [x_3 \ y_3 \ z_3]^T + \lambda_2 [x_4 - x_3 \ y_4 - y_3 \ z_4 - z_3]^T \quad (19)$$

各 링크의 任意的 두 點間의 距離벡터는 두 媒介變數化된 點들의 位置벡터의 差異로 定義될 수 있다.

$$D(\lambda_1, \lambda_2) = R_Q - R_P \quad (20)$$

이제 두 링크間의 最短距離를 求하기 위해서  $\lambda_1$ 과  $\lambda_2$ 에 대한 다음의 2次式이 數式化될 수 있다.

$$\text{Minimize } \|D\|^2_{1,2} \quad (21.a)$$

$$\text{subject to } 0 \leq \lambda_1 \leq 1, 0 \leq \lambda_2 \leq 1. \quad (21.b)$$

上記 式의 解는 IMSL의 ZXMWD와 같은 잘 알려진 2次計画法 소프트웨어를 利用하여 쉽게 얻을 수 있다.

$k$ 時間段階에서의 從로봇의 링크  $l$ 과  $k+1$ 時間段階에서 主로봇의 링크  $l'$ 間의 最短距離를  $d(x_l[k], x_{l'}[k+1])$ 라고 表示하면, 두 臺의 로봇間의 衝突回避條件은 式 (17)을 擴張하여 다음과 같이 나타낼 수 있다.

$$\begin{aligned} & -\langle J^T[k] \nabla d_{ll'}[k], \Delta q_s[k] \rangle \\ & \leq d(x_l[k], x_{l'}[k+1]) - d_{\min}, \\ & l'=1, \dots, n' \text{ for } l=1, \dots, n \end{aligned} \quad (22)$$

여기서  $n'$ 는 主로봇의 링크의 數,  $\nabla d_{ll'}$ 는  $x_l[k]$ 點에서 從로봇의  $l$ 번째 링크와 主로봇의  $l'$ 번째 링크를 連結하는 距離函數의 準勾配이다.

任意 形狀을 갖는 固定 또는 靜止障礙物의 境遇도 障礙物을 部分的인 線分들의 단힌 連結로 모델링함으로써 쉽게 包含시킬 수 있다. 任意 形狀의 障礙物을 線分으로 精確하게 모델링하는 경우, 衝突回避를 위한 條件式들은 크게 늘어나게 되나 線形計画法을 利用함으로써 그 複雜性을 줄일 수 있다.

## 2.4 線形計画法에 의한 軌跡計劃

이제  $k$ 段階에서의 最適化問題는 다음과 같이 定式化될 수 있다.

$$\begin{aligned} & \text{Minimize } \max_{\Delta q_s[k]} \{ \omega_j | \Delta X_s[k] \\ & - J_s[k] \Delta q_s[k]_j | \} + \sum_{i=1}^n \gamma_i | \Delta q_s[k]_i | \quad (23) \\ & \text{subject to } -\langle J^T[k] \nabla d_{ll'}[k], \Delta q_s[k] \rangle \end{aligned}$$

$$\leq d(x_l[k], x_{l'}[k+1]) - d_{\min}, \quad l'=1, \dots, n' \quad (24)$$

$$q_s^{i \min} \leq q_s[k+1] \leq q_s^{i \max} \quad (25)$$

$$\Delta q_s^{i \min} \leq \Delta q_s[k] \leq \Delta q_s^{i \max} \quad (26)$$

最適化指數 式 (23)은 關節角의 增分과  $\Delta X_s[k] = J_s[k] \Delta q_s[k]$ 系의 解에 對한 체비셰프(Chebyshev)意味의 誤差의 加重을 나타낸다. 式 (24)는 衝突回避를 爲한 線形條件이고 式 (25)와 (26)은 各各 關節角과 關節速度의 制限條件이다.

上記 式을 標準 線形計画法의 問題<sup>(9)</sup>로 바꾸기 위해 다음과 같은 陰이 아닌 變數들을 導入한다.

$$x_i - x_{n+i} = \Delta q_i \quad i=1, \dots, n \quad (27)$$

$$x_{2n+1} = \max_j | \omega_j (\Delta X - J \Delta q)_j | \quad j=1, \dots, m \quad (28)$$

여기서  $x_k \geq 0$ ,  $k=1, \dots, 2n+1$ 이고 時間段階와 從로봇을 爲한 記號는 便宜上 省略했다. 그러면, 式 (24), (25), (26)의 制限條件을 갖는 最適化問題 式 (23)은 다음과 같이 다시 쓸 수 있다.

$$\text{Minimize } x_{2n+1} + \sum_{i=1}^n \gamma_i x_i + \sum_{i=1}^n \gamma_i x_{n+i} \quad (29)$$

$$\text{subject to } -\sum_{i=1}^n (J_i^T \nabla d_{ll'})_i (x_i - x_{n+i}) \leq$$

$$d(x_l, x_{l'}) - d_{\min}, \quad l'=1, \dots, n' \text{ for } l=1, \dots, n \quad (30)$$

$$x_i - x_{n+i} \leq q_{i \max} - q_i, \quad i=1, \dots, n \quad (31.a)$$

$$x_{n+i} - x_i \leq q_i - q_{i \min}, \quad i=1, \dots, n \quad (31.b)$$

$$x_i - x_{n+i} \leq \Delta q_{i \max}, \quad i=1, \dots, n \quad (32.a)$$

$$x_{n+i} - x_i \leq -\Delta q_{i \min}, \quad i=1, \dots, n \quad (32.b)$$

$$-\omega_j \sum_{i=1}^n J_{ji} (x_i - x_{n+i}) - x_{2n+1} \leq -\omega_j \Delta X_j, \quad j=1, \dots, m \quad (33.a)$$

$$\omega_j \sum_{i=1}^n J_{ji} (x_i - x_{n+i}) - x_{2n+1} \leq \omega_j \Delta X_j, \quad j=1, \dots, m \quad (33.b)$$

$$x_k \geq 0, \quad k=1, \dots, 2n+1 \quad (34)$$

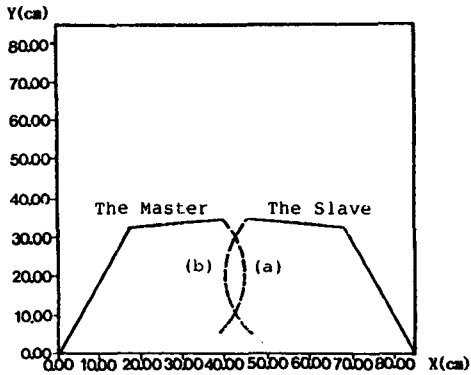
$\Delta q_i$ 가 陽의 값을 갖기 위해서  $x_{n+i}$ 가 零이 되고  $\Delta q_i$ 가 陰의 값을 갖기 위해서  $x_i$ 가 零이 되어야 되므로 다음과 같은 等值가 成立하게 된다.

$$\begin{aligned} & \text{Minimize } x_{2n+1} + \sum_{i=1}^n \gamma_i | x_i - x_{n+i} | \\ & \Leftrightarrow \text{Minimize } x_{2n+1} + \sum_{i=1}^n \gamma_i (x_i + x_{n+i}) \end{aligned}$$

식 (30)은 衝突回避를 위한 條件이고 식 (31)과 (32)는 各各 關節角과 關節速度制限式이다. 식 (33)은 새로운 變數  $x_{2n+1}$ 의 導入으로 인하여 생기는 附加制限條件이다. 每時間段階에서 上記問題는 線形計劃法을 利用하여  $x_1, \dots, x_{2n+1}$ 에 대해서 풀 어지고 식 (27)을 利用하여 最適值  $\Delta q_1, \dots, \Delta q_n$  이 얻어진다.

2.5 컴퓨터 檢證 및 考察

提示된 알고리즘의 有用성을 보이기 위하여 初期 形狀이 Fig. 2에 나타나 있는 것과 같은 두 臺의 2 링크 平面 로봇이 導入되었다. 動作優先度는 왼쪽 의 로봇에 주어졌다. 從로봇은  $T=0.417$ 초 동안



(a) The Master's desired trajectory  
(b) The Slave's desired trajectory

Fig. 2 Dual robot arms in a common workspace

Table 1 Link lengths and initial configurations

	Master	Slave
link lengths [m]	$l_{m1}=0.37$ $l_{m2}=0.23$	$l_{s1}=0.37$ $l_{s2}=0.23$
initial configuration [degree]	$q_m^1=62$ $q_m^2=-57$	$q_s^1=118$ $q_s^2=57$

Table 2 Joint angle and velocity constraints of the slave

	Lower bounds	Upper bounds
joint angle [degree]	$q_s^{1min}=-180$ $q_s^{2min}=0$	$q_s^{1max}=180$ $q_s^{2max}=180$
joint velocity [rad/sec]	$\dot{q}_s^{1min}=-2.0$ $\dot{q}_s^{2min}=-2.5$	$\dot{q}_s^{1max}=2.0$ $\dot{q}_s^{2max}=2.5$

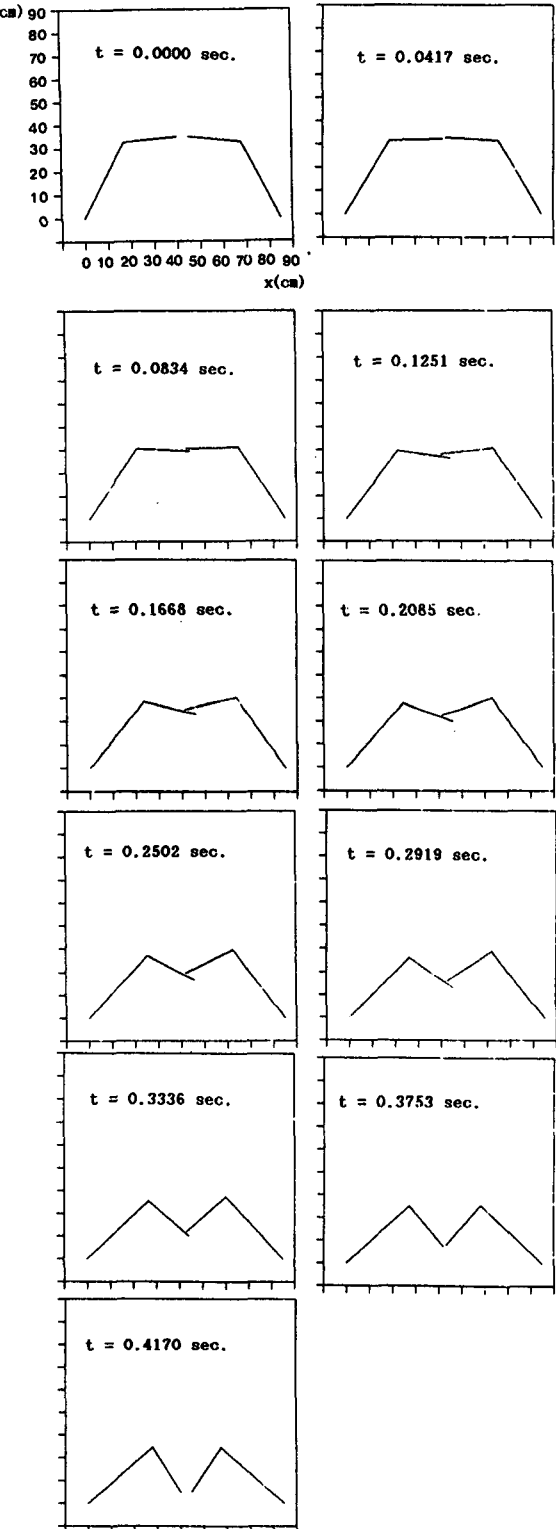


Fig. 3 Simulation results for dual robot arms

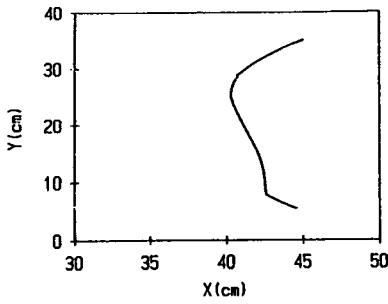
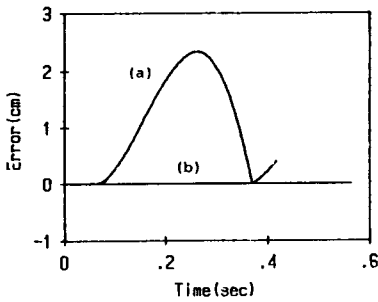
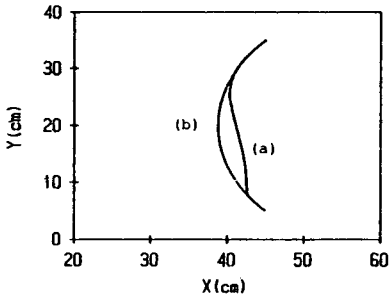


Fig. 4 The end-effector trajectory of the slave



(a) After 1st Iteration with  $T=0.417$ sec.  
(b) After 2nd Iteration with  $T=0.561$ sec.

Fig. 5 The trajectory error of the slave



(a) After 1st Iteration with  $T=0.417$ sec.  
(b) After 2nd Iteration with  $T=0.561$ sec.

Fig. 6 The improvement of the end-effector trajectory through learning process

中心點이 (0.6, 0.2, 0.0)[m]이고 始點과 終點이 各各 (0.45, 0.35, 0.0)[m]와 (0.45, 0.05, 0.0)[m]인 孤를 따라 움직이고 主로봇은 中心點이 (0.25, 0.20, 0.0)[m]이고 始點과 終點이 各各 (0.4, 0.35, 0.0)[m]와 (0.4, 0.05, 0.0)[m]인 孤를 따라 움직인다. 두 로봇의 링크 길이와 初期 形狀은 各各 Table 1에 나타나 있고 終로봇의 關

節角과 關節速度 制限範圍는 Table 2에 나타나 있다. 加重值  $\gamma_1=2.0$ ,  $\gamma_2=1.0$ ,  $\omega_1=10.0$ ,  $\omega_2=10.0$  를 갖는 境遇에 學習過程을 거치지 않은 終로봇 尖端部의 軌跡이 Fig. 3에 나타나 있다. Fig. 3과 4로부터 두 대의 로봇은 衝突을 回避했다는 것을 알 수 있으나 終로봇의 軌跡은 0.0001인  $\epsilon^*_s$ 보다 많이 離脫했음을 알 수 있다. 이 離脫을 줄이기 위해 加重值들을 動的으로 變換시키는 것이 훨씬 合理的이나 그런 加重值들을 찾는 것은 매우 어려운 問題이다. 反面에 Fig. 5와 6은 時間段階가 增加함에 따라 誤差가 減少하는 것을 나타냄으로써 從로봇의 尖端部의 軌跡이 漸次的으로 改善되어지는 것을 보여주고 있다. Fig. 5와 6으로부터 두번째 反復段階에서부터  $\epsilon^*_s$ 보다 작은 0근처로 最大 軌跡 誤차가 줄어들음을 알 수 있다.

첫번째 反復段階에서 時間段階를 600개로 잡았을 때 VAX-8700으로 約 200CPU時間이 所要되었고 線形計劃法의 問題를 풀기 위해 IMSL의 ZX3LP가 使用되었다.

### 3. 結 論

3次元 同一作業空間內에서 두 臺의 로봇의 衝突 回避 軌跡計劃 알고리즘이 指定된 經路를 따라 可能한 한 最大 等速度를 낼 수 있도록 反復 學習概念을 利用하여 提案되었다. 本 알고리즘은 다음과 같은 意味에서 매우 效果的이라고 判明되었다.

- (1) 두 臺의 로봇의 尖端部뿐만 아니라 모든 링크들 간의 衝突을 同時에 考慮했고,
  - (2) 한 空間에서 다른 空間으로의 寫像(mapping)을 要求하지 않는다.
- 또 恒常 唯一한 解를 주는 線形計劃法을 利用할 수 있는 形態로 數式化되었다.

3次元 多로봇시스템의 境遇도 各 로봇의 軌跡이 動作優先도에 따라 順次的으로 計劃된다면 쉽게 擴張될 수 있을 것이다. 提示된 알고리즘의 數學的 收斂性을 證明하는 問題와 各 로봇의 動力學的 要素를 包含하는 軌跡計劃 問題에 關한 研究가 계속 進行되어야 할 것이다.

### 後 記

本 論文은 1989年度 文敎部 支援 韓國學術振興財團의 一般課題 學術研究 助成費에 依하여 研究되어



진 것으로 研究支援에 對하여 關係者 여러분께 深  
 深한 謝意를 表하는 바입니다.

참 고 문 헌

- (1) Freund, E. and Hoyer, H., 1986, "Pathfinding in Multi-Robot Systems: Solution and Applications", Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 103~111.
- (2) Lee, B.H. and Lee, C.S.G., 1987, "Collision-Free Motion Planning of Two Robots", IEEE Trans. Syst. Man, and Cybern., Vol. SMC-17, No. 1, Jan./Feb., pp. 21~32.
- (3) Yuh, J., 1987, "On-Line Adaptive Collision Avoidance Motion Control Strategy for Two Planar Robots", Proc. IEEE Int. Conf. Syst. Man., Cybern., pp. 21~25.
- (4) Shin, Y.S. and Bien, Z., 1988, "A Novel Method of Collision-Free Trajectory Planning for Two Robot Arms", Proc. IEEE Int. Conf. Syst. Man, and Cybern., Beijing, China, Vol. 2, pp. 791~794.
- (5) Fortune, S., Wilfong, G., and Yap, C., 1986, "Coordinated Motion of Two Robot Arms", Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1216~1223.
- (6) Erdmann, M. and Lozano-Peréz, T., 1986, "On Multiple Moving Objects", Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1419~1424.
- (7) Tournassoud, P., 1986, "A Strategy for Obstacle Avoidance and Its Application to Multi-Robot Systems", Proc. IEEE Int. Conf. Robotics Automat., San Francisco, pp. 1224~1229.
- (8) Gilbert, E.G., and Johnson, D.W., 1985, "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles", IEEE J. Robotics Automat., Vol. RA-1, No. 1, Mar., pp. 21~30.
- (9) Generozov, V. L., 1984, "Algorithms for Trajectory Planning in the Presence of Obstacles", Tekhnicheskaya Kibernetika, No. 1.
- (10) Shin, K.G. and Mckay, M.D., 1986, "Selection of Near-MinimumTime Geometric Path for Robotic Manipulators", IEEE Trans. Automat. Contr., Vol. AC-31, No. 6, June, pp. 501~511.
- (11) Shin, K.G. and Mckay, M.D., 1986, "A Dymanic Programming Approach to Trajectory Planning of Robotic Manipulators", IEEE Trans. Automat. Contr., Vol. AC-31, No. 6, June, pp. 491~500.
- (12) Shin, K.G. and Mckay, M.D., 1985, "Minimum Time Control of Robotic Manipulators with Geometric Path Constraints", IEEE Trans. Automat. Contr., Vol. AC-30, No. 6, June, pp. 531~541.
- (13) Bobrow, J.E., Dubowsky, S., and Gibson, J.S., 1985, "Time-Optimal Control of Robotic Manipulators Along Specified Paths", Int. J. Robot. Res., Vol. 4, No. 3, Fall, pp. 3~17.
- (14) Oh, S.R., Bien, Z., and Suh, I.H., 1988, "An Iterative Learning Control Method with Application for the Robot Manipulators", IEEE J. Robotics Automat., Vol. 4, No. 5, Oct., pp. 508~514.