

로봇 매니플레이터의 카르테시안 직선운동을 위한 경로점의 최소화

(Determination of the Minimum Number of Intermediate Points for the Robot Manipulator Cartesian Straight Motion)

吳世鉉*, 全洪兌*

(Se Hyun Oh and Hong Tae Jeon)

要約

복잡한 작업공간에서 로봇 매니플레이터의 경로(path) 혹은 궤적(trajjectory)을 설정할 때, 카르테시안(cartesian) 공간상에서 임의의 두 점간의 직선운동은 타월한 잇점들을 갖는다. 그러나 이러한 직선운동을 수행하는 것은 간단한 일이 아니다. 표준적인 방법으로써 직선 경로에 충분한 경로점을 설정하여 조인트(joint) 공간상에서 두 개의 인접한 경로점을 선형적으로 보간시키는 근사적인 방법이 행해지고 있다. 이러한 방법에서 최소의 경로점을 결정하는 것은 여러면에서 매우 중요하다고 할 수 있다.

본 논문에서는 이러한 목적을 효과적으로 성취하기 위한 새로운 방법이 제시된다. 이 결과의 효용성은 PUMA 560 매니플레이터(manipulator)를 사용하여 입증된다.

Abstract

Between any two points in the Cartesian space, the straight motion has many distinct advantages in path/trajjectory plannings, specially in a crowded workspace. But to achieve this motion is a nontrivial task. One standard way to approximate this motion is to place enough intermediate points along the desired path and linearly interpolate two adjacent intermediate points in the joint space.

In this approach, however, the determination of the minimum number of intermediate points is very important from several aspects.

A scheme that can effectively accomplish this purpose is established in this paper. This scheme is based on several search methods. The results are demonstrated using the PUMA 560 series manipulator.

I. 서론

주어진 작업 공간내의 임의의 두 점 사이를 로봇 매니플레이터(manipulator)가 수행할 수 있는 경로

(path)는 무한히 많지만, 그중 직선 경로는 다른 경로들에 비해 많은 잇점들을 가지고 있다.[1] 즉, 직선 경로는 비교적 예측이 쉽고, 두 점 사이의 최단 거리로 실제 작업공정에서 흔히 사용되며 또한 최소의 관성력을 갖는다. 그러나 로봇 매니플레이터는 다관절 조합체로 end-effector가 직선경로를 수행하도록 제어하는 일은 어려운 문제로 간주되고 있다.

*正會員, 中央大學校 電子工學科
(Dept. of Elec. Eng., Chungang Univ.)
接受日字: 1987年 7月 13日

Whitney [13]와 Paul [7]은 각각 inverse Jacobian과 inverse kinematics을 이용한 차동 직선운동(differential straight line motion) 방식을 제시하였다. 이 방식은 많은 계산량(매 샘플링 마다 inverse kinematics 혹은 Jacobian을 계산함)을 필요로 하고 실제 구동에 있어 조인트의 여러 제약 조건들을 만족시키지 못하며, 또한 유연성(smoothness)이 결여된 운동을 낳는다는 단점들을 내포하고 있다. 상기의 문제점을 효율적으로 해결하기 위해 Taylor [11]와 Lin [4] 등이 각각 조인트 공간(joint space) 상에서 1, 2차 혼합 함수 및 3차 함수를 이용하여 근사화된 직선 운동을 수행하기 위한 조인트체적 산출 방법을 제시하였다. 본 논문에서는 함수의 차수가 높아지면 매니퓰레이터의 진동이 많이 발생하고 수명(life cycle)이 단축된다는 점에 착안하여, 1, 2차 혼합 함수를 근사화된 직선운동을 위한 조인트 체적의 기저함수로 사용한다.

그림 1에서 연속적인 두 점($q^0 - q^1$ 혹은 $q^1 - q^2$) 사이의 선형 보간은 이웃 구간에서의 속도의 불연속성을 초래한다. 속도의 연속성을 유지하기 위해 변이시간 τ_1 동안 일정한 가속도를 가해야 한다. 일반적으로 변이시간 τ_1 은 경로점으로 부터 조인트 체적의 과도한 이탈을 방지하기 위해 $\min \{t_1/\tau_1, t_2/\tau_2\} > K$ (K 는 상수)의 조건을 만족해야 한다. 만약 구간 수행시간 t_1 과 t_2 가 미리 주어지고, 계산된 변이시간 τ_1 이 상기 조건을 만족시키지 못할 경우, t_1 혹은 t_2 의 증가를 필요로 한다. 이는 q^0 와 q^1 사이에서 매니퓰레이터의 주행시간을 증대시킨다. 따라서 그림 1의 예를 참고로 하면 경로점의 수가 증가할수록 전체 주행시간이 증대된다는 사실을 쉽게 알 수 있다. 또한 경로점에서 일정 가속 구간의 설정은 매니퓰레이터 수명에 심각한 영향을 미치는 급격한 동작(jerk)의 증대를 가져오기 때문에 가급적 경로점의 수를 최소화시키는 것이 필요하다. 따라서 로봇 매니퓰레이터 직선경로 제어에 있어 경로점의 최소화는 매우 중요한 문제로 인식된다.

경로점을 결정하기 위해 Taylor는 Bounded Deviation Paths(BDP)를 제안하였다. 그러나 Taylor의 방법은 경로점의 최소화를 얻는데 미흡한 것으로 판단된다. 이는 Taylor의 방법이 "직선운동을 조인트 공간 상에서 선형보간에 의해 근사화 시켰을 때 대부분의 매니퓰레이터는 최대오차를 구간의 중간점에 갖는다"라는 가정하에서 이루어졌다는 점에 기인된다. 이러한 가정의 모호성은 Brady [1]가 제시한 간단한 2개의 축으로 이루어진 평면(planar) 매니퓰레이터의 직선운동에서도 쉽게 입증된다.

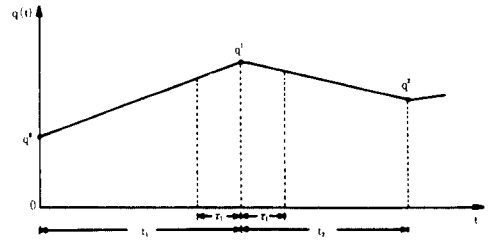


그림 1. 선형구간 사이에서의 변이구간 설정
Fig. 1. Transition between linear joint motions.

본 논문에서는 이러한 가정없이 주어진 허용오차 내에서 근사화된 직선운동을 위한 경로점의 수를 최소화시킬 수 있는 방법을 제시하고자 한다.

II. 위치(Position) 오차와 방위(Orientation) 오차의 정의

카르테시안 공간상에서 임의의 두점 X_1 와 X_2 사이(주행시간 t_d)의 직선경로(straight line path)는 다음과 같이 서술된다.

$$X(\lambda) = \begin{bmatrix} R(\lambda) P(\lambda) \\ 0 & 1 \end{bmatrix} \quad (1)$$

$$p(\lambda) = p_1 + \lambda(p_2 - p_1) \quad (2)$$

$$R(\lambda) = R_1 \text{Rot}(k, \lambda\theta) \quad (3)$$

여기서 $P(\lambda)$ 와 $R(\lambda)$ 는 각각 위치벡터 및 방위 매트릭스를 나타내며, $\lambda (=t/t_d)$ 는 normalized 시간을 나타낸다. 또한 벡터 K 와 각도 θ 는 $\text{Rot}(K, \theta) = R_1^{-1} \cdot R_2$ 로 부터 결정된다. X_1 와 X_2 에 대응되는 조인트 값들 q_1 와 q_2 라 한다면, 두 조인트값들 사이를 선형적으로 보간시켜준 조인트값 $\hat{q}(\lambda)$ 와 그에 대응되는 카르테시안 공간상의 경로 $\hat{X}(\lambda)$ 는 다음과 같이 나타낼 수 있다.

$$\hat{q}(\lambda) = q_1 + \lambda(q_2 - q_1) \quad (4)$$

$$\hat{X}(\lambda) = \begin{bmatrix} \hat{R}(\lambda) \hat{p}(\lambda) \\ 0 & 1 \end{bmatrix} \quad (5)$$

여기서 $\hat{q}(\lambda) \in R^n$ (n 자유도인 경우), $\hat{P}(\lambda) \in R^3$, $\hat{R}(\lambda) \in R^{3 \times 3}$, $f(\cdot)$ 는 kinematics 방정식을 나타낸다.

두 점 X_1 와 X_2 의 조인트 값 q_1 와 q_2 를 선형적으로 보간 시켰을 경우, 매니퓰레이터의 end-effector는 원래 설정했던 카르테시안 직선경로($X(\lambda)$)에서 이탈하게 된다. 이때 발생하는 오차는 위치오차 $e_p(\lambda)$ 와 방위오차 $e_R(\lambda)$ 로 구별되며 이는 다음과 같다 [11]. 그림 2는 이들 오차를 카르테시안 공간상에서 표시해 준 것이다.

$$e_p(\lambda) = |\hat{p}(\lambda) - p(\lambda)| \quad (6)$$

$$e_R(\lambda) = |\text{angle part of } R^{-1}(\lambda)\hat{R}(\lambda)| \quad (7)$$

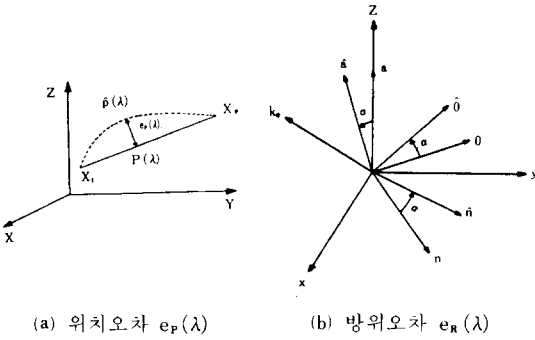


그림 2. 위치오차와 방위오차의 정의
Fig. 2. Definition of $e_p(\lambda)$ and $e_R(\lambda)$.

III. 최소의 경로점들 설정방법

카르테시안 공간상의 출발점 X_s 와 목표점 X_d 사이의 직선경로는 다음과 같이 서술된다.

$$X(\beta) = \begin{bmatrix} R(\beta) & p(\beta) \\ 0 & 1 \end{bmatrix} \quad (8)$$

$$p(\beta) = p_s + \beta(p_d - p_s) \quad (9)$$

$$R(\beta) = R_s \text{Rot}(k_s, \beta \theta_s) \quad (10)$$

여기서 $\beta \in [0, 1]$ 는 X_s 와 X_d 사이의 normalized 거리를 나타내며, 벡터 K_s 와 각도 θ_s 는 $R_{01}(K_s, \theta_s) = R_s^{-1} \cdot R_d$ 로 부터 결정된다.

위와 같은 식으로 카르테시안 공간상의 직선경로상에 있는 모든 지점의 위치와 방위가 β 로써 정의된다.

1. 최대오차 탐색방법

그림 3과 같이 카르테시안 공간상의 임의의 두 경로

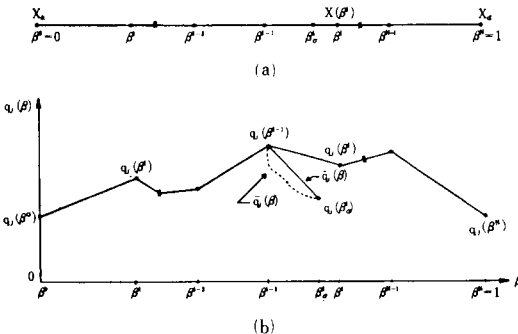


그림 3. (a) Normalized 거리(β)로 나타낸 카르테시안 직선경로
(b) 조인트 공간상에서 j번째 조인트의 선형보간
Fig. 3. (a) A Cartesian straight line path represented in a normalized distance. β .
(b) Linear interpolations of joint j in the joint space.

점 β^{i-1} 과 β^i 사이에서 j번째 조인트의 선형보간 함수 $\hat{q}_j(\beta)$ 는 식(4)에 의해 아래와 같이 쓸 수 있다.

$$\hat{q}_j(\beta) = q_j(\beta^{i-1}) + \frac{\beta - \beta^{i-1}}{\beta^i - \beta^{i-1}} |q_j(\beta^i) - q_j(\beta^{i-1})| \quad (11)$$

$$\text{여기서 } \lambda \left(= \frac{t}{t_d} \right) = \frac{\beta - \beta^{i-1}}{\beta^i - \beta^{i-1}}$$

조인트 공간상의 오차 $e_j(\beta)$ 는 다음과 같이 쓸 수 있다¹⁰⁾.

$$e_j(\beta) = \bar{q}_j(\beta) - \hat{q}_j(\beta) = \frac{(\beta - \beta^{i-1})(\beta - \beta^i)}{2} \ddot{q}_j(\xi_\beta), \quad j=1, 2, \dots, 6 \quad (12)$$

여기서 \bar{q}_j 는 설정된 경로에 대응되는 실제의 조인트 함수, $\ddot{q}_j \triangleq \frac{d^2 \hat{q}_j}{d\beta^2}$, $\beta^{i-1} \leq \xi_\beta \leq \beta^i$

일반적으로 조인트들의 미소변화에 따르는 위치와 방위의 변화율은 아래와 같다.

$$\begin{bmatrix} dx \\ dy \\ dz \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} = J \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix} \quad (13)$$

여기서 $J \in R^{6 \times 6}$ 는 Jacobian 매트릭스를 나타내고, $[dx \ dy \ dz]^T$ 와 $[\delta_x \ \delta_y \ \delta_z]^T$ 는 각각 기준좌표계에 대한 end-effector의 위치와 방위의 변화율을 나타낸다. 또한 $[dq_1, dq_2, dq_3, dq_4, dq_5, dq_6]^T$ 는 조인트의 미소 변화율을 나타낸다.

만약 $dq_j = e_j(\beta)$ 로 가정해 보면, 식(13)은 다음과 같이 나타낼 수 있다.

$$\begin{bmatrix} \hat{d}_x \\ \hat{d}_y \\ \hat{d}_z \\ \hat{\delta}_x \\ \hat{\delta}_y \\ \hat{\delta}_z \end{bmatrix} = J \begin{bmatrix} e_1(\beta) \\ e_2(\beta) \\ e_3(\beta) \\ e_4(\beta) \\ e_5(\beta) \\ e_6(\beta) \end{bmatrix} = \frac{(\beta - \beta^{i-1})(\beta - \beta^i)}{2} J \begin{bmatrix} \ddot{q}_1(\xi_\beta) \\ \ddot{q}_2(\xi_\beta) \\ \ddot{q}_3(\xi_\beta) \\ \ddot{q}_4(\xi_\beta) \\ \ddot{q}_5(\xi_\beta) \\ \ddot{q}_6(\xi_\beta) \end{bmatrix} \quad (14a)$$

$$\triangleq \frac{(\beta - \beta^{i-1})(\beta - \beta^i)}{2} \begin{bmatrix} \hat{e}_1(\beta) \\ \hat{e}_2(\beta) \\ \hat{e}_3(\beta) \\ \hat{e}_4(\beta) \\ \hat{e}_5(\beta) \\ \hat{e}_6(\beta) \end{bmatrix} \quad (14b)$$

위의 결과를 이용하면 식(6), (7)에서 정의한 위치와 방위오차는 다음과 같이 쓸 수 있다.

$$e_p(\beta) = \left| \frac{(\beta - \beta^{i-1})(\beta - \beta_o^i)}{2} \hat{e}_p(\beta) \right| \quad (15)$$

$$e_n(\beta) = \left| \tan^{-1} \left\{ \frac{(\beta - \beta^{i-1})(\beta - \beta_o^i)}{2} \hat{e}_n(\beta) \right\} \right| \quad (16)$$

여기서 $\hat{e}_p(\beta) = \{\hat{e}_1^2(\beta) + \hat{e}_2^2(\beta) + \hat{e}_3^2(\beta)\}^{1/2}$ 이고 $\hat{e}_n(\beta) = \{\hat{e}_1^2(\beta) + \hat{e}_2^2(\beta) + \hat{e}_3^2(\beta)\}^{1/2}$ 이다.

식(15), (16)을 살펴보면 $d\hat{e}_p(\beta)/d\beta = d\hat{e}_n(\beta)/d\beta = 0$ 인 경우는 Taylor의 가정대로 최대오차가 unimodal이면 정 중앙에 존재하지만, 실제로 있어서는 $\hat{e}_p(\beta)$ 나 $\hat{e}_n(\beta)$ 는 상수가 될 수 없다. 따라서, 이는 각각의 매니플레이터 마다 다르므로 다음의 두가지 경우로 나눌 수 있다.

(a) $\frac{d\hat{e}_p(\beta)}{d\beta}, \frac{d\hat{e}_n(\beta)}{d\beta}$ 가 아주 작은 경우

(unimodal인 경우)

(b) $\frac{d\hat{e}_p(\beta)}{d\beta}, \frac{d\hat{e}_n(\beta)}{d\beta}$ 가 큰 경우

(multimodal인 경우)

1) Unimodal인 경우의 최대오차 탐색

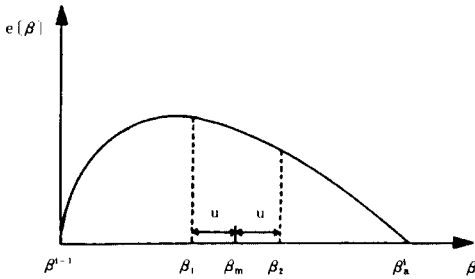


그림 4. 오차의 분포가 Unimodal인 경우.
Fig. 4. Unimodal case of the deviation.

오차가 unimodal인 경우 최대오차 탐색을 위해 Fibonacci 탐색방법을 사용한다. 탐색 방법은 그림 4와 같이 탐색점 β_m 을 기준으로 양측에 β_1 과 β_2 를 설정하여 각각의 오차값을 계산한후, 그중 작은쪽을 β^{i-1} 이나 β_o^i 로 놓고 위의 과정을 반복하여 최대오차의 값을 찾을 수 있는 방법이다. 탐색점 β_m 은 Fibonacci 수열을 이용하여 설정되며 상세한 알고리즘은 다음과 같다.

[알고리즘 1 : 최대 위치오차 (e_m^*) 탐색]

[단계 1] $u, r (<< 1)$ 을 선택. 식(8), (9), (10)을 이용하여 $X(\beta^{i-1})$ 과 $X(\beta_o^i)$ 를 계산. 그리고 $X_1 = X(\beta^{i-1}), X_2 = X(\beta_o^i)$ 로 놓음

[단계 2] $\beta_1 = \beta^{i-1} + \beta_o^i/2 - u, \beta_2 = \beta^{i-1} + \beta_o^i/2 + u$ 로 놓는다. 만일 $\beta_o^i - \beta^{i-1} < r$ 면, $e_m^* = e_p(\beta^{i-1} + \beta_o^i/2)$ 로 하여 정지. 그렇지 않으면 계속.

[단계 3] 식(1)~(7)에 의해 $e_p(\beta_1)$ 과 $e_p(\beta_2)$ 를 계산

[단계 4] 만일 $e_p(\beta_1) > e_p(\beta_2)$ 면, $\beta_o^i = \beta_2$ 로 해서 [단계 2]로 그렇지 않으면 계속

[단계 5] 만일 $e_p(\beta_1) < e_p(\beta_2)$ 면, $\beta^{i-1} = \beta_1$ 으로 해서 [단계 2]로. 그렇지 않으면 $e_m^* = e_p(\beta_1 + \beta_2 / 2)$ 로 해서 정지

여기서, r 은 알고리즘의 정지를 위해 설정해주는 수렴기준값이고 u 는 탐색점의 양측의 오차값을 찾기위해 설정한 탐색폭이다.

[알고리즘 1' : 최대 방위오차 (e_m^*) 탐색]

알고리즘 1과 똑같은 과정이고 단지 e_p 와 e_m^* 를 e_n 과 e_n^* 로 대체함.

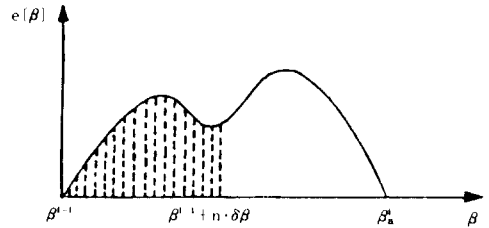


그림 5. 오차의 분포가 Multimodal인 경우
Fig. 5. Multimodal case of the deviation.

2) Multimodal인 경우의 최대오차 탐색

이 경우에는 효율적인 탐색방법은 없고 단순히 샘플링 구간을 설정해서 매회 마다의 오차를 계산해 계속 비교함으로써 최대오차를 찾는 직접 탐색방법 (direct search method)를 이용한다. 이 경우의 알고리즘은 다음과 같다.

[알고리즘 2 : 직접 탐색방법]

[단계 1] $N (>> 1)$ 선정. $\delta\beta = \beta_o^i - \beta^{i-1} / N$ 을 계산. $m = 0, e_1 = e_2 = 0$ 로 놓음.

[단계 2] 식(8), (9), (10)에 의해 $X(\beta^{i-1})$ 과 $X(\beta_o^i)$ 를 계산 $X_1 = X(\beta^{i-1}), X_2 = X(\beta_o^i)$ 로 놓음

[단계 3] $\beta = \beta^{i-1} + m \cdot \delta\beta$ 로 놓음. 만일 $m > N$ 면 $e_m^* = e_1, e_m^* = e_2$ 로 해서 정지. 그렇지않으면 $\lambda = \beta - \beta^{i-1} / \beta_o^i - \beta^{i-1}$ 로 하여 계속.

[단계 4] 식(1)~(7)에 의해 $e_p(\lambda)$ 와 $e_n(\lambda)$ 를 계산.

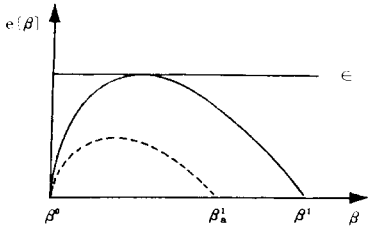
[단계 5] 만일 $e_1 < e_p(\lambda)$ 면, $e_1 = e_p(\lambda)$ 로 . 만일 $e_2 < e_n(\lambda)$ 면, $e_2 = e_n(\lambda)$ 로 놓음.

[단계 6] $m = m + 1$ 로 하여 [단계 3]으로.

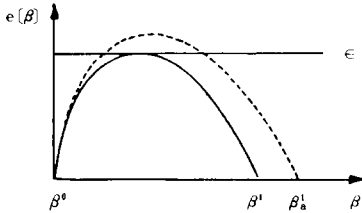
2. 최소의 경로점들 선택 알고리즘

그림 6의 예로 최소의 경로점 β^i 을 결정하는 과정을 살펴본다.

(a)와 같은 경우는 가상 경로점 (β_o^i) 구간내의 최대오



(a) $e_M(\beta'') < \epsilon$ 인 경우



(b) $e_M(\beta'') > \epsilon$ 인 경우

(e_M : 최대오차, ϵ : 허용오차, β'' : 첫번째 구간에서 설정한 가상 경로점)

그림 6. 허용오차 (ϵ)에 의해 설정되는 첫번째 경로점 (β').

Fig. 6. Initial intermediate point (β') determined within the tolerance deviation ϵ .

차 $e_M(\beta'')$ 가 허용오차 (ϵ) 보다 적으므로 그림과 같이 구간을 더 확장시킬 수 있다. (b)의 경우는 (a)의 경우와 반대로 구간을 축소시켜야 한다.

위의 내용을 위치오차와 방위오차를 함께 고려하면 다음의 3 가지 상태로 나눌 수 있다.

1. $e_M^p = \epsilon_p$, $e_M^r < \epsilon_r$ 이거나, $e_M^p < \epsilon_p$, $e_M^r = \epsilon_r$ 인 경우
2. $e_M^p < \epsilon_p$, $e_M^r < \epsilon_r$ 인 경우
3. $e_M^p > \epsilon_p$ 이거나 $e_M^r > \epsilon_r$ 인 경우

여기서, e_M^p : 최대 위치오차, e_M^r : 최대 방위오차
 ϵ_p : 허용 위치오차, ϵ_r : 허용 방위오차

1의 경우는 β'' 이 그대로 경로점 β' 으로 선정되고, 2의 경우는 forward 탐색방법을 이용하여 허용오차 내에서 구간을 최대 확장시켜 β' 을 선정한다. 또한 3의 경우는 backward 탐색방법을 이용하여 e_M^p 와 e_M^r 들이 허용오차 내에 존재하도록 구간을 축소하여 β' 을 선정한다.

첫 경로점 (β')을 선택한 후 같은 방법으로 나머지 경로점들도 모두 구할 수가 있다. 결과적으로 각 경로점들 사이의 구간은 허용오차 내에서 최대이므로 전체 경로점들의 수는 최소가 될 수 있다.

매 경로점 결정을 위한 가상 경로점들은 다음과 같

이 지정된다.

$$\beta''_o = K (\ll 1) \tag{17}$$

$$\beta''_i = 2\beta^{i-1} - \beta^{i-2}, i > 2 \tag{18}$$

forward 탐색방법과 backward 탐색방법의 알고리즘은 다음과 같다.

[알고리즘 3 : Forward 탐색방법]

[단계 1] $\delta\beta, v1, v2 (\ll 1)$ 를 선정. 알고리즘 1과 1' (또는 알고리즘 2)에 의해 Δe_M^p 와 Δe_M^r 계산

$$\begin{cases} \Delta e_M^p = e_M^p(\beta''_o + \delta\beta) - e_M^p(\beta''_o), \\ \Delta e_M^r = e_M^r(\beta''_o + \delta\beta) - e_M^r(\beta''_o). \end{cases}$$

[단계 2] $\beta''_o = \beta''_o + N \cdot \delta\beta$ 로 놓음

$$\text{여기서 } N = \max\left(\frac{\epsilon_p - e_M^p(\beta''_o)}{\Delta e_M^p}, \frac{\epsilon_r - e_M^r(\beta''_o)}{\Delta e_M^r}\right)$$

으로 결정됨.

[단계 3] 알고리즘 1과 1' (또는 알고리즘 2)에 의해 $e_M^p(\beta''_o)$ 와 $e_M^r(\beta''_o)$ 계산

만일 $e_M^p(\beta''_o) > \epsilon_p$ 혹은 $e_M^r(\beta''_o) > \epsilon_r$ 이면, [단계 5]로 그렇지 않으면 계속.

[단계 4] 만일 $|e_M^p(\beta''_o) - \epsilon_p| < v1$ 혹은 $|e_M^r(\beta''_o) - \epsilon_r| < v2$ 이면, $\beta' = \beta''_o$ 로 해서 정지. 그렇지 않으면 $\beta''_o = \beta''_o$ 로 하여 [단계 2]로.

[단계 5] $\beta''_o = (\beta''_o + \beta''_o)/2$. 알고리즘 1과 1' (또는 알고리즘 2)를 이용하여 $e_M^p(\beta''_o)$ 와 $e_M^r(\beta''_o)$ 을 계산.

[단계 6] 만일 $e_M^p(\beta''_o) > \epsilon_p$ 혹은 $e_M^r(\beta''_o) > \epsilon_r$ 이면, $\beta''_o = \beta''_o$ 로 해서 [단계 5]를 반복. 그렇지 않으면 [단계 7]로.

[단계 7] 만일 $|e_M^p(\beta''_o) - \epsilon_p| < v1$ 혹은 $|e_M^r(\beta''_o) - \epsilon_r| < v2$ 이면, $\beta' = \beta''_o$ 로 해서 정지. 그렇지 않으면 $\beta''_o = \beta''_o$ 로 해서 [단계 5]로.

여기서 $v1, v2$ 는 알고리즘의 정지를 위해 설정한 수렴 기준값이다.

[알고리즘 4 : Backward 탐색방법]

아래 사항을 제외하고는 알고리즘 3과 같다.

i) [단계 1]에서 $\Delta e_M^p = e_M^p(\beta''_o) - e_M^p(\beta''_o - \delta\beta)$ 이고 $\Delta e_M^r = e_M^r(\beta''_o) - e_M^r(\beta''_o - \delta\beta)$ 로 결정된다.

ii) [단계 2]에서 $\beta''_o = \beta''_o - N \cdot \delta\beta$ 이고 N 은 다음과 같이 결정된다.

$$N = \begin{cases} \frac{e_M^p(\beta''_o) - \epsilon_p}{\Delta e_M^p} \triangleq N1, \text{ 만일 } e_M^p > \epsilon_p, \\ e_M^r \leq \epsilon_r \text{ 이면} \\ \frac{e_M^r(\beta''_o) - \epsilon_r}{\Delta e_M^r} \triangleq N2, \text{ 만일 } e_M^r \leq \epsilon_r, \\ e_M^p > \epsilon_p \text{ 이면} \\ \max(N1, N2), \text{ 만일 } e_M^p > \epsilon_p, \\ e_M^r > \epsilon_r \text{ 이면} \end{cases}$$

- iii) [단계 3]에서 만일 $e_M^p(\beta_o^p) < \epsilon_p$, $e_M^r(\beta_o^r) < \epsilon_r$ 이면, [단계 4]로. 그렇지 않으면 $\beta_o = \beta_o^p$ 로 해서 [단계 2]로.
- iv) [단계 4]에서 "단계 2"를 "계속"으로 대체한다.
- v) [단계 5]에서 " $\beta_o = \beta_o^p$ "을 " $\beta_o^p = \beta_o^p$ "으로 대체한다.
- vi) [단계 6]에서 " $\beta_o = \beta_o^p$ "을 " $\beta_o = \beta_o^p$ "으로 대체한다.

이상의 알고리즘을 종합하여 최소의 경로점들을 선택하는 전체 알고리즘을 나타내면 다음과 같다.

[알고리즘 5 : 최소의 경로점들 선택 알고리즘]

- [단계 1] $K, \epsilon_p, \epsilon_r$ 선택. $L = 0, \beta_o = K$ 로 놓음.
- [단계 2] 만일 $\beta_o \geq 1$ 이면, $L = 1, \beta_o^p = 1$ 로 하여 [단계 3]으로.
- [단계 3] 알고리즘 1과 1' (또는 알고리즘 2)에 의해 e_M^p 와 e_M^r 을 계산.
- [단계 4] 만일 $e_M^p = \epsilon_p, e_M^r \leq \epsilon_r$ 이거나 $e_M^p \leq \epsilon_p, e_M^r = \epsilon_r$ 이면, $\beta^1 = \beta_o$ 로 하여 [단계 5]로. 만일 $e_M^p < \epsilon_p, e_M^r < \epsilon_r$ 이면, [단계 6]으로. 그렇지 않으면 [단계 7]로.
- [단계 5] 만일 $L = 1$ 이면 정지. 그렇지 않으면 [단계 8]로.
- [단계 6] 만일 $L = 1$ 이면 정지. 그렇지 않으면 forward 탐색방법을 이용해서 β^1 를 계산하여 [단계 8]로.
- [단계 7] Backward 탐색방법을 이용하여 β^1 를 계산.
- [단계 8] 만일 $L = 1$ 이면, $\beta_o = 1$ 로 해서 [단계 3]으로. 그렇지 않으면 $\beta_o^p = 2\beta^{1-1} - \beta^{1-2}$ 로 하여 [단계 2]로.

IV. 컴퓨터 시뮬레이션 및 결과

시뮬레이션에 이용된 PUMA 매니플레이터의 제원은 그림 7과 표 1에 설명된다. 지정해준 카르테시안 직선경로의 출발점(X_s)과 목표점(X_d) 그리고 허용오차는 다음과 같다.

$$X_s = \begin{bmatrix} 1 & 0 & 0 & 10 \\ 0 & \cos 0^\circ & -\sin 0^\circ & 10 \\ 0 & \sin 0^\circ & \cos 0^\circ & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$X_d = \begin{bmatrix} 1 & 0 & 0 & 40 \\ 0 & \cos 45^\circ & -\sin 45^\circ & 40 \\ 0 & \sin 45^\circ & \cos 45^\circ & 40 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\epsilon = 0.4\text{cm}, \epsilon_r = 0.01\text{rad}$.

상기 자료를 이용하여 Taylor 방식과 본 논문의 방

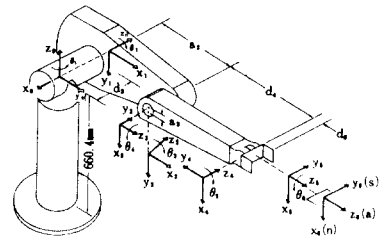


그림 7. PUMA 매니플레이터
Fig. 7. PUMA manipulator.

표 1. PUMA 매니플레이터의 링크 파라메타
Table 1. Link parameter of the PUMA manipulator.

Joint	θ_i	α_i	a_i	d_i	Joint range
1	90	-90	0	0	-160to+160
2	0	0	431.8mm	149.09mm	-- 225to45
3	90	90	-20.32mm	0	-- 45to225
4	0	-90	0	433.07mm	110to170
5	0	90	0	0	-- 100to100
6	0	0	0	56.25mm	-- 266to266

법에 의해 구해진 각각의 경로점들과 그에 대한 오차의 분포는 표 2와 그림 8 그리고 그림 9에 나타난다.

이상의 시뮬레이션 결과와 같이 Taylor 방식에 비하여 본 논문에서 제시한 알고리즘이 같은 허용 오차 내에서 최소의 경로점들을 선택함을 알 수 있다. 다만, 제안된 알고리즘 수행시간이 Taylor 방식보다 길지만 off-line 시스템 측면에서는 큰 차이가 없으리라 생각된다.

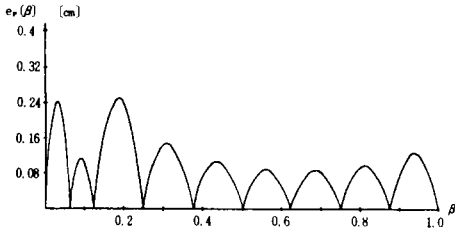
V. 결 론

주어진 카르테시안 직선경로를 조인트 공간상에서 선형적으로 보간 시켜줄 경우 이미 언급한 바와 같이 최소의 경로점들을 선택해 주는 것은 매우 중요

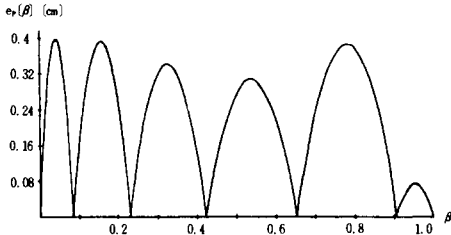
표 2. Taylor의 BDP와 제안된 방식에 의해 결정된 경로점들

Table 2. Intermediate points determined by Taylor's BDP and the proposed method.

	β^1	β^2	β^3	β^4	β^5	β^6	β^7	β^8
Taylor	0.0625	0.125	0.25	0.375	0.5	0.625	0.75	0.875
Alg. 5	0.08599	0.22906	0.42213	0.65238	0.90663			



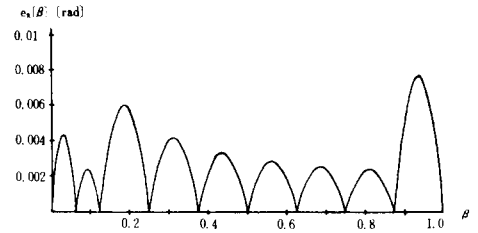
(a) Taylor



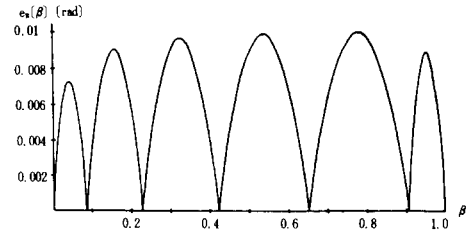
(b) Alg5

그림 8. $e_p(\beta)$ 의 분포도

Fig. 8. The derived positional deviation.



(a) Taylor



(b) Alg5

그림 9. $e_r(\beta)$ 의 분포도

Fig. 9. The derived orientational deviation.

하며, 본 논문이 제시한 방법이 타당함은 시뮬레이션을 통하여 입증 되었다. 이 방식은 3차 보간함수를 이용하는 경우에 대해서도 확장시킬 수 있을 것이다.

앞으로의 연구분야는 로봇트 경로 연구분야에서 가장 큰 제약조건이 되고있는 singular point 혹은 그 근처에서의 알고리즘의 수렴을 보장해 주는 것이 되겠다.

參 考 文 獻

- [1] M. Brady, "Trajectory Planning," in Robot Motion: Planning and control (M. Brady, et al., Eds.), Cambridge, Ma: MIT Press, Chapter 4, 1982.
- [2] R.A. Finkel, "Constructing and Debugging Manipulator Programs," Ph. D, thesis, Stanford University, 1976.
- [3] B.S. Gottfried, "Introduction to Optimization Theory," (B.S. Gottfried, et al.). Englewood Cliffs, N.J: Plentice-Hall, pp. 68-73, 1973.
- [4] C.S. Lin, P.R. Chang, and J.Y.S. Luh "Formulation and optimization of cubic polynomial joint trajectories for industrial robots." *IEE Trans. Auto. Contr.*, vol. AC-28, no. 3, pp. 1066-1074, Dec. 1983.
- [5] J.Y.S. Luh, M. W. Walker, and R.P.C. Paul, "On-line computational scheme for mechanical manipulators," *ASME Trans. J. Dyn. Syst. Measurement Contr.*, 102 (2): 69-76, 1980.
- [6] R.P.C. Paul, *Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm.*, Ph.D. thesis, Stanford University, 1972.
- [7] R.P.C. Paul, "Manipulator Cartesian path control," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-9, pp. 702-711, Nov. 1979.
- [8] R.P.C. Paul, B. Shimano, and G.E. Mayer, "Kinematic control equations for simple manipulators," *IEEE Trans. Syst. Man Cybern.*, SMC-11(6): 449-455, 1981.
- [9] D.L. Pieper, "The Kinematics of Manipulators Under Computer Control," Ph.D. thesis, Stanford University, 1968.
- [10] P.M. Prenter, "Splines and Variational Methods," New York: Wiley, 1975.
- [11] R.H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM J. Res. and Develop.*, 23(4): 424-

- 436, 1979.
- [12] D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses." *IEEE Trans. Man-Mach. Syst.*, MMS-10(2):47-53, 1969.
- [13] D.E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators.", *ASME Trans. J. Dyn. Measurement Contr.*, 94(4): 303-309, 1972.
-