

Bezier Spline을 이용한 용접 로봇의 새로운 Weaving Motion 궤적 생성 알고리즘

정원지*, 김대영⁺, 서영교⁺⁺, 홍형표⁺⁺, 홍대선⁺⁺⁺

(논문접수일 2003. 11. 21, 심사완료일 2004. 2. 5)

A New Planning Algorithm of Weaving Trajectory Using Bezier Spline for A Welding Robot

Won-Jee Chung*, Dae-Young Kim⁺, Young-Kyo Seo⁺⁺, Hyung-Pyo Hong⁺⁺, Dae-Sun Hong⁺⁺⁺

Abstract

In this paper, we propose a new weaving trajectory algorithm for the arc welding of a articulated manipulator. The algorithm uses the theory of Bezier spline. We make a comparison between the conventional algorithms using Catmull-Rom curve and the new algorithms using Bezier spline. The proposed algorithm has been evaluated based on the MATLAB environment in order to illustrate its good performance. Through simulations, the proposed algorithm can result in high-speed and flexible weaving trajectory planning so that it's trajectory cannot penetrate into a base metal compared to the conventional algorithm using Catmull-Rom curve.

Key Words : 6-axis Articulated Manipulator(6축 수직 다관절 로봇), Weaving Trajectory(위빙 궤적), Simple Weaving(직선 위빙), Triangular Weaving(삼각 위빙), Bezier Spline, Catmull-Rom Curve, Base metal(모재)

1. 서론

최근 숙련된 용접 기능사의 부족, 열악한 작업 환경의 기피, 고임금화로 인하여 용접 로봇을 도입하는 사례가 많

다. 단순 용접 뿐만 아니라 복잡한 용접에서도 로봇을 많이 사용하고 있다. 따라서 용접 로봇의 다양화와 고기능화가 더욱 더 요구되고 있다. 특히 로봇의 위빙 동작은 유연한 로봇 동작의 대표적인 예로서 숙련된 사람이 용접 할

* 창원대학교 기계설계공학과 (wchung@changwon.ac.kr)

주소: 641-773 경남 창원시 사림동 9번지

+ 창원대학교 기계설계공학과 대학원

++ 두산메카텍(주) 연구개발부

+++ 창원대학교 기계설계공학과

때와 같은 궤적을 생성해야 한다. 용접 로봇에 새로운 위빙 동작을 적용함으로써 여러 가지 용접이 가능하고, 생산성을 높일 수 있으며 용접 품질을 향상시킬 수 있다. 본 연구는 6축 수직다관절 로봇의 용접 시 위빙 동작에 관한 것이다. 기존 DRC/DR6(두산메카텍(주)의 산업용 로봇 명칭)에 사용된 Catmull-Rom Curve를 이용한 위빙 동작 알고리즘을 보완하기 위해 Bezier Spline⁽¹⁾을 이용한 새로운 알고리즘을 제안한다. 연구수행에 있어서는 MATLAB⁽²⁾에 의한 시뮬레이션을 통해 실제 로봇에 적용시킨다.

2. 새로운 위빙 운동 알고리즘

Bezier Spline을 이용한 새로운 위빙운동은 직선 위빙 운동과 삼각 위빙 운동으로 나누어진다.

2.1 직선 위빙 운동 알고리즘

Fig. 1은 직선 위빙 운동 알고리즘의 순서도이다. 직선 위빙 운동 알고리즘에서 시점 $V_1(x_1, y_1, z_1)$, 종점 $V_{n+1}(x_{n+1}, y_{n+1}, z_{n+1})$, 참조점 $Ref(x_R, y_R, z_R)$, d (진폭), q (smooth상수), n (위빙 횟수)은 사용자가 입력하는 값이다. Fig. 2와 같이 먼저 시점과 종점을 연결한 \vec{B} 의 단위 벡터인 \vec{u} 를 구한다. \vec{u} 를 이용하여 시점과 종점을 n 등분하는 점 V_i ($i=1, 2, 3, \dots, n+1$)를 구한다. V_i, V_{i+1} 을 이용하여 V_i, V_{i+1} 의 중간점 M_i ($i=1, 2, 3, \dots, n$)를 구하고, M_i 와 $\pm q$ 을 이용하여 P_i ($i=1, 2, 3, \dots, 2n$)를 구한다. 또한, H좌표를 이용하여 \vec{u} 에 수직이면서

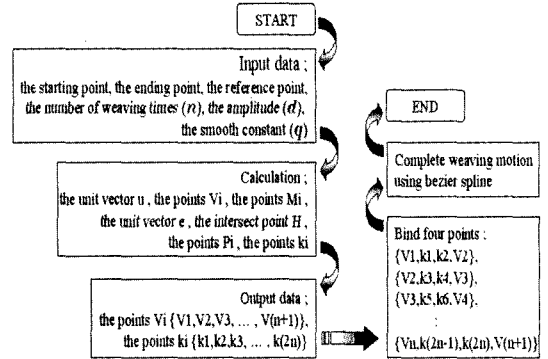
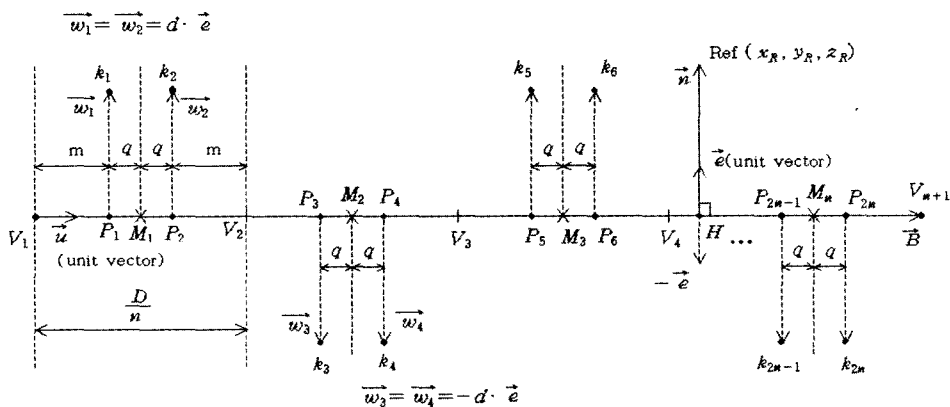


Fig. 1. Simple weaving motion flow chart

참조점을 지나는 단위벡터 \vec{e} 를 구한 후 \vec{e}, d, P_i 를 이용하여 k_i ($i=1, 2, 3, \dots, 2n$)를 구한다.

시점과 종점을 n (n 은 정수) 등분한 첫 번째 구간의 Bezier Spline 조정점들은 (V_1, k_1, k_2, V_2) 이 된다. 즉, 시점 (V_1), 진폭과 Smooth 정도에 결정되어지는 \vec{e} 방향의 두 점 (k_1, k_2), n 등분된 점들 중 두 번째 점 (V_2)으로 이루어진다. 첫 번째 구간의 위빙 방향은 \vec{e} 방향과 동일하며 그 반대로 가능하다. 두 번째 구간의 Bezier Spline 조정점들 (V_2, k_3, k_4, V_3) 으로 이루어지며, 위빙 방향은 앞 구간의 위빙 방향과 반대가 된다. 이렇게 반복하여 n 번째 구간까지 구한 Bezier Spline을 연결하여 직선 위빙 운동을 완성한다.



d : weaving amplitude, q : smooth constant

Fig. 2 Simple weaving motion

2.2 삼각 위빙 운동 알고리즘

Fig. 3은 삼각 위빙 운동 알고리즘의 순서도이다. Fig. 4의 삼각 위빙 운동에서 $a_1, b_1, c_1, d (=c_n), c$ (smooth 상수), step은 사용자가 입력하는 값이다.

$$\vec{u} = \left(\frac{x_{n+1} - x_1}{D}, \frac{y_{n+1} - y_1}{D}, \frac{z_{n+1} - z_1}{D} \right) \quad (1)$$

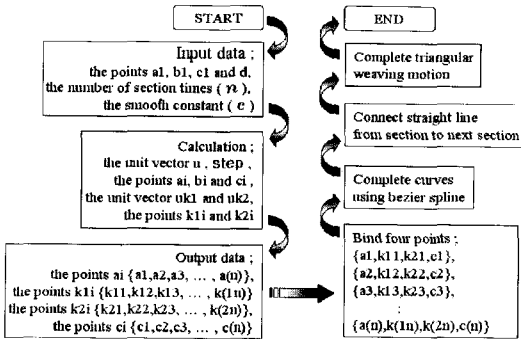


Fig. 3 Triangular weaving motion flow chart

$$D = \sqrt{(x_{n+1} - x_1)^2 + (y_{n+1} - y_1)^2 + (z_{n+1} - z_1)^2} \quad (2)$$

식 (1)과 같이 먼저 c_1, d 를 연결한 벡터의 단위벡터를 구하고, 식 (1)의 단위벡터를 이용하여 $(a_1, a_2, \dots, a_n), (b_1, b_2, \dots, b_n), (c_1, c_2, \dots, c_n)$ 을 구한다.

$$\vec{u}_{k1} = \left(\frac{x_{a1} - x_{b1}}{D_1}, \frac{y_{a1} - y_{b1}}{D_1}, \frac{z_{a1} - z_{b1}}{D_1} \right) \quad (2)$$

$$D_1 = \sqrt{(x_{a1} - x_{b1})^2 + (y_{a1} - y_{b1})^2 + (z_{a1} - z_{b1})^2}$$

$$k_{1i} = b_i + \vec{u}_{k1} \times c \quad (i=1, 2, 3, \dots, n) \quad (3)$$

$$\vec{u}_{k2} = \left(\frac{x_{c1} - x_{b1}}{D_2}, \frac{y_{c1} - y_{b1}}{D_2}, \frac{z_{c1} - z_{b1}}{D_2} \right) \quad (4)$$

$$D_2 = \sqrt{(x_{c1} - x_{b1})^2 + (y_{c1} - y_{b1})^2 + (z_{c1} - z_{b1})^2}$$

$$k_{2i} = b_i + \vec{u}_{k2} \times c \quad (i=1, 2, 3, \dots, n) \quad (5)$$

그리고, 식 (2), 식 (4)에서 구한 단위벡터와 Bezier Spline의 Smooth 정도에 따라 (b_1, b_2, \dots, b_n) 으로 부터 식 (3), 식 (5)를 이용하여 $(k_{11}, k_{12}, \dots, k_{1n}), (k_{21}, k_{22}, \dots, k_{2n})$ 을 구한다. 여기서 구한 점들은 모두 Bezier Spline의 조정

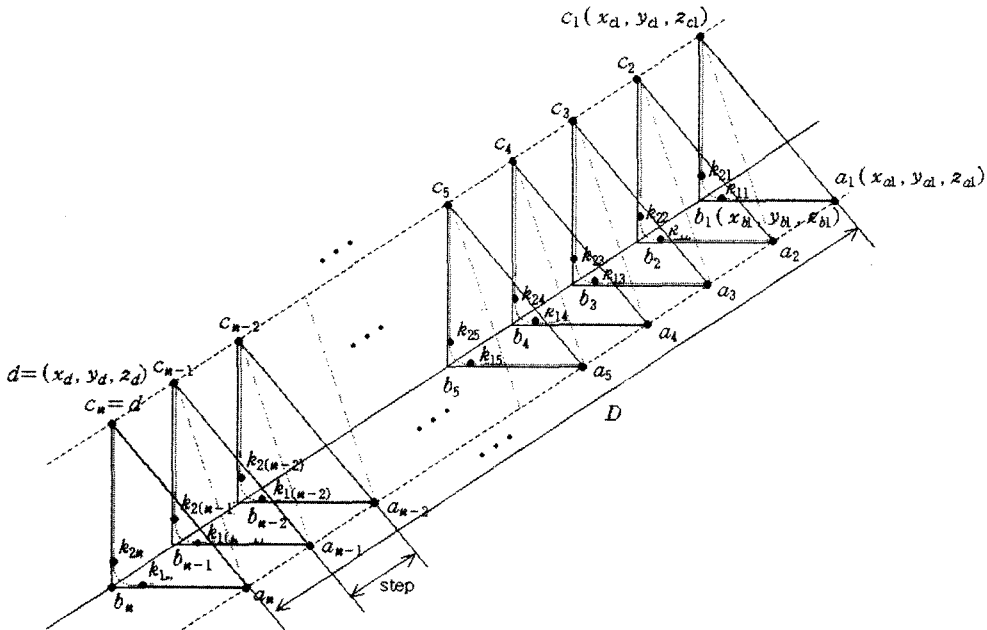


Fig. 4 Triangular weaving motion

점들이 된다. 첫 번째 구간의 Bezier Spline 조정점들은 $(a_1, k_{11}, k_{21}, c_1)$ 이 되고, 두 번째 곡선의 조정점들은 $(a_2, k_{12}, k_{22}, c_2)$ 가 된다. 첫 번째 곡선과 두 번째 곡선은 c_1 과 a_2 를 이용한 직선 (3)으로 연결된다. 이렇게 n 번째 구간까지 반복하여 삼각 위빙 운동을 완성한다.

3. 기존 알고리즘과의 비교

기존의 Catmull-Rom 곡선^(4,5)을 이용한 알고리즘과 새롭게 제안한 알고리즘을 비교하였다.

Catmull-Rom 곡선에서 $(n+1)$ 개의 경유점이 주어졌을 때, i 번째 구간의 생성곡선 $C_i(t)$ 를 행렬로 나타내면 다음과 같다.

$$C_i(t) = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -c & 2-c & c-2 & c \\ 2c & c-3 & 3-2c & -c \\ -c & 0 & c & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}$$

여기서, c 는 곡선과 경유점을 지나는 직선과의 근접정도를 나타내며 Fig. 5에서처럼 c 값이 작을수록 직선에 근접한 곡선이 된다.

3.1 직선 위빙 운동

직선 위빙 운동에서 기존의 Catmull-Rom Curve를 이용한 알고리즘은 c (단, $0 \leq c = 1$)라는 상수를 통해 곡선의 부드러운 정도를 조절할 수 있지만, 한 구간의 곡선을 얻기 위해서는 4개의 조정점을 이용한 2번의 계산이 필요하다. 반면에 Bezier Spline을 이용한 알고리즘은 조정점에 의해 곡선의 부드러운 정도를 제어할 수 있고, 4개의 조정점을 이용한 1번의 계산으로 한 구간의 곡선을 얻을 수 있다. 따라서, 계산량이 절반으로 줄어들어 실제 로봇 구현에 있어

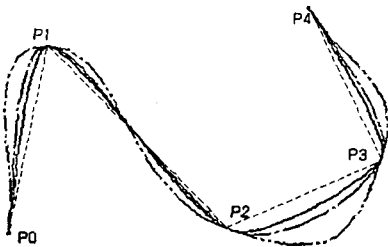


Fig. 5 Example of Catmull-Rom curve according to smoothness factor 'c'

서 용이하다.

3.2 삼각 위빙 운동

삼각 위빙 운동에서도 기존의 Catmull-Rom Curve을 이용한 알고리즘은 c (단, $0 \leq c = 1$)라는 상수를 통해 곡선의 부드러운 정도를 조정한다. 그러나, Fig. 6처럼 Smooth 상수 $c=0$ 인 경우 완전한 삼각형 용접 궤적을 나타내지만, 나머지 구간에서는 용접 궤적이 실제 용접 경로를 벗어

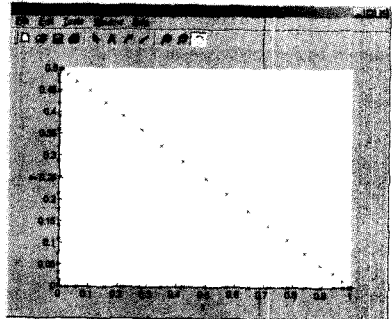
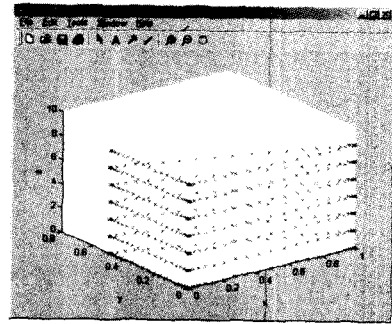
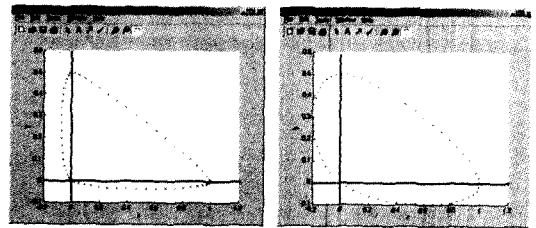
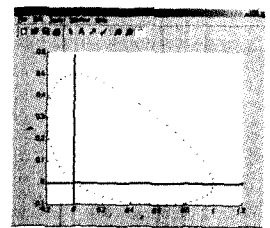


Fig. 6 Triangular weaving motion using Catmull-Rom curve at $c=0$



(a) $c=0.3$



(b) $c=0.8$

Fig. 7 Triangular weaving motion using Catmull-Rom curve

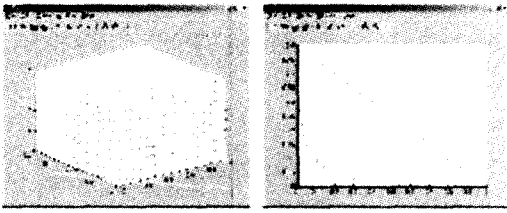


Fig. 8 Triangular weaving motion using Bezier Spline at $c=0.05$

나 용접하고자 하는 물체에 파고드는 단점이 있다. Fig. 7처럼 실제 용접경로는 실선이지만, 용접 궤적은 실제 용접 경로를 벗어나 모재를 침범한다. 또한, Smooth 상수 c 가 증가할수록 실제 용접경로에서 많이 벗어난다.

반면, Bezier Spline을 이용한 새로운 알고리즘은 Fig. 8처럼 용접 궤적이 모재를 침범하지 않는다.

또한, c (단, $0 \leq c = 1$)의 변화에 따라 삼각형 꼭지점 부근에서 곡선의 부드러운 정도를 조절할 수 있다.

4. 시뮬레이션

앞서 제안한 새로운 직선 위빙 운동 및 삼각 위빙 운동 알고리즘을 MATLAB에서 시뮬레이션 한 결과이다.

4.1 직선 위빙 운동

Fig. 9는 임의로 시점 (5,8,0), 종점 (54,37,0), 참조점 (45,23,0), 진폭 $d=2$, Smooth 상수 $q=0.4$, 위빙 횟수 $n=18$ 인 경우 MATLAB 시뮬레이션 결과이다.

Fig. 10은 임의의 네 개의 연속된 직선인 경우 위빙 하는 것이다. 이 때에는 각 직선간의 교점과 각 직선의 참조점만 추가해서 입력하면 된다. 특히, 각 직선의 위빙이 끝나는 부분과 그 다음 직선이 시작하는 부분은 항상 중첩 없이 부드럽게 연결되도록 하였다.

4.2 삼각 위빙 운동

Fig 8은 임의로 $a_1(1,0,0)$, $b_1(0,0,0)$, $c_1(0,0,5,0)$, $d(0,0,5,8)$, $c=0.05$, $step=1$ 인 값을 입력한 경우이다.

기존의 Catmull-Rom Curve를 이용한 알고리즘의 결과처럼 용접물을 파고드는 부분이 나타나지 않으며, Smooth 상수 c 로서 곡선의 부드러운 정도를 조절할 수 있다.

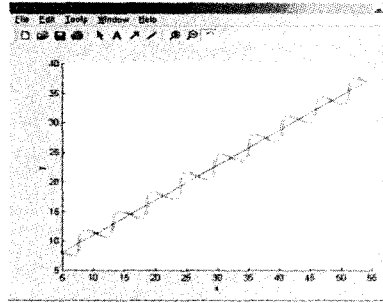
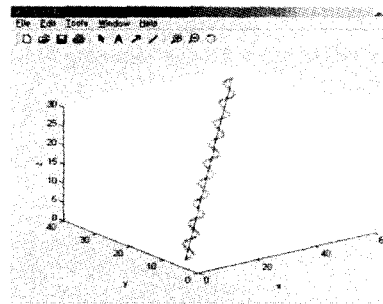


Fig. 9 Simple weaving motion result based on MATLAB

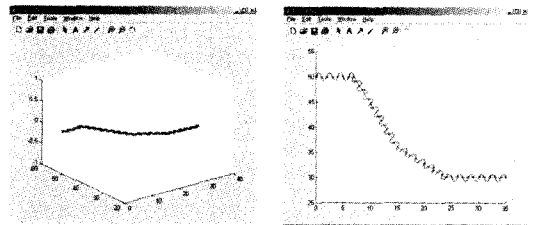


Fig. 10 Simple weaving motion of continuity line

5. 결론

본 논문에서는 수직다관절 매니퓰레이터의 아크 용접 시 위빙 운동에 관한 새로운 알고리즘을 제안하였다. MATLAB을 이용하여 제안된 Bezier Spline을 이용한 알고리즘과 기존의 Catmull-Rom 곡선을 이용한 알고리즘을 비교해 보았다. MATLAB으로 시뮬레이션 한 결과 계획되는 용접 궤적의 정확도나 계산량면에서 제안된 알고리즘이 더 우수함을 알 수 있었다. 특히, 삼각 위빙에서 모재를 침범하지 않는 정확한 용접궤적을 생성하면서도 계산량이 적어 처리시간(processing time)을 줄일 수 있을 것으로 사료되었고, 현

대 로봇에 요구되는 고속화 및 유연한 로봇 동작에 적합한 알고리즘임을 알 수 있었다.

후 기

본 연구는 한국과학재단 지정 창원대학교 공작기계기술 연구센터(RRC) 및 두산메카텍(주)의 지원에 의한 것입니다. 또한, 본 과제는 산업자원부의 출연금 등으로 수행한 지역전략산업 석·박사 연구인력 양성사업의 연구결과입니다.

참 고 문 헌

- (1) Lee, H. C., Chae, S. W., Choi, Y., 1996, *Computer Graphics and Geometric Modeling for Engineers*, Sigma Press, pp. 276~280.
- (2) Solc, F., 2000, "Matrix Model of Robot in

Matlab Simulink," *Lecture notes in computer science*, pp. 250~258.

- (3) Yi, S. H., Yang, S. H., and Park, Y. K., 2001, "Generation of Constant Orientation in Industrial Robot," *Transactions of the Korean Society of Machine Tool Engineers*, Vol. 10, No. 2, pp. 95~102.
- (4) Suh, I. H., Yuh, H. J., and Eum, K. S., 1993, "Weaving Generation Technology I," *Summer Workshop*, Research Center for New Technology of Control and Instrumentation.
- (5) Suh, I. H., Yuh, H. J., and Eum, K. S., 1994, "Weaving Generation Technology II," *Winter Workshop*, Research Center for New Technology of Control and Instrumentation.