

베지어 곡선을 이용한 로봇 축구 항법의 개선

정태영^a, 이귀형^{a*}

An Improvement of Navigation in Robot Soccer using Bezier Curve

Tae-Young Jung^a, Gui-Hyung Lee^{a*}^a Department of Mechanical System Design Engineering, Seoul National University of Science and Technology,
232, Gongneung-ro, Nowon-gu, Seoul 01811 Korea

ARTICLE INFO

Article history:

Received	21	September	2015
Revised	14	October	2015
Accepted	27	October	2015

Keywords:

Bezier curve
Robot soccer
Obstacle
Avoid

ABSTRACT

This paper suggests a new method for making a navigation path by using Bezier curve in order to improve the navigation performance used to avoid obstacles during a robot soccer game. We analyzed the advantages and disadvantages of both vector-field and limit-cycle navigation methods, which are the mostly widely used navigation methods for avoiding obstacles. To improve the disadvantages of these methods, we propose a new design technique for generating a more proper path using Bezier curve and describe its advantages. Using computer simulations and experiments, we compare the performance of vector-field navigation with that of Bezier curve navigation. The results prove that the navigation performance using Bezier curve is relatively superior to the other method.

1. 서론

항법이란 로봇이 주변 환경정보를 받아들여 자신이 이동해야 할 경로를 생성하고 생성된 경로를 따라 이동하도록 제어하는 일련의 행동을 일컫는다. 항법은 지도화, 경로생성 및 선택, 경로추종 등의 세 가지 작업으로 나눌 수 있다^[1]. 로봇의 항법에 대한 연구는 오랫동안 다양한 방법들이 제안되었다. 초기의 연구는 로봇 제어기의 연산능력의 한계 때문에 실제로 구현되기 어려운 점이 많았다. 따라서 많은 이론적인 연구에도 불구하고 실제로 구현되어 로봇에 이용된 항법 방식은 계산량이 적은 몇 가지뿐이었다. 그러나 현재에는 PC의 발전과 고성능의 마이크로컨트롤러의 등장으로 다양한 항법을 적용할 수 있게 되었다^[3]. 일반적으로 이동 로봇의 경로 설정은 초기 위치로부터 원하는 목적지까지 이동하는 작업의 수행이

가능하도록 연속적인 상태의 전이 방법들로 구성한다. 로봇의 현재 위치와 상태 변수들과 주변영역들에 대한 정보들이 지속적으로 탐색되어 현재 상태의 정보를 통해 경로설정을 수행해 나간다. 일반적으로 목적지까지의 최단경로는 단 하나만이 존재할 수 있지만, 단순히 이동을 위한 목적의 경로는 많은 수가 존재하며 다양한 알고리즘들에 의해서 이러한 경로들의 차이가 발생한다^[6,7]. 로봇 축구 경기에서 축구 로봇은 2.6 m/s 이상 고속으로 주행한다. 이처럼 빠른 주행으로 인해 상대팀 또는 자신의 팀의 로봇과 충돌하면 이동시간의 증가와 몸체의 고장 등 큰 손상을 입는다. 이로 인해 장애물 회피 알고리즘이 매우 중요시 되고 있다. 본 논문에서는 로봇 축구에서 기존의 장애물 회피 항법에 대하여 분석한 후, 문제점을 해결할 수 있는 방법으로 베지어 곡선을 이용한 장애물 회피 항법을 연구하여 제안하였다.

* Corresponding author. Tel.: +82-2-970-6325

Fax: +82-2-974-8270

E-mail address: ghlee@seoultech.ac.kr (Gui-Hyung Lee).

2. 장애물 회피 항법

로봇 축구에서 장애물 회피 항법으로는 벡터장 항법과 limit-cycle 항법이 많이 쓰이고 있다. 벡터장 항법은 통합 항법 방식 중의 한 방법으로, 가상 중력장 방식에서 발생하는 문제점들을 해결하고, 양방향 구동식 로봇에 적용 가능하도록 개발된 방법이다^{4,8)}. 이 방법은 빠른 로봇의 제어에 적합하며, 이동 장애물이 존재해도 충돌을 회피할 수 있다는 장점이 있다. 또한 로봇을 원하는 위치와 각도로 동시에 제어하여 수렴시킬 수 있기 때문에 로봇 축구에서 공을 치는 행동을 쉽게 구현할 수 있다. limit-cycle 항법은 벡터장 항법에 비하여 모든 영역의 위치에서 원하는 원으로 로봇을 움직이게 하는 필드를 간단한 식으로 표현할 수 있는 이점이 있다¹⁾. 또한 벡터장 항법에서 나타나는 로봇이 척력의 역작용으로 원하는 방향이 아닌 위쪽으로 이동하는 문제점을 해결할 수 있다⁵⁾.

2.1 벡터장 항법의 사용시 일반적인 문제점

Fig. 1에서는 로봇이 두개의 장애물을 회피하면서 볼까지 이동하는 경우를 보여준다²⁾. 로봇이 장애물과 근접했을 경우 경로의 심한 변화를 피할 수 없으며, 그림에서와 같이 로봇이 이동 중 4번의 급격한 변화가 발생한다. 이는 로봇 축구 경기에서 축구 로봇이 고속으로 주행할 때 장애물과 충돌할 수 있고 로봇의 관성 및 원심력에 의해 경로를 이탈할 수 있다.

Fig. 2(a)에서는 볼이 장애물의 offset영역 내부에 위치할 경우 단위 벡터의 방향이 offset경계 영역으로 발산하기 때문에 로봇이 볼까지 접근하지 못하는 것을 보여주며, Fig. 2(b)에서는 로봇이 장애물을 지나간 뒤에도 장애물에서 나오는 척력의 작용으로 인하여 로봇이 원하는 방향이 아닌 위쪽으로 가는 것을 보여 준다^{1,5)}.

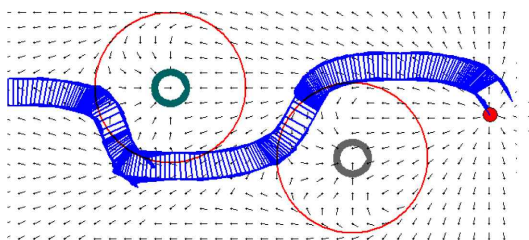


Fig. 1 Rapid change of path in vector field navigation

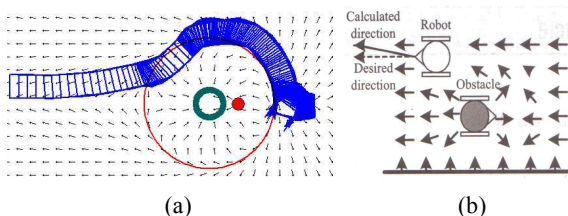


Fig. 2 Effect of repulsion in vector field navigation

로봇과 단위 벡터의 각도 차이가 크면, 로봇은 단위 벡터와 방향 각을 맞추기 위해 불필요한 이동을 한다. Fig. 3에서는 로봇이 단위 벡터와 방향을 맞추기 위해 불필요한 이동이 많이 발생한 것을 보여준다.

벽에 근접해 있을 경우 Fig. 4와 같이 로봇의 방향과 상관없이 볼 방향으로 로봇의 양 바퀴 속도를 다르게 주어 벽에서 탈출한다¹⁾. 그러나 Fig. 5처럼 로봇이 경기장 모서리에 근접해 있을 경우 boundary 영역이 중복(왼쪽 벽, 위쪽 벽)된다. Fig. 5의 첫 번째 그림에서 로봇이 왼쪽 벽 boundary 영역에 속하므로(또는 위쪽 벽) clockwise 방향으로 회전하여 빠져 나오면 다시 위쪽 boundary 영

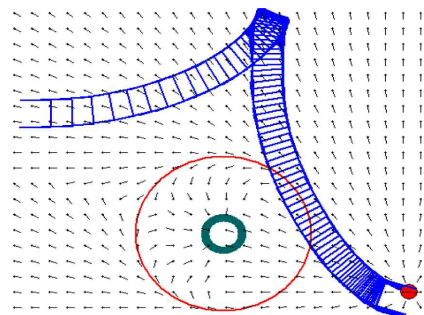


Fig. 3 Generation of unnecessary path

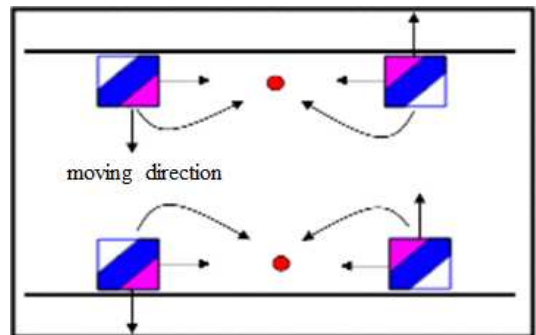


Fig. 4 Difficulty of avoiding boundary wall

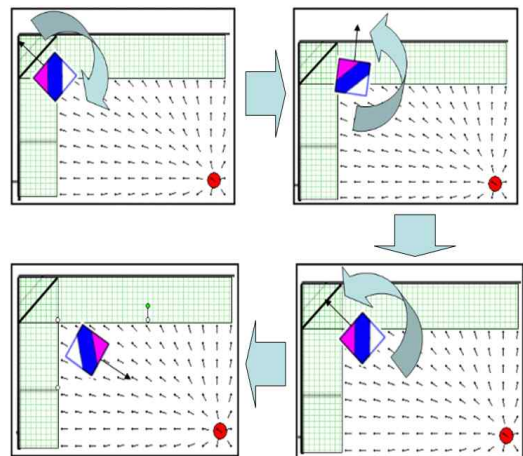


Fig. 5 Difficulty of avoiding boundary corner

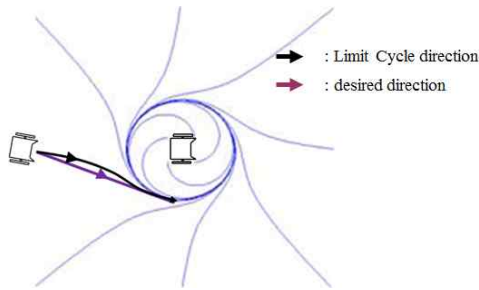


Fig. 6 Improper path planning in limit-cycle navigation

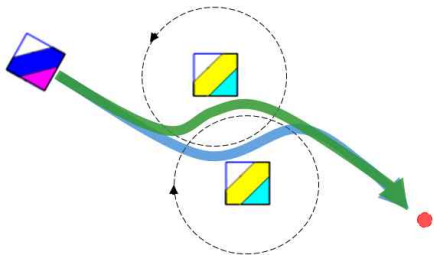


Fig. 7 Collision of robot and obstacles in limit-cycle navigation

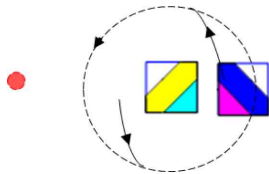


Fig. 8 Collision of robot and obstacles in limit-cycle navigation

역에 속하게 된다. 이후 로봇은 다시 counterclockwise 방향으로 회전하여 boundary 영역을 완전히 벗어나 벡터장 방향으로 볼까지 이동하게 되는데, 이처럼 벽에서 벗어날 때 다소간 어려움이 발생한다.

2.2 Limit Cycle 항법 사용 시 일반적인 문제점

limit cycle 항법을 사용할 경우, 로봇이 limit cycle의 이동경로에 접근 시 두 번의 방향전환이 일어난다 (Fig. 6). 또한, 로봇이 장애물과 매우 근접해 있을 경우, limit cycle을 따라 이동하는데 Fig. 7과 Fig. 8과 같이 장애물과 충돌할 수 있다.

3. 베지어 곡선을 이용한 경로 설계

회피하려는 장애물의 중심을 원점으로 잡고 안전한 회피영역의 양끝점인 A, B를 구하기 위해 Fig. 9(a)에서 TA, TB (tentative A, B) 위치를 정한다. 여기서 TA, TB의 길이는 각 장애물의 A, B의 길이와 같다. TA, TB의 좌표를 회전시키기 위해 Fig. 9(b)에서 로봇의 중심부터 장애물의 중심까지 선을 이어 ϕ 를 구하고, π 에서 ϕ 를 뺀 θ 값을 구한다.

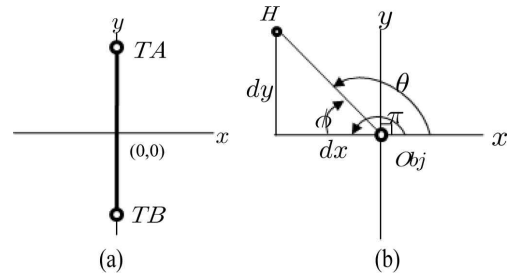


Fig. 9 Decision of degree of A, B

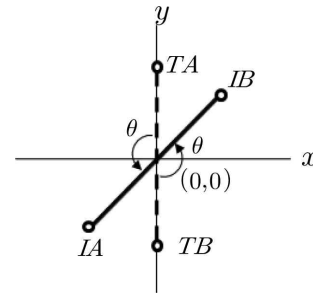


Fig. 10 Decision of degree of A, B

H : 로봇의 중심점 좌표 (H_x, H_y)

Obj : 장애물의 중심점 좌표 (Obj_x, Obj_y)

$$dx = Obj_x - H_x$$

$$dy = Obj_y - H_y$$

$$\phi = \tan^{-1}(dy / dx)$$

$$\theta = \pi - \phi$$

(1)

Fig. 9에서 TA, TB의 위치를 변환하기 위해 식 (1)에서 구한 θ 만큼 좌표를 회전시켜 각 장애물에서 A, B의 좌표를 구하기 위한 초기위치 IA, IB (initial A, B)의 좌표를 Fig. 10과 같이 구한다.

$$IA_x = TA_x \cos \theta - TA_y \sin \theta$$

$$IB_x = TB_x \cos \theta - TB_y \sin \theta$$

$$IA_y = TA_x \sin \theta + TA_y \cos \theta$$

$$IB_y = TB_x \sin \theta + TB_y \cos \theta$$

(2)

여기서 TA_x, TB_x 가 0이므로

$$IA_x = -TA_y \sin \theta, \quad IB_x = -TB_y \sin \theta$$

$$IA_y = TA_x \cos \theta, \quad IB_y = TB_x \cos \theta$$

(3)

이 된다.

원점 (0,0)에서 좌표 변환식을 이용하여 구한 IA, IB의 좌표를 장애물의 중심좌표에 더해 최종 A, B의 좌표를 Fig. 11과 같이 구한다.

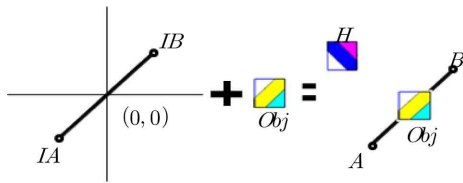


Fig. 11 Decision of degree of A, B

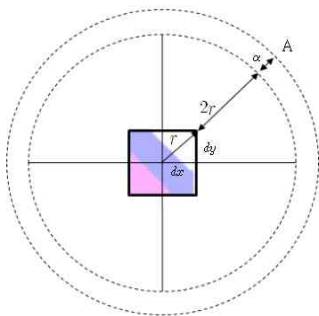


Fig. 12 Decision of safe area

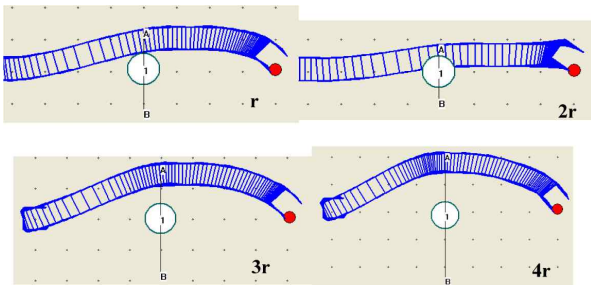


Fig. 13 Decision of safe area

H : 로봇

Obj : 장애물

$$A_x = IA_x + Obj_x, \quad B_x = IB_x + Obj_x$$

$$A_y = IA_y + Obj_y, \quad B_y = IB_y + Obj_y$$

(4)

로봇 이동시 장애물과 충돌하지 않기 위한 A, B의 길이는 장애물의 중심에서 다음 식 (5)를 이용하여 결정한다. Fig. 12에서처럼 A, B의 길이는 같기 때문에 A의 길이만 구한다.

$$r = \sqrt{dx^2 + dy^2}$$

$$A = 3r + \alpha$$

(5)

Fig. 13은 장애물 회피를 위한 A의 길이와 장애물의 반경 r사이의 관계를 정하기 위한 시뮬레이션이다. r값이 작으면 로봇의 이동 궤적이 부드러우나 장애물과 충돌할 가능성이 커지고 r의 값이 크면 장애물과 충돌을 하지 않으나, 로봇의 이동 궤적이 큰 것을 볼 수 있다. 본 논문에서는 로봇과 장애물의 충돌을 피하기 위해서 A와 B의 길이를 적절하게 3r로 정하여 베지어 곡선 항법에 적용하

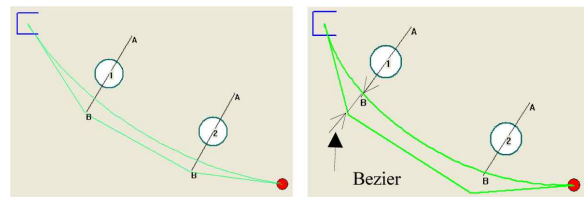


Fig. 14 Decision of offset point of Bezier curve-1

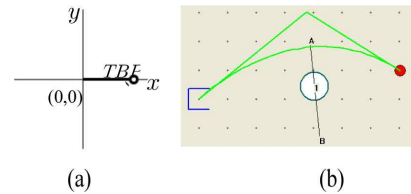


Fig. 15 Decision of offset point of Bezier curve-2

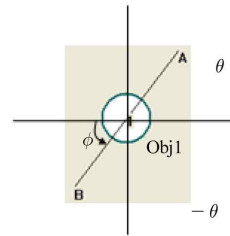


Fig. 16 Decision of offset point of Bezier curve-3

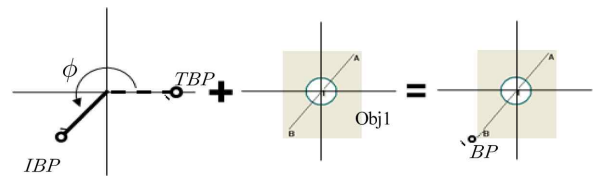


Fig. 17 Decision of offset point of Bezier curve-4

였다.

각 장애물의 A 또는 B에 베지어 컨트롤 포인트를 생성하면 베지어 곡선이 장애물의 안쪽에(Fig. 14) 만들어진다. 로봇과 장애물간의 충돌 방지를 위해 A, 또는 B에 Bezier offset을 주어 안전영역을 확보한다.

Fig. 15에서 Fig. 17까지는 Fig. 14의 Bezier offset point를 구하는 과정을 보여준다. 먼저 Fig. 15(a)에서 (0,0)의 좌표에 TBP(tentative Bezier point)의 위치를 정한다. 여기서 TBP의 길이는 Fig. 14에서 Bezier offset의 길이와 같은데, TBP의 길이에 따라 베지어 곡선의 휨 정도가 달라진다. 본 논문에서는 TBP의 길이를 130 pixel로 정하였으며, Fig. 15(b)에서는 TBP의 길이를 130 pixel로 정하였을 때, 베지어 곡선을 생성한 것을 보여준다.

Fig. 16에서 장애물의 중심에서 B 사이의 theta각도를 구하고, B가 3사분면에 있으므로 theta+pi를 하여 좌표변환을 위한 phi값을 구한다.

$$\theta = \tan^{-1}((Obj1_y - B_y)/(Obj1_x - B_x))$$

$$\phi = \theta + \pi \tag{6}$$

TBP의 좌표를 식 (6)에서 구한 ϕ 만큼 회전시켜 IBP(initial Bezier point)의 좌표를 구하고, 장애물의 중심좌표를 더해 최종 BP(Bezier point)의 좌표를 구한다.

$$IBP_x = TBP_x \cos \phi - TBP_y \sin \phi$$

$$IBP_y = TBP_x \sin \phi + TBP_y \cos \phi \tag{7}$$

식 (7)에서 TBP_y 가 0이므로

$$IBP_x = TBP_x \cos \phi, \quad IBP_y = TBP_x \sin \phi$$

$$BP_x = IBP_x + Obj1_x$$

$$BP_y = IBP_y + Obj1_y \tag{8}$$

이 된다.

Fig. 18부터 Fig. 20까지는 로봇이 볼까지 이동 중 필요 없는

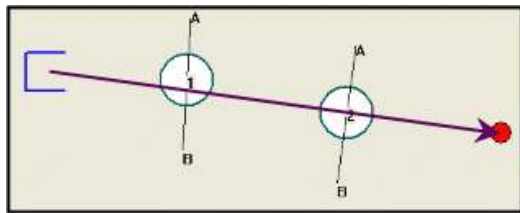


Fig. 18 Path planning-1

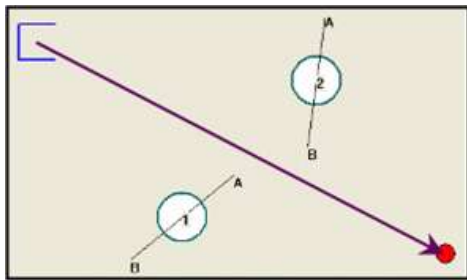


Fig. 19 Path planning-2

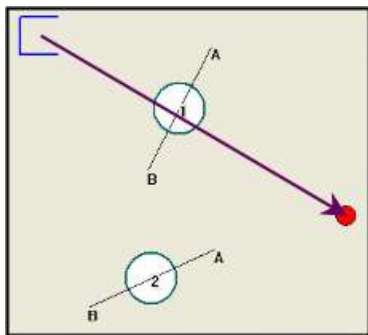


Fig. 20 Path planning-3

경로를 제거하는 방법을 보여준다. 불필요한 경로를 생성하지 않기 위하여 우선 로봇의 중심에서 볼의 중심까지 선을 잇는다. 만약 각 장애물의 \overline{AB} 와 교차점이 생기면 해당 장애물에 대해서 경로를 생성한다. 교차점이 하나만 생기면 하나의 장애물에 대해서 경로를 생성하고 (Fig. 18), 교차점이 두 개가 생기면 두 개의 장애물에 대해서 경로를 생성한다 (Fig. 19). 만약 교차점이 발생하지 않으면 직접 볼 방향으로 경로를 생성한다 (Fig. 20).

4. 베지어 곡선 방법을 이용한 장애물 회피

베지어 곡선을 이용한 방법은 원(장애물을 회피하기 위한 안전영역)의 영향을 받지 않기 때문에 로봇이 이동 중 급격한 변화를 줄일 수 있고, 목표지점까지 부드러운 주행을 할 수 있다 (Fig. 21). 장애물과 볼이 근접했을 경우 벡터장 항법과는 다르게 베지어 곡선을 이용한 방법은 척력이나 인력의 영향을 받지 않기 때문에 로봇이 장애물을 회피한 후 볼로 이동하는 모습을 Fig. 22에서 보여준다. 이동 경로에서 로봇의 진행방향과 볼의 각도차가 클 경우, 로봇은 제안된 각도차의 범위까지 제자리 회전을 한 후 이동을 하기 때문에 불필요한 움직임을 하지 않는다 (Fig. 23).

베지어 곡선 방법을 사용할 때 경로 설계시 볼까지 이동할 수 있는 여러 이동경로의 거리를 계산한다. 볼까지 이동할 수 있는 경

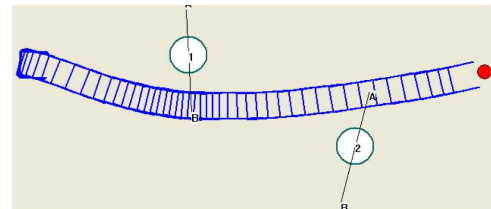


Fig. 21 Evasion of obstacles with small-changing path in Bezier curve navigation

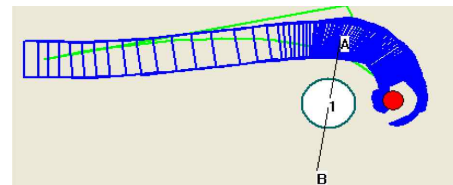


Fig. 22 Approach path to ball in Bezier curve navigation

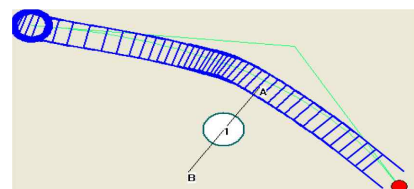


Fig. 23 Small-changing path in Bezier curve navigation

로 중 가장 짧은 경로를 결정하고, 베지어 곡선을 생성하여 주행하기 때문에 벡터장 항법보다는 이동거리, 이동 중 회전변화에서 우수하다. Fig. 24는 볼까지 최단거리를 계산한 후 베지어 곡선을 형성하고 로봇이 볼까지 이동하는 시뮬레이션을 보여준다.

로봇 축구 경기에서 장애물 회피시 많이 사용되는 원을 이용한 항법과 베지어 곡선 항법을 비교해 보면 원을 이용한 장애물 회피는 로봇이 원에 접근 시 두 번 이상의 급격한 회전 변화를 갖는다 (Fig. 25). 이는 구동 시 로봇의 회전에 의해 발생하는 구심력을 고려할 때 로봇이 경로를 이탈할 확률이 커지고 전류소모, 모터의 부하 등 로봇의 주행에 불리하게 작용한다. 그러나 베지어 곡선 항법을 이용하면, Fig. 26처럼 이런 문제점들을 줄일 수 있으며 매 순간 볼까지 최단거리를 계산하여 주행할 수 있다.

Fig. 27은 4개의 장애물에 대해서 벡터장 항법과 베지어 곡선 항법을 비교한 시뮬레이션의 예이다. 그림에서 보듯이 벡터장 항법은 각 장애물에서 로봇의 급격한 변화를 갖는 반면에 베지어 곡선 항법은 로봇의 이동 경로의 궤적이 유연한 것을 볼 수 있다.

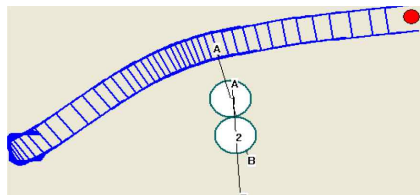


Fig. 24 Shortest path in Bezier curve navigation

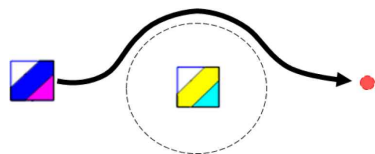


Fig. 25 Collision of robot and obstacle when using other navigation

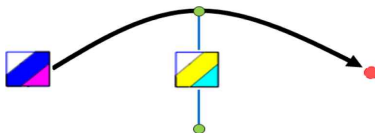


Fig. 26 Navigation path using Bezier curve navigation

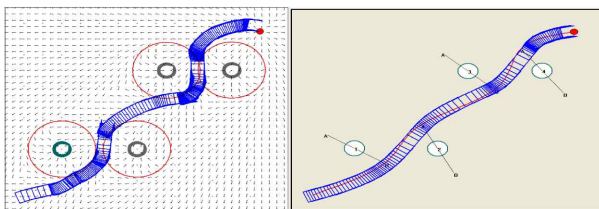


Fig. 27 Simulation of navigation of vector field and Bezier curve when 4 obstacles

5. 시뮬레이션 및 실험 결과

시뮬레이션과 실험의 조건은 Fig. 27처럼 1.5×1.3 m²의 경기장 그라운드에서 왼쪽 아래와 오른쪽 위에 각각 2개의 장애물(테니스공)을 설치하고 왼쪽 아래 모서리에서 출발하여 오른쪽 위 모서리에 도착할 때까지의 주행시간을 측정하였다.

Fig. 28의 시뮬레이션의 결과, 베지어 곡선 항법이 벡터장 항법보다 이동거리, 이동시간이 단축된 것을 볼 수 있다. 4번째의 이동 시간 경우는 시뮬레이션 상에서 1 ms마다 로봇의 좌표가 변하므로 벡터장 항법보다 느리게 볼까지 이동하였다. 그러나 실제 로봇에 적용 시 베지어 곡선 항법이 벡터장 항법보다 이동시간이 단축되는 것을 실험에서 보여준다. 시뮬레이션의 결과, 벡터장 항법에 비해 베지어 곡선 항법 사용 시 이동거리는 평균 95.3 pixel, 이동시

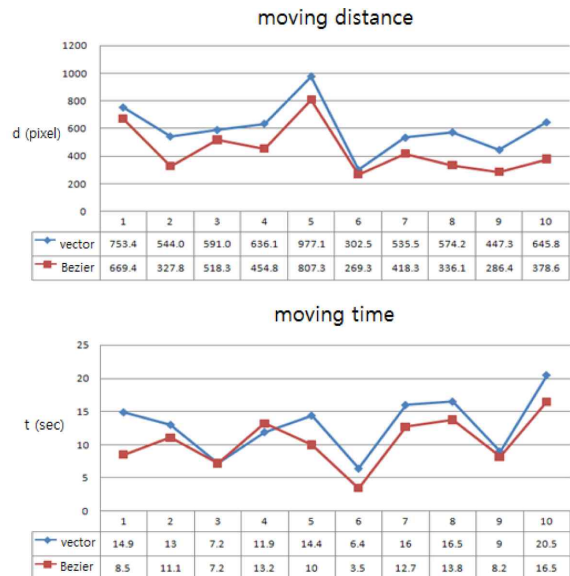


Fig. 28 Comparison of simulation results



Fig. 29 Comparison of experimental results

간은 평균 4.81 sec 단축되었다.

Fig. 29의 실험결과에서 보듯이, 이동거리와 이동시간에서 베지어 곡선 항법이 벡터장 항법보다 우수한 것을 볼 수 있다. 이동거리 단축은 각 로봇의 배터리 소모를 줄여 주고, 이동시간의 단축은 좀 더 빨리 로봇이 좋은 위치로 이동할 수 있는 이점이 된다. 실험결과, 베지어 곡선 항법이 평균 이동시간이 0.38초, 이동거리는 25.87cm 만큼 단축되었으며, 전체적으로 시뮬레이션 결과와 비슷한 실험결과가 나타났다.

6. 결론

본 논문에서는 로봇 축구 경기에서 장애물 회피 항법 중 가장 많이 쓰이는 벡터장 항법과 limit-cycle 항법의 장단점을 분석한 후, 이를 개선하기 위하여 베지어 곡선 항법을 새롭게 제시하였으며 기존 항법들의 문제점들을 효과적으로 해결할 수 있었다.

로봇 축구 경기에서 주행 시 발생하는 여러 가지 경우에 있어서, 각각의 항법들의 이동거리, 이동시간 등의 주행특성을 계산하기 위해 Visual C++를 이용하여 시뮬레이터를 제작하였고, 성능을 비교하기 위하여 주행결과 값을 비교분석하였다. 최종적으로 실험을 통하여 벡터장 항법과 베지어 곡선 항법의 성능을 비교하였으며, 결과적으로 베지어 곡선 항법이 상대적으로 우수함이 증명되었다.

장애물 회피 시뮬레이션 결과, 이동시간은 19.3%, 이동거리는 25.6% 개선되었으며, 장애물 회피 실험결과 이동시간은 23.5%, 이동거리는 27.3% 개선되었다. 특히, 실험에서 시뮬레이션 상에서 볼 수 없었던 로봇의 관성과 회전에 의해 발생하는 구심력의 영향을 볼 수 있었으며, 이로 인해 시뮬레이션 결과와 다소 차이가 났다. 그러나 결과적으로 베지어 곡선 항법이 벡터장 항법 보다 성능이 개선된 것을 알 수 있었다.

시뮬레이션에서 로봇의 이동경로와 실험에서 로봇의 이동경로가 다소 틀린 경우가 있었는데, 이는 시뮬레이션 상에서는 처음 계획된 경로로 로봇이 이동한 반면에, 실험에서는 로봇이 이동 중 로봇의 관성과 회전에 의해 발생하는 구심력에 의해 경로를 이탈 시, 이탈된 위치에서 경로를 재설정하기 때문에 이동경로의 차이

를 보였다.

본 논문에서는 베지어 곡선을 로봇 축구에서 사용하는 장애물 회피 항법에 적용시켰다. 로봇 축구 경기 전체 시간동안 본 논문에서 제안한 베지어곡선 항법을 실제로 적용해 본 결과를 검토해 보아야 할 것이다. 향후 B-spline 곡선이나 Hermite 곡선 등을 적용하여 여러 가지 경우에서 각각의 곡선 중 가장 좋은 곡선을 장애물 회피 항법에 적용해야 할 것이다.

후 기

이 연구는 서울과학기술대학교 교내학술연구비 지원으로 수행되었습니다.

References

- [1] Kim, J. H., Shim, H. S., Kim, S. H., 2002, Robot Soccer Engineering, Brain Korea, Republic of Korea.
- [2] Lee, H. C., 1997, Computer Graphics and Shape Modeling, Sigma Press, Republic of Korea.
- [3] Moshe, S., 1995, An Introduction To Artificial Life, Exploration in Artificial Life, Swiss.
- [4] Kim, D. H., Kim, J. H., 2003, A Real-time Limit-cycle Navigation Method for Fast Mobile Robots and its Application to Robot Soccer, Robotics and Autonomous Systems, 42:1 17-30.
- [5] Jung, Y. H., 2006, A Real Time Avoidance for Mobile Robot Using Nearness Diagram, Limit-cycle and Vector Field Method, A Thesis for a Master, Korea Aerospace University, Republic of Korea.
- [6] Latombe, J. C., 1991, Robot Motion Planning, Kluwer Academic Publishers, NewYork.
- [7] Bae, S. H., Choi, D. Y., 1994, Path Planning for the Obstacle Avoidance of the Autonomous Mobil Robot, ICROS Conference, 1:2 766-771.
- [8] Kim, D. H., 2008, Limit-cycle Navigation Method for Mobile Robot, Control Conference, 320-324.