

논문 2009-461E-3-6

Limit-cycle 항법의 동역학적 안정성 분석

(Analysis of Dynamic Stability of Limit-cycle Navigation Method)

김 동 한*, 강 수 혁****, 이 은 진****, 고 국 원**, 남 상 엽***

(Donghan Kim, Soo-Hyeok Kang, Eun-Jin Lee, Kuk Won Ko, and Sang Yep Nam)

요 약

본 논문에서는 Limit-cycle 항법의 안정성을 분석하고 path follower를 제안한다. Limit-cycle 항법은 2차 비선형함수의 특징을 이용하여, 빠른 이동로봇의 움직임을 제어하기 위해 제안되어졌다. Limit-cycle의 수렴반경과 방향을 조절하여 충돌 없이 장애물을 회피하고, 목표점에 도달할 수 있다. 하지만 지금까지 Limit-cycle 항법의 안정성은 연구되어지지 않았다. 따라서 본 논문에서는 이러한 안정성을 분석하고, Limit-cycle 항법을 로봇축구에 적용하여, 시뮬레이션과 실제 실험에서 제안된 방법을 검증한다.

Abstract

Because the stability of obstacle avoidance ability is important for the safe operation of mobile robots, this paper deals with the analysis of dynamic stability of Limit-cycle navigation method that was proposed by authors. Limit-cycle navigation method is fast and easy to implement for fast moving mobile robots using limit-cycle characteristics of the 2nd-order nonlinear function. It can be applied to robots in dynamically changing environment such as the robot soccer. By adjusting the radius of the motion circle and the direction of the obstacle avoidance, the mobile robot can avoid the collision with obstacles and move to the destination point. The stability of Limit-cycle navigation method is analyzed with a linear model. To demonstrate the effectiveness and applicability, it is applied to the robot soccer. Simulations and real experiments ascertain the merits of the proposed method.

Keywords : limit-cycle, navigation, dynamic stability

I. 서 론

이동로봇의 장애물 회피 항법은 그동안 많은 연구자들에 의해서 제안되어 왔다^[1~5]. 이러한 항법은 크게 deliberative 방법과 reactive 방법, 두 가지로 나눌 수

있다. deliberative 방법은 주위 환경에 대한 정확한 모델을 가지고 임무 수행에 필요한 계획을 만드는 것으로, 이론적으로 센싱, planning 그리고 acting의 세 단계로 나눌 수 있으며, 이를 통해서 원하는 목표까지의 궤적을 만들어 나가게 된다. 하지만 실제 실험하는 경우엔 매순간 환경이 변하는데 따른 새로운 경로 계획 수립과 환경에 대한 불확실성으로 야기되는 실패 때문에 많은 문제가 발생한다. 이에 반해, reactive 방법^[3~10], [11]은 로봇 주위의 환경에 대해서 모델이 조금 필요하거나 거의 필요 없으며, 로봇의 센서와 구동기 사이에 적절한 전달 함수를 이용해서 연결시킨다. 일반적으로 비교적 적은 계산량을 필요로 하므로, 로봇으로 하여금 아주 적은 시간 동안에 행동을 취하도록 하고, 계속적

* 정회원, **** 학생회원, 경희대학교 전자전파공학과
(Dept of Electronics Eng, KyungHee Univ)

** 정회원, 선문대학교 정보통신공학과
(Dept of I & C Eng, SunMoon Univ)

*** 평생회원, 국제대학 IT계열
(Dept of Information Technology, KookJe College)

※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임
(R01-2008-000-20352-0).

접수일자: 2009년5월4일, 수정완료일: 2009년9월9일

인 행동의 계산을 통해서 환경의 불확실성을 이겨낼 수 있다. 그렇지만, 표현과 추론 방법의 한계로 인해 순수한 reactive 시스템은 목표 지향적이지 못하다. 이러한 한계를 극복하기 위해서 2차 비선형 함수의 limit-cycle 특성을 이용한 새로운 방법을 제안했다^[12]. 이것은 장애물이나 로봇이 추가/감소되거나 환경이 계속 변화하는 상황에서도 적용될 수 있는 새로운 방법이며, reactive 방법의 유연성과 deliberative 방법의 목표지향적이라는 장점을 합친 것으로, 경로를 얻기 위해 2차 비선형 함수의 limit-cycle 특성을 이용한다. 이를 통해 local minima와 원치 않는 장애물의 영향을 극복하며, 로봇과 목표 사이에 여러 개의 장애물이 있는 경우에도 쓰일 수 있도록 확장한다. 하지만 기존의 Limit-cycle 방법에 path-generation은 안정성이 증명되었지만 path-following 부분에서는 현재까지 동역학적 안정성에 대한 분석이 없어서 시간경과에 따라서 로봇이 최종 목표까지 도달할 수 있다는 것을 증명하지 못했다^[12].

따라서 본 논문에서는 빠른 장애물이 많은 상황에서도 적용하기 용이한 Limit-cycle 방법의 동역학적 안정성을 증명하고 이를 위한 path-following 방법을 제안한다. 또한 가상실험과 실제 로봇에 적용하여 우수성을 입증하고자 한다.

다음 II장에서는 Limit-cycle 방법을 간단하게 설명하고, III장에서 path-following 방법을 제안하며, 이동 로봇의 동역학적 모델을 통한 Limit-cycle 방법의 안정성을 증명한다. 모의실험과 실험 결과는 IV장에서 보여지며, V장에서 결론을 맺는다.

II. Limit-cycle 방법

1. Limit-cycle

다음의 2차 비선형 시스템을 고려하자^[13].

$$\begin{aligned}\dot{x}_1 &= x_2 + x_1(1 - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + x_2(1 - x_1^2 - x_2^2)\end{aligned}\quad (1)$$

그리고, 이 함수의 Lyapunov 함수를 다음과 같이 가정한다.

$$V(x) = x_1^2 + x_2^2 \quad (2)$$

시스템의 궤적을 따라 $V(x)$ 를 미분한 값은 다음과 같다.

$$\begin{aligned}\dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_1^2(1 - x_1^2 - x_2^2) - 2x_1x_2 \\ &\quad + 2x_2^2(1 - x_1^2 - x_2^2) = 2V(x)(1 - V(x))\end{aligned}\quad (3)$$

위의 식에서 $V(x)$ 의 미분값은 $V(x) < 1$ 인 경우엔 양수이고, $V(x) > 1$ 인 경우엔 음수이다. 그러므로, $0 < c_1 < 1$ 의 경우 $V(x) = c_1$ 인 level 평면에서는 모든 궤적이 밖으로 향하고, $c_2 > 1$ 인 경우엔 $V(x) = c_2$ 의 level 평면에서는 모든 궤적이 안으로 향한다. 이것은 다음과 같은 주기적인 영역,

$$M = \{x \in R^2 | c_1 \leq V(x) \leq c_2\} \quad (4)$$

이 positively invariant하다는 뜻이다. 또한 닫혀있고, bounded 되어 있으며, 원점 $x = 0$ 이 유일한 equilibrium point이기 때문에 Poincare-Bendixson 정의에 의해 M 안에는 한 개의 주기적인 궤도가 존재한다. 이는 어떠한 $c_1 < 1$ 과 어떠한 $c_2 > 1$ 에도 성립하므로, c_1 과 c_2 는 1에 접근한다. 결국엔 집합 M 은 그림 1과 같이 단위 원으로 수렴한다. 이런 주기적인 궤도를 limit-cycle이라고 한다.

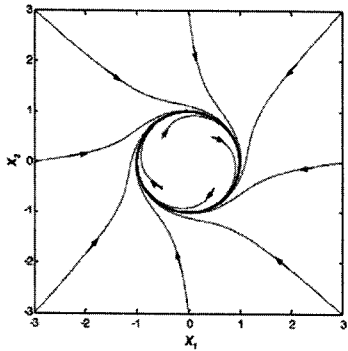
그림 1(a)은 식 (1)의 x_1 과 x_2 을 x -축과 y -축에 각각 대응시켜 phase portrait을 나타낸 것이다. 원점을 포함한 모든 점 (x_1, x_2) 에서의 궤적은 앞서 설명한 시계방향의 단위원으로 수렴한다. 반시계방향의 단위원으로 수렴시키기 위해서는 앞의 2차 비선형식 (1)을 다음과 같이 바꿔야 한다.

$$\begin{aligned}\dot{x}_1 &= -x_2 + x_1(r^2 - x_1^2 - x_2^2) \\ \dot{x}_2 &= x_1 + x_2(r^2 - x_1^2 - x_2^2)\end{aligned}\quad (5)$$

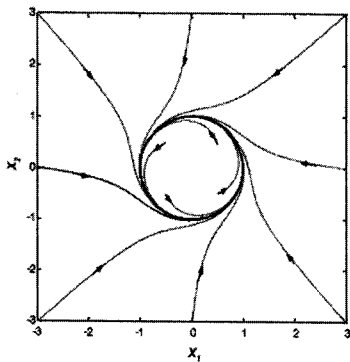
그러면, 모든 궤적은 그림 1(b)와 같이 반시계 방향의 단위원으로 수렴한다. 단위원이 아닌 일반적인 반지름 r 에 대한 식 (1)의 일반적인 형태는 다음과 같다.

$$\begin{aligned}\dot{x}_1 &= x_2 + x_1(r^2 - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + x_2(r^2 - x_1^2 - x_2^2)\end{aligned}\quad (6)$$

지금까지 limit-cycle의 방향과 크기를 조절하는 방법을 설명했다. 앞으로 이를 이용해서 limit-cycle 방법을 만들도록 한다. 이 방법은 로봇으로 하여금 장애물을 회피하는데 어느 방향으로, 어느 정도로 피할지를 정할 수 있게 한다.



(a)clockwise



(b)counter-clockwise

그림 1. Limit-cycle의 모양
Fig. 1. Phase portrait of limit-cycle.

2. Limit-cycle 항법

본 절에서는 limit-cycle을 이용한 항법의 과정을 설명한다.

1. Global 좌표계, Σ_{xyo} 에서 로봇과 목표 사이를 잇는 직선 l 을 다음과 같은 식을 이용하여 그린다.

$$ax + by + c = 0 \tag{7}$$

2. 직선 l 이 지나는 Variable 장애물은 disturbing 장애물, O_d 로 간주하고, 지나지 않는 장애물은 non-disturbing 장애물, O_n 으로 간주한다.
3. O_d 가 없다면 목표로 향해서 간다.
4. 그림 2에서, 가장 가까운 disturbing 장애물의 중심에서 직선 l 까지의 거리 d 를 다음의 식으로 구한다.

$$d = \frac{aQ_x + bQ_y + c}{\sqrt{a^2 + b^2}} \tag{8}$$

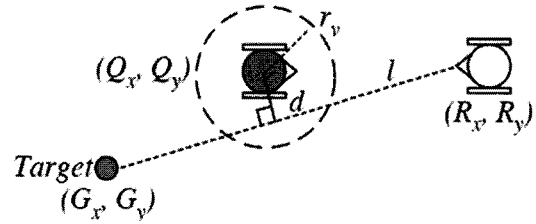


그림 2. 회전방향의 계산
Fig. 2. Decision of rotational direction.

여기서 (Q_x, Q_y) , (G_x, G_y) 와 (R_x, R_y) 는 각각 장애물, 목표 그리고, 로봇의 xy -값이 된다. d 와 r_v 를 추가하고, x_1 과 x_2 를 global 좌표계 Σ_{xyo} 의 x 와 y 으로 바꾸면, 다음과 같이 확장된다.

$$\begin{aligned} \dot{x} &= \frac{d}{|d|}y + x(r_v^2 - x^2 - y^2) \\ d &= -\frac{d}{|d|}x + y(r_v^2 - x^2 - y^2) \end{aligned} \tag{9}$$

여기서 x 와 y 는 장애물로부터의 상대 위치이다. 위의 식에서 d 가 양수이며, 로봇은 장애물 O_d 를 시계방향으로 피한다. 마찬가지로 음수이면, 반시계 방향으로 피한다. r_v 는 다음의 식과 같이 로봇과 장애물의 크기와 상대 위치에 따라 정해진다.

로봇이 움직이면서 l 은 계속 변하고, 이에 따라 과정 2~4를 임무를 완수할 때까지 반복한다.

III. Limit-cycle 항법의 안정성

1. Path-following controller 설계

본 논문에서는 미끄러짐이 없고 순수하게 구름만이 있는 차동 구동 방식의 이동 로봇을 고려한다. 속도 벡터, $Q = [v \ w]^T$ 는 로봇의 중심 속도, v 와 로봇의 중심을 기준으로 하는 회전 각속도, w 로 구성된다. 이 속도 벡터 Q 와 자세 벡터, $P = [x \ y \ \theta]^T$ 와는 다음의 로봇 기구학식으로 관련지어질 수 있다.

$$\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = J(\theta)Q \tag{10}$$

$$Q = [v \ w]^T = \begin{bmatrix} \frac{V_r + V_l}{2} & \frac{V_r - V_l}{L} \\ \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_l \\ V_r \end{bmatrix} \quad (11)$$

여기서 V_l 과 V_r 는 각각 로봇의 왼쪽, 오른쪽 바퀴의 속도이다. 그러므로 로봇은 이 V_l 과 V_r 로 자세를 제어할 수 있다. 그림 2(b)과 같이, 현재 위치 (x, y) 에서 로봇의 원하는 방향인 θ_d 을 계산하고, 로봇의 각도 θ_r 를 이용하면, 다음과 같이 각도 오차 θ_e 가 구해진다.

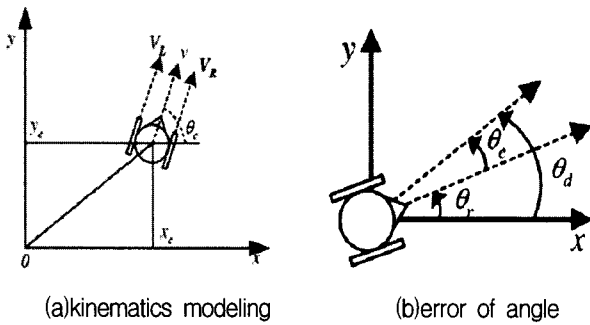


그림 3. 로봇 모델링
Fig. 3. Robot modeling.

$$\theta_e = \theta_d - \theta_r \quad (12)$$

이를 이용해서 로봇이 원하는 방향 θ_d 로 움직이게 하는 제어 방법은 다음의 식과 같이 구할 수 있다.

$$\begin{aligned} V_l &= v - K_1\theta_e - K_2 \int \theta_e dt \\ V_r &= v + K_1\theta_e + K_2 \int \theta_e dt \end{aligned} \quad (13)$$

여기서 K_1, K_2 는 비례상수이며, v 는 거리에 따른 중심속도이다. 일반적으로 거리에 비례하는 값을 주거나 일정한 값을 가지다가 정해진 거리에 도달하면 급격하게 줄어드는 기하급수 함수를 이용한다. 이것은 실험적으로 찾아낸 것이며, 본 논문에서는 후자의 것을 이용한다.

2. 동역학적 안정성 증명

본 논문에서 다루는 이동로봇의 동역학식은 다음과 같다.

$$I \ddot{\theta} = (u_r - u_l)L \quad (14)$$

여기서 I 는 로봇의 inertia이며, u_r 과 u_l 은 각각 오른쪽 바퀴와 왼쪽 바퀴의 힘이다. L 은 두 바퀴 사이의 거리의 반이다. 식(14)은 non-holonomic 제약을 나타내며, 이는 일반적인 이동로봇이 갖는 제한이기도 하다. 식(13)을 식(14)에 대입하기 위해 미분하면 다음과 같다.

$$\begin{aligned} \dot{V}_l &= \dot{v} - K_1\dot{\theta}_e - K_2\theta_e \\ \dot{V}_r &= \dot{v} + K_1\dot{\theta}_e + K_2\theta_e \end{aligned} \quad (15)$$

위 식에서 왼쪽 항은 왼쪽 바퀴와 오른쪽 바퀴의 가속에 해당하며, 이는 외측 바퀴와 오른쪽 바퀴의 힘에 비례한다고 할 수 있다. 따라서 이를 식(14)의 u_r 과 u_l 에 각각 대입하면, 다음과 같다.

$$\begin{aligned} \ddot{\theta}_r &= M(\dot{v} + K_1\dot{\theta}_e + K_2\theta_e - \dot{v} + K_1\dot{\theta}_e + K_2\theta_e)L \\ &= 2M(K_1\dot{\theta}_e + K_2\theta_e)L \end{aligned} \quad (16)$$

또한 식(12)에서 목표 각도인 θ_d 는 한 샘플링 주기 동안에 일정한 값을 갖는다. 따라서 이를 두번 미분하게 되면 다음과 같은 식이 된다.

$$\ddot{\theta}_r = \ddot{\theta}_e \quad (17)$$

따라서 식(16)는 다음과 같이 정리된다.

$$\ddot{\theta}_r = 2M(K_1\dot{\theta}_e + K_2\theta_e)L \quad (18)$$

마지막으로 로봇의 바퀴를 포함한 기구부와 바닥과의 마찰 C 를 고려한 항을 추가하면 다음과 같은 최종 식이 유도된다.

$$\ddot{\theta}_e = (C - 2MK_1L)\dot{\theta}_e - 2K_2\theta_e L = 0 \quad (19)$$

위의 식은 두개의 실근을 갖게 된다. 따라서 θ_r 이라는 각도를 갖는 이동로봇이 목표 각도인 θ_d 로 수렴하도록 하는 조건은 다음과 같은 정리로 나타낼 수 있으며, 식(19)의 두 실근이 모두 음의 값을 갖게 되면, 각도 오차 θ_e 가 빠르게 0으로 수렴하게 되어, 로봇의 각도 θ_r 이 목표 각도인 θ_d 로 수렴하도록 하게 된다.

위의 증명과정으로부터 정리 1이 정리된다.

정리 1. 식(14)를 만족하는 차륜 구동방식의 이동로봇이 식 (9), (13)의 Limit-cycle 항법을 따라 갈 경우, 로봇의 각도 θ_r 이 목표 각도인 θ_d 로 수렴하도록 하는 제어입력 K_1 과 마찰계수 C 가 존재한다.

본 절에서는 Limit-cycle 항법을 따라 이동하는 로봇이 목표하는 방향으로 항상 수렴한다는 것을 증명했다. 다음 절에서는 이를 모의실험과 실제 실험에 적용하여 우수성과 안정성을 보이고자 한다.

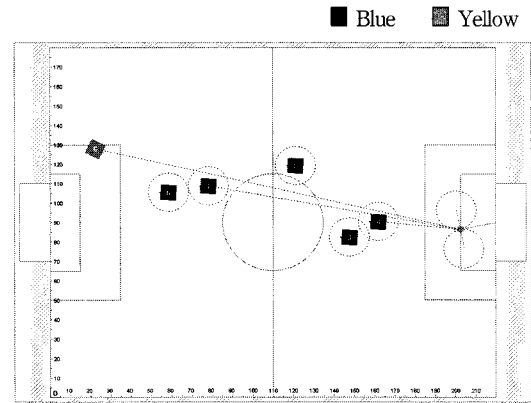
IV. 모의실험과 실험 결과

1. 모의실험

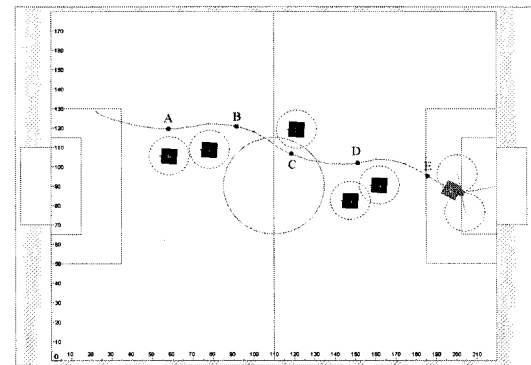
제안된 limit-cycle 항법을 KMC(Kyunghee Multi Cooperation)팀의 시뮬레이터를 사용하여 모의실험을 했다. 실제 로봇 축구 상황을 고려하기 위하여 FIRA의 Middle League MiroSot^[14]의 규칙을 준수하였다.

KMC팀의 시뮬레이터에서 로봇은 상대 로봇을 피해서 상대 골대 쪽으로 공을 차게 된다. 모의실험에서는 상대골대의 중앙으로 공을 찰 수 있도록 로봇이 공을 차는 각도를 고려하여 수렴반경을 정의하였다. 실제 로봇 축구 상황에서는 상대 골키퍼의 위치에 따라 공을 차는 방향을 달리하면 된다.

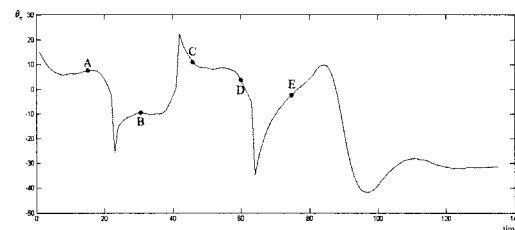
그림 4에서 Yellow팀의 0번 로봇은 경기장에 배치된 Blue팀의 로봇들을 피해서 공을 골대 쪽으로 차야 한다. 그림 4(a)에서 로봇은 (20, 130)에서 출발한다. 공으로 향하다가 Blue팀의 0번, 1번 로봇을 만나게 된다. 시계 방향으로 로봇을 피한 후 2번 로봇을 시계방향으로 피한다. 마지막으로 3번, 4번 로봇을 반 시계 방향으로 피하면 모든 장애물을 피하게 된다. 목표점인 공에 도달하면, 로봇은 공을 상대골대 가운데로 차게 된다. 그림 4(b)는 각 장애물에서 limit-cycle에 대한 로봇과의 각도 오차를 보여준다. 반 시계방향으로 로봇을 회피할 때 각도오차는 감소하며, 시계방향으로 로봇을 회피할 때 각도오차는 증가한다. 로봇이 공을 상대 진영으로 차면, limit-cycle과 로봇과의 각도오차는 일정한 각도로 그림 4(c)처럼 수렴하게 된다. 그림 4(c)를 보면, x축은 시간을 나타내며 sampling time이 30ms이므로 로봇은 목표지점까지 4.2초(140sample × 30ms)만에 도달한다.



(a) 상황



(b) 모의실험 결과



(c) 각 지점에서의 각도오차

그림 4. 로봇축구 예제1

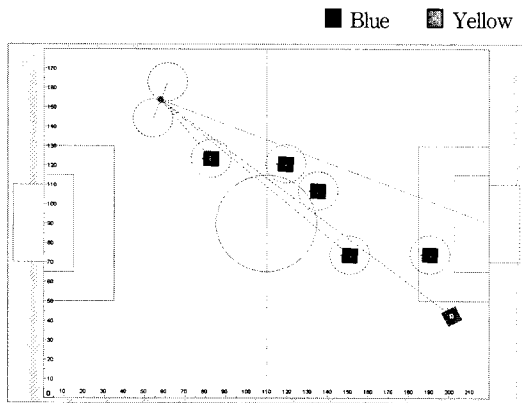
Fig. 4. Robot soccer example1.

실험환경은 표 1과 같다.

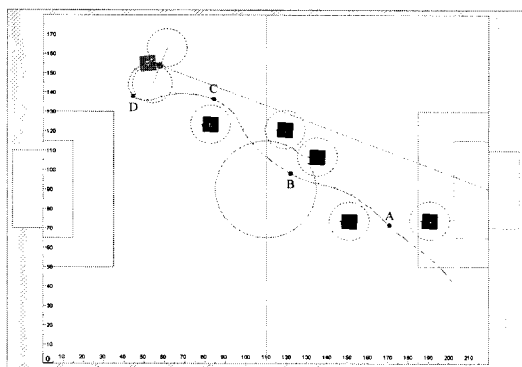
표 1. 시뮬레이션 환경

Table 1. Simulation environment.

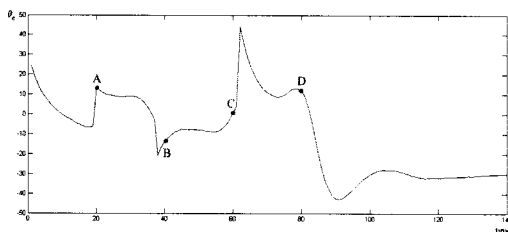
경기장 규격	220cm(가로)*180cm(세로) (골대제외) 250cm(가로)*180cm(세로) (골대포함)
로봇 규격	8cm(가로)* 8cm(세로)
개발환경	Microsoft Visual Studio 2005



(a) 상황



(b) 모의실험 결과



(c) 각 지점에서의 각도오차

그림 5. 로봇축구 예제2

Fig. 5. Robot soccer example2.

다음으로 그림 5의 로봇 축구 상황을 보자. 공이 Yellow팀 진영에 있고, Yellow팀의 로봇이 상대 진영 있을 때 상대 로봇을 회피 한 후, 공을 상대진영 끝대로 차는 경우이다. 그림 4의 경우와 마찬가지로 limit-cycle을 따라 각 로봇들을 시계, 반 시계 방향으로 회피한 후 목표점에 도달하여 공을 차게 된다.

이번 절에서는 limit-cycle 방법을 실제로 로봇 축구 상황을 적용한 시뮬레이터를 사용하여 모의실험을 했다. 로봇이 각 상황에서 일정한 원으로 움직여 장애물을 피하였으며, 목표점인 공에 도달하여 공을 차서 그 공을 상대 끝대로 향하는 득점상황을 보였다. 각각의 장애물을 회피하여 최단경로로 목표점인 공에 도달하는 것을

확인하였으며, 두 개의 가상의 variable 장애물을 가정하여 로봇 축구에 성공적으로 적용했다.

2. 실험 결과

실제 이동 로봇을 활용한 실험적 검증을 위해 로봇 축구 시스템을 사용하였다. 그림 6의 전체 시스템은 로봇과 호스트 컴퓨터, 비전 시스템, 통신 시스템으로 이루어져 있다. 비전시스템 정보를 이용하여 호스트 컴퓨터는 로봇의 제어 신호를 만들어내고, 만들어진 제어 신호는 통신 시스템을 통하여 축구 로봇에게 전달되는 구조이다.

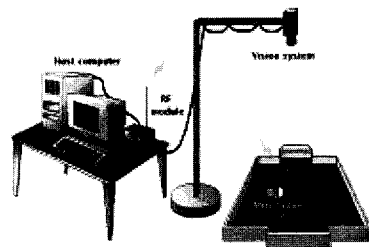


그림 6. 로봇축구 시스템

Fig. 6. Robot soccer system.

실험에 호스트 컴퓨터로 사용된 기종은 IBM PC 이며, 비전 시스템에는 Uniqvision UC-685 10-bit color digital CCD Camera가 사용되었다. 비전 시스템의 노이즈로 위치와 방향에 약간의 오차가 있을 수 있으나 실험에 영향을 주지 않을 정도이므로 무시할 수 있다. 사용된 소프트웨어는 Visual C++로 제작되었다. 사용된 로봇은 가로 세로 높이 각각7.5cm 크기이며 DSP TMS320F2811 마이크로 컨트롤러와 2개의 DC 모터로 동작한다.

a. 장애물이 없을 경우

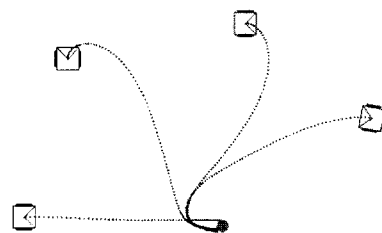


그림 7. 장애물이 없을 때 주행 경로

Fig. 7. Without Obstacles trajectories.

그림 7은 장애물이 없을 때 로봇의 주행을 나타낸 그림이다. Limit-cycle의 모양대로 주행이 잘 이루어지고 있음을 알 수 있다.

b. 기본 장애물 회피

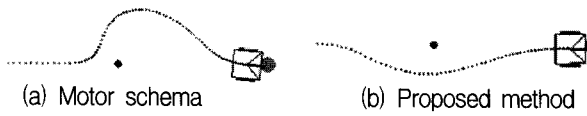


그림 8. 기본 장애물 회피
Fig. 8. Basic Obstacle avoidance.

장애물이 하나 있을 때 장애물을 회피하는 경로를 나타낸 그림이다. 그림 8(a)에서는 기존의 모터 스키마 방법으로, 로봇의 속도와 방향을 고려하지 않았을 경우이며, 로봇의 장애물회피가 비효율적인 것을 알 수 있다. 하지만 그림 8(b)의 이번 연구를 통해 고안한 limit-cycle방법을 보면 로봇이 장애물을 자연스럽게 피하는 것을 볼 수 있다.

c. 복잡한 장애물 회피

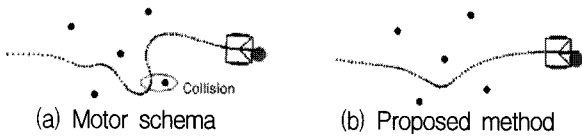


그림 9. 복잡한 장애물 회피
Fig. 9. Complex obstacles avoidance.

장애물이 복잡하게 배치되어 있을 경우를 살펴보면, 그림 9(a)의 기존 모터 스키마 방법에서는 장애물과 마주칠 때마다 심하게 튕겨나가면서 비효율적인 경로를 그리고 있으며 마지막엔 장애물과 충돌을 피하지 못하였다. 반면 그림 9(b)의 limit-cycle 방법에서는 장애물이 하나있는 경우와 마찬가지로 효율적이고 자연스럽게 장애물을 피하는 것을 볼 수 있다.

이번에는 장애물 주변에서 장애물의 바람직하지 영향이 얼마나 제거 되었는지 살펴보자. 그림 10(a)의 모터 스키마 방법에서는 장애물이 경로에 방해가 되지 않는 위치에 있음에도 불구하고 장애물의 척력이 로봇의 경로에 영향을 미쳐서 로봇이 멀리 돌아가게 됨을 알 수 있다. 즉, 이러한 경우는 장애물 주변에서 장애물의 영향으로 비효율적인 주행을 하게 되는 경우이다.

반면, 그림 10(b)의 limit-cycle 방법은 장애물이 있을 경우와 없을 경우가 거의 차이가 없다. 즉, 모터 스키마 방법과는 다르게 장애물의 바람직하지 않은 영향을 거의 받지 않는 것이다.

d. 장애물 주위에서 로봇의 움직임

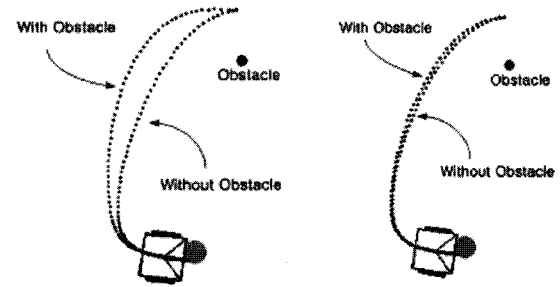


그림 10. 장애물 주위에서 로봇의 움직임
Fig. 10. Motion of the robot around obstacle.

그림 11은 여러 개의 고정 장애물이 있는 상황에서 장애물을 피하여 오른쪽에서 왼쪽으로 이동하는 다양한 경우의 실험결과를 보이고 있다. 시뮬레이션과 마찬가지로 로봇이 효율적으로 장애물 회피를 하는 것을 알 수 있었다.

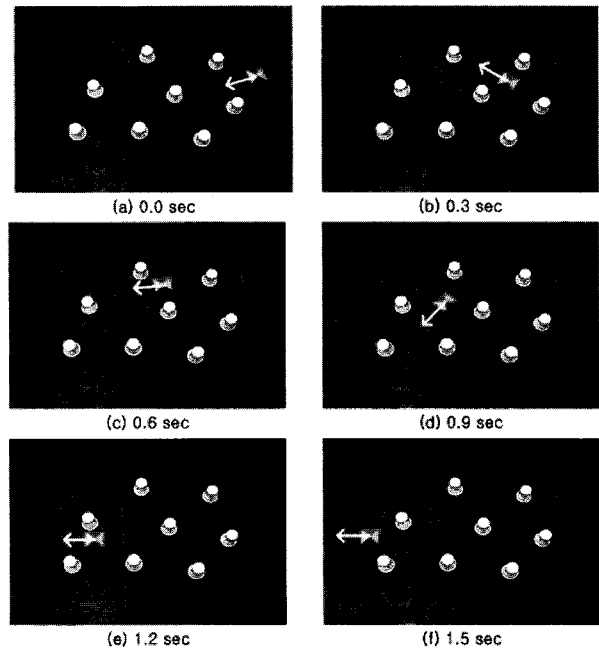


그림 11. 실험결과
Fig. 11. Experimental results.

본 절에서는 다양한 실험을 통해서, limit-cycle의 장점을 보였다. Potential field법의 local minima와 원하지 않는 장애물의 영향을 회피하는 원의 방향과 크기를 조절함으로써 해서 해결했다. 가상의 variable 장애물을 가정하여, 목표점에 도달했을 때 원하는 방향으로 향하도록 했다. 이것은 로봇 축구에서 공을 상대 골대로 차는

경우에 사용되었다. 이 장에서의 실험 결과는 limit-cycle 항법이 reactive 방법처럼 유연하고, deliberative 방법처럼 목표지향적임을 나타내며, 로봇 축구와 같이 변화하는 환경의 경우에도 잘 적용됨을 보였다.

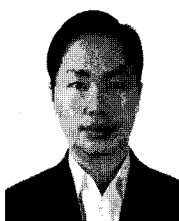
V. 결 론

본 논문에서는 2차 비선형 함수의 limit-cycle 특성을 이용해서 로봇 축구와 같은 환경이 빠르게 변화하는 경우에 대해서도 적용 가능한 새로운 항법을 제안했다. 로봇이 장애물을 회피하는 방향뿐만 아니라 거리를 조절함으로 해서 어떠한 경우에도 원하는 목표로 갈 수 있었다. 제안한 항법의 우수성을 보이기 위해 컴퓨터를 통한 모의실험과 실제 로봇을 이용한 실험 결과를 보였다.

참 고 문 헌

- [1] Ronald C. Arkin, "Motor Schema-Based Mobile Robot Navigation," *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 92-112, 1989.
- [2] Ronald C. Arkin, "Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation," *Robotics and Autonomous Systems*, vol. 6, pp. 105-122, 1990.
- [3] Rodney A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal on Robotics and Automation*, vol. RA-2, no. 1, 1986.
- [4] Dong-Han Kim, Yong-Jae Kim, K-C Kim, Jong-Hwan Kim and P. Vadakkepat, "Vector Field Based Path Planning and Petri-net Based Role Selection Mechanism with Q-learning for the Soccer Robot," *Intelligent Automation and Soft Computing*, vol. 6, no. 1, pp. 75-88, 2000.
- [5] Yong-Jae Kim, Jong-Hwan Kim and Dong-Soo Kwon, "Evolutionary Programming-based Univector Field Navigation Method for Fast Mobile Robots," *IEEE Trans. on Systems Man, and Cybernetics*, vol. 31, no. 3, pp. 450-458, 2001.
- [6] Hyun-Sik Sim, Myung-Jin Jung, Heung-Soo Kim, Jong-Hwan Kim and P. Vadakkepat, "A Hybrid Control Structure for Vision Based Soccer Robot System", *Intelligent Automation and Soft Computing*, *Intelligent Automation and Soft Computing*, vol. 6, no. 1, pp. 89-101, 2000.
- [7] Moon-Su Lee, Myung-Jin Jung and Jong-Hwan Kim, "Evolutionary Programming-based Fuzzy Logic Path Planner and Follower for Mobile Robots," *Congress on Evolutionary Computation*, vol. 1, pp. 139-144, San Diego, USA, July, 2000.
- [8] Marc G. Slack, "Navigation Template: Mediating Qualitative Guidance and Quantitative Control in Mobile Robots," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 23, no. 2, pp. 452-466, 1993.
- [9] Y. Koren and J. Borenstein, "Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation," *Proceedings of the IEEE Conference on Robotics and Automation*, vol. 2, pp. 1398-1404, 1991.
- [10] Erann Gat, "Navigation Templates: Enhancements, Extensions and Experiments," *IEEE International Conference on Robotics and Automation*, vol.1, pp. 541-547, 1993.
- [11] E. Rimon, D. Koditschek, "Exact navigation using artificial potential functions," *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 5, pp. 501-518, 1992.
- [12] Keun-Ho Rew and Donghan Kim, "Limit-cycle Navigation Method for Fast Mobile Robots," *Journal of Institute of Control, Robotics and Systems*, Vol. 14, No. 11, pp. 1130-1138, 2008.
- [13] Hassan K. Khalil, *Nonlinear Systems*, 2nd ed., Prentice Hall, chap. 7, pp. 289-312, 1996.
- [14] <http://www.fira.net/soccer/mirosot/MiroSot%20Rules.doc>

저 자 소 개



김 동 한(정회원)
1995년 KAIST 전기 및
전자공학과 학사 졸업
1998년 동대학원 석사 졸업
2003년 동대학원 박사 졸업
2003년~2004년 일리노이
주립대학 Post Doc

2004년~2007년 삼성전자 무선사업부
책임연구원.
2007년~현재 경희대학교 전자정보대학 조교수
<주관심분야 : 로봇제어, 다개체 지능제어>



강 수 혁(학생회원)
2005년~현재 경희대학교
전자공학과 재학
<주관심분야 : 로봇제어, 퍼지제어>



이 은 진(학생회원)
2003년~현재 경희대학교
전자공학과 재학
<주관심분야 : 로봇제어, 뉴럴네트
트워크>



고 국 원(정회원)
1992년 서울대학교 농업기계학과
학사 졸업
1994년 한국과학기술원
정밀공학과 석사 졸업
2001년 한국과학기술원
기계공학과 박사졸업
2002년~현재 선문대학교 정보통신공학과 조교수
<주관심분야 : 영상처리, 자동제어, 로봇제어>



남 상 엽(평생회원)
2002년 단국대학교
전자공학과 박사 졸업.
1987년~1992년 삼성종합기술원
정보시스템연구소
주임연구원
1992년~1998년 (주)모토로라반도
체통신 기술연구소 차장

1998년~현재 국제대학 IT계열 교수
<주관심분야 : 지능형로봇, 임베디드 시스템,
USN/RFID, 홈네트워크>