이동 로봇의 자율주행을 위한 목표점 지향 반사 주행 방식

論 文
53D-4-6

On a Goal-Directed Reactive Navigation Method for a Mobile Robot

吳 勇 煥* · 尹 道 榮** · 吳 尙 錄*** · 朴 貴 泰§
(Yong-Hwan Oh · Do-Young Yoon · Sang-Rok Oh · Gwi-Tae Park)

Abstract - This paper proposes two contributions. One is an analysis for the limit of the subject of goal-directed reactive robot navigation, and the other is an effective navigation method employing the scheme of the subject. The analysis for the subject is presented in order to clarify the limit of the method. On the basis of the analysis, a safety-guaranteeing and deadlock-free reactive navigation method is newly proposed. The proposed method has a simple behavior-based frame such that it can make the required navigation tasks such as obstacle avoidance, deadlock resolving, and etc. with a very small set of behaviors in entirely unknown environments such as a living room, an office, and etc. Some results of experiments show these validities.

Key Words : Reactive navigation, Goal-directed navigation, Mobile robot, Behavior-based robotics, Deadlock

1. 서 론

절대적인 위치 정보에 기반하는 로봇의 주행[1] 방식(absolute navigation [1])이 자기위치확인(localization) 도구를 이용한 전역적인(global) 정보를 이용하는 반면에 반사적인 주행(reactive navigation) 방식은 센서로부터 얻는 전적으로 지역적(local)이고 상대적인(relative) 정보만을 이용하여 온라인으로 단계적으로(step-by-step) 경로를 생성하는 것을 그 특징으로 한다. 이렇게 지역적이고, 부분적이며 불완전한 형태의 정보에 의존하기 때문에 반사적인 주행 방식들은 몇 가지 문제점을 가지게 된다 [2-3]. 그러나 반사적인 주행 방식의 단순한 구조와 변화하는 환경에서의 적응성은 이 방식이 실제의 주행 시스템을 구성하는데 있어 중요한 일부분으로 사용되도록 하고 있다 [4]. 절대적인 주행 방식과 반사적인 주행 방식을 혼용하는 실제의 많은 주행 시스템에서 반사적인 주행 방식은 전체 주행 시스템의 하나의 요소로서 활발히 사용되고 있음을 볼 수 있다 [3], [5-7]. 비용의 측면에서 보면 주행 시스템은 절대적인 주행 방식이 차지하는 비중이 증가할수록 비용과 복잡도가 증가한다. 따라서 얼마나 많은 비중을 절대적인 주행 방식과 반사적인 주행 방식에 각각 할당할 것인가를 결정하는 것이 필요하다. 목표점 지향 반사 주행 방식(goal-directed reactive navigation)은 반사적 방식의 일종이지만 엄밀하게는 일종의 절대적/반사적 주행 방식의 혼합형이라고 볼 수도 있다. 왜냐하면 이 방식은 현재위치에서 목표점으로 향하는 방향각을 사용하게 되는데 이 정보가 전역적인 것이기 때문이다. 그러나 이는 로봇이 목표점으로 접근하기 위한 최소한의 전역적인 정보이다. 목표점에 대한 전역적인 정보가 전무한 상태로는 목표점에 접근하기 불가능하기 때문이다. 현재 위치에서의 목표점으로의 방향각은 목표점의 방향각과 거리를 모두 사용하는 완전한 형태의 전역적인 정보와 비교해 볼 때 약한 형태의 전역적인 정보라고 볼 수 있다. 따라서 목표점의 거리를 배제하고 방향각만을 필요로 하기 때문에 완전한 형태의 목표점 정보보다 지역적인 방식으로 획득할 수 가 있다. 예를 들어 단순한 형태의 비젼 시스템이나 유도 시스템(beacon)을 가지고도 복잡한 자기위치확인 시스템 없이 손쉽게 목표점으로의 방향각을 얻을 수가 있다 [8].

이 논문은 우선 이러한 목표점 지향 반사 주행 방식의 한계를 분석적으로 제시한다. 이 작업은 전체적인 주행 전략을 설계할 때 반사 주행 방식을 어느 범위까지 사용해야 하는가의 현실적인 범위를 명확히 하는데 필요하다. 두 번째로 이러한 분석의 토대위에서 하나의 목표점 지향 반사 주행 방식을 새로이 제안한다. 이 방식은 특히 행위 기반(behavior-based) 접근 방식[9]을 중심으로 하는데 이를 통하여 단순한 형태의 제어 구조로써 자율주행에서 요구하는 여러 가지 임무들을 성공적으로 수행할 수 있다. 제안한 주행 방식은 반사 주행 반식의 한계를 높은 성공 확률로 극복할 수 있다.

* 正 會 員 : 한국과학기술연구원 선임연구원 · 工博
** 正 會 員 : 고려대학교 전기공학과 박사과정
*** 正 會 員 : 한국과학기술연구원 책임연구원 · 工博
§ 正 會 員 : 고려대학교 전기공학과 敎授 · 工博
接受日字 : 2003年 12月 12日
最終完了 : 2004年 2月 19日

1) 본 논문에서는 로봇의 자율 주행을 위한 경로 계획에 중점을 두고 있어 경로 추종을 위한 로봇의 제어에 관한 문제를 배제하기 위하여 holonomic 로봇을 가정하였다.

2. 목표점 지향 반사 주행 방식의 한계에 대한 분석

목표점 지향 반사 주행 방식을 사용하는 기존의 여러 방식들이 있다. 이 방식들에는 전위장(potential field) 방식[10], 벽면 따라가기 방식(wall following)[11] 등의 기본적인 반사 주행 방식들과 임시적인 목표점 변경(local target switching) 방식[3], 조화 전위장(harmonic potential field) 방식[12] 등과 같은 기본적인 방식의 문제점들을 완화하기 위하여 제안된 방식들이 있다. 이러한 방식들은 핵심적인 알고리즘의 측면에서 성공적이고 유용한 것들이다. 그러나 "과연 임의의 모든 형태의 지형에서 자율적으로 주행 할 수 있는 목표점 지향 반사 주행 방식이 존재하는가?"라는, 현재까지는 해답이 불명확한 질문에 대한 언급을 이들 기존의 연구에서는 찾아볼 수 없다. 이 대답은 불행히도 "아니오" 이다. 이 논문은 이러한 목표점 지향 반사 주행 방식의 한계를 분석적으로 제공한다. 반사주행 방식에서 가장 심각한 문제점은 주행 중에 교착상태(deaklock)에 빠지는 것이다. 이 문제는 로봇의 짧고 지엽적인 시야에서 발생하는 문제이다. 또 다른 문제로 장애물 회피가 있지만 이는 비교적 해결하기 쉬운 문제이다. 따라서 반사주행 방식의 한계점은 교착상태에 대한 분석을 통하여 알 수 있다. 우선 주행 알고리즘과 로봇의 센서 장치에 대한 몇 가지 가정을 하고 분석을 시작한다. 이 가정들은 대부분의 반사주행 방식들이 모두 취하고 있는 것들이어서 분석의 일반성을 크게 해치지는 않는다 [3], [11], [13-14].

1. 센서 장치에 대한 가정

Assumption 1.1: 사용 가능한 정보는 현재 위치에서의 목표점으로의 방향각과, 지역적인 센서 데이터 뿐이다.

Assumption 1.2: 센서의 감지 범위는 주위 환경(벽, 장애물 등)의 크기에 비하여 충분히 짧다.

Assumption 1.3: 짧은 시간동안(short-term) 저장한 지역적인 정보를 사용할 수 있다.

2. 주행 알고리즘에 대한 가정

Assumption 2.1: 교착상태에서 벗어나기 위한 알고리즘은 교착상태에 빠졌음을 감지하는 위치(deadlock detection point)와 교착상태에서 벗어나 정상 상태로 되는 지점(end of deadlock point)의 두 개의 특징점을 사용한다.

Assumption 1.1은 해당 주행 알고리즘이 목표점 지향 반사 주행 방식임을 말한다. Assumption 1.2는 해당 주행 임무가 알고리즘의 측면에서 무의미한 사소한(nominal) 문제가 아님을 말해준다. 로봇을 주행시키기 위한 알고리즘은 전적으로 사용 가능한 센서 정보의 질과 종류에 의존한다. 만약 지형의 특징을 인식할 수 있는 알고리즘이 탑재된 비젼 센서와 같이 충분히 멀고 넓은 감지 범위를 가지는 센서 시스템을 장착한 로봇이라면 문제는 사소한 것이 되고 만다. 예를 들어 로봇은 이런 경우에는 처음부터 아예 교착상태나 복잡한 지역으로 진입하지 않고 그 지역을 우회하여 목표점으로 갈 것이다. 설사 교착상태에 빠진 경우에도 주위에 멀리 떨어져 있는 탈출 지점을 길고 넓은 센서 감지 범위를 이용해 쉽게 발견할 수 있다. 이를 다른 관점에서 본다면, 로봇에게 이러한 센서 시스템을 장착하는 것이 또 하나의 현실적인 해결책이기도 하다. 이 논문에서는 이 것을 배제하고 주행 알고리즘적인 측면을 다루고자 한다. Assumption 1.3은 교착상태를 해결하기 위한 최소한의 필요조건이다. 만약 로봇의 이전 단계의 주행 정보를 전혀 기억하지 않고 지역적인 센서의 현재 데이터 외에 아무런 정보도 허용되지 않는다면 교착상태에 빠졌음을 알 수 없으면 따라서 그 상태를 벗어나기도 불가능하다. 로봇 자신의 주행상태의 정보를 얻는 한 가지 방법은 주행거리계(odometry)를 사용하는 것이다. 다만 여기서 강조하고자 하는 것은 주행거리계 정보가 절대적인 자기 위치확인 데이터로 이용되지 않고 부정확한 값임을 감안하고 지역적인 형태의 정보로 사용된다는 것이다. 이렇게 얻어진 정보는 짧은 시간동안 기억장치에 기억해 둔다. 마지막으로 Assumption 2.1은 교착상태 해소 알고리즘에 필요한 공통적인 사항이다. 교착상태에 빠지지 않은 정상상태의 목표점으로 향하는 주행 전략을 가지고 교착상태를 빠져나갈 수 없으며, 교착상태 해소를 위한 주행 전략을 가지고는 목표점에 도달할 수 없기 때문에 교착상태와 정상상태에서의 주행 전략은 반드시 달라야 한다. 따라서 정상 주행 전략에서 교착 상태 탈출 전략으로 주행 전략을 바꾸기 위한 교착상태에 빠졌음을 감지하는 지점이 필요하고, 마찬가지로 교착상태 전략에서 정상 전략으로 바꾸기 위한 교착상태 탈출 지점을 알 필요가 있다. 대부분의 반사 주행 방식은 이 두 가지의 특징점을 여러 가지 다양한 형태로서 사용하고 있다.

어떤 반사주행 알고리즘 α에 대하여 교착상태에 빠지지 않은 정상상태의 목표점으로 향하는 주행 전략을 $N_n(\alpha)$라 하고, 교착상태에서의 주행 전략을 $N_d(\alpha)$, 이 지역에 들어와서 주행 전략 N_n을 사용하면 반드시 교착 상태로 빠지게 되는 지점들의 집합을 $D(\alpha, N_n)$이라고 하자. 또한 사용가능한 모든 정보 즉, 감지 범위 r을 가지는 센서로부터의 데이터와 목표점으로의 방향각, 그리고 저장된 지역적인 주행정보를 가지고 교착상태에서 정상 전략으로 바꾸는 지점의 판단 조건을 $s_e(r, \alpha)$라 하고, $R(N_d)$를 로봇이 교착상태 전략 N_d상태에서 지나게 되는 지점들의 집합이라고 하자.

Conjecture 1: 알고리즘 α에 관계없이 임의의 개수의 위치 $p \square R(N_d), p \square D(\alpha, N_n)$에서 탈출조건 $s_e(r, \alpha)$를 가지는 지형을 항상 만들 수 있다.

Theorem 1: 일반적인 임의의 형태의 지형에서 주행이 가능한 Assumption 1.1, 1.2, 1.3, 2.1을 채용하는 목표지향 반사 주행 알고리즘은 존재하지 않는다.

Proof: 로봇은 $D(\alpha, N_n)$의 정의에 의해서 오직 주행 전략 N_d에 의해서만 $D(\alpha, N_n)$에서 탈출할 수 있고, 마찬가지로 오로지 N_n에 의해서만 목표점에 도달할 수 있다. 로봇은 어떤 지점 p에서 탈출 조건 $s_e(r, \alpha)$를 감지하면 목표점으로 접근하기 위하여 주행 전략을 N_d에서 N_n으로 바꿀

것이다. 만약 지점 p가 $p \square D(\alpha, N_n)$이면 로봇은 p에서 N_n을 쓰게 되므로 다시 교착상태에 빠질 것이다. Conjecture 1에 의하여 $p \square R(N_d)$이므로 로봇은 항상 전략N_d를 가지고 지점 p를 만나게 되고, p는 탈출 조건 $s_e(r, \alpha)$를 가지므로 전략은 N_n으로 바뀌고 이때 $p \square D(\alpha, N_n)$이기도 하므로 로봇은 다시 교착상태에 빠지게 된다. 이 과정이 무한이 반복되고 결국 로봇은 성공적인 목표점으로의 주행을 수행하지 못하게 되므로 Theorem 1은 참이다.

위에서 지점 $p \square R(N_d), p \square D(\alpha, N_n)$를 거짓 교착상태 탈출점, 즉 FEOD(False End Of Deadlock Point)라 부를 수 있다. 즉, 적어도 최소한 하나 이상의 FEOD가 존재하기만 하면 로봇은 교착상태에서 빠져나올 수 없다. 그림 1이 Theorem이 말하고자 하는 바를 보여주고 있다. 우리는 로봇의 센서 시스템의 입장에서 보기에 탈출 조건 $s_e(r, \alpha)$과 똑같은 조건으로 보이는 지형은 그 탈출 조건을 결정하는 알고리즘을 고려하여 항상 교착상태로 유도되는 영역 $D(\alpha, N_n)$안의 경로 $r(N_d)$에 만들어 넣을 수가 있다. 비록 어떠한 지형의 경우는 이러한 조금은 특이한 형태라고 할 수도 있는 FEOD를 가지지 않는 경우도 있겠으나 이는 일반적으로 기대할 사항은 아니다. 그림 1에서 지점 "1"과 "2"는 동일한 목표점으로의 방향각과 센서 데이터를 갖는다. 즉, 센서의 감지 범위 r을 참고하여 보면 진짜 교착상태 탈출점 TEOD(True End Of Deadlock point) 1과 FEOD 2에서 로봇은 전적으로 동일한 정보를 얻게 된다.

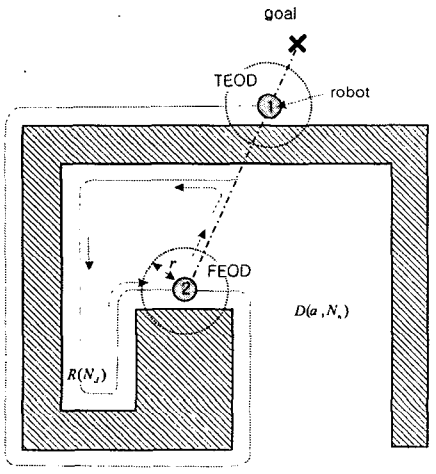


그림 1. 거짓 교착상태 탈출(FEOD)의 예
Fig. 1 An example of the shape of false end of deadlock point.

위의 분석의 유용성은 어떤 환경이 주어졌을 때 어떤 하나의 순수한 반사주행 방식 α만으로 주행 시스템을 구성할 수 있는가에 대한 판단을 할 수 있다는 것이다. 다른 관점에서 본다면 주어진 환경에서 사용할 수 있는 교착상태 해소 알고리즘을 선택하는데 유용하게 사용할 수 있다. 선택의 기준은 주어진 환경에서 그 알고리즘이 FEOD를 갖는지의 여부이다. 예를 들어 순수한 "BUG" 알고리즘[11]만을 사용한다면, 그림 1의 지점 2와 같은 FEOD를 쉽게 만들 수 있다. 이 때 $s_e(r, "BUG")$는 목표점으로의 방향각이 처음 교착상태를 감지한 지점에서의 것과 같고, 센서 데이터는 목표점 방향으로 장애물이 없고, 그 반대 쪽은 장애물이 있는 것으로 보이는 조건이다. 이 조건은 지점 1과 2에서 완벽하게 동일하다. 다른 예로, 순수한 "local target switching" 알고리즘[3]만을 사용한다면 마찬가지로 FEOD를 쉽게 만들 수가 있다. 그림 2에서 로봇이 loop 1에 갇혔을 때 로봇은 주행 전략을 바꾸어 목표점을 가상 목표점으로 바꾸어 그 가상 목표점으로 향하는 전략을 사용하게 된다. 이 전략으로 로봇은 loop 1을 탈출하여 벽을 따라가다가 열린 지점(opening point) A에 도착하여 탈출조건을 감지하여 목표점을 다시 원래의 목표점으로 바꾸어 진행하여 결국 loop 2를 통하여 목표점으로 접근하게 될 것이다. 이 때 탈출 조건 $s_e(r, "Local\ Target\ Switching")$는 오른쪽(왼쪽) 근접 센서의 데이터가 "FAR"값을 나타내는 것이다. 그러나 그림에서 점선부분과 같은 "J"자 모양의 지형을 더한다면 loop 3에 거짓 열린 지점이 존재하게 된다. 이 FEOD는 결국 로봇을 다시 교착 상태로 유도하고 만다. 따라서 이 알고리즘은 또 다른 교착상태 해소 알고리즘의 도움을 받아야만 원래 목표점으로의 도착을 보장할 수 있을 것이다.

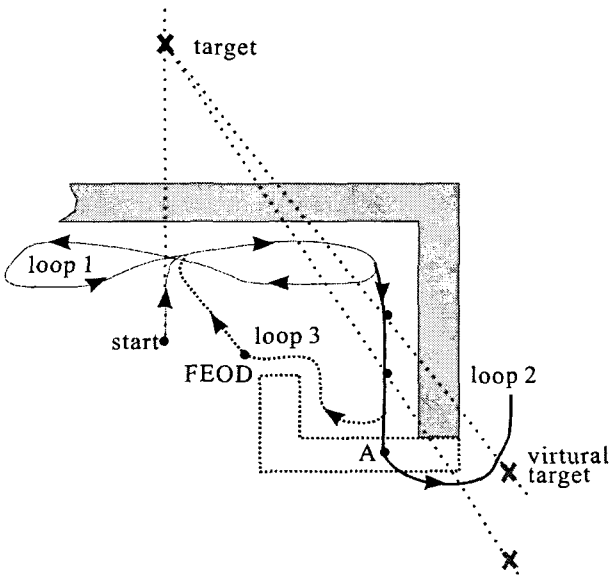


그림 2. "Local Target Switching" 알고리즘에 대한 FEOD
Fig. 2 An FEOD for the "Local Target Switching" algorithm.

또 다른 예로서 [13]에서 $s_e(r, "Yun")$는 목표점까지의 거리의 변화이다. 즉, 목표점으로의 거리는 교착상태를 감지하여 벽면 따라가기 방식으로 주행 전략을 바꾸면서 증가하기 시작한다는데 착안하여 목표점으로의 거리가 다시 줄어들기 시작하는 지점을 탈출 지점으로 선택하는 것이다. 이 전략은 [13]에서 선택한 "bench", "corner", "dead-end" 등의 형태의 지형에서는 잘 작동하지만, 역시 FEOD는 손쉽게 만들

수가 있다. 예를 들어 그림 2와 같은 "J"자형의 지형에서는 점선 지형의 시작 지점에서부터 목표점까지의 거리가 줄어들기 시작하고 로봇이 주행 전략을 다시 N_n으로 바꾼다면 교착상태로 다시 빠지므로 이 지역은 $D("Yun", N_n)$에 속한 지역이고 그 지점은 FEOD가 된다. 위의 예들에서 보듯이 순수한 목표점 지향 반사 주행 알고리즘은 일반적인 해법이 아님을 손쉽게 증명할 수가 있다.

이러한 분석에 따르면 순수한 목표점 지향 반사 주행 전략은 FEOD가 없는 환경에서만 혼자서 사용될 수 있음을 알 수 있다. 그러나 일반적인 경우, 특히 알려지지 않은 환경에서는 이를 기대할 수 없으므로 반사 주행 방식은 반사적인 방식이 아닌 전역적인 전략과 더불어 사용되어야 할 것이다. 아니면 두 번째의 차선책은 모든 지형에서 확률적으로 높은, 완벽한 해답이 아닌, 성공률을 가지도록 하는 순수한 반사 주행 방식을 개발하는 것에 만족해야 할 것이다. 경로 계획을 위한 통상의 속도 벡터에 잡음(Noise) 벡터를 더하여 교착상태를 빠져나오는 방법[15] 등이 이러한 확률적인 접근 방식의 대표적인 예이다. 이러한 전략에서는 성공적인 주행 알고리즘의 하나의 판단 기준인 최단거리 경로는 교착상태를 해결하는 댓가로 상당한 수준으로 포기하여야만 한다. 이러한 결론에서 다음 절에서부터 하나의 새로운 방식의 순수한 반사 주행 방식을 제안한다. 이 방식은 완벽하지 않지만 높은 확률의 성공적인 주행을 기대할 수 있는 방식이다.

3. 새로운 목표점 지향 반사주행 방법

제안한 주행 방식은 단순한 제어 구조로써 충돌로부터의 안전을 보장하며, 높은 성공률로 교착상태를 탈출할 수 있는 방식이다. 이 방식은 순수한 반사주행 방식으로서 로봇의 자기중심적인(ego-centric) 지역적인 센서 정보와 짧은 기간 동안 저장된 지역적인 과거의 주행 정보만을 사용한다. 한편 알고리즘의 전체적인 구조는 행위기반 로봇 제어 구조에 의하여 구성하여 자율주행에서 요구하는 모든 임무를 극히 적은 세 개의 원소를 가지는 행위 집합을 가지고 수행할 수 있도록 하였다. 전체 행위 집합은 두개의 일반적인 요소 행위(general primitive behavior)와 하나의 특화된 행위(dedicated to specific mission)의 세 가지 행위만으로 구성하였다.

3.1. 행위기반 주행 알고리즘의 전체 구조

정상적인 상태에서는 목표점 지향 방법에 따라서 로봇은 비교적 작은 장애물들을 피하면서 목표점으로의 방향각을 따라서 움직인다. 만약 로봇이 커다란 장애물이나 협곡(box canyon) 같은 형태의 어려운 지형을 만나 같은 지역에서 반복적인 주행을 계속하는 교착 상태에 빠지게 되면 그 상태를 탈출하기 위하여 벽면을 따라가는 행위를 하게 된다. 이것은 반사 주행 방식의 가장 일반적인 접근이다. 새로이 제안한 주행 방식은 이러한 접근의 새로운 구체적인 알고리즘 및 구현방식에 대한 것이다.

제안된 주행 방식은 로봇의 자율 주행을 위하여 "추출된(factored out)" 다음의 행위 집합을 기반으로 한다:

$$B = \{InSafety,\ Mv2Target,\ DetcDlock \tag{1}$$

$InSafety$와 $Mv2Target$은 요소 행위로서 주행에 필요한 대부분의 모든 임무들은 이 두 가지 요소 행위들의 적절한 조합으로 만들어 낼 수가 있다. 이러한 단순한 구조가 이 방식의 중요한 장점의 하나이다. $InSafety$는 로봇의 주행 방향과 속도를 조정하여 사물에 충돌하지 않는 안전한 상태를 항상 유지하는 역할을 한다. 이는 매 경로 계획 주기(planning step) 마다 장애물이 있어 충돌의 위험이 있는 방향으로 로봇이 이동하지 못하게 하는 방식이다. 이동 속도는 가장 가까운 장애물의 표면에도 도달하지 못하도록 주위 환경을 참조하여 결정된다. $Mv2Target$은 $InSafety$에 의해서 정해지 속도와 제한된 이동 방향 하에서 로봇을 정해진 목표점으로 향하게 한다. $Mv2Target$은 "목표점 (Target)"이 필요에 따라 변하는 매개변수변경가능(parameterizable) 행위이다. 이 두 가지의 요소 행위가 적절히 조합되어 상황에 맞게 새로이 만들어 지는 창발행위(emergent behavior)가 생성되어 장애물 회피, 속도 제어, 교착상태 해소, 목표점 도달 등의 주행에 필요한 임무들이 수행된다. 또 다른 하나의 행위 $DetcDlock$은 교착상태를 감지하는 역할을 하는 특화된 행위이다.

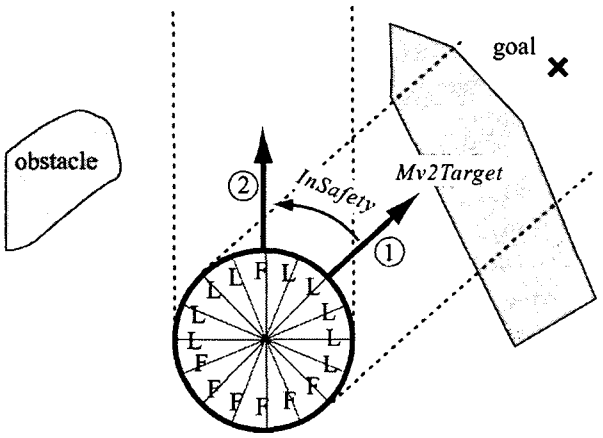
3.2. 충돌로부터의 안전 상태 유지

안전한 기동을 위하여 로봇의 속도와 이동 방향은 주위의 환경을 고려하여 결정되어야 한다. 이동 방향의 측면에서 보면 장애물 회피 방식은 두 종류로 구분될 수 있는데 [16], 하나는 열린 공간(free space)을 직접 찾는 방식이고, 다른 하나는 전위장을 이용하는 방식이다. 제안한 방식은 전자의 방식에 해당한다. 또한 로봇의 이동 방향은 몇 개의 방향으로 양자화하여 다루는 것이 현실적이며 이 때 방향은 주로 로봇의 센서 시스템의 구성에 의존한다. 예를 들어 본 논문에서는 전체의 연속적인 360° 의 방향을 로봇의 원형 몸체에 둘러진 초음파 띠(sonar ring)의 음향축(acoustic axis)에 따라 16가지의 방향으로 양자화하여 사용하였다. 행위 $InSafety$는 양자화된 각 방향이 이동하기에 안전한가의 여부를 판단하여 안전하지 못한 방향으로는 로봇이 움직이지 못하도록 즉시 가상적으로 로봇의 몸체를 방향을 잠근다 (virtual locking). $InSafety$는 다음과 같이 구현되었다:

$$\begin{aligned} &\square\, InSafety\,[17] \\ &\quad \angle \boldsymbol{v}: \text{Lock virtually the unsafe directions} \\ &\qquad \text{to move to.} \\ &\quad |\boldsymbol{v}| = d_{\min}/t_s - \beta, \end{aligned} \tag{2}$$

여기서 $\angle \boldsymbol{v}$와 $|\boldsymbol{v}|$는 각각 속도 벡터의 방향각과 크기를 나타내고, $d_{\min}$, t_s, $\beta(>0)$는 각각 최근접 장애물 표면까지의 거리, 경로 계획 주기, 충돌 방지를 위한 여분의 속도 상수이다. 어떤 특정한 방향의 안정성 여부를 판단하는 방식

은 여러 가지가 쓰일 수 있다. 본 논문에서 사용한 16개의 초음파 센서를 이용하는 방식을 부록에 첨부하였다. 다른 행위 즉, 이 논문에서는 $Mv2Targe$ 의 이동하고자 하는 방향은 이 $InSafety$에 의하여 제한을 받는다. 이 원리를 그림 3에 나타내었다. 그림 3에서 보이는 바와 같이 만약 어떤 행위가 원하는 방향이 $InSafety$에 의하여 잠겼다면 대신에 원하는 방향에서 가장 가까운 잠기지 않은 방향으로 움직이게 된다. 이러한 가상적인 로봇 몸체의 잠금 방식은 매우 간단하고도 효과적인 방식이다. 왜냐하면 이후에는 어떤 행위들을 실행할 때도 로봇은 장애물과의 충돌을 함께 고려하지 않아도 되기 때문이다. 또한 전위장을 쓸 때의 평형점, 넓은 장애물, 좁은 통로 등과 같은 비교적 약한 교착상태는 $InSatey$가 전위장 방식을 쓰지 않고 열린 공간을 찾는 방식을 사용함으로 인하여 자연스럽게 해결된다. 또한 앞에서도 언급했듯이 $InSafety$는 벽면 따라가기 같은 다른 창발 행위를 생성하는데 쓰이는 요소 행위의 역할을 한다.



L:Locked, F:Free (by *InSafety*)
① original desired direction to move by *Mv2Target*
② modified direction by *InSafety*

그림 3 행위 *InSafety*의 안전 보장 역할

Fig. 3 Safety guaranteeing role of the behavior *InSafet*

로봇의 속도는 주위 환경의 복잡도에 의해서 조정이 된다, 즉 복잡한 환경에서는 천천히 (미세 제어) 움직이고, 자유로운 공간에서는 빨리 (거친 제어) 움직이게 한다. (2)에서 속도는 가장 근접한 장애물의 표면과의 거리에 비례하도록 정해지고 있음을 볼 수 있다. 위와 같은 속도와 이동 방향의 조정을 통하여 $InSafety$는 근본적으로 로봇을 항상 충돌로부터 안전한 상태을 유지하도록 한다.

3.3. 목표점에의 도달

행위 $Mv2Target$은 로봇을 특정한 목표점의 방향으로 움직이게 한다. $Mv2Target$ 역시 요소 행위로서 여러 가지 임무를 수행하는데 기여한다. 구현 형태는 다음과 같다:

$$\square Mv2Target$$
$$\angle \boldsymbol{v} = \angle \boldsymbol{t} \quad \text{where, } \mathbf{t}: \text{target position vector from currentpoints.} \tag{3}$$

t가 전역적인 원래의 목표점으로 정해지면 이 행위는 전역적인 원래의 목표점으로 도달하게 하는 역할을 하게 된다. 정상 상태 즉, 교착상태에 빠지지 않은 상태에서는 로봇은 $Mv2Target|_{target=globalgoal}$ 에 의해서 항상 전역적인 원래의 목표점을 향하여 움직인다.

3.4. 벽면 따라가기

벽면 따라가기 알고리즘은 긴 장애물이나 교착상태에서 빠져나오는데 아주 유용하게 쓰이는 방법이다. 그러나 이 벽면 따라가기 기능을 실제로 구현하는 것은 현실적으로 그다지 쉬운 일이 아니다. 벽면 따라가기 기능은 보통 초음파나 적외선 센서 같은 근접 센서를 사용하여 구현이 된다. 수 많은 종류의 벽면 따라가기 알고리즘이 이미 발표되었다. Turennout[14]와 Bemporad[18]는 벽면으로부터 정해진 일정 거리를 유지하면서 벽면을 따라가기 위하여 로봇의 현재 위치와 방향각(heading angle)의 피드백(posture feedback)방식을 사용하였다. 여기서 두 경우 모두 다 직선 형태의 매끄러운 벽면과 완벽한 전역적인 로봇의 위치와 방향각 즉, (x, y, θ)을 안다고 가정하였다. Tustel[19]은 같은 목적을 위하여 퍼지로직을 사용하기도 하였다. 일반적인 형태의 벽면의 경우에 대해서는 Ando[20]가 벽면의 형태를 "전방 벽면(front wall)", "우측전방 벽면(front-right wall)" 등의 몇 가지로 분류하여 각 형태에 맞는 벽면 따라가기 알고리즘을 따로 구현하였다. 또 다른 방식은 벽면의 형태 등에 관계없이 일반적인 벽면 따라가기를 할 수 있는 핵심 행위를 구현하는 것이다 [21]. 이 방식은 본 논문의 방식이기도 하다. [21]에서 Yata는 로봇으로부터 장애물의 가장 가까운 지점으로의 벡터에 직각인 방향으로 움직인다는 전략으로 간단하면서도 효과적인 방식을 제안하였다. 이 방식은 모든 형태의 벽면을 잘 따라갈 수 있으나 단점으로 원래 따라가던 벽면과 인접한 다른 벽면이 더 가까울 경우 다른 벽면으로 갈아타는 경우가 생길 수 있고, 복잡한 환경에서는 잘 동작하지 않는다는 점이 있다. 본 논문에서는 새로운 방식의 벽면 따라가기 기능을 요소 행위인 $InSafety$와 $Mv2Target$을 이용하여 구현하였다. 이 방식은 아주 단순한 구조로 만들어 졌으며 동시에 위에서 소개한 기존의 방식들의 단점을 모두 극복하였다. 제안된 벽면 따라가기 기능은 $Mv2Target$의 "target"을 (4)에서 보는 바와 같은 값으로 정하여 구현한다.:

$$\square \text{Wall} - \text{following}$$
$$\angle \mathbf{v} \text{ is made by Mv2Target with } \angle \boldsymbol{t} = \angle \boldsymbol{c} \pm \pi/4, \text{ the robot.} \quad \textit{where, } \boldsymbol{c}: \text{current position vector of therobot.} \tag{4}$$

(4)의 벽면 따라가기 방식 역시 *InSafety*에 의해서 제약을 받으면서 실행이 된다. 이렇게 *InSafety*와의 조합으로 특별한 형태의 벽면에 제한되지 않고, 다른 인접한 벽면으로 갈아타지 않으며, 복잡한 공간에서도 원하는 벽면을 잘 따라가는 행위를 구현하였다. 이 벽면 따라가기 알고리즘을 그림 4에 설명하였다. *Mv2Target*은 로봇을 우측전방 (또는 좌측전방)으로 향하도록하고, 이는 *InSafety*에 의하여 우측전방에서 가장 가까운 장애물이 없는 방향으로 변형되면서 자연스럽게 벽면에서 떨어지지 않고 일정한 거리를 유지하면서 벽면과 평행한 방향으로 움직이게 된다. 이 벽면 따라가기 방식은 다음에 설명하는 바와 같이 교착상태를 탈출하는 데에도 쓰이게 된다. 한편 (4)의 벽면 따라가기 방식은 구현의 측면에서 볼 때 실제적인 강점을 가진다. 왜냐하면 두개의 요소 행위들로 간단히 구현이 되며, 벽면의 형태를 고려할 필요가 없고, 벽면에서 일정거리 이상 떨어지지 않는가를 주시할 필요가 없고, 벽면과 평행하게 진행하는 문제도 자연스럽게 해결되기 때문이다. 이 모든 벽면 따라가기의 요구 사항이 단순히 두개의 요소 행위 *InSafety*와 *Mv2Target*에 의하여 만들어 지는 것이다.

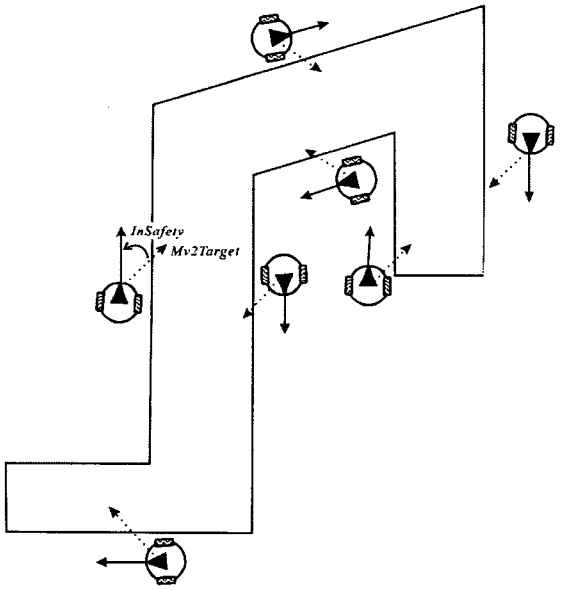


그림 4 행위 *Mv2Target*와 *InSafet* 에의한 벽면 따라가기

Fig. 4 Wall-following by *Mv2Target* and *InSafet*

3.5. 교착상태 해소

교착상태를 해소하기 위한 알고리즘은 교착상태 감지와 탈출의 두 가지 작업으로 이루어진다. 앞에서 증명했듯이 교착상태를 해소하는 완벽한 반사 주행 방식은 존재하지 않는다, 본 논문에서 제안하는 방식은 확률적으로 높은 성공률을 기대할 수 있는 방식이며, 이 때 최단 주행거리라는 최적성은 포기되었다.

1) **교착상태 감지 방법** : 만약 로봇이 목표점 방향각외에 아무런 전역적인 정보를 사용하지 않는다면 교착상태를 감지하기 위해서 반드시 지역적인 과거의 주행 정보를 일정량 기억하여 사용하여야 한다. 본 논문에서는 연속하는 두 계획 주기에서의 주행거리계에 의한 위치의 차를 이용하였다. 행위 *DetcDlock*의 교착상태 감지 알고리즘은 일정 시간 동안의 연속하는 두 계획 주기에서의 위치의 차들의 총합은 교착상태에 빠졌을 때 "0"-실제 구현에서는 "0"에 가까운 값-을 가지게 된다는 사실을 바탕으로 제안되었다:

$$\left|\sum_{k=0}^{W-1} x(t-k)-x(t-k-1)\right| + \left|\sum_{k=0}^{W-1} y(t-k)-y(t-k-1)\right| \approx 0 \quad (5)$$

where, $x(t)$, $y(t)$: position coordinate at time t obtainedbyodometry,
W: width of time window

(5)는 주행거리계에 의한 자기위치확인 정보의 오류가능성을 고려한 식이다. 만약 이 자기위치확인 정보가 이상적으로 정확하다면 다음의 (6)과 같이 구현될 것이다:

$$[x(t), y(t)]^T \approx [x(t-W), y(t-W)]^T \quad (6)$$

그러나 바람직한 교착상태 감지 알고리즘은 주행거리계와 같이 누적 오차를 포함하는 자기위치확인 정보의 정확성에 되도록이면 적게 의존하는 것이다. (6)은 두 위치에서의 전역적인 원점에서부터 누적되어 계산되어온 절대 위치를 기반으로 하는 반면 (5)는 절대 위치에 의존하지 않고 단순히 연속된 위치의 차만을 사용하기 때문에 보다 적은 교착상태 감지 오차를 가지게 된다. (5)는 다음에서 보는 바와 같이 교착상태 감지와 탈출의 두 가지 알고리즘에 동시에 사용되므로 본 논문에서 교착상태를 다루는 방식의 핵심 사항이다. 그림 5에 (5)의 원리를 설명하였다. 그림 5에서 로봇이 교착상태에 빠지면 반드시 그 지역에서의 반복적인 주행이 일어나고, 그 반복적 주행의 경로는 모양은 일정하지 않지만 어떤 형태이든 모두 폐곡선을 이루게 된다. 반복되는 경로의 폐곡선의 구체적인 모양은 중요하지 않다. 다만 경로의 시작점과 끝점이 거의 일치하게 되므로 (5)는 참임을 알 수 있다. (5)를 사용한 교착상태 감지의 성공 확률은 사용가능한 시간축 단계의 수 $W_{\max}$가 커짐에 따라 증가하게 된다. 그러나 반면 필요로 하는 메모리의 양 역시 증가하는 단점이 있다. 이 장단점을 (7)에 정리하였다:

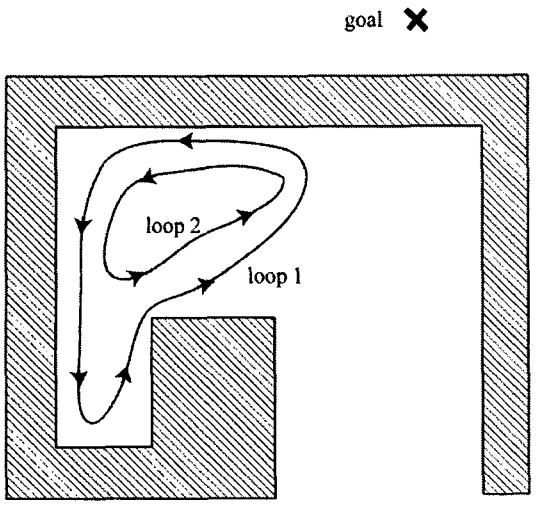


그림 5 교착상태에서의 반복 주행
Fig. 5 Limit cycle in deadlock region

$$P_{detect} \propto W_{max},\ M_{required} \propto W_{max}$$
$$where,\ P_{detect}: \text{probability to detect deadlock},\quad W_{required}: \text{amount of memory required} \tag{7}$$

그림 5에서 loop 1이 loop 2보다 긴 시간을 가진다고 하자. 제안된 감지 알고리즘의 W_{max}가 loop 2에 해당하는 값으로 작게 맞추어져 있다면 loop 1의 교착상태는 감지하지 못할 확률이 높다. 결과적으로 (7)의 감지율과 필요한 메모리양의 장단점을 고려하여 W_{max}를 정해야 할 것이다.

2) **교착상태 탈출 알고리즘** : 기존의 많은 교착상태 탈출 알고리즘은 전형적으로 교착상태를 나타내는 비교적 쉬운 환경에서 주로 효과적으로 빨리 탈출하는 것을 목표로 삼고 있으며, 그 대상으로 삼은 지형에서 효과적인 성능을 보였다. 그러나 알고리즘이 적용된 지형들은 앞에서 언급한 바와 같이 치명적인 형태 즉, FEOD를 포함하지 않고 있다 [3], [11], [13]. 본 논문에서는 이러한 기존의 방식과 다르게 치명적인 FEOD를 포함하는 일반적인 모든 종류의 지형에 적용되며, 확률적으로 높은 성공률로 교착상태를 탈출할 수 있는 방식을 제안한다. 교착상태의 미로를 탈출하기 위한 많은 방식이 있지만 본 논문은 우측(좌측)벽면 따라가기(right hand-contact wall following) 방식을 채택하였다. 왜냐하면 짧은 감지거리의 센서를 가지고 알려지지 않은 임의의 형태의 미로를 빠져나가는데 가장 신뢰할 만한 방식이 벽면 따라가기 방식이기 때문이다. 이 전략은 [11]에 잘 설명되어 있다. 그러나 Theorem 1에서 본 것과 같이 순수한 벽면 따라가기 만으로 일반적인 교착상태의 미로를 빠져나갈 수는 없으며, 따라서 본 논문에서는 다음과 같은 알고리즘을 제안한다.

제안한 알고리즘의 주요 착안점은 Theorem 1에 따라 거짓 탈출 조건,즉 FEOD를 교착상태의 미로 안에서 감지하는 것이다. 이를 위하여 로봇은 처음 탈출조건을 감지했을때 (5)의 교착상태 감지 알고리즘을 초기화하여 바로 다시 시작하는 것이다. 그런 다음 다음으로 탈출조건이 감지되었을 때 (5)의 합을 검사한다. 이 때 합이 0에 가까운 값이라면 그 지점은 이전의 탈출 조건이 감지된 지점이고, 따라서 로봇은 그동안 같은 loop를 돈 것이 되므로 그 지점은 FEOD일 가능성이 아주 큰 지점이다. 이렇게 (5)를 사용하여 FEOD를 감지하면 로봇은 이를 무시하여 정상상태의 주행전략으로 바꾸지 말고 그대로 벽면 따라가기를 계속하면 된다. 그림 6에 이러한 "거짓 탈출 조건 배제 (false escape-screening)" 알고리즘을 도해하고 있다. 탈출 조건은 목표점과 최초의 교착상태 감지점 "0"을 포함하는 점선상에 로봇이 오게 될 때 감지된다. 지점 "0"에서 최초로 교착상태가 감지되었다고 하면, 로봇은 다음의 탈출 조건이 감지되는 지점 "1"에 도달하기까지 벽면 따라가기를 수행한다. "1"에서 탈출조건이 감지되므로 로봇은 다시 정상상태의 목표점을 향하는 주행전략으로 바꾸고 동시에 교착상태 감지 알고리즘 (5)를 다시 초기화하여 합을 시작한다. 로봇은 다시 교착상태에 빠져 다시 이전과 같은 벽면을 따라 loop A를 이동하게 될 것이며 다시 "1"의 지점을 만날 때 (5)의 합이 0에 가까운 값이 되므로 "1"이 FEOD임을 알게된다. 따라서 "1"의 탈출 조건을 무시하고 loop C를 따라 벽면 따라가기를 시행한다. 또다른 FEOD "2"도 이와 같은 방식으로 무시하고 지나쳐 벽면 따라가기를 계속하여 결국 진짜 탈출 지점인 "3"에서 교착상태를 빠져나가게 된다.

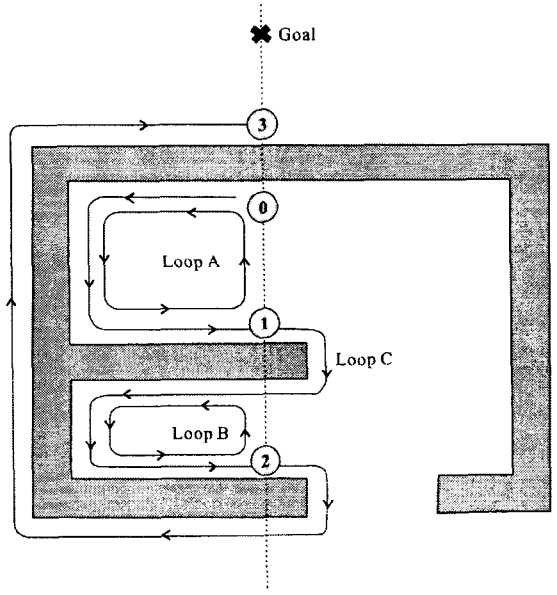


그림 6 교착상태 해소를 위한 "거짓 탈출 조건 배제" 알고리즘
Fig. 6 The false escape-screening algorithm for escaping from deadlock

결론적으로 교착상태 감지 알고리즘 (5)가 이상적으로 완벽하다면 로봇은 임의의 형태의 교착상태를 완벽하게 탈출할 수 있을 것이다. 즉, 오차의 요인이 큰 주행거리계 등을 사용

한 실제 구현시의 (5)의 성공률이 전체 반사 주행 알고리즘의 성공률을 결정하게 되는 것이다. 보다 일반적으로 말해 FEOD를 감지하는 도구의 현실적인 불완전성이 목표점 지향 반사 주행 알고리즘의 불완전성을 초래하는 것이다. 이 것이 바로 Theorem 1이 의미하는 내용이다.

4. 실험 결과

제안한 반사 주행 방법을 알려지지 않은 실내 환경에 적용하여 실험하였다. 주행 환경은 자율주행이 요구하는 여러 가지 특징적인 형태의 지형과 임의로 분포하는 장애물들이 있는 사무실 환경을 모방하여 만들었다. 또한 실험실의 복도 같은 실제 환경에서도 실험을 실시하였다. 직접 제작한 이동 로봇 MARI(Mobile Agent with Realistic Intelligence)는 둥근 원통형의 몸체를 가지고 있으며 몸체 둘레에 16개의 초음파 센서를 고르게 배치시켰다. 목표점의 방향각과 목표점 인식 기능은 현재로서는 오직 주행거리계를 사용하여 구현하였다. 차후에 이를 단순한 비젼 알고리즘을 추가하여 성능을 개선할 계획이다. 그림 7에 로봇의 모습과 제작된 주행 환경을 보였다.

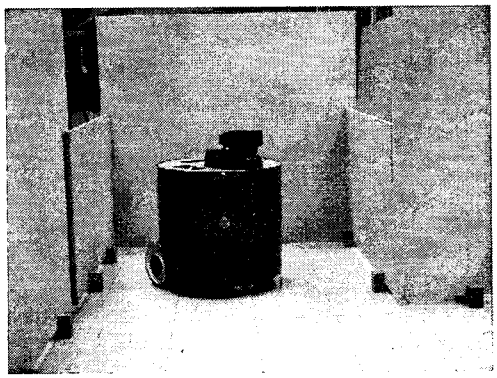


그림 7 자체 개발 로봇 MARI
Fig. 7 The developed robot "MARI".

4.1. 벽면따라가기 성능

그림 8에 실제 실험실의 복도에서 행한 실험으로부터 얻은 주행 위치들을 샘플링하여 연결한 주행 궤적을 표시하였다. 실험시 0.5m로 설정한 벽면과의 거리를 비교적 잘 유지하면서 성공적으로 벽면 따라가기를 수행하고 있음을 볼 수 있다.

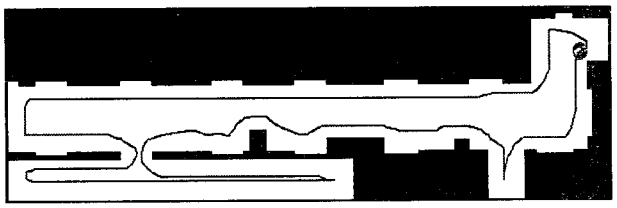


그림 8 복도에서의 벽면 따라가기 결과
Fig. 8 Wall-following performance in a corridor.

4.2. 교착상태 해소 성능

그림 9는 (5)를 사용하여 교착상태를 감지하여 제안된 거짓 탈출 조건 배제 방식을 적용하여 교착상태를 해소하는 실험의 예를 보이고 있다. 실험 환경은 두개의 FEOD를 포함하고 있음을 볼 수 있다.

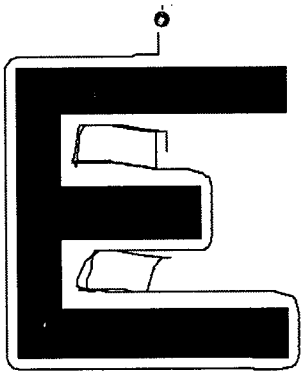


그림 9 교착상태 감지 및 해소 실험 결과
Fig. 9 An example of deadlock detection and resolving with two FEOD.

4.3. 전체 성능 평가

자율주행에 필요한 각종 판단 기준에 대한 성공 혹은 실패율을 집계하였다. 표 1에 설정한 자율주행 임무에서의 판단 기준을 요약하였다. 또한 표 2에 여러 번의 실험과정에서 얻은 데이터로부터 얻은 전체적인 성능의 통계 값을 나타내었다. 표 2에서 실패율은 표 1의 기준을 참조하여 다음 (8)에 의하여 결정되었다. 다만 속도 제어는 예외이다.

$$failute\ rate = \frac{\#\ of\ failure \times 100}{\#\ of\ corresponding\ situation}[\%] \quad (8)$$

표 2에서 보는 바와 같이 제시된 알고리즘이 여러 가지 다양한 임무에 대하여 성공적인 자율주행을 할 수 있음을 보여주고 있다. 장애물 회피 성능이 예상했던 대로 가장 높은 성공률은 보여주고 있다. 이는 임무 자체가 쉬운 측면과 다른 행위들에 비하여 $InSafety$에게 최우선으로 실행될 우선권을 주었기 때문임을 알 수 있다. 교착상태 해소도 비교적 높은 성공률을 보이고 있다. 여기서 다시 한번 강조할 것은 순수한 목표점 지향 반사 주행 방식만으로는 완벽한 교착상태 해소 성능을 기대할 수 없다는 것이다. 몇몇 불만족한 성능 데이터는 모두 센서 데이터의 질에서 비롯된 것이다. 이는 이상적인 센서 정보를 가정한 시뮬레이션에서 보이는 완벽한 성능을 볼 때 알고리즘의 측면에서는 오차가 없음을 알 수 있기 때문이다. 한편 표 2의 교착상태 해소 통계는 감지와 추정 이론 (detection and estimation theory [22])에서 말하는 type I 오차 즉, 교착상태가 발생했음에도 이를 감지하지 못하는 감지 실패 (missing detection) 오차를 나타낸 것이다.

표 1 자율주행의 성능 판단 기준

Table 1 Criteria for autonomous navigation

Criteria	Condition	Success	Failure
Avoiding obstacle	When obstacles exist within $t_s\|v\|$	If doesn't collide	If collide
Detecting deadlock	When cyclic behavior occurs	If detect deadlock	If cannot detect
Resolving deadlock	When deadlock is detected	If escape from deadlock	If cannot escape
Speed Control		Measured by the ratio of (current velocity to maximum velocity)/(degree of clutter of the faced environment)	
Arriving at goal	When start	If arrive at goal	If cannot arrive

표 2 실험 결과에 대한 통계:실패율 [%]

Table 2 Some examples of statistics of performance: Failure rate [%]

Criteria / Episodes	Avoiding obstacle	Detecting deadlock	Resolving deadlock	Speed Control	Arriving at goal
1	3/47=6.4	3/21=14.3	4/18=22.2	86	0/3=0
2	3/61=4.9	5/26=19.2	3/21=14.3	80	0/6=0
3	1/52=1.9	4/21=19.0	3/17=17.6	85	1/9=10.1
4	2/50=4.0	3/24=12.5	2/21=9.5	76	1/4=25
5	1/35=2.9	3/18=16.7	4/15=26.7	83	0/4=0

반면 type II 오차 즉, 교착상태가 발생하지 않았는데도 감지했다고 판단하는 거짓 감지 (false alarm) 오차는 지극히 낮은 비율로 발생하였다. 이는 보통의 주행 상태의 대부분을 차지하는 교착상태에 빠지지 않은 정상상태에서는 교착상태 해소 알고리즘에 의하여 방해 받지 않고 순탄하게 목표점으로 접근함을 말해 준다.

표 2의 교착상태 감지 결과는 $W_{\max} = 100\,[\text{sampling} \times]$로 고정했을 때의 결과이다. $W_{\max}$가 커짐에 따라 (7)에서 기대한 것처럼 감지율도 같이 증가하는 것을 확인할 수 있다. 이 (7)의 경향성을 그림 10에 나타내었다.

5. 결 론

본 논문은 목표점 지향 반사 주행 방식의 한계를 명시한 중요성이 있다. 임의의 형태의 모든 지형에 적용될 수 있는 순수한 목표점 지향 반사 주행 방식의 존재에 대한 공통적인 의문점에 대한 해답을 제시하였다.

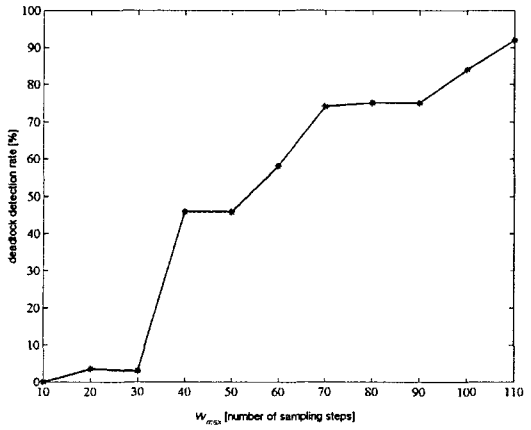


그림 10 최대 사용 가능한 이동 합의 크기에 따른 교착상태 감지율

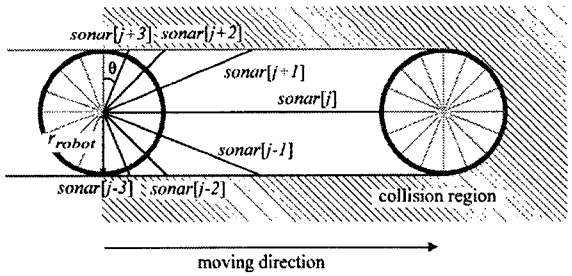
Fig. 10 Deadlock detection rate with maximum available width of time window for moving sum.

또한 이 분석을 바탕으로 하여 새로이 충돌로부터의 안전성을 보장하고 교착상태를 해소할 수 있는 순수한 반사 주행 방식을 제한하였다. 제안된 방식은 장애물 회피, 목표점 도착, 벽면 따라가기, 교착상태 해소 등 자율 주행에 필요한 모든 기능을 단 세 개의 행위 *InSafety*, *Mv2Target*, *DetcLlock*를 가지고 구현하였다. 이 구조는 매우 단순하면서도 효과적인 것이다. 특히 충돌위험 방향으로 움직이지 못하도록 로봇의 몸체를 가상적으로 잠그는 전략에 의한 안전 상태 유지 알고리즘과, 새로이 제안된 교착상태 감지 방식과 벽면 따라가기 방법을 통한 교착상태 해소 알고리즘은 성공적으로 그 유용성을 보여주었다. 따라서 제안된 반사 주행 방식에 대한 한계 분석과 새로운 주행 방식은 알려지지 않은 환경에서의 주행 시스템을 설계하는데 도움을 줄 것이다. 향후에는 보다 정확한 목표점 방향각과 목표점 인식을 위하여 단순한 비젼 시스템을 추가하여 제안된 주행 방법의 성능을 높일 계획이다.

부 록

초음파센서를 이용한 특정 방향의 충돌 가능성 여부의 판별법

본 논문에서 사용한 원통형 로봇 몸체에 균일하게 배치한 16개의 초음파 센서를 이용하여 각 초음파센서의 음향축(acoustic axs)을 기준으로한 16개의 방향에 대한 충돌 위험을 판별하는 규칙을 다음 (A.1)과 같이 세웠다.



그림A. 1 초음파센서를 이용한 특정 방향의 충돌 안전 여부 판별

Fig.A. 1 Deciding whether the direction corresponding to jth sonar will be locked or not.

(A.1)은 장애물의 표면이 심하게 돌출된 곳이 없이 비교적 매끄럽다고 가정하였을 때, 이동중에 충돌로부터의 안전을 보장받으려면 진행하고자 하는 방향의 초음파 센서의 데이터가 한 계획 주기 동안 예상되는 움직일 거리보다 길어야 하고, 동시에 그 주변의 모든 6개의 초음파 센서의 값도 각각의 초음파의 방향으로 로봇의 진행시 휩쓸고 갈 최소한의 거리보다 커야 함을 의미한다.

Moving direction $j, j=0,1,1,\cdots,15$, is locked in oreder not to move to

if any one of following conditions is true;

condition 1 : $sonar[j] \le |v|t$

condition 2 : $sonar[j-k] + r_{robot} \le \frac{r_{robot}}{cos((4-k)\theta)},\ k=1,2,3$

condition 3 : $sonar[j+k] + r_{robit} \le \frac{r_{robot}}{\cos((4-k)\theta)},\ k=1,2,3$

where, sonar$[j]$: the measured distance by jth sonar sensor,

sonar$[j \pm k]$: the kth neighbor sonar values from j, $j \pm k$ result in the circular index within 1 to 15

r_{robot} : robot radius

$|v|$: decided speed for next step,

t : sampling periode

θ : equally distributed angle of sonar sensors, 22.5° here

참 고 문 헌

[1] J. M. Roberts, E. S. Duff, P. I. Corke, P. Sikka, G. J. Winstanley, and J. Cunningham, "Autonoumous control of underground mining vehicles using reactive navigation," in Proc. IEEE Intern. Conf. Robotics and Automation, San Francisco, CA, 2000, pp. 3790-3794.

[2] A. Mukerjee, and A.D. Mali, "Reactive robots and amnesics: A comparative study in memoryless behavior," IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 292, No. 2, pp. 216 -226, May 1999.

[3] W. L. Xu, and S. K. Tso, "Sensor-based fuzzy reactive navigation of a mobile robot through local target switching," IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 29, NO. 3, pp. 451-459, August, 1999.

[4] M. Piaggio, G. Vercelli, R. Zaccaria, "A reactive sensor-based system for solving navigation problems of and autonomous robot," in Proc. IEEE/RSJ Intern. Conf. Intelligent Robots and Systems, IROS'97, Grenoble, France, 1997, Vol. 1, pp. 238 -243.

[5] M. Alwan, and P. Y. K. Cheung, "Combining goal-directed, reactive and reflexive navigation in autonomous mobile robots," in Proc. Australian New Zealand Conf. Intelligent Information Systems, Adelaide,

Australia, 1996, pp. 346 349.

[6] J. R. Asensio, J. M. M. Montiel, and L. Montano, "Goal directed reactive robot navigation with relocation using laser and vision," in Proc. IEEE Intern. Conf. Robotics and Automation, Detroit, Michigan, 1999, pp. 2905-299.

[7] J. Vandorpe, H. V. Brussel, and H. Xu, "LiAS: A reflexive navigation architecture for an intelligent mobile robot system," IEEE Trans. Industrial Electronics, Vol. 43, NO. 3, pp. 432-440, June, 1996.

[8] H. R. Everett, Sensors for Mobile Robots. Theory and Application, Wellesley, Massachusetts: A K Perters, Ltd., 1995.

[9] R. Pfeifer, and C. Scheier, Understanding Intelligence, MA: MIT Press, 1999.

[10] Y. Koren, and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in Proc. IEEE Intern. Conf. Robotics and Automation, Sacramento, CA, 1991, pp. 1398-1404.

[11] V. J. Lumelsky, and A. A. Stepanov, "Path Planning Strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," Algorithmica, 2, pp. 403-430, 1987.

[12] J. O. Kim, and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," IEEE Trans. Robotics and Automation, Vol. 8, NO. 3, pp. 338-349, June, 1992.

[13] X. Yun, and K.-C. Tan, "A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning," in Proc. IEEE intern. conf. Robotics and Automation, Monterey, CA, 1997, pp.421-426.

[14] P. van Turennout, and G. Honderd, "Following a Wall with a Mobile Robot using Ultrasonic Sensors," in Proc. IEEE/FSJ Int'l Conf. Intelligent Robots and Systems. Raleigh, NC, 1992, PP. 1451-1456.

[15] R. C. Arkin, "Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior," in Proc. IEEE Intern. Conf. Robotics and Automation, Raleigh, NC, 1987, pp.264-271.

[16] P. C. Y. Sheu, and Q. Xue, Intelligent Robotic Planning System, Farrer Road, Singapore: World Scientific, 1993.

[17] D. Y. Yoon, S. R. Oh, G. T. Park, and B. J. You, "A Behaviour-Based Approach to Reactive Navigation for Autonomous Robot," in Proc. 15th IFAC World Congress, Barcelona, SPAIN, 2002.

[18] A. Bemporad, M. Di Marco, and A. Tesi, "Wall-Following Controllers for Sonar-Based Mobile Robots," in Proc. 36th Conf. Decision & Control, San Diego, CA, 1997, pp. 3063-3068.

[19] E. Tunstel, and M. Jamshidi, "Embedded Fuzzy Logic-Based Wall-Following Behavior for Mobile Robot Navigation," in Proc.First Intern. Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference. The Industrial Fuzzy Control and Intelligent Systems Conference, and the NASA Joint Technology, NAFIPS/IFIS/NASA '94, San Antonio, Texas, 1994, pp. 329 330.

[20] Y. Ando, and S. Yuta, "Following a Wall by an Autonomous Mobile Robot with a Soanr-Ring," in Proc. IEEE Intern. Conf. Robotics and Automation, Nagoya Aichi, Japan, 1995. pp.2599 2606.

[21] T. Yata, L. Kleeman, S. Yuta, "Wall Following Using Angle Information Measured by a Single Ultrasonic Transducer," in Proc. IEEE Intern. Conf. Robotics & Automation, Leuven, Belgium, 1998, pp. 1590-1596.

[22] T. K. Moon, and W. C. Stirling, Mathematical Methods and Algorithms, Upper Saddle River, NJ: Prentice Hall, 2000.

저 자 소 개

오 용 환(吳勇煥)

1991년 포항공과대학교 기계공학과 (공학사). 1993년 포항공과대학교 기계공학과 (공학석사). 1999년 포항공과대학교 기계공학과 (공학박사). 2001년~현재 KIST 선임연구원. 관심분야는 인간형 로봇 플랫폼 및 메커니즘, 센서기반 동적 보행, 로봇 매니퓰레이터의 강인한 컴플라이언스 제어, 시각 및 촉각 기반 로봇 매니퓰레이터의 제어

윤 도 영(吳尙錄)

1995년 서강대학교 전자공학과 (공학사). 1997년 서강대학교 전자공학과 (공학석사). 2004년 고려대학교 전기공학과 (공학박사), 현재 대우조선해양(주) 로봇연구소 선임연구원. 관심분야는 회로 및 시스템, 지능 로봇.

오 상 록(尹道榮)

1980년 서울대학교 전자공학과 (공학사). 1982년 한국과학기술원 전기 및 전자 공학과 (공학석사). 1987년 한국과학기술원 전기 및 전자공학과 (공학박사). 1988년~현재 KIST 책임연구원. 2003년~현재 정보통신부 지능형서비스로봇 프로젝트 메니저. 관심 분야는 robotics and automation, intelligent control and machine learning, advanced Control system theory.

박 귀 태(朴貴泰)

1975년 고려대학교 전기공학과 (공학사). 1977년 고려대학교 전기공학과 (공학석사). 1981년 고려대학교 대학원 전기공학과 (공학박사). 1981년~현재 고려대학교 전기공학과 교수. 현재 IBS Korea회장. 관심분야는 intelligent controls, image processing, mobile robotics, home automation. intelligent building system(IBS).