

최적 경유점을 갖는 전역 DWA 에 기반한 이동로봇의 주행

Mobile Robot Navigation based on Global DWA with Optimal Waypoints

함종규, 박종태, 송재복*
(Jong-Gyu Ham, Joong-Tae Park, and Jae-Bok Song)

Abstract : The dynamic window approach(DWA) is a well known technique for reactive collision avoidance. It shows safe and efficient performance in real-world experiments. However, a robot can get stuck in local minima because no information about the connectivity of the free space is used to determine the motion. The global DWA can solve this problem of local minima by adding a navigation function. Even with the global DWA, it is still difficult for a robot to execute an abrupt change in its direction, for example, entering from the corridor to a doorway. This paper proposes a modified global DWA using the included angles of waypoints extracted from an optimal path. This scheme enables the robot to decelerate in advance before turning into the doorway. Therefore the robot can reach the goal position more safely and efficiently at high speeds.

Keywords : real-time obstacle avoidance, gradient method, global dynamic window approach, mobile robot

I. 서론

자율 이동로봇의 주행을 위하여 로봇은 운동 알고리즘을 사용하는데, 이는 크게 경로계획과 장애물회피 등으로 나눌 수 있다[1]. 경로계획 알고리즘은 환경에 대한 정보를 지도 형태로 미리 알고 있다고 가정하고 지도에 표시된 장애물과 충돌하지 않는 최적의 경로를 로봇의 현재 위치에서 목표점 까지 생성한다. 그러나 로봇이 환경을 언제나 완전하게 알 수는 없으며, 환경 역시 시간에 따라서 변화하므로, 경로계획 법만을 이용한 주행에서는 지도를 갱신하여도 지도갱신과 경로계획의 수행에 소요되는 시간 때문에 동적 장애물 회피에는 어려움이 따른다. 그러므로 동적 환경에서 장애물과의 충돌을 방지하기 위해서는 실시간 장애물 회피 알고리즘이 필요하다. 일반적으로 실시간 장애물 회피에서는 로봇의 장애물에 대한 회피 반응을 빠르게 하기 위하여 실시간으로 얻는 센서정보로부터 장애물과 충돌하지 않는 로봇 운동을 결정한다.

초기의 실시간 장애물 회피 알고리즘은 장애물로부터의 척력과 목표점에서의 인력을 이용한 포텐셜 필드 방식[2]을 기반으로 로봇이 안전할 수 있는 거리를 유지하도록 하여 장애물과의 충돌을 방지하였다. 이후 센서정보를 이용하여 장애물 점유확률을 1차원의 도수분포(frequency distribution)로 변환시켜 장애물의 점유확률이 작은 곳으로 주행하는 Vector Field Histogram approach(VFH)[3], 로봇의 물리적 구속조건이 포함되어 장애물을 회피할 수 있는 속도를 선택하는 Curvature Velocity Method(CVM)[4]가 제시되었다.

현재 일반적으로 사용되고 있는 실시간 장애물 회피 알고

리즘으로는 Dynamic Window Approach(DWA)가 있다[5]. DWA는 로봇의 기구학적 구속조건을 포함시켜, 로봇에 의하여 변화할 수 있는 속도영역에서 로봇과 충돌하는 장애물을 회피 하면서 목표점으로 빠르게 주행할 수 있는 속도를 선택하는 방법이다. 결과적으로 DWA는 적은 연산량에 의한 빠른 수행속도로 실제 환경에서 우수한 장애물 충돌 회피 성능을 보인다. 그러나 실시간 장애물 회피 알고리즘만을 이용한 로봇 주행은 전체 공간에 대한 정보 결여로, 주행 중 국부최소(local minimum)에 빠져 목표점에 도달하지 못할 수 있는 단점을 가지고 있다. 이러한 국부최소 문제는 Brock[6]이 제시한 경로계획법인 주행함수(navigation function)[7]를 이용하여 환경정보를 추가한 전역 DWA에 의하여 보완할 수 있다.

그러나 전역 DWA는 많은 경우에 경유점 근처에서 적절한 감속을 하지 못하여 경로를 벗어나 크게 회전하거나, 좁은 영역에서 진입에 실패하는 문제점을 갖고 있다. 이에 본 논문에서는 국부최소에 빠지지 않으며 최적경로를 계획하는 구배법(gradient method)[8]에서 경유점을 추출하고, 경유점들의 사이 각도를 이용하여 감속이 필요한 부분에서 충분히 감속할 수 있는 전역 DWA를 제안한다. 이로써 로봇은 보다 안정적인 고속주행을 할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 DWA를 소개하며, 3장에서는 최적경로 계획이 가능한 구배법을 설명한다. 4장에서는 구배법에서 경유점을 추출하는 방법과 경유점 사이 각도를 이용하는 전역 DWA를 소개한다. 5장에서는 경유점만을 이용한 전역 DWA와 사이 각도를 이용하여 감속이 가능한 전역 DWA의 차이를 실험을 통하여 비교하며, 6장에서 결론을 도출한다.

II. Dynamic Window Approach(DWA)

DWA에서 dynamic window란 로봇을 구동하는 액츄에이터의 한계에 의해서 제한된 전체 속도영역 중에서 로봇의 현재 속도를 기준으로, 그림 1의 점선 안의 흰색영역과 같이 단위 시간(Δt) 동안 최고 가속도(a_{max})에 의하여 변화할 수 있는 v_r 과 v_f 영역을 의미한다.

* 책임저자(Corresponding Author)

논문접수 : 2006. 10. 30., 채택확정 : 2007. 4. 4.

함종규, 박종태 : 고려대학교 메카트로닉스 협동과정 대학원
(maze09@korea.ac.kr/geullu@korea.ac.kr)

송재복 : 고려대학교 기계공학과(jbsong@korea.ac.kr)

※ 본 연구는 산업자원부 지원으로 수행하는 21C 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었음.

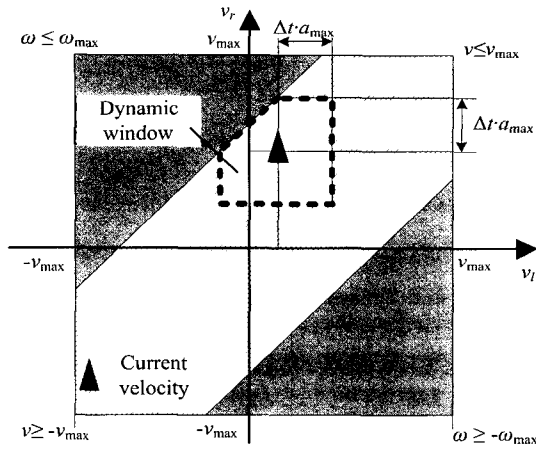


그림 1. Dynamic window.
Fig. 1. Dynamic window.

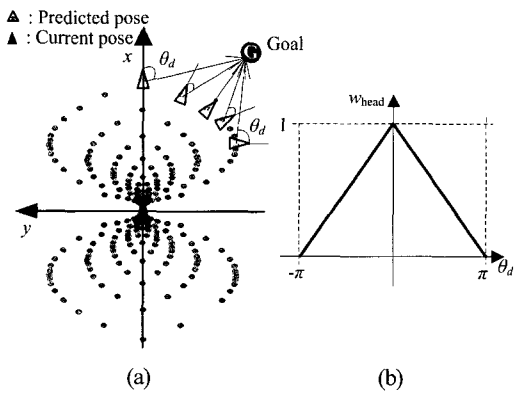


그림 2. Heading objective(w_{head})의 계산.
Fig. 2. Computation of heading objective(w_{head}).

여기서, v_{max} 는 로봇의 최고 속도이고, ω_{max} 는 최고 각속도이다. 이와 같이 설정된 dynamic window 내의 (v_b, v_r)마다 다음과 같이 3개의 목적함수(objective function)의 합으로 표시되는 전체 목적함수(overall objective function)인 w^* 가 존재하게 된다.

$$w^* = \alpha_{head} \cdot w_{head} + \alpha_{speed} \cdot w_{speed} + \alpha_{clear} \cdot w_{clear} \quad (1)$$

여기서, $w_{head}, w_{speed}, w_{clear}$ 는 로봇의 방향, 속도 및 충돌에 관련된 목적함수이고, $\alpha_{head}, \alpha_{speed}, \alpha_{clear}$ 는 $w_{head}, w_{speed}, w_{clear}$ 의 가중치이다. 이때 전체 목적함수 w^* 가 최대가 되는 로봇 속도(v_b, v_r)를 선택하여 출력한다. 해당 함수의 가중치를 조절함에 따라 DWA의 로봇 운동 특성이 결정된다.

로봇의 방향과 관련되는 heading objective function은 단위 시간 동안 이동하는 로봇 위치를 그림 2(a)와 같이 예상하고, 예상된 위치에서 목표점까지의 각도 차이(θ_d)를 그림 2(b)의 선도에 의하여 w_{head} 를 구한다.

로봇 속도와 관련된 speed objective function은 그림 3과 같이 v_b 와 v_r 의 모든 영역에서 로봇의 이동속도(v)가 증가할수록 w_{speed} 의 값도 증가하며, v_{max} 에 이르러 w_{speed} 는 1이 된다.

Clearance objective function는 로봇과 장애물이 충돌하지 않을수록 w_{clear} 를 크게 한다. 로봇과 장애물의 충돌에 대한 계산은 로봇과 장애물 사이의 거리[s]를 이용한 방식과 충돌시

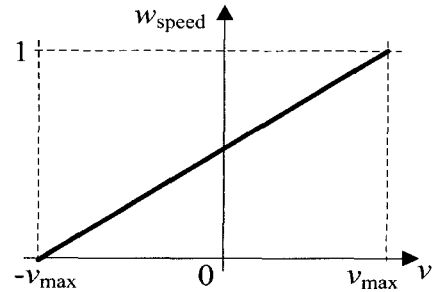


그림 3. Speed objective function (w_{speed})의 계산.
Fig. 3. Computation of speed objective function (w_{speed}).

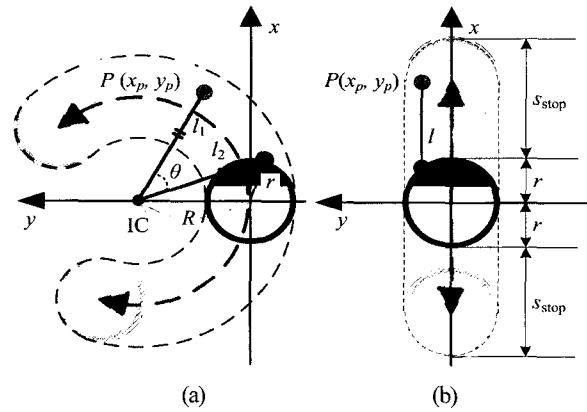


그림 4. 충돌예측.
Fig. 4. Collision prediction.

간[l]을 이용한 방식으로 구분된다. 로봇과 장애물 간의 거리를 이용한 방법은 로봇 회전 시와 고속주행 시에 발생하는 충돌회피에 취약하므로 충돌까지 소요되는 시간을 사용하여 충돌 계산을 수행하였다.

충돌예측은 그림 4와 같이 로봇이 회전운동을 하는 경우와 병진운동을 하는 경우로 나누며, 그림 4에서 빗금 친 영역과 같이 로봇이 이동하는 영역에 존재하는 장애물과의 충돌 시간을 계산하여 충돌을 예측하게 된다. 그림 4(a)와 같이 회전운동 하는 경우에는 장애물과 장애물이 로봇과 충돌할 때의 사잇각인 θ 를 ω 로 나누어 충돌시간(t_{col})을 구한다. 병진운동 때는 그림 4(b)에서 보듯이 장애물과 로봇의 거리인 l 을 v 로 나누어 t_{col} 을 계산한다. 이와 같이 dynamic window 내의 각 (v_b, v_r)마다 t_{col} 을 구하고, (2)와 같이 w_{clear} 를 정규화한다.

$$w_{clear}(v) = \begin{cases} 0 & (t_{col} \leq T(v)) \\ \frac{t_{col} - T(v)}{T_{max} - T(v)} & (T(v) < t_{col} < T_{max}) \\ 1 & (t_{col} \geq T_{max}) \end{cases} \quad (2)$$

여기서, $T(v)$ 는 해당 속도 (v_b, v_r)에서 정지할 때까지 소요되는 시간이며, T_{max} 는 v_{max} 에서 정지할 때까지 소요되는 시간이다.

이와 같이 DWA는 dynamic window 내의 각 (v_b, v_r)마다 각각의 목적함수의 결과를 구하고, (1)에 의하여 계산된 w^* 중에서 가장 큰 값을 가질 때의 (v_b, v_r)을 속도로 선택한다. 물론 이 속도는 현재 상황에서 목표점을 향하여 장애물을 회피하며 가장 빠르게 주행할 수 있는 속도에 해당한다.

III. 구배법

Konolige에 의하여 제안된 구배법은 장애물 위치, 바닥의 미끄러짐 정보와 같은 환경특성에 따라 결정되는 고유비용 (intrinsic cost)과 목표점까지의 주행거리에 따라 결정되는 근접비용(adjacency cost)을 합하여 주행비용(navigation cost)을 구하고, 이를 바탕으로 각 위치에서의 주행함수(navigation function)을 계산한다. 그리고 주행함수가 감소하는 방향을 따라 목표점까지의 경로를 구하게 되는데, 이 경로가 최소비용을 갖는 최적경로에 해당하게 된다.

근접비용과 고유비용을 계산한 뒤에, 현재위치에서 목표점으로 가는 경로를 결정하는 방법으로 주행함수의 개념을 사용한다. 경로 P 는 로봇 자체의 반경만큼 장애물을 확장한 지도내의 경로점의 수열이다.

$$P = \{p_1, p_2, \dots, p_N\} \quad (3)$$

여기서, p_i 는 경로상의 경로점(waypoint)를 의미하며, N 은 시작점(새로운 주행비용의 계산시마다 로봇의 현재위치가 새로운 경로의 시작점이 됨)과 목표점을 포함한 경로상의 모든 경로점의 개수를 나타낸다. 최종점인 p_N 은 목표점(goal)에 해당한다. 또한, 경로 P 의 비용함수 $F(P)$ 는 다음과 같다.

$$F(P) = \sum_{i=1}^N I(p_i) + \sum_{i=1}^N A(p_i, p_{i+1}) \quad (4)$$

여기서, $I(p_i)$ 는 점 p_i 에서의 고유비용이고, $A(p_i, p_{i+1})$ 는 점 p_i 에서 p_{i+1} 까지의 이동에 대한 근접비용이다. (4)는 경로 P 를 통하여 목표점에 도달하는 경우의 비용을 나타내는데, 시작점에서 목표점까지 주행하는 경로는 다수개가 존재할 수 있으므로 이들 경로 중에서 최적의 경로를 선택하는 것이 필요하다. 이를 위하여 다음과 같이 주행함수(navigation function)을 정의한다.

$$N_k = \min_{P_k} F(P_k) \quad (5)$$

여기서, P_k 는 점 k 에서 시작하는 경로를 나타낸다. N_k 는 점 k 에서의 주행함수로, 그 값은 점 k 에서 시작하는 다수의 경로 P_k 중에서 최소비용을 갖는 경로의 비용으로 부여된다. 즉, 로봇이 어떤 점에 있을 때 이 점에서의 주행함수는 이 점에서 목표점까지 도달할 수 있는 최소비용을 의미한다.

앞서 정의한 근접비용은 두 점 사이의 이동에 관련된 경로 비용으로, 대부분의 경우 이동거리에 비례하여 경로비용을 부여한다. 즉, 목표점을 한 점으로 설정하면, 목표점으로부터의 거리에 비례하여 증가하는 비용이 각 셀에 부여된다. 이때, 각 셀에 부여할 비용을 계산하기 위하여, 목표점으로부터 주변의 셀로 확장해 나가면서 거리를 산출하여야 하는 체계적인 방법이 필요한데, 이를 위하여 본 연구에서는 Konolige에 의해서 제안된 LPN 알고리즘을 사용하였다. LPN 알고리즘은 셀과 셀 사이의 거리를 유클리드 거리로 계산하며, 상하좌우 및 대각선의 8방향으로 확장하여 근접비용을 갱신해 나가는 방법이다.

주어진 환경지도 상의 각 셀에 대하여 계산한 주행비용에 기초하여 시작점에서 목표점까지의 경로를 생성하게 되는데,

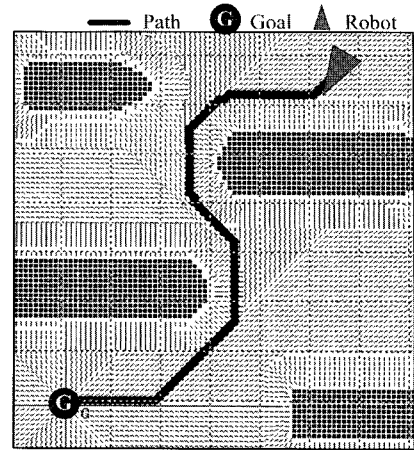


그림 5. 구배법으로 구한 최적경로.

Fig. 5. Optimal path found using gradient method.

그 방법으로 주행비용이 가장 빠르게 감소하는 방향으로 모든 셀에 대하여 기울기(gradient)를 생성한다. 그림 5에서 장애물에서는 로봇을 밀어내는 방향으로, 목표점에서는 끌어당기는 방향으로 기울기가 생성된 것을 볼 수 있다. 이와 같이 지도전역에 대하여 생성된 기울기에 대하여, 목표점에서 로봇위치까지 급격히 감소하는 기울기를 검색하며 경로를 생성하여 로봇을 주행시키면, 지도상에 나타나는 장애물을 회피하면서 목표점까지 최소의 비용으로 도달하는 최적경로가 생성된다.

IV. 전역 DWA

전역 DWA(global)는 3장에서 설명한 구배법을 이용하여 최종 목표점에 도달하기 위한 중간 목표인 경유점을 추출하고 경유점 간에는 DWA로 주행함으로써, 극부 최소를 회피하면서 목표점까지 안정하게 도달하는 주행 방법이다. DWA는 w_{head} 와 w_{speed} 에 의하여 로봇이 목표점을 향하여 빠르게 이동하게 하기 때문에 경유점 사이에 장애물이 존재하지 않는 직선으로 연결된다면, 로봇은 DWA의 특성에 잘 부합하며 주행할 수 있다. 이에 구배법의 결과로 얻는 최적경로에서 이러한 장애물에 걸리지 않는 직선을 찾아 직선의 시작점과 끝점을 경유점으로 설정하도록 하였다. 이에 대하여 아래에 자세히 설명한다.

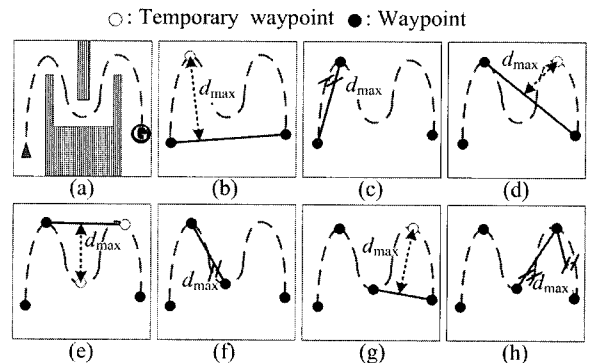


그림 6. 직선 분할 기법을 이용한 경유점 추출.

Fig. 6. Waypoint extraction using line segmentation algorithm.

그림 6은 경유점을 추출하는 방법으로, [10]에서 제시된 레이저 스캐너의 데이터로부터 직선성분을 구하기 위하여 사용된 방법을 활용하였다. 그림 6(a)와 같은 환경에서 구배법에 의하여 점선과 같은 최적경로가 생성되었다고 가정하자. 경로의 시작점과 종점을 연결하여 직선을 만든다. 이 직선과 모든 경로 사이의 최대 거리(d_{max})를 갖는 점을 찾고 d_{max} 가 임계치보다 클 경우에는 이 점을 임시 경유점으로 설정한다 (b). 시작점과 임시 경유점을 양 끝점으로 하는 구간에서 위의 과정을 반복하여 d_{max} 를 새로이 찾고, 이 d_{max} 가 임계치보다 작다면 임시 경유점은 경유점으로 등록된다(c). 경유점을 설정한 후에는 경유점을 시작점으로 하고 경로의 마지막 점을 끝점으로 하는 구간 (d)에서 새로이 찾은 d_{max} 가 임계치보다 작을 때까지 (b)에서 (c)로의 과정을 반복하여 (h)와 같이 경유점을 구한다. 여기서 최적경로는 c-obstacle[7]에 의하여 최소한 로봇 반경만큼 장애물로부터 떨어져서 생성되어 있으므로, 경로상의 경유점 역시 최소한 로봇 반경만큼 장애물로부터 떨어져 존재하게 된다. 그러므로 임계치를 로봇 반경으로 정하면 경유점 사이에는 장애물이 존재하지 않는 직선으로 연결되어, 로봇은 경유점을 이용하여 목표점으로 빠르게 이동할 수 있다.

그림 7은 로봇이 복도를 지나가다가 목표점이 있는 방으로 진입하는 경우를 나타낸다. 이 경우에 로봇은 DWA의 speed objective function에 의하여 최대한 빠른 속도로 주행하려고 하므로 충분히 회전하지 못한다. 더불어 다음 경유점에서의 전환시점에 대한 적절한 방안이 없기 때문에 회전을 위한 속도 (v_b, v_r)의 선택이 늦어져 그림 7(d)에서와 같이 경유점을 지나쳤다가 다시 되돌아 오게 된다. 이를 해결하기 위해서는 방향을 전환하여야 하는 상황에서 로봇이 충분히 감속

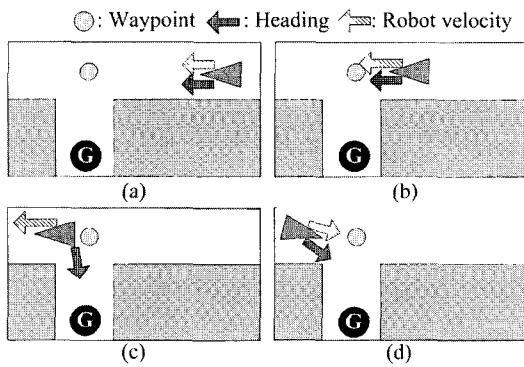


그림 7. 전역 DWA의 문제점.
Fig. 7. Problem of global DWA.

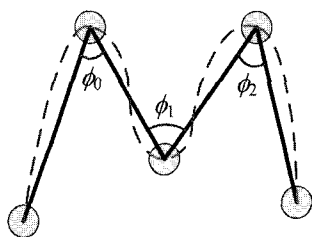


그림 8. 직선 간의 사이각(ϕ).
Fig. 8. Included angles between lines(ϕ).

할 수 있어야 하며, 다음 경유점에 대한 정보를 제공해야 하는 시점에 대한 정확한 근거가 마련되어야 한다.

이에 본 논문에서는 구배법의 최적경로를 이용하여 국부 최소에 빠지지 않는 경유점을 제공하고 경유점들을 연결하는 직선의 각도를 이용하여 speed objective function의 정규화식을 새로이 제안한다. 이로써 로봇은 경유점에서 회전할 수 있도록 충분히 감속시킬 수 있는 근거를 마련하였다. 또한, 경유점 전환 시점은 동적 윈도우 내의 (v_b, v_r)에 의하여 경유점에 도착할 수 있는 시간을 예측하고, 최단 도착시간이 최대 정지시간(T_{max})보다 짧을 때로 규정하였다.

그림 8에서의 ϕ 는 경유점을 연결하는 직선 간의 각도이다. 이 각도를 speed objective function의 입력으로 사용하기 위하여, 그림 3에서 w_{speed} 를 나타내는 식

$$w_{speed} = \frac{v + v_{max}}{2 \cdot v_{max}} \tag{6}$$

을 다음과 같이 변형하였다.

$$w_{speed} = 1 - \left| \frac{v - (v_{max} - w_{intercept})}{(2 - w_{grad}) \cdot v_{max}} \right| \tag{7}$$

여기서, w_{grad} 는 w_{speed} 식의 기울기를 조절하고, $w_{intercept}$ 는 y 절편에 영향을 준다. 이 두 변수는 (8)과 같이 ϕ 에 의하여 결정된다.

$$w_{grad} = \frac{(\pi - \phi)}{\pi}, \quad w_{intercept} = \frac{(\pi - \phi)}{\pi} \cdot v_{max} \tag{8}$$

그림 9는 ϕ 가 $\pi, 0$ 인 경우의 speed objective function을 나타내는 선도이다. 그림 9(a)에서 보듯이 ϕ 가 π 인 경우는 직선 경로의 주행에 해당하므로, 기존의 speed objective function과 동일하게 속도가 증가할수록 w_{speed} 가 크게 설정된다. 그림 9(b)와 같이 ϕ 가 0인 경우에는 로봇이 제자리에서 180° 회전하여야 다음 경유점으로 향할 수 있으므로 w_{speed} 는 속도가 0에 가까울수록 큰 값을 갖는다.

그림 10과 같은 환경에서 최적경로로부터 추출된 경유점의 사이 각도는 114°, 102°이다. 이 각도로부터 (7)과 (8)에 의하여 그림 11과 같이 speed objective function이 변경된다. 기존의 speed objective를 나타내는 그림 3의 선도는 그림 11과 같이 ϕ 에 의하여 최고속도의 63% 또는 56%일 때 w_{speed} 가 1이 되는 선도로 변한다. 이처럼 기존의 speed objective에서는 속도가 클수록 w_{speed} 가 커지는 것과 달리 ϕ 에 의하여 적절한

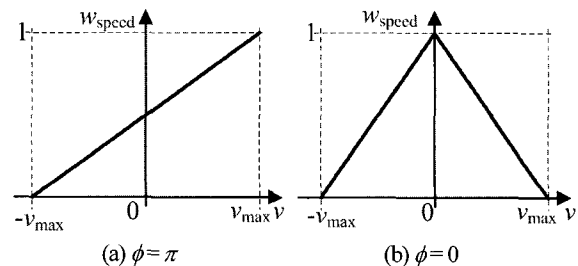


그림 9. ϕ 에 의하여 변경된 speed objective function.
Fig. 9. Speed objective function modified by ϕ .

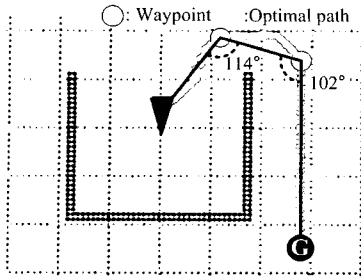


그림 10. 경유점을 이용한 전역 DWA.
Fig. 10. Global DWA using waypoints.

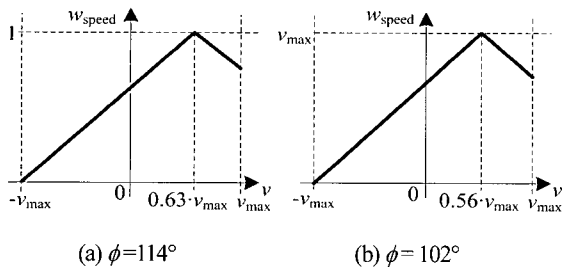


그림 11. ϕ 가 114°와 102°일 때 speed objective function.
Fig. 11. Speed objective function for $\phi = 114^\circ$ and $\phi = 102^\circ$.

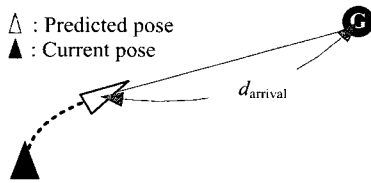


그림 12. $d_{arrival}$ 의 예측.
Fig. 12. Prediction of $d_{arrival}$.

속도를 선택할 수 있다. 그러나 시작점에서 목표점까지 모든 주행구간에서 변경된 speed objective function에 ϕ 를 적용한다면 로봇은 언제나 최고속도로 주행할 수 없게 된다. 그러므로 감속이 필요치 않은 일반적인 경우에는 (8)의 ϕ 를 π 로 하여 그림 9(a)와 같이 빠른 속도일수록 w_{speed} 가 큰 값을 갖도록 한다. 더불어 그림 10과 같은 ϕ 를 (7)에 적용하여 그림 11과 같은 선도로 속도를 조절하고, 다음 경유점으로 목표를 전환하는 시점은 다음과 같이 정의한다.

그림 12와 같이 DWA의 목적함수인 heading objective function을 수행할 때 예상되는 로봇 위치에서 목표점까지의 거리($d_{arrival}$)를 구한다. $d_{arrival}$ 를 v 로 나누면 목표점에 도착할 때까지 시간($T_{arrival}$)을 알 수 있다. $T_{arrival}$ 가 최대 정지시간(T_{max})과 비교하여 같거나 짧아질 때, 다음 경유점으로 전환하고 경유점 사이 각도를 (8)에 적용한다. 이는 로봇이 최고속도로 주행하더라도 $T_{arrival}$ 이 T_{max} 와 같다면 경유점에서 정지할 수 있기 때문에 경유점을 지나치기 전에 충분한 감속을 하여 안정하게 다음 경유점으로 주행할 수 있다.

이와 같이 경유점 간의 사잇각을 이용한 speed objective function의 변환과 다음 경유점에 대한 정보를 제공하는 시점의 명확한 근거를 통하여 로봇은 목표점까지 안정적인 고속 주행을 할 수 있다.

V. 실험 및 결과

그림 13(a)와 같이 복도 환경이 포함된 실험환경(10m×10m)을 대상으로 주행 실험을 수행하였다. 주어진 격자지도의 한 격자는 10cm×10cm이므로, 이 환경은 100×100개의 점유격자로 구성되어 있다. 격자지도를 바탕으로 구배법에 의하여 계획된 최적경로에서 경유점을 추출하고, 경유점을 차례로 수행하는 전역 DWA를 실험하였다.

그림 14는 로봇이 매 200ms마다 움직인 경로를 연속적으로 표현한 것으로, (a)는 기존의 전역 DWA, (b)는 본 과제에서 제안한 전역 DWA의 결과를 나타내었다. 기존의 방식에서는 로봇의 빠른 속도로 이동하다 좁은 영역으로 진입하는 경우에 경유점에서 로봇의 속도를 미리 감속하는 수단이 없기 때문에 경유점을 지나치는 문제점이 발생하였다. 결국 로봇은 최적경로를 벗어나게 되며, 최악의 경우에는 극부 최소에 빠져 목표점까지 도달하지 못하게 될 수도 있다. 그러나 본 논문에서 제안한 방식에서는 각 경유점 사이의 각도 정보를 이용하여 w_{speed} 를 조절하므로, 그림 14(b)와 같이 경유점을 도달하기 전에 충분히 감속하여 경로를 잘 추종하게 되며, 병진 이동 구간에서는 빠른 속도로 주행할 수 있다.

그림 15와 같이 복도에서 최고속도로 주행하다 좁은 입구로 들어갈 때 기존의 전역 DWA를 사용하면 위의 문제가 더욱 심각해져 로봇은 목표점에 도착하기 어려울 수도 있다. 그러므로 그림 15(a)에서와 같이 최고속도를 인위적으로 줄이는 방법(여기서는 400mm/s)을 사용하여야 했다. 그러나 본

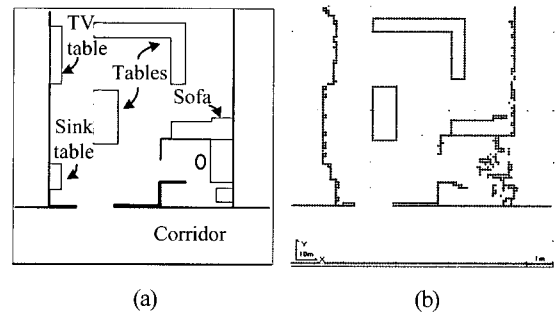


그림 13. 실험 환경.
Fig. 13. Experimental environment.

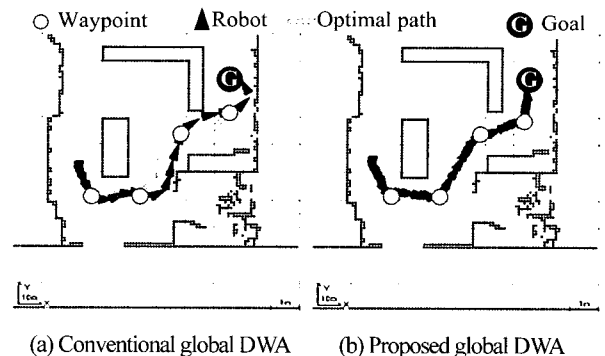


그림 14. 기존의 전역 DWA와 제안한 전역 DWA에서의 로봇 이동경로의 비교(최고속도 = 900mm/s).
Fig. 14. Comparison of the proposed global DWA with the conventional one when $v_{max} = 900\text{mm/s}$.

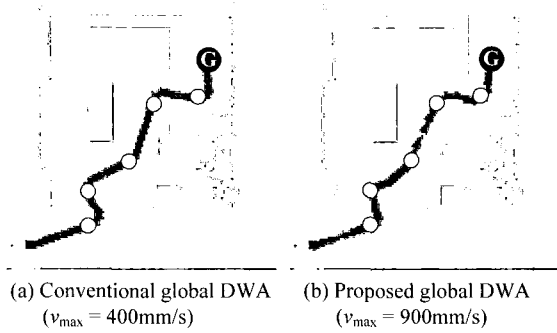


그림 15. 복도환경에서의 기존의 전역 DW와 제안한 전역 DWA의 주행.

Fig. 15. Comparison of the proposed global DWA with the standard one in corridor.

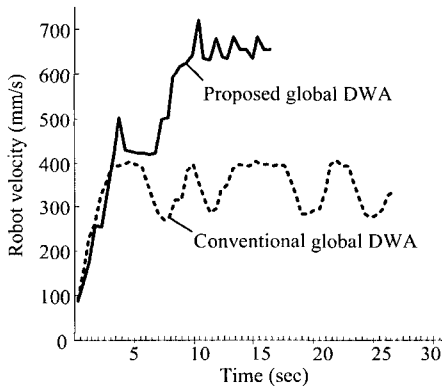


그림 16. 기존의 전역 DWA와 제안한 전역 DWA의 속도 비교.
Fig. 16. Velocity comparison of the proposed global DWA with standard one.

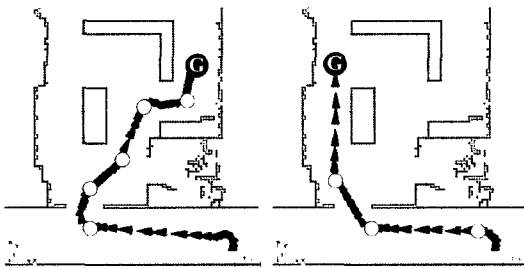


그림 17. 제안한 전역 DWA의 주행 결과.
Fig. 17. Result of proposed global DWA.

논문에서 제안한 전역 DWA는 최고속도가 900mm/s인 설정을 변경하지 않고 주행하였음에도 불구하고, 그림 15(b)처럼 기존의 전역 DWA에 비하여 최적경로와 비슷한 주행을 빠르게 한 것을 확인할 수 있다.

그림 16은 동일한 환경에서 기존의 전역 DWA와 제안한 전역 DWA의 주행에서의 출력 속도를 비교한 것이다. 그림 16에서 보듯이 기존의 방법에서는 v_{max} 를 400mm/s로 제한하여야만 목표점까지 도달할 수 있었기에 약 27초의 주행 시간이 걸렸으나, 제안한 방법에서는 16초에 목표점까지 도착하

였다. 이와 같이 제안된 방법을 사용하면 로봇은 직선 구간에서 고속주행이 가능하며 회전이 필요한 영역에서는 충분히 감속할 수 있다.

그림 17은 그림 15와 같이 복도가 포함된 환경에서 좁은 영역을 통과하여 목표점으로 주행한 모습이다. 로봇은 복도에서 최고속도로 주행하다 방안으로 들어오기 전 경유점에서 점차 감속하여 최적 경유점을 통과하면서 DWA로 생성된 경로를 따라 자연스러운 주행을 하고 있음을 관찰할 수 있다.

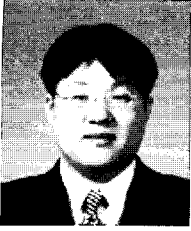
위의 실험 결과에서 보듯이 본 논문에서 제안한 방법은 각 경유점을 연결한 각도를 이용하여 w_{speed} 를 조절함으로써, 매 경유점에서 신속한 감속이 가능해져 기존의 전역 DWA보다 빠르면서도 안정된 주행으로 목표점에 도달할 수 있다.

VI. 결론

본 논문에서는 구배법으로 작성한 최적경로에서 경유점을 추출하고, 이들 경유점 사이의 각도 정보로부터 경유점 근처에서의 로봇의 속도를 사전에 조절함으로써, 신속하고도 자연스러운 주행이 가능한 전역 DWA 기법을 제시하였다. 로봇이 각 경유점에 도착하게 될 시간을 계산하여 최대 정지시간보다 짧을 때, 경유점 사이의 각도에 따라 DWA의 speed objective의 변환식을 이용하여 경유점에서 감속함으로써 로봇은 경유점 근처에서도 경로를 크게 벗어나지 않고 목표점에 안정적으로 도달할 수 있다.

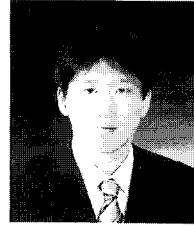
참고문헌

- [1] R. Siegwart, *Introduction to Autonomous Mobile Robots*, A Bradford Book, pp. 258-290, 2004.
- [2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.
- [3] J. Bornstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278-288, 1991.
- [4] R. Simmons, "The curvature-velocity method for local obstacle Avoidance," *Proc. of International Conf. on Robotics and Automation*, 1996.
- [5] D. Fox, W. Burgard, and S. Thrun. "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, pp. 23-33, 1997.
- [6] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *Proc. of IEEE International Conf. on Robotics and Automation*, 1999.
- [7] J.-C. Latombe, *Robot motion planning*, 4th Edition, Kluwer academic publishers, 1996.
- [8] K. Konolige, "A gradient method for real-time robot control," *Proc. of International Conf. on Intelligent Robots and Systems*, pp. 639-646, 2000.
- [9] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," *Proc. of International Conf. on Robotics and Automation*, pp. 13-19, 2003.
- [10] L. Zhang, "Line segment based map building and localization using 2D laser rangefinder," *Proc. of International Conf. on Robotics and Automation*, vol. 3, pp. 2538, 2000.



함 종 규

1977년 9월 10일생. 2004년 한국외국어대학교 디지털정보공학과(공학사). 2006년 고려대학교 메카트로닉스학과(공학석사). 현재 삼성전자 근무중. 관심분야는 주행알고리즘 개발.



박 중 태

1979년 11월 14일생. 2005년 고려대학교 전산학과(이학사). 2007년 고려대학교 메카트로닉스학과(공학석사). 현재 고려대학교 메카트로닉스학과 박사과정 재학중. 관심분야는 이동로봇의 주행 및 물체인식알고리즘 개발.



송 재 복

1960년 8월 5일생. 1983년 서울대학교 기계공학과(공학사). 1985년 서울대학교 기계설계학과(공학석사). 1992년 MIT 기계공학과(공학박사). 1993년~현재 고려대학교 기계공학과 교수. 관심분야는 로봇 팔의 설계 및 제어, 이동로봇의

주행, 햅틱스.