

Path Planning for Autonomous Mobile Robots by Modified Global DWA

윤희상* · 박태형†

(Hee-Sang Yoon · Tae-Hyoung Park)

Abstract – The global dynamic window approach (DWA) is widely used to generate the shortest path of mobile robots considering obstacles and kinematic constraints. However, the dynamic constraints of robots should be considered to generate the minimum-time path. We propose a modified global DWA considering the dynamic constraints of robots. The reference path is generated using A* algorithm and smoothed by cardinal spline function. The trajectory is then generated to follow the reference path in the minimum time considering the robot dynamics. Finally, the local path is generated using the dynamic window which includes additional terms of speed and orientation. Simulation and experimental results are presented to verify the performance of the proposed method.

Key Words : Autonomous mobile robots, Path planning, Dynamic window approach (DWA), Robot dynamics.

1. 서론

자율이동로봇은 각종 센서를 이용하여 위치 및 장애물과 같은 환경을 인식하고, 주어진 지도에서 현재 위치에서 목표점까지 이동 가능한 경로를 생성하고, 생성된 경로를 따라 주행하는 로봇이다. 자율이동로봇은 환경인식을 통해 목적지까지 안전하고 빠르게 주행하는 것이 핵심이며, 주행성능을 높이기 위한 연구들이 진행되어왔다.

자율이동로봇은 경로계획을 통해 이동경로를 생성하고, 이동경로를 따라 주행하게 된다. 경로계획은 지도가 주어진 상태에서 전체경로를 생성하는 전역경로계획과 센서값들로부터 환경을 인지하며 목표점에 도달하는 지역경로계획이 있다. 전역경로계획은 가시도그래프(V-graph), 보르노이 다이어그램(voronoi-diagram), 구배법, A*알고리즘 등 다양한 방법들이 연구되었으나 경로이탈 또는 새로운 장애물이 발견시 다시 경로를 생성해야하므로 실시간으로 적용하기엔 문제점이 있다[1]. 반면에 DWA(dynamic window approach)[2][3], EB(elastic band)[4], VFH(vector field histogram)[5] 등과 같은 지역경로계획 방법들의 경우 센서기반으로 실시간으로 주변 장애물을 인식하여 목표점까지 이동하는 방법이다. 지역경로계획 방법은 새로운 장애물 발견시 실시간 처리가 가능하고, 로봇의 동역학 특성을 고려가 가능한 장점이 있지만, 지역적인 제한으로 인해 지역최소문제에 빠져 목표점에 도달하지 못하는 경우가 발생할 수 있다. 최근에는 전역경

로계획과 지역경로계획의 장점을 결합하여 전역경로계획을 통해 초기경로를 생성하고, 이 경로를 지역경로계획을 통해 주행하는 연구들이 있다[6][7]. 전역 DWA의 경우 전역경로 생성 방법인 NF1, A*, D* 등을 이용하여 초기경로를 생성하고[7][8], 중간 경유점들을 DWA의 목표점으로 설정하여, 지역최소문제를 해결하고 실시간으로 장애물회피가 가능하도록 하였다[9]. 그러나 전역 DWA의 경우 경유점 전환시 로봇의 동역학 특성을 고려하지 않아 경로이탈 문제가 발생하여 로봇이 주행이 불안정해 지는 문제가 있다.

본 논문은 기존의 전역 DWA의 경로이탈 문제를 해결하고, 로봇이 안정적인 주행이 가능한 수정된 전역 DWA 방법을 제안한다. 수정된 전역 DWA는 전역경로계획에서 기존방법과 유사하게 A*알고리즘을 적용하여 초기경로를 탐색하고, 초기경로에 대해 곡선생성 방법인 카디널스플라인(cardinal spline)[10][11]으로 곡선경로를 생성한다. 또한 생성된 곡선경로에 대해 동역학 특성이 고려된 속도프로파일[12][13]을 생성한다. 지역경로계획에서는 전역경로계획에서 생성한 곡선경로와 속도프로파일을 기반으로, 가상경로를 생성하고, 수정된 DWA목적함수를 적용하여 경로이탈이 없이 안정적으로 곡선경로를 주행한다. 시뮬레이션과 실험을 통

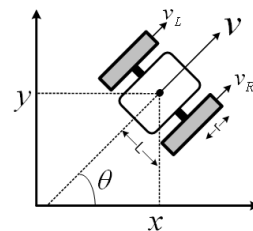


그림 1 로봇 모델.

Fig. 1 The robot model.

* 정희원 : 충북대 제어로봇공학과 박사과정

† 교신저자, 정희원 : 충북대 전자공학부 및 컴퓨터정보통신 연구소 교수

E-mail : taehpark@cbnu.ac.kr

접수일자 : 2010년 12월 14일

최종완료 : 2011년 1월 19일

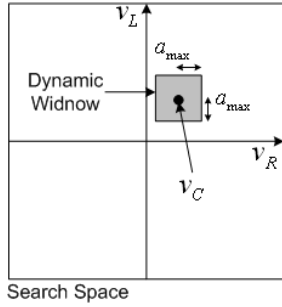


그림 2 탐색공간.
Fig. 2 The search space.

해 경로생성 및 주행 성능을 평가한다.

2. DWA (Dynamic Window Approach)

먼저, 본 논문에서는 그림 1과 같이 두 개의 독립적인 구동가능한 바퀴를 가지는 차동구동방식의 자율이동로봇을 대상으로 한다. 각각의 바퀴로 구동되는 로봇은 최대속도 및 최대가속도 범위 내에서 제어되어야한다. DWA는 로봇이 이동 가능한 최대속도 범위를 그림 2와 같이 탐색공간 (search space)으로 정의하고, 로봇의 현재속도 v_c 를 기준으로 최대가속도 범위내에 있는 영역을 동적윈도우로 설정한다. 또한 동적윈도우내의 좌우바퀴의 속도에 대해 목적함수 (objective function)를 계산하여 최대가 되는 속도를 찾아 로봇을 제어하는 방법이다[2]. DWA는 로봇의 기구학 및 동역학 제약조건을 고려할 수 있으며, 장애물과의 충돌예측을 통해 장애물 회피도 가능하다. 동적윈도우는 격자로 구성되며, 이는 실시간 계산에 유리하다.

DWA는 동적윈도우내에서의 각 바퀴의 속도 정보로 예상 가능한 지점 p_n 지점을 찾고, 목표점 p_t 사이에 방향, 속도 및 충돌 관련된 함수들로 목적함수 O 를 구성한다. 다음은 DWA의 목적함수를 나타낸다.

$$O = \gamma_{head} w_{head} + \gamma_{speed} w_{speed} + \gamma_{clear} w_{clear} \quad (1)$$

여기서 $w_{head}, w_{speed}, w_{clear}$ 는 각각 로봇의 방향, 속도, 장애물과의 충돌에 관련된 함수이며, $\gamma_{head}, \gamma_{speed}, \gamma_{clear}$ 는 각 함수의 가중치를 나타낸다.

DWA의 목적함수 중 방향함수 w_{head} 는 목표지점과의 방향을 수치화한 함수로, 그림 3과 같이 예상지점 p_n 과 목표지점 p_t 사이의 각 θ_{tn} 를 계산한다. θ_{tn} 이 0인 경우는 목표점과 방향이 일치하므로 최대값인 1을 설정하고, $-\pi, \pi$ 인 경우는 목표점 방향과 반대 방향으로 최소값인 0을 다음과 같이 설정한다.

$$w_{head} = 1 - \frac{|\theta_{tn}|}{\pi} \quad (2)$$

로봇이 동일한 경로를 이동시에는 속도가 빠를수록 목표점까지 도달시간이 빨라진다. DWA에서는 속도가 빠를수록

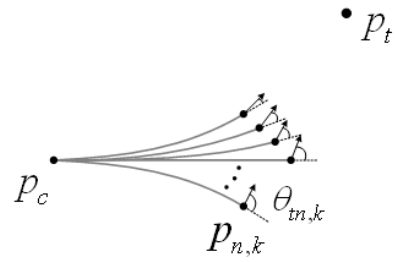


그림 3 DWA 방향함수.
Fig. 3 The Direction function of DWA.

1에 가까운 값이 되도록 속도함수 w_{speed} 를 설정한다. 동적윈도우의 선속도 v_n 이 최대속도 v_{max} 일 경우에 1, $-v_{max}$ 일 경우에 0이 된다. 속도함수 w_{speed} 는 다음과 같다.

$$w_{speed} = \frac{(v + v_{max})}{2v_{max}} \quad (3)$$

DWA는 장애물과의 거리나 충돌예상시간을 이용하여 충돌함수를 설정할 수 있다[5]. 장애물과의 거리를 이용한 방법의 경우, 센서정보로부터 장애물과의 거리정보로부터 단순히 사용할 수 있으나, 로봇이 회전 시 장애물과의 거리 표현이 정확하지 않으며, 장애물과의 충돌거리가 로봇의 속도에 따라서 달라진다. 이런 문제를 보완하여 최근에는 장애물과의 거리 대신, 장애물과의 충돌예상시간을 함수로 사용한다.

충돌시간으로 설정한 충돌함수 w_{clear} 는 로봇이 직선 및 회전 시 장애물까지의 충돌시간 t_{col} , 현재속도에서 최대가속도로 정지 할 때의 소요시간 $T(v)$, 최대속도 v_{max} 에서 정지시까지의 최소시간을 T_{max} 등을 설정하고, 직선운동과 회전운동으로 분리하여 계산한다. 직선운동인 경우 장애물까지의 거리 ρ 와 현재속도 v 로 나누어 구하고, 회전운동인 경우에는 장애물까지의 회전각 $\theta_{obstacle}$ 와 로봇의 회전속도 ω 를 이용하여 충돌시간 t_{col} 을 계산한다.

$$t_{col} = \begin{cases} \rho/v & : w = 0 \\ \theta_{obstacle}/\omega & : w \neq 0 \end{cases} \quad (4)$$

또한 현재 속도에서 최대가속도로 정지 까지 소요시간 $T(v)$, 정지시까지 a_{max} 로 감속할 경우 시간 T_{max} 는 각각 다음과 같다.

$$T(v) = \frac{\max(v_R, v_L)}{a_{max}} \quad (5)$$

$$T_{max} = \frac{v_{max}}{a_{max}} \quad (6)$$

충돌함수 w_{clear} 는 충돌시간 t_{col} 을 기준으로 $T(v)$ 보다 작은 경우는 장애물과 충돌이며, T_{max} 보다 큰 경우는 장애물이 없는 경우, 그 외의 경우에 대해 다음과 같이 충돌함수의

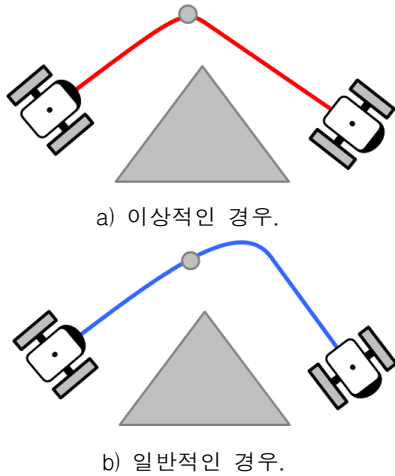


그림 4 전역DWA의 지역목표점 전환.
Fig. 4 Switching of Local target points.

값을 결정한다.

$$w_{dear} = \begin{cases} 0 & : t_{col} \leq T(v) \\ \frac{t_{col} - T(v)}{T_{max} - T(v)} & : T(v) < t_{col} < T_{max} \\ 1 & : t_{col} \geq T_{max} \end{cases} \quad (7)$$

DWA는 동적윈도우에서 방향, 속도, 충돌함수를 (2), (3), (7)을 이용하여 각각 구하고, (1)을 이용하여 목적함수 O 를 구하여, 이 목적함수가 최대가 되는 값을 다음상태의 좌우바퀴의 속도로 선택하는 과정을 목표지점에 도달하기까지 반복한다. DWA는 이와 같이 전역경로 없이, 센서만을 이용하여 현재지점에서 목표지점까지 이동한다. 따라서 기존의 센서기반 경로계획과 마찬가지로 국부최소에 빠질 수 있다.

이러한 국부최소문제를 해결하기 위해 전역경로계획으로 전역경로를 생성하여, 중간 경유점들에 대해 DWA를 적용한 전역DWA에 대한 연구가 있다. 전역DWA는 전역경로 생성을 위해 NF1, A*, D*등과 같은 방법을 이용하며, 생성된 경로에서 특징점이 되는 점을 추출하여 이를 지역목표점으로 설정하고, 이 지역목표점들에 대해 DWA를 적용하는 방법이다[7][14]. 따라서 DWA의 장점을 그대로 유지하면서 단점인 국부최소문제를 해결하였다.

전역DWA는 이상적인 경우 그림 4(a)와 같이 지역목표점에 도달하여 새로운 지역목표점으로 전환된 후 이동한다. 그러나 지역목표점 전환 시 로봇이 지역목표점에 진입하는 속도가 빠른 경우 지역목표점 전환 후 그림 4(b)와 같이 빠른 속도에 의해 경로를 이탈하는 문제가 발생할 수 있다. 이로 인해 경로이탈에 따른 장애물 충돌 가능성이 발생할 수 있으며, 길어진 경로에 따라 로봇의 이동시간 또한 늘어날 수 있다. 이 문제점들을 해결하기 위해 로봇의 최대속도를 제한하여 경로이탈을 줄이거나, 지역목표점 전환을 빠르게 한 방법이 연구되었으나, 최대속도를 줄이는 경우 이동시간이 늘어나게 되고, 지역목표점 전환을 빠르게 할 경우에는 이동시간은 빠르지만 장애물과의 충돌 및 경로이탈 문제는 여전히 존재한다. 본 논문에서는 이러한 속도에 의한 경로

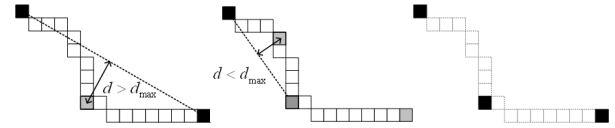


그림 5 최적경유점 추출과정.
Fig. 5 The process of selecting optimal way points.

이탈 문제를 줄여 전체 이동속도를 줄이는 수정된 전역DWA를 제안한다.

3. 수정된 전역 DWA

수정된 전역DWA는 전역경로계획으로 전역경로를 생성하고, 생성된 전역경로에 대해 수정된 DWA와 목적함수들을 이용하여 지역경로계획으로 경로를 따라 주행한다. 전역경로계획에서는 기존의 전역DWA와 같은 단순 전역경로생성과 함께 로봇의 방위각과 동력학 특성을 고려한 부드러운 경로 및 속도프로파일을 생성한다. 지역경로계획에서는 전역경로계획에서 생성된 곡선경로를 경로이탈 없이 빠르게 주행하기 위해, 기존의 DWA를 수정한 가상경로 기반의 목적함수를 적용한다.

3.1 전역경로계획

본 논문에서는 전역경로계획을 위해 격자(grid)기반으로 지도(map)를 구성하고, 경로탐색알고리즘인 A*알고리즘을 적용하여 초기경로를 생성한다. A*알고리즘으로 생성된 경로는 직선으로 연결된 경로로, 연결지점에서 로봇이 연속적으로 이동이 불가능하다. 기존 전역DWA의 경우 이러한 연결지점을 지역목표점으로 설정하였으나, 본 논문에서는 연결지점을 부드럽게 지나는 곡선경로를 생성한다. 곡선경로는 로봇의 동력학 제약조건을 고려하고, 방위각 고려가 가능한 곡선경로생성방법인 카디널스플라인을 적용하여 생성한다.

3.1.1 초기경로생성

A*알고리즘은 시작점에서 목표점까지 예측치를 이용하여 지도의 전체 영역을 탐색하는 전역탐색에 비해 빠른 경로탐색이 가능하다[1]. A* 알고리즘의 평가함수 $f(n)$ 은 시작점에서 현재지점까지의 비용을 나타내는 비용함수 $g(n)$ 과 현재 지점에서 목표점까지의 예측치를 나타내는 휴리스틱함수 $h(n)$ 으로 나타낸다.

$$f(n) = g(n) + h(n) \quad (8)$$

A* 알고리즘은 시작점으로부터 평가함수 $f(n)$ 이 작은 지점을 우선 탐색하여 목표점에 도달한 경우 역탐색을 통해 시작점에서 목표점까지 경로를 생성한다. A*알고리즘은 장애물을 회피하는 경로를 빠르게 탐색하지만 A*알고리즘에서 생성된 경로는 곡선경로 생성 시, 불필요한 경유점들을 포함하고 있다. 따라서 불필요한 경유점들을 제거하여 곡선경로 생성 시 필요한 최적경유점을 추출한다. 최적경유점

추출 방법으로는 다각형 근사화에 사용된 방법을 사용한다 [15].

다각형 근사화 방법은 그림 5와 같이 가장 멀리 떨어져 있는 두 점을 찾아서 직선으로 연결하고, 나머지 경유점들 중 가장 먼 점을 찾아 이 값이 d_{max} 보다 큰 경우에는 근사화 점으로 추가하여 반복 수행하고, 작은 경우에는 다음구간으로 이동하여 최종점이 될 때까지 반복적으로 적용한다. 다각형 근사화 방법을 이용해 추출한 초기경로의 경유점은 P 라 정의하고 다음과 같이 나타낸다.

$$P = \{p_1, p_2, \dots, p_g\} \quad (9)$$

3.1.2 곡선경로생성

초기경로에서 추출된 경유점들에 대해 곡선경로를 생성한다. 주어진 점들을 부드러운 곡선으로 연결하는 곡선생성 방법으로는 보간법과 근사법이 있다. 보간법은 점들사이를 지나가는 곡선을 생성하며, 근사법은 점은 곡선생성을 위한 제어점으로 사용하여 점 근처를 근사하는 곡선을 생성한다. 본 논문에서는 주어진 경유점에 대해 곡선경로를 생성하기 위해 에르미트보간(Hermite interpolation)을 기초로한 카디널스플라인(cardinal spline)을 적용한다[10].

에르미트보간은 두 점과 두 점에서의 접선벡터가 주어진 경우 접선벡터에 영향을 받아 곡선을 생성 한다. 에르미트 보간을 이용하여 2차원 공간상에서 곡선을 표현하기 위해 파라미터 u ($0 < u \leq 1$)를 정의하고, 다음과 같은 행렬로 표현 한다.

$$p(u) = [u^3 \ u^2 \ u \ 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(0) \\ p(1) \\ p'(0) \\ p'(1) \end{bmatrix} \quad (10)$$

에르미트 보간은 두 점 $p(0), p(1)$ 과 두 점에서의 접선벡터가 $p'(0), p'(1)$ 로 주어진 경우 (10)을 이용하여 그림 6(a)와 같은 곡선을 생성한다. 카디널스플라인은 두 점사이의 곡선을 생성하는 에르미트 보간을 기초로 다수의 점에 에르미트 보간을 적용한 것으로, 그림 6(b)와 같이 곡선과 곡선이 만나는 점에서는 접선벡터를 동일하게 설정하여, 부드럽게 연결한다. 시작점과 목표점에서는 로봇의 방향각을 이용하여 접선벡터를 설정하고, 중간 경유점에서는 주변 두 점 사이의 벡터를 이용하여 다음과 같이 접선벡터 p'_k 를 설정한다.

$$p'_k = (1-\alpha)(p_{k+1}-p_{k-1}) \quad \text{단 } , 0 \leq \alpha \leq 1 \quad (11)$$

여기서 α 는 접선벡터의 가중치로 이 값에 따라 곡선의 장력이 결정된다.

3.1.3 속도프로파일 생성

본 논문에서는 전역경로계획에서 생성한 곡선경로에 대해 기구학 및 동력학을 고려한 이상적인 속도프로파일을 계산한다. 이는 실제 지역경로계획에서 속도함수에 사용되며, 속도프로파일을 이용하여 로봇이 회전시 충분한감속이 가능하

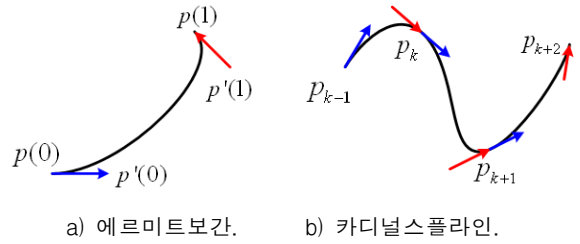


그림 6 곡선생성 예.
Fig. 6 Examples of generating curve.

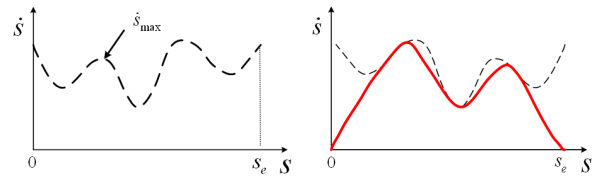


그림 7 속도프로파일.
Fig. 7 The velocity profile.

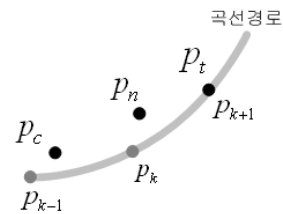


그림 8 지역목표점 설정.
Fig. 8 Set of local target point.

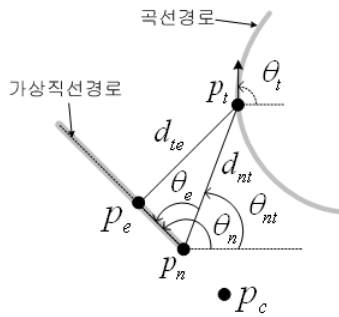
게 된다. 로봇의 속도프로파일을 구하기 위해서는 로봇의 동력학 특성을 고려해야하며, 본 논문에서는 기구학 및 동력학식을 기본으로 하여 참고문헌[12]에서 사용된 궤적계획 알고리즘을 이용하여 속도프로파일을 구한다. 이 궤적계획 알고리즘은 그림 7(a)와 같이 생성된 곡선경로에서 최대 속도범위를 설정하고, 이에 대해 기구학 및 동력학 제약조건에 고려하여 정방향 및 역방향에 대해 속도프로파일을 계산하여, 그림 7(b)와 같은 최종 속도프로파일을 구한다. 최종 속도프로파일은 지역경로계획에서 속도함수로 사용되며, 이는 로봇이 회전 또는 정지 시 속도정보를 제공하여 경로추종시 로봇이 속도에 의해 경로에서 벗어나지 않도록 한다.

3.2 지역경로계획

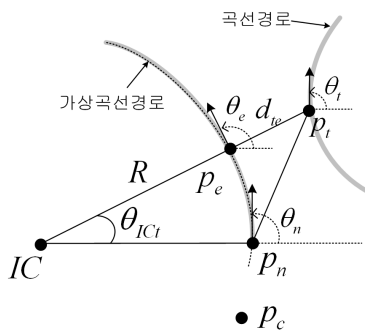
전역경로계획에서 생성한 곡선경로와 속도프로파일을 이용하여 기존의 DWA를 수정한 지역경로계획방법을 제안한다. 수정된 DWA는 미끄러짐이나 새로운 장애물에 강인하고, 경유점 전환 문제를 해결하여, 경로이탈 없이 목표점까지 로봇이 주행이 가능하도록 설계한다. 수정된 DWA는 곡선경로의 한 점을 지역목표점을 설정하고, 가상경로를 생성하여 지역목표점과 유사한 가상경로를 선택하여 주행한다.

3.2.1 지역목표점 설정

지역경로계획에서 목표점을 정의한다. 기존 전역DWA의



a) 가상직선경로.



b) 가상곡선경로.

그림 9 가상경로.
Fig. 9 Virtual paths.

경우 NF1, A*로 생성한 경로에서 직선으로 연결지점이 지역목표점이다. 본 논문에서는 전역경로계획에서 직선경로가 아닌 곡선경로를 생성하였으며, 따라서 직선연결점이 아닌 곡선경로의 한 지점을 지역목표점으로 선택한다.

곡선경로에서 지역목표점 설정을 위해 그림 8과 같이 구간별 파라미터로 표현된 곡선경로에서 현재지점을 p_c , 동적윈도우의 속도에 따라 예상되는 지점을 p_n 이라 정의한다. 지역목표점은 로봇이 진행하게 되는 예상지점 p_n 보다 멀리 있는 곡선경로의 지점을 찾아 지역목표점을 설정한다. 즉 곡선경로의 임의의 지점을 p_k 라 할 때, p_k 와 p_c 사이의 거리가 p_n 과 p_c 사이의 거리보다 크게 되는 p_k 를 지역목표점을 p_t 로 설정한다.

3.2.2 가상경로 생성

로봇이 곡선경로를 따라 이탈 없이 주행하기 위해서는 곡선경로와 실제 주행경로와의 차이가 작고, 곡선경로와 주행경로의 방향의 차이가 작을수록 로봇은 곡선경로를 따라 주행하게 된다. 또한 앞서 전역경로에서 생성한 속도프로파일을 이용하면, 현재 지역목표점에서의 최대속도를 설정할 수 있으며, 이를 통해 급격한 회전 시 빠른 속도에 따른 경로이탈 문제를 해결할 수 있다. 본 논문에서는 동적윈도우의 로봇의 좌우바퀴 속도에 따른 가상경로를 생성하여, 곡선경로의 지역목표점과 거리, 방향, 속도 등을 비교한다.

동적윈도우에서의 가상경로는 로봇이 회전하는 경우와 직진하는 경우로 나누어 생성한다. 먼저 동적윈도우상에서 선

택된 속도가 좌우바퀴가 동일한 경우는 로봇이 직진으로 이동하는 상태로, 이 경우 그림 9(a)와 같이 예상지점 p_n 에서 로봇의 방위각 θ_n 을 기준으로 가상직선경로를 생성한다. 여기서 가상직선경로와 지역목표점 p_t 가 수직으로 만나는 지점을 p_e 로 설정한다. p_e 는 가상직선경로와 p_t 와의 동일선상의 지점을 나타내며, 두 점 사이의 거리는 d_{te} 로 정의하고 다음과 같이 계산한다.

$$d_{te} = d_{nt} \times \sin(\theta_{nt} - \theta_n) \tag{12}$$

여기서 d_{nt} 와 θ_{nt} 는 p_t 와 p_n 사이의 거리 및 사이각 이다. d_{te} 는 그 크기가 0에 가까울수록 가상경로가 곡선경로의 지역목표점에 근접하게 된다. 그러나 d_{te} 가 0에 가까워 지역목표점에 근접하더라도 p_e 지점에서의 방위각 θ_e 과 p_t 에서의 방위각 θ_t 의 차이가 크면 가상경로는 곡선경로를 이탈하게 된다. 따라서 d_{te} 외에도 θ_t 와 θ_e 의 차이를 α 로 정의하고, 다음과 같이 계산한다.

$$\alpha = \theta_e - \theta_t \tag{13}$$

가상직선경로의 경우 $\theta_e = \theta_n$ 동일하다. 따라서 가상직선경로에서는 (12)(13)을 이용하여 d_{te} 와 α 를 계산한다.

다음으로 로봇의 두 바퀴가 서로 다른 경우는 로봇이 좌우바퀴의 속도 차이로 인해 회전 이동하는 상태로, 이 경우 그림 9(b)와 같이 가상곡선경로를 생성한다. 가상곡선경로는 p_n 에서의 선속도 v 및 각속도 w 를 이용하여 회전반경 $R = v/w$ 을 구한다. 회전반경 R 을 이용하면, 가상곡선경로의 중심점은 다음과 같이 계산할 수 있다.

$$\begin{pmatrix} x_{IC} = x_n - R\cos(90 - \theta_n) \\ y_{IC} = y_n + R\sin(90 - \theta_n) \end{pmatrix} \tag{14}$$

중심점과 반경 R 을 이용하여 그림 9(b)와 같이 p_n 에서의 반경 R 을 갖는 가상곡선경로를 생성한다. 가상곡선경로에서도 가상직선경로와 같이 동일선상의 지점 p_e 를 설정한다. p_e 는 회전중심점과 지역목표점 사이를 직선으로 연결하여 이 직선과 가상곡선경로의 교점으로 한다. p_e 를 설정 후 가상곡선경로와 지역목표점의 거리 d_{te} 를 계산한다. 가상곡선경로에서 d_{te} 는 가상곡선경로의 중심점과 지역목표점 p_t 와의 차이로 계산할 수 있다.

$$d_{te} = |R| - \sqrt{(x_t - x_{IC})^2 + (y_t - y_{IC})^2} \tag{15}$$

즉 d_{te} 가 0에 가까울수록 가상곡선경로가 p_t 에 근접한 것이며, 0보다 큰 경우는 가상곡선경로의 회전반경안쪽에 p_t 가 위치한 것이며, 0보다 작은 경우는 회전반경 바깥쪽에 p_t 위치한다.

가상곡선경로에서도 d_{te} 와 더불어 가상곡선경로와의 근접성을 나타내는 방위각차이 α 를 계산한다. 먼저 p_e 에서의 방위각 θ_e 는 다음과 같이 계산한다.

$$\theta_e = \tan^{-1}\left(\frac{x_t - x_{IC}}{y_t - y_{IC}}\right) + \left(90 \times \frac{R}{|R|}\right) \quad (16)$$

가상곡선경로에서의 α 는 가상직선경로와 동일하게 (13)을 이용한다.

3.2.3 목적함수

가상직선경로 및 가상곡선경로는 곡선경로의 지역목표점 p_t 와의 거리 d_{te} 및 방위각 차이 α 가 모두 0에 가까울수록 지역목표점 p_t 에 근접한다. 그러나 두 값중 하나는 0에 근접하나 다른 값이 클 경우에는 오히려 경로를 벗어날 수 있으므로 이 두 값을 적절한 목적함수를 통해 최적의 값을 선택해야한다. 본 논문에서는 기존의 DWA의 목적함수를 수정하여 사용한다. 먼저 전체 목적함수는 다음과 같다.

$$O^* = \gamma_{clear} w_{clear} + \gamma_{distance} w_{distance} + \gamma_{head} w_{head} + \gamma_{speed} w_{speed} \quad (17)$$

여기서 w_{clear} 는 기존 DWA와 동일한 충돌함수이며, w_{head} 는 가상경로와 지역목표점과의 방위각 차이를 나타내며, $w_{distance}$ 는 가상경로와 지역목표점과의 떨어진 거리를 나타내고, w_{speed} 는 지역목표점의 속도대비 현재 로봇의 속도를 함수로 나타낸다. 또한 $\gamma_{clear}, \gamma_{distance}, \gamma_{head}, \gamma_{speed}$ 는 각 목적함수의 가중치를 나타낸다.

동적원도우의 속도에 따른 예상지점 p_n 에서 가상경로 생성하여 d_{te} 와 α 를 계산하였다. d_{te} 와 α 를 이용하여 곡선경로에 근접한 가상경로를 선택할 수 있도록 수정된 DWA의 목적함수를 정의한다.

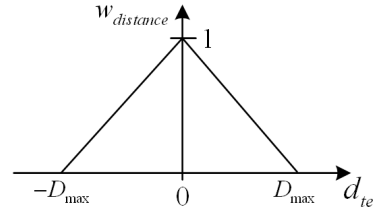
먼저 거리함수는 $w_{distance}$ 로 정의하며, 곡선경로와 가상경로와의 차이를 나타내는 d_{te} 를 정규화한 함수이다. 동적원도우상에서 계산된 d_{te} 를 $d_{te}(h)$, $h \in DW$ 라 정의 할 때, $w_{distance}(h)$ 는 곡선경로와 가상경로와의 거리 최대값 D_{max} 를 이용하여 다음과 같이 계산한다.

$$w_{distance}(h) = \begin{cases} 0 & |d_{te}| > D_{max} \\ 1 - \frac{|d_{te}(h)|}{D_{max}} & |d_{te}| < D_{max} \end{cases} \quad \text{단, } h \in DW \quad (18)$$

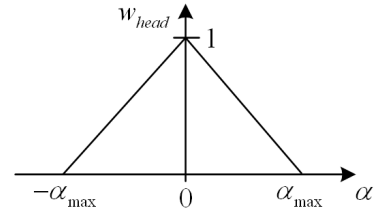
$w_{distance}$ 는 그림 10(a)와 같이 $|d_{te}|$ 가 0일 경우 최대값이 1을 가지며 D_{max} 보다 큰 경우 0이 된다. D_{max} 의 값은 클 경우 로봇이 주행시 바이어스가 커지며, 너무 작을 경우 불안정해 질 수 있다. 본 논문에서는 D_{max} 의 값을 10cm로 설정하였다.

다음으로 곡선경로의 지역목표점과 가상경로의 방위각차이를 나타내는 α 를 이용하여 방향함수 w_{head} 를 정의한다. 지역목표점과 가상경로의 방위각 차이인 α 에서 최대값을 α_{max} 라 정의하고, 다음과 같이 방향함수 w_{head} 를 계산한다.

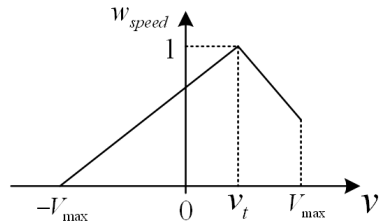
$$w_{head}(h) = \begin{cases} 0 & |\alpha| > \alpha_{max} \\ 1 - \frac{|\alpha(h)|}{\alpha_{max}} & |\alpha| < \alpha_{max} \end{cases} \quad \text{단, } h \in DW \quad (19)$$



a) 거리함수.



b) 방향함수.



b) 속도함수.

그림 10 수정된 DWA의 목적함수.

Fig. 10 Objective functions of modified DWA.

w_{head} 는 $|\alpha|$ 가 0일 때 최대값 1을 가지며, α_{max} 보다 크면 0이 된다. 최대 방위각 차이인 α_{max} 는 너무 작을 경우 잦은 방향전환이 이루어질 수 있으며, 본 논문에서는 $\pi/4$ 를 설정하였다. α 에 대한 방향함수 w_{head} 는 그림 10(b)에 나타내었다.

마지막으로 수정된 DWA 목적함수는 속도함수 w_{speed} 이다. 속도함수는 기존 DWA에서는 최대속도일 때 최대값으로 설정하였으나, 수정된 DWA에서는 곡선경로 생성 후 계산한 속도프로파일을 이용한다. 전역경로계획에서 계산한 속도프로파일은 곡선경로의 임의의 점에서의 허용 가능한 속도를 나타낸다. 따라서 곡선경로상에 있는 지역목표점 p_t 에서도 허용 속도값 v_t 를 가진다. 수정된 DWA의 속도함수 w_{speed} 는 지역목표점의 허용속도 v_t 와 동적원도우상의 선속도 $v(h)$ 를 이용하여 다음과 같이 설정한다.

$$w_{speed}(h) = 1 - \frac{|v_t - v(h)|}{v_t + v_{max}} \quad \text{단, } h \in DW \quad (20)$$

속도함수 w_{speed} 는 그림 10(c)에 나타내었으며, 선속도 $v(h)$ 가 허용속도 v_t 와 같은 경우 최대값 1을 가지게 된다.

수정된 DWA의 거리, 방향 및 속도함수 $w_{distance}, w_{head}, w_{speed}$ 는 동적원도우에서 각각 계산되며, 최종 목적함수 O^* 는 (17)을 이용하여 최대값을 갖는 좌우바퀴의 속도를 선택하게 된다.

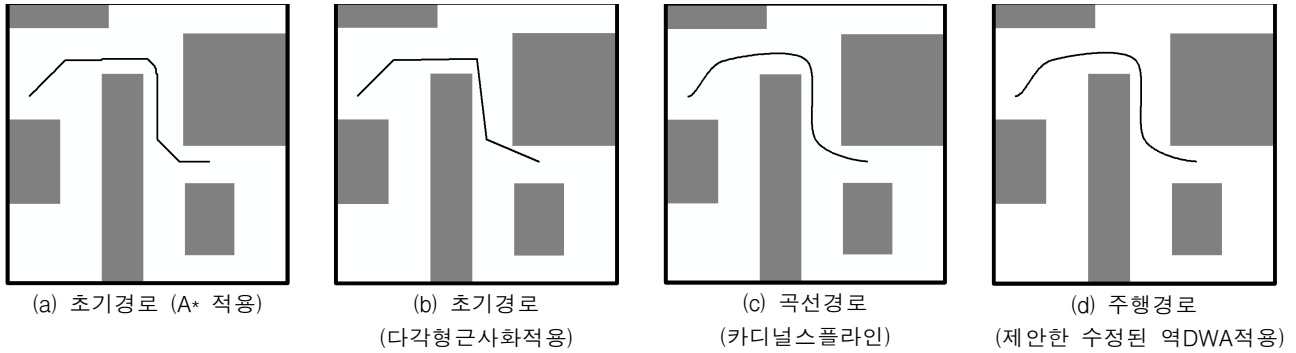


그림 11 경로생성 결과
 Fig. 11 Results of path generation.

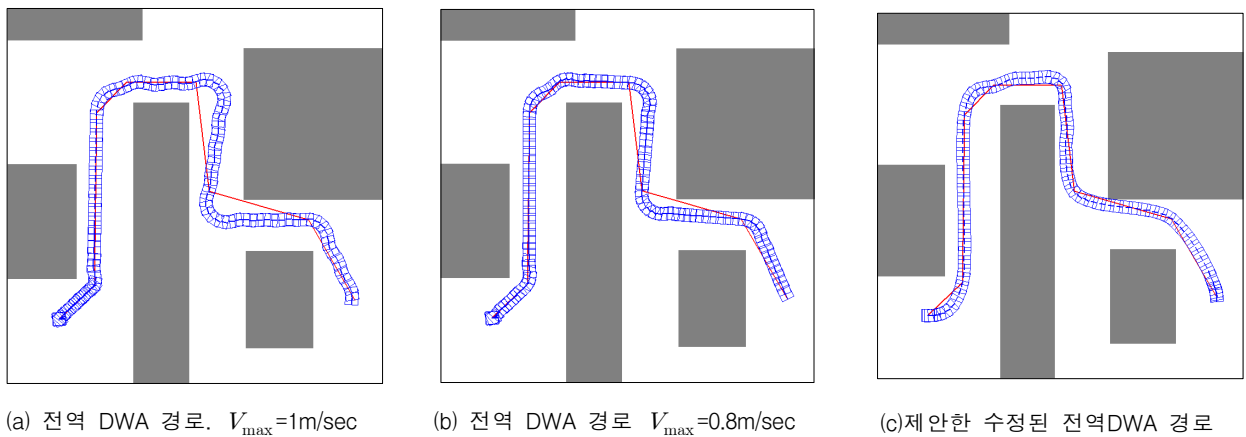


그림 12 전역 DWA 비교.
 Fig. 12 Comparison of global DWA.

4. 시뮬레이션

본 논문에서 제안한 수정된 전역 DWA의 성능 평가를 위해 시뮬레이션을 수행하였다. 먼저 시뮬레이션은 IBM-PC 기반에서 Matlab 7.3을 이용하여 구현하였다. 시뮬레이션에 사용된 로봇은 좌우바퀴 사이의 거리가 0.68m, 최대속도는 1m/sec, 최대가속도는 0.4m/sec²로 설정하였다. 사용된 지도는 20m×20m평면에 장애물이 배치된 지도이며, 전역경로계획에서는 지도의 격자크기를 10cm/grid로 설정하였고, 지역 경로계획에서는 격자 크기를 1cm/grid로 설정하였다. 수정된 DWA의 업데이트 주기는 100msec 이며, 동적윈도우의 크기는 격자 하나당 1cm/sec²로 설정하여, 현재속도와 가속, 감속을 고려하여 총 9×9의 격자로 구성된다.

먼저 제안한 수정된 전역 DWA를 이용한 경로생성과정을 시뮬레이션 하였다. 그림 11은 제안한 수정된 전역 DWA를 이용하여 생성된 경로를 나타낸다. 시작점은 (6.7m, 1.5m, π/2), 목표점은 (11.4m, 14.6m, π/2)인 경우 그림 11(a)는 전역경로계획에서 A*알고리즘으로 생성한 초기경로이며, 그림 11(b)는 다각형근사화 알고리즘을 통해 불필요한 경유점을 제외하고 추출된 최적경유점을 보여준다. 그림 11(c)는 최적경유점에 대해 카디널스플라인으로 시작점 및 목표점에서 방위각을 고려하여 생성한 곡선경로이며, 그림 11(d)는 그림 11(c)의 곡선경로를 지역경로계획에서 수정된

DWA를 이용하여 모의 주행한 경로이다. 전역경로계획에서 생성한 곡선경로와 지역경로계획에서 곡선경로를 따라 모의 주행한 경로가 유사한 것을 볼 수 있다.

기존의 전역 DWA는 지역목표점 전환시 로봇의 속도에 의해 경로이탈 문제가 발생할 수 있으며, 이로 인해 주행이 불안정해지거나 이동시간이 늘어나는 문제점을 가지고 있다. 제안한 수정된 전역 DWA는 이런 문제점을 줄일 수 있으며, 이를 시뮬레이션을 통해 비교하였다. 시뮬레이션에 사용된 전역 DWA의 전역경로는 수정된 전역 DWA와 동일하게 A*알고리즘과 다각형근사화로 생성하였으며, 생성된 경유점들을 지역목표점으로 설정하여 DWA를 적용하였다. 기존 전역 DWA의 경우 로봇의 최대 속도 설정에 따라 경로 이탈이 크게 발생할 수 있다. 그림 12(a)는 가속도는 0.4 m/sec²고정하고, 최대속도를 1m/sec로 설정했을 경우에 전역 DWA를 통해 로봇이 주행한 경로를 보여준다. 지역목표점 전환 시 로봇의 속도로 인해 경로이탈이 발생한 것을 볼 수 있으며, 이 때 이동시간은 약 56초이다. 다음으로 그림 12(b)는 최대속도를 0.8m/sec로 낮추고 전역 DWA를 적용하여 생성한 주행경로를 보여준다. 그림 12(a)에 비해 최대속도를 낮추었지만 이동시간은 약 56초로 유사한 것을 볼 수 있으며, 이는 최대속도를 낮추었지만 경로이탈이 줄어들어 전체 이동시간에는 변화가 없었다. 그러나 여전히 지역목표점 전환시점에서 불안정하게 주행하는 것을 볼 수 있다. 마

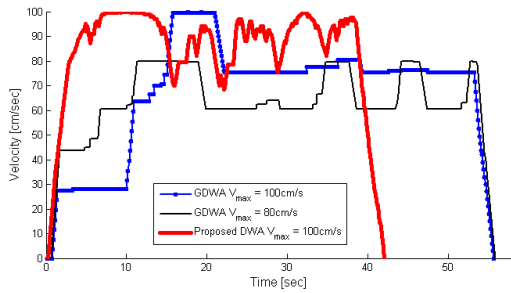


그림 13 속도 비교.
Fig. 13 Comparison of velocity

지막으로 제안한 수정된 전역 DWA는 그림 12(c)와 같이 최대속도는 1m/sec로 설정하였지만, 그림 12(a)의 전역 DWA와 같은 경로이탈이 발생하지 않으며, 주행경로 또한 그림 12(b)에 비해 안정적으로 부드러운 것을 볼 수 있다. 또한 이동시간도 약 42초 정도로 기존 전역 DWA에 비해 빠르게 이동하는 것을 확인할 수 있다.

그림 13은 그림 12의 비교 대상들의 속도를 나타낸다. 기존 DWA는 지역목표점이 전환되는 회전 구간에서 속도가 줄어 전체 이동구간에서 최대속도 비율이 낮은 편이나, 제안한 수정된 전역 DWA는 최대속도에 근접하게 주행한 것을 볼 수 있으며, 이로 인해 전체 주행시간이 단축되었다.

5. 실험

시뮬레이션은 전역경로계획과 지역경로계획을 통해 기존 전역 DWA와 수정된 전역 DWA를 비교하였다. 제안한 수정된 전역DWA는 경로이탈 없이 부드럽게 주행하는 것을 확인하였으며, 이를 실제로 로봇에 적용하여 경로이탈에 대한 성능을 평가하였다. 먼저 실험용 로봇으로는 엔티렉스사의 NT-TriWheel을 기구부를 기초로 STM32F103VB기반 보드를 이용하여 로봇 구동부를 제작하였다. 로봇의 전역경로계획 및 지역경로계획을 하는 주제어기는 Intel Core2duo 기반 PC를 이용하였다. 실험에 사용된 로봇은 그림 14에 나타내었다. 실험은 7m×7m 공간에서 이루어졌으며, 로봇의 최대속도는 1m/sec, 최대가속도는 0.3m/sec²이며, DWA의 업데이트 주기는 100msec로 동적원도우의 크기는 7×7로 구성하였다. 경로이탈여부를 잘 판단할 수 있도록 경유점을 90도로 꺾어서 배치하였으며, 꺾어진 경유점에 대해 기존 전역 DWA와 수정된 전역 DWA를 로봇에 적용하여 경로이탈 정도를 실험하였다.

실험 시 기존 전역 DWA에서 최대속도를 1m/sec로 설정한 경우, 시작점(2m,3m)에서 두 번째 경유점(6m,3m)으로 이동시 경로이탈이 심해, 최종 경유점(6m,7m)에 도달하지 못하였다. 이 문제를 해결하기 위해 최대속도를 0.8m/sec로 줄이고 전역 DWA를 적용한 경우와 최대속도는 그대로 두고 경유점 전환시점을 1m 전방으로 설정하여 전역 DWA를 적용 경우를 제안한 수정된 DWA와 비교하였다.

그림 15는 로봇에 기존 전역 DWA 및 수정된 전역 DWA에 의해 실제 주행한 경로를 나타내며 경유점은 작은 원으로 나타내었다. 최대속도를 1m/sec로 설정하고 경유점

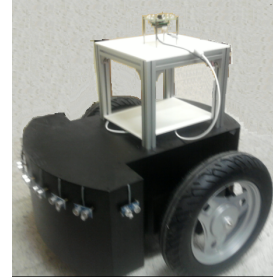


그림 14 NT-Triwheel.
Fig. 14 NT-Triwheel.

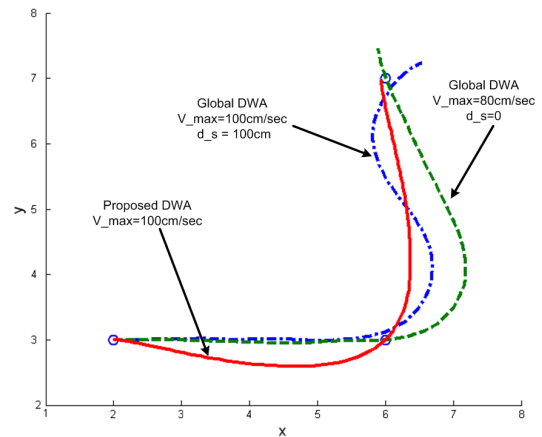


그림 15 실험 주행경로.
Fig. 15 Paths by experiment.

전환을 빠리한 경우엔 경로이탈은 줄일 수 있지만 속도가 충분히 줄이지 않은 상태에서 회전하여 다음 경유점까지 이동시 불안정하게 이동한 것을 확인할 수 있다. 다음으로 최대속도를 0.8m/sec로 줄인 경우에는 최대속도가 1m/sec일 때 보다 안정적인 주행은 가능하지만 경유점 전환 후 경유점 진입 속도에 의해 경로이탈이 보다 심한 것을 볼 수 있다. 마지막으로 제안한 수정된 DWA의 경우 경유점에 대해 생성된 곡선경로를 따라 주행하여, 경로이탈 없이 경유점들을 안정적으로 주행하면서 통과한 것을 확인할 수 있다. 이는 시뮬레이션과 동일한 결과로써 제안한 수정된 DWA가 전역경로계획에서 곡선경로와 속도 프로파일을 생성하여 로봇이 회전시 안정적으로 주행할 수 있는 최대속도를 미리 계산하였고, 이를 기반으로 지역경로계획에서 곡선경로를 따라 주행하도록 설계된 DWA의 목적함수를 통해 경로이탈 없이 안정적으로 주행할 수 있다.

6. 결론

본 논문은 기존의 전역 DWA의 경로이탈을 줄여 전체 이동시간을 단축시키는 수정된 전역 DWA를 제안하였다. 수정된 전역 DWA는 전역경로계획에서 A*알고리즘과 다각형근사화를 통해 최적경유점을 선택하고 이를 카디널스플라인을 이용하여 곡선경로를 생성하고, 곡선경로에 대해 최적의 속도프로파일을 계산하였다. 지역경로계획에서는 곡선경로와 최적속도프로파일을 기준으로, 지역목표점을 설정하여

가상경로를 통해 수정된 DWA의 목적함수를 적용하여 로봇이 부드럽게 주행하도록 하였다.

제안한 수정된 전역DWA를 기존 전역DWA와 비교를 통해 경로이탈을 줄여 안정적으로 주행하고, 이로 인해 전체 이동시간도 단축 되는 것을 시뮬레이션을 통해 확인하였다. 또한 수정된 전역 DWA를 실제 로봇에 적용하여 테스트를 통해 성능을 확인하였다. 제안한 방법은 다수의 알고리즘 적용이 필요한 전역경로계획은 오프라인으로 계산하였으며, 지역경로계획은 온라인으로 실시간으로 처리하여 실제 로봇 시스템에 실용적으로 적용할 수 있으리라 예상된다.

감사의 글

본 연구는 교육과학기술부와 한국연구재단의 지역 혁신인력양성사업으로 수행된 연구임.
이 논문은 2010년 교육과학기술부로부터 지원받아 수행된 연구임(지역거점연구단육성사업/충북 BIT 연구중심대학육성사업단).

참 고 문 헌

[1] R. Siegwart, I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, The MIT Press, 2004.
[2] D. Fox, W. Burgard, S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics and Automation Magazine, pp. 23-33, 1997.
[3] E. Demeester, M. Nuttin, D. Vanhooydonck, G. Vanacker, H. Van Brussel, "Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2694 - 2699, 2005.
[4] S. Qinlan, O.Khatib, "Elastic bands: connecting path planning and control," In Proceedings of IEEE Int. Conf. on Robotics and Automation, 1993.
[5] J. Borenstein, Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," Journal of Robotics and Automation, vol. 7, pp. 278-288, 1999.
[6] O. Brock, O. Khatib, "High-speed navigation using the global dynamic window approach," In Proceeding of the IEEE Intl. Conf. on Robotics and Automation, pp. 341-346, 1999.
[7] R.Philippsen, R.Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot", In Proceeding of the IEEE Intl. Conf. on Robotics and Automation, pp. 446-451, 2003.
[8] A. Stentz, "Optimal and Efficient Path Planning for Partially - Known Environments," ICRA'94, IEEE International Conference on Robotics and Automation, vol. 4, pp. 3310-3317, 1994.
[9] 함종규, 박중태, 송재복, "최적경유점을 갖는 전역 DWA에 기반한 이동로봇의 주행", 제어자동화시스템공학 논문지, 13권, 7호, 624-630, 2007.

[10] V. B. Anand, computer graphics & geometric modeling for engineers, J. Wiley, 1993.
[11] 윤희상, 박태형, "카디널스플라인을 이용한 자율이동로봇의 곡선경로 생성방법", 전기학회논문지, 59권, 4호, 803-808, 2010.
[12] 윤희상, 박태형, "동적프로그래밍을 이용한 자율이동로봇의 동작계획", 제어자동화시스템공학 논문지, 16권, 1호, 53-60, 2010.
[13] M. Yamamoto, M. Iwamura, A. Mohri, "Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles," IEEE/RSJ Int. Conf. On Intelligent Robot and Systems, vol. 4. pp. 2958-2963, 1999.
[14] M. Seder, K. Macek, I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," in In Proceeding of the 31st Annual Conference of the IEEE Industrial Electronics Society, pp. 1785-1790, 2005.
[15] D. Douglas, T. Peucker, "Algorithms for the reduction of the number of points required for represent a digitized line or its caricature," Canadian Cartographer, pp. 112-122, 1973.

저 자 소 개



윤 희 상 (尹 熙 相)

2004년 충북대 전기전자컴퓨터공학부 졸업. 2006년 동 대학원 제어계측공학과(석사). 2006년~2007년 충북대 부설 유비쿼터스 바이오정보기술 연구센터 주임연구원. 2007년~현재 충북대 제어로봇공학과 박사과정. 관심분야는 로보틱스, 영상처리, 임베디드시스템 등.



박 태 형 (朴 泰 炯)

1988년 서울대 제어계측공학과 졸업. 1992년 동 대학원 제어계측공학과(석사). 1994년 동 대학원 제어계측공학과(박사). 1994년~1997년 삼성 테크윈(주) 정밀기기 연구소 선임 연구원. 1997년~현재 충북대 전자공학부 교수. 2000년~2001년 Univ. of Toronto 방문교수. 관심분야는 로보틱스 및 자동화, 전자조립 및 검사 시스템 등.