



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년10월12일
(11) 등록번호 10-1784500
(24) 등록일자 2017년09월27일

(51) 국제특허분류(Int. Cl.)
B25J 9/16 (2006.01)

(52) CPC특허분류
B25J 9/1666 (2013.01)
B25J 9/162 (2013.01)

(21) 출원번호 10-2016-0073692

(22) 출원일자 2016년06월14일

심사청구일자 2016년06월14일

(56) 선행기술조사문헌

KR101503903 B1

JP2013152551 A

(73) 특허권자

인하대학교 산학협력단

인천광역시 남구 인하로 100, 인하대학교 (용현동)

(72) 발명자

노진홍

인천광역시 남동구 소래역로 94 1104동 601호 (논현동, 단풍마을휴먼시아11단지)

허옥열

서울특별시 성동구 한림말길 50 (옥수동, 옥수하이츠아파트) 101동 502호

(뒷면에 계속)

(74) 대리인

이원희

전체 청구항 수 : 총 5 항

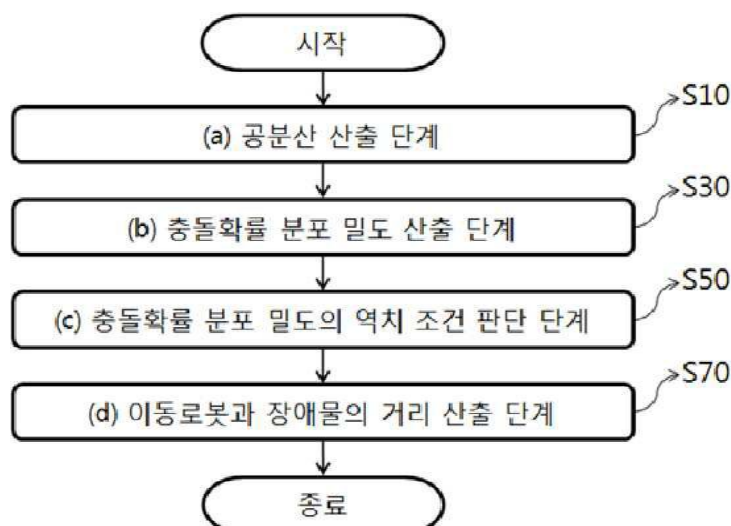
심사관 : 김태수

(54) 발명의 명칭 **확률론적 환경에서 이동로봇의 장애물 회피를 위한 거리 결정 방법**

(57) 요약

본 발명은, 이동로봇의 장애물 회피를 위한 거리 결정 방법에 있어서, 상기 이동로봇의 위치 및 공분산, 상기 장애물의 위치 및 공분산을 산출하는 (a)단계; 상기 이동로봇과 상기 장애물의 상대적인 충돌확률 분포를 공분산의 변수로 표현하고, 상기 충돌확률 분포의 좌표계를 고유벡터로 변환하여 충돌확률 분포 밀도를 산출하는 (b)단계; 상기 충돌확률 분포 밀도와 상기 이동로봇의 위치 간 마할라노비스(Mahalanobis distance)의 거리 및 상기 충돌확률 분포 밀도의 역치 조건을 이용하여 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는지 판단하는 (c)단계; 및 상기 (c)단계에서 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는 것으로 판단되는 경우, 라그랑주 승수를 이용하여 상기 이동로봇과 상기 장애물 사이의 거리를 산출하는 (d)단계를 포함하여, 위치의 불확실성을 고려하여 이동 경로를 산출한다.

대표도 - 도1



(52) CPC특허분류
B25J 9/1676 (2013.01)

(72) 발명자

박중훈

서울특별시 용산구 이촌로 100-14 (이촌동, 스카이
빌라) 301호

장성룡

경기도 안산시 단원구 광덕동로 78 (고잔동, 네오
빌주공아파트) 609동 202호

공지예외적용 : 있음

명세서

청구범위

청구항 1

이동로봇의 장애물 회피를 위한 거리 결정 방법에 있어서,

- (a) 상기 이동로봇의 위치 및 공분산, 상기 장애물의 위치 및 공분산을 산출하는 단계;
- (b) 상기 이동로봇과 상기 장애물의 상대적인 충돌확률 분포를 공분산의 변수로 표현하고, 상기 충돌확률 분포의 좌표계를 고유벡터로 변환하여 충돌확률 분포 밀도를 산출하는 단계;
- (c) 상기 충돌확률 분포 밀도와 상기 이동로봇의 위치 간 마할라노비스(Mahalanobis distance)의 거리 및 상기 충돌확률 분포 밀도의 역치 조건을 이용하여 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는지 판단하는 단계; 및
- (d) 상기 (c)단계에서 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는 것으로 판단되는 경우, 라그랑주 승수를 이용하여 상기 이동로봇과 상기 장애물 사이의 거리를 산출하는 단계를 포함하여, 위치의 불확실성을 고려한 것을 특징으로 하는 이동로봇의 장애물 회피를 위한 거리 결정 방법.

청구항 2

제 1 항에 있어서,

상기 (c)단계는,

하기의 [관계식 1]에 따라 마할라노비스(Mahalanobis distance)의 거리와 상기 충돌확률 분포 밀도의 역치 조건을 판단하는 것을 특징으로 하는 이동로봇의 장애물 회피를 위한 거리 결정 방법.

[관계식 1]

$$d_M^2(\mathbf{0}, S^2; \mathbf{s}) \leq -2 \ln \left(\rho \sqrt{(2\pi)^n |S|^2} \right)$$

여기서, d_M 은 마할라노비스(Mahalanobis distance)의 거리, $\mathbf{s}$ 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치를 의미한다.

청구항 3

제 1 항에 있어서,

상기 (d)단계는,

상기 (c)단계에서 상기 충돌확률 분포 밀도가 역치보다 높은 조건에서 최단 거리를 하기의 [수학식 1]에 따라 산출하는 것을 특징으로 하는 이동로봇의 장애물 회피를 위한 거리 결정 방법.

[수학식 1]

$$d_u(\mathbf{s}, S^2; \rho) = \{\min \|\mathbf{e} - \mathbf{s}\| : \mathbf{e} \in E(S^2; \rho)\}$$

여기서, d_u 는 상기 이동로봇과 상기 장애물 사이의 거리, s 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌 확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치, e 는 E 에 포함된 점, E 는 상기 장애물의 영역을 의미한다.

청구항 4

제 3 항에 있어서,

상기 (d)단계는,

상기 [수학식 1]의 해를 산출하기 위하여 라그랑주 승수를 이용한 알고리즘(root-finding algorithm)을 사용하고, 상기 알고리즘은 하기의 [수학식 2]에 따른 것을 특징으로 하는 이동로봇의 장애물 회피를 위한 거리 결정 방법.

[수학식 2]

$$L(e, s, S^2; \rho, \lambda) = f(e, s) + \lambda g(e, S^2; \rho)$$

여기서, L 은 [수학식 1]에 대한 라그랑지언(Lagrangian), s 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, λ 는 라그랑주 승수, ρ 는 상기 충돌확률 분포 밀도의 역치, g 는 제약조건 함수, f 는 최적화 대상 함수, e 는 장애물의 영역에 포함된 점을 의미한다.

청구항 5

제 4 항에 있어서,

상기 (d)단계는,

상기 [수학식 2]의 연산 범위 조건을 하기의 [수학식 3]에 따라 산출하는 것을 특징으로 하는 이동로봇의 장애물 회피를 위한 거리 결정 방법.

[수학식 3]

$$g(\lambda) = 0, \quad g(\lambda) = g(s, S^2; \rho, \lambda)$$

여기서, g 는 제약조건 함수, s 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌 확률 분포 밀도의 역치, λ 는 라그랑주 승수를 의미한다.

발명의 설명

기술 분야

[0001] 본 발명은 이동로봇의 장애물 회피를 위한 거리 결정 방법에 관한 것으로서, 특히 장애물 위치의 불확실성을 고려하여 확률론적 환경에서 이동로봇의 장애물 회피를 가능하게 하는 거리 결정 방법에 관한 것이다.

배경 기술

[0003] 이동 로봇은 목표 지점까지 장애물과의 충돌 없이 이동이 가능해야 한다. 이를 위해서 이동 로봇의 장애물 회피를 위한 많은 접근법들이 제안되어 왔다. 로봇은 장애물의 위치와 로봇의 위치가 정확하게 결정된다면 충돌없이

이동 경로를 설정할 수 있다. 하지만, 로봇은 측정시의 노이즈가 존재하는 환경 하에서 제어되는 것이 일반적이다. 이에 따라, 로봇은 장애물의 위치를 확률론적인 센서 모델에 의존하여 평가하여야 한다.

[0004] 충돌 확률의 분포는 위치에 대한 공분산과 연관될 수 있다. 그러나, 충돌 확률 분포는 경계가 없기 때문에 로봇은 장애물로부터 의도한 거리를 유지할 수 없게 된다. 그러므로, 로봇은 충돌 확률 분포의 경계 조건(boundary)을 결정하는 능력을 소유해야 하고, 이 경우 거리 계산은 경계로부터의 거리로 계산될 수 있을 것이다. 종래의 거리결정 접근법을 살펴보면 다음과 같다.

[0005] 가장 간단한 접근법으로, 위치에 대한 공분산을 계산하지 않고 거리를 산출하는 방법이 있다. 만약, 높은 정확도의 측정 장비, 센서 등의 사용이 가능하다면 이동 로봇의 장애물 회피에 있어서 공분산의 고려 없이 거리를 계산하는 본 방법이 가장 빠르고 손쉬운 해결책이 될 수 있다. 그러나, 공분산은 주변 환경에 영향을 고려하는 변수로서, 장애물의 이동이나 예측은 공분산을 증가시킨다. 이러한 환경적 요인을 생각하면, 위치의 불확실성을 고려할 수 있는 다른 접근법을 고려함이 바람직하다.

[0006] 다른 접근법으로, 점유 격자(occupancy grid)를 사용하여 충돌 확률 분포를 계산하는 방법이 있다. 장애물들은 점유 격자 내에 표현되고, 장애물과의 거리는 가장 근접한 점유 격자 유닛까지의 거리로 표현된다. 점유 격자는 전술한 첫 번째 방법과는 달리 위치의 불확실성을 고려한 접근법이다. 그러나, 점유 격자를 이용한 이동 로봇의 경로 계산은 많은 연산량 때문에 다수의 시간을 요하는 단점이 있다.

[0007] 또 다른 접근법으로, 오차 타원(Error ellipsoid) 방법이 있다. 그러나, 오차 타원 방법은 통계학 기반의 중요성에 의존한 접근법이다. 오차 타원 방법에는 거리 또는 충돌 확률과 같은 물리학적 개념이 포함되지 않아, 실제 적용시 비교적 큰 오차가 유발되는 단점이 있다.

[0008] 전술한 세 가지의 대표적인 접근 방법은 이동 로봇과 장애물의 거리를 결정함에 있어서 적합하지 못한 한계를 갖는다. 위치의 불확실성 때문에, 연산 시간과 충돌 확률은 충돌의 확률을 무시할 수 있을 때까지 계산되어야 한다. 장애물 회피 방법은 이동 로봇과 장애물 사이에서 환경을 고려하여 거리를 계산하는 것이 요구된다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 한국 공개특허공보 제10-2011-0047844호

발명의 내용

해결하려는 과제

[0011] 따라서 본 발명은 위치의 불확실성을 고려한 이동로봇의 장애물 회피를 위한 거리 결정 방법을 제공하고자 한다. 보다 상세하게, 본 발명은 이동로봇과 장애물 사이의 상대적인 위치와 공분산을 파악하여, 특정 충돌확률 밀도를 기준으로 할 때 상대적인 분포와의 거리를 라그랑주 승수법을 이용하여 계산할 수 있는 거리 결정 방법을 제공하고자 한다. 이 경우, 연산 범위를 설정하여 해를 계산함으로써 적은 연산량과 빠른 속도로 충돌확률을 고려한 거리 결정 방법을 제공하고자 한다.

과제의 해결 수단

[0013] 상기 목적을 달성하기 위하여 본 발명은, 이동로봇의 장애물 회피를 위한 거리 결정 방법에 있어서, 상기 이동로봇의 위치 및 공분산, 상기 장애물의 위치 및 공분산을 산출하는 (a)단계; 상기 이동로봇과 상기 장애물의 상대적인 충돌확률 분포를 공분산의 변수로 표현하고, 상기 충돌확률 분포의 좌표계를 고유벡터로 변환하여 충돌확률 분포 밀도를 산출하는 (b)단계; 상기 충돌확률 분포 밀도와 상기 이동로봇의 위치 간 마할라노비스(Mahalanobis distance)의 거리 및 상기 충돌확률 분포 밀도의 역치 조건을 이용하여 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는지 판단하는 (c)단계; 및 상기 (c)단계에서 상기 이동로봇의 위치가 상기 장애물의 영역 외에 존재하는 것으로 판단되는 경우, 라그랑주 승수를 이용하여 상기 이동로봇과 상기 장애물 사이의 거리를 산출하는 (d)단계를 포함하여, 위치의 불확실성을 고려한 것을 특징으로 한다.

[0014] 바람직하게, 상기 (c)단계는 하기의 [관계식 1]에 따라 마할라노비스(Mahalanobis distance)의 거리와 상기 충

돌확률 분포 밀도의 역치 조건을 판단할 수 있다.

[관계식 1]

$$d_M^2(\mathbf{0}, S^2; \mathbf{s}) \leq -2 \ln \left(\rho \sqrt{(2\pi)^n |S|^2} \right)$$

여기서, d_M 은 마할라노비스(Mahalanobis distance)의 거리, $\mathbf{s}$ 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치를 의미한다.

바람직하게, 상기 (d)단계는 상기 (c)단계에서 상기 충돌확률 분포 밀도가 역치보다 높은 조건에서 최단 거리를 하기의 [수학식 1]에 따라 산출할 수 있다.

[수학식 1]

$$d_u(\mathbf{s}, S^2; \rho) = \{\min \|\mathbf{e} - \mathbf{s}\| : \mathbf{e} \in E(S^2; \rho)\}$$

여기서, d_u 는 상기 이동로봇과 상기 장애물 사이의 거리, $\mathbf{s}$ 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치, $\mathbf{e}$ 는 E 에 포함된 점, E 는 상기 장애물의 영역을 의미한다.

바람직하게, 상기 (d)단계는 상기 [수학식 1]의 해를 산출하기 위하여 라그랑주 승수를 이용한 알고리즘(root-finding algorithm)을 사용하고, 상기 알고리즘은 하기의 [수학식 2]에 따른 것일 수 있다.

[수학식 2]

$$L(\mathbf{e}, \mathbf{s}, S^2; \rho, \lambda) = f(\mathbf{e}, \mathbf{s}) + \lambda g(\mathbf{e}, S^2; \rho)$$

여기서, L 은 [수학식 1]에 대한 라그랑지언(Lagrangian), $\mathbf{s}$ 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, λ 는 라그랑주 승수, ρ 는 상기 충돌확률 분포 밀도의 역치, g 는 제약조건 함수, f 는 최적화 대상 함수, $\mathbf{e}$ 는 장애물의 영역에 포함된 점을 의미한다.

바람직하게, 상기 (d)단계는 상기 [수학식 2]의 연산 범위 조건을 하기의 [수학식 3]에 따라 산출할 수 있다.

[수학식 3]

$$g(\lambda) = 0, \quad g(\lambda) = g(\mathbf{s}, S^2; \rho, \lambda)$$

여기서, g 는 제약조건 함수, $\mathbf{s}$ 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치, λ 는 라그랑주 승수를 의미한다.

발명의 효과

본 발명에 따르면, 로봇과 장애물의 공분산을 입력 변수로 하여 위치의 불확실성을 고려한 거리를 라그랑주 지수 및 연산 범위의 조건을 설정하여 재귀적 알고리즘으로 해를 산출한다. 따라서, 본 발명은 노이즈가 존재하는 환경 하에서 제어될 수 밖에 없는 이동로봇에 있어서, 확률론적 환경을 고려하여 거리를 계산하기 때문에 이동로봇의 무인 제어, 이동로봇의 자체 경로 설정에 특히 적합한 방법을 제공한다.

또한, 본 발명에 따른 방법은 거리 계산의 개념으로, 적용에 대한 별도의 고려가 필요하지 않다. 이 경우, 유클리드 거리 대신 본 발명에 따라 계산된 거리를 사용한다면 기존의 경로계획법에도 위치 불확실성을 고려하는 기

능을 추가할 수 있다. 이동로봇은 자신과 장애물의 위치 추정상태에 따라 여유거리를 조절할 수 있을 것이다.

도면의 간단한 설명

- [0034] 도 1은 본 발명의 실시예에 따른 이동로봇의 장애물 회피를 위한 거리 결정 방법의 흐름도를 나타낸다.
- 도 2는 (a)단계의 수행결과 표현된 이동로봇과 장애물의 각 영역 및 장애물 영역에 대한 로봇의 상대적인 위치를 그래프로 출력한 모습이다.
- 도 3은 본 발명의 실시예에 따른 이동로봇의 장애물 회피를 위한 거리 결정 방법의 알고리즘을 수행한 시뮬레이션 결과를 나타낸다.
- 도 4는 실험을 위한 무작위로 장애물이 배치된 상황의 시작점 환경과 이동로봇을 컴퓨터 상에 맵핑한 디스플레이 화면을 나타낸다.
- 도 5는 도 4의 실험환경에서 장애물의 각 영역을 설정하여 이를 회피하는 이동로봇 경로지점 및 목표지점까지의 이동결과를 표시한 실험 결과를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0035] 이하, 첨부된 도면들에 기재된 내용들을 참조하여 본 발명을 상세히 설명한다. 다만, 본 발명이 예시적 실시 예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일 참조부호는 실질적으로 동일한 기능을 수행하는 부재를 나타낸다.
- [0036] 본 발명의 목적 및 효과는 하기의 설명에 의해서 자연스럽게 이해되거나 보다 분명해 질 수 있으며, 하기의 기재만으로 본 발명의 목적 및 효과가 제한되는 것은 아니다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이, 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다.
- [0037] 도 1은 본 발명의 실시예에 따른 이동로봇의 장애물 회피를 위한 거리 결정 방법의 흐름도를 나타낸다. 도 1을 참조하면, 본 발명의 실시예에 따른 이동로봇의 장애물 회피를 위한 거리 결정 방법은 공분산을 산출하는 (a)단계(S10), 충돌확률 분포 밀도를 산출하는 (b)단계(S30), 충돌확률 분포 밀도의 역치 조건을 판단하는 (c)단계(S50), 및 이동로봇과 장애물의 거리를 산출하는 (d)단계(S70)를 포함할 수 있다.
- [0038] 본 발명의 실시예에 따른 상기의 (a) 내지 (d)단계는 컴퓨터 수단을 통해 수행되는 프로그램 명령 형태로 수행될 수 있으며, 이는 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 포함한다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 설계되거나 컴퓨터 소프트웨어 담당자에게 공지되어 사용 가능한 것일 수 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크, 자기 매체, CD-ROM, DVD와 같은 광기록 매체(Optical media), 플롭티컬 디스크(Floptical disk)와 같은 자기-광 매체(Magneto-optical media), 롬(ROM), 램(RAM), 및 플래시 메모리 등의 프로그램 명령을 저장하고 수행하도록 구성되는 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 인터프리터를 사용하는 컴퓨터의 고급 언어 코드를 포함한다. 본 실시예로, (a) 내지 (d)단계는 컴퓨터 프로세서에 의하여 수행되며 이동로봇의 장애물 회피를 위한 거리결정 프로그램이 저장된 기록매체에 저장될 수 있다. 이하, 컴퓨터에서 수행되는 각 단계를 설명한다.
- [0039] (a)단계(S10)는 이동로봇의 위치 및 공분산, 장애물의 위치 및 공분산을 산출할 수 있다. 도 2는 (a)단계(S10)의 수행결과 표현된 이동로봇과 장애물의 각 영역 및 장애물 영역에 대한 로봇의 상대적인 위치를 그래프로 출력한 모습이다.
- [0040] $N(r, R)$ 는 이동 로봇의 평가된 위치를 나타낸다. 여기서 'N'은 다변수의 정규 분포이고, 'r'은 위치 벡터의 평균이며, 'R'은 공분산 행렬을 의미한다. 이와 유사하게 $N(q, Q)$ 는 장애물의 평가된 위치를 나타낸다. 도 2의 (a)를 참조하면, 이동로봇의 위치는 $N\left(\begin{bmatrix} 0.20 \\ 0.25 \end{bmatrix}, \begin{bmatrix} 0.004 & 0.002 \\ 0.002 & 0.002 \end{bmatrix}\right)$, 장애물의 위치는 $N\left(\begin{bmatrix} -0.07 \\ -0.10 \end{bmatrix}, \begin{bmatrix} 0.012 & -0.003 \\ -0.003 & 0.006 \end{bmatrix}\right)$ 로 나타낼 수 있다. 도 2의 (b)는 상대적인 위치를 나타낸

것으로, Relative position은 $\begin{bmatrix} -0.3804 \\ -0.2252 \end{bmatrix}$ 로서 푸른색 '+' 로 표현되었고, 장애물의 영역인 Obstacle region은 $E\left(\begin{bmatrix} 0.0079 & 0 \\ 0 & 0.0161 \end{bmatrix}; 10^{-2}\right)$ 로서 붉은색 타원형으로 표현되었다. 본 예시에서 상대적인 위치와 장애물 영역의 가장 가까운 거리는 0.0768[m]로 계산된다.

[0041] (a)단계(S10)는 이동로봇의 거리 결정 계산에 요구되는 입력 변수값을 설정하는 단계로 이해될 수 있다. 거리 계산을 위해서 설정되어야 하는 변수로는 이동로봇의 위치와 공분산(r,R), 장애물의 위치와 공분산(q,Q), 충돌 확률 분포 밀도의 역치(ρ), 허용 거리오차(ϵ)가 될 수 있다.

[0042] (b)단계(S30)는 이동로봇과 장애물의 상대적인 충돌확률 분포를 공분산의 변수로 표현하고, 충돌확률 분포의 좌표계를 고유벡터로 변환하여 충돌확률 분포 밀도를 산출할 수 있다.

[0043] (b)단계(S30)에서는 (a)단계(S30)에서 설정된 이동로봇의 위치와 공분산 N(r,R) 및 장애물의 위치와 공분산 N(q,Q)을 토대로 상대적인 충돌확률 분포를 표현할 수 있는데, 이는 N(r-q,R+Q)와 같이 나타낼 수 있다. 이후, N(r-q,R+Q)는 충돌 회피를 위해 분석된다.

[0044] 분석을 위해서, 새로운 좌표 시스템이 수행된다. 공분산의 고유 벡터는 충돌확률 분포를 나타내는 $Q + R = VS^2V^T$ (고유값 분해)로 표현될 수 있다. 여기서 $V = [v_1, \dots, v_n]$ 이고, Q+R의 고유벡터 ' v_i '를 구성성분으로 하는 $n \times n$ 의 스퀘어 행렬이다. 'S'는 충돌확률 분포 밀도이며 $S = \text{diag}_i(\sigma_i)$ 이고, 표준 편차 σ_i 의 대각 행렬이다. 새로운 좌표계에서 'V'를 사용하면, 충돌 확률 분포는 $s \sim N(0, S^2)$ 로 표현될 수 있다. 여기서 '0'은 제로 벡터를 의미한다. 또한, 이동 로봇의 상대 위치는 $s = V^T(r - q)$ 로 표현될 수 있다. 이후, 충돌확률 분포 밀도는 's' 위치에서의 밀도를 의미한다. 실제로, 이동로봇은 항상 충돌확률 분포 밀도가 0이 아닌 값을 갖는다. 그러므로 이동로봇을 위치시킬 타겟 지점을 충돌확률 없이 설정시키기란 불가능하다. 이러한 상황에서, 높은 충돌 확률의 영역을 피하는 것으로 경로를 설정하는 것이 가장 바람직할 것이다. 이러한 영역은 충돌확률 분포 밀도의 역치 ' ρ ' 로 설정된다. 만약 어느 위치이든 역치 ' ρ ' 값보다 충돌확률 분포 밀도가 높다면, 이는 장애물이 존재하는 영역으로 이해될 수 있다. 상기의 비교 과정은 하기의 (c)단계(S50)에서 수행된다.

[0045] (c)단계(S50)는 충돌확률 분포 밀도와 이동로봇의 위치 간 마할라노비스의 거리 및 충돌확률 분포 밀도의 역치 조건을 이용하여 이동로봇의 위치가 장애물의 영역 외에 존재하는지 판단할 수 있다.

[0046] (c)단계(S50)는 하기의 [관계식 1]에 따라 마할라노비스(Mahalanobis)의 거리와 충돌확률 분포 밀도의 역치 조건을 판단할 수 있다.

[0047] [관계식 1]

$$d_M^2(0, S^2; s) \leq -2 \ln(\rho \sqrt{(2\pi)^n |S|^2})$$

[0048]

[0049] 여기서, d_M 은 마할라노비스(Mahalanobis)의 거리, s는 이동로봇의 상대적 위치, S는 충돌확률 분포 밀도, ρ 는 충돌확률 분포 밀도의 역치를 의미한다. 장애물 영역의 경계는 $N(0, S^2)$ 의 마할라노비스 거리와 동일하므로, 상기의 [관계식]을 통해 이동로봇의 상대적 위치가 장애물 영역의 내외에 있는지 여부를 판단할 수 있다.

[0050] (d)단계(S70)는 (c)단계(S50)에서 이동로봇의 위치가 장애물의 영역 외에 존재하는 것으로 판단되는 경우, 라그

량주 승수를 이용하여 이동로봇과 장애물 사이의 거리를 산출할 수 있다.

(d)단계(S70)는 (c)단계(S50)에서 상기 충돌확률 분포 밀도가 역치보다 높은 조건에서 최단 거리를 하기의 [수학식 1]에 따라 산출할 수 있다.

[수학식 1]

$$d_u(s, S^2; \rho) = \{\min \|e - s\| : e \in E(S^2; \rho)\}$$

여기서, d_u 는 이동로봇과 장애물 사이의 거리, s 는 상기 이동로봇의 상대적 위치, S 는 충돌확률 분포 밀도, ρ 는 충돌확률 분포 밀도의 역치, e 는 E 에 포함된 점, E 는 상기 장애물의 영역을 의미한다.

여기서 역치 ' ρ ' 는 하기의 [관계식 2]의 조건을 만족한다.

[관계식 2]

$$\bar{E}(S^2; \rho) \geq 0 \Leftrightarrow \rho \in (0, (2\pi)^{-n/2} |S_{max}|^{-1}]$$

여기서, $\bar{E}$ 는 장애물의 경계 영역을 의미한다. 장애물 E 는 확률론적인 환경하에 주의가 요구되는 영역상에 부피를 갖고 존재한다. 만약, $|S_{max}|$ 값이 알려졌다면 상기의 [관계식 2]에 따라 역치의 범위가 결정된다.

$|S_{max}|$ 값은 센서 모델 또는 실험을 통해서 획득될 수 있다.

(d)단계(S70)는 [수학식 1]의 해를 산출하기 위하여 라그랑주 승수를 이용한 알고리즘(root-finding algorithm)을 사용하고, 상기 알고리즘은 하기의 [수학식 2]에 따른다.

[수학식 2]

$$L(e, s, S^2; \rho, \lambda) = f(e, s) + \lambda g(e, S^2; \rho)$$

여기서, L 은 [수학식 1]에 대한 라그랑지언(Lagrangian), s 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, λ 는 라그랑주 승수, ρ 는 상기 충돌확률 분포 밀도의 역치, g 는 제약조건 함수, f 는 최적화 대상 함수, e 는 장애물의 영역에 포함된 점을 의미한다.

(d)단계(S70)는 [수학식 2]의 연산 범위 조건을 하기의 [수학식 3]에 따라 산출할 수 있다.

[수학식 3]

$$g(\lambda) = 0, \quad g(\lambda) = g(s, S^2; \rho, \lambda)$$

여기서, g 는 제약조건 함수, s 는 상기 이동로봇의 상대적 위치, S 는 상기 충돌확률 분포 밀도, ρ 는 상기 충돌확률 분포 밀도의 역치, λ 는 라그랑주 승수를 의미한다.

즉, [수학식 2]의 해를 찾는 알고리즘(root-finding algorithm)은 [수학식 3]의 연산 범위 내에서 조건을 만족하는 해를 구하게 된다. [수학식 3]에서 $g(\lambda)$ 는 $\lambda \in [a, b]$ (조건 1)에서 연속적이고, $g(a)g(b) \leq 0$ (조건 2)를 만족해야 한다. 따라서 $[a, b]$ 의 간격은 조건 1,2를 만족할 수 있도록 설정되어야 한다.

연산 범위의 초기화 및 상기 $[a, b]$ 를 산출하기 위한 알고리즘의 실시예는 하기의 [그림 1]과 같다.

[0069] [그림 1]

입력:	상대위치와 공분산 (s, S) 중동확률밀도 역치 ρ
출력:	초기화된 범위 $[a, b]$
01 $a \leftarrow -\sigma_{min}^2$ 02 $b \leftarrow -\sigma_{min}^2 + \sqrt{\left(-2 \ln \left(\rho \sqrt{(2\pi)^n S ^2}\right)\right)^{-1} \sum_{i=1}^n s_i^2 \sigma_i^2}$ 03 if $b < 0$ then $c \leftarrow (a+b)/2$ 04 else $c \leftarrow 0$ 05 while $g(c) < 0$ 06 $b \leftarrow c, c \leftarrow (a+b)/2$ 07 return $[c, b]$	

[0070]

[0071] 상기의 알고리즘으로 초기화된 연산 범위 내에서 조건을 만족시키는 해를 구하는 root-finding 알고리즘을 구현한 실시예는 [그림 2]와 같다.

[0072] [그림 2]

입력:	로봇의 위치와 공분산 (r, R) 장애물의 위치와 공분산 (q, Q) 중동확률밀도 역치 ρ 허용 거리오차 ϵ 초기 검색범위 $[a, b]$
출력:	라그랑주 승수 λ
01 if $ g(a) < g(b) $ then swap(a, b) 02 $c \leftarrow a, m \leftarrow true$ 03 while $(g(b) \neq 0) \wedge (g(z) \neq 0) \wedge (\sqrt{f(b)} - \sqrt{f(a)} > \epsilon)$ 04 if $(g(a) \neq g(b)) \wedge (g(b) \neq g(c))$ then 05 $z \leftarrow \frac{\{y-g(a)\}\{y-g(b)\}}{\{g(c)-g(a)\}\{g(c)-g(b)\}}c + \frac{\{y-g(b)\}\{y-g(c)\}}{\{g(a)-g(b)\}\{g(a)-g(c)\}}a + \frac{\{y-g(c)\}\{y-g(a)\}}{\{g(b)-g(c)\}\{g(b)-g(a)\}}b$ 06 else $z \leftarrow b - g(b) \frac{b-a}{g(b)-g(a)}$ 07 if $\{z \in [(3a+b)/4, b]\} \forall \dots$ 08 $\{(m = true) \wedge (\sqrt{f(z)} - \sqrt{f(b)} \geq \sqrt{f(b)} - \sqrt{f(c)} /2)\} \forall \dots$ 09 $\{(m = false) \wedge (\sqrt{f(z)} - \sqrt{f(b)} \geq \sqrt{f(c)} - \sqrt{f(d)} /2)\} \forall \dots$ 10 $\{(m = true) \wedge (\sqrt{f(b)} - \sqrt{f(c)} \geq 2\epsilon)\} \forall \dots$ 11 $\{(m = false) \wedge (\sqrt{f(c)} - \sqrt{f(d)} \geq 2\epsilon)\}$ then 12 $z \leftarrow (a+b)/2, m \leftarrow true$ 13 else $m \leftarrow false$ 14 $d \leftarrow c, c \leftarrow b$ 15 if $g(a)g(z) < 0$ then $b \leftarrow z$ 16 else $a \leftarrow z$ 17 if $ g(a) < g(b) $ then swap(a, b) 18 return b	

[0073]

[0074] 전술한 [그림 1]의 $[a, b]$ 를 산출하기 위한 알고리즘과 [그림 2]의 root-finding 알고리즘을 포함하여, (a) 내지 (d)단계를 수행하는 알고리즘의 실시예는 [그림 3]과 같다.

[0075] [그림 3]

입력: 로봇의 위치와 공분산 (r, R) 장애물의 위치와 공분산 (q, Q) 충돌확률밀도 역치 ρ 허용 거리오차 ϵ
출력: 위치불확실성을 고려한 거리 $d_v(s, S^2; \rho)$
<pre> 01 $(V, S^2) \leftarrow eigenDecomposition(Q + R)$ 02 $s \leftarrow V^T(r - q)$ 03 if $\max_i \sigma_i = 0$ then return $\ s\$ 04 if $d_M^2(0, S^2; s) \leq -2 \ln(\rho \sqrt{(2\pi)^n S^2 })$ then return 0 05 $[a, b] \leftarrow initializeInterval(s, S^2, \rho)$ 06 $\lambda \leftarrow rootFinding(s, S^2, \rho, \epsilon, a, b)$ 07 return $\sqrt{f(\lambda)}$ </pre>

[0076]

[0077] 도 3은 상기의 알고리즘을 컴퓨터 프로세서 상에서 수행하여 본 발명의 실시예에 따른 이동로봇의 장애물 회피를 위한 거리를 결정한 시뮬레이션 결과를 나타낸다. 도 3의 (a)에서 이동로봇의 위치는

$$N\left(\begin{bmatrix} 0.20 \\ 0.25 \\ -0.05 \end{bmatrix}, \begin{bmatrix} 0.004 & 0.002 & 0.001 \\ 0.002 & 0.002 & -0.001 \\ 0.001 & -0.001 & 0.003 \end{bmatrix}\right) \text{ 이고, 장애물의 위치는 } N\left(\begin{bmatrix} -0.07 \\ -1.00 \\ 1.00 \end{bmatrix}, \begin{bmatrix} 0.020 & -0.010 & -0.005 \\ -0.010 & 0.010 & 0.007 \\ -0.005 & 0.007 & 0.015 \end{bmatrix}\right) \text{ 이다.}$$

이들의 확률분포는 도 3의 (a)에서 푸른 영역(이동로봇의 영역) 또는 붉은 영역(장애물의 영역)으로

표현되었다. 도 3의 (b)에서 이동로봇의 상대적 위치는 $\begin{bmatrix} -0.4477 \\ 0.0691 \\ -0.1127 \end{bmatrix}$ 이고, 푸른색 't' 로 표현되었으며, 장애물

의 영역은 $E\left(\begin{bmatrix} 0.0816 & 0 & 0 \\ 0 & 0.1274 & 0 \\ 0 & 0 & 0.1764 \end{bmatrix}; 1\right)$ 이고 타원체로 표현되었다. 도 3의 (c)에서 [그림 1]의 [a, b]를 산출하기 위한 알고리즘을 사용하여 연산 범위(searching interval)를 표현하였으며, 도 3의 (d)에서 도 3(c)의 연산 범위 내의 조건으로 [그림 2]의 root-finding algorithm을 사용하여 라그랑주 승수를 산출한 것을 표현하였다.

[0078] 본 알고리즘을 적용하여 이동로봇의 장애물 회피를 위한 거리결정 방법을 실험하였으며, 그 결과는 도 4 및 도 5와 같다. 도 4는 실험을 위한 무작위로 장애물이 배치된 상황의 시작점 환경과 이동로봇을 컴퓨터 상에 맵핑한 디스플레이 화면을 나타낸다. 도 5는 도 4의 실험환경에서 장애물의 각 영역을 설정하여 이를 회피하는 이동로봇 경로지점 및 목표지점까지의 이동결과를 표시한 실험 결과를 나타낸다. 도 5를 참조하면, 장애물의 영역에서 위치의 불확실성을 타원체로 표현되며, 장애물과의 가장 근접한 지점을 판단하여 장애물을 회피하는 경로가 설정됨을 확인할 수 있다.

[0079] 이상에서 대표적인 실시예를 통하여 본 발명을 상세하게 설명하였으나, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 상술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리 범위는 설명한 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 특허청구범위와 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태에 의하여 정해져야 한다.

부호의 설명

[0081]

S10: 공분산을 산출하는 (a)단계

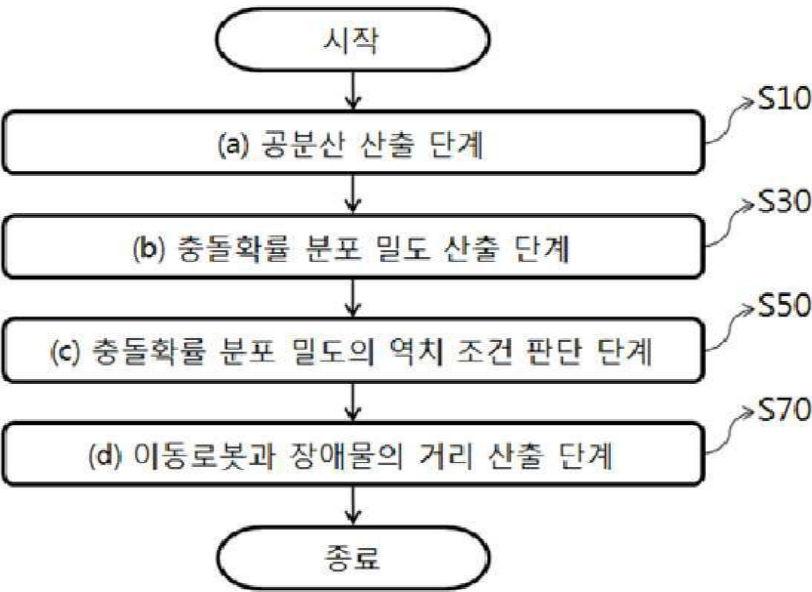
S30: 충돌확률 분포 밀도를 산출하는 (b)단계

S50: 충돌확률 분포 밀도의 역치 조건을 판단하는 (c)단계

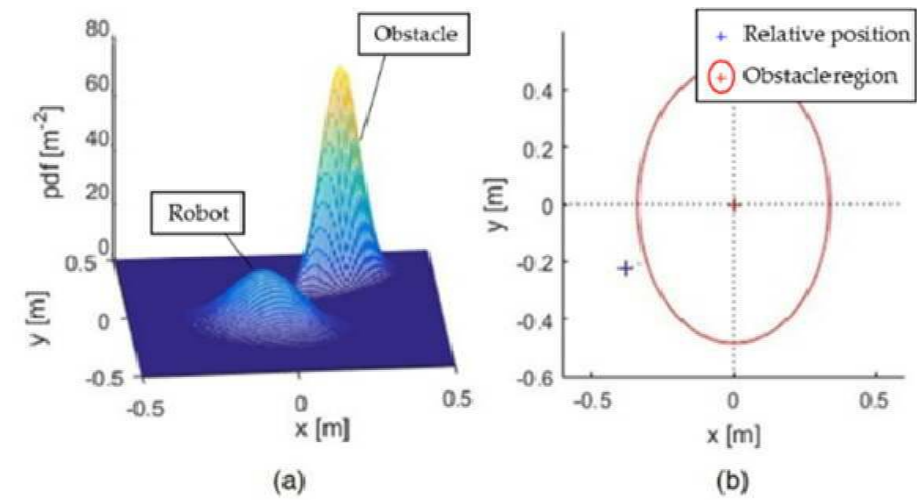
S70: 이동로봇과 장애물의 거리를 산출하는 (d)단계

도면

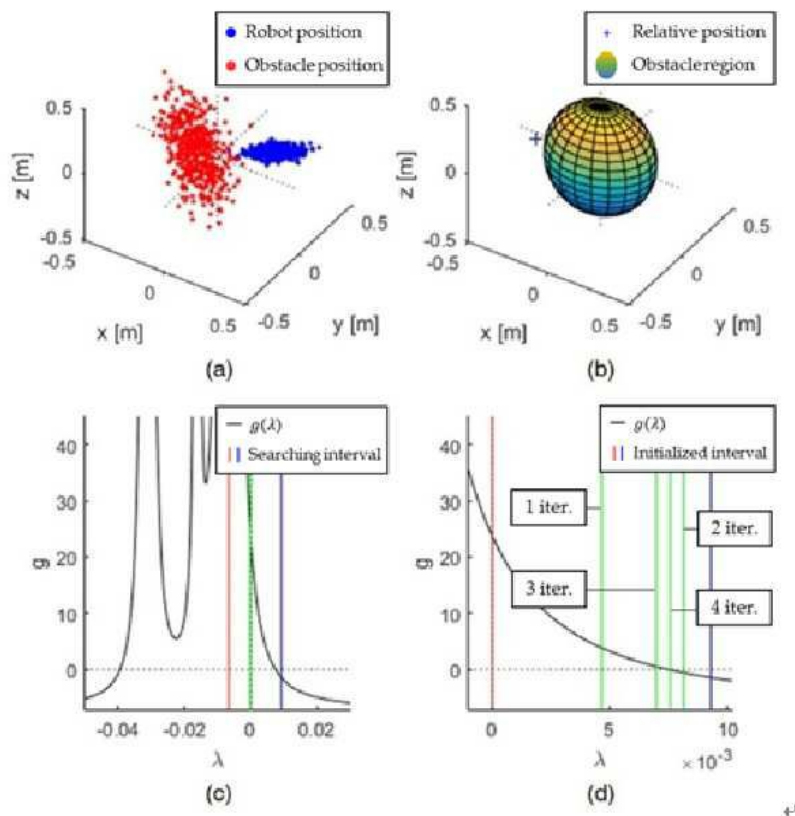
도면1



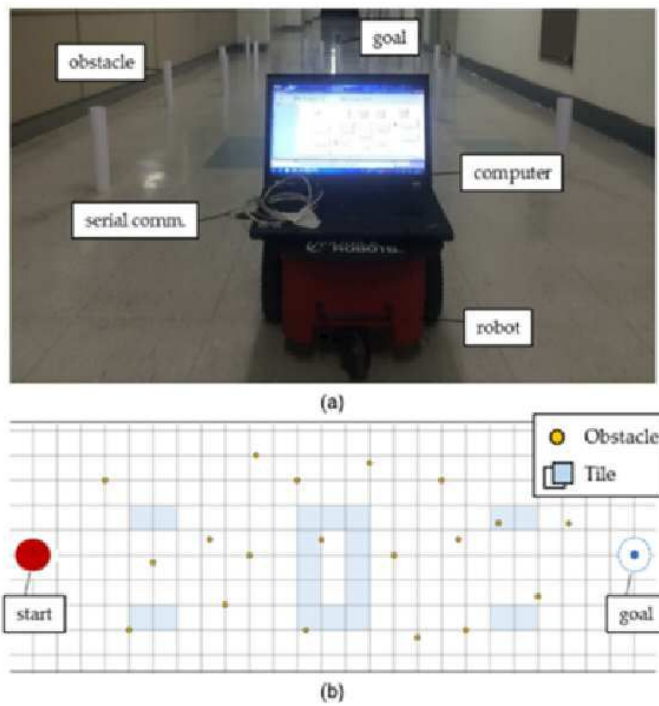
도면2



도면3



도면4



도면5

