

유전 알고리즘을 이용한 이동로봇의 주행 오차 보정

Correction of Traveling Error for a Mobile Robot Using a Genetic Algorithm

박병규, 이기성

홍익대학교 공과대학 전기제어공학과

Byunggyu Park, Keeseong Lee

요 약

일반적으로 직진 경로를 주행하는 이동로봇의 오차 보정을 위해서는 PI 제어기의 계수 보정이 필요하다. 본 논문에서는 직진 경로를 주행하는 이동로봇의 경우 양쪽 부하의 불균형에 의해서 생기는 진행 방향과 움직임의 오차를 이동로봇의 양쪽 바퀴에서 얻어진 엔코더의 값으로 측정하고, 측정되어진 엔코더의 값을 이용하여 방향과 움직임의 오차를 보정해주는 알고리즘을 PI제어기와 유전알고리즘을 사용하여 최적의 파라미터를 구할 수 있는 방법을 제안하였다.

I. 서 론

이동로봇이 주행을 할 때 두바퀴의 속도를 같은 속도로 움직이더라도 이동로봇 양쪽 바퀴에 들어가는 부하가 다를 경우 경로를 이탈하는 경우가 생긴다. 이럴 경우 이동로봇은 원하는 경로가 아닌 다른 경로로 이탈하기 쉽다. 본 논문은 이러한 이동로봇의 경로이탈방지를 위한 제어기로는 PI제어기를 사용하였으며, PI제어기의 계수의 최적해를 구하기위해서 유전 알고리즘을 이용하였다.

II. 본 론

1. 이론적 배경

1.1 주행 보정 제어기

1.1.1 주행기록

이동로봇의 거리에 대한 정보는 절대위치와 상대위치를 통하여 알 수 있다. 그 중 로봇의 주행한 거리 즉, 상대위치는 이동로봇의 바퀴에서 얻어진 엔코더의 펄스값으로 알아낼 수 있다.

주행기록을 나타내는 식은 (1)과 같다.

$$c_m = \frac{\pi D_n}{nC_e} \quad (1)$$

식(1)에서, c_m = 거리 변환 인자 이며,

D_n : 바퀴의 지름 이며,

C_e : 엔코더 분해능 이며,

n : 기어비 이다.

한편, 주행거리 변환식은 식 (2)와 같다[3].

$$\Delta U_i = c_m N_i \quad (2)$$

여기서, ΔU_i 는 이동한 거리의 변화량이며,

N_i 는 엔코더에서 받아들인 펄스의 수이다.

1.2 주행 알고리즘

고정된 절대 좌표에서의 이동로봇의 위치의 표현은 3개의 값들로 가능하다. 이동로봇의 중심 좌표 x , y 와 방향 θ 이다. 그림 1은 초기의 위치 (x_0, y_0, θ_0) 에서 새로운 위치 (x_f, y_f, θ_f) 로 이동을 하였을 때 진행을 나타낸다[1,2]. 식 (3)과 (4)는 이동로봇의 초기위치에서 새로운 위치로 이동하였을 때 변화된 각과 거리를 나타낸다.

$$\phi = \arctan \frac{y_f - y_0}{x_f - x_0} \quad (3)$$

$$d = \sqrt{(x_f - x_0)^2 + (y_f - y_0)^2} \quad (4)$$

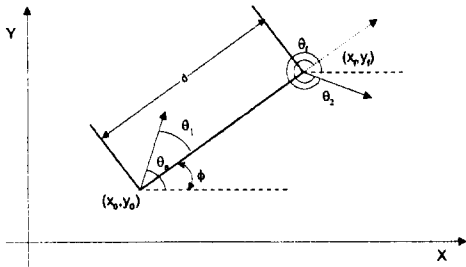


그림 1. 새로운 위치로의 경로 진행

1.1.3 PI-제어기

이동 로봇의 두 개의 엔코더에서 받은 펄스의 값들로 오차신호 E_i 를 발생시킨다.

$$E_i = N_{Li} - N_{Ri} \quad (5)$$

여기서, N_{Ri} = 오른쪽 엔코더의 펄스 수 이며,

N_{Li} = 왼쪽 엔코더의 펄스 수 이다.

오차신호 E_i 가 0가 아니라는 것은, 한쪽의 모터가 다른 한쪽의 모터보다 빠르다는 것을 나타내고, E_i 는 양 또는 음의 값을 갖는다. 식 (6)처럼 오차신호 E_i 는 보정속도 M_i 로 다시 정의되어진다. 이것이 P(Proportional)-제어기이다.

$$M_i = K_p E_i \quad (6)$$

여기서, K_p 는 비례상수 이다.

식 (7), (8)은 양쪽의 모터의 변화된 속도이다.

$$v_L = v_{ref} - M_i \quad (7)$$

$$v_R = v_{ref} + M_i \quad (8)$$

여기서, v_R = 오른쪽 모터의 속도 이며,

v_L = 왼쪽 모터의 속도 이며,

v_{ref} = 처음 출발때의 속도 이다.

이런 방법으로, P-제어기는 두 모터들의 속도를 같아지도록 하여 보정해 준다. 그러나, 이동로봇의 양쪽 바퀴 부하가 다르면, 위의 P-제어기로 구성된 이동로봇은 경로를 다시 이탈할 수 있다. 이러한 문제를 보완하기 위해 P-제어기에 적분기(Integration)를 덧붙혔다. PI-제어기는 이동 로봇의 속도가 아닌 출발점에서부터 이동한 만큼의 전체 펄스수를 같게 해주는데 목적을 둔다.

식 (9)는 PI-제어기를 나타낸다.

$$M_i = K_c \sum_{n=0}^i E_n + K_p E_i \quad (9)$$

여기서 K_c 는 적분 상수

이와 같은 PI-제어기는 단지 작은 계산량만을 요구하고, 구성하기가 쉽다[1,2].

그림 2는 이동로봇의 두 바퀴간 부하가 불균등할 때의 P-제어기 보정효과(그림 2a)와 PI-제어기 보정효과(그림 2b)를 나타낸 그림이다. 여기서 P-제어기로 보정하였을 때는 주어진 경로를 따라가다 경로를 이탈하게되고, PI-제어기로 보정하였을 때는 경로를 이탈한 경우에도 다시 경로를 찾아간다.

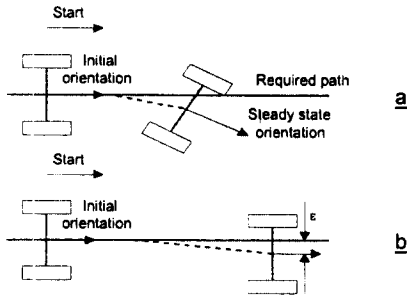


그림 2. 이동로봇의 경로 보정효과
(a. P-제어기 b. PI-제어기)

본 논문에서 구성한 유전알고리즘의 시스템 구조는 그림 3과 같다.

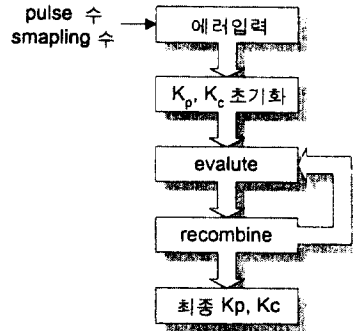


그림 3. 전체 시스템의 구조

1.2 유전 알고리즘

유전 알고리즘은 미지의 함수 $Y=f(x)$ 를 최적화하는 해 x 를 찾는 모의 진화(Simulated Evolution)형의 탐색 알고리즘이다. 유전 알고리즘에서는 생물학적 유전인자인 염색체에 해당하는 문자열을 가지고 생물과 같은 재생산(Reproduction), 교배(Crossover), 돌연변이(Mutation)를 거쳐서 다음 세대의 새로운 자손(Offspring)을 인공적으로 만들어 낸다.

다음은 본논문에서 사용한 SSGA의 알고리즘이다.

- step 1. 초기 개체군(population)을 초기화한다.
- step 2. 개체군 내에서 적합도가 가장 높은 개체 두 개를 선택한다.
- step 3. 선택한 두부모개체에 대해서 균일교배를 수행하여 자식 개체를 생성한다.
- step 4. 생성된 자식개체에 돌연변이 연산을 수행한다.
- step 5. 개체군 내에서 가장 낮은 적합도를 가진 개체 두개를 개체군에서 삭제한다.
- step 6. 생성된 자식개체를 개체군에 삽입한다.
- step 7. 정해진 세대수만큼 또는 목적하는 값이 될 때까지 2~6을 반복한다.

2. PI 계수 선택 메카니즘

2.1 전체 시스템의 개요

2.2 주행 오차 매개변수

본 논문에서는 유전 알고리즘을 이용하여 K_p, K_c 를 찾아내는 것이다. 한세대 개체군은 두 상수의 그림 3과 같이 나열한 모양으로 임의의 K_p, K_c 를 발생 시키도록 하였다.

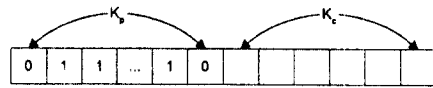


그림 4. 유전자형

2.3 평가 함수

평가 함수의 선택은 유전 알고리즘의 성능을 좌우하게 된다. 본 논문에서는 정해진 도착점의 위치와 실제 도착한 이동로봇의 위치의 선정된 평가 함수는 다음과 같다.

$$Fitness = \frac{1}{1 + E^2} \quad (10)$$

4. 실험

실험환경은 HIVD(Hong-Ik Vehicle Dog) 이동로봇을 사용하였다. 주행속도는 평균 0.05

m/s, 엔코더에서의 샘플링 시간은 50ms로 하였다. 이동로봇의 시작좌표의 위치와 각도는 (0,100cm, 0°)으로 하여 기준점으로 잡았고 목표 도착점의 위치와 각도는 (100cm, 100cm, 0°)로 x축으로 1m 직진 주행을 하도록 하였다. 표 1, 표2와 그림4는 주행 보정전,후의 이동로봇이 이동하였을 때 좌표를 보여준다. 그림 4에서의 좌표는 목표 도착점에서의 실제 도착점을 상대 좌표로 나타낸 것이다.

좌 표 x (cm)	좌 표 y(cm)	각도(°)	시간(sec)
108	76.9	-26	16.80
112.6	71.6	-28.5	17.65
109.9	82.8	-22.5	16.75
107.4	73.8	-31	17.30
112.7	85.2	-16	16.95

표 1. 보정전 이동로봇의 위치 및 결과

좌 표 x (cm)	좌 표 y(cm)	각도(°)	시간(sec)
102.1	97	-4	15.15
103.6	94.7	-6.5	15.95
100.6	95.5	-9	15.25
102.6	96.3	-7	15.50
104.3	98	-3	15.35

표 2. 보정후 이동로봇의 위치 및 결과

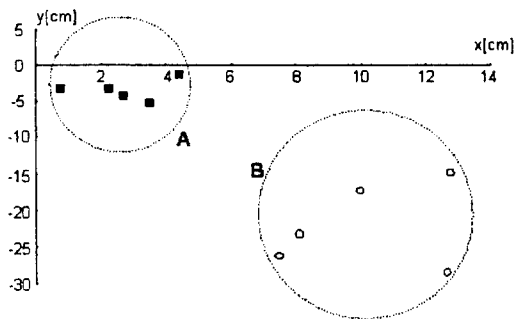


그림 5. 보정전, 후의 이동로봇의 도착위치 (A. 보정후, B. 보정전)

III. 결론

본 논문에서 보정전보다 보정후의 이동로

봇이 더 목표하는 곳에 더 가깝고, 정확하게 도착할 수 있음을 보여준다. 본 논문에서는 유전 알고리즘 중에서 SSGA를 사용하여 최적의 해를 찾는데 부모세대의 해를 생존시켜 그세대보다도 더 열등한 자식세대를 갖지 않도록 하여 수렴성을 보장할 수 있었고, 사용한 PI제어기와 유전 알고리즘으로 이동로봇의 경로의탈을 피할 수 있었다. 앞으로 이동로봇이 좀 더 정확한 위치와 방향으로 가기 위해서는 기구학적 특성, 엔코더의 분해능, 캐스터등의 영향도 고려 해야 한다고 본다.

참고 문헌

- [1] J. Borenstein and Y. Koren, "A Mobile Platform for Nursing Robots," *IEEE Trans. on Industrial Electronics*, Vol. 32, No. 2, pp. 158-165, 1985.
- [2] J. Borenstein and Y. Koren, "Motion Control Analysis of a Mobile Robot," *Transactions of ASME, J. of Dynamics, Measurement and Control*, Vol. 109, No. 2, pp. 73-79, 1985.
- [3] J. Borenstein and Y. Koren, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," *IEEE Trans. on Robotics and Automation* Vol. 12, No. 6, pp. 869-880, December 1996.
- [4] B. Soucek, *Dynamic, Genetic and Chaotic Programming*, John Wiley & Sons, 1991
- [5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1994.
- [6] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.