

PID 제어를 이용한 자율주행 로봇의 경로 추적 및 군집주행

김서연, 정현서, 정은주, 김은경

국립한밭대학교

{syeon, hyunseo, eunju}@edu.hanbat.ac.kr, ekim@hanbat.ac.kr

Path Tracking and Platooning of Autonomous Driving Robots Using PID Control

Seo Yeon Kim, Hyun Seo Jeong, Eun Ju Jeong, Eunkyung Kim
Hanbat National University

요 약

본 논문에서는 자율주행 로봇의 경로 추적과 군집 주행 시스템을 구현한다. 로봇이 트랙을 벗어나지 않고 주행하기 위하여 피드백 제어의 한 종류인 PID 제어를 적용한다. 초음파 센서를 이용하여 로봇 간 거리를 측정하고, 로봇 간 거리가 감소하면 후행 로봇의 속도를 감소시켜 거리를 유지하고, 안전거리 이하로 줄어드는 경우에는 후행 로봇이 정지하도록 설계하였다. 로봇 간 거리가 벌어지면 후행 로봇의 속도를 증가시켜 안전거리를 유지하면서 군집 주행을 가능하도록 하고 구현된 경로추적 시스템과 군집주행 알고리즘은 효과적으로 동작하는 것을 확인한다.

1. 서 론

4차 산업혁명 시대에 자율주행 기술은 주요 혁신 기술 중 하나이다[1]. 자율주행을 위해 PID 제어는 자율주행 로봇(또는 차량)이 위치와 속도를 정밀하게 조절하여 경로를 안정적으로 추적할 수 있게 한다. 특히 두 대 이상의 차량이 협력하여 군집을 형성하고 주행하는 군집주행에서는 차량 사이의 상호작용과 통신 그리고 차량 간의 적절한 거리 유지를 필요로 한다. 본 논문에서는 PID 제어를 이용한 자율주행 로봇의 경로 추적 및 군집주행에 대해 연구한다. PID 제어를 통해 로봇의 경로 추적 성능을 향상시키고 2대 이상의 로봇이 일정한 간격을 유지하며 효과적으로 군집주행을 수행하는 알고리즘을 제안한다. 또한 제안한 알고리즘의 동작을 실제 자율주행 로봇에 구현하여 확인한다.

2. PID 제어를 이용한 경로 추적 및 군집주행 시스템 설계

자율주행 로봇은 목표 경로를 따라가기 위하여 조향 조작이 필요하다. 조향 제어는 피드백 제어의 한 종류인 PID 제어 방식[2]을 사용한다. PID(Proportional Integral Derivative) 제어기는 P 제어(Proportional Control), I 제어(Integral Control) 및 D 제어(Derivative Control)를 이용한다. 각 제어를 통해 목표 값에 가까워지도록 하는 것이 PID 제어의 목적이다.

P 제어는 P gain(K_p)을 사용하여 목표값과 실제 측정값 사이의 차이인 오차에 비례하게 제어량을 만들어서 시스템의 출력이 목표 값에 가까워지도록 한다.

I 제어는 오차의 누적된 값을 이용하여 제어량을 조정하는 적분 제어기로 I gain(K_i)을 사용하여 오차가 누적되는 속도가 증가하고 이에 제어량이 더 강해진다. P 제어만으로 해결하기 어려운 정상상태 오차를 제거하고 목표값에 더욱 근접하게 도달할 수 있게 한다.

D 제어는 오차의 변화율에 비례하여 D gain(K_d)값을 이용하여 급격한 변화에 대한 반응을 제어하고 시스템의

반응 속도를 개선하고 오버슈트를 줄여 안정적으로 목표 값에 도달하도록 하는 미분 제어기다.

PID 제어기의 제어식은 다음과 같다[2].

$$\text{조작량} = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \quad (1)$$

각 PID 제어의 최적화된 값은 보다 세밀한 방식으로 결과를 도출하는 수동 튜닝을 우선적으로 적용한다.

경로 추적 시, [그림 1]처럼 8개의 센서가 일렬로 배열되어 있는 LSA(Light Sensor Array)센서를 사용하여 각 센서가 측정한 조도 값을 리스트 형태로 저장하고 활용한다. 이 센서 값을 활용하여 자율주행 로봇이 트랙 위의 흰색 선을 따라 이동이 가능하도록 한다.

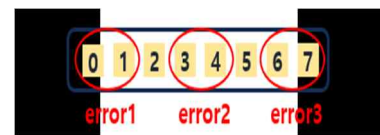


그림 1 error 값 계산 방법

로봇의 섬세한 조작을 위해 각 센서에서 얻은 값을 기반으로 오차값을 계산한다. 8개의 센서 값 중에서 왼쪽 끝 2개의 값의 평균을 이용한 오차값은 다음과 같다.

$$\text{error1} = th - (\text{sensorvalue}[0] + \text{sensorvalue}[1])/2 \quad (2)$$

여기서 th란 목표값을 나타내며, 이는 로봇이 센서를 통해 트랙에서 측정한 흰색 선의 값을 나타낸다. 중앙의 오차값(error2), 오른쪽의 오차값(error3)도 이와 같이 계산된다. 이 오차값들은 로봇이 트랙 위에서 벗어나지 않도록 조작량을 결정하는데 활용한다. 조작량은 PID 제어 식(1)에 따라 계산된다. 이 공식을 통해 각 오차값에 대한 3가지 조작량이 생성되며, 로봇의 움직임을 섬세하게 제어하여 트랙을 정확하게 주행할 수

있도록 한다.

```

Algorithm 1 조향 제어
1: correction1 - 왼쪽 앞 2개의 센서 값을 기반으로 계산된 조작량
2: correction2 - 중앙 2개의 센서 값을 기반으로 계산된 조작량
3: correction3 - 오른쪽 앞 2개의 센서 값을 기반으로 계산된 조작량
4:
5: procedure STERLINGCONTROL(correction1, correction2, correction3)
6:   if (sensurvalue[0] ≤ a) and ((sensurvalue[3] + sensurvalue[4]) ≥ b) and (sensurvalue[7] ≤ c) then
7:     correction1을 사용한 강한 우회전
8:   else if (sensurvalue[0] ≤ d) and (sensurvalue[6] ≥ e) and (sensurvalue[7] ≤ f) then
9:     correction2를 사용한 약한 우회전
10:  else if (sensurvalue[0] ≥ g) and ((sensurvalue[4] + sensurvalue[5]) ≥ h) and (sensurvalue[7] ≥ i) then
11:    correction3을 사용한 강한 좌회전
12:  else if (sensurvalue[3] ≥ j) and ((sensurvalue[4] + sensurvalue[5]) ≤ k) and (sensurvalue[7] ≥ l) then
13:    correction3을 사용한 약한 좌회전
14:  else if (sensurvalue[0] ≥ x) and ((sensurvalue[4] + sensurvalue[5]) ≥ y) and (sensurvalue[7] ≥ x) then
15:    조향 제어 없음-직진
16:  end if
17: end procedure
  
```

Algorithm1 은 3 가지 조작량의 사용 기준을 나타낸다. 왼쪽 끝의 0 번 센서, 중앙의 3 번 및 4 번 센서, 그리고 오른쪽 끝의 7 번 센서를 기준으로 정해진다. 로봇에 장착된 센서마다 측정 값이 다르기 때문에, 측정 후에 적절한 기준을 설정해야 한다.

```

Algorithm 2 군집 주행에서 로봇 간 거리 유지
1: Input: Distance between robots measured by ultrasonic sensors
2: Output: Speed control commands for the robot
3: procedure CONTROLDISTANCE(distance)
4:   if distance ≤ 35 then
5:     if distance ≤ 27 then
6:       Stop the robot
7:     else if 27 < distance ≤ 35 then
8:       Reduce speed
9:     end if
10:  else if distance > 35 then
11:    Increase speed
12:  end if
13: end procedure
  
```

Algorithm 2 는 초음파 센서를 이용해 로봇 간 거리를 유지하는 알고리즘이다. 이 알고리즘의 핵심은 로봇 간 거리 측정과 이에 따른 속도 조절이다. 먼저, 초음파 센서를 이용해 선행 로봇과의 거리를 실시간으로 측정한다(1 행). 이 거리가 설정된 최소 안전거리인 35cm 이하로 줄어들면 속도를 약 0.12m/s 로 감소시켜 거리를 유지하고, 27cm 이하로 줄어들 경우에는 정지하도록 설계하였다(4-8 행). 로봇 간 거리가 35cm 이상으로 벌어지면 후행 로봇의 속도를 약 0.17m/s 로 증가시켜 35cm 의 거리를 유지하면서 군집 주행이 가능하도록 설계하였다(10-11 행).

3. 실험 방법 및 결과



그림 2 선행로봇(좌)과 후행로봇(우)

앞서 설계한 자율주행 및 군집주행 시스템을 통합하여 검증하기 위해 실험 트랙에서 구동을 진행하였다. [그림 2]는 본 연구에 사용한 선행 로봇과 후행 로봇이다. 후행 로봇은 선행 로봇과의 거리를 측정하기 위해 초음파 센서가 앞쪽에 장착되어 있다. 라인을 추적하기 위해 이용하는 LSA 센서도 앞쪽 하부에 장착되어 있다.

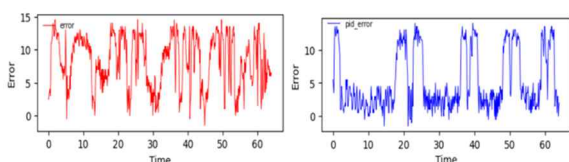


그림 3 PID 제어 전(좌)과 제어 후(우)

[그림 3]은 선행 로봇이 경로 추적을 할 때 PID 제어 적용 전과 후를 비교한 데이터를 보여준다. x 축은 로봇의 트랙 2 바퀴 완주 시간을 나타낸 것이고 y 축은 PID 제어를 이용한 경로 추적 및 군집주행 시스템 설계에서의 error2 를 나타낸다. 그래프에서 볼 수 있듯이, PID 를 적용하지 않는 경우 일정한 패턴 없이 error 값이 급격하게 변동하며 불안정적인 주행을 보여준다. 이는 직선 구간과 코너 구간에서 흰색 선을 정확하게 추적하지 못 하고 있다는 것을 보여준다. 또한, 로봇이 트랙을 벗어나는 문제가 발생하여 벗어난 로봇을 트랙으로 옮겨주는 작업이 필요했다. 따라서 PID 제어 전 주행에서는 시간이 더 소요되며 비효율적이라는 것을 알 수 있다. PID 를 적용하는 경우 error 값이 비교적 일정한 패턴으로 유지되며, 직진 구간에서 급격한 변동 없이 흰색 선을 잘 따라가고 있는 것을 보인다. 코너 구간에서 흰색 선을 잠시 벗어났지만 곧바로 다시 흰색 선으로 돌아와 안정적으로 주행하는 것을 보여준다. 이는 PID 제어가 효과적으로 작동하고 있음을 나타낸다.

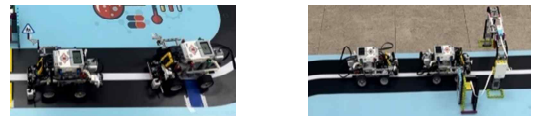


그림 4 주행 중 거리 유지(좌)와 정지 시 안전거리 확보(우)

[그림 4]는 실험에서 두 로봇이 동시에 출발한 후 안전거리를 확보하여 주행 및 정지하는 모습이다. 특히 선행 로봇이 신호등 앞에서 멈추는 경우에도 후행 로봇이 충돌하지 않고 멈추는 것을 확인하였다. 이를 통해 시스템의 자율주행과 군집주행 기능이 실험 환경에서도 성공적으로 동작함을 확인하였다(추가 결과 [3] 참조).

4. 결론

본 논문에서는 자율주행 로봇의 경로 추적과 군집주행 시스템을 구현하였다. 로봇이 트랙을 벗어나지 않고 원활한 주행을 위해 피드백 제어의 한 종류인 PID 제어를 적용하였으며 또한, 초음파 센서를 이용하여 선행로봇과 후행로봇 간의 거리를 측정하고 일정한 거리를 유지하는 군집주행 알고리즘을 설계하였다.

향후 연구에서는 LiDAR, RADAR, GPS 등의 추가 센서 기술을 도입하여 자율주행 로봇의 주변 환경 인식 능력과 군집 주행 능력을 향상시키는데 중점을 두어 고도화된 자율 주행 시스템에 대해 연구할 예정이다.

ACKNOWLEDGMENT

본 연구는 2024 년 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업의 연구결과 (2022-0-01068)와 대학 ICT 연구센터사업의 연구결과(IITP-2024-RS-2024-00437886)로 수행되었음.

참 고 문 헌

- [1] 김현규, 허진수. "자율주행 기술 연구 동향 및 전망." The Journal of The Korean Institute of Communication Sciences 35, no. 5 (2018): 3-12.
- [2] 정세영. "A study on steering position control system of mobile robot using PID self-tuning," Doctoral Dissertation, 목원대, 2017.
- [3] "Path Tracking and Platooning of Autonomous Robots Using PID Control," <https://github.com/iSWLab/Path-Tracking-and-Platooning-of-Autonomous-Driving-Robots-Using-PID-Control>