

미로 내 목표물 이동을 위한 군집로봇 효율성에 관한 연구

이병준*, 우균*, 정성엽**, 전준기**, 김형일**, 김한빛**, 김민국**
 부산대학교*, 한국과학영재학교**
 imjuni@pusan.ac.kr

A Study on the Efficiency of Swarm-bot for Carrying Objects in a Maze

Sung-Youb Jung*, Joon-Key Jeon*, Hyeong-Il Kim*, Han-Bit Kim*
 *Korea Science Academy

요 약

군집로봇이란 하나의 일을 단순한 작업을 하는 여러 대의 로봇을 이용하여 수행하는 로봇들을 일컫는 말이다. 본 연구에서는 군집로봇을 이용하여 미로를 탐색하고 스스로 작업을 분할하여 목표물을 운반하는 로봇 제어프로그램을 작성했다. 또한 작성된 제어 프로그램으로 로봇 수와 작업 목표물 개수에 따른 군집로봇의 효과에 대하여 실험하였다. 작성된 제어 프로그램을 시뮬레이션 환경에 적용하여 실험한 결과 군집로봇을 사용하면 작업시간을 단축시킬 수 있었다. 또한 작업을 효과적으로 수행하기 위해서 로봇 수를 늘리는 것은 일정 한계까지 작업 효과가 증가하며 그 이후에는 오히려 작업 효과를 저해함을 알 수 있었다.

1. 서론

로봇의 발명으로, 인간이 하기 힘든 작업을 로봇이 대체하기 시작했다. 이런 로봇 중 외부 환경을 인식하고 자기 판단에 의해서 행동하는 로봇을 지능형 로봇이라고 한다^[1]. 이중 군집로봇은 여러 로봇이 각각의 일을 분담하여 처리하는 로봇들을 말한다. 군집로봇들은 복잡한 단일 로봇에 비해 기능이 떨어지지 않, 협력하여 작업을 효율적으로 처리할 수 있고, 단일 로봇이 불가능한 일을 진행할 수도 있다. 또한 결함이 생겼을 때 군집로봇은 그 로봇을 제외한 다른 로봇들로 작업을 완수할 수도 있다.

일반적으로 병렬 프로그래밍 환경에서는 CPU 숫자를 증가시키에 따라 속도가 선형적으로 증가하지 않는다. Amdahl의 법칙에 따르면, 속도증가를 저해하는 여러 요인으로 인하여 그림 1 과 같은 속도효율을 보인다.

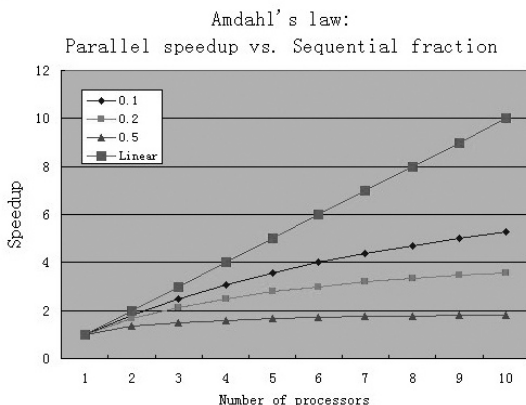


그림 1 Amdahl 법칙에 따른 속도증가

군집로봇 또한 로봇 수를 증가 시킴에 따라 작업효율이 선형적으로 증가하지 않을 것이다.뿐만 아니라 군집로봇은 통신시간과 군집행위를 위한 대기시간과 같은 다양한 작업효율 저하요인으로 인하여 CPU와는 달리 속도가 더 느려질 우려가 있다. 따라서 작업량에 따른 적합한 로봇 수를 찾아낼 필요성이 있다.

본 연구는 먼저 Pyro 시뮬레이션 환경을 사용하여 군집지능 제어프로그램을 작성하고 시뮬레이션 한다. Pyro 시뮬레이션 환경을 이용하여 군집로봇이 가지는 효과에 대하여 검증하고자 한다. 따라서 미로를 제작하고, 텔레이 방식의 군집지능을 설계한 후, 작업량 및 로봇 수에 따른 작업 효율을 살펴본다.

2. 연구환경

2.1 Pyro

Python Robotics의 약자로, Python 언어로 제작된 로봇 개발 플랫폼이다. Pyro는 Bryn Mawr 대학에서 커리큘럼을 개설하여 교육한 적이 있는, 교육적 목적과 실제 로봇에 적용하여 개발 가능한 개발 목적에 동시에 부합하는 로봇 개발 플랫폼이다^[3]. 로봇 하드웨어에 대한 세부적인 지식이 없더라도 프로그램을 작성하고 인공지능을 테스트할 수 있다.

2.2 플레이어 / 스테이지(Player / Stage)

Stage는 Pyro에서 지원하는 시뮬레이터로, 2차원 다중 로봇 시뮬레이터이다. 플레이어는 일종의 미들웨어로, Pyro와 Stage를 연결하는 역할을 한다. 또한 로봇에 장착된 각종 장치를 통제하는 역할을 겸하고 있다. 초음파 및 레이저 센서, 카메라와 같은 장치를 지원한다.

2.3 VMWare

리눅스환경에서만 구동되는 Stage 시뮬레이터를 사용하기 위하여 사용한 프로그램이다. 운영체제를 새로 설치하지 않고도 윈도우에서 리눅스를 구동할 수 있도록 해준다. 본 연구에서는 Debian 리눅스를 사용하였다.

3. 목표물 이동 알고리즘

우리가 가정한 시나리오는 단순한 미로 상황에서, 출발점에서 목적지까지 도착하여 목표물을 가지고 돌아오는 것이다. 본 연구에서는 리더 로봇이 먼저 미로를 탐색하고, 이 후 최단경로를 다른 로봇들에게 알린다. 최단경로를 일정하게 나누어 로봇을 배치하고, 각자에게 주어진 구역에서 릴레이 방식으로 물건을 운반하게 된다. 전체적인 과정은 다음과 같다.

- ① 리더 로봇이 벽을 따라 미로를 탐색
- ② 미로를 탐색하면서 최단경로 생성
- ③ 도착 후 다른 로봇에게 최단경로 전달
- ④ 리더로봇이 물건을 찾아 짐을
- ⑤ 나머지 로봇들이 맡은 구역으로 이동
- ⑥ 각자 주어진 구역에서 릴레이 형식으로 물건을 주고받아, 목적지에서 출발지로 운반한다.

3.1 벽 따라가기

하나의 벽으로 이루어진 가장 단순한 미로를 생각하여, 왼쪽 벽을 짚고 가면서 미로를 탈출하는 방법을 사용하였다.

3.2 물건집기

Pyro 에서 지원하는 카메라 장치와 그립퍼 장치를 이용한다. 먼저 카메라에 물체가 보일 때까지 회전한다. 목표물을 발견하면 카메라에서 영상처리 기법을 이용하여 가장 가까운 목표물을 계산해 낸다. 회전과 전진을 이용하여 목표물에 접근하여, 그립퍼 장치로 잡을 수 있을 때 그립퍼 장치를 작동시켜 잡는다. 잡지 못하면 그립퍼 장치를 열고, 뒤로 이동한 뒤 다시 찾는다. 마지막으로 처음 물건을 찾기 시작했던 지점으로 돌아간다.

3.3 최단경로 구성

벽 따라가기 알고리즘을 실행하면서, 최단경로를 계산하는 데 필요한 알고리즘이다. 가장 제한적인 격자형 미로를 가정하였으므로, GPS 장치를 통해서 로봇이 어느 위치에 있는지 알 수 있다. 이동하는 방향에 대한 정보를 채워가면서, 막힌 길을 이동하였을 경우에 백트래킹기법을 사용하여 최단경로를 찾는다.

3.4 최단경로 이동 알고리즘

최단경로 구성 알고리즘을 통해서 얻은 방향이 적힌 최단경로로 이동하는 알고리즘이다. 먼저 같은 방향으로 저장되어 있는 정보들을 종합하여, 현재 위치에서 직선거리로 얼마만큼 연결되어있는지 찾는다.

목적지의 좌표를 알게 되면, 현재 위치와 x 축으로 얼마나 떨어져 있는지, y 축으로 얼마나 떨어져 있는지 알 수 있다. 현재 로봇의 각도를 GPS 장치를 통해 알 수 있다. 두 벡터를 외적하여, 양수가 나오면 좌측으로, 음수가 나오면 우측으로 회전한다. 외적 값이 0에 가까워 지면 목표지점까지의 거리에 비례하는 속도로 신속하게 목표지점으로 이동한다.

3.5 릴레이 방식을 이용한 목표물 이동

앞서 가정한 시나리오와 같이, 탐색 목적으로 리더 로봇을 하나 보낸다. 리더로봇은 벽 따라가기 알고리즘을 통해 미로를 탐색하고 최단경로를 얻은 후 다른 로봇과 작업을 시작한다.

물건을 운반하는 업무를 분담하는 방법은 두 가지를 생각할 수 있다.

- 모든 로봇이 왕복하면서 일을 하는 방법
- 구간을 나누어 서로 바통을 전달하는 식으로 릴레이를 하는 방법

본 연구에서는 두 번째 방법을 채택하였다.

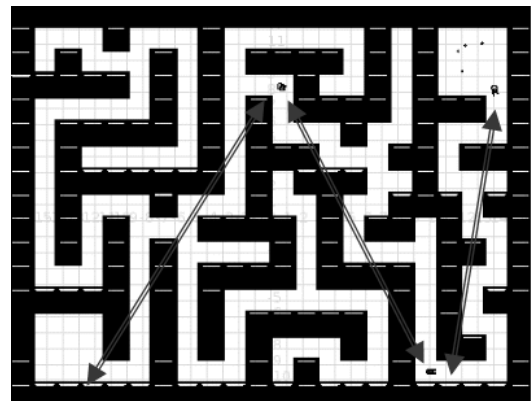


그림 2 로봇 3대의 릴레이 방법

4. 실험 및 연구결과

4.1 실험순서

본 연구에서는 군집 로봇을 사용할 때 로봇 개수와 목표물 개수에 따른 변화를 실험하였다. 이는 군집 로봇의 효과를 살펴보고, 작업에 필요한 로봇 개수를 파악하여 효과적인 군집 제어 프로그램 작성을 가능하게 해준다. 실험은 목표물 수와 시간과 관계, 로봇 수와 시간관계, 목표물과 로봇 수에 따른 효과를 실험하였다.

4.2 실험 1 목표물 수와 소요되는 시간의 관계

로봇 수가 일정할 때, 목표물 수에 따른 소요시간의 상관관계를 찾기 위한 실험이다. 로봇의 수는 4 개로 고정하였고, 목표물의 수를 1 개에서 7 개까지 증가시키면서 실험했다. 결과는 다음과 같다.

표 1 목표물 수와 소요시간의 관계

목표물	1	2	3	4	5	6	7
시간(분)	2.28	3.28	4.20	5.25	6.27	7.25	8.25

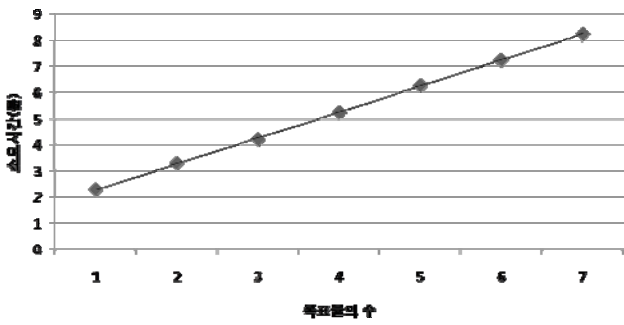


그림 3 목표물의 수와 소요시간의 관계 그래프

실험 결과로 보았을 때, 로봇의 수가 일정하면 목표물의 수가 증가함에 따라서 소요되는 시간이 선형적으로 증가한다는 것을 알 수 있다.

4.1 실험 2 로봇 수와 소요되는 시간의 관계

목표물의 수가 일정할 때, 로봇의 수에 따른 시간의 상관관계를 찾기 위한 실험이다. 이 실험에서는 목표물의 수를 4 개로 고정하였으며, 로봇의 수를 1 개에서 7 개로 늘려가며 실험하였다. 결과는 다음과 같다.

표 2 로봇의 수와 소요시간의 관계

로봇수	1	2	3	4	5	6	7
시간(분)	2.28	3.28	4.20	5.25	6.27	7.25	8.25

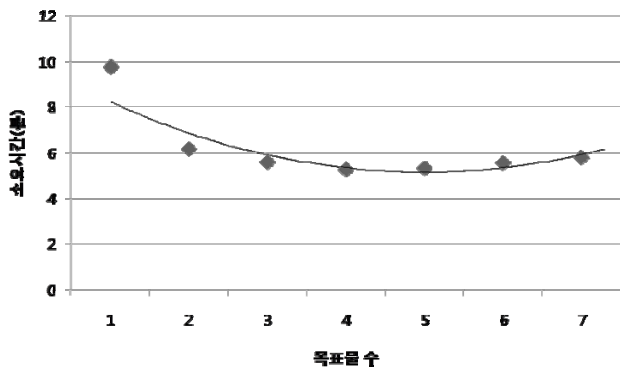


그림 4 로봇의 수와 소요시간의 관계 그래프

위 결과를 볼 때, 로봇의 수가 증가한다고 항상 좋지는 않다는 것을 알 수 있다. 특히 목표물의 수가 4 개일 때는 로봇의 수가 4 개일 때 가장 효율이 좋다는 것을 알 수 있다.

4.2 실험 3 목표물과 로봇 수에 따른 효율성

이번 실험에서는 로봇의 수와 목표물의 수를 모두 변화시켜가며 실험하였다. 결과는 다음과 같다.

표 3 목표물과 로봇 수에 따른 효율성

목표물의 수

	1	2	3	4	5	6	7
1	1.55	4.32	7.07	9.77	12.6	15.3	18
2	1.58	3.13	4.65	6.18	7.68	9.23	10.8
3	1.75	2.98	4.25	5.58	6.9	8.25	9.52
4	2.28	3.28	4.2	5.25	6.27	7.25	8.25
5	2.85	3.67	4.51	5.31	6.13	6.97	7.79
6	3.42	4.12	4.84	5.55	6.23	6.92	7.62
7	3.99	4.59	5.2	5.79	6.37	6.97	7.58

로봇의 수

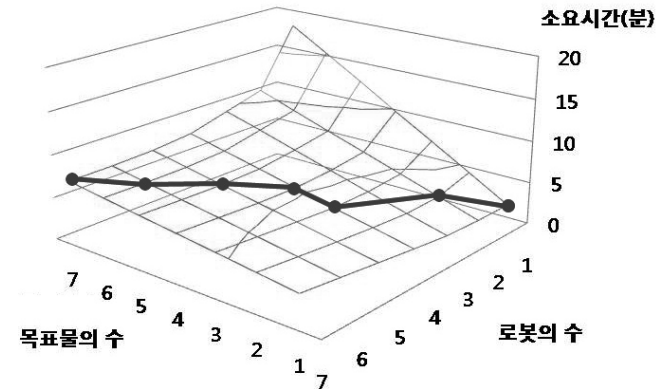


그림 5 목표물과 로봇 수에 따른 효율성 그래프

위의 표와 그래프를 보았을 때, 목표물의 수가 높아질 수록 효율이 가장 높은 로봇의 수가 증가하는 것을 볼 수 있다. 이때 로봇의 수가 너무 많아지게 되면 걸리는 시간이 늘어난다는 것을 확인할 수 있다.

5. 결론

본 연구에서는 군집 지능 로봇 제어 프로그램을 개발하고, 군집 지능의 효과에 대해서 검증하였다.

군집 지능 로봇 제어 프로그램은 리더 로봇을 이용하여 미로를 통과할 수 있으며 최단 경로를 도출할 수 있었다. 뿐만 아니라 릴레이 형태로 작업을 수행함에 따라 작업시간을 단축시킬 수 있었다.

또한 군집 지능 로봇 제어 프로그램이 가지는 효과에 대한 검증을 진행하였다. 군집 지능은 다수 개의 로봇을 사용하는 만큼, 다수 개의 로봇이 군집 지능을 수행하기 위해서 준비 시간과 의사소통 시간을 필요로 하므로 최적화된 로봇의 수를 찾는 것은 효과적인 군집 지능 제어를 위해서 필요하다.

본 연구에서는 목표물 개수와 로봇 개수에 따른 시간 변화를 살펴보고 두 개의 변위 요소를 모두 사용하여 실험을 진행하여 시간 변화량을 살펴보았다. 실험 결과에서는 로봇 개수를 선형적으로 증가시키면 따라서 작업 효율이 선형적으로 증가하지 않았다. 그 이유는 다음과 같이 추론할 수 있다.

- 처음 일을 시작할 때 걸리는 지연시간
- 물건을 주고받을 때 걸리는 지연시간
- 상호 의사소통을 통한 지연시간

실험 전에 예측한 것과 같이 작업 준비에 따른 지연시간, 물건을 주고받을 때 걸리는 지연시간 등으로 인하여 선형적인 증가를 하지 않았다. 또한 본 연구에서는 의사소통에 따른 지연시간이 없었지만 실제로 통신 장비를 사용한 군집 지능 제어 프로그램에서는 의사소통을 위한 지연시간이 발생할 수 있으며 선형적인 작업효율 증가를 저해할 수 있다. 만약 이러한 지연시간을 유발하는 요소가 없는 이상적인 환경이라면 로봇을 많이 투입하는 것이 효과적이지만, 실제로는 지나치게 작업효율을 저하시킬 수 있다는 것을 알 수 있었다.

6. 향후연구

본 연구는 앞으로 크게 2 방향으로 생각해 볼 수 있다. 첫 번째는 소프트웨어적인 효율 개선이다. 소프트웨어 적인 효율 개선은 다양하게 생각해 볼 수 있다. 현재 미로 탐색 알고리즘은 복잡한 미로를 통과하기 힘들기 때문에 미로 탐색 알고리즘을 개선할 것이다. 또한 릴레이 알고리즘을 개선할 수 있다. 직진 주행을 할 경우 현재 좀 더 높은 속도로 움직이도록 설계 되어 있다. 이 경우 릴레이 구간을 세분하여 효율적인 방식으로 작업을 진행하도록 개선할 수 있다. 두 번째는 하드웨어 환경 구현이다. 현재 다양한 이동 로봇이 시판되고 있으며 우리가 사용한 Pyro 환경은 이동 로봇 5종 이상을 지원한다. 이러한 지원로봇을 이용하여 하드웨어 환경을 구현할 것이다.

참고문헌

- [1] 산업자원부:지능형로봇산업 비전 및 발전전략
- [2] Jeffrey Elkner, "Using Python in High School Computer Science Program", 8th International Python Conference
- [3] Blank, D.S., Kumar, D., Meeden L., and Yanco, H.(2004) Pyro: A Python-based Versatile Programming Environment for Teaching Robotics. Journal of Education Resources in Computing(JERIC)
- [4] Brian P. Gerkey, Richard T. Vaughan, Andrew Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems", Proceedings of the International Conference on Advanced Robotics, 317-323, Coimbra, Portugal, June 30-July 3, 2003
- [5] Auke Jan Ijspeert, Alcherio Martinoli, Aude Billard, Luca Maria Gambardella, (2001) "Collaboration through the exploitation of local interactions in autonomous collective robots: the stick pulling experiment"
- [6] Shervin Nouyan, Marco Dorigo, "Chain Based Path Formation in Swarms of Robots", IRIDIA, CoDE, Universite Libre de Bruxelles, Brussels, Belgium
- [7] Vito Trianni, Marco Dorigo (2005) "Emergent Collective Decisions in a Swarm of Robots", Proceedings of the 2005 IEEE Swarm Intelligence Symposium
- [8] Marco Dorigo, "Swarm-bot: An Experiment in Swarm

Robotics", Proceedings of SIS 2005 - IEEE Swarm Intelligence Symposium
 [9] Eric D. Taillard, "Ant Systems", Technical Report IDSIA-05-99, 1999