

동적인 환경에서 강인한 멀티로봇 제어 알고리즘 연구

홍성우*(부경대 대학원 기계공학과), 안두성(부경대 기계공학과)

Study for Control Algorithm of Robust Multi-Robot in Dynamic Environment

S. W. Hong(Mech. Eng. Dept., PKNU), D. S. Ahn(Mech. Eng. Dept., PKNU)

ABSTRACT

Abstract In this paper, we propose a method of cooperative control based on artificial intelligent system in distributed autonomous robotic system. In general, multi-agent behavior algorithm is simple and effective for small number of robots And multi-robot behavior control is a simple reactive navigation strategy by combining repulsion from obstacles with attraction to a goal. However when the number of robot goes on increasing, this becomes difficult to be realized because multi-robot behavior algorithm provide on multiple constraints and goals in mobile robot navigation problems As the solution of above problem, we propose an architecture of fuzzy system for each multi-robot speed control and fuzzy-neural network for obstacle avoidance. Here, we propose an architecture of fuzzy system for each multi-robot speed control and fuzzy-neural network for their direction to avoid obstacle. Our focus is on system of cooperative autonomous robots in environment with obstacle. For simulation, we divide experiment into two method. One method is motor schema-based formation control in previous and the other method is proposed by this paper Simulation results are given in an obstacle environment and in a dynamic environment.

Key Words . multi-robot (멀티 로봇), 퍼지 시스템(Fuzzy system), 신경망(neural network), fuzzy-neuro system(퍼지뉴로 시스템), behavior algorithm(행동 알고리즘)

1. 서론

지능이 있는 개체를 구현하는 것은 인공지능 및 로봇틱스 등의 분야에서 추구하는 궁극적인 목표라 할 수 있다. 여기에서 개체라 함은 주어진 일과 처한 환경에 대해, 센서를 통해 환경을 인식하고, 적절한 상황판단과 의사결정을 한 후, 가지고 있는 효과기들을 이용해 환경에 반응을 하는 모든 것으로 정의된다. 협력하는 군집 로봇의 장점은 사람들이 협력하여 어떤 목적을 달성하는 경우와 같다. 다시 말해서 군집 로봇은 여러 로봇이 협력하여 단일 로봇이 실행 불가능한 일을 인간의 위험을 배제하고 스스로 작업을 수행하는 자율 다중이동로봇의 구현이라고 할 수 있다.

멀티로봇을 제어하기 위한 방법은 여러 가지가 제시되어 왔다. 지금까지 연구되어온 방법은 크게 3가지로 나누어 살펴볼 수 있다. 첫째로 하나의 물체를 다수 로봇이 붙잡아서 협력하여 원하는 목적지로 물체를 이동시키는 방법이다[1][7]. 이는 로봇의 상호 위치제어뿐만 아니라 힘, 속도, 기구학 제어까지 고려해야 한다. 둘째로 다수 로봇이 이동시 formation을

유지하며 장애물을 회피해 가는 것이다[2][3][5]. 마지막으로 로봇축구 시스템같은 동적인 환경을 주어진게 할 수 있는 시뮬레이션 시스템을 구축하여 축구라는 형태로 멀티 로봇 행동 알고리즘을 구현하는 것이다[8]. 여기에서 두 번째 방법은 멀티로봇 각자의 센서로 서로간에 거리를 유지함으로써 집단의 형태를 서로 유지하려고 하는 작용에 중점을 둔 방법이다. 이는 동물들이 포식자의 공격을 이겨내기 위해 떼, 군집, 무리를 이루어 생활하는 것과 흡사하게 본을 딴 것으로 이동 중에도 이 집단의 형태가 무너지지 않는 것이 중요하다고 하겠다[5]

이 논문에서는 멀티 로봇이 집단 형태를 유지하며 미지의 환경을 통과하는 방법을 소개한다. 로봇은 자신이 지니고 있는 센서만으로 주변의 정보를 알아낸다 지역적 정보만으로 무수히 변화하는 환경을 알아내기 위해서는 적응할 수 있는 특별한 알고리즘이 필요하다. 이를 위하여 인공지능을 이용하게 되고 그 중에서 퍼지와 신경망을 적용하여 환경에 적응할 수 있음을 보여주하고자 한다 로봇은 그 환경에 따라 자신이 움직여야 할 방향과 속력을 퍼지와 신경망을 이용하여 결정하도록 하였으며 그 유효성

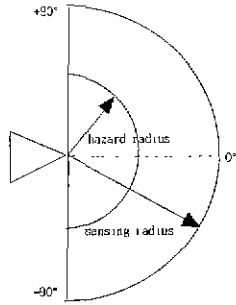


Fig 1 Sensor range of an Autonomous Mobile Robot

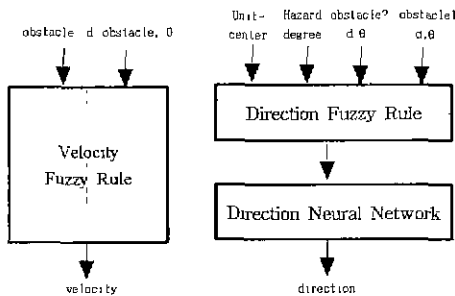


Fig. 2 Block diagram of Autonomous Mobile Robot system

은 컴퓨터 시뮬레이션을 통해 테스트하였다. 컴퓨터 상의 시뮬레이션을 실제 세계에서 완전히 적용하기 어렵다는 것은 주지의 사실이나 이는 진화나 실제 환경의 샘플링 등을 통해 줄일 수 있다는 연구 결과가 있다[5] 논문의 구성은 2에서 이 논문에서 제시한 제어기 원리를 소개하고 3에서는 시뮬레이션을 이용한 실험을 보여주고, 4에서 결론을 언급한다.

2. SYSTEM OVERVIEW

2.1 개요

여기에서 멀티로봇의 군집 행동제어 방식은 Tucker, Manaf의 방식을 기본으로 한다. 이들은 로봇의 형태를 unit-center중심으로 하여 군집 형태를 유지하도록 하였으며 각 로봇의 행동방식은 motor schema-based formation control를 바탕으로 한다. 하지만 위 이론을 완전히 따르지는 않는다. 논문에서 위 방법과 제시된 방법의 확실한 비교를 위해 기본적인 원리만을 이용하여 시뮬레이션을 실행하고 본 논문에서 제시된 방법과 비교하게 된다. 즉 기본의 방법으로는 장애물 회피시 로봇 자신을 기준으로 센싱된 방향의 반대 방향으로만 피하게 된다. 로봇은 독립된 자율 이동로봇이며 이들은 각기 독립된 제어기를 가지고 있다.

motor schema-based formation의 방법은 계산량이 적고 로봇의 센서로 정보를 수집하여 벡터 계산으로

움직임을 결정할 수 있는 장점이 있다. 또한 군집에서 자신의 위치를 적절하게 찾아 갈수 있어 장애물 통과시에도 형태 유지에 요긴하다. 하지만 위에서 제시된 방식은 전체 면적의 일정 범위내의 장애물 통과시 높은 신뢰성을 보였으며 장애물 또한 일정한 크기라는 전체 조건이 있었다. 그리고 움직이는 물체에 대해서는 고려되지 않았다. 여기에서 지름이 각기 다른 장애물이 랜덤하게 분포되어 있을 경우와 움직이는 장애물일 경우 행동방식에 대해 연구해 보겠다. 왜냐하면 장애물을 통과시 행동 제어에 가장 어렵기 때문이다.

여기에서 사용된 자율 이동로봇은 Fig. 1에서 보여주는 것과 같으며 전체조건은 ①로봇은 지역적 센싱 반경내 내용만을 알고 있고 ②센싱 영역은 전방 180°이며, ③각 로봇은 독립된 제어기를 가지며 행동한다고 두겠다.

2.2 퍼지와 신경망

본 논문에서는 제어방법으로 인공지능기법인 뉴럴 네트워크와 퍼지이론을 도입한다. 퍼지 시스템(fuzzy system)은 퍼지 규칙 베이스에 의해 구성된 시스템이나 퍼지값의 파라미터(parameter)를 갖는 시스템을 말하고 신경망은 인간이나 동물들이 가지고 있는 생물학적인 뇌의 신경세포(neuron)을 모델화하여 인공적으로 지능을 만드는 것이다. 이 신경망은 구조화된 지식을 얻기 어려운 문제에 대해서 우수한 성능을 보인다. 퍼지 제어와 뉴럴 네트워크는 서로 장·단점을 내포하고 있다.

퍼지제어는 구조화된 지식을 쉽게 표현할 수 있고 또한 애매한 정보도 쉽게 표현하고 처리할 수 있다. 반면 학습 기능에 대한 메카니즘이 내재되어 있지 않아 시스템 구성시간은 상대적으로 짧으나 만족할 만한 성능을 보이도록 시스템을 조정(tuning)하는 데는 상당한 시간이 필요하다. 신경망은 학습능력과 병렬성을 가지고 있어 자기 조정 기능을 가지고 있으며 분류(classification), 연상 기억(associative), 함수 근사(function approximation)에 사용될 수 있다.

위와 같은 상호 장단점을 상호 보완하는 측면이 있기 때문에 결합됨으로서 성능 향상을 기대할 수 있다. 본 논문에서는 퍼지와 뉴럴 네트워크를 이용하여 효과적인 제어가 되고자 했다. Fig. 2는 전체 제어기 블록도를 나타낸다. 사용된 제어기 부분은 속도와 방향 제어기 부분을 분리해서 결과값을 받는다. 장애물, 목표지점에서는 속도를 줄여 장애물 충돌 및 목표지점을 지나치는 잘못을 없애고자 했으며 방향은 퍼지에서 전처리를 하고 뉴럴네트워크를 이용하여 비퍼지화를 하는 방법으로 다변적인 환경에서 적절한 로봇 방향을 구한다. 방향을 결정하는 제어기에서 퍼지부분에 입력되는 변수는 센싱된 장애물까지 거리와 각도,

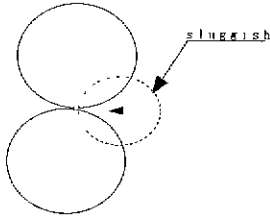


Fig 3(a) Case stop robot when right-left symmetry obstacle avoid in motor schema-based control

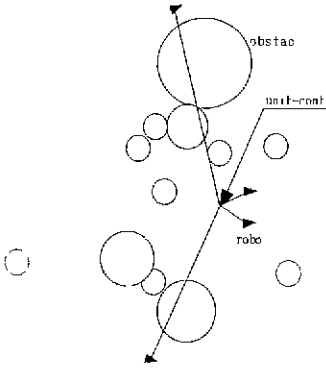


Fig. 3(b) Case go out robot when obstacles avoid

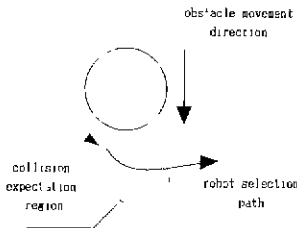


Fig. 3(c) Case robot avoidance when obstacles move

위험도(hazard degree), unit-center와의 거리로 이루어진다. 자세한 설명은 뒤에서 하기로 한다.

2.3 Previous work와 차이점

Behavior-Based formation control에서 motor schema-based control 방식을 사용한 연구를 거론하겠다. 위 방법은 벡터 계산식으로 되어있어 제어하기에 간단하고 여러 로봇의 formation을 유지하며 움직임을 구현할수 있는 장점이 있으나 장애물 통과시 잘못된 지점에 빠질 우려가 있다. Fig. 3은 한 예를 나타내었다. Fig. 3(a)는 로봇을 기준으로 좌우의 장애물에 대해 회피하기 위한 방향을 잡아야 하지만 좌우 대칭적인 장애물로 앞으로 나갈 입구가 막힌 상태에서 어느 방향으로 가야할 지 모르는 상황이다. 즉, 이때의 방향값은 대략

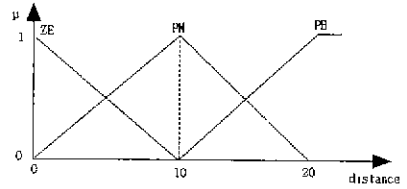


Fig. 4 Distance membership function of the smallest distance obstacle

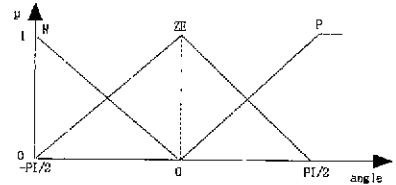


Fig 5 Angle membership function of the smallest distance obstacle

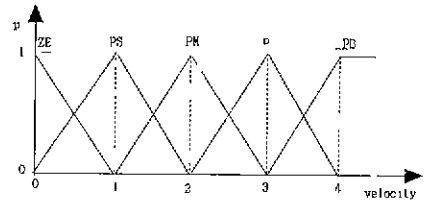


Fig. 6 Velocity membership function of obstacle collision avoidance

Table 1. Fuzzy rules of velocity

angle \ dist	ZE	PM	PB
N	P	PS	PB
ZE	PM	P	P
P	P	PS	PB

$$\theta_{magnitude} = \frac{S-r}{S}$$

S : 로봇의 센싱거리,
r : 로봇과 장애물과의 거리
방향은 Fig. 1을 기준으로 로봇 센싱각도의 + 위치일 경우 - 방향으로 -위치일 경우 + 방향으로 움직인다. 따라서 Fig 3(a)의 경우 좌우 대칭이므로 로봇은 직선방향으로 가게 되고 막혔거나 좁소해도 계속해서 방향은 직선방향만을 가지게 된다. Fig. 3(b)는 많은 장애물을 분포할 시에 형태에서 크게 벗어날 때가 있다. 이때 로봇은 unit-center에서 점점벗어나게 된다. 일정범위를 넘어서면 unit-center에 따라 나머지 로봇들이 각각의 위치를

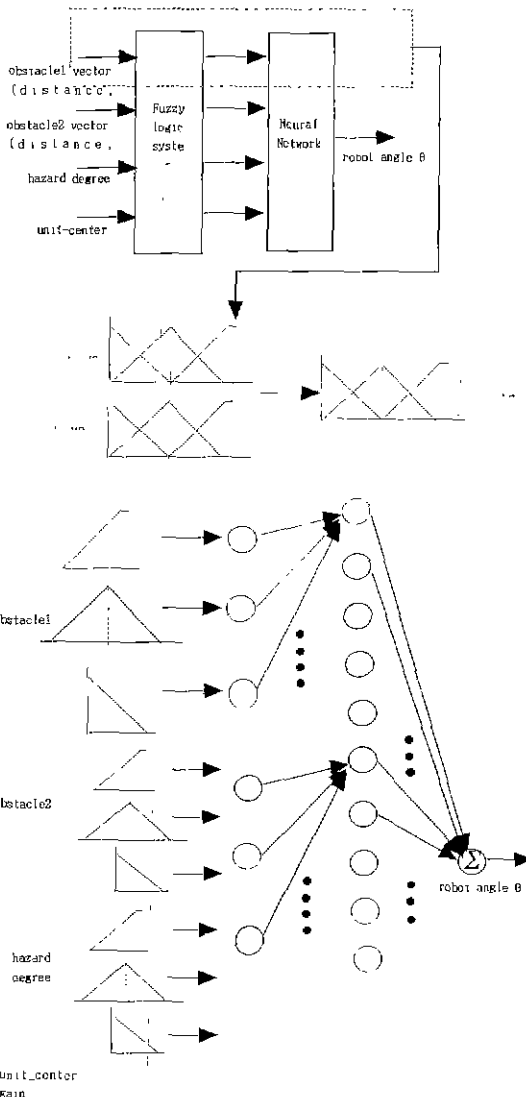


Fig. 7 Example of determine robot angle using fuzzy-neural network

변경을 한다. 이때 한 로봇이 계속해서 장애물이 생겨 위로 올라가면 다른 로봇 또한 동반상승하여 자칫 로봇은 계속해서 위로 상승하여 완전히 이탈을 하게 되는 경우가 생긴다. 실제로 많은 장애물을 두었을 때 이러한 현상은 있었다. Fig. 3(c)는 움직이는 물체의 경우로
경우 장애물의 운동방향으로 로봇은 회피하게 되어 로봇의 운동방향과 직각인 경우를 살펴 보면 점선 원상에서 충돌이 생긴다. 왜냐하면 로봇에서는 장애물이 + 방향에 있고 따라서 자신은 -방향으로 피해야 하기 때문이다.

이를 해결하기 위하여 Tucker과 Maria는 nose

Table 2. Fuzzy rules of obstacle

angle \ dist	ZE	PM	PB
N	N	N	ZE
ZE	P	P	ZE
P	P	P	ZE

Table 3. Fuzzy rules of hazard degree

hazard degree \ change degree	ZE	PM	PB
ZE	P	S	ZE
PM	P	P	S
PB	P	P	P

term을 두어 5 second간 랜덤한 모든 방향($0-2\pi$)에 대해 움직일수 있는 방향을 계산하는 방법을 사용하였다. 하지만 시간이 언제나 지체되고 그 장소에서 계산량이 급격히 많아지게 된다는 단점이 있다. 또한 센싱의 한계로 또 다른 에러에 빠질 우려가 높다.

이에 본 논문에서는 퍼지와 신경망을 이용하여 이러한 단점을 보완하고자 한다.

2.4 제안된 제어 이론

2.4.1 속력 제어기

로봇의 속력은 퍼지 제어기를 이용한다. 속력은 장애물과 목표지점에 가까울 시에는 속력을 줄여서 (motor schema control에서는 빨리 벗어나기 위해 장애물에 가까울수록 속력을 높인다) 속력을 높이다 충돌할 가능성을 없애고 조심해서 통과하기 위함이다. 또한 가고자하는 위치의 목표지점에 가까울 때도 속력을 줄이는 것은 자칫하면 지나칠 수 있기 때문이다 사용한 퍼지의 소속도 함수는 로봇과 가장 가까이 있는 장애물을 선택하며 Fig. 4는 장애물과의 거리에 대한 것이며, Fig. 5에서는 로봇을 기준으로 장애물의 각도를 나타낸 것이다. Fig. 6은 속력을 출력하기 위한 소속도 함수를 보여주며 사용한 퍼지 규칙은 Table. 1에 나타나 있다. 본 논문에서는 Mamdani min implication operator와 max-min composition을 이용하였고 디퍼지화 기법 (defuzzification method)은 무게중심법(Mean of Maxima)을 사용하였다.

2.4.2 방향 제어기

Fig. 7은 로봇의 적합한 방향을 구하기 위하여

구성한 퍼지와 뉴럴 네트워크의 블록 다이어그램을 나타냈다. 그림에서 보듯이 입력변수를 퍼지 논리에 각각 적용을 시키고 출력 소속도 함수값을 뉴럴네트워크 입력 layer에 들어가게 된다. 퍼지 제어기의 입력변수는 로봇 센싱 범위에 있는 모든 장애물의 각각의 거리와 각도, formation을 유지하기 위한 robot team의 unit_center, 지역적 함정과 동적인 장애물등을 피하기 위한 hazard degree가 있다.

신경망은 퍼지 시스템과 분리되어 있으며 학습방법은 출력 노드의 오차를 연결된 이전 노드로 역전과 시켜 입력노드에 이를 때까지 반복하면서 연결강도를 수정하고, 오차가 0.05이하이면 학습이 완료된다.

- 장애물 : 전처리되는 퍼지 논리 회로에 입력되는 변수는 로봇과 장애물과의 거리와 로봇에서 바라본 장애물까지의 각도다. Table. 2는 퍼지 규칙을 나타내었다. 고려하는 장애물은 총 2개로 로봇이 움직이면서 만나게될 센싱 범위내의 장애물은 2개를 넘지 않는다는 전제를 둔다. 사용되는 퍼지 소속도 함수는 속력제어기에서 사용된 소속도 함수와 일치한다. 각각 장애물은 퍼지 제어기를 통해 출력 퍼지 소속도 함수에서 각각 멤버 함수를 뉴럴네트워크에 뉴런에 입력 시킨다. 총 2개의 장애물만을 고려하고 멤버함수는 3개씩이므로 장애물 하나에 뉴런의 수는 3개. 총 6개의 뉴런이 있게 된다. 만일 하나의 장애물만이 센싱되면 다른 하나의 장애물에는 출력 소속도 함수가 모두 0이 되도록 한다.

- Hazard degree : 장애물을 만나 회피하면서 로봇의 hazard radius내에 들어오게 되면 Hazard degree값이 증가하게 된다. Hazard부분은 가장 가까운 장애물과의 hazard degree와 hazard 증가율 소속도 함수로 나누어 진다. Table. 3은 위험도와 위험도 증가율에 대한 퍼지 규칙을 나타내었다. 만일 로봇이 Fig. 3(a)와 같은 상황이라면 로봇이 계속적으로 전진하면서 장애물과의 거리가 가까워 가는 hazard degree값은 증가하고 함께 증가율도 적게 커질 것이다. 만일 움직이는 장애물이라면 증가율의 폭은 커진다. hazard degree와 증가율을 구하는 식은 다음과 같다.

$$\delta_{degree} = \frac{r_{hazard} - d_{r-ob}}{r_{hazard}}$$

$$d \delta_{change} = \delta_{degree_{current}} - \delta_{degree_{previous}}$$

δ_{degree} : hazard degree

$d \delta_{change}$: the ratio of hazard degree change (1 to 0)

r_{hazard} : robot hazard radius

d_{r-ob} : distance between robot and obstacle



Fig. 8. Environment simulation

$\delta_{degree_{previous}}$: hazard degree in previous

$\delta_{degree_{current}}$: hazard degree in current

- Unit-center : 로봇의 장애물 회피 운동 중에 군집의 중심에서 벗어나는 것을 예방하기 위한 항목이다. 로봇이 장애물 회피중 일정범위를 넘어서면 다음 장애물에서 unit-center 방향으로 돌아서도록 유도하고자 하는 것이며 이부분은 normalizing된 gain값으로 처리한다.

3. 실험 방법 및 결과

3.1 시뮬레이션 환경

Fig. 8은 실험하기위한 시뮬레이션 환경을 구현한 것이다. 실험하는 formation은 V자 형을 중심으로 한다. 그외에도 diamond, column, line형 등이 있으나 column와 line은 상대적으로 접근하기 쉬우며 diamond형 보다 V자 형이 구현하기 까다롭기 때문이다. 로봇은 4대를 기준으로 실시하였다. 환경은 200(m)× 600(m)운동장에 지금 6m의 원형 장애물이 랜덤하게 흩어져 있다고 가정한다. 로봇의 크기는 1m이며 센싱가능 거리는 20m라고 하겠다.

실험 방법은 장애물 종류에 따라 3가지로 나누어져 있다. 첫째, 로봇이 장애물 사이로 통과하는 경우로서 그 폭이 좁아서 로봇이 통과하기에 적당하지 못하거나 막혔을 때 로봇의 움직임을 살펴본다. 둘째, 다수 장애물 통과로 unit-center에서 크게 벗어났을 경우 로봇의 움직임을 본다. 셋째, 움직이는 장애물에 대해 살펴본다. 장애물의 운동은 수직운동으로 한정되어 있다.

3.2 재시원 제어기를 위한 시뮬레이션

각각의 조건에 대한 결과를 Fig. 9, Fig. 10, Fig. 11에서 보여주고 있다. Fig. 9는 Fig. 3. (a)에서의 경우 제시된 제어기를 썼을 때 움직이는 경로를 나타내었다. Fig. 10의 경우는 Fig. 3. (b)의 경우의 로봇 경로이다. Fig. 11의 경우 움직이는 장애물일 경우 로봇이 회피하는 모습을 보인다. 이때 장애물의 속력은 로봇의 1/2로 정하였다.

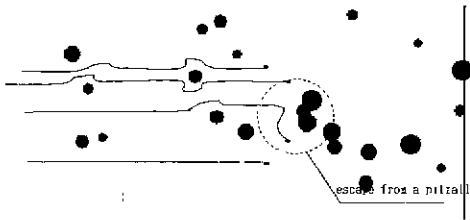


Fig. 9. Simulation 1

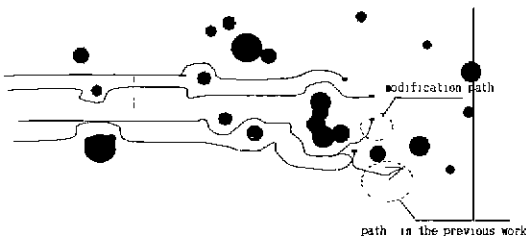


Fig. 10. Simulation 2

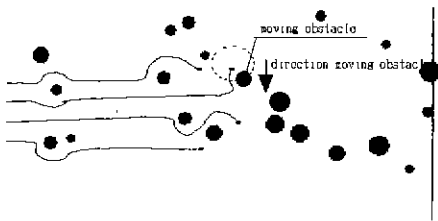


Fig. 11. Simulation 3

4. 결론

우리는 멀티로봇의 행동을 위한 제어기를 설계를 하였다. 여러 가지 상황에 적당한 값을 가지기 위하여 신경망과 퍼지의 장단점을 이용하여 속도 제어기에는 퍼지 시스템을 방향 제어기에는 퍼지와 신경망 제어기를 합성하여 구성하였다. 이 논문에서는 formation을 유지하는 것은 기존의 방법을 사용하였으나 장애물 회피시의 단점을 보완하고자 하는 것에 중점을 두었다. 그리고 시뮬레이션 결과 여러 가지 상황별로 좀더 강인한 면을 보이고 있다. 또한 Fig. 10에서 unit center에서 벗어나더라도 전체가 아닌 하나의 로봇된의 위치 수정으로 움직임의 결약성도 보였다.

향후, 상황에 따라 적합한 formation을 판단하고 학습 능력을 자생적으로 이룰수 있도록 하고 하드웨어적으로 구성하여 실세계에 구현 할 것이다.

참고문헌

1. ZhiDong WANG, Majid Nili Ahmadabadi, Eiji NAKANO, and Takayuki TAKAHASHI "A Multiple Robot System for Cooperative Object Transportation with Various Requirements on Task Performing", IEEE International Conference on Robotics & Automation, pp. 1226-1233, Detroit, Michigan, May 1999.
2. Jaydev P. Desai, Vijay Kumar, James P Ostrowski "Control of changes in formation for a team of mobile robots", IEEE International Conference on Robotics & Automation, pp. 1556-1561, Detroit, Michigan, May 1999.
3. Toshio FUKUDA, Hiroo MIZOGUCHI, Kosuke SEKIYAMA, Fumihito ARAI "Group Behavior Control for MARS(Micro Autonomous Robotics System)", IEEE International Conference on Robotics & Automation, pp 1550-1555, Detroit, Michigan, May 1999.
4. O. Miglino, H. H. Lund and S. Nolfi, "Evolving mobile robots in simulated and real environments", *Artificial Life*, vol. 2, no. 4, pp. 417-434, MIT Press, 1996.
5. Tucker Balch, Ronald C. Arkin. "Behavior-Based Formation Control for Multirobot Teams", IEEE Transaction on Robotics and Automation, vol. 14, no. 6, pp. 926-939, DECEMBER 1998.
6. Timothy B. Gold, James K. Archibald, Richard L. Frost. "A Utility Approach to Multi-Agent Coordination", IEEE International Conference on Robotics & Automation, pp. 2052-2057, San Francisco, CA, April 2000.
7. Kazuhiro Kosuge, Yasuhisa Hirata, "Motion Control of Multiple Autonomous Mobile Robots Handling a Large Object in Coordination", IEEE International Conference on Robotics & Automation, pp. 2666-2673, Detroit, Michigan, May 1999.
8. Sung Ho Kim, Jong Suk Choi, and Byung Kook Kim "Development of BEST Nano-Robot Soccer Team", International Conference on Robotics & Automation, pp 2680-2685, Detroit, Michigan, May 1999.