

계층적 구조에 의한 다중이동로봇의 협동제어에 관한 연구

박성규\*, 이기성  
홍익대학교 전기제어공학과

A Study on the Cooperative Control of Multiple Mobile Robots Using a Hierarchical Structure

Sung-Kyu Park\* , Keeseong Lee  
Dept. of Electrical and Control Engineering Hongik University

**Abstract** - A hierarchical structure for the cooperative control of multiple mobile robots using coordinates of objects obtained from vision system is proposed. The order-level perceives environments represented by workspace sets. The algorithm selects an object to be moved using an object discrimination part and determines the robot actions. The action-level generates a trajectory of each wheel velocity of robot. The simulation results show the effectiveness of the proposed algorithm.

1. 서 론

지능이 있는 개체를 구현하는 것은 인공지능이나 로봇 텍스 등의 분야에서 추구하는 궁극적 목표 중의 하나라고 할 수 있다. 여기서 개체라 함은 주어진 일과 처한 환경에 대해 센서를 통해 환경을 인식하고 적절한 상황 판단과 의사결정을 한 후 가지고 있는 효과기들을 이용해 환경에 반응을 하는 모든 것으로 정의[1]된다. 지능의 정의는 관점에 따라 조금씩 다를 수 있다. 제어이론의 관점에서 볼 때 지능이란 환경을 감지하고 결정을 내리고 행동을 제어하며 추론 인식 이해등의 기능을 포괄하여 말한다. 특히 환경이 매우 동적이고 불확실한 요소를 많이 가지고 있다면 그러한 환경 속에서의 개체는 더욱 높은 지능정도가 요구된다. 더욱이 주어진 일이 많거나 복잡하여 단일 개체가 아닌 다 개체 시스템이 사용될 경우 개체들간의 상호 협동이 필요하다[3]. 주어진 환경과 일에 대해 어떤 행동을 할 것인가를 결정하는 것은 개체에게 주어진 가장 기본적인 문제라고 할 수 있으며 로봇의 행동에 대한 결정이 실시간에 대응할 수 있으면 행동 결정단계에 있어서 매우 효과적이며 간략화 되어야 하고, 또한 그 결정에 정확성이 뒤따라야 한다. 로봇이 어떤 행동을 선택할 것인가라는 개체의 행동선택문제에 대한 연구들이 있어왔다[4][5][6]. 그 중에는 동물의 행동 선택에 초점을 맞추어 계층적 결정 구조와 그것을 시험하기 위한 모의 환경을 구현하고 많은 다른 행동 선택기들과의 비교 연구도 있었으며[4,5], 다개체들 간의 협동을 위한 행동 선택기를 제안하고 실내의 가구나 바닥 청소 같은 비교적 정적인 모의 환경 하에서의 결과를 보인 연구도 있었다[6]. 하지만 실제 세계는 그 환경이 동적이며 불확실한 것이 대부분이다. 또한 계층적 구조의 명령 전달 방식이 도입되었다[9]. 계층적 제어구조는 시스템의 신뢰도와 작업완성도를 개선시킬 수 있으며 전체적인 제어구조의 설계가 용이한 장점이 있다. 이에 본 논문에서는 계층적 제어구조를 이용하여 비교적 간단하면서도 신뢰성있는 제어를 사용하고 협동에 관계되는 대표적인 문제들을 효율적으로 해결하는데 목적을 두었다.

2. 제어 구조

제어기는 복잡하고 예측하기 어려운 작업환경의 상황을 보다 빠르고 효과적으로 인식하고 판단하기 위해서 계층적 제어구조를 가진다. 기존에 사용되었던 유전 알고리즘을 이용한 제어구조는 최적값을 찾을 때 유용하게 사용되기는하나 그 최적값을 찾기까지의 시간과 연산량이 늘어나는 단점을 가진다. 정해진 시간하에서 로봇이 효과적으로 주어진 작업을 마치기엔 무리가 따른다. 본 논문에서 사용한 제어구조는 유전알고리즘이나 신경회로망보다는 보다 간단한 구조로 구성되어 있으며 또한 연산량이 적다는 장점이 있다. 명령계층에서는 비전 시스템에서 인식되어 획득된 물체의 좌표와 로봇의 좌표, 목표점의 좌표 물체와 목표점과의 각도, 로봇과 물체와의 각도 등의 데이터를 토대로 로봇이 처해있는 환경을 단순화하고 환경에 따라 나뉜 로봇의 상태에 따라 로봇이 취해야 할 적절한 행동을 결정하게 되고 행동계층에서는 명령계층에서 로봇의 상태를 결정짓는 데이터를 사용하여 transport point와 push point를 계산한 뒤 로봇이 이동할 수 있는 속도케적을 생성하여 로봇을 구동하게 된다.

2.1 명령 계층

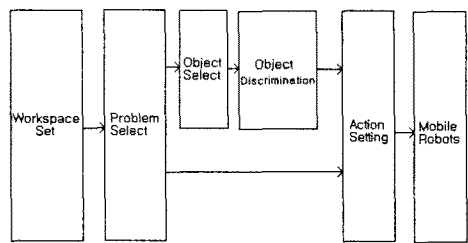


그림 1. 제어 구조

명령계층은 Workspace Set과 로봇, 물체, 목표점이 각각 지니고 있는 각도를 이용하여 작업환경의 상태를 판단하고 로봇이 놓여있는 상태를 결정짓고 결정되어진 상태의 로봇들은 각각 주어진 상태에서 취하게 되는 행동을 결정짓게 되는데, 여기에서 로봇이 각 상태에 따라 취할 수 있는 행동 부분으로 구성되어 있다.

Workspace Set은 작업환경 내에 존재하는 목표점과 운반할 물체, 로봇의 상태를 간략하게 나타낸 모델이며 이들이 지니고 있는 각각의 X 좌표에 관한 내림차순의 순서로 나열되는 행태를 지니게 된다. 로봇의 경우 로봇의 중심좌표를 기준으로 한다. Workspace Set에서는 작업환경 내에서 로봇과 공, 목표점이 나열되는 상태만 알면 되므로 만일 로봇의 좌측, 또는 우측에 운반되어질

물체가 놓여 있는 상태라 하더라도 그것은 로봇의 앞에 있다고 가정한다. 그래서 8개의 상태만으로 작업환경에서 일어날 수 있는 상황을 효과적으로 모델링 할 수 있다.

로봇과 이동시킬 물체와 목표점과의 관계를 위에서 언급한 방법대로 나열한다면,

로봇 - 물체 - 목표점, 물체 - 로봇 - 목표점, 목표점 - 로봇 - 물체, 목표점 - 물체 - 로봇인 4종류로 크게 분류할 수 있다.

이것을 다시 표현하면 다음과 같다.

$$\begin{aligned} WS_{RA} &= \{ R_fOG, R_bOG, OR_fG, OR_bG \} \\ WS_{LA} &= \{ GO R_f, GO R_b, G R_fO, G R_bO \} \end{aligned} \quad (1)$$

$WS_{RA}$  : 목표점이 오른쪽일 때의 Workspace Set

$WS_{LA}$  : 목표점이 왼쪽일 때의 Workspace Set

$R_fOG$  : Robot's Forward - Object - Goal

$R_bOG$  : Robot's Backward - Object - Goal

$OR_fG$  : Object - Robot's Forward - Goal

$OR_bG$  : Object - Robot's Backward - Goal

$GO R_f$  : Goal - Object - Robot's Forward

$GO R_b$  : Goal - Object - Robot's Backward

$G R_fO$  : Goal - Robot's Forward - Object

$G R_bO$  : Goal - Robot's Backward - Object

또한 명령계층에서 로봇의 상태를 결정하는 요소로는 로봇과 공이 가지는 각도와 공과 목표점이 가지는 각도가 사용되는데 로봇의 각도는 오른쪽을 보고 있는 상태를 0라고 정의하고 그로부터 시계방향으로 각도가 증가한다. 이와 같은 각도 체계는 비단 로봇에 국한된 것이 아니며 작업환경 내에 존재하는 모든 물체의 각도에 적용된다.

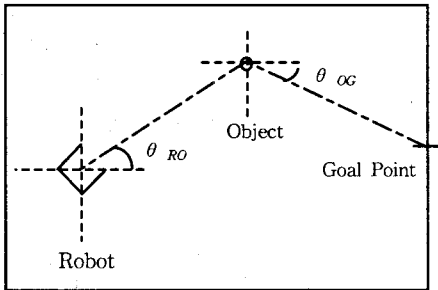


그림 2. 로봇과 물체, 물체와 목표점과의 각도

그림 2는 로봇과 물체가 이루는 각도  $\theta_{RO}$  와 물체와 목표점이 이루는 각도  $\theta_{OG}$  를 나타내고 있다. 매 샘플링 시간의 Workspace set과 위의 그림에서 보여진 각도에 따라 결정된다.

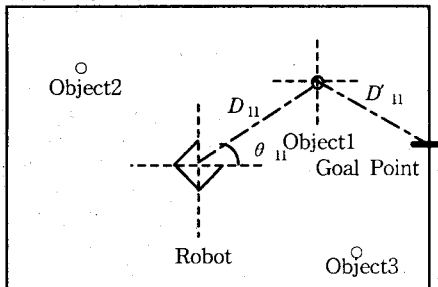


그림 3. 로봇과 물체 사이에 생성되는 거리와 각도

그림 3은 로봇과 각 물체간에 생길 수 있는 각각의 변수의 생성과정을 나타낸다. 각 로봇에서 운반시킬 물체의 선택 과정은 각 로봇과 운반시킬 물체가 이루는 각, 로봇과 물체 사이의 거리, 목표점과 물체사이의 거리에

workspace set에서 표현된 상태에 따라서 결정된  $w_j$  에 의해서 계산된 값들의 최소값으로 운반할 물체를 선택하며, 이것은 다음과 같은 식으로 표현될 수 있다.

$$X_1 = R_1 \min\{(\theta_{11} + D_{11} + D_{12})w_j, \dots, (\theta_{1n} + D_{1n} + D_{12})w_j\} \quad (2)$$

$$X_2 = R_2 \min\{(\theta_{21} + D_{21} + D_{22})w_j, \dots, (\theta_{2n} + D_{2n} + D_{22})w_j\} \quad (3)$$

$R_1$  : 로봇 1,  $R_2$  : 로봇 2

$D_{1i}$  : 로봇 1과 물체의 거리

$D_{1i}$  : 목표점과 로봇 1 선택한 물체의 거리

$D_{2i}$  : 로봇 2과 물체의 거리

단,  $D_{2i}$  : 목표점과 로봇 2 선택한 물체의 거리

$w_j$  : Object Weight

$j = \begin{cases} 0 & \text{물체가 로봇의 뒤에 있을 경우} \\ 1 & \text{물체가 로봇의 앞에 있을 경우} \end{cases}$

$w_0 = 0, w_1 = 1$

$i = 1, 2, \dots, n$

그러나, 위의 식으로는 운반하고자하는 물체의 상태가 서로 인접해 있거나 두 로봇이 같은 물체를 선택했을 경우 해결할 수 없는 단점을 가진다. 그러므로 다음에 보여지는 식에 따라 운반할 물체의 상태를 식별한다.

$$\sqrt{|O_{2y} - O_{1y}|^2 + |O_{2x} - O_{1x}|^2} = f \quad (4)$$

$O_{1x}$  : 로봇 1이 선택한 물체의 X 좌표

단,  $O_{1y}$  : 로봇 1이 선택한 물체의 Y 좌표

$O_{2x}$  : 로봇 2이 선택한 물체의 X 좌표

$O_{2y}$  : 로봇 2이 선택한 물체의 Y 좌표

$f = \begin{cases} 0 < f < 2r & \text{서로 같은 물체를 선택한 상태} \\ 2r < f < 2.5r & \text{물체가 서로 인접한 상태} \\ f > 2.5r & \text{서로 다른 물체를 선택한 상태} \end{cases}$

단, 물체의 반지름은  $r$ 로 가정한다.

(a) 서로 같은 물체를 선택했을 경우

$$X_3 = \min\{D_{OR1}, D_{OR2}\} \quad (5)$$

단,  $D_{OR1}$  : 로봇 1과 물체의 거리

$D_{OR2}$  : 로봇 2과 물체의 거리

(7)식으로 결정된  $X_3$ 값을 바탕으로 최소거리를 가지는 로봇이 물체의 우선권을 가지며, 다른 로봇은 다른 물체를 선택한다.

(b) 물체가 서로 인접해있는 경우

$$\frac{O_{x2} + O_{x1}}{2} = \text{push point X} \quad (6)$$

$$\frac{O_{y2} + O_{y1}}{2} = \text{push point Y} \quad (7)$$

(c) 서로 다른 물체일 경우

계산되어진 point로 각각의 로봇이 이동한 후 궤적을 따라 이동하게된다.

상태가 결정되어진 로봇들은 각각 주어진 상태에서 취하게 되는 행동을 결정짓게 되는데, 여기에서 로봇이 각 상태에 따라 취할 수 있는 행동에는 transport와 push 요소로 구성되며 Action의 결정 역시 명령계층에서 매 샘플링 시간의 State에 따라서 결정되어지며, 로봇의 목표점은 오른쪽으로 가정한다. 편의상 1대의 로봇에 관한 경우만 살펴보기로 한다.

$$\begin{cases} R_1 = R_fOG \text{ or } OR_bG \text{ and } \{ (0 < f < 2r \text{ or } f > 2.5r) \} \\ R_1 = \{ \text{Transport} \} \end{cases} \quad (8)$$

$$\begin{cases} R_1 = R_fOG \text{ or } OR_bG \text{ and } \{ 2r < f < 2.5r \} \\ R_1 = \{ \text{push} \} \end{cases} \quad (9)$$

$R_1$  과  $R_2$  는 각각 로봇을 나타내며  $f$ 는 (4)식에서 얻어진 값이다.

## 2.2 행동 계층

행동계층에서는 명령계층에서 부여받은 transport point 또는 push point에서 로봇이 상태에 대해 적절하게 이동할 수 있는 양 바퀴의 속도를 생성한다.

그림 4은 로봇이 회전해야 하는 각도를 결정하기 위한 그림이며 로봇 진행방향의 원래의 각도를  $\theta$ , 로봇의 중심으로부터 goal point까지의 각도를  $\phi$ 라 하면 로봇의

회전각  $\psi$  은 다음과 같다.

$$\psi = \theta - \varphi \quad (10)$$

따라서 로봇의 이동방향은  $\psi=0$ 을 만들 수 있는 방향으로 회전속도를 결정하며 다음의 식으로 양쪽 바퀴의 속도를 구할 수 있다.

$$w_l[t] = w[t-1] + \Delta w \quad (11)$$

$$v_l[t] = v \quad (12)$$

$$v_r[t] = \frac{v[t] + R \times \omega}{r} \quad (13)$$

$$v_l[t] = \frac{v[t] - R \times \omega}{r} \quad (14)$$

단,  $v$ 는 로봇중심의 속도,  $v_r$ 은 우측바퀴의 속도,  $v_l$ 은 좌측바퀴의 속도이며  $R$ 은 바퀴간의 거리,  $r$ 은 바퀴의 반지름이다.  $\Delta w$ 는 각속도 증가분,  $\omega$ 는 각속도를 나타낸다.

### 3. 시물레이션

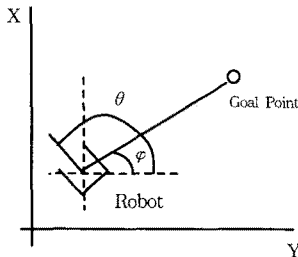


그림 4. 로봇의 이동

#### 3.1 산재되어있는 물체의 정리 문제

주어진 작업 환경 내에서 로봇에 주어진 작업의 양이 많아 작업시간이 오래 걸릴 경우 한대 이상의 로봇이 주어진 작업을 함께 한다면 작업시간이 단축될뿐더러 작업의 효율을 높일 수 있을 것이다. 그러므로 로봇들이 산재되어있는 물체들을 정해진 일정한 장소로 운반하는 것이 목적이다. 편의상 작업환경의 가로와 세로의 길이를 2m로 가정하고, 작업환경 내에는 두 대의 로봇이 각각 존재하며 로봇의 이동속도는 5cm/s, 목표지점은 우측 사각형으로 설정하였고 운반할 물체의 크기는 로봇의 1/4의 크기로 가정한다.

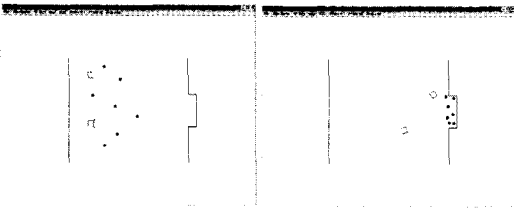


그림 5

그림 6

그림 5는 두 대의 로봇과 이동될 물체의 상태를 나타내고 그림 6은 로봇에 의해서 물체가 목표지점안으로 이동되어진 후의 상태를 나타낸다. 표1에서는 로봇이 이동시킬 물체의 이동 점을 찾을 때까지의 연산시간과 로

	유전 알고리즘	계층적 제어구조
연산 시간	97ms	28.6ms
로봇의 총 이동거리	17.49m	17.23m

표 1. 산재되어있는 물체의 정리시

봇의 총 이동거리를 비교한 것이다.

#### 3.2 물체 밀어내기 문제

하나의 로봇이 운반하기 힘든 물체의 운반에 여러 로봇을 함께 사용하면 일을 수행할 수가 있을 것이다. 그러나 여러 개의 로봇이 있어도 서로 협조가 되지 않는다면 그 일을 수행하기는 어려워진다. 그러므로 로봇들이 일단 물체 앞으로 모인 후 동시에 상자를 밀어낼 수 있는 로봇의 물체 이동 점을 찾는 것이 목적이다. 밀어낼 물체는 로봇의 두 배의 크기로 가정하였다.

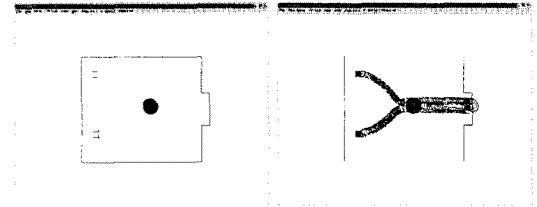


그림 7

그림 8

그림 7은 두 대의 로봇과 이동시킬 물체가 설정된 상태를 나타내고 그림 8은 로봇이 물체를 작업환경 밖으로 이동시키기 위해 밀어낼 수 있는 이동점으로 접근한 후의 상태를 나타내며 표2에서는 로봇이 이동시킬 물체의 이동 점을 찾을 때까지의 연산시간과 로봇의 이동거리를 비교한 것이다.

	유전 알고리즘	계층적 제어구조
연산 시간	42ms	22ms
로봇의 총 이동거리	1.78m	1.73m

표 2. 물체 밀어내기 문제시

### 4. 결 론

본 논문에서 제안하고 있는 알고리즘은 로봇 주위의 동적인 작업환경의 상태를 최대한 단순화하여 각 로봇에 공격과 수비의 상태를 주어주고 그 주어진 상태에 따라 로봇의 행동 양식이 결정이 되며 행동 양식에 따라 로봇이 이동할 때 필요한 속도가 주어지는 구조 방식이다. 본 논문에서 제안된 제어구조는 계층적 구조에 근간을 두고 있으며 기존의 유전 알고리즘이나 신경회로망을 이용한 로봇의 협동제어에서 발생할 수 있었던 연산량 증가를 보완하고 구조를 보다 단순화하면서도 주어진 작업환경의 상황을 효과적으로 모델링함으로써 성능향상을 유도하였으며 제안된 제어구조의 성능을 시물레이션을 통해 확인하였다.

#### (참 고 문 헌)

- [1] S. Russell and P. Norvig, *Artificial Intelligence*, Prentice-Hall, 1995.
- [2] R. C. Baker and B. Charlie, "Nonlinear unstable systems," *International Journal of Control*, Vol. 23, No. 4, pp. 123-145, May, 1989.
- [3] A. B. K. Y. Uny Cao, A. S. Fukunaga and F. Meng, "Cooperative Mobile Robots: Antecedents and directions," in *Intelligent Robots and Systems*, IEEE Computer Society Press, pp. 226-234, August 1995.
- [4] T. Tyrrell, *The Technical Writer's Handbook*, Mill

Valley, Seoul, 1989.

- [5] T. Tyrrell, "The use of hierarchies for action selection," in *From animals to animats 2*, The MIT Press, pp. 138-147, 1993.
- [6] L. E. Parker, "Adaptive action selection for cooperative agent teams," in *From animals to animats 2*, The MIT Press, pp. 442-450, 1993.
- [7] Atsushi Fujimori, Peter N. Nikiforuk, Madan M. Gupta, "Adaptive Navigation of Mobile Robots with Obstacle Avoidance" *IEEE Trans. Robotics and Automat.* Vol. 13 No. 4 August 1997.
- [8] David B. Reiter, Francois G Pin, "Time-optimal Trajectories for Mobile Robots With Two Independently Driven Wheels" *The International Journal of Robotics Research* Vol. 12 No.2 February 1993.
- [9] 이인재, 백승민, 손경오, 국태용, "다중이동로봇의 계층적 제어" 제어자동화시스템공학회 논문지, 제 13권, pp. 252-255, 1998.