

# 강한 결합조건을 갖는 두 이동로봇의 협동 운동계획

## Cooperative Motion Planning of Two Tightly-Coupled Mobile Robots

이 승 환, 이 승 하, 이 연 정

(Seung Hwan Lee, Seung-ha Lee and Yun-Jung Lee)

**Abstract** : In this paper, we propose a cooperative motion planning algorithm for two tightly-coupled mobile robots. Specifically, the considered cooperative work is that two mobile robots should transfer a long rigid object along a predefined path. To resolve the problem, we introduce a master-slave concept for two mobile robots having the same structure. According to the velocity of the master robot and the positions of two robots on the path, the velocity of the slave robot is determined. The slave normally tracks the master's motion, but in case that the velocity of the slave exceeds the velocity limit, the roles of the robots should be interchanged. The effectiveness of the proposed algorithm is proved by computer simulations.

**Keywords** : mobile robot, path-tracking, cooperation, motion planning, master-slave

### 1. 서론

현대의 공장에서는 공산품을 생산하는 방식은 다품종 소량생산이 주가 되어가고 있다. 이러한 상황에서는 작고 단순한 로봇 여러 대의 협동작업을 통하여 복잡한 작업을 수행하는 편이 거대한 전용로봇을 사용하는 것보다 더 쉽게 변화하는 환경에 적응할 수 있다.

이와 같은 이유에서 지금까지 여러 방식의 다중 협동로봇에 대한 연구[1]-[6]가 활발히 진행되어 왔다. 하지만 지금까지의 대부분의 협동작업 로봇에 관한 연구는 로봇들이 각각 짐을 실어 나를 때의 교통 관제 방법 [1][2]이나 여러 대의 로봇이 서로 방해하지 않고 목표위치에 잘 도달하는 방법[3], 잔디 깎기나 청소작업처럼 여러 대의 로봇이 효율적으로 특정 영역을 빠짐없이 쓸고 지나가야 하는 방법[4]등과 같이 각각의 로봇이 혼자 할 수 있는 일을 여러 대가 함께 할 때 서로 방해받지 않고 효율적으로 작업을 수행하는 방법이 주로 연구되어 왔다. 이러한 협동작업 문제들은 로봇들간의 결합조건이 약한 (loosely-coupled) 경우로 볼 수 있다.

그러나 협동작업의 종류에 따라서는 한 대의 로봇으로 주어진 작업을 수행할 수 없고 여러 대의 로봇이 서로 긴밀하게 협동을 해야만 할 수 있는 일이 있다. 예를 들면 하나의 로봇으로는 운반하기 어려운 긴 물체를 두 로봇이 각각 한쪽 끝을 지지하여 운반해야 하는 경우는 약한 결합조건이 아닌 강한 결합조건(tightly-coupled condition)을 갖는 협동작업이라 할 수 있다. 이와 같은 강한 결합조건을 갖는 협동작업의 경우는 약한 결합조건일 때에 전혀 고려되지 않았던 각각의 로봇들간의 협력 문제를 고려해야만 한다. 즉, 약한 결합조건인 경우는 상대 로봇을 움직이는 장애물로 간주하여 특정 범위이내로

서로 근접하지 않으면 상대 로봇의 위치 및 속도가 자신의 움직임을 결정하는데 크게 관여하지 않지만, 강한 결합조건을 갖는 경우는 다른 로봇들의 위치와 속도가 하나의 로봇의 움직임을 결정하는데 있어 항상 고려해야 할 중요한 요소가 된다.

본 논문에서는 상기한 강한 결합조건을 갖는 협동작업 문제의 하나로써, 주어진 경로(path)를 따라 일정 길이의 물체를 두 대의 이동로봇이 협동하여 운반해야 하는 경우, 경로를 이탈함이 없이 제한된 속도 범위 내에서 각 로봇의 운동계획(속도 및 주종역할변환) 문제를 다루고자 한다.

이러한 강한 결합조건을 갖는 경우 문제를 해결하기 위해서 두 가지 방식의 접근방법을 생각해 볼 수 있다. 운반해야 하는 물체의 전체적인 이동 경로(trajectory)를 미리 결정하고 그에 따라 각각의 자체 지능이 없는 로봇들의 속도와 위치 등을 중앙집중식으로 결정하는 방식 [5]이며, 다른 한가지 방식은 본 논문에서 적용한 방법으로 어느 정도의 지능을 가진 각각의 로봇들이 주위의 경로(path)와 운반해야 하는 물체에 의해 발생하는 결합조건을 상호 통신을 통해 교환함으로써 각각의 로봇이 어떠한 속도로 움직일지를 결정하는 분산식 방법이다.

구체적으로 제안된 방법을 설명하면, 두 이동로봇들의 이동 속도는 각 로봇이 위치한 주변의 부분적인 경로에 대한 정보를 상호 통신을 통하여 알려줌으로써, 매 순간 각 로봇이 위치한 부분경로에 따라 분산식으로 각각의 속도를 결정한다. 또한, 경로의 형태에 따라서는 두 로봇이 경로를 이탈하지 않고 일정 길이를 유지하기 위해 결정된 속도가 제한된 범위를 벗어나는 경우가 발생하는데, 이러한 문제를 해결하기 위하여 본 논문에서는 주-종(master-slave) 역할 교체 개념을 새롭게 도입하였다. 제안된 방법은 자체 제작한 윈도우즈 프로그램을 이용한 모의 실험을 통하여 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에

서 다루고자하는 문제에 대해 상세히 기술하고, 3장에서는 각 부분경로에서의 두 로봇의 위치와 속도 관계식을 유도하고, 주-종 로봇의 역할 및 주-종 역할 교체에 대해 기술하며, 제안된 알고리즘의 모의실험을 통한 검증 결과에 대해 4장에서 언급하고, 마지막으로 5장에서 결론을 맺는다.

II. 문제 설정

먼저 공사에서 긴 버팀목이나 파이프를 두 사람의 인부가 특정지역으로 이동시키는 경우를 생각해 보자. 두 인부는 최종 목표를 알고 있기는 하지만 최종 목표에 도달하기 위한 완전한 경로를 가지고 있지는 않다. 단지 인부들은 시각정보를 통하여 최종 목표에 도달하기 위해서 그때그때 설정한 자기 주변의 부분적인 경로에 대한 정보만을 가지고 있다. 이러한 경우 버팀목을 이동시키기 위해서 보통은 앞의 인부가 주(master)가 되어 종(slave)이 된 뒤쪽의 인부를 이끌고 나간다 엄밀히 얘기 하면 앞의 인부가 시각정보에 의해 임의로 결정한 자신의 부분경로를 임의의 속도로 진행을 하면 뒤쪽의 인부는 우선 자신의 부분경로를 시각정보에 의해 결정하고 버팀목으로부터 전달되어오는 힘을 감지하여 앞의 인부의 속도와 위치를 추정한 후에 앞의 인부의 속도와 위치에 맞추어 얼마만한 속도로 자신의 부분경로를 진행해야 할지를 결정한다. 이때 자신이 가져야 할 속도나 위치가 현실적으로 불가능한 것이라면 앞의 인부에게 잠시 서달라는 요청을 한 후에 이번에는 이전과는 반대로 뒤의 인부 마음대로 속도를 결정하고 앞의 인부가 뒤의 인부 속도에 맞추어 자신의 속도를 결정하여 즉, 주-종 역할을 교체하여 계속 진행해 나아간다. 이러한 과정을 반복하여 두 인부는 버팀목을 원하는 위치까지 옮긴다. 주-종 개념을 정리하면 자신의 속도를 상대에게 제한을 받지 않고 임의대로 설정하는 쪽을 '주'라 부르며 상대의 속도에 따라 자신의 속도가 결정되는 쪽을 '종'이라고 한다.

본 논문에서 고려하는 협동작업 문제는 위에서 설명한 인부가 버팀목을 나르는 문제와 유사하며, 자동화된 공장에서 무인 운반차 두 대가 긴 강성 물체를 바닥에 그려진 정해진 경로를 따라 옮기는 문제와도 같은 것이다. 구체적으로 다루고자하는 문제를 정리하면 다음과 같다.

문제 : 주어진 경로(path)를 따라 출발점에서 목표점까지 일정 길이의 강체를 두 대의 이동로봇이 협동하여 운반하기 위한 각 로봇의 이동 속도를 구한다. 단, 두 로봇은 형태나 구조가 동일한 동형 로봇(homogeneous robots)이며, 상호 통신을 통하여 각자의 위치, 속도에 대한 정보를 주고받을 수 있다고 가정한다. 제한조건으로서, 이동시 두 로봇은 경로를 이탈하지 않아야 하고 이동 속도는 특정 범위내의 값을 가져야 한다.

이러한 문제를 해결하기 위해 본 논문에서 제안하는 알고리즘은 기본적으로 앞서 언급한 "두 인부가 버팀목 나르는 방식"을 모사한 것으로 중앙 제어용 외부 컴퓨터가 없이 두 로봇의 자율적인 상호 통신에 의한 분산 제어 방식을 갖는다.

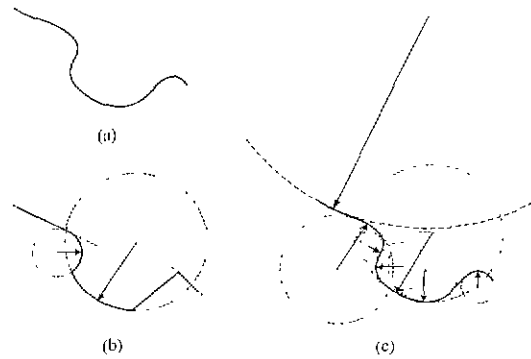


그림 1. 임의의 곡선에 대한 직선과 원호의 근사법. Fig. 1. Approximation of an arbitrary curve with arcs and lines.

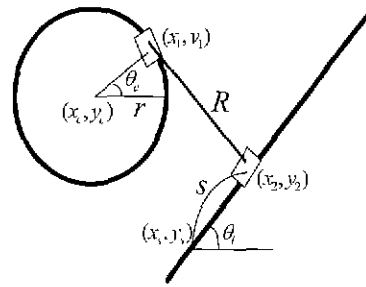


그림 2 직선과 원호 상에 위치하는 로봇들. Fig. 2. Robots on a line and a circle

III. 협동 운동계획 알고리즘

1. 부분 경로

본 논문에서 고려하는 전체경로는 출발점에서 목표까지 부분경로의 조합으로 이루어지고, 이러한 부분경로는 직선 또는 원호로만 표현된다고 가정한다

이와 같은 가정이 타당한 이유는 그림 1에서 보는 바와 같이 임의의 모양을 갖는 곡선에 대해서도 원호와 직선으로 근사화 할 수 있기 때문이다. 그림 1의 (a)는 임의의 곡선이고 그림 1의 (b)는 이를 두 개의 원호와 세 개의 직선으로 근사화 한 것이며 더 정밀도를 높이기 위해서는 그림 1의 (c)와 같이 보다 많은 수의 원호와 직선으로 나타낼 수 있다.

이러한 부분경로의 정의가 필요한 이유는 종 로봇은 사람들처럼 힘센서를 통하여 상대방의 위치나 속도를 알아내는 방식이 아니라, 통신을 통하여 주 로봇의 부분경로와 위치, 속도를 전달받은 후에 그 정보와 자신의 부분경로, 그리고 위치를 고려하여 타당한 종 로봇의 속도를 구해야하기 때문이다.

주-종 로봇이 위치한 각 부분경로의 형태에 따라 위치 및 속도관계가 달라지게 되므로, 다음에서는 본 논문에서 가정한 부분경로들의 조합인 직선-직선, 직선-원호, 원호-원호 세 가지 경우에 대한 로봇간의 위치 및 속도에 관한 관계식을 유도한다.

1) 직선과 원호

한 대의 로봇은 원호에 있고 다른 한대는 직선에 있을 때, 각각 로봇의 속도에 대한 관계는 다음과 같다.

그림 2에서  $x_c, y_c$ 와  $r$ 은 각각 원점의 좌표와 원호의 반지름을 나타내며,  $x_s, y_s$ 와  $\theta_s$ 은 직선을 표현하기 위한 변수로 각각 직선의 기준점과 직선의 기울기를 나타낸다. 또,  $R$ 은 운반해야하는 강체의 길이를 나타낸다.

이 경우 각 로봇의 위치는 다음과 같이 표현된다.

$$\begin{aligned} x_1 &= r \cos \theta_c + x_c \\ y_1 &= r \sin \theta_c + y_c \\ x_2 &= s \cos \theta_1 + x_s \\ y_2 &= s \sin \theta_1 + y_s \end{aligned} \quad (1)$$

여기서,  $s$ 는 직선 위의 기준점에서 로봇까지 거리를 나타내며,  $\theta_c$ 는 기준축(여기서는 양의  $x$ 축)에서 로봇까지의 각도를 나타낸다.

길이가  $R$ 인 강체를 운반해야 하므로 로봇들이 반드시 지켜야할 위치조건은 다음과 같다.

$$\begin{aligned} (r \cos \theta_c + x_c - s \cos \theta_1 - x_s)^2 \\ + (r \sin \theta_c + y_c - s \sin \theta_1 - y_s)^2 = R^2 \end{aligned} \quad (2)$$

(2)의 위치관계를 속도에 관한 식으로 변환시키기 위하여 시간에 관해 미분을 하고 정리를 하면 다음과 같은 식이 유도된다.

$$\begin{aligned} \omega_c \cdot \{ r \cos \theta_c \cdot (y_c - y_s) \\ + r \sin \theta_c \cdot (x_c - x_s) \} \\ = v_s \cdot \{ r \cos(\theta_1 - \theta_c) - s \\ + \cos \theta_1 \cdot (x_c - x_s) + \sin \theta_1 \cdot (y_c - y_s) \} \end{aligned} \quad (3)$$

(3)을 선속도만으로 표현하면 최종적으로 다음과 같은 식으로 표현된다.

$$\begin{aligned} v_c \cdot \{ \cos \theta_c \cdot (y_c - y_s) \\ + \sin \theta_c \cdot (x_c - x_s) \} \\ = v_s \cdot \{ r \cos(\theta_1 - \theta_c) \\ - s + \cos \theta_1 \cdot (x_c - x_s) + \sin \theta_1 \cdot (y_c - y_s) \} \end{aligned} \quad (4)$$

(4)를 이용하여 중심이  $(x_c, y_c)$ 이며 반지름이  $r$ 인 원호에  $(x_1, y_1)$ 의 위치에 주 로봇이 존재하며 속도가  $v_c$ 일 경우, 한 점  $(x_c, y_c)$ 를 지나는 기울기가  $\theta_1$ 인 직선위에  $(x_2, y_2)$ 의 위치에 있는 종 로봇의 속도는  $v_s$ 로 결정할 수 있다. 즉 이러한 속도 관계로 진행할 경우 두 로봇의 거리는  $R$ 로 일정하게 유지할 수 있다.

2) 직선과 직선

그림 3에서  $(x_{s1}, y_{s1}), (x_{s2}, y_{s2})$ 와  $\theta_1, \theta_2$ 는 미리 주어지는 부분경로 중 직선을 표현하기 위한 변수로 각각 기준점과 직선의 기울기를 나타낸다.

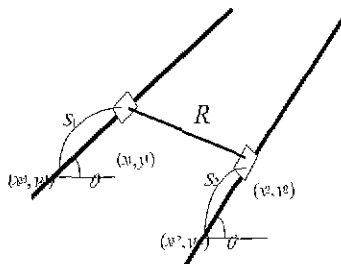


그림 3. 직선과 직선 상에 위치하는 로봇들.  
Fig. 3. Robots on two lines.

직선 위의 로봇의 위치는 다음과 같이 표현된다.

$$\begin{aligned} x_1 &= s_1 \cos \theta_1 + x_{s1} \\ y_1 &= s_1 \sin \theta_1 + y_{s1} \\ x_2 &= s_2 \cos \theta_2 + x_{s2} \\ y_2 &= s_2 \sin \theta_2 + y_{s2} \end{aligned} \quad (5)$$

로봇의 거리를 일정하게 유지하기 위하여 만족시켜야 할 조건은 다음과 같다.

$$\begin{aligned} (s_1 \cos \theta_1 + x_{s1} - s_2 \cos \theta_2 - x_{s2})^2 \\ + (s_1 \sin \theta_1 + y_{s1} - s_2 \sin \theta_2 - y_{s2})^2 = R^2 \end{aligned} \quad (6)$$

직선과 원호일 경우와 유사한 방법으로 양변을 시간에 관하여 미분하여 로봇의 선속도 관계식을 유도하면 다음과 같이 표현된다.

$$\begin{aligned} v_1 \{ s_1 - s_2 \cos(\theta_1 - \theta_2) \\ + \cos \theta_1 \cdot (x_{s1} - x_{s2}) + \sin \theta_1 \cdot (y_{s1} - y_{s2}) \} \\ = v_2 \{ -s_2 + s_1 \cos(\theta_1 - \theta_2) \\ + \cos \theta_2 \cdot (x_{s1} - x_{s2}) + \sin \theta_2 \cdot (y_{s1} - y_{s2}) \} \end{aligned} \quad (7)$$

3) 원호와 원호

원호 상에 위치하는 로봇의 위치는 다음과 같이 표현된다.

$$\begin{aligned} x_1 &= r_1 \cos \theta_1 + x_{c1} \\ y_1 &= r_1 \sin \theta_1 + y_{c1} \\ x_2 &= r_2 \cos \theta_2 + x_{c2} \\ y_2 &= r_2 \sin \theta_2 + y_{c2} \end{aligned} \quad (8)$$

만드시 지켜야하는 로봇간의 거리 조건은 다음과 같다.

$$\begin{aligned} (r_1 \cos \theta_1 + x_{c1} - r_2 \cos \theta_2 - x_{c2})^2 \\ + (r_1 \sin \theta_1 + y_{c1} - r_2 \sin \theta_2 - y_{c2})^2 = R^2 \end{aligned} \quad (9)$$

직선과 원호일 경우와 유사한 방법으로 양변을 시간에

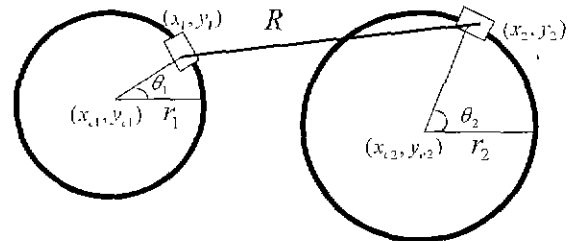


그림 4. 원호와 원호 상에 위치하는 로봇들.  
Fig. 4. Robots on circles

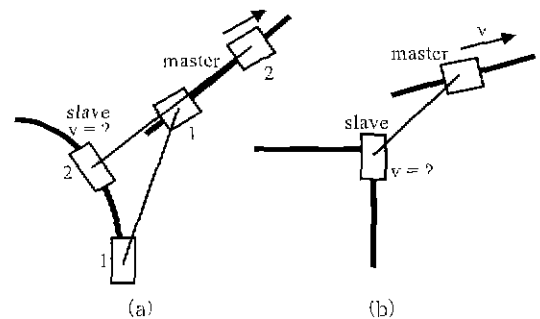


그림 5. 주종 로봇의 역할을 교체해야하는 상황 예.  
Fig. 5. Examples for the situation of master-slave switching.

관하여 미분하여 로봇의 선속도 관계식을 유도하면 다음과 같이 표현된다.

$$\begin{aligned} & v_1 \{ \cos \theta_1 \cdot (y_{c1} - y_{c2}) \\ & \quad - \sin \theta_1 \cdot (x_{c1} - x_{c2}) + r_2 \sin(\theta_1 - \theta_2) \} \\ & = v_2 \{ \cos \theta_2 \cdot (y_{c1} - y_{c2}) \\ & \quad - \sin \theta_2 \cdot (x_{c1} - x_{c2}) + r_1 \sin(\theta_1 - \theta_2) \} \end{aligned} \quad (10)$$

한편, 이와 같은 결과는 기계구조학에서 다루는 사절기구(four-bar mechanism)의 RRRP, PRRP, RRRR 형상을 통해서도 유도할 수 있다.

4) 부분경로 상에서 주종 로봇의 속도 설정

긴 강형 물체를 운반하는 두 대의 로봇은 서로간의 거리를 엄격히 지켜야만 한다. 이러한 제한조건을 만족하기 위하여 두 대의 로봇의 속도는 다음과 같은 방식으로 설정한다. 로봇 중에 임의의 한 대를 주 로봇으로 설정하고 속도를 임의로 정한다. 그리고 주종 로봇의 부분경로의 형태에 따라 위의 (4), (7), (10) 중 하나를 선택하여 주 로봇의 속도와 위치와 종 로봇의 위치에 따라 종 로봇의 속도를 결정한다.

2. 주종 로봇 역할 교체

제안한 알고리즘에서 주종 로봇의 역할 교체는 매우 중요하다. 긴 물체를 곡선의 경로로 운반할 경우 주 로봇의 임의의 속도에 대응하는 종 로봇의 속도를 상기한 관계식으로 구했을 경우 그 값이 최대 제한 속도보다 크게 될 수가 있다. 이러한 경우 주종 로봇의 역할을 교체하는 것이 필요한데, 이에 대해 그림 5와 같은 예를 통해 살펴보자. 그림 5의 (a)에서 주 로봇과 종 로봇이 1번 위치에 있을 경우에는 제안한 방법으로 주 로봇과 종 로봇의 속도를 구하면 양쪽 다 타당한 값을 구할 수 있으나 1번 위치에서 2번 위치로 로봇이 이동해감 따라 종 로봇의 속도는 주 로봇에 비해 아주 큰 값을 가져야만 하며, 주 로봇의 이동방향과 종 로봇의 이동방향이 수직이 되는 순간에는 (4)에서  $\theta_c = \theta_1$ 이고  $\frac{y_c - y_1}{x_c - x_1} = \tan \theta_c$

인 경우가 되어 주 로봇의 속도에 대응하는 종 로봇의 속도를 (4)로부터 구하면 무한대가 되어 물리적인 의미를 상실한다. 또, 그림 5의 (b)는 주종 로봇이 모두 직선으로 이루어진 부분경로의 경계점에 있는 경우로 주 로봇의 속도에 대응하는 종 로봇의 속도를 현재 위치인 경계점 좌측 부분경로로부터도 타당한 종 로봇의 속도를 구할 수도 있고 우측 부분경로로부터도 타당한 종 로봇의 속도를 구할 수도 있지만 구해진 속도에 의한 다음 위치는 경로 상에 존재하지 않는다.

이러한 두 가지 경우 중 한가지가 발생했을 때 각각의 로봇들의 주종 역할을 바꾸어 다시 속도를 찾으면 주종 양측에 모두 타당한 속도를 얻을 수 있다. 즉 그림 5의 (a)에서 종 로봇의 속도가 최대 제한 속도( $V_{max}$ ) 범위를 벗어나기 전에 주 로봇과 종 로봇의 역할을 교체하여 원호 상에 있는 로봇의 속도를 임의로 설정하고 그 속도에 따라 직선 상의 로봇의 속도를 설정한다. 이와 같이 주종 로봇의 역할이 바뀌어 2번 위치에 도달했을 경우 원호 위의 로봇이 주가 되고 직선 위의 로봇이 종

이 되며, 이 때 속도 관계식으로부터 구한 주 로봇의 속도에 따른 종 로봇의 속도는 거의 0이 되어 양측 모두가 물리적으로 실현 가능한 속도를 가지게 된다. 그림 5의 (b)의 경우에도 주종 로봇의 역할을 바꾸어 속도를 다시 설정하면 양측에 모두 타당한 값을 가지게 된다.

이와 같은 주종 역할 교체를 통하여 긴 물체를 주어진 곡선 경로를 따라서 이동시키는데 있어 위의 예에서 보듯이 상황에 따라 주 로봇이 종 로봇을 이끌고 진행하거나 뒤에서 밀고 가기도 한다. 따라서, 두 로봇은 주어진 협동작업을 수행함에 있어 전진 운동만이 아닌 때론 후진을 하기도 해야한다.

3. 전체적인 운동계획 알고리즘

본 절에서 서술하는 운동계획의 흐름은 두 로봇 전체를 총괄한 제어 흐름이다. 즉 외부에서 작업을 수행하는 로봇들을 바라볼 때 각각의 로봇이 작업을 수행하는 순서를 의미한다.

단계 1 : 임의로 한 로봇에게 주 역할을, 다른 한 로봇에게는 종 역할을 하도록 설정한다. 두 로봇의 위치를 초기화하고 주 로봇의 속도를 임의로 설정한다.

단계 2 : 각각의 로봇은 미리 주어진 지도 정보와 현재 위치에 따른 개별적인 부분경로를 설정한다.

단계 3 : 주 로봇의 부분경로와 속도, 위치에 따라 위에서 제시한 각각의 로봇간의 속도의 관계에 따라 현재 종 로봇의 부분경로에서의 속도를 계산한다.

단계 4 : 계산된 종 로봇속도의 크기가 최대 속도( $V_{max}$ )보다 크거나 계산된 종 로봇의 위치가 현재 부분경로를 이탈해야만 하는 값인지 체크하여 종 로봇의 속도와 위치의 타당성을 검사한다.

단계 5 : 종 로봇의 속도가 타당하지 못할 경우 종 로봇이 주 로봇에게 주와 종의 역할 교체를 요구하는 신호를 보낸 후 이번에는 이전에 종이었던 로봇이 주가 되어 속도를 임의로 설정하고 이전에 주었던 로봇이 종이 되어 로봇간의 부분경로 및 위치 관계에 따라 속도를 다시 계산하여 설정한다.

단계 6 : 만일 종 로봇의 속도와 위치가 타당할 경우 미리 설정된 시간( $\Delta t$ )동안 주 로봇과 종 로봇 각각 자신의 속도로 진행할 경우 다음 위치를 계산한다.

단계 7 :  $\Delta t$  이후의 위치가 경로를 벗어나게 될 경우  $\Delta t$  를 감소시켜 다음 부분경로의 시작까지만 로봇이 진행되도록 하고, 그렇지 않을 경우는  $\Delta t$  동안 정해진 속도로 진행하고 위치값을 갱신한다.

단계 8 : 로봇의 위치가 최종 목표에 도달했는지 검사하여 최종 목표에 도달했으면 종료하고 그렇지 않으면 단계 2부터 다시 시작한다.

4. 각각의 로봇에 대한 알고리즘 구현

상기한 3절에서는 협동을 하는 두 로봇 전체에 대한 운동계획의 흐름에 중점을 두어 설명을 했다. 그러나, 실제 구현을 위해서는 두 로봇 전체에 대한 협동 알고리즘은 필요치 않고 각각의 로봇에 대한 분산적인 제어 알고리즘이 필요하다.

제안된 각 로봇의 제어 방법은 메시지 구동 방식을

기반으로 한다. 즉 특정 메시지가 도착하면 그 메시지에 해당하는 작업을 수행한 후 다시 메시지를 기다리는 것을 반복하는 방식으로 상대방 로봇과의 상호 통신을 하면서 협동작업을 수행해 나간다. 이러한 메시지 구동 방식에 대해 자세히 살펴보면 다음과 같다.

1) 전환 메시지(Switch Message)가 발생하지 않았을 때

그림 6의 (a)에서 주 로봇에게 출발 메시지(Start Message)가 주어지면 알고리즘이 시작된다. 출발 메시지를 받은 주 로봇은 임의의 속도를 설정한 후에 종 로봇에게 주 로봇의 속도와 위치, 부분경로를 속도 메시지(Velocity Message)에 실어서 보낸다. 속도 메시지를 받은 종 로봇은 속도 메시지에 함께 실려온 주 로봇의 정보를 가지고 자신의 속도를 결정한다. 이때 결정된 속도가 타당하기를 검사하여 타당하면 결정된 속도로 진행할 시간 간격을 실행시간 메시지(Runtime Message)에 실어서 보낸다. 실행시간 메시지를 받은 주 로봇은 실행시간 메시지에 실려온 진행시간만큼 진행을 한 후에 종 로봇에게 종료 메시지(Done Message)를 보낸다. 종 로봇은 진행시간만큼 진행을 하고 종료 메시지를 주 로봇에게 받으면 출발 메시지를 주 로봇에게 보낸다.

2) 전환 메시지가 발생했을 때

그림 6 (b)의 경우는 (a)와 달리 주 로봇의 속도 메시지를 받고 종 로봇의 속도를 결정했을 때 결정된 속도가 타당하지 않은 경우로서, 종 로봇 쪽에서 주 로봇 쪽으로 전환 메시지(Switch Message)에 자신이 원하는 속도와 현재 위치, 부분경로를 실어서 보낸다. 전환 메시지를 받은 주 로봇은 종으로 역할이 바뀌면서 전환 메시지에 실려온 정보를 가지고 자신의 속도를 계산해 설정한다. 그리고는 실행시간 메시지를 발생시킨다. 이 이후로는 단지 역할만 바뀌었을 뿐 전환 메시지가 발생하지 않았을 경우와 완전히 동일하다.

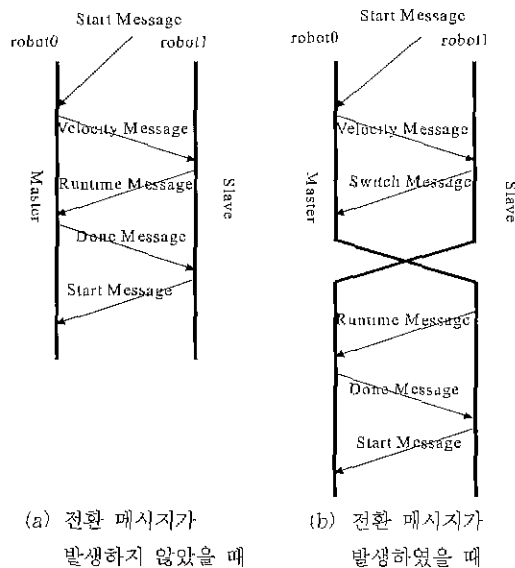


그림 6. 주종 로봇간의 메시지 전달.

Fig. 6. Message transmission between masterslave robots.

#### IV. 모의실험

##### 1. 모의실험용 프로그램 (Cobot)

제안된 제어 알고리즘의 타당성을 검증하기 위하여 협동로봇과 작업환경을 Visual C++를 사용하여 소프트웨어적으로 구현하였다. "Cobot"이라 불리는 이 프로그램의 전체적인 모습은 그림 7과 같다.

그림 7에서 좌측 에디터 윈도우에 원하는 형태의 경로와 로봇의 초기위치를 정해진 형식에 따라 텍스트로 편집(edit)하여 리셋(reset)버튼을 누르면 우측 그래픽 화면에 원하는 형태의 경로가 그려지고, 로봇들이 출발점에 위치한다. 그 후 출발신호에 따라 제안한 알고리즘에 의해서 로봇들이 긴 물체를 경로에 따라 운반하는 모습과 로봇의 속도 및 위치 등에 관한 여러 가지 상태 정보를 보여주게 된다.

Cobot 프로그램에서 분산식 로봇 제어의 특징을 충분히 나타내기 위하여 각각의 소프트웨어 로봇을 구현한 모듈은 독립적인 보조 쓰레드(thread)이며 메인 쓰레드는 각각의 소프트웨어 로봇을 제어하기보다는 단지 소프트웨어 로봇이 보내주는 정보를 화면에 보여주는 기능만 가지고 있을 뿐이다. 모든 소프트웨어 로봇들은 동형이며 하드웨어 로봇들이 양방향 통신을 하는 것과 동일하게 소프트웨어적인 로봇들은 윈도우 환경에서 상호 메시지를 주고받음으로써 양방향 통신을 한다.

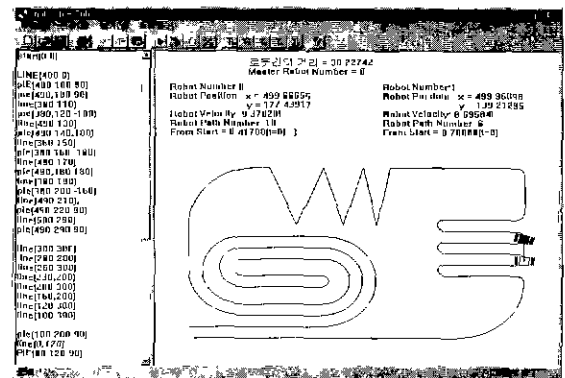


그림 7. Cobot 프로그램.

Fig. 7. Cobot program

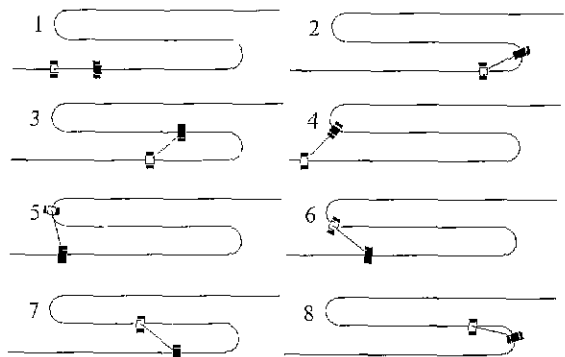


그림 8. S자형 경로를 따라 이동하는 과정

Fig. 8. Process of the movements tracking the S-typed path.

2. 모의실험 결과

모의실험을 위한 경로와 각각 로봇에 대한 설정값은 다음과 같다.

- 선행로봇의 초기 위치 : (80cm, 0cm)
- 선행로봇의 초기 속도 : 8cm/s
- 후행로봇의 초기 위치 : (0cm, 0cm)
- 후행로봇의 초기 속도 : 8cm/s
- 각 로봇의 최대 속도 : 10cm/s
- 강체 운반물의 길이( $R$ ) : 80cm
- 초기 주 로봇 : 선행로봇

먼저 그림 8과 같은 S자형의 경로가 주어질 경우, 모의 실험 결과를 부분적으로 자세히 살펴보자. 그림 8은 S자 경로를 따라 이동하는 모습을 진행 순서에 따라 나열한 것이다. 여기서, 주 로봇은 흑색으로, 종 로봇은 백색으로 각각 표시한다.

그림 8의 각 단계에서 두 로봇은 표 1에서 보는 바와 같이 전진과 후진을 반복하고 역할을 바꾸어 가며 곡선의 경로를 수행한다. 표 1과 그림 8을 함께 살펴보면 4번과 5번 사이에서 한번의 주종 로봇의 역할 교체가 일어났음을 알 수 있다. 이 사실을 그림 8에서 살펴보면 4번과 5번 사이에서 그림 5의 (a)번과 같은 경우가 발생되어 종 로봇의 속도가 제한 속도 범위를 벗어난 값을 가져야 하므로 주종 역할 교체가 이루어졌음을 알 수 있다. 그림 8 이후의 진행은 1번에서 8번까지 진행의 역으로 생각하면 유사한 경우가 되므로 상세한 진행 과정에 대한 설명은 생략한다.

그림 8에서 주어진 경로는 3개의 직선들과 2개의 원호들로 이루어져 있으며 제일 하단 좌측을 출발점으로 하고 제일 상단 우측을 목표점으로 한다. 이들 5개의 부분경로에 대한 구체적인 파라미터를 표 2와 같이 설정한 경우, S자형으로 주어진 경로를 이동할 때의 각 로봇의 속도는 그림 9와 같다. 이 경우 전후진을 반복하며 경로를 통과함을 알 수 있으며, 주종 역할 변환은 7번 발생하였다.

표 1. 각 단계에서 로봇의 역할 및 진행 방향.

Table 1. The role of robots and direction of movement at each step

순서	선행 로봇		후행 로봇	
	역할	속도의 방향	역할	속도의 방향
1	주	전진	종	전진
2	주	전진	종	전진
3	주	전진	종	후진
4	주	전진	종	전진
5	종	후진	주	전진
6	종	후진	주	전진
7	종	후진	주	전진
8	종	후진	주	전진

표 2. 각 부분경로의 파라미터.

Table 2. Parameters of each partial paths.

번호 (시작위 치부터)	모양	기준점 ( $x_p, y_p$ ) 또는 중심 ( $x_c, y_c$ )	기울기 ( $\theta_p$ ) 또는 원호각 ( $\theta_c$ )
1	직선	(0cm, 0cm)	0
2	원호	(400cm, 25cm)	$\pi$
3	직선	(400cm, 50cm)	$-\pi$
4	원호	(100cm, 75cm)	$-\pi$
5	직선	(100cm, 100cm)	0

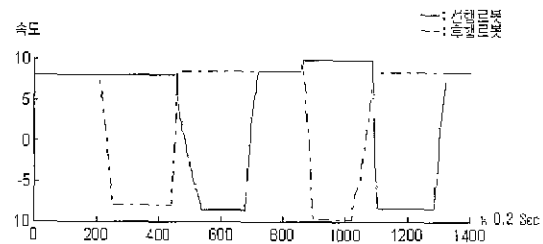


그림 9. S자형 경로를 통과하는 과정에서의 로봇들의 속도 곡선.

Fig. 9. Velocity curves in the process of passing through the S-typed path.

다음으로 다양한 경우를 포함한 비교적 복잡한 그림 10과 같은 경로에 대한 모의실험을 수행하였다. 그림 10에서 짙은 색의 로봇이 주 로봇을 의미한다. 출발점에서는 선행 로봇이 주 로봇이었으나 반복적인 주종 역할 교체를 통하여 두 로봇이 일정 길이의 강체를 주어진 복잡한 경로를 따라 잘 운반함을 알 수 있다.

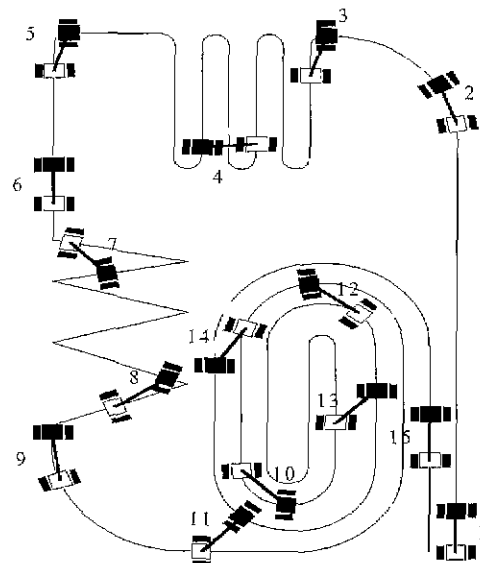


그림 10. 복잡한 경로에 대한 모의실험 결과.

Fig. 10. Simulation result for a complicated path.

그림 10의 최종 모의실험에서 경로와 각각 로봇에 대한 설정값은 다음과 같다.

- 선행로봇의 초기 위치 : (0cm, 40cm)
- 선행로봇의 초기 속도 : 8cm/s
- 후행로봇의 초기 위치 : (0cm, 0cm)
- 후행로봇의 초기 속도 : 8cm/s
- 각 로봇의 최대 속도 : 10cm/s
- 강체 운반물의 길이( $R$ ) : 40cm/s
- 초기 주 로봇 : 선행로봇
- 경로가 그려진 전체 영역의 크기 :  
400cm × 500cm
- 전체 부분경로의 개수 : 43개

3. 로봇간의 거리오차

제안한 알고리즘이 이상적으로 수행되기 위해선 주 로봇의 속도가 연속적으로 변함에 따라 종 로봇의 속도도 연속적으로 변화를 해야만 한다. 하지만 통신속도의 한계 때문에 종 로봇이 주 로봇의 상태를 무한히 빠른 속도로 읽어 들일 수는 없다. 따라서 통신에서 받아들인 주 로봇의 정보를 이용하여 종 로봇의 속도를 구하고 종 로봇은 미리 설정한  $\Delta t$  동안 주 로봇의 속도가 일정하다고 가정하고  $\Delta t$  동안 진행하게 된다. 그러므로 반드시 유지해야 하는 로봇간의 거리에 오차가 발생하게 된다.

이러한 오차를 줄이기 위해서는  $\Delta t$ 를 줄이거나 로봇의 속도를 줄여야만 한다.  $\Delta t$ 는 로봇의 통신시간이나 계산시간에 의해 제한되므로 줄이는데 한계가 있다. 따라서 거리오차를 허용한계 이하로 줄이기 위해서는 주 로봇의 속도와 종 로봇의 속도를 감소하여야만 한다.

V. 결론

본 논문에서는 임의의 곡선 경로를 통하여 두 대의 로봇이 긴 강체를 옮기는 문제를 분산제어 방법으로 해결하였다. 기존의 연구에서는 두 대 이상의 로봇 제어는 주로 각각의 로봇이 서로 강하게 결합되어 있지 않은, 즉 각각의 로봇은 서로 움직이는 장애물 이상의 의미가 없는 경우에 대해 주로 연구 되어왔다. 반면 본 논문에서는 긴 강체를 두 대의 로봇이 협동하여 옮기는 문제와

같은 강한 결합조건을 갖는 로봇의 협동 문제를 설정하고, 제시한 협동작업을 적절히 수행하기 위한 알고리즘을 제안하였다. 구체적으로는 각 부분경로 조합에 대한 두 대의 로봇간의 속도 관계식의 유도와 새롭게 제안한 주종 역할 교체 방법을 근간으로 하는 협동 운동계획 알고리즘을 제안하였다. 제안된 제어 방법은 일반적인 사람들의 협동방식의 특징인 상호 통신과 주위 상태에 따라 자신의 행동방식을 결정하는 특성을 모사한 방법이다. 제안된 알고리즘의 타당성은 모의실험용으로 개발된 Cobot 프로그램을 이용하여 다양한 경로에 대하여 검증하였다.

현재까지의 연구에서는 단지 두 로봇의 속도 결정에 주안점을 두었으나, 앞으로는 가속도 특성을 함께 고려한 협동 운동계획 알고리즘에 대해 계속 연구하고자 한다.

참고문헌

- [1] R. Alami, F. Robert, F. Ingrand, and S. Suzuki, "Multi-robot cooperation through incremental planning," *IEEE Conference on Robotics and Automation*, vol. 3, pp. 2573-2579, 1995
- [2] R. Alami, and M. Khatib et al., "A general framework for multi-robot cooperation and its implementation on a set of three hilare robots," *The Fourth International Symposium on Experimental Robotics*, ISER '95 Stanford, California, June 30-July 2, 1995
- [3] 김도윤, 정명진, "동적 환경에서 강화학습을 이용한 다중 이동 로봇의 제어," *Proc. of the 11th KACC*, vol. 2, pp 944-947, October, 1996.
- [4] T. Balch and R. C. Arkan, "Communication in reactive multiagent robotic systems," *Technical Report GIT-CC-95-47*, Georgia Tech. College of Computing, pp. 1-25, 1994.
- [5] Y. J. Kanayama, "Rigid body motion analysis towards rotary vehicle," *IEEE Trans. Robotics and Automation*, vol. 13, no. 1, February, 1997.
- [6] J. S. Bay, "Design of the army-ant cooperative lifting robot," *IEEE Robot Automat. Mag.*, vol. 2, no. 1, March 1995.



이 승 환

1972년 4월 27일생. 1995년 경북대학교 전자공학과 졸업. 1998년 동대학원 석사, 1998년-현재 LG전자 생산기술원 연구원, 관심분야는 제어시스템 프로그래밍.



이 승 하

1966년 7월 30일생. 1988년 경북대학교 전자공학과 졸업. 1990년 한국과학기술원 전기 및 전자공학과 석사, 1995년 한국과학기술원 전기 및 전자공학과 박사, 1995년-1996년 한국과학기술원 전기 및 전자공학과 위촉연구원. 1996년-현재 경북대학교 전자전기공학부 국책계약교수. 관심분야는 자동화시스템, 퍼지시스템, 지능제어, 로봇틱스.

이 연 정

제어·자동화·시스템공학 논문지 제 2권, 제 1호, 참조.