공학 연구논문집 제25권 제2호 pp. 113~125, 울산대학교
Journal of Engineering Research Vol. 25, No. 2, pp. 113~125, 1994.
University of Ulsan.

대규모 자유주행방식 AGV 시스템을 위한 중앙운행제어

盧瑩植
電氣工學科

<요 약>

본 논문에서는 다수의 자유주행방식 AGV들을 대규모 물류자동화 시스템에 사용하는 경우 중앙운행제어에 필요한 경로계획 및 교통제어 알고리즘을 제시하였다. 중앙제어 시스템은 사전에 정의된 경로망 정보를 이용하여 작업장소에 이르는 최적 경로를 A* 알고리즘을 이용하여 탐색하고 탐색 결과를 최적 AGV 선정과 최적 경로 주행 명령어 구성에 이용한다. 이때 특정 직선경로 주행시 소요되는 시간에 영향을 주는 여러 요소를 고려하여 최소 시간 경로를 얻기 위한 평가지표를 제안하였다. 또한 경로망을 적절히 구성하여 매우 복잡한 교통환경에서 효율적으로 사용할 수 있는 교통통제 방식을 제안하고 이를 위한 제어 알고리즘을 개발하였다. 개발된 알고리즘을 자동차 차체 조립공장의 AGV 시스템에 적용하여 시뮬레이션 함으로써 검증하였다.

A Central Navigation Control for Large-Scale Free-Ranging AGV Systems

Ro, Young-Shick
Dept. of Electrical Eng

<Abstract>

In this paper, a path-planning and traffic control algorithm for a central navigation control is proposed which can be accepted in large-scale material handing automation systems using free-ranging AGVs. The central navigation control system has the predefined vehicle path network, and can searches for an optimal route to the goal station by A* algorithm, which be used to the optimal vehicle selection and to its routing commands. A heuristic evaluation function is

proposed for minimum time path-planing, which include various elements to effect the estimation of driving-time in a specified line path. Also, some useful traffic control methods is propsed for the complex environments, and its control algorithm is developed using the zone-blocking and the path blocking approachs. Computer simulation of a material handling system in the car-body assembly shop is conducted to demonstrate the acceptibility of the proposed control concepts in large-scale AGV system.

1. 서 론

AGV(Automatic Guided Vehicle)는 FA(Flexible Automation)를 위한 효과적인 운송수단으로 여러 분야에서 사용되고 있다. 그러나 기존의 AGV는 바닥에 설치된(유도선, 광테이프, 등) 연속적인 주행경로에 의존하여 운행하여야 하므로 적용이 제한되고, 일단 주행경로가 설치되면 경로의 변경이 어렵고 시간과 경비가 소요되는 문제가 있다. 따라서 기존의 AGV로 적용이 어려웠던 복잡한 주행환경에서 보다 유연한 물류 자동화를 실현하기 위하여 설치된 주행경로 없이 이동할 수 있는 자유주행방식 AGV의 사용이 적극 검토되고 있다[1-2].

한편 자유주행방식 AGV를 이용하여 대규모 물류자동화 시스템을 구축하는 경우 중앙운행제어의 역활이 대단히 중요하다. 대규모 물류자동화 시스템에서는 작업공간이 광범위하고 다수의 AGV를 요구하나 정해진 이동통로와 작업장소에 정해진 시간내에 운반작업을 완료해야 하므로 이동경로가 매우 복잡하다. 따라서 개개의 AGV의 자율성과 지능을 향상시키는 것보다 전체 AGV들을 통제하는 중앙운행제어 시스템의 기능을 높히고 AGV는 단순한 주행기능만 갖도록 하는 것이 유연성, 경제성, 신뢰성, 등의 면에서 효율적이다. 이러한 경우 AGV 시스템의 성능은 AGV 자체의 성능보다 중앙운행제어 시스템의 성능에 의존한다고 볼 수 있다.

작업환경 정보를 사전에 알고있는 경우 자유주행 AGV(혹은 Mobile Robot)의 주행경로 결정에 대한 연구[3-5]와 다수의 AGV들의 상호 충돌 회피에 대한 연구[6-7]는 많이 진행되어 왔다. 그러나 물류자동화 시스템에서 AGV의 주행경로는 한정되어 있으며 일반적으로 생산라인 사이의 직선통로로 한 작업지점에서 다른 작업지점으로 주행한다. 또한 AGV가 대기장소에서 대기 중이거나 혹은 작업을 종료하고 대기장소로 이동하는 중에 호출된다. 이때 최단시간에 이동할 수 있는 주행경로를 사전에 마련하거나 실시간으로 탐색하는 것은 대규모 작업환경에서 매우 어렵다. 또한 다수의 AGV가 한정된 통로를 같이 사용하는 경우 AGV 자체의 지능을 이용하여 교통장애 문제를 해결하는 것은 거의 불가능하다. 이러한 문제를 효과적으로 해결하는 방법은 사전에 AGV의 주행 가능한 경로를 정의하는 것이다. 이러한 사전에 정의된 경로를 이용하여 현재의 위치에서 목표 작업지점에 이르는 최적경로를 실시간으로 탐색하는 문제와 다수의 AGV가 원활하게 계획된 경로를 주행하기 위한 교통제어 문제는 적절한 중앙운행제어 시스템에 의하여 비교적 쉽게 해결될 수 있다. Yamashita[8] 및 Crowley [9] 는 통로 양 끝을 노드로 사전에 정의하고 노드망을 구성하여 경로계획하는 방법을 제안하였다. 한편 Kanayama와 Yuta는 연속적인 직선경로에 의한 주행경로 표현법

을 제시하였고, 이를 이용한 이동명령어 시스템에 대하여 연구하였다[10-14].

본 논문은 대규모 물류자동화 시스템에서 자유주행방식 AGV를 효과적으로 관리하기 위하여 중앙운행제어에 의한 실시간 경로계획과 교통제어에 대해 다룬다. 이때 자유주행방식 AGV는 언제 어느 위치에서나 작업공간상의 위치를 인식가능하고 중앙운행제어 시스템과 통신가능하다고 가정한다. 자유주행방식 AGV의 주행경로는 직선경로들을 상호 연결한 경로망을 이용하여 사전에 정의하고, 경로망내의 AGV의 현위치에서 작업위치에 이르는 최소 시간 경로를 찾기 위한 평가지표를 제시한다. 제시된 평가지표를 이용하여 최적 경로 탐색을 수행하고 이를 특정 운반작업을 위한 최적 AGV 선정과 AGV를 최적경로를 따라 주행하도록 하는 이동명령어 구성에 사용한다. 경로망의 합류점에서의 교통제어 방법으로는 비교적 단순한 형태의 교통환경에 손쉽게 적용할 수 있는 구역통제 방식과 복잡한 교통환경에서 교통장애나 불필요한 교통통제를 피할 수 있는 경로통제 방식을 제안하고 제어알고리즘을 제시한다. 경로통제의 경우 대규모 AGV 시스템에서 흔히 요구되는 교통제어 문제에 대한 두가지 예를 소개하고 구체적인 방법론을 제시한다. 제안된 중앙운행제어 시스템의 타당성을 검증하기 위하여 자유주행방식 AGV의 사용이 검토되고 있는 한 물류자동화 시스템에 대하여 컴퓨터 시뮬레이션한다.

2. 경로계획

2-1. 경로 표현 및 경로망

자유주행방식 AGV는 공간상의 자신의 위치를 항상 알고 있으므로 AGV를 직선경로를 따라 주행시키기 위해서는 경로좌표계와 경로길이에 대한 정보를 전달해 주는것으로 충분하다. 이를 위하여 본 연구에서는 단일 직선경로를 경로의 시작점의 위치 (x, y), 경로의 방향 θ, 및 경로의 길이 d로 표현하고, 이러한 직선경로를 따라 AGV를 주행시키기 위해서 다음과 같은 이동명령어를 이용하였다.

move(x, y, θ, d);

AGV를 원하는 주행경로를 따라 목표지점까지 이동시키기 위해서는 연속적인 직선경로 이동명령을 주는 것이 필요하다. AGV는 이동명령을 FIFO(First In First Out) 명령버퍼에 저장하고 자체에 내장된 직선경로 추종 동작제어 알고리즘에 의하여 이동명령이 전달된 순서에 따라 직선경로를 주행한다. 이때 경로종점에 이르기 전 일정한 위치(경로 변환점)에서 다음 직선경로를 따라 주행하고 만약 다음에 정의된 경로가 없는 경우는 현 경로의 종점에 정지한다. 그림 1은 연속적인 직선경로 정보(x1, x2, x3, x4, x5)를 이용하여 우회전, 좌회전, U-turn, 경로 변경, 등의 다양한 동작을 보여주는 예이다.

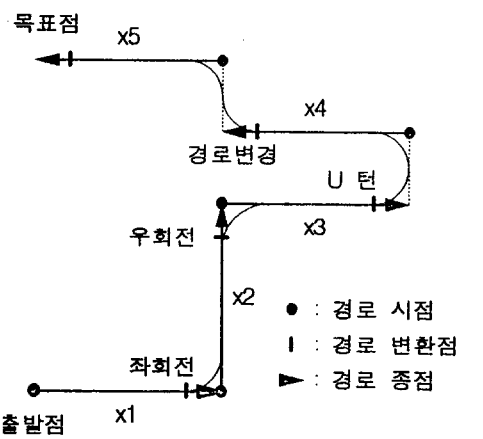


그림 1. 연속적인 직선경로 및 AGV 주행경로

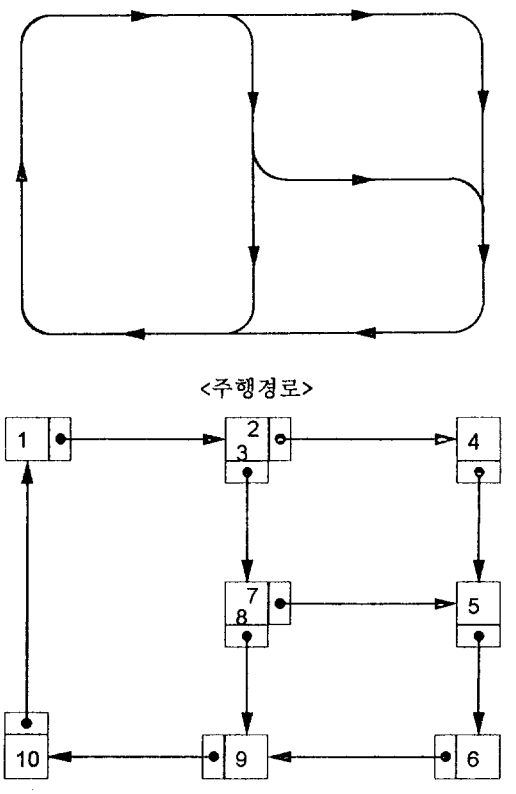
직선경로 정보들을 이용하여 사전에 AGV의 주행경로를 정의하고 출발점에서 목표점에 이르는 최적의 경로를 찾아 AGV를 주

행시키기 위해서는 연속주행이 가능한 직선 경로 정보들을 상호 연결한 경로망의 구성이 필요하다. 여기서는 직선경로 정보의 데이타 구조를 다음과 같이 정의하여 직선경로를 표현하고 경로망을 구성할 수 있도록 하였다.

```
typedef structure {
int name;
flot x, y, θ, d;
PATH* successor[];
} PATH;
```

여기서 name은 각 직선경로에 부여된 고유의 번호이고, successor는 한 직선경로에 연결된 다른 직선경로들을 나타낸다.

그림 2는 사전에 정의된 AGV의 주행경로와 이를 직선경로 정보를 이용하여 표현한 경로망의 구성의 예를 보여준다.



<주행경로>

<경로망>

그림 2. AGV 주행경로 및 경로망

2-2. 최적 경로 및 AGV 선정

AGV는 항상 하나의 직선경로상에 위치하고 AGV의 이동 목표지점인 작업장소(station)도 다른 직선경로상에 위치하므로 AGV를 현재의 위치에서 작업장소까지 이동시키기 위한 주행경로를 탐색하는 문제는 사전에 정의된 경로망내의 시작경로 S에서 목표경로 G에 이르는 직선경로들의 리스트를 찾는 문제가 된다. S에서 G에 이르는 방법은 여러개 존재할 수 있으며 이중 최적 경로를 효과적으로 찾기 위하여 최소 비용 그래프 탐색 알고리즘(mininum cost graph search algorithm)으로 잘 알려진 A* 알고리즘을 사용하였다[15]. 여기서 AGV의 최적 경로라 함은 작업장소까지 최소의 시간에 도달할 수 있는 주행경로를 말한다.

탐색 대상인 직선경로 C 에 대한 평가지표 f를 다음과 같이 선정한다.

$$f(C)=g(C)+h(C) \qquad (2\text{-}1)$$

여기서 g(C)는 시작경로 S에서 출발하여 직선경로 C를 주행완료 하는데 예상되는 시간 이고, h(C)는 C에서 목표경로 G에 도착하기 위하여 필요한 시간에 대한 경험적 예상치를 표시한다.

시작경로에서 한 직선경로를 지나는 데 걸리는 시간은 경유한 경로길이, 경로의 혼잡도, 경로의 연결형태 등에 관계되고 AGV의 직선경로 주행은 경로변환 및 경로주행으로 구성되므로 식(2-1)의 예상시간 g(C)를 다음과 같이 정의한다.

$$g(C)=g(P)+trans(P, C)+travel(C) \qquad (2\text{-}2)$$

여기서 g(P)는 시작경로 S에서 C 이전의 직선경로 P를 주행하는 예상되는 시간이고, trans(P, C)는 P에서 C로 경로전환에 따른 추가 소요 시간이며, travel(C)는 C를 주행하여 경로변환점에 이르는 데 걸리는 예상시간이다. 식(2-2)의 trans(P, C)는 다시 다음과 같이 정의한다.

trans(P, C)={dist(P, C)+αangle (P, C)}/vel (2-3)

여기서 dist(P, C)는 P의 종점과 C의 시점사이의 거리이고, angle(P, C)는 P와 C 사이의 연결각도이며, α는 이에대한 하중을 나타내는 상수이다. 또한 vel은 AGV의 디폴트 속도를 나타낸다. 식(2-2)의 travel(C)는 다음과 같이 정의한다.

travel(C)=length(C)/vel+traffic (C) (2-4)

여기서 length(C)는 직선경로 C의 경로길이 이고, traffic(C)는 C를 주행하는 동안 발생할 수 있는 교통 장애에 의한 손실시간을 나타낸다. 일반적으로 AGV가 주행하는 경로는 작업자, 전동차, 등과 공유하여 사용하므로 특히 AGV가 피해야할 혼잡한 경로는 traffic(C)를 크게 선정함으로써 다른 경로로 우회하도록 유도할 수 있다. 식(2-1)의 평가지표 f(C)에서 h(C)는 C의 종점과 G의 시점의 직선주행 시간을 계산하여 구한다. 이 시간은 C에서 G에 이르는 실제의 최적경로를 주행하는 시간보다 항상 적으므로 A* 알고리즘의 경험적 예상치로 사용 가능하다.

이상의 평가지표를 A* 알고리즘에 적용하면 AGV의 현위치에서 작업지점까지 최소시간으로 도달할 수 있는 경유 직선 경로들의 리스트와 예상시간을 얻을 수 있다. 대부분의 AGV 시스템에서 작업공간 내에 다수의 AGV가 존재하고 한 작업이 특정 AGV에만 부여되지 않으므로 필요한 작업을 최소 시간에 수행할 수 있는 AGV를 찾는 문제가 발생된다. 이는 현재 작업대기 장소에서 대기중에 있거나 작업을 완료하고 대기장소로 돌아가는 AGV들에 대하여 최적경로와 예상시간을 구하여 이중 예상시간이 최소인 AGV를 선택함으로써 쉽게 해결할 수 있다.

3. 교통제어

3-1. 구역 통제

경로망은 다수의 AGV에 의해 공유되므로 한 직선경로에 두개 이상의 직선경로가 연결되어 있는 지점(합류점)에서 AGV들의 상호 충돌 문제가 발생될 수 있다. 이러한 합류점에서의 교통 문제는 AGV가 지니고 있는 충돌방지 기능(전방 감지 센서, 범퍼, 등)이나 지적인 동적 장애물 회피 알고리즘에 의하여 해결할 수 없는 경우가 많으므로 중앙운행제어 시스템에서 적절한 교통제어를 하는 것이 바람직하다.

가장 간단하면서 많이 사용되온 접근방식은 구역통제이다. 이 방식은 합류점 주위의 일정한 크기의 구역을 교통통제 구역으로 정하고 이 구역내에는 단지 하나의 AGV만 주행이 허용되도록 한다. 기존의 AGV 시스템에서는 AGV의 진입, 통과, 통제, 등을 위하여 지상에 각종 센서 및 통신점을 설치하였으나 자유주행방식 AGV 시스템은 중앙운행제어 시스템에서 실시간으로 모든 AGV의 위치를 감시할 수 있고 경로망에 대한 정보를 사전에 지니고 있으므로 구역통제를 소프트웨어에 의하여 쉽게 실현할 수 있다.

합류점에서 구역통제를 위한 데이타 구조는 다음과 같다.

```
typedef structure {
PATH* input[];
PATH* output[];
int act_v;
} ZONE_BLOCK;
```

여기서 input은 통제구역으로 들어오는 직선경로들을 표시하고, output은 통제구역을 빠져나가는 직선경로를 표시하며, act_v는 현재 통제구역을 점령하고 있는 AGV를 나타낸다.

교통제어를 위하여 AGV는 중앙운행제어 시스템의 진입통제 명령에 의하여 일시적으

로 정지하고 통제해제 명령으로 정지된 운행을 계속하는 것으로 가정하였다. 구역통제를 위한 제어 알고리즘은 다음과 같다.

1) i 번째 ZONE_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있으면 그 AGV가 통제구역 내에 있는지 조사하여 통제구역을 벗어났으면 act_v의 내용을 제거하여 점령상태를 해제하고 다른 AGV의 진입을 허용한다.

2) i 번째 ZONE_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있지 않으면 입력구역에 있는 AGV들 중에 정해진 우선순위에 따라 act_v에 등록하여 통제구역을 점령시킨다.

3) i 번째 ZONE_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있으면 현재 구역을 점령하고 있는 AGV 이외의 입력구역에 있는 AGV들을 진입통제한다.

4) 1),2),3)의 과정을 모든 ZONE_BLOCK에 대하여 수행한다.

그림 3은 입력경로가 둘이고 출력경로가 둘인 경우 구역통제를 보여준다. 한편 그림 4는 입력경로 및 출력경로가 둘인 두개의 통제구역이 인접해 있는 경우로 구역통제에 의한 교통제어를 할 수 없는 하나의 예를 보여준다. 이 경우 a1, b1에서 입력된 서로 다른 AGV에 의하여 통제구역 A 및 B가 각각 점령되고 점령한 AGV의 경로가 각각 a3, b3 일때 두 AGV는 서로를 진입통제하여 정체가 발생된다. 만약 두 통제구역을 합하여 하나의 구역으로 설정하는 경우 이러한 정체는 방지할 수 있으나 불필요한 진입통제가 발생된다. 즉 a2로 진행하는 AGV에 의하여 b2로 진행하는 AGV를 불필요하게 정지 시키게 된다. 일반적으로 출력경로가 둘 이상이고 두개 이상의 통제구역이 근접해 있는 경우 AGV의 진행 경로에 따라 교통제어를 하는 것이 효율적이다.

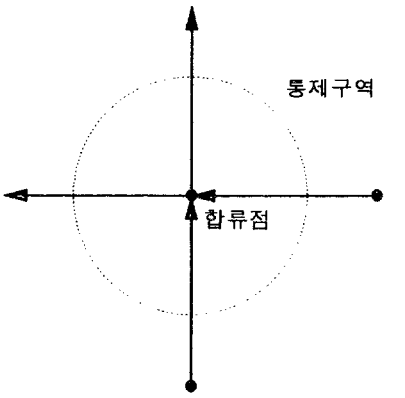


그림 3. 구역통제의 예 1

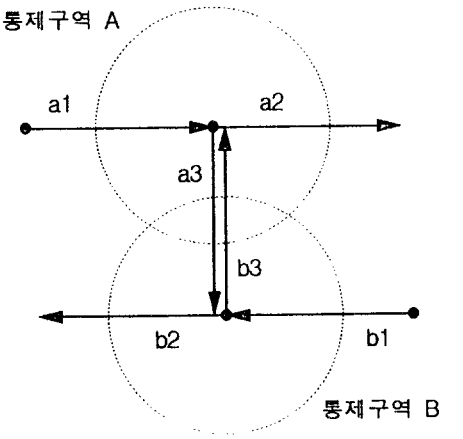


그림 4. 구역통제의 예 2

3-2. 경로 통제

경로통제는 출력경로가 많은 합류 지점에서 AGV의 진행경로에 따라 교통제어하는 방식을 말한다. 이러한 방식은 경로망을 구성할 때 AGV의 진행경로에 따라 각각 다른 입력경로를 정의함으로써 가능하다. 여기서는 대규모 AGV 시스템에서 볼 수 있는 두가지의 대표적인 교차로 통행을 위한 경로통제 방식을 예로든다.

CASE 1: 사거리 교차로 통행

그림 5는 주경로가 서로 교차하는 사거리의 통행을 경로통제 하기위한 경로망의 구

성을 보여준다.

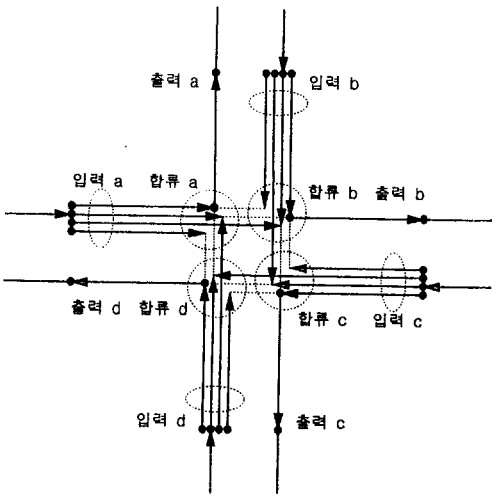


그림 5. 사거리 교차로 통행시 경로망

각 입력에서 진행가능한 경로는 좌회전 경로(l), 직진경로(s), 우회전 경로(r), U-턴 경로(u)로 4개의 경로가 존재하며 총 16개의 경로가 교통제어 구역내에 존재한다. 이중 하나의 경로가 어떤 AGV에 의해 점령된 경우 이 경로와 상호 교차하여 충돌이 발생할 가능성이 있는 경로들은 진입이 통제되어야 한다. AGV에 의해 점령된 경로를 구성하는 직선경로들의 집합을 활동경로(active path)라 하고, 이때 진입이 통제되어야할 입력구역에 있는 직선경로들의 집합을 통제경로(blocked path)라 한다. 각각의 진행가능한 경로들에 대해서 활동경로와 통제경로를 구하기 위하여 다음과 같은 입력행렬 I와 출력행렬 O를 정의한다.

$I=[I_{ij}]$

$O=[O_{il}]$ (3-1)

여기서,

i=a, b, c, d

j=l, s, r, u

Iij : 입력 i 에서 경로 j로 진행하기 위한 입력 직선경로

Oil : 출력 i 에 위치한 직선경로

AGV가 입력 i에서 경로 j로 진행하는 경우, 이 경로가 통과하는 합류점들과 합쳐지는 모든 다른 경로들은 진입이 통제되어야 한다. 그러나, 활동경로를 제외한, 같은 입력 i에서 시작되는 다른 경로들은 통제할 필요가 없다. 이러한 접근방식에 따라 활동경로와 통제경로를 집합연산에 의해 구하면 다음과 같다.

for $i=a, b, c, d$

$A_{il}=(I_i \cap I_i) \cup O_i$,

$B_{il}=(Z_i \cap \overline{I_i}) \cup (I_i \cap I_l)$

$A_{is}=(I_i \cap I_s) \cup O_{i+1}$,

$B_{is}=((Z_i \cup Z_{i+1}) \cap \overline{I_i}) \cup (I_i \cap I_s)$

$A_{ir}=(I_i \cap I_r) \cup O_{l+2}$,

$B_{ir}=((Z_i \cup Z_{i+1} \cup Z_{i+2} \cap \overline{I_i}) \cup (I_i \cap I_r)$

$A_{iu}=(I_i \cap I_u) \cup O_{i+3}$,

$B_{iu}=((Z_i \cap Z_{i+3}) \cap \overline{I_i}) \cup (I_i \cap I_u)$ (3-2)

$Z_i=I_i \cup (I_{i+1} \cap I_u) \cup (I_{i+2} \cap I_s) \cup (I_{i+3} \cap (I_s \cup I_r))$

여기서,

i+1 : abcda 순으로 i 다음 입력 혹은 출력

Aij : AGV가 입력 i에서 경로 j로 진행하는 경우 활동경로

Bij : AGV가 입력 i에서 경로 j로 진행하는 경우 통제경로

Zi : 합류 i로 들어가는 입력 직선경로들의 집합

Ii : 입력행렬 I의 i 열의 원소들의 집합

Ij : 입력행렬 I의 j 행의 원소들의 집합

Oi : 출력행렬 O의 i 열의 원소들의 집합

이상의 활동경로와 통제경로를 이용하여 경로통제를 하기위한 데이타 구조는 다음과 같다.

```
typedef structure {
int act_v;
int input_v;
int block_v[];
```

PATH* act[];
PATH* input;
PATH* block[];
} PATH_BLOCK;

여기서 act_v는 현제 경로를 점령중인 AGV, input_v는 act_v 이외의 입력경로에 위치한 AGV, block_v는 act_v에 의해 통제되고 있는 AGV들을 나타낸다. 또한 act는 활동경로, input은 입력경로, block은 통제경로를 각각 표시한다.

이러한 정보구조를 이용한 경로통제 알고리즘은 다음과 같다.

1) 모든 PATH_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있으면 통제경로에 있는 AGV들을 block_v에 기록하고 진입통제를 한다. 또한 경로를 점령중인 AGV가 활동경로 내에 있는지 조사하여 없으면 act_v 및 block_v의 내용을 제거하여 점령상태를 해제한다.

2) 현재 진입통제 상태이나 모든 PATH_BLOCK의 block_v에 등록되어 있지 않은 AGV들을 조사하여 진입통제 상태를 해제시킨다.

3) i 번째 PATH_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있지 않고 입력경로에 진입허용 상태인 AGV가 존재하면 이를 act_v에 등록함으로써 경로를 점령시킨다.

4) i 번째 PATH_BLOCK에 대하여 현재 점령되어 있는가 조사한다. 만일 점령되어 있으면 현재 경로를 점령하고 있는 AGV 이외의 통제경로에 있는 AGV들을 block_v에 등록하고 진입통제한다.

5) 3),4) 의 과정을 모든 PATH_BLOCK에 대하여 수행한다.

CASE 2 : 주경로와 양방향 경로의 교차통행

그림 6은 주 경로와 한쪽 끝이 차단된 양방향 경로가 교차하는 경우 경로망의 구성을 보여준다.

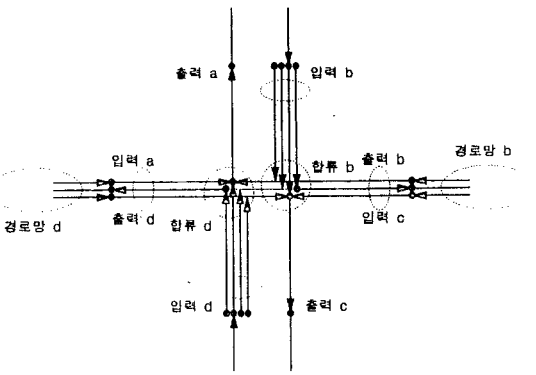


그림 6. 주경로와 양방향 경로의 교차통행시 경로망

이러한 경로는 대규모 AGV 시스템에서 흔히 볼 수 있는 것으로 AGV가 지게차형태 이거나 작업장소가 한쪽 끝이 막힌 좁은 통로에 위치하는 경우에 발생된다. 이 경우 진행가능한 경로는 입력 a, c에서 각각 2개, 입력 b, d에서 각각 4개로 총 12개가 존재하고 CASE 1에 다음과 같은 제한 조건이 있는 경우로 볼 수 있다.

$$\begin{aligned} &for\ i=a,c \\ &I_i \cap I_s = \emptyset \\ &I_i \cap I_u = \emptyset \\ &Z_i = Z_{i+1} \end{aligned} \tag{3-3}$$

$$\begin{aligned} &for\ i=b,d \\ &Z_i = Z_i \cup Z_{i+1} \end{aligned} \tag{3-4}$$

따라서 활동경로와 통제경로는 CASE 1의 식(3-2)로 부터 다음과 같이 얻을 수 있다

$$\begin{aligned} &for\ i=b,d \\ &A_{il} = (I_i \cap I_l) \cup O_i, \\ &B_{il} = (Z_i \cap I_i) \cup (I_i \cap I_i) \\ &A_{is} = (I_i \cap I_s) \cup O_{i+1}, \\ &B_{is} = ((Z_i \cup I_i) \cup (I_i \cap I_s) \\ &A_{ir} = (I_i \cap I_r) \cup O_{i+2}, \\ &B_{ir} = ((Z_i \cup Z_{i+2}) \cap I_i) \cup (I_i \cap I_r) \end{aligned}$$

$A_{iu}=(I_i \cap I_u) \cup O_{i+3}$,

$B_{iu}=((Z_i \cup Z_{i+2}) \cap I_i) \cup (I_i \cap I_u)$ (3-6)

$Z_i=I_i \cup I_{i+1} \cup ((I_{i+2} \cap (I_r \cup I_u)) \cup (I_{i+3} \cap I_r)$

그러나 한쪽 끝이 막혀있는 양방향 경로의 경우 한 AGV가 양방향 경로로 집입했을 때 그 AGV가 양방향 경로를 빠져나갈 때까지 다른 AGV에 의한 양방향 경로의 진입을 통제하여야 한다. 이를 위하여 양방향 경로가 시작되는 직선경로(그림 6에서 출력 b, 및 출력 d)로 부터 이를 빠져나가는 직선경로들(그림 6에서 출력 a, 및 c)을 활동경로로 추가로 지정한다. 이러한 활동경로 와 통제경로를 구하면 다음과 같다.

for $i=b, d$

$A'_{il}=O_i \cup IO_i \cup A_{(i+1)l} \cup A_{(i+1)r}$,

$B'_{il}=(I_i \cap I_l) \cup (I_{i+2} \cap I_r)$

$A'_{ir}=O_{i+2} \cup IO_{i+2} \cup A_{(i+3)l} \cup A_{(i+3)r}$,

$B'_{ir}=(I_i \cap I_r) \cup (I_{i+2} \cap I_l)$ (3-7)

여기서,

A'ij : 출력 i로 양방향 통로에 진입한 후 입력 i+1의 경로 j로 진행하는 직선경로들의 집합

B'ij : AGV가 활동경로 A'ij에 있는 경우 통제하여야 하는 입력 직선경로들의 집합

IOi : 경로망 i에 있는 모든 직선경로들의 집합

이상의 활동경로와 통제경로를 이용하여 경로통제를 하기위한 정보구조 및 제어 알고리즘은 CASE 1의 경우와 동일하다.

4. 시뮬레이션

제안된 중앙운행제어의 적용 예로 자율주행방식 AGV의 사용이 검토되고 있는 X사의 자동차 차체 조립공장 프레스물 운반공정 자동화를 컴퓨터 시뮬레이션 하였다. 프레스물은 프레스 공장에서 생산되어 팔레트에 일정양이 적재되고 팔레트 저장구역에 저장된다. 차체 조립공장의 각 작업장소에서는 서로 다른 종류의 프레스물을 소비하며 빈 팔레트가 발생하면 중앙제어 시스템에 팔레트 공급요구를 한다. 따라서 AGV의 운반작업은 빈 팔레트를 팔레트 저장구역내의 팔레트 회수장소로 이동시키고, 실 팔레트를 조립장소에 공급하는 것으로 이루어 진다. 이러한 작업을 수행하기 위하여 구성된 AGV 시스템의 규모와 AGV 사양은 표 1에, AGV의 주행경로 및 작업장소를 직선경로 표현법과 경로망을 이용하여 설계한 예는 그림 7에, 교차로 주행중인 AGV 들에대한 교통제어 상태는 그림 8에서 각각 보여준다.

표 1. AGV 시스템의 규모 및 AGV 사양

AGV 시스템 규모	작업공간	200(L) x 160(H) [m]
	작업장소	61 개소
	운 반 물	25종, 2(W) x 2(L) [m], 최대 1.5 [ton]
	운반빈도	최소 20 분, 최대 200 분
	대기장소	3 개소
	직선경로	205 개
	구역통제	9 구역
	경로통제	14 구역 110 경로
	AGV 수	10 대
AGV 사양	크 기	2(W) x 4(L) [m]
	주행속도	1 [m/sec]
	회전반경	2 [m]
	충돌방지	전방감지 센서, 범퍼
	작업시간	90 [sec/pallet_handing]

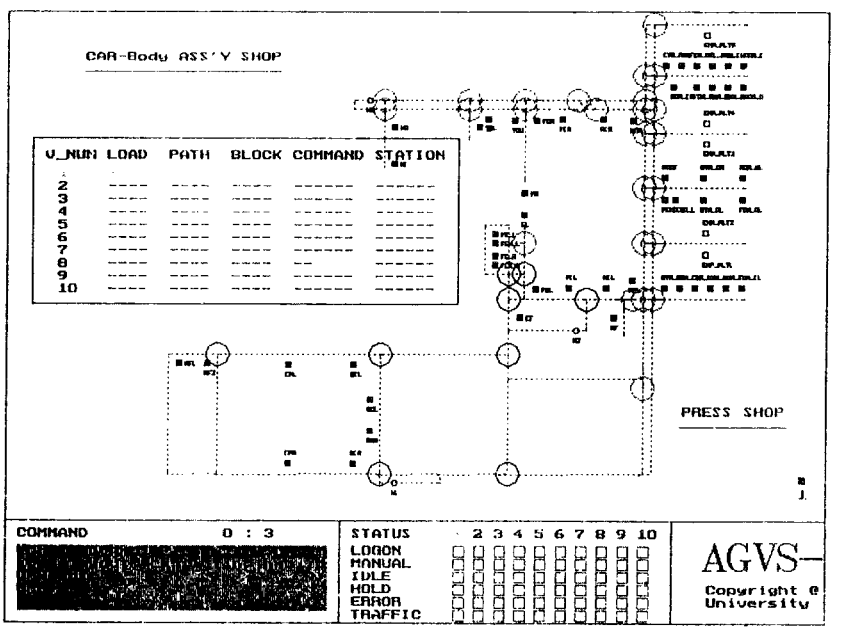


그림 7. AGV 주행경로 및 작업장소

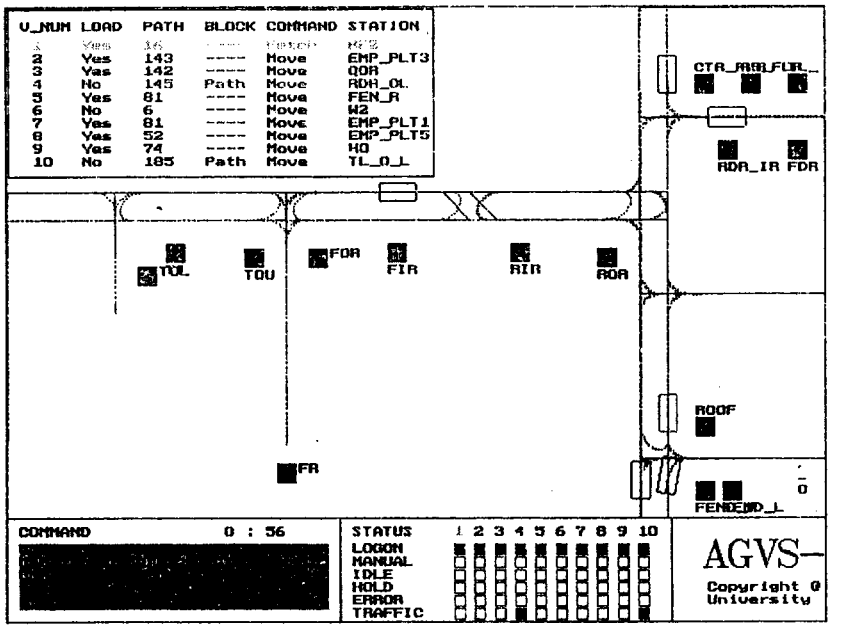


그림 8. 교차로 주행시 교통제어

시뮬레이션은 다음과 같은 방법으로 200 시간 이상 수행 하였다.

1) 차제 조립공장의 각 작업장소의 팔레트 소비시간을 고려해 일정한 시간간격으로 중앙운행제어 시스템에 작업요구를 하도록 한다.

2) 중앙운행제어 시스템은 작업장소로 부터 팔레트 운반요구가 있을 때 현재 대기장소에서 대기 중이거나 작업이 완료되어 대기장소로 이동하는 AGV들 중에서 요구된 작업을 최소 시간에 수행할 수 있는 AGV를 선택하여 주행할 최적경로와 작업내용을 전달한다. 또한 AGV의 현재의 위치 및 경로를 일정시간(여기서는 0.5 sec) 간격으로 감시하며 교통제어 구역에서는 구역통제 혹은 경로통제를 수행한다.

3) AGV는 지시된 경로를 따라 작업장소로 이동하여 팔레트 이재 혹은 적재작업을 수행하고 부여된 작업이 완료되면 지정된 대기장소로 이동한다.

시뮬레이션 결과 제안된 경로 계획 알고리즘에 의하여 AGV가 적절한 경로로 이동하여 작업을 수행함을 볼 수 있었고, 이동중에 AGV의 상호 충돌이나 진입통제에 의한 교통장애가 전혀 발생되지 않고 불필요한 교통통제에 의한 작업시간 증가 현상도 없음을 확인할 수 있었다. 그러나 AGV의 주행경로 선정 및 교통제어 방법은 사전에 정의된 경로망에 전적으로 의존하고 이는 시스템의 규모나 주행환경에 따라 다양한 형태를 가질 수 있으므로 적용 시스템에 따라 적절한 수정과 개선이 요구된다. 또한 전체적인 생산성을 향상시키기 위하여는 경로망 설계에 대한 보다 체계적인 연구가 필요하다.

5. 결 론

본 논문은 자유주행방식 AGV를 이용한 대규모 물류자동화 시스템에 효율적으로 적용할 수 있는 중앙운행제어에 의한 경로계획과 교통제어 방법론을 제시하였다. 자유주행방식 AGV의 주행경로를 정의하기 위

하여 단위 직선경로를 상호 연결한 경로망을 제안하였고, 이때 직선경로는 시작점 위치, 경로 방향 및 경로 길이로 표현하였다.

AGV의 현재 위치에서 작업지점까지 이동하기위한 최소시간 경로 탐색 문제는 제안된 경로망에 A* 알고리즘을 적용하여 해결하였으며 이를 위한 평가지표를 제시하였다. 평가지표에는 주행경로의 길이에 따른 주행시간, 회전양에 따른 지체시간, 혼잡도에 의한 대기시간, 등의 요소로 구성하여 경험적인 요인들이 최적 경로를 선정하는데 반영되도록 하였다. 이러한 최적경로 탐색 알고리즘에 의하여 경유해야할 직선경로의 리스트와 소요 예상시간을 구할 수 있었고, 이때 예상시간은 특정 운반작업을 하기위한 최적 AGV를 선정하는 데에 사용하였고 직선경로 리스트는 AGV를 최적경로를 따라 주행하도록 이동명령어를 구성하는 데에 사용하였다.

또한 제안된 경로망을 이용하여 기존의 AGV 시스템에서 실현하기 어려웠던 복잡한 교차로에서의 교통제어 방법을 개발하고 이에 대한 교통제어 알고리즘을 제시하였다. 교통문제가 발생되는 합류점에서 교통제어 방법으로 출력이 하나인 경우 적용할 수 있는 구역통제 방식과 출력이 여러개 있는 복잡한 교통환경에 적용할 수 있는 경로통제 방식을 제안하였다. 구역통제 방식은 추가적인 직선경로 설정이 필요없고 제어 알고리즘도 비교적 간단하다는 장점이 있으나 출력이 여러개 존재하는 경우 교통장애가 발생할 수 있고 불필요한 교통통제를 일으킬 수 있다. 이러한 문제를 해결하기 위하여 합류점에서 AGV의 진행경로들을 경로망에 구체적으로 표현하여 진행 경로에 따라 입력경로를 통제하는 경로통제 방식을 제안하였다. 경로통제 방법의 구체적 예로 대규모 AGV 시스템에서 흔히 볼 수 있는 사거리 교차로 통행과 주경로와 양방향 경로의 교차로 통행을 들었고, 각각의 경우 경로망 구성방법과 입출력 행렬을 이용한 활동경로 및 통제경로를 구하는 식을 제시하였다. 또한 이러한 경로망과 교통정보를 이용한 경로통제 제어 알고리즘을 개발하였다.

제안된 중앙운행 시스템을 자유주행방식 AGV를 이용한 자동차 차체 조립공장의 물류 자동화 시스템에 적용하였고 컴퓨터 시뮬레이션을 통하여 유용성을 검증하였다.

참고문헌

[1] T. Tsumura, "AGV in Japan", Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1477-1484, 1994.

[2] T. Tsumura, "Trends of Mechatronics in Material Handling and Handling Automation", Jour. of Systems, Control and Information, Vol. 37, No. 6, pp327-331, 1993.

[3] P. Hart, N. J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Sys. Sci. Cyber, SSC-4(2) : pp. 100~107, 1968.

[4] R. A. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Trans. Sys. Man. Cyber., SMC-13(3), pp. 190~197, 1983.

[5] S. Kambhampati and L. S. Davis, "Multiresolution Path Planning for Mobile Robots", IEEE Journal of Robotics and Automation, Vol. RA-2, No. 3, 1986.

[6] K. Fujimura and J. Samet, "A Hierarchical Strategy for Path Planning Among Moving Obstacles", IEEE Trans. on Robotics and Automation,

Vol. 5, No. 1, 1989.

[7] K. R. Harinarayan and V. J. Lumelsky, "Sensor-based Motion Planning for Multiple Robots in An Uncertain Environmemt", Proc. of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1485-1497, 1994.

[8] T. Yamashita, "Start of Autonomous Mobile Robots' Operation in Clean Room", IEEE/RSJ International Workshop on Inteligent Robotics and Systems, pp. 512-519, 1989.

[9] O. Causse and J. L. Crowley, "Navigation with Constraints for an Autonomous Mobile Robot", Proc. of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1899-1905, 1994.

[10] Y. Kanayama and S. Yuta, "Vehicle Path Specfication by Sequence of Straight Lines", in Proc. IEEE Journal of Robotics and Automation, vol. 4, no. 3, pp. 265~276, June, 1988.

[11] N. Tsuda, Y. Kanayama and S. Yuta, "Implementation of RCS - OS for Intelligent Robot," JRSJ, vol. 3, no. 5, pp. 432~443, 1985.

[12] Y. Kanayama, "Concurrent Programming of Intelligent Robots," in Proc. 8th Int. Joint Conf. on AI, pp. 834~838, 1983.

[13] Y. Kanayama, T. Noguchi, "Locomotion Functions for a Mobile Robot Language," IEEE RSJ International Workshop on Intelligent Robots Systems, pp. 542~549, 1989.

[14] S. Yuta and Y. Kanayama, "An Implementation of MITCHI - A Locomotion Command System for Intelligent Mobile Robots," in Proc, 2nd Int. Conf. on Advanced Robotics, pp. 127~134, 1985.

[15] N. J. Nilsson, Principles of Artificial Intelligence, Palo Alto, CA: Tioga, 1980.