

가상 AGV를 이용한 물류자동화 시뮬레이션

김태후, 홍현주, 권석근, 노영식
울산대학교 전기전자정보시스템공학부

Material Handling Automation Simulation Using Virtual AGV

Tai-who Kim, Hyun-ju Hong, Seok-geon Kweon, Young-shick Ro
School of Electric-Electronic Information System Engineering University Of Ulsan

Abstract - 본 논문에서는 다수의 AGV를 이용한 물류 자동화 시뮬레이션에 대해 연구하였다. 각각의 AGV가 독립적으로 명령을 수행하도록 가상의 AGV를 제안하고 이미 연구된 AGV의 주행경로 표현방법, 경로망 구성방법을 이용하였다. 작업요구시 주어진 작업을 가장 빨리 수행 할 수 있는 AGV를 선정하고 최적 경로 계획을 수행하였다. 그리고 작업을 수행 중인 AGV는 경로 합류점에서의 교착상태(dead lock)와 충돌이 발생할 수 있는데 이를 방지하기 위한 통제구역을 정의하고 교통제어를 수행하였다.

테스트할 때는 공유메모리를 이용하여 시스템의 안정성, 효율성을 검증하고, UDP/IP를 이용할 때는 이동명령 전달의 손실, 통신두절, 통신프로토콜에 의한 명령전송 지연 시간 등을 고려한 통신알고리즘을 실험하거나 검증하고자 할 때 사용되어 질 수 있다.

본 논문에서는 AGVMS의 최적이동경로설정 탐색, 교통제어, 최적의 VAGV를 선정하는 알고리즘의 효율성과 안정성을 검증하는 것에 중점을 두고 있으므로 하나의 PC에서 공유메모리를 이용하여 VAGV와 AGVMS와의 통신을 하였다.

1. 서 론

산업현장에서 물류자동화를 위해 다수의 AGV를 운용하는 방식을 사용한다. 자율주행방식의 AGV를 이용하여 대규모 물류자동화 시스템을 구축하는 경우 각각의 AGV의 성능보다도 이를 관리하는 시스템의 성능이 더욱 중요하다. 왜냐하면 작업공간이 광범위하고 다수의 AGV를 정해진 이동경로로 이동시켜야하며 정해진 시간 내에 작업요구를 완료해야 하기 때문이다. 따라서 물류 처리속도를 향상시키며 최소의 AGV를 가동하기 위해서는 물류자동화시스템을 시뮬레이션할 필요성이 있다.

본 논문에서는 가상AGV의 개념을 도입한 AGVMS(AGV Manager System)을 구현하였으며 실제 물류시스템의 성능을 평가하는데 필요한 정보를 제공할 수 있도록 하였다.

2. 본 론

2.1 VAGV(Virtual AGV)

AGVMS의 테스트를 위해 실제 AGV를 사용하는 안정성에 대한 보장이 없는 상태에서 대단히 위험하고 많은 손실이 있을 수 있다. 이를 방지하기 위하여 VAGV를 사용하여 AGVMS를 테스트하고 후에 실제의 AGV에 적용할 수 있도록 하는 것이 필요하다. VAGV를 명령처리부, 모션제어부, VAGV정보부로 구성하였다. 명령처리부에서는 AGV관리자와 통신을 담당하며 순차적인 명령은 Qucuc에 저장하고 즉각적인 명령은 바로 수행하도록 하는 기능을 한다. 또 VAGV정보를 전송하는 기능을 담당한다. 모션제어부는 충돌방지기능과 기구학 및 동역학을 고려하여 설계된 속도제어를 담당하고, VAGV의 상대위치를 계산한다. VAGV의 종류에 따른 기구학과 동역학을 선택 해야한다. VAGV 정보부는 VAGV의 위치와 속도 및 상태를 감시, 보고하기 위한 정보저장을 담당한다. 위치와 속도는 모션제어부에서 갱신(update)되고 명령처리부에서는 명령어 버퍼상태, 내부 센서 상태정보를 갱신한다. 이러한 VAGV의 구성과 동작 흐름도는 그림 1에 나타내었다. VAGV와 AGV관리자와의 통신은 둘이상의 PC에서 UDP/IP를 기반으로 수행 될 수도 있고, 하나의 PC에서는 공유메모리를 통한 통신을 할 수 있다. 따라서 AGV관리자의 교통제어알고리즘이나 이동경로설정, 최적의 AGV를 선정하는 과정은

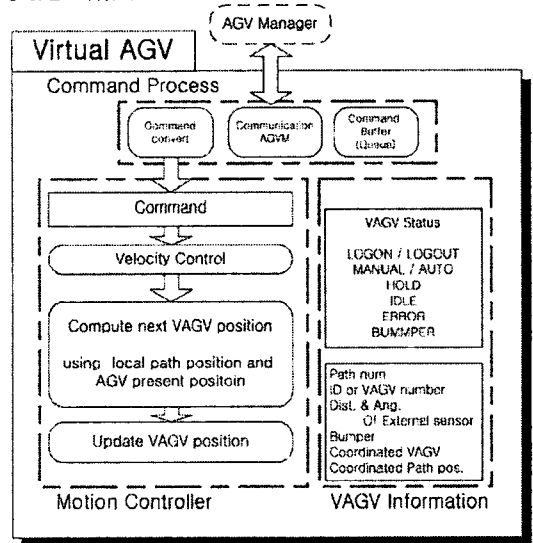


그림 1. VAGV의 구성과 동작 흐름도

N	상태	모션번호	X(cm)	Y(cm)	θ	v(m/s)	ω (rad/s)
1	ST Idle	1	1852.5	11361.0	-179.4	0.0	0.000
2	ST Idle	1	16800.0	3231.2	-93.0	100.0	0.000
3	ST Idle	1	18615.4	7760.0	-173.4	0.0	0.000
4	ST Idle	1	13065.2	3393.0	-0.0	99.9	0.001
5	ST Idle	1	15747.4	3350.0	0.0	100.0	0.000
6	ST Idle	1	14367.5	4950.0	-0.0	0.0	0.000
7	ST Idle	1	17224.2	9549.8	-179.3	0.0	0.000
8	ST Idle	1	8000.3	11547.8	93.8	0.0	0.000
9	ST Idle	1	18215.5	13250.0	-141.7	0.0	-0.892
10	ST Idle	1	15177.4	12150.0	-180.0	0.0	0.000

그림 2. VAGV의 정보

2.2 AGV 관리자

AGV관리자는 다수의 VAGV를 관리하며 VAGV제어에 관련된 명령인 단위작업명령과 모션명령을 처리한다. 단위작업명령은 일련의 물류를 운반하는 명령에 해당되고, 모션명령은 물류를 운반하기 위해서 이동해야 할 경로를 따라서 수행하도록 하는 명령과 물류를 VAGV에 싣고, 내리는 명령에 해당된다. 일단 단위작업명령은 버퍼에 저장되고 수행을 기다리게 되며 이전 단위작업이 완료되면 버퍼에 저장되어진 단위작업명령이 수행된다.

이때 단위작업이 가장 빨리 수행되어질 수 있는 경로를 최적의 경로 선택 알고리즘을 사용하여 경로를 선택하고 이를 수행하기 위한 일련의 모션명령으로 변환되어 VAGV에 전달된다. 모션명령이 전달된 VAGV는 독립적으로 모션명령을 수행하며 단위작업이 완료될 때까지 AGV관리자에 의해 관리된다. 또한 교통제어를 수행하여야 할 경로나 합류점의 상태를 감시하여 충돌과 교착상태를 방지한다.

Order	TASK/명	Order	유형	상태	STATION	거리	경로	구분	이동시간
1	RF1	YES	MOV...	EMP.P...	5			W1	
2	RF1	YES	MOV...	EMP.P...	157	Path		W1	
3	RF1	YES	MOV...	EMP.P...	124			W1	
4	RF1	YES	MOV...	EMP.P...	77			W1	
5	RF1	YES	FEI...	EMP.P...	66			W2	
6	F01	YES	MOV...	EMP.P...	309			W2	
7	F01	NO	MOV...	FDR.LR	187	Path		W2	
8	F01	NO	DEL...	EMP.P...	72			W3	
9	F01	NO	MOV...	EMP.P...	82			W3	
10	TOU	YES	FEI...	TL.O.U	74			W3	

그림 3. AGV관리자

2.3 작업요구관리자

작업요구관리자는 작업요구(Order) 명령을 처리한다. 작업요구는 사용자가 발생시킬 수도 있고 작업장소에서 센서감지에 의하여 발생된 신호를 분류 처리하여 만들어질 수 있다. 입력된 작업요구명령은 일단 버퍼에 저장되었다가 순차적으로 수행되고 작업요구명령을 가장 빨리 수행할 수 있는 최적의 VAGV가 결정되면 작업요구표(Order Table)을 참조하여 일련의 단위작업명령(Task Command)으로 변환되어 AGV관리자로 전달한다. 단위작업명령은 사전에 계획하여 DB관리자의 작업요구표에 등록할 수 있다.

2.4 DB관리자

DB관리자는 경로(path), 통제구역(traffic zone), 작업장소(station), 작업요구표등의 정적 데이터를 저장, 관리한다. VAGV의 가동율과 작업요구처리시간은 항상 계산되어야 하므로 이러한 동적인 데이터는 AGV관리자에서 갱신되도록 하였다.

2.5 명령관리자

사용자에 의한 명령을 처리하는 기능을 하고 입력받은 명령의 종류에 따라 각각의 관리자로 전달된다. 상위수준의 명령을 줄 수도 있고 VAGV를 직접 명령을 전달할 수도 있다. 또, 현재 작업요구명령의 대기상태를 알려주어 현재 대기 중인 작업요구명령의 수를 확인 할 수 있다.

Station	Name	위치	유형	상태	이동시간
1	G01L	11 1000	Left	Exchanging	
2	G01R	11 2650	Left	Exchanging	
3	G02L	11 1250	Left	Exchanging	
4	G02R	11 2850	Left	Exchanging	
5	C01	12 300	Right	Exchanging	
6	C02	2 4400	Right	Exchanging	
7	C03	12 3000	Right	Exchanging	
8	C04	2 2300	Right	Exchanging	
9	RF1	0 3600	Right	Exchanging	
10	RF2	14 3600	Left	Exchanging	
11	WT	10 1550	Com.	Wait	

그림 4. DB관리자

Station	Name	위치	유형	상태	이동시간
1	1070	4150	0	1500	3
2	2650	4150	0	1500	3
3	1250	4150	0	1500	3
4	2850	4150	0	1500	3
5	300	1500	0	1500	3
6	4400	1500	0	1500	3
7	3000	1500	0	1500	3
8	2300	1500	0	1500	3
9	3600	1500	0	1500	3
10	3600	250	180	2200	11
11	3600	250	180	2200	11
12	1550	150	180	1700	18

그림 5. 명령관리자

2.6 그래픽관리자

그래픽관리자는 시뮬레이션중에 사용자가 관찰해야할 대상(Path, Station, VAGV등)을 화면에 표시하고 교통제어가 필요로 하는 부분을 확대하여 사용자가 관찰할 수 있도록 하였다.

2.7 경로 계획 알고리즘

AGV는 항상 하나의 직선경로상에 위치하고 AGV의 이동목표지점인 작업장소도 다른 직선경로상에 위치하므로 AGV를 현재의 위치에서 작업장소까지 이동시키기 위한 주행 경로를 탐색하는 문제는 사전에 정의된 경로망내의 시작경로에서 목표경로에 이르는 직선경로들의 리스트를 찾는 문제가 된다. AGV의 최적경로를 작업장소까지 최소의 시간에 도달할 수 있는 주행 경로로 정의하고 이러한 최적경로를 찾기 위해서 A*알고리즘을 사용하였다. 탐색대상인 직선경로 C에 대한 평가지표 f를 다음과 같이 선정한다.

$$f(C) = g(C) + h(C)$$

여기서 g(C)는 시작경로에서 출발하여 직선경로 C를 수행 완료하는데 예상되는 시간이고 h(C)는 C에서 목표경로에 도착하기 위한 시간에 대한 경험적 예상치를 표시한다. 시작경로에서 한 직선경로를 지나는데 걸리는 시간은 경유한 경로길이, 경로의 혼잡도, 경로의 연결형태 등에 관계되고 AGV의 직선경로 주행을 경로변환 및 경로주행으로 구성되므로 예상시간 g(C)를 다음과 같이 정의할 수 있다.

$$g(C) = g(P) + trans(P, C) + travel(C)$$

여기서 g(P)는 시작경로에서 C이전의 직선경로 P를 수행하는데 예상되는 시간이고, trans(P,C)는 P에서 C로 경로전환에 따른 추가소요시간이며, travel(C)는 C를 수행하여 경로변환점에 이르는 데 걸리는 예상시간이다. trans(P,C)는 다시 다음과 같이 정의된다.

$$trans(P, C) = \frac{dist(P, C) + \alpha angle(P, C)}{vel}$$

여기서 dist(P,C)는 P의 종점과 C의 시점사이의 거리이고, angle(P,C)는 P와 C사이의 연결 각도이며, α는 가장 상수이다. 또한 vel은 VAGV의 보통속도를 나타낸다. travel(C)는 다음과 같이 정의한다.

$$travel(C) = \frac{length(C)}{vel} + traffic(C)$$

여기서 length(C)는 직선경로 C의 경로길이이고, traffic(C)는 C를 주행하는 동안 발생할 수 있는 교통장애에 의한 손실시간을 나타낸다. 평가지표 f(C)에서 h(C)는 C의 종점과 G의 시점의 직선거리를 최대 속도로 주행하는데 필요한 시간을 계산하여 구한다. 이시간은 C에서 G에 이르는 실제 가능한 어떠한 경로로 주행하는 시간보다 항상 같거나 적으므로 A*알고리즘의 경험적 예상치로 사용 가능하다. 이상의 평가지표를 A*알고리즘에 적용하면 AGV의 현 위치에서 작업지점까지 최소 시간으로 도달할 수 있는 경유 직선 경로들의 리스트와 예상시간을 얻을 수 있다.

2.8 교통제어 알고리즘

경로망은 다수의 VAGV에 의해 공유되므로 한 직선 경로에 2개 이상의 직선경로가 연결되어 있는 지점(합류점)에서 VAGV들의 상호 충돌 문제가 발생할 수 있다. 합류점에서의 교통 문제는 VAGV가 지나고 있는 충돌 방지 기능(전방감지 센서, 범퍼등)이나 지적인 동적 장애물 회피 알고리즘에 의하여 해결할 수 없는 문제가 있다. 이러한 문제점을 중앙통제에 시스템에서 적절한 교통제어를 하는 것이 바람직하다. 본 논문에서 사용된 구역통제와 경로통제 방법은 다음과 같다. 구역통제는 합류점 주위의 일정한 크기의 구역을 교통통제 구역으로 정하고 이구역내에는 오직 하나의 VAGV만 수행이 허용되도록 한다(그림 6). 입·출력경로가 많은 합류 지점에서 구역통제만으로 교통통제를 하면 교착상태가 발생할 수도 있다. 따라서 VAGV의 진행경로에 따라 교통제어하는 방식의 경로통제가 필요하다(그림 7). 경로망을 구성할 때 VAGV의 진행경로에 따라 각각 다른 입력 경로를 정의한다. 현재의 경로가 점령중이거나 이미 경로통제구역을 진입한 VAGV가 있을 경우 입력 경로를 통제함으로써 교착상태를 피할 수가 있다.

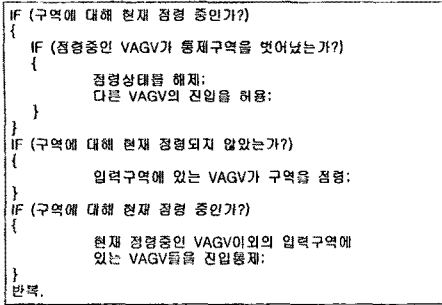


그림 6. 구역통제 알고리즘

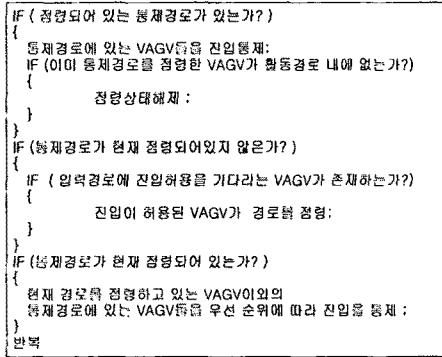


그림 7. 경로통제 알고리즘

2.9 시뮬레이션

Microsoft VC++ 6.0을 사용하여 개인용 컴퓨터로 동작될 수 있도록 제작 되었다. 그림 8은 제한된 AGVMS의 전반적인 구조를 나타내고 그림 9는 전체 작업공간을 3개의 지역으로 나누어 시뮬레이션한 그림이다. 그림 9의 (A)부분은 AGV관리자로서 VAGV의 상태, 작업의 진행상태 및 교통제어에 대한 정보를 확인할 수 있다. 그림 9의 (B)부분은 DB관리자로서 AGV의 가동율과 작업요구명령의 평균실행시간을 확인할 수 있다. 그림 9의 (C) 부분은 그래픽관리자로서 경로, 작업장소, VAGV, 통제구역 등을 나타내고 있다. 그림 10은 그래픽관리자를 나타내는 화면이고 교통통제 알고리즘이 제대로 수행되는지 확대하여 관찰하는 그림으로 붉은 색의 원은 구역통제와 경로통제를 실시할 부분을 나타낸다.

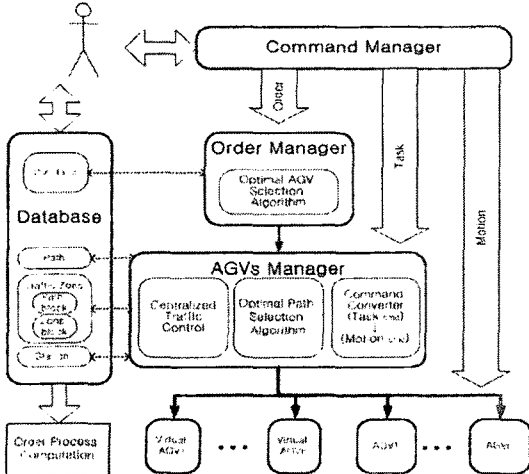


그림 8. AGVMS의 구조도

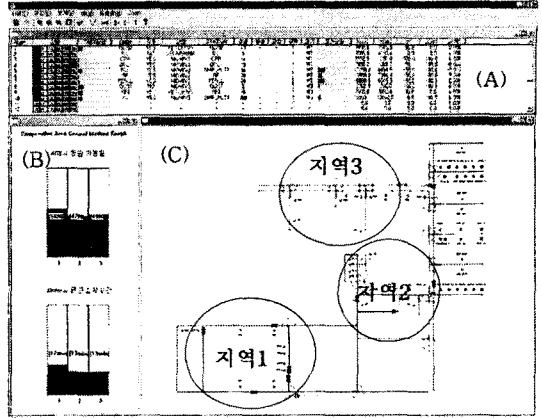


그림 9. AGVMS의 시뮬레이션

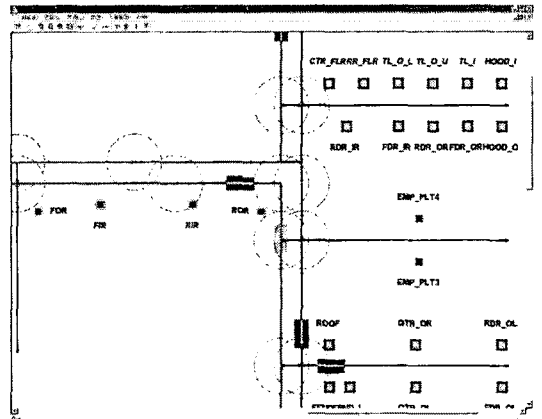


그림 10. 교통제어 수행

3. 결 론

가상 AGV를 이용한 물류자동화 시뮬레이션에 필요한 구조를 제안하였고 제안된 구조로 시뮬레이션 프로그램을 작성하였다. 본 논문은 참고문헌[1]의 향후과제로써 기존의 시스템에 VAGV의 개념을 도입하여 물류자동화 시뮬레이션을 수행하였다. 추후 VAGV의 개념을 확대하여 실제 AGV에도 대처가 가능하도록 할 것이고 UDP/IP를 이용한 이동명령전달의 손실, 통신두절, 명령 전송지연시간을 고려한 시뮬레이션에 대해 연구를 진행 할 것이다.

[참 고 문 헌]

- [1] 노영식, "AGV의 중앙운영제어를 위한 경로계획 및 교통제어", KIEE Vol. 44, No. 3, 1995
- [2] T. Tsumura, "AGV in Japan", Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1477-1484, 1994.
- [3] Subbarao Kambhampati and Larry S. Davis, "Multiresolution Path Planning for Mobile Robots", IEEE Journal of Robotics and Automation, Vol. RA 2, No. 3, 1986.
- [4] Y. Kanayama and S. Yuta, "Vehicle Path Specification by Sequence of Straight Lines" in Proc. IEEE. Journal of Robotics and Automation. Vol. 4, No. 3, No. 5, pp. 432-443, 1985