

다중 쓰레드 기법을 이용한 AGV 의 PC 기반 분산제어 시스템

전성재*, 조연상**, 박흥식#

PC Based Distributed Control System of AGV with Multi-Thread Method

Sung Jae Jun*, Yon Sang Cho** and Heung Sik Park#

ABSTRACT

With the recent progress in flexible manufacturing systems (FMS) in industry, increasing attention has been given to Automatic Guided Vehicle (AGV) systems. An AGV is a self-powered unit for transporting materials between stations without needing to be controlled by an operator. Such a system has several sensors to recognize the external state, and it is designed to travel between stations automatically without external assistance. To manage each device automatically in real time it requires a distributed controller with a main computer as the host, as well as a number of micro-controllers. In this study, an AGV system with dual motor drive was constructed. A Pentium 4 personal computer was set up as the main host for the distributed control, and this communicated with other micro-controllers in the management of the motor. The speed of each motor was also controlled by a micro-controller.

Key Words : AGV(Automatic Guided Vehicle), PC-based system(PC 기반 시스템), Dual motor drive(이륜속도차), Distributed control(분산제어), Hardware module(하드웨어 모듈), Software module(소프트웨어 모듈), Multi threading method(다중 쓰레드 기법), Multi process(다중처리)

1. 서론

최근 생산성 향상과 제품의 품질향상을 위하여 생산의 자동화가 활발히 이루어짐에 따라 물류 운반의 자동화를 위한 AGV 시스템에 대한 관심이 높아지고 있다.¹⁻³ 그리고 컴퓨터 기술의 발달과 함께 표준 환경으로 통합된 개방형 시스템과 비전 기술을 활용하여 지능적인 운행이 가능한 시스템이 요구됨에 따라서 PC 를 기반으로 하는 로봇 및

자동화 시스템에 대한 연구가 많이 이루어지고 있다.⁴⁻⁶ 일반적으로 PC 를 기반으로 하는 제어기의 하드웨어 플랫폼(hardware platform)은 다양한 주변 장치와의 연계성 및 적응성이 우수하고 범용성을 갖추고 있으며, 비전 기술의 도입 및 그래픽 인터페이스(graphic interface) 환경의 제공으로 지능적이고 편리한 운용이 가능하다는 장점을 가지고 있다.⁷

특히 본 연구에서는 AGV 시스템의 장애물 검

접수일: 2004 년 12 월 29 일; 게재승인일: 2005 년 7 월 8 일

* 동아대학교 기계공학과 대학원

** 동아대학교 기계공학과

동아대학교 기계공학과

E-mail parkhs@daunct.donga.ac.kr Tel. (051) 200-7653

출 및 유도선 검출을 위하여 3 대의 CCD 카메라를 이용하였기 때문에 처리해야 될 많은 영상 데이터를 쉽고 빠르게 처리하기 위해서는 PC 를 기반으로 하는 하드웨어 플랫폼의 사용이 필수적이라고 할 수 있다.

그러나 GUI(Graphic User Interface) 운영체제인 Windows 2000은 수 ms(millisecond)의 서보 루프(servo loop)를 보장 받지 못하고 다른 장치를 처리하는 동안 발생하는 시간지연으로 인하여 인터럽트(interrupt)의 실시간 처리가 힘들다는 문제점을 가지고 있다.⁸ 따라서 Windows 2000을 실시간 제어에 활용하기 위해서는 제어기를 분산제어시스템으로 구성하고 이에 대한 실시간 제어성능을 검토할 필요가 있다.^{9, 10}

분산제어시스템¹¹은 다수의 컴퓨터를 네트워크로 연결하여 각 장치의 제어를 분담하여 동시에 실시간 처리하는 시스템이다. 이는 중앙집중방식인 범용제어기^{12, 13}에 비해 실시간 제어를 위하여 고성능, 고가의 시스템을 요구하지 않고 일반 PC 및 표준화된 여러 장비를 그대로 이용하여 저렴하고 용이하게 시스템을 개발할 수 있다는 장점이 있다. 그래서 PC 를 기반으로 하는 제어기의 설계는 대부분 분산제어 방식을 이용하고 있으며 최대한 실시간성 및 신뢰성을 높이기 위한 운용 소프트웨어에 대한 연구를 활발히 하고 있다. 본 연구에서는 이러한 운영 소프트웨어 개발 기법으로 다중 쓰레드 기법(multi thread method)을 도입하였다.

다중 쓰레드 기법^{14, 15}은 하나의 프로세스(process)에서 쓰레드라고 하는 수행중인 독립된 작업을 여러 개 두어 동시에 다중작업을 하는 것이다. 이는 수행 중인 작업을 마친 후에 다음 작업을 수행하는 단일 쓰레드 처리 방법과는 달리 여러 장치를 효율적으로 동시에 제어할 수 있다는 장점을 가지고 있다. 예를 들어, 스테레오 처리에 걸리는 시간이 300ms, 주행라인 검출 시간이 200ms 이고 모터 컨트롤러와의 여러 가지 정보 처리에 걸리는 시간이 50ms 라고 하면 단일 쓰레드 처리 방식에서는 550ms 마다 한번씩 각 작업을 수행할 수 있다. 그리고 영상처리 과정에 처리가 다소 늦어진다면 그대로 다른 작업에 영향을 미치므로 실시간성을 유지할 수 없다. 따라서 최대한 실시간성을 확보하기 위해서는 다른 여러 장치와의 데이터 입출력 및 여러 계산 처리하는 부분을 각각 독립적인 쓰레드로 구성하여 다중 수행하는 다중 쓰레드 기

법이 필요하다. 그리고 이러한 방법들은 Windows NT 의 RTX(real-time extension)¹¹, RTOS(real-time operation system)¹¹를 이용하는 것보다 경제적이고 빠르게 운용 소프트웨어를 개발할 수 있다는 장점을 가지고 있다.

따라서 본 연구에서는 분산처리에 의한 이륜 속도차 방식의 AGV 시스템을 개발하기 위하여 펜티엄 4 PC 를 메인 제어기로 하여 스테레오 비전에 의한 장애물 검출, 비전에 의한 유도선 검출 및 이륜 속도제어를 하였다. 그리고 PC 를 기반으로 한 실시간 분산 제어시스템을 위하여 각 장치의 하드웨어 및 소프트웨어 모듈을 구성하였으며, 각 소프트웨어 모듈은 다중 쓰레드 기법을 사용하여 수행하였다. 그리고 이에 대한 실제 구동 실험을 통하여 제어성능을 검토함으로써 Windows 2000을 운영체제로 하는 PC 를 이륜속도차 AGV 시스템을 위한 분산제어기로 활용하고자 하였다.

2. 분산제어 모듈의 구성

2.1 하드웨어 모듈

본 연구에서 개발하고자 하는 AGV는 3대의 CCD 카메라로부터 영상신호를 입력 받아 스테레오 비전에 의한 장애물 검출 및 유도선 추적 처리하게 되므로 이러한 과정 중에 시간지연이 발생하여 모터 제어의 실시간성을 확보할 수 없게 된다. 이와 같은 문제점을 해결하고 시간지연을 최소화하기 위해서 빠른 응답 속도로 실시간 처리를 요구하는 부분에 개별적으로 마이크로 컨트롤러(micro controller)를 이용하여 분산처리를 하도록 구성하고 주행상황, 유도선 추적, 장애물 감지 및 대처 등과 같이 비교적 느린 응답 속도를 요구하는 부분은 PC를 기반으로 하는 메인 컨트롤 모듈에서 처리하도록 하였다. Fig. 1은 이륜속도차 AGV의 구동 모터 제어 및 영상처리를 위하여 PC 를 기반으로 하는 분산처리 하드웨어 모듈을 나타낸 것으로 하드웨어 모듈은 크게 메인 컨트롤 모듈(main control module), 명령 컨트롤 모듈(command control module), 모터 및 초음파 컨트롤 모듈(motor & ultra-sonic control module)로 구성되어 있다. 메인 컨트롤 모듈은 3채널 멀티 프레임 그레버(multi frame grabber)를 장착하여 장애물에 대한 스테레오 영상신호 처리 및 유도선에 대한 영상신호 처리를 하고 명령 컨트롤 모듈로부터 받은 모터 및 초음

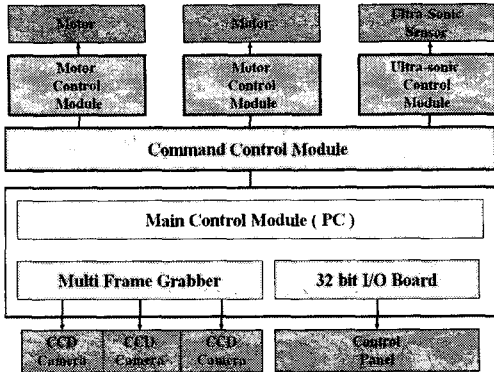


Fig. 1 Structure of hardware module

과 감지기의 상태를 분석하여 AGV 에 대한 제어를 총괄한다. 명령 컨트롤 모듈은 메인 컨트롤 모듈과 다른 모듈 사이에 원활한 통신이 이루어지도록 적절하게 명령 및 데이터를 분배하는 모듈로 메인 컨트롤 모듈과는 RS232 시리얼 통신을 하고 다른 모듈과는 인터럽트에 의한 8비트 데이터 통신을 하는 구조로 되어있다. 모터 컨트롤 모듈은 모터의 동작 지령을 위한 신호를 출력하는 것으로 각 모터에 부착하여 독립적으로 실시간 속도를 제어한다. 그리고 초음파 컨트롤 모듈은 초음파 센서를 통하여 초음파를 출력하고 반향되는 초음파를 감지하여 거리를 측정하여 메인 컨트롤 모듈에 전달한다. 이러한 구조는 PC 를 기반으로 하는 메인 컨트롤 모듈의 운영체제가 windows 2000을 기반으로 하기 때문에 수 ms 의 서보 루프를 보장받기 힘들다는 단점을 보완하기 위한 것이다. Fig. 2는 8비트 마이크로 컨트롤러인 8051을 이용하여 자체 제작한 컨트롤 모듈의 사진을 나타낸 것이다.

2.2 소프트웨어 모듈

본 연구에서는 AGV 의 통합운영을 위하여 메인 컨트롤 모듈인 PC 의 운영체제로 Microsoft Windows 2000을 사용하였다. 그러나 이를 실시간 운영체제로 하여 소프트웨어 모듈을 구축하기 위해서는 주기적인 서보 루프의 보장과 다른 모듈의 처리과정에 발생하는 시간지연 최소화 등과 같은 여러 가지 문제점을 해결해야 한다. 이러한 문제점을 해결하기 위하여 각 모터는 전용 마이크로 컨트롤러로 제어를 함으로써 주기적인 서보 루프를 확보하고, 각 소프트웨어 모듈은 다중 쓰레드 처리

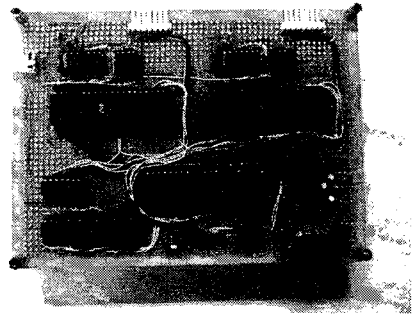


Fig. 2 Photo of control module

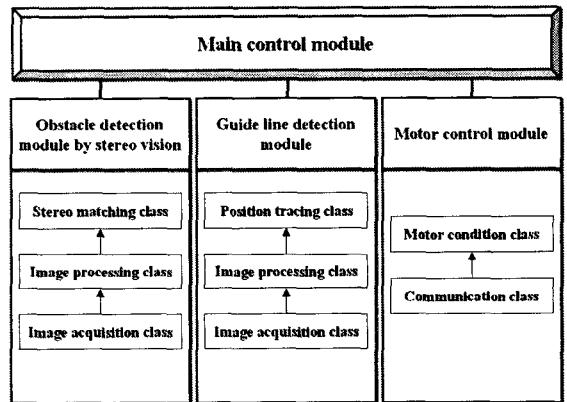


Fig. 3 Structure of integration software module

기법을 이용하여 시간지연을 최소화할 필요가 있다. 따라서 본 연구에서 개발한 통합 소프트웨어는 비주얼 C++ 클래스로 모듈화 하였으며, Fig. 3의 소프트웨어 모듈의 구성도에 나타난 바와 같이 스테레오 비전에 의한 장애물 검출, 유도선 검출 및 모터 제어 처리 등과 같은 실시간성을 요구하는 모듈은 다중 쓰레드 처리에 의해 최대한 실시간성을 보장받을 수 있도록 구성하였다.

여기서 스테레오 비전에 의한 장애물 검출 모듈은 CCD 카메라로부터의 영상신호를 컴퓨터 메모리에 저장하는 영상획득 클래스, 노이즈 제거 및 threshold 처리를 위한 영상처리 클래스 그리고 물체를 검출하고 장애물 정보를 추출하는 스테레오 매칭 클래스로 구성되어 있다. 그리고 유도선 검출 모듈은 장애물 검출 모듈과 마찬가지로 영상획득 및 영상처리 클래스와 유도선에 대한 정보를 추출하여 주행 상황을 전달하기 위한 위치추적 클

래스로 구성되어 있다. 그리고, 모터 제어 모듈은 명령 컨트롤 모듈과 통신하기 위한 통신 클래스, 상황에 따른 모터의 속도 조절, AGV의 위치계산 및 주행거리 계산을 위한 모터 상태 클래스로 구성되어 있다. 이러한 각 모듈들은 하드웨어 모듈과의 데이터 전송 및 처리 클래스로 구성되어 독립적인 쓰레드로 분산 처리되고 있으며 통합 소프트웨어 모듈은 각 모듈에서 독립적으로 처리된 결과를 분석하여 주행 상황을 관리 감독하도록 하였다.

3. 이륜속도차 AGV에 적용

3.1 AGV 시스템

Fig. 4는 본 연구에서 개발한 하드웨어 및 소프트웨어 분산처리에 의한 이륜속도차 방식 AGV 시스템의 사진을 나타낸 것이며, Fig. 5는 이륜속도차 방식 구동부의 사진이다. 차량의 총 중량은 약 100 kg으로 전방의 상부에는 스테레오 비전에 의한 장애물 검출을 위하여 2대의 CCD 카메라가 장착되어 있다. 그리고 프레임의 중앙 하부에는 좌우바퀴를 구동하기 위하여 1/50의 유성기어식 감속기가 부착된 2대의 서보 모터를 장착하였으며, 구동 바퀴는 직경 200 mm인 튜브식 타이어를 사용하였다. Fig. 6은 이와 같이 구성된 이륜 속도차 AGV의 좌표계 모델을 나타낸 것으로 x-y 평면상을 운동하는 AGV 시스템의 중심 C의 좌표를 (x,y), AGV의 전진각을 θ , 좌우 구동 바퀴의 회전각을 θ_R, θ_L 라고 두면 시스템의 위치벡터 \bar{q} 는 식 (1)과 같이 나타낼 수 있다.

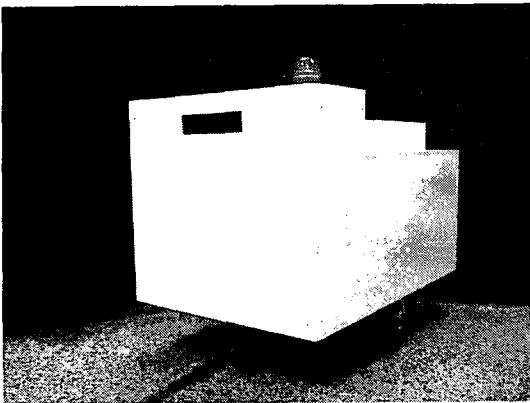


Fig. 4 Photo of AGV system with dual motor drive

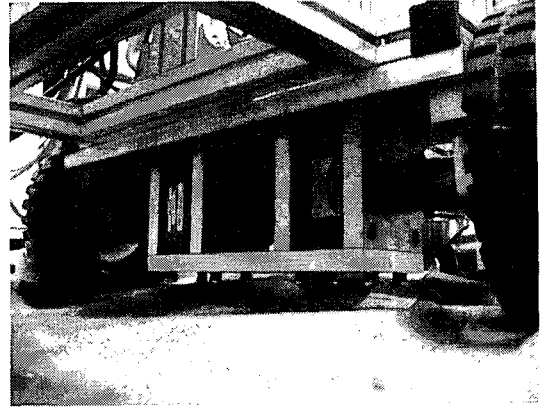


Fig. 5 Photo of dual motor drive part

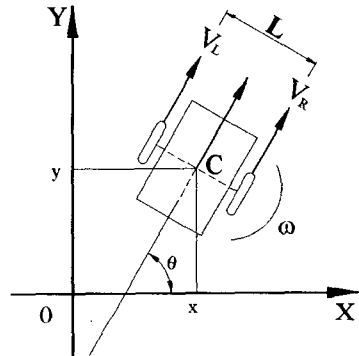


Fig. 6 Coordinate system model of AGV

$$\bar{q} = [x, y, \theta, \theta_R, \theta_L] \quad (1)$$

그리고 AGV의 중심점 C에서의 속도벡터 \bar{v} 는 식 (2), 식 (3) 및 식 (4)와 같이 접선속도 v 및 회전 각속도 ω 로 나타낼 수 있다.

$$v = \frac{v_R + v_L}{2} = r \frac{\omega_R + \omega_L}{2} \quad (2)$$

$$\omega = \frac{v_R - v_L}{2} = r \frac{\omega_R - \omega_L}{2}$$

$$\bar{v} = [v, \omega]^T \quad (3)$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (4)$$

여기서 바닥과 바퀴의 접촉면에서 바퀴면에 수직인 방향으로의 미끄러짐이 없다고 할 때, 비

홀로노믹(Non-holonomic) 제한 조건으로 식 (5)와 같이 나타낼 수 있으며, 이때 이동거리 및 회전반경은 식 (6), 식 (7)과 같이 구할 수 있다.

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (5)$$

$$D = \int \frac{v_R + v_L}{2} dt \quad (6)$$

(D : moving distance)

$$R = \frac{L}{2} \frac{v_R + v_L}{v_R - v_L} = \frac{L}{2} \frac{\omega_R + \omega_L}{\omega_R - \omega_L} \quad (7)$$

(R : turning radius)

3.2 서보 모터의 속도제어

본 연구에서는 바퀴 구동을 위하여 1/50 감속기가 부착된 서보 모터를 사용하였다. Fig. 7은 모터의 구동 드라이버 및 컨트롤 모듈의 구성도를 나타낸 것이다. 모터는 서보 드라이버를 통하여 closed loop에 의한 PWM 속도제어를 하고, 제어 신호의 입출력은 8bit 마이크로프로세서인 8051을 사용하였다. 그리고 속도는 8bit DAC (digital analog converter) 를 통하여 0~5V 사이의 전압 레벨로 지령하도록 하였다.

Fig. 8은 속도 설정 전압에 따른 구동바퀴의 원주 속도를 나타낸 것으로 약 0.7 V부터 4 V까지는 전압이 증가함에 따라서 속도가 선형적으로 증가하지만 4V 이상에서는 약 36m/min로 속도가 일정하게 유지되는 것을 볼 수 있다. 따라서 본 연구에

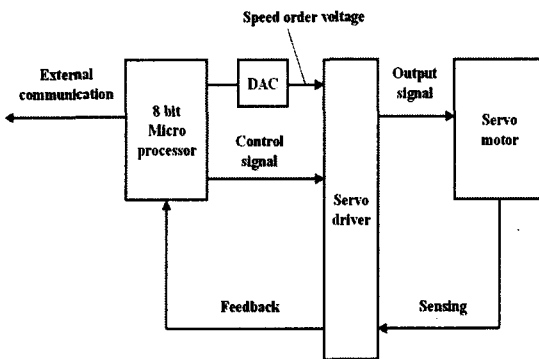


Fig. 7 Block diagram of servo motor driver circuit

서는 바퀴의 최대 원주 속도를 30 m/min로 설정하여 속도 전압을 3V 이하에서 지령하도록 하였다.

4. 실험 및 결과

Fig. 9는 구동바퀴의 속도를 30m/min으로 설정하였을 때의 응답 곡선을 나타낸 것으로 속도응답은 약 1.5 초에서 정상상태에 도달하였으며, 정상상태 응답은 기준 속도에 대하여 0.66%의 오차를 가지고 있다. 이는 30 m/min 이내의 느린 속도로 운행되는 AGV 시스템에 대하여 매우 정밀한 속도 추정 성능을 가지고 있으며 적절한 응답특성을 나타낸다고 생각된다.

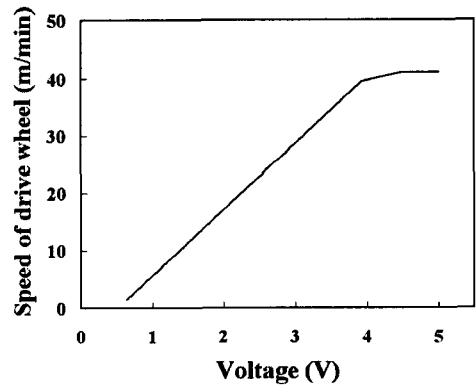


Fig. 8 Effect of speed order voltage on circumferential speed

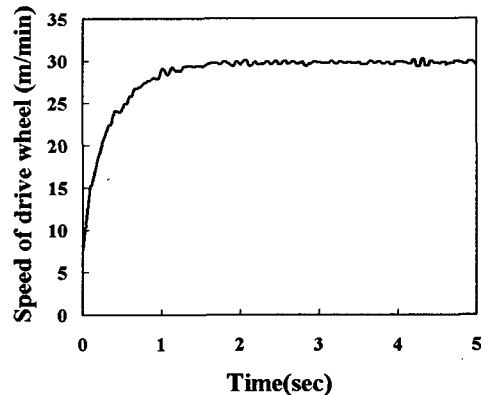
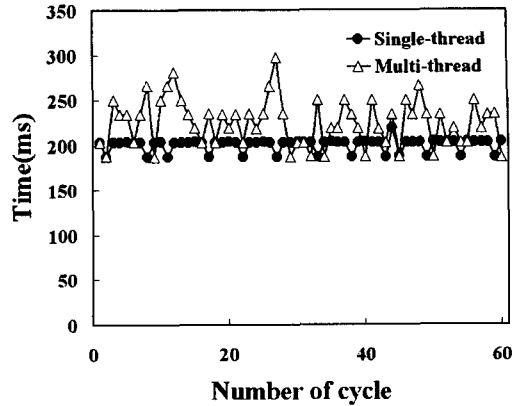


Fig. 9 Speed response curve of drive wheel

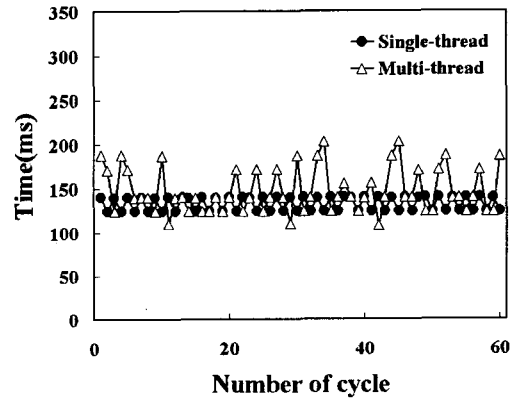
Fig. 10 은 분산처리를 위한 소프트 모듈을 구성하여 개별적으로 쓰레드를 실행한 경우와 3가지 모듈을 동시에 다중으로 실행한 경우의 사이클 처리 시간을 나타낸 것이다. (a) 스테레오 비전에 의한 장애물 검출 모듈은 다른 모듈을 수행하지 않고 개별적으로 쓰레드를 수행하였을 경우, 처리시간이 약 180~200 ms 로 비교적 안정되게 작동하는 것을 볼 수 있다. 그러나 다른 모듈과 함께 다중 쓰레드로 수행하였을 경우에는 처리시간이 약 180~300 ms 로 수행하는 것을 볼 수 있다. 그리고 (b) 유도선 검출 모듈은 개별적으로 쓰레드를 수행하였을 경우, 약 120~140 ms 의 처리시간을 나타내지만 다중 쓰레드로 수행하였을 경우에는 약 120~200 ms 의 처리시간을 나타내고 있다. 이것은 다른 쓰레드 처리에 의한 시간 지연 때문에 최대 80~120 ms 정도의 지연이 발생된 것이라고 생각된다. 그리고 (c) 모터 컨트롤 모듈은 다중 쓰레드 수행 시에 20~40 ms 로 처리되고 있음을 알 수 있다.

여기서 각 모듈을 단일 쓰레드로 구성하여 처리하면 각 모듈은 순차적으로 처리되기 때문에 개별 쓰레드 처리시간의 총 합인 310~360 ms 마다 한번씩 수행되지만, 다중 쓰레드로 구성하여 처리하였을 경우에는 각 모듈은 최대 300, 200 및 40 ms 마다 한번씩 독립적으로 수행한다. 따라서 AGV 의 최대 속도를 30m/min 으로 하였을 때 장애물 검출은 300ms, 유도선 검출은 200ms 그리고 모터 컨트롤은 50 ms 마다 처리하는 것을 목표로 하였기 때문에 비록 시간지연이 발생하더라도 다중 쓰레드 기법을 통하여 충분히 실시간성을 유지한다고 생각되며, Windows 2000을 기반으로 하였을 경우, 단일 쓰레드 내에서 각 모듈을 순차적으로 처리하는 것보다 다중 쓰레드 기법에 의하여 분산 처리를 하는 것이 유용하다고 생각된다.

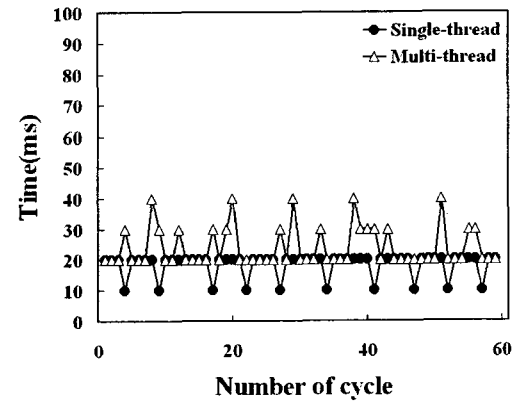
AGV를 30 m/min의 속도로 직선 주행하였을 경우, 정상상태에서 이동거리 50mm마다 측정된 순간 오차를 Fig. 11에 나타내었다. 정상상태에서 직선 주행 시에는 $\pm 0.05\text{mm}$ 이내로 매우 정밀하게 이동하는 것을 알 수 있다. 그리고 Fig. 12는 회전 반경 1 m로 회전운동을 하였을 때 정상상태에서의 순간 회전반경의 오차를 나타낸 것으로 회전반경 1 m에 대하여 $\pm 10\text{mm}$ 이내의 오차를 가지고 있다. 이러한 결과로 볼 때, 하드웨어 및 소프트웨어적으로 모듈화 하여 다중 쓰레드 기법에 의한 분산처리를 수행하면 주행 시에 스테레오 비전에 의한 장애물



(a) Obstacle detection module



(b) Guide line detection module



(c) Motor control module

Fig. 10 Comparison of cycle time on single and multi-thread processing of software module

검출 및 주행라인 검출에 의하여 발생할 수 있는 시간지연을 최소화하여 주어진 경로를 매우 정밀하게 추종하여 운동할 수 있다고 생각된다.

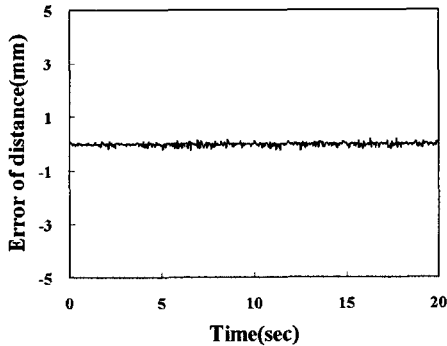


Fig. 11 Error of instantaneous distance in linear motion

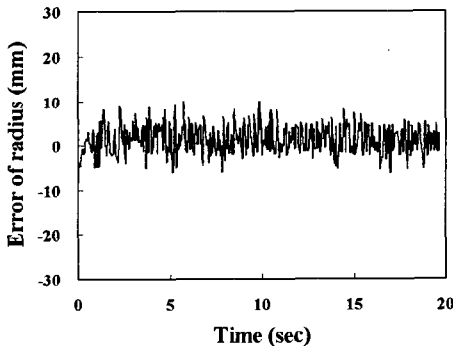


Fig. 12 Error of instantaneous radius in rotational motion

5. 결론

본 연구에서는 분산처리에 의한 이륜속도차 방식의 AGV 시스템을 개발하기 위하여 펜티엄 4 PC를 메인 제어기로 구성하고, 실시간 분산처리를 하기 위하여 각 장치의 하드웨어 및 소프트웨어 모듈을 구성하여, 이에 대한 실제 구동 실험을 통하여 제어성능을 검토한 결과 다음과 같은 결론을 얻었다.

1) 구동바퀴의 속도를 30 m/min 으로 설정 하였을 때, 속도응답은 약 1.5sec 에서 정상상태에 도달 하였으며, 정상상태응답은 기준 속도에 대하여 0.66%의 오차로써 30m/min 이내의 느린 속도로 운행되는 AGV 시스템에 대하여 매우 정밀한 속도 추정성능을 가지고 있다.

2) 각 모듈의 처리시간은 장애물 검출은 300 ms, 유도선 검출은 200 ms 그리고 모터 컨트롤은 50 ms 이내로써 충분히 목표로 하는 실시간성을 유지하고 있다.

3) 직선 주행 시에는 정상상태에서 이동거리 50mm 마다 측정된 순간 오차가 ± 0.05 mm 이고, 회전 반경 1 m 로 회전운동을 하였을 때 정상상태에서의 순간 회전반경의 오차는 ± 10 mm 이내 이다. 이는 다중 쓰레드 기법에 의한 분산처리를 수행하면 다른 작업 중에 발생할 수 있는 시간지연을 최소화하여 주어진 경로를 정밀하게 추종하여 운동할 수 있다고 생각된다.

4) 메인 제어기를 Windows 2000 을 운영체제로 하는 PC 를 기반으로 하더라도 하드웨어 및 소프트웨어적으로 모듈화 하여 단일 쓰레드 내에서 각 모듈을 순차적으로 처리하는 것보다 다중 쓰레드 기법에 의하여 분산 처리를 하는 것이 유용하다

후 기

본 연구는 산업자원부의 지원혁신 인력양성사업의 연구결과로 수행되었음.

참고문헌

1. Tanchoco, J. M. A., "Material Flow Systems in Manufacturing," Chapman & Hall, 1994.
2. Dunkin, Ann, "Automated Guided Vehicle Systems : An Introduction," Jr. of IE, pp. 47-51, 1994.
3. Wright, P. K., Bourne, D. A., "Manufacturing Intelligence," Addison Wesley, 1988.
4. Wright, P. K., "Principles of open-architecture manufacturing," Journal of Manufacturing Systems, Vol. 14, No. 3, pp. 187-202, 1995.
5. Babb, M., "PCs: The foundation of open architecture control system," Control Engineering, pp.75-82, 1996.
6. Kim, J. G., Choi, K. H. and Hong, K. S. "A PC-Based Open Robot Control System : PC-ORC," Journal of Control, Automation and System Engineering, Vol. 6, No. 5, pp. 415-425, 2000.
7. Smith, J. S., "Design and implementation of FMS control software," Proc. of the 1996 FAIM Conf.,

- 1996.
8. Julian, T., "Beginning Windows NT Programming," Information Publishing Group, 1999.
 9. Chang, S. O. and Lee, J. K., "Development of a hydraulic servo system real-time simulator using a one-board micro-processor and personal computer," Proceedings of the 4th Asia-Pacific Conference on Control and Measurement, pp.128-132, July, 2000.
 10. Frankin, Gene F., Powell, J. David and Michael Workman, "Digital control of dynamic systems," Addison Wesley, 1997.
 11. Kang, S. K., Seo, S. H., "Open soft CNC system," Cheong Moon Gak Pub., 2003.
 12. Lee, S. H. and Bae, Y. H., "Development of Heterarchical control system through automated plant monitoring," Journal of the KSPE, Vol. 14, No. 1, pp. 108-118, 1997.
 13. Park, J. Y. and Chang, P. H., "Development of high performance universal controller based on multiprocessor," Journal of the KSPE, Vol. 10, No. 4, pp. 227-235, 1993.
 14. Asche, Ruediger R., "Win32 Multithreading Performance," Microsoft MSDN Library, 1996.
 15. Cohen, Aaron and Woodring, Mike, "Win32 Multithreaded Programming," O'Reilly, 1997.