

CPLD를 이용한 이륜 속도차방식 AGV 제어기 설계 및 구현

*진중호, 백한석, 한석봉
경상대학교 전자공학과

Design and Implementation of the Dual Motor Drive AGV Controller Using CPLD

*Jung-Ho Jin, Han-Suk Baek, Seok-Bung Han

Department of Electronic Engineering, Gyeongsang National University.

*jhjin@vlsi.gsnu.ac.kr

요약

본 논문에서는 AGV(Automatic Guided Vehicle)를 제어하기 위한 hard-wired 제어기를 설계하였고, CPLD(Complex Programmable Logic Device)를 이용하여 구현하였다. 제안된 제어기는 자율주행을 위한 유도장치 제어기, 모터 제어장치, 입출력 sequence 제어기 등을 포함하고 있다.

마이크로프로세서에 의해 구현된 기존방식에 비해 hard-wired 제어방식을 사용하므로 복잡한 프로그램 과정을 줄일 수 있다. 또한 메모리, 조합논리, 순서논리 회로를 쉽게 추가할 수 있어 제품의 개발시간 단축, 제품 크기 축소, 난이도 등에서 발생하는 총 제작비용 등을 감소시킬 수 있다.

제어기는 VHDL을 이용하여 동작적 기술 방법으로 설계되었으며, Altera사의 MAX+Plus II를 사용하여 합성하였고, EPF10K10LC84-4 디바이스로 구현하여 AGV 모형(Line-tracer)에 적용시켜 동작을 확인하였다.

Abstract

This paper describes the design and implementation of a hard-wired AGV controller using CPLD(Complex Programmable Logic

Device). The proposed controller manages a guidance equipment, motor and I/O sequence controller for a self-control traveling.

Compared with a conventional μ -processor, the CPLD controller using a hard-wired control method can reduce a difficult programming process. Also, the total costs of production are reduced, such as development time, product's size and difficulty, because memory, combinational logic and sequential logics are implemented by CPLD.

The Controller designed using behavioral description method with VHDL and was synthesized by MAX+Plus II of the ALTERA co. We implemented controller using EPF10K10LC84-4 device.

1. 서론

AGV는 자율주행을 하며 물건 등을 나르거나 다른 일을 하는 로봇을 말한다. 차량 자체에 외부 인식할 수 있는 각종 센서들이 내장되어 있어 외부 도움 없이도 판단하여 주행할 수 있다. 응용분야는 복지(가정, 호텔, 도서관), 산업(반도체, 원자력발전소), 의료(의료기기 반송), 장애자(무인 휠체어), 방재(화재 감시)용으로 용도에 따라 매우 광범위하게 사용된다.

이러한 AGV의 기능을 구현한 Line-tracer(모형)은 주로 마이크로프로세서에 의해 제어되는

본 연구는 반도체설계교육센터(IDEA)의 지원을 받아 수행되었습니다.

데, 제한된 자원의 효율적 제어를 위한 프로그래밍 과정과 실제 마이크로프로세서에 의해 구현될 수 없는 주변 소자들이 필요하게 된다.

본 논문에서는 hard-wired 방식으로 CPLD 제어기를 설계하였고 Line-tracer에 적용시켜 동작을 확인하였다. 이 원칩 제어기는 FSM(Finit State Machine) 형태의 제어블록, 스텝모터 상발생기, 주변 MSI급 소자들을 모두 포함하고 있다. 따라서 제어가 쉬워지고, 제품의 크기, 생산가격을 줄일 수 있다. 제안된 제어기는 ALTERA사의 FLEX10K10LC84-4를 이용하여 구현하였다.

II. 본론

1. AGV의 이동원리

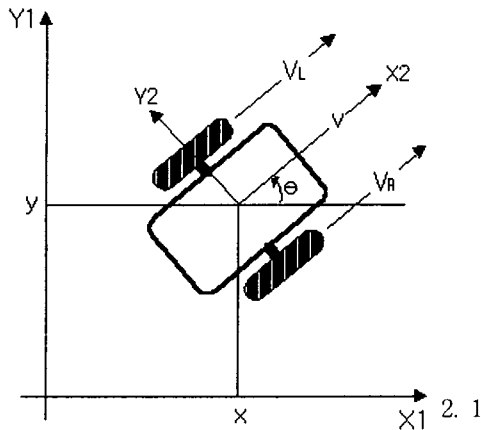


그림 1. AGV 좌표계 및 기구학 모델

그림 1은 AGV가 위치한 공간에서 나타나는 좌표이다. X1-Y1 좌표계는 AGV 외부에 기준점을 둔 절대 좌표계이다. 그리고 X2-Y2 좌표계는 AGV 중심점을 원점으로 하고 로봇이 향한 방향을 x축으로 하는 좌표계이다. AGV의 자세(Posture) P는 아래와 같이 정의할 수 있다.

$$P = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

여기서, x, y는 AGV의 중심점의 좌표이고, θ 는 AGV의 앞부분이 향하는 방향각이다. 궁극적으로 AGV 구동부에 적절한 출력을 가해 원하는 자세를 얻기 위해 이 값들을 맞추어야 한다. 이와 같은 바퀴의 회전 속도와 그에 따라오는 자세

관계를 나타낸 식을 기구학 식이라 한다. 기구학 식을 풀기 위해 몇 개의 조건을 가정한다. AGV 바퀴는 회전축의 바닥면의 수직선분과 직교하며, 또한 바퀴와 바닥사이의 접촉면에서 pure rolling과 non-slipping의 조건을 만족한다. Pure rolling은 미끄러짐이 없는 굴림 운동이라는 뜻으로 바퀴 표면의 점들이 연속적으로 바닥 표면의 점들과 일대일 접촉을 할 때를 말한다. Non-slipping은 나아가는 방향의 수직 성분으로 미끄러지지 않는 것을 말하는데 이동 중 옆으로 미끄러지지 않는다는 것을 의미한다.

2. AGV의 기구학식

왼쪽 바퀴와 오른쪽 바퀴의 각속도를 ω_L, ω_R 이라고 하면 non-slipping 조건에 의해 양 바퀴의 지면과의 접촉점 속도는 아래와 같다.(식 2.1)

$$V_R = r\omega_R, V_L = r\omega_L$$

(단, r은 바퀴의 반지름)

로봇 중심의 선속도를 v, 양 바퀴의 선속도를 각각 V_L, V_R 이라 하고 반시계 방향으로의 AGV 각속도를 ω 라 하면, v, $\omega, \omega_L, \omega_R$ 은 아래와 같은 관계를 가진다.(식 2.2)

$$\omega = \frac{V_R - V_L}{L} = r \frac{\omega_R - \omega_L}{L}$$

$$v = \frac{V_R + V_L}{2} = r \frac{\omega_R + \omega_L}{2}$$

(단, L은 양 바퀴 사이의 거리)

$(x \ y \ \theta)^T$ 와 $(v \ \omega)^T$ 은 다음과 같은 기구학식에 의해 연관 관계를 가진다.(식 2.3)

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

식 2.3을 보면 제어할 수 있는 구동기는 2개인 데 반해 도달해야 하는 위치 및 각도자유도는 3이다. 따라서 현재 AGV 자세에서 원하는 자세로 제어할 때 제한조건이 발생한다[3]. non-slipping을 이용하면 아래와 같은 non-holonomic 제한조건을 구할 수 있다.(식 2.4)

$$H \cdot \dot{p} = (\sin \theta \ - \cos \theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \dot{x} \sin \theta - \dot{y} \cos \theta = 0$$

(단, H는 바퀴에 수직인 단위벡터)

제한된 조건은 식(2.4)와 같이 되며, 이것은 $\tan \theta = \dot{y}/\dot{x}$ 와 같이 정리될 수 있다. 이것은 로봇의 순간 진행방향이 로봇이 향하는 각도 θ 와 같아야 한다는 의미이다.

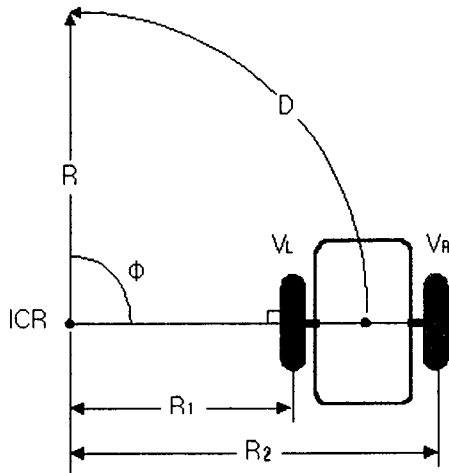


그림 2. AGV의 순간회전중심 및 이동거리, 회전각도

양 바퀴의 속도 V_L 과 V_R 은 그림 2에 표시된 순간회전중심(ICR)에서의 바퀴까지 거리 R_1 , R_2 와 비례한다.

$$V_L : R_1 = V_R : R_2 \quad (\text{식 2.5})$$

$$V_L / (R - \frac{L}{2}) = V_R / (R + \frac{L}{2}) \quad (\text{식 2.6})$$

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} \quad (\text{식 2.7})$$

식 2.7을 보면 AGV가 직진할 경우는 $R = \infty$ 일 경우이며 $V_R = V_L$ 일 때이다. 또한, AGV가 제자리에서 회전(Spin turn)하기 위해서는 $R = 0$ 일 경우이며 $V_R = -V_L$ 일 경우이다.

실제 모터의 회전은 그림 3와 같이 가감속 구간이 있다. 양 바퀴의 가감속 시간이 같다면, AGV는 원호를 그리며 진행하게 된다. 이때 이동거리와 회전반경, 회전각도에 대한 식은 아래

와 같다.

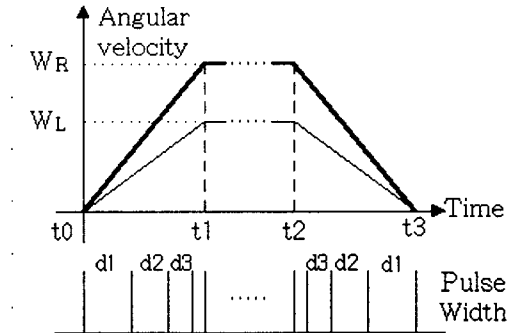


그림 3. 양 바퀴 속도 profile 및 pulse width

이동 거리(2.8) :

$$\begin{aligned} D &= \int_{t_0}^{t_1} \frac{V_L + V_R}{2} dt \\ &= \frac{1}{2} r \frac{\omega_L + \omega_R}{2} (t_3 - t_0 + t_2 - t_1) \end{aligned}$$

회전 반경(2.9) :

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{\omega_R + \omega_L}{\omega_R - \omega_L}$$

회전 각도(2.10) :

$$\begin{aligned} \psi &= \frac{D}{R} \\ &= \frac{r}{2L} (\omega_R - \omega_L) (t_3 - t_0 + t_2 - t_1) \end{aligned}$$

2. 3 AGV 가감속 테이블

실제 모터의 회전은 그림 3과 같이 가속, 등속, 감속 구간을 가지게 된다. 가속, 감속을 위한 펄스폭에 관한 식은 아래와 같이 나타낼 수 있다.

$$d_2 = \frac{d_1}{\sqrt{2}}, \quad d_3 = \frac{d_2}{\sqrt{3}}, \quad \dots, \quad d_n = \frac{d_1}{\sqrt{n}}$$

즉, n배로 가속하기 위해서는 최초 펄스폭의 $1/\sqrt{n}$ 만큼 분주해 주어야 한다.

III. 제어기 설계 및 Line-tracer 구현

3. 1 제어기 설계

위 식에서 계산된 값을 내부 ROM에 LUT(Look-Up Table) 형태로 저장하고 조건에 따라 Table을 참조하여 분주비를 조정하여 양 바퀴 속도를 제어하게 된다. 그림 4는 제안된 제어기의 Block diagram이다.

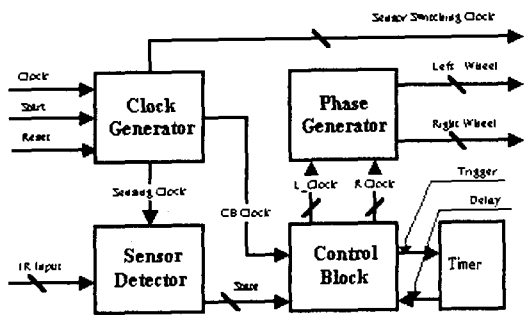


그림 4. AGV 제어기의 블록다이아그램

제어부는 클럭 발생부, 센서 감지부, 제어블록, 상발생부, 타이머로 구성되어 있다. 클럭 발생부는 센서 스위칭, 입력신호 래치, 제어블록 클럭 공급을 한다. 센서 감지부는 스위칭 클럭과 동기된 클럭으로 센서 입력을 감지하고 분석하여 시작, 종료, 가속, 감속, 좌.우회전, 좌.우 기울어짐 등 AGV 상태를 제어블록에 전달한다. 이 신호들이 제어블록에 전달되면 직진, 회전, 자세 보정 등 주행에 필요한 클럭을 LUT를 이용하여 발생시킨다. 상 발생부에서는 제어블록의 가변클럭을 이용하여 스텝모터 구동에 필요한 1-2상 여자신호를 발생시킨다[1]. 그림 5는 제어기의 시뮬레이션 파형이다.

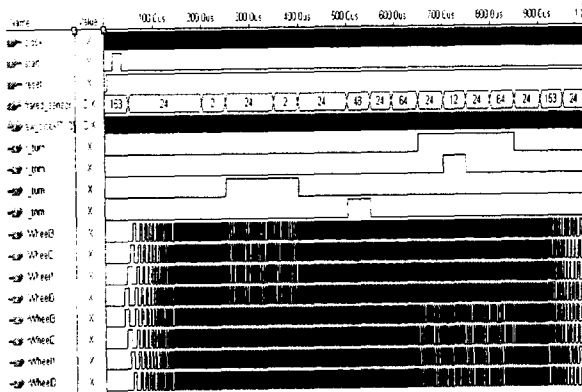


그림 5. 제어기의 시뮬레이션 파형

3.2 센서부 설계

감지부는 AGV가 자율주행하는데 필요한 유도라인을 인식하는데 사용된다. Line-tracer는 8개의 적외선 LED와 포토 TR를 사용하였다. 적외선이 흰색 유도라인에는 반사되고 검은색 바닥에는 흡수되는 성질을 이용하였다. 센서의 동작주파수는 1KHz이며 외부 조명 등 외란 제거를 위

해 400Hz 차단주파수를 갖는 HPF를 사용하였다.

3.3 구동부 설계

디지털 신호로 제어하기 쉬운 스텝모터를 유니폴라 방식으로 제어하였다. 제어기에서 발생한 1-2상 여자신호는 스텝모터 드라이브인 Sanken사의 SLA7024에 인가되어 스텝모터를 구동시킨다[2].

IV. 결론

본 논문에서는 AGV 제어를 위한 CPLD 제어기를 제작하였다. 이 제어기는 기존의 마이크로프로세서로 구현된 제어기에 비해 메모리소자, 센서제어부, 상발생부 등을 포함하고 있어 제어 용이하며 제품의 크기 및 생산가격을 줄일 수 있다. 아래 그림 6은 CPLD 제어기를 사용하여 제작한 Line-tracer이다[4][5]. 실제 모형에 적용해 본 결과 정상적인 주행능력을 확인하였다.

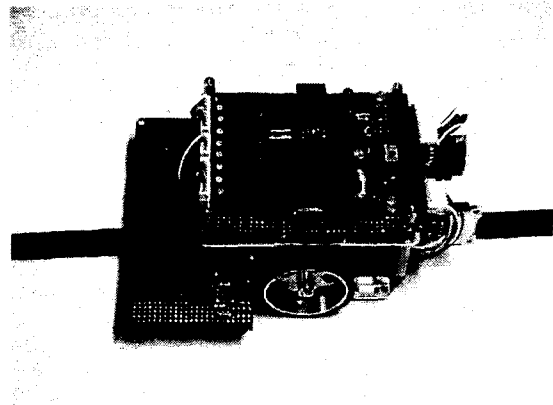


그림 6. CPLD 제어기를 사용한 Line-tracer

참고문헌

- [1] 박세현, "VHDL 기본과 활용", 그린출판사, Pp. 253-258, 1998
- [2] 신정환, "C로 미는 로봇 I", Ohm사, Pp283-326, 1996
- [3] 노봉환 외3. "역학", 희중당, Pp238-240, 1994
- [4] Altera, "Configuring FLEX10K Devices", Application Note 59, August 1998
- [5] Altera, "Configuration EPROMs for FLEX10K Devices", Data Sheet, October 1998