LabVIEW를 이용한 고속 인쇄기의 레지스터 컨트롤러의 개발에 관한 연구

권혁기*, 홍선기* 이덕형*
호서대학교*

A study for register controller of high speed printing machine which uses the LabVIEW

Hyukl-Ki Kwon*, Sun-Ki Hong*, Duck-Hyung Lee*
Hoseo Uni versity*

Abstract - 기존 고속 인쇄기의 인쇄 속도인 250[mpm]의 두 배 속도인 500[mpm]의 고속 인쇄에서도 사용할 수 있는 고성능 레지스터 컨트롤러를 개발함에 있어 LabVIEW라는 GUI기반의 언어 사용하여 개발 시간을 상당히 단축시켰으며, 실제 인쇄와 유사한 시뮬레이션을 통해 기존 컨트롤러의 성능과 비교하고자 한다.

1. 서 론

최근의 그라비어 인쇄 에서는 sectional 타입의 인쇄기가 사용되고 있다. 메인 모터 하나로 구동되는 방식에서 축을 없애고, 각 인쇄 실린더를 개별적인 모터로 구동하는 방식이다. 이 방식의 경우 레지스터 에러를 기존의 compensator roll 타입의 인쇄기에서 compensator roll을 움직여 스팬 길이를 변화를 통하여 에러를 보상한 반면 Sectional 타입의 인쇄기에서는 각 프린팅 유닛의 모터의 속도를 변화를 통하여 에러를 보상한다. 이러한 Sectional 타입의 장점은 고속의 인쇄가 가능하며, 각 인쇄 부분에 대하여 빠른 오차 보정이 가능하고, 주변 Shaft 등을 없애 구조적으로 간단해지는 등의 여러 장점을 가진다.

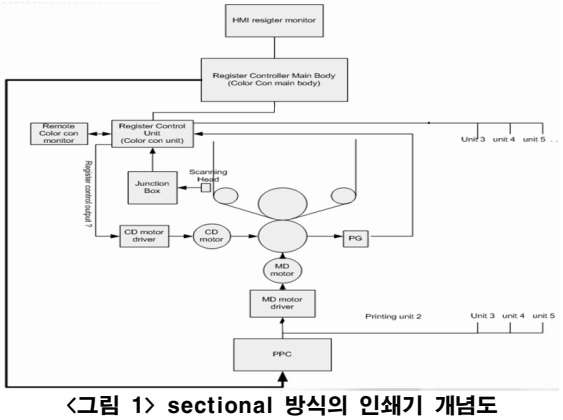
이러한 Sectional 타입에 인쇄기에 적합한 인쇄물의 오차를 판단하고 보정하는 레지스터 컨트롤러를 GUI기반의 LabVIEW와 cRIO 라는 리얼타임 IO 장치를 이용하여 구현해 보고자한다.

2. 본 론

2.1 인쇄기의 기본 동작

최근의 인쇄기는 Sectional 타입의 인쇄기가 많이 사용되고 있다. Sectional 타입의 인쇄기란 메인 모터를 축에 물리는 방식에서 벗어나 각 도 마다의 모터를 개별적으로 구동하여 각각의 오차를 각 도의 모터로 속도제어를 하는 것이다. 이는 각각의 도를 구동하기 위한 모터 드라이버를 사용하여 구동을 하고 이 모터는 PPC를 통해 제어된다. 각 도의 1회전마다 광센서를 통해 원단에 인쇄되는 레지스터 마크를 입력 받고 에러를 생성한 후 인접한 도와의 비교를 통해 오차를 계산한 다음 네트워크를 통하여 레지스터 컨트롤러와 PPC에 전송하게 된다. 레지스터 컨트롤러는 전달 받은 오차의 양을 화면에 출력하여 인쇄 상황을 파악할 수있게 하며 PPC는 전달 받은 오차의 만큼 각 도의 상을 조절하여 오차를 보정하게 된다.

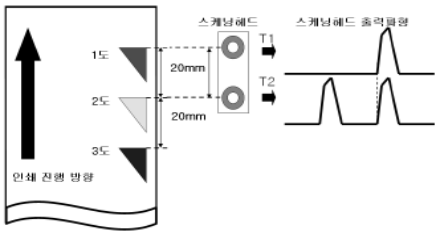
보정 방식은 레지스터 컨트롤러의 Pulse 형태로 보정 파형을 받은 다음 PWM을 이용하여 미리 정해져 있는 각도만큼 각 도를 조정하게 된다. 정해져 있는 보정의 양은 센서의 타입이나 성능 등을 고려하여 조정된다. Sectional 타입의 인쇄기 개념도는 <그림 1>에서 보이는 바와 같다.



<그림 1> sectional 방식의 인쇄기 개념도

2.2. 레지스터 마크 입력 설계

기존보다 발전된 성능의 레지스터 컨트롤러 설계를 위해 기존 방식의 입출력에 관한 정의와 개발될 시스템에 맞는 접목이 필요하다. 인쇄기가 정방향 인쇄를 할 경우 <그림 2>에서 보는 바와 같이 스케닝헤드의 신호를 받는다. 두 개의 각각의 레지스터 마크의 간격은 20mm로 떨어져 있어 동일한 시간대에 신호가 나오면 오차가 없는 것이고 차이가 있으면 오차 값이 발생하는 것이다. T1T2의 신호는 버퍼 IC를 이용하여 구형파로 바꾸어 DSP가 인식 가능하도록 한다. 이 때 인쇄기의 속도를 계산하고 그 속도에 의해 파형의 폭과 파형과 파형의 간격을 결정하여 실제 시스템에 가장 근접한 신호를 만들게 된다.



<그림 2> 레지스터 마크의 센서 신호 인식 개념도

2.3 LabVIEW를 이용한 레지스터 컨트롤러 알고리즘 설계

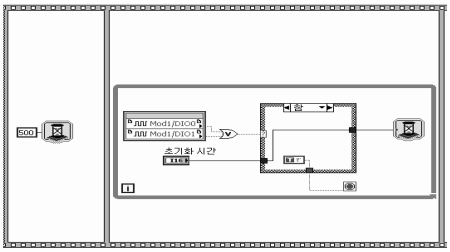
GUI 기반의 LabVIEW를 사용함으로서 개발 시간의 많은 단축을 가져왔다. 텍스트 기반의 C언어를 이용할 때에 비하여 약 10배 이상의 개발에 필요한 구현시간을 가져왔다. 또한 개별적인 UI의 제작이 필요 없는 부분도 상당한 장점이 되었다.

<그림 3>은 FPGA부분의 프런트패널 모습으로 FPGA 동작 시 사용자가 확인할 필요가 있는 부분이나 조정이 가능 한 부분을 나타내며, 이 부분을 통해 인쇄기가 실제 동작 시 데이터의 확인 및 조정이 가능하다.

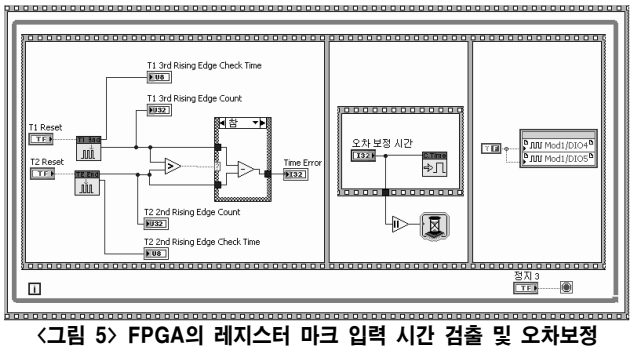
<그림 3> FPGA의 프런트패널의 모습

<그림 4>와 <그림 5>는 FPGA부분의 레지스터 마크의 입력에 관한 블록 다이어그램의 모습이다. 레지스터 컨트롤러 동작의 초기화하는 부분으로 인쇄기의 동작의 초기화 하는 부분으로 인쇄기가 최초 동작 시 발생 가능한 레지스터 마크의 입력 오류를 막기 위한 부분, 레지스터 컨트롤러의 초기화 이후 T1의 두 번째 상승 Edge 검출 시간과 T2의 세 번째 Edge의 검출 시간을 체크하여 오차 시간을 확인하는 부분, cRIO에서 연산된 오차보정 시간을 받아 정확한 시간만큼 보정 파형을 출력하는 부분과 보정 파형 출력 이후 보정 파형 출력을 종료하는 부분을 나

타낸 모습이다.

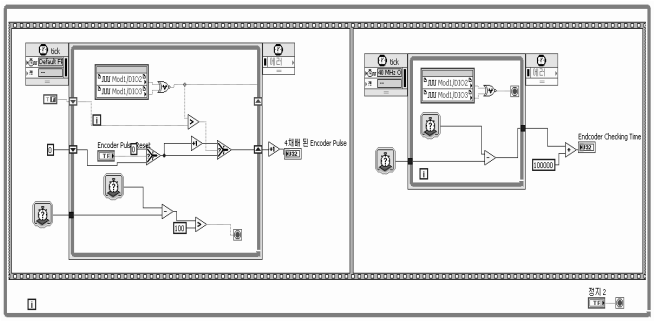


〈그림 4〉 FPGA의 레지스터 마크 입력 초기화 블록다이어그램



〈그림 5〉 FPGA의 레지스터 마크 입력 시간 검출 및 오차보정

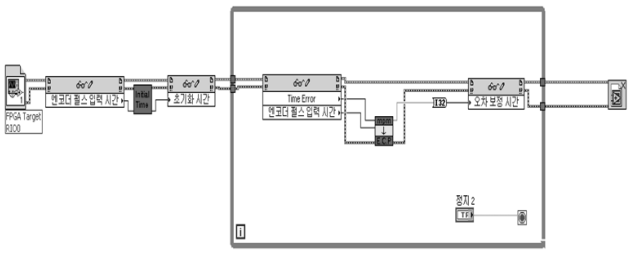
<그림 6>은 Encoder Pulse를 4채배하여 지정한 Edge의 수가 되면 그 시간을 검출하는 부분으로 <그림 4>와 <그림 5>의 오차 검출 및 보정 부분과 독립적으로 동작한다.



〈그림 6〉 FPGA의 Encoder Pulse 입력 시간 검출

2.4. cRIO를 이용한 오차 보정 작업

FPGA가 수행될 동작들이 컴파일 완료되면, cRIO는 FPGA가 동작하는 데 필요한 데이터 들을 연산하여 FPGA에 입력하는 역할을 한다. <그림 7> cRIO의 동작을 보여주는 블록다이어그램이며, cRIO에서는 FPGA에서 수집된 데이터들을 가지고 연산하여 오차보정에 필요한 시간들을 입력하는 역할을 한다.

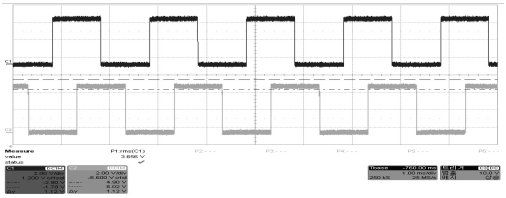


〈그림 7〉 cRIO 동작 블록다이어그램

2.5 제작 레지스터 컨트롤러의 시뮬레이션

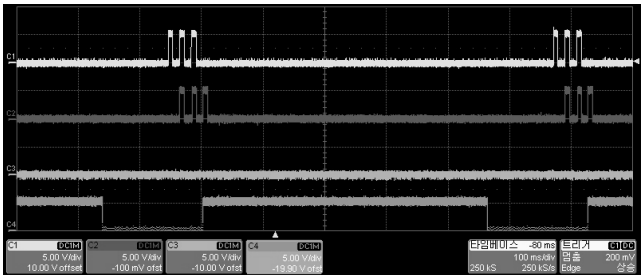
FPGA와 cRIO에 대한 설계가 완료 된 후 실제 인쇄기를 동작 시킬 경우 발생되는 파형들을 임의 파형 발생기를 통해서 제작해보았다.

<그림 8>의 경우 Encoder에서 발생되는 A,B상을 발생시켜 본 모습이다. 인쇄기에 장착된 Encoder의 경우 1회전에 512개의 펄스가 발생되며, 4채배 했을 경우 2048개의 펄스로 구별 할 수 있다. 또한 이 신호의 주기를 변경 시켜 mpm을 변환 가능하다. 이 신호를 가지고 cRIO 모듈에 입력하여 MT method를 이용하여 정확한 속도 검출여부를 판단하였다.



〈그림 8〉 Encoder Pulse A상과 B상의 모습

<그림 9>는 임의 파형 발생기를 통해 제작한 T1,T2의 신호 +오차 보정 신호, - 오차 보정 신호이며, 오차여부를 판단할 수 있도록 T2신호에 대하여 약 1.5msec의 오차를 주어 오차 검출 여부를 시뮬레이션 해보았다. 인쇄기의 Encoder Pulse와 T1,T2신호를 같이 입력하여 인쇄기의 속도와 레지스터 마크의 오차를 검출한 후 정확한 오차 보정 신호를 출력하는지 확인하기 위한 시뮬레이션 모습이다.



〈그림 9〉 T1, T2신호 입력 시 보정 파형의 발생 모습

<그림 10> cRIO 프런트패널의 동작 모습을 보여주는 그림이다. <그림 8>과 <그림 9>에서 입력되는 신호들을 입력받아 오차와 속도를 계산하여 보정 파형의 출력 시간이 나온 모습을 보여준다. 이 데이터와 오실로스코프를 이용해 직접 파형의 시간차를 확인한 결과 동일함을 볼 수 있었다.

〈그림 10〉 cRIO의 프런트패널 동작 모습

3. 결 론

LabVIEW를 통해 기존 PPC에서 하는 동작들을 동일하게 구현하고자 하였다. LabVIEW를 사용함으로서 개발 시간을 텍스트 기반의 언어를 사용할 때와 비교하여 1/10 정도로 단축시킬 수 있었다. 그리고 FPGA를 이용하여 입출력 설정을 빠르게 할 수 있었다. 또한 cRIO를 통한 실시간 입출력 제어가 가능했다. 이를 통해 기존 레지스터 컨트롤러의 동작과 비교하였을 때 동일 한 성능을 나타내는 것을 확인 할 수 있었다.

추가적으로 인쇄기와 연동 실험을 통한 레지스터 컨트롤러의 동작을 확인 후 기존 레지스터 컨트롤러의 기능뿐 아니라 추가적인 기능을 부여 할 수 있을 것으로 예상된다.

[참 고 문 헌]

[1] 설승기, “전기기기 제어론”, 도서출판 브레인 코리아, 2002

[2] 장중학, “고속 인쇄기의 레지스터 컨트롤러의 오차 보정에 관한 연구”,대한전기학회 하계학술대회,1809~1810, 2006

[3] 권혁기, “TMS320F2812를 이용한 고속 인쇄기의 레지스터 컨트롤러의 오차 보정 개선에 관한 연구”,대한전기학회 하계학술대회, 1581~1582, 2007

[4] NATIONAL INSTRUMENTS, "Data Acquisition & Signal Conditioning", NATIONAL INSTRUMENTS, 2003

[5] NATIONAL INSTRUMENTS, "CompackRIO and LabVIEW Development Fundamentals", NATIONAL INSTRUMENTS, 2007