레졸버 기반의 절대위치 검출 센서 드라이버의 FPGA 구현

FPGA Implementation of Resolver-based Absolute Position Sensor Driver

전 지 혜, 신 동 윤, 양 윤 기, 황 진 권, 이 창 수*
(Ji-Hye Jeon, Dong-Yun Shin, Yoon-Gi Yang, Jin-Kwon Hwang, and Chang-Su Lee)

Abstract : Absolute position detector which is one of the major equipment in the field of factory automation, not only perceives the absolute position of the rotary machine but also outputs switch data according to the given angle. Absolute position detector is composed of sensor module and its controller. In this paper, a sensor driver is implemented using FPGA with VHDL. This chip has a less form factor than conventional circuit. A test shows reliable precision within THD(total harmonic distortion) of 0.2% which can be applicable commercially. Also, FPGA-based phase error compensation methods were newly discussed. In the future, more research will be conducted to enhance the precision by the introduction of 3-phase transformer.

Keywords : RVDT(Rotary Variable Differential Transformer), FPGA, VHDL

I. 서론

산업현장에서 사용하는 산업용 장비들의 정밀도를 높이기 위한 노력은 끊임없이 이루어지고 있다. 장비의 정밀도는 신뢰성과도 직결되어 있기 때문에 그만큼 중요한 것이다. 이러한 산업용 장비 중 변압기는 많은 분야에서 사용되고 있는 중요한 장비이며 이에는 여러 종류가 있다. 그 중에서도 회전형 가변 차동 변압기(RVDT)는 기기의 절대 위치를 검출하는 트랜스듀서로 회전하는 변압기의 원리를 이용한 것이다[1]. 또한 RVDT는 절대 회전각을 검출하는 센서로서 일반적으로 산업용 모터의 축에 장착되어 모터의 절대 회전각과 회전 속도를 검출하는 데 사용하고 있다. 모터의 절대 회전각을 측정하기 위해서는 R/D(RVDT to digital) 변환기를 사용한다[8]. 이 R/D 변환기는 센서 드라이버 또는 센서 컨트롤러라고도 하며 센서를 구동하는 입력 파형인 정현파와 여현파를 생성한다. 모터에 장착되어 있는 회전자의 회전에 따른 출력 파형과 입력 파형을 비교하여 위상차에 대한 정보를 알게 되면 모터의 회전각을 알 수 있다[2,3]. 이것은 파형의 교차되는 지점, 즉 영점 교차(zero-crossing)에 대한 위치 정보를 알게 되는 것과 같다[6,7]. 이러한 역할을 하는 R/D 변환기를 VHDL 언어를 기반으로 한 FPGA 단일 칩으로 구현함으로써 기계적인 오차에 대한 보상을 통해 정밀도를 높일 수 있다. FPGA 단일 칩의 구현은 설계 및 수정을 용이하게 하고 여러 부품을 사용하는 것보다 면적을 줄일 수 있다. 뿐만 아니라 개발 비용이 감소하고 개발 시간도 단축할 수 있는 장점을 가진다[4,5]. 하지만 이렇게 FPGA로 구현하였을 경우에도 소자의 정밀도로 인한 오차가 발생하는 것을 실험적으로 알 수 있다.

본 연구에서는 센서 드라이버를 FPGA 단일 칩으로 구현하여 실험 하였다. RVDT 센서 드라이버는 컨트롤러의 종류 및 구성에 따라 여러 가지로 분류를 할 수 있는데 본 연구에서는 단회전(single-turn) RVDT 센서 드라이버와 다회전(multi-turn) RVDT 센서 드라이버를 구현 및 실험한다. 또한 FPGA 기반의 위상 오차 보상 방법에 대해 제안하고 실험한다.

II. 센서 드라이버의 원리와 FPGA의 구현

1. 단회전 RVDT 센서 드라이버

센서 드라이버(R/D 변환기)는 RVDT 센서를 구동하기 위해서 정현파와 여현파를 생성한다. 이 파형들을 RVDT 센서의 입력신호로 인가하며 RVDT 센서에서 나오는 출력 파형과 비교하여 위상차에 대한 정보를 검출하는 역할을 한다. RVDT 센서로 보내지는 두 입력 파형을 생성하기 위해서는 LUT(Look-Up Table)과 DAC를 이용하게 된다[9]. 이 입력 파형의 정밀도를 높여주는 것 역시 중요하므로 VHDL을 기반으로 하는 FPGA 단일 칩으로 구현한다. 실제 구현에는 저가의 CPLD 칩을 이용하였다. 전체적인 흐름을 살펴보면 입력 파형으로 쓰일 정현파 또는 여현파의 정보를 가진 데이터를 ROM의 LUT를 이용하여 센서의 입력으로 전

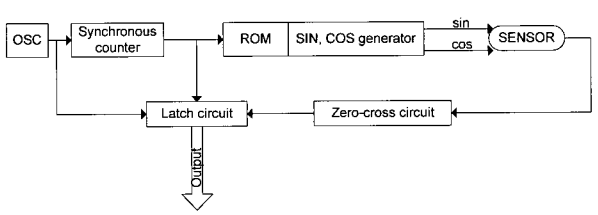


그림 1. 단회전 RVDT 센서 드라이버의 블록 다이어그램.
Fig. 1. Block diagram of single-turn RVDT sensor driver.

* 책임저자(Corresponding Author)
논문접수 : 2007. 8. 5., 채택확정 : 2007. 9. 1.
전지혜, 신동윤, 이창수 : 수원대학교 전자공학과
(jhjeon@suwon.ac.kr/blackdoy@nate.com/cslee@suwon.ac.kr)
양윤기 : 수원대학교 정보통신공학과(ygyang@suwon.ac.kr)
황진권 : 우석대학교 소방안전학과(jkhwang@woosuk.ac.kr)

달하고 센서의 출력 파형을 영점 교차회로를 거친 후 그 정보를 CPLD로 전달하면 그 순간의 위상차 데이터를 래치에 저장되고 외부로 출력하여 컨트롤러가 받는 것이다.

단회전 RVDT 센서 드라이버의 블록 다이어그램은 위의 그림 1과 같이 구성된다. Isin(wt)와 Icos(wt)의 1차 여기 전압의 데이터는 ROM의 LUT에 저장되어 있으며 ROM 앞에 연결된 동기 카운터의 값이 어드레스 값으로 사용된다. 이 ROM의 두 파형을 D/A변환과 증폭을 거쳐서 RVDT 센서의 1차 여기 입력으로 만드는 것이다. 2차 여기 입력은 영점 교차 회로가 만든다. 영점 교차 회로를 거치기 전 증폭 및 노이즈 제거를 하기 위한 필터링 과정을 거치게 된다. 파형의 영점교차가 이루어지는 순간 1차, 2차 여기 전압의 위상차 데이터는 래치에 저장되고 외부로 출력되어 컨트롤러가 받는 것이다. 입력 오실레이터의 주파수 값을 f_x, 동기 카운터 값을 N, 1차 여기 전압의 캐리어 주파수 값을 f_c라 하면 그 관계는 (1)과 같이 나타낼 수 있다.

$$f_x = N \times f_c \tag{1}$$

$$\frac{\theta_{\max}}{N} \tag{2}$$

$$N = \frac{primary\ exicitation\ wave\ period(t = \frac{1}{f} = \frac{2\pi}{w})}{clock\ pulse\ frequency(t_0 = \frac{1}{f_0})} \tag{3}$$

위상차를 θ라고 한다면 분해능은 (2)를 이용하여 구할 수 있다. 이 때 동기 카운터 값 N은 (3)을 이용하여 구할 수 있다. N의 값을 설정함에 따라서 분해능의 크기를 조절할 수 있는데 이 N값을 조정할 때에는 오실레이터 주파수나 시스템적인 상황도 변한다는 것을 고려해야 한다. 본 연구에서 사용한 단회전 RVDT 센서의 파라미터 값은 N= 1440, f_c =5kHz f_x =7.2MHz이다.

단회전 RVDT 센서 드라이버에서 RVDT 센서 컨트롤러로 출력 신호를 내보낼 때 컨트롤러가 이 신호를 받는 방법은 2가지로 볼 수 있다. 하나는 LSB_EN, MSB_EN 핀을 이용하여 high시 LSB, MSB를 출력하여 8bit의 데이터를 2번 읽어 들이는 방법이고 또 다른 하나는 센서 드라이버와 RVDT 센서의 핀 연결처럼 한번에 11bit의 데이터를 읽어 들이는 방법이다. 본 연구에서는 2가지 방법으로 신호를 획득하였는데 이것은 한 주기인 360°를 1440으로 샘플링 하는 것을 의미한다. 즉 0.25°간격으로 데이터를 읽어 들이게 된다. 또한 init핀의 신호가 high에서 low로 떨어지는 순간을 인터럽트로 받아들여 11bit의 데이터를 5khz의 init 주파수의 간격으로 읽어 들이게 되는 것이다. RVDT 센서 컨트롤러는 init핀의 인터럽트 신호로 모든 동작을 하게 된다. Busy핀은 입력과 출력의 파형이 영점 교차하는 상황에서 신호가 발생하게 된다. 이 busy신호가 발생한 후에 일정한 주기의 경과 후 안정된 값이 래치 되도록 구성한다. Busy신호의 값이 low일 때 RVDT 센서는 데이터를 가져가도록 구성되었다. 또한 이 busy신호는 입, 출력 파형의 영점 교차의 위치에 따라서 주기가 변하게 된다.

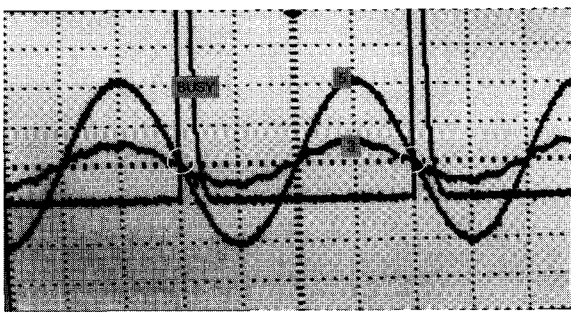


그림 2. 센서 드라이버의 입, 출력 파형과 BUSY 신호.

Fig. 2. Input and output waves of sensor driver and busy signal.

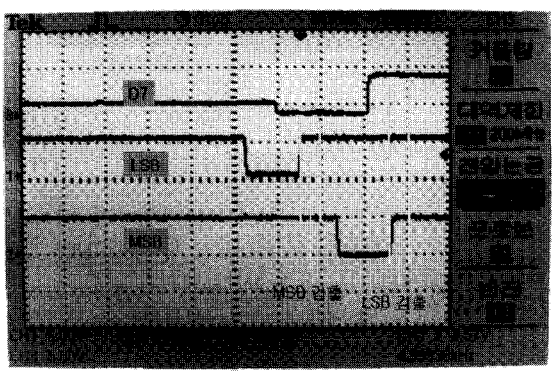


그림 3. 센서 드라이버의 데이터 신호와 RVDT로부터 들어오는 LSB, MSB enable 신호.

Fig. 3. Count data of sensor driver and LSB, MSB enable signal from RVDT.

그림 2는 센서 드라이버의 입·출력 파형과 이 파형들이 영점 교차하는 순간의 busy 신호가 high로 되는 것을 실험적으로 확인한 것이다. Fault핀은 RVDT 센서가 연결이 끊어졌거나 입력파형이 불안정하게 들어올 경우 신호가 high에서 low로 떨어지게 되는데 컨트롤러는 이것을 감지하고 에러를 표현한다. 11bit의 데이터 신호를 읽어 들일 때에는 busy신호의 발생여부에 따라서 데이터 신호를 래치 시키면 되지만 8bit의 데이터 신호를 2번 읽어 들이는 경우에는 조금 다르게 구성해야 한다. 이 방법의 경우 RVDT 센서로부터 드라이버로 들어오는 LSB, MSB의 enable 신호가 데이터를 읽어 들이는 데 중요한 역할을 한다.

그림 3은 8bit의 데이터를 2번 읽어 들이는 경우를 실험적으로 확인한 것이다. LSB 신호가 enable되면 하위비트 8bit인 DB0부터 DB7까지 래치 되서 컨트롤러로 출력 되고 MSB 신호가 enable이 되면 나머지 상위비트인 DB8부터 DB10까지 래치 되어 컨트롤러로 출력 된다.

2. 다회전 RVDT 센서 드라이버

다회전 RVDT 센서 드라이버의 블록 다이어그램은 그림 4와 같이 구성될 수 있다. 센서 드라이버에서 생성되는 정현파, 여현파의 주파수는 10kHz이고 2^{17}의 분해능으로 RVDT 센서에 신호를 보내도록 구성한다. 이렇게 구성된 다회전 RVDT 센서의 경우 32회전까지 검출이 가능하기 때문에 센서의 회전수에서 2^5의 분해능이 결정되고 센서 1

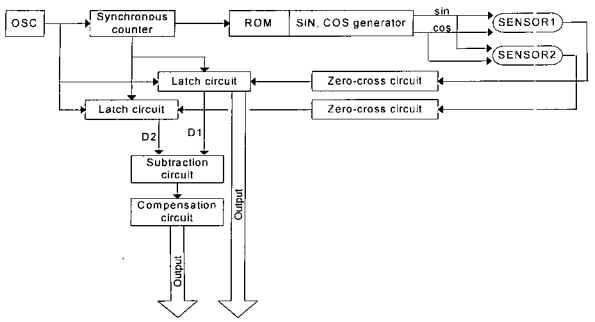


그림 4. 다회전 RVDT 센서 드라이버의 블록 다이어그램.
Fig. 4. Block diagram of multi-turn RVDT sensor driver.

(S1)의 출력 신호를 이용한 한 회전 내 분해능으로는 $2^{12}=4096$이 필요한 것을 알 수 있다.

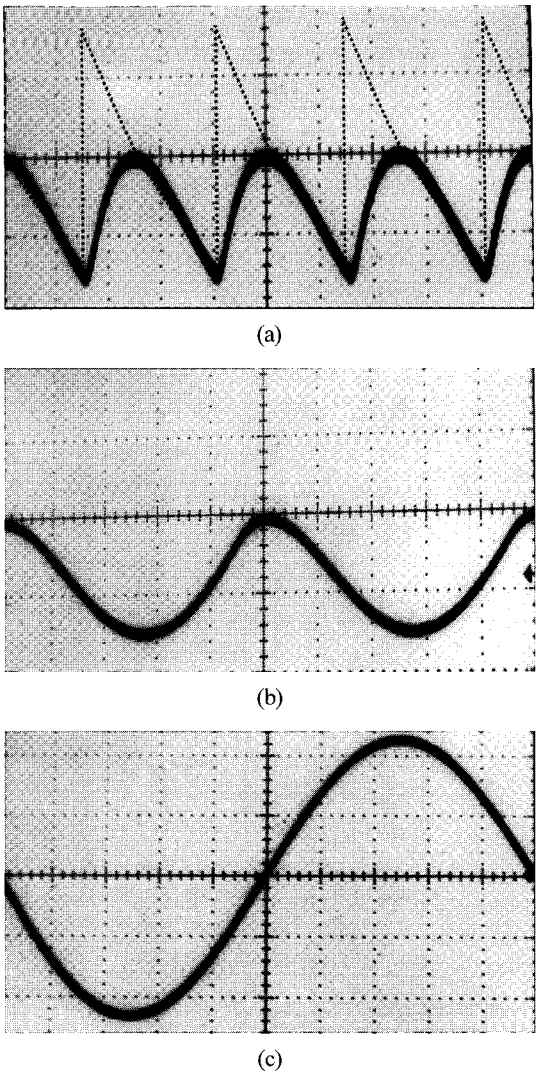
단회전 센서 드라이버의 경우에서 살펴본 것처럼 다회전 센서 드라이버에서도 전체 회로 구성에서 필요한 파라미터 값인 f_x, N 그리고 f_c를 계산해볼 필요가 있다. $N=2^{12}$이고 $f_c=10$kHz이므로 $f_x=40.96$MHz의 오실레이터 주파수를 계산할 수 있다. 위의 그림에서 볼 수 있는 synchronous counter는 정현파, 여현파 생성을 위한 클럭을 만들어 주고 RVDT 센서의 출력 데이터의 분해능을 위한 클럭을 만든다. 이 때 RVDT 센서의 입력 파형을 만들기 위해 필요한 클럭은 정현파 1 주기(0°~359°)를 512로 분해하는 것을 의미하므로 10kHz × 512 = 5.12MHz가 된다. 이런 모든 것이 VHDL로 설계가 되며 메인 오실레이터 주파수가 $f_x=40.96$ MHz이므로 클럭이 4번 발생할 때마다 정현파, 여현파를 생성한다.

정현파, 여현파를 생성하는 방법으로는 2가지가 있다. 첫 번째 방법은 0°~359°의 1주기를 512로 분해하는 LUT를 만드는 방법이고 두 번째 방법은 0°~89°($\frac{1}{4}$주기)를 128로 분해하는 LUT를 만드는 방법이다. 이 두 가지를 이용하여 정현파를 만들어 낼 수 있는데 본 연구에서는 후자의 방법이 전자의 방법에 비해 FPGA로 구현할 때 게이트를 적게 쓰기 때문에 후자의 방법을 이용하여 구현한다.

III. FPGA로 구현한 센서 드라이버의 실험

그림 5는 FPGA로 정현파를 생성할 때 단계별로 파형을 만드는 과정을 보여준다. 그림 5(a)는 점선과 같이 1/4 주기인 0°~89°의 파형이 반복되게 나타낸 것이다. 그림 5(b)는 (a)의 반복된 파형 중 2번째 주기의 파형의 데이터를 역순으로 배열시켜 1/2주기의 파형을 만든 것을 실험적으로 보여준다. 이렇게 만들어진 파형은 다시 2번째 주기의 파형의 데이터를 음 위상에서 양 위상으로 반전시켜 그림 5(c)와 같이 완성된 정현파의 형태를 만들게 되는 것이다. 이렇게 구현된 정현파는 THD를 통하여 이상적인 정현파와 어느 정도 차이가 있는지를 알 수 있다.

표 1은 본 연구에서 실험을 통해 확인해 본 단회전, 다회전 RVDT 센서 입력으로 사용되는 정현파에 대한 THD 데이터를 정리해 놓은 것이다. THD는 왜곡률이기 때문에



(a)

(b)

(c)

그림 5. 정현파의 단계별 생성.
Fig. 5. A step-by-step generation of sine wave.

표 1. RVDT 센서 입력 정현파의 THD 데이터.
Table 1. THD data of RVDT sensor input sine wave.

항목	주파수	THD
단회전 입력	5kHz	0.2%
다회전 입력	10kHz	0.1%

작을수록 정현파의 왜곡률이 낮다는 의미로 이상적인 파형에 가깝다는 의미를 가진다. 본 실험에서 획득한 THD는 현재 시중에 나와 있는 타사 제품과 비교해 보았을 때 단회전의 경우 0.2%, 다회전의 경우 0.1% 이하의 낮은 값을 갖는다. RVDT 센서의 입력 파형으로 사용하기에 신뢰성이 있으며 어느 정도 안정적이라는 것을 확인할 수 있었다.

그림 6에서는 정현파와 여현파의 진폭차이를 확인할 수 있다. 두 파형의 여기 전류의 진폭 차이는 RVDT 센서의 위상 오차에 영향을 미친다. 따라서 위상 오차를 줄이기 위해서는 두 파형의 진폭을 같게 만들어야 한다. 그러기 위해서는 정현파와 여현파의 증폭 회로에서의 변화가 필요하다.

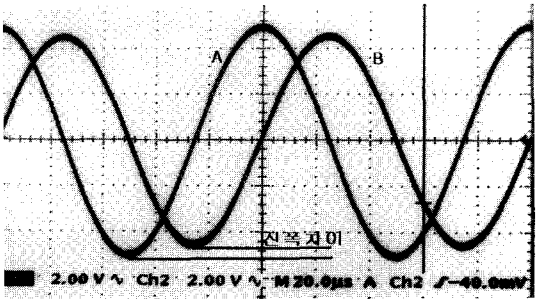


그림 6. 진폭차이가 있는 정현파와 여현파.
Fig. 6. Sine and cosine wave with the amplitude difference.

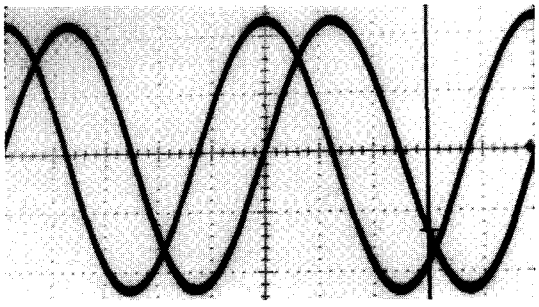


그림 7. 그림 6을 교정한 정현파와 여현파.
Fig. 7. Revised sine and cosine wave of Fig. 6.

정현파, 여현파를 생성하는 실제로 구성한 증폭 회로에서의 진폭 차는 회로를 구성하는 소자들의 오차로 인한 것이기 때문에 두 파형의 진폭을 서로 같게 만들어 주기 위해서는 여현파의 증폭 회로부분의 가변저항을 이용하는 것이다. 그림 7은 진폭을 같게 보상해 준 정현파와 여현파이다.

진폭을 보상하여 두 파형을 생성하는 증폭 회로 뒤에는 추가적으로 RVDT센서와 센서 드라이버의 연결 상태를 확인하는 회로도가 붙게 된다. 이 회로도는 연결 여부에 따라 FAULT신호를 제어하게 되는 것인데 연결된 상태에서는 high 신호를 출력하고 단절된 상태에서는 low 신호를 출력한다.

이렇게 생성된 정현파와 여현파는 RVDT 센서로 들어가게 되고 입력 정현파와 비교해 센서가 회전한 위상차를 갖는 파형을 출력한다. 이 센서의 출력 파형을 이용해 위상 데이터 값을 출력해야 하는데 이 로직에서 위상 데이터 값을 2개로 나누어 출력하게 된다. D0~D11의 12bit 데이터는 만들어진 정현파와 RVDT 센서의 출력 정현파의 위상차를 체크하여 한 주기의 위상인 2π를 12bit로 분해하여 출력한 것이다. D12~D16의 5bit 데이터는 2개의 센서의 출력 정현파 간의 위상차를 체크하여 32회전의 회전 수 데이터를 출력한다. 이 때 센서의 출력 정현파는 아날로그 신호이기 때문에 FPGA에서 정확히 읽을 수 없다. 따라서 센서 출력 파형들은 영점 교차 회로를 거치게 하여 펄스 모양처럼 만들어 주어야 한다. 센서의 출력 파형은 입력 파형과 비교하여 센서 축이 회전한 위상만큼의 위상차를 보이므로 한 주기 내 위상 데이터를 읽어 들이는 초기 시작점을 같게 한다면 절대 위치 검출은 가능해진다. 그러므로 영점 교차 회로를 거친 신호가 low로 변하는 순간 데이터를 읽어 들여 한 회전 내 절대 위상 데이터를 검출한다.

다회전 RVDT 센서의 경우 센서 2개가 각각 출력 파형을 내보내므로 각 센서가 회전한 만큼 위상 검출이 가능하다. 2개의 센서 S1과 S2의 기어톱니 수의 비가 31:32로 설정되어 있다. 센서가 한 바퀴를 돌 때 위상 데이터는 한 바퀴 내 위상 2π가 4096으로 분해되어 출력된다고 한다면

표 2. 다회전 센서의 초기점 맞추는 단계.
Table 2. The stage to set the initial spot of multi-turn sensor.

1. 스위치를 OFF 상태로 둔다.
2. 센서를 돌려 위상 데이터를 0으로 한다.
3. 다회전 센서의 기어 결속을 푼다. 그리고 센서를 돌리면 S2만 회전하게 된다.
4. 스위치를 ON 상태로 바꾼다.
5. 센서를 돌려 위상 데이터를 0으로 한다.
6. 다회전 센서 결속을 하고 스위치를 OFF로 한다.

표 3. 센서 간 위상차에 따른 회전 수.
Table 3. The rotation number according to phase difference of sensors.

위상차(count)	위상차 X(rad)	회전수
0000~0127	$0\pi/16 \le x < 1\pi/16$	0
0128~0255	$1\pi/16 \le x < 2\pi/16$	1
0256~0383	$2\pi/16 \le x < 3\pi/16$	2
0384~0511	$3\pi/16 \le x < 4\pi/16$	3
0512~0639	$4\pi/16 \le x < 5\pi/16$	4
0640~0767	$5\pi/16 \le x < 6\pi/16$	5
0768~0895	$6\pi/16 \le x < 7\pi/16$	6
0896~1023	$7\pi/16 \le x < 8\pi/16$	7
1024~1151	$8\pi/16 \le x < 9\pi/16$	8
1152~1279	$9\pi/16 \le x < 10\pi/16$	9
1280~1407	$10\pi/16 \le x < 11\pi/16$	10
1408~1535	$11\pi/16 \le x < 12\pi/16$	11
1536~1663	$12\pi/16 \le x < 13\pi/16$	12
1664~1791	$13\pi/16 \le x < 14\pi/16$	13
1792~1919	$14\pi/16 \le x < 15\pi/16$	14
1920~2047	$15\pi/16 \le x < 16\pi/16$	15
2048~2175	$16\pi/16 \le x < 17\pi/16$	16
2176~2303	$17\pi/16 \le x < 18\pi/16$	17
2304~2431	$18\pi/16 \le x < 19\pi/16$	18
2432~2559	$19\pi/16 \le x < 20\pi/16$	19
2560~2687	$20\pi/16 \le x < 21\pi/16$	20
2688~2815	$21\pi/16 \le x < 22\pi/16$	21
2816~2943	$22\pi/16 \le x < 23\pi/16$	22
2944~3071	$23\pi/16 \le x < 24\pi/16$	23
3072~3199	$24\pi/16 \le x < 25\pi/16$	24
3200~3327	$25\pi/16 \le x < 26\pi/16$	25
3328~3455	$26\pi/16 \le x < 27\pi/16$	26
3456~3583	$27\pi/16 \le x < 28\pi/16$	27
3584~3711	$28\pi/16 \le x < 29\pi/16$	28
3712~3839	$29\pi/16 \le x < 30\pi/16$	29
3840~3967	$30\pi/16 \le x < 31\pi/16$	30
3968~4095	$31\pi/16 \le x < 32\pi/16$	31

S1과 S2의 초기점이 0으로 같게 설정된 상태에서 다회전 센서가 한 바퀴 돌 때 S1은 0으로 돌아오지만 S2는 32 기어이므로 한 회전의 1/32만큼 차이가 발생한다. 이 차이를 이용해 다회전 센서의 회전수를 검출할 수 있다. 이 경우 센서간의 위상차로 회전 수 검출이 가능하지만 센서 S1, S2의 초기 시작점이 같아야 하는 초기 조건이 필요하다. 초기 시작점이 같은 이상적인 동작에서는 S1이 초기점으로 돌아오는 순간 회전수가 바뀌어야 하는데 둘의 초기 점이 다른 경우 S1이 초기점으로 돌아오는 순간 두 센서의 위상차는 1/32를 벗어날 수 있다. 예를 들어 S1의 초기 시작점은 0이고 S2의 초기 시작점이 16이라면 다회전 센서의 반 바퀴 회전만으로 회전수가 바뀔 수 있다. 실제로 제작된 센서에서는 초기점을 파악할 수 없으므로 회로 상으로 LED와 스위치를 출력부분에 추가하여 확인할 수 있다. 스위치를 OFF할 때는 센서 S1의 위상 데이터와 회전수가 출력되고 ON이 되면 S2의 위상 데이터와 회전수가 출력된다. 표 2는 초기점을 맞추는 단계이다.

실제로는 위의 과정대로 한다고 하도 센서 자체의 오차가 1° 정도 존재하므로 약간의 오차 보정을 해야 한다. 한 회전 내 카운터는 4096(2^{12}분해능) step을 가지며 위상 데이터가 증가함에 따라 센서 간 위상차도 커진다. (4)는 센서 간 위상차를 구하는 공식이다. 여기서 P는 회전 내 위상 데이터, R은 회전수이다. 표 3은 두 센서 간 위상차에 따른 회전수를 나타낸 것이다.

$$X = \frac{P}{32} + (128 \times R) \tag{4}$$

위에서 언급한 것처럼 센서 자체의 오차로 인하여 회전 내 위상 데이터 D0~D11이 0이 되는 순간과 회전 수 데이터 D12~D15가 변하는 시점이 정확히 맞지 않는다. 예를 들면 표 3에서 볼 수 있듯이 이상적인 회전에서는 2회전의 초기 카운터 값은 256인데 실제로 오차가 약 40정도 발생한다고 하면 카운터는 216이 되고 결국 1회전으로 인식 되는 것이다. 이러한 오차 보정을 위해서 회전 내 위상 데이터를 8곳으로 분할하여 회전 내 위치에 따라서 위상 차 데이터를 보정해준다. 표 4는 회전 내 위치에 따른 위상 차 데이터의 보정 계산 값이다. 이러한 방법으로 보정할 경우 안정적인 값으로 인식되어 적합한 데이터를 출력할 수 있다.

표 4. 회전 위치에 따른 위상차 데이터 보정.

Table 4. Phase difference data revision according to a rotational location.

회전 내 카운터(count)	센서 간 위상차 보정 계산(count)	센서 간 위상차 보정 계산(rad)
0000~0511	위상차+64	위상차+$\frac{1}{32}\pi$
0512~1023	위상차+48	위상차+$\frac{3}{128}\pi$
1024~1535	위상차+32	위상차+$\frac{1}{64}\pi$
1536~2047	위상차+16	위상차+$\frac{1}{128}\pi$
2048~2559	위상차-16	위상차-$\frac{1}{128}\pi$
2560~3071	위상차-32	위상차-$\frac{1}{64}\pi$
3072~3583	위상차-48	위상차-$\frac{3}{128}\pi$
3584~4095	위상차-64	위상차-$\frac{1}{32}\pi$

IV. FPGA 기반의 오차 보상 알고리즘

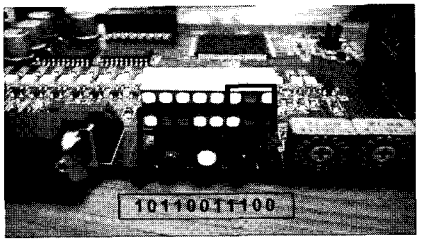
1. RVDT 입력 파형의 오차 보상

앞에서 언급한 것처럼 FPGA로 구현된 센서 드라이버는 RVDT 센서를 구동하기 위해서 정현파와 여현파를 생성한다. 정현파와 여현파의 위상차를 90°로 정확히 맞추는 것은 안정적인 데이터 출력을 위한 중요한 과정이다. 다회전 RVDT 센서 드라이버의 경우, 90°의 정확한 위상차를 맞추기 위하여 정현파의 한 회전(360°) 내 분해능인 4096의 step 값을 갖는데 이것을 4로 나누어 90°를 1024 카운터의 분해능 값으로 표현할 수 있다. 이론적으로 여현파와 정현파는 (5)와 같이 나타낼 수 있다.

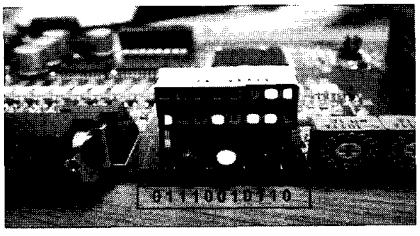
$$\cos\text{위상}\,step = \sin\text{위상}\,step + 1024step(90^\circ\text{에 해당}) \tag{5}$$

그러나 실험에서 두 파형의 위상차는 소자 간의 오차로 인하여 90°에서 벗어난다. 총 12bit 가운데 최하위 비트(LSB)는 감지된 데이터의 안정성 고려를 위해 소프트웨어적으로는 사용하지만 하드웨어 실험에서는 불필요하므로 표현을 생략한다. 그래서 상자로 표시한 것처럼 11bit가 표시된다. 또한 정현파와 여현파의 위상차가 90°(1024step)가 정확한지 관심을 갖는 것이기 때문에 11bit 중에서 최상위 2bit 역시 회전수에 관련된 bit이므로 고려하지 않았다.

그림 8과 같이 실제 실험에서 정현파와 여현파의 위상차가 1024 step에서 벗어났음을 알 수 있다(110011100-110010110=110(0), 상위 2bit는 고려하지 않음). 즉, 두 파형이 12



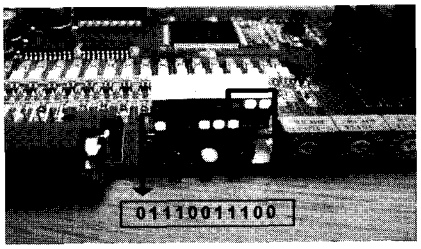
(a) Non-revised value of sine wave



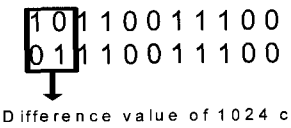
(b) Non-revised value of cosine wave

그림 8. 파형 보상 전의 정현파와 여현파.

Fig. 8. The picture of two waves before wave compensation.



(a) Compensated value of cosine wave



(b) Compensated count value after error compensation

그림 9. 파형 보상 후의 출력 카운터 값 사진.

Fig. 9. Picture of counter value after wave compensation.

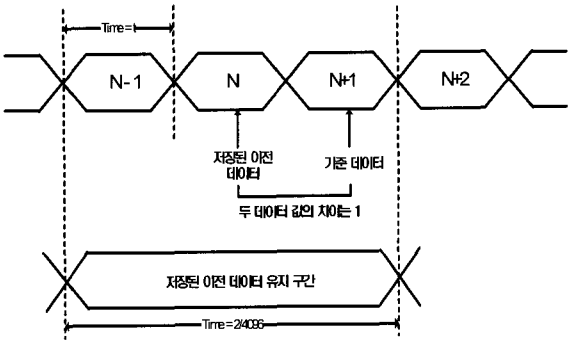
step만큼의 오차를 보임을 확인할 수 있다.

이것은 센서의 정밀도에 영향을 미친다. 이러한 문제점을 해결하기 위하여 벗어난 step값 12를 고려하여 오차를 보상해야한다(1024step-12step = 1012step). 그림 9(a)는 정확히 90°의 위상차를 위해 오차 값을 고려하여 교정된 여현파의 모습이다. 정현파는 기준이 되므로 그림 8(a)와 같다. 그림 9(b)에서 오차 보상 후 두 파형의 위상차가 정확히 90°임을 확인할 수 있다.

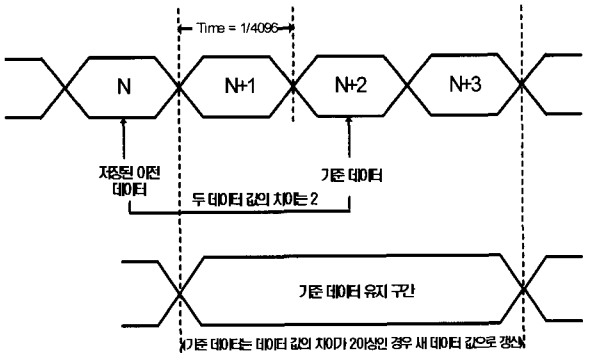
2. 응답속도를 고려한 데이터의 안정화

다음으로 FPGA 처리속도로 인해 생기는 오차 보상 방법에 대해 소개한다. 실험에서 처리 속도에 따른 카운터값의 출력은 불안정하다는 것을 확인할 수 있었다. 이것은 전반적인 장비의 구성에 있어서 MCU와 포토커플러(photo coupler)의 응답 속도(response time) 차이에 의한 것이다. MCU는 10kHz로 일정하게 데이터를 출력해주지만 이 데이터를 받는 포토커플러와 응답 속도가 맞지 않아 출력 데이터를 받는 디바이스의 속도가 데이터 출력 속도를 따라가지 못해 불안정한 데이터가 출력되는 것이다. 불안정한 데이터 출력을 방지하기 위한 기본적인 방법으로 일정구간 데이터 값의 평균을 이용하는 방법이 있다. 이 방법을 이용하면 데이터의 값의 급 변화가 일어나는 것을 방지할 수 있다. 하지만 이 방법을 사용하면 연속적으로 출력되는 두 데이터의 경계가 MCU의 빠른 데이터 출력 속도로 인해 모호해질 수 있고 MCU 다음에 연결된 디바이스가 제대로 된 데이터를 받을 수 없어서 속도를 맞추기 위해 딜레이를 사용해야 한다. 딜레이를 사용하면 응답시간이 증가하기 때문에 고주파(10kHz)를 이용한 데이터 처리가 무의미해진다.

본 논문에서 제안하는 새로운 방법은 현재 데이터의 2 차이만큼의 비교를 통해 안정적인 데이터를 출력하는 방법이다. 현재 데이터와 과거에 래치된 데이터를 비교해서 두 데이터 값의 차가 2보다 크거나 같은 경우 현재 데이터를 저장하고 출력한다. 저장된 이 값은 미래에 들어올 값과 비교의 대상이 된다. 만약 두 데이터 값의 차가 2보다 작다면 과거에 래치된 데이터를 출력한다.



(a) The case when the difference of two contiguous data is lower than two



(b) The case when the difference of two contiguous data is greater than two

그림 10. 두 인접한 데이터 값의 차이를 이용한 보상 방법.

Fig. 10. Compensation method according to the difference of two contiguous data.

그림 10은 위에서 언급한 새롭게 제시된 방법을 예를 들어 도식화한 것이다. 그림 10(a)는 두 데이터의 값의 차이가 2보다 작은 경우이고 (b)는 2와 같거나 큰 경우이다. 000이라는 데이터 값을 저장된 이전 데이터라 가정하여 논하겠다. 그림 10(a)의 경우 000 다음에 들어오는 001은 1만큼의 데이터 값의 차를 가지므로 저장된 이전 데이터인 000이 유지된다. 그림 10(b)의 경우 010은 000과 2만큼의 데이터 값의 차를 가지므로 이 데이터를 출력하고 새롭게 저장하여 다음 데이터와 비교할 때 과거의 값이 되고 새로운 비교 데이터 값이 기준 데이터가 되도록 기준을 갱신한다. 그림에서 360의 한 회전은 4096 카운터로 샘플링 되므로 한 데이터의 발생부터 소멸까지를 t(1/4096)로 표현하였다.

그러나 이 방법만으로는 안정된 데이터 값을 출력하기 위해 부족할 수 있다. 센서는 양쪽으로 회전이 가능하기 때문에 정방향과 역방향으로 센서가 회전 방향이 변할 때 대칭되는 데이터 값에서의 출력이 정확히 일치 하는지도 고려해야 한다. 회전 방향이 변한다는 것은 데이터가 증가하다가 방향을 바꾸기 위해 순간 정지가 발생하는 것을 의미하고 그 순간 포착된 데이터를 기준으로 하여 데이터의 안정화를 이루어야 한다. 데이터가 000부터 110까지 존재한다고 가정하고 정방향, 역방향으로 모두 변하는 경우를 살펴보면 표 5와 같다.

표 5. 센서 위치와 출력 데이터의 비교.

Table 5. The comparison between sensor location and output data.

센서의 위치(정방향)						
000	001	010	011	100	101	110
출력 데이터						
000	000	010	010	100	100	110
센서의 위치(역방향)						
101	100	011	010	001	000	
출력 데이터						
110	100	100	010	010	000	

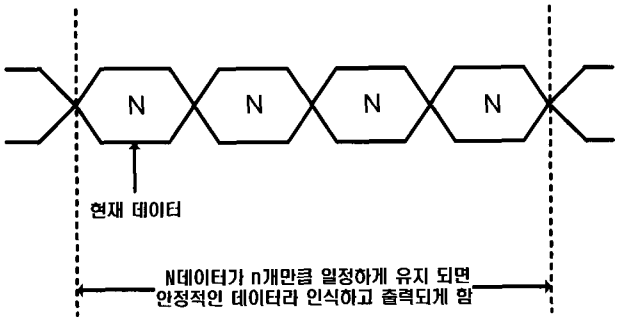


그림 11. 두 데이터 값의 차이를 이용한 방법.

Fig. 11. The method using the difference of two data value.

표 5는 센서 회전에 따른 출력 데이터의 관계를 나타내고 있다. 센서는 정방향 회전을 하다가 110데이터를 기점으로 다시 역방향 회전을 하고 있다. 이 때 정방향, 역방향의 입력 데이터가 같은 경우 출력 데이터 값도 같게 나와야 하지만 센서의 위치 001, 011, 101의 경우 출력 데이터 값이 정방향, 역방향 회전에서 센서의 위치에 따라 다른 2개의 값을 갖는다. 그렇기 때문에 안정된 출력 데이터 값이 무엇인지 구분하기가 어렵다.

이러한 오차를 보정하기 위한 방법으로 양방향의 센서의 위치가 같은 경우 출력되는 다른 2개의 데이터 중 1개의 데이터가 n번 연속적으로 유지되는 경우 안정적인 데이터로 판단하고 출력한다. 본 연구에서는 실험적으로 n=100으로 설정하였다. 시간으로 환산하면 10kHz당 100개의 데이터가 유지되므로 0.01초(100/10kHz)동안 한 데이터가 유지됨을 알 수 있다. 즉 0.01초 동안 데이터가 안정된 구간에 위치한다면 안정된 출력을 얻을 수 있다. 본 연구에서 구현된 다회전 센서 드라이버의 경우 이 방법을 적용하였을 때 한 바퀴 분해능이 4096 카운터를 가지므로 이 분해능을 시간에 따라 환산하였을 때 40.96초(4096/100)를 얻을 수 있고 이 시간 동안 센서가 한 바퀴 회전을 한다면($\approx$ 1.46rpm) 정방향, 역방향의 위치에 상관없이 시간적으로 안정적인 데이터 값을 검출할 수 있게 된다.

실제 실험 조건에서 MCU 보드는 10kHz로 일정하게 데이터를 출력해주지만 그 데이터를 받는 포토커플러 단의 응답속도가 충분이 빨라야 안정된 데이터를 출력할 수 있다. MCU는 고정적인 속도로 내보내지만 포토커플러는 MCU에서 데이터를 처리하는 주파수에 맞는 스펙의 제품을 선택해야 한다. 그렇지 않으면 falling edge, rising edge 발생시 서로 타이밍이 달라서 데이터가 경계에 걸려 불안정하게 출력될 수 있다. 따라서 본 연구를 통해 구현된 기기는 데이터의 입·출력이 연결되는 포토커플러와 MCU를 비롯한 D/A 출력단의 저주파 필터의 응답속도를 고려하여 설계하였다.

V. 결론

본 연구에서는 RVDT 센서를 안정적으로 제어하고 신뢰성 있는 절대 위치 데이터를 얻기 위한 RVDT 센서 드라이버의 FPGA 단일 칩에 의한 오차 보정 방법에 중점을 두었다. 센서 드라이버의 FPGA 단일 칩 구현은 앞에서 언급하였던 것처럼 가격, 면적 측면의 장점뿐만 아니라 알고리즘의 변경에 따른 수정이 용이하다. 실제 센서 구동 입력 파형의 구현은 단회전, 다회전 2가지 센서에 대하여 센서 드라이버를 구현하였고 각각 0.2%와 0.1%의 좋은 THD 값을 얻었다.

또한 절대위치 검출기 대한 구현 및 실험 결과, 그리고 오차 보상 방법에 대해서 소개하였다. 제안된 방법은 아날로그 회로에서 발생할 수 있는 절대 위치 오차를 디지털적으로 VHDL 코드만을 변환하는 것으로 오차 보상이 가능하였다. 아날로그적으로도 어느 정도 안정된 데이터를 획득할 수 있었지만 불가피하게 발생하는 오차에 대해 알아보고 연구를 통해 2가지 보상 방법을 제시하였다. 첫 번째 방법은 실제 구현 시 기계적으로 발생되는 오차를 고려하여 정현파, 여현파의 위상차를 정확하게 90°로 맞추는 방법이고 두 번째 방법은 처리 속도를 고려하여 안정된 출력 데이터 값을 획득하는 것이다. 제안된 두 가지 방법에 의한 오차 보상 방법으로 실험한 결과 안정된 절대 위치 데이터를 출력할 수 있었다.

연구에서 제안된 방법은 신뢰성 있는 산업용 절대 위치 검출기에 적용이 될 수 있으며 향후 이를 바탕으로 RVDT 센서 정밀도, 신뢰도가 더 높은 3상 RVDT에 대한 연구의 기반이 될 것으로 판단한다.

참고문헌

[1] D.-Y. Shin, Y.-G. Yang, and C.-S. Lee, "RVDT phase error compensation for absolute displacement measurement," ICASE, 2006.

[2] R. Pallas-Areny and J. G. Webster, *Sensors and Signal Conditioning.*, Wiley : NewYork, 1991.

[3] F. Yassa and S. Garvericks, "Multichannel digital demodulator for LVDT/RVDT position sensors," IEEE J.Solid-State Circuits, vol. 25, pp. 441-445, Apr. 1990.

[4] Charles H. Roth, J, "Digital systems design using VHDL," Thomson Learning. 1998.

[5] S.-H. Lee, Y.-S. Park, G.-J. Park, and J.-H. Lee, *Digital Logical Circuit Design Using ALTERA MAX+PLUS2*, Bogdoo Publishing co. 2005.

[6] Encoder vs. Resolver-Based Servo Systems application note, ORMEC Corp.

[7] Control Sciences Inc. application data, Resolvers as

Velocity and Position Encoding Devices.

[8] A Discussion of Encoder Technology's Converter and Impedance Detector Technology, Encoder Tech. Corp, 2000.

[9] Dong-Yoon Shin, "Implementation of RVDT sensor and its controller for an absolute position detection," A graduation thesis of The University of Suwon, 2006.

전 지 혜

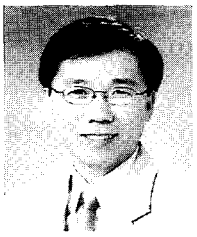
2007년 수원대학교 전자공학과 학사. 2007년~현재 동 대학원 석사과정. 관심분야는 영상처리, 제어계측, OCT.

신 동 윤

2005년 수원대학교 전자공학과 학사. 2007년 동 대학원 석사. 현재 포스테크 주임 연구원. 관심분야는 영상처리, OCT, 제어계측, 임베디드 시스템.

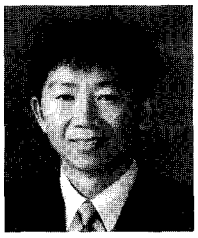
양 윤 기

1989년 서울대학교 제어계측공학과 학사. 1991년 동 대학원 석사. 1996년 박사. 현재 수원대학교 정보통신공학과 부교수. 관심분야는 통신 시스템, 컴퓨터 응용기술.



황 진 권

1985년 서울대학교 제어계측공학과 학사. 1987년 동 대학원 석사. 1997년 박사. 현재 우석대학교 소방안전학과 교수. 관심분야는 무선통신, 신호처리, 제어시스템.



이 창 수

1985년 서울대학교 제어계측공학과 학사. 1987년 동 대학원 석사. 1997년 박사. 현재 수원대학교 전자공학과 부교수. 주요 연구분야는 영상처리, OCT, 통신 시스템, 웨이블릿 응용, 검사 자동화.