

**다축 힘제어 시스템을 위한 웹기반 감시시스템 구현**

남승욱\*, 이현철\*, 남현도\*, 강철구\*\*  
 \* 단국대학교 전기공학과, \*\* 건국대학교 기계공학과

**Implementation of Web Based Monitoring Systems for Multi-Axis Force Control Systems**

Seung-Uk Nam\*, Hyun-Chul Lee\*, Hyun-Do Nam\*, Chul-Goo Kang\*\*  
 \* Department Of Electrical Engineering, Dankook University, \*\* ME, Konkuk University

**Abstract** - In this paper, web based monitoring systems are implemented for multi-axis force control systems of an intelligence robot. A web based monitoring system is implemented by porting Linux at embedded systems which include a Xscale processor. A device driver is developed to receive data from multi-axis force sensors in Linux operation systems. To control this device driver, a socket program for the Labview is also developed.

**1. 서 론**

컴퓨터와 인터넷 기술의 발달은 산업 현장에서의 인터넷 기반 제어와 감시를 가능하게 했다. 다축 힘 제어 센서를 사용한 자동화 제조 공정인 분포 제어 시스템과 로봇 힘 제어 시스템은 웹기반 기술로 향상 되었다[1-4].

본 논문은 6축 힘/토크 센서를 사용한 지능형 로봇의 힘 제어 시스템을 위해 웹 기반 감시 및 제어 시스템을 구현하였다. 스트레인 게이지를 응용한 다축 힘 센서에서 감지된 힘 정보는 오차를 수반하고 있으므로 신호라인에서 발생하는 노이즈의 처리를 위해 A/D 컨버터를 사용하여 이 데이터를 Xscale 프로세서를 이용한 임베디드 시스템에서 처리하는 시스템을 구현하였다.

웹 기반 감시 시스템 구현을 위하여 Xscale 프로세서를 기반으로 한 임베디드 시스템에 리눅스를 포팅하고 [5] TCP/IP 소켓프로그램으로 소켓을 열어 Xcale 프로세서에 연결된 다축 힘 센서의 출력을 LAN 망을 통한 LabVIEW에 의하여 감시를 하는 웹 기반 감시 시스템을 구현하였다.

리눅스 운영 시스템에서 다축 힘 센서의 데이터를 받기 위하여 디바이스 드라이버를 개발하고 이 디바이스 드라이버를 제어하기 위한 응용프로그램으로 TCP/IP 소켓 프로그램을 사용하여 다축 힘 센서에서 받은 데이터를 LabVIEW를 통해 수신을 하게 된다. A/D 컨버터를 사용한 디지털화로 신호의 안정화, 노이즈 영향 감소를 윈도우상의 LabVIEW를 통해 가시하면서 센서의 성능을 평가하고 앞으로 센서의 개선 방안 방법에 대한 방법의 타당성을 검증하는 시스템을 구현하였다.

**2. 힘 센싱 시스템**

힘/토크 측정 방법에는 힘 측정을 위해 만든 탄성구조물에 여러 개의 스트레인 게이지를 부착하여 구조체에 힘/토크가 가해졌을 때 그에 비례하는 스트레인 게이지의 변형량을 측정하여 Wheatstone 브리지기를 통해 미소 전압으로 바꾸고 이 미소 전압을 증폭기로 증폭시켜 나온 전압 값을 측정한다. 이 데이터 값으로 탄성변형과 힘의 관계식으로부터, 센서에 작용하고 있는 세 방향의 힘과 세 방향의 토크를 구하는 것이다[1,2].

**2.1. 힘 센서의 신호처리**

힘/토크 센서에 힘을 가하면 스트레인 게이지에서 미세

한 저항의 변화가 생기게 되는데 이를 Wheatstone 브리지를 사용하여 전압으로 바꾸게 된다. 이때 나오는 신호는 미소 전압으로 이를 OP-AMP를 사용해 전압 증폭을 하게 된다. 증폭된 전압으로 탄성체의 변형량을 측정하게 된다. 신호에 포함된 노이즈를 제거하기 위하여 저주파 통과필터를 통과시킨다. 본 논문에서는 신호 통과 대역에서 평탄 특성을 유지하면서 회로를 단순화하기 위하여 다음 그림 1과 같은 Sallen-Key Topology의 2차 Butterworth 저주파통과필터를 사용하였다. 그리고 일반적으로 사용되는 기계시스템의 대역폭과 제어용 샘플링 시간을 고려하여 차단주파수를 102Hz로 선정하였다.

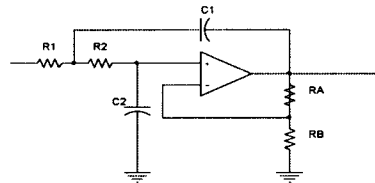


그림 1. 2차 Butterworth 저주파통과필터

이 저주파통과필터의 전달함수 H(s)는 다음과 같다.

$$H(s) = \frac{1/(R_1 R_2 C_1 C_2)}{s^2 + \frac{R_1 + R_2}{R_1 R_2 C_1} s + \frac{1}{R_1 R_2 C_1 C_2}} \quad (1)$$

본 논문에서 사용한 2차 Butterworth 저주파통과필터의 각 소자값으로  $R_1=R_2=110k\Omega$ ,  $C_1=0.02\mu F$ ,  $C_2=0.01\mu F$ 이고,  $R_A=R_B=0$ 을 사용하였다[3,4].

**2.2 다축 힘 센서의 힘 감지 원리**

스트레인 게이지를 이용한 6축 힘/토크 센서의 힘/토크 감지 과정을 도식적으로 나내면 그림 2와 같다.

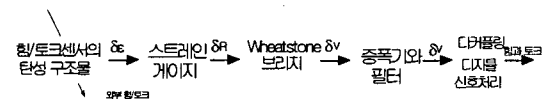


그림 2. 힘/토크 정보 감지과정

Hooke의 법칙을 만족하는 탄성영역에서 변형이 일어난다고 가정하면 식 (2)와 같다.

$$Cf = \epsilon \quad (2)$$

여기서 f는 센서에 작용하는 3개의 힘 성분과 3개의 모멘트성분으로 구성된  $n \times 1$  힘 벡터이고,  $\epsilon$ 은 탄성구조물의 6 지점에서 측정되는  $m \times 1$  표면변형률벡터이고, C는 이 두 벡터 사이의 관계를 표시하는  $m \times n$  행렬로 컴플라이언스(compliance) 행렬 또는 보정행렬(calibration)

로 불려진다. 컴플라이언스 행렬은 상수행렬로서, FEM 해석으로 얻을 수 있으나, 실험에 의하여 구하는 것이 힘 감지 오차를 줄일 수 있다. 일차독립인 6 개의 힘  $f$  에 대해 6 번의 실험을 수행하고, 이 때 얻어지는  $\epsilon$  으로부터  $C$  를 얻을 수 있다. 벡터  $\bar{f}$  에는 힘과 모멘트가 함께 포함되어 있어 사용하는 단위에 따라  $\bar{C}$  행렬의 성질이 변할 수 있다. 따라서 이런  $\bar{C}$  행렬의 변하는 성질을 정규화를 통해 제거한다. 벡터  $\bar{f}$  과  $\bar{\epsilon}$  을 정방향렬  $N_f, N_\epsilon$  로 정규화 시킨다.

$$\bar{f} = N_f f, \quad N_f = \text{diag}\{f_{1M}, f_{2M}, \dots, f_{nM}\} \quad (3)$$

$$\bar{\epsilon} = N_\epsilon \epsilon, \quad N_\epsilon = \text{diag}\{\epsilon_{1M}, \epsilon_{2M}, \dots, \epsilon_{nM}\} \quad (4)$$

$N_\epsilon$  과  $N_f$  로  $\bar{C}$  를 정규화 시킨 컴플라이언스 행렬을 구하면 다음과 같다.

$$C = N_\epsilon^{-1} N_f \bar{C} \quad (5)$$

위 식으로 정규화 된 변형률과 외력은  $C f = \epsilon$  의 관계식이 성립한다.

$$f = (C^T C)^{-1} C^T \epsilon \quad (6)$$

위 식으로 컴플라이언스 행렬을 안다면 힘의 변형률로 센서의 힘의 크기를 구할 수 있다.

### 3. 내장형 시스템과 웹기반 감시시스템

본 논문에서는 타겟 보드로는 한백전자사에서 제작한 Intel Xscale PAX255 Processor(400MHz), 128M Bytes SDRAM, 32M Bytes 플래쉬 메모리를 내장한 HBE-EMPOS II 보드를 사용하였다. PC를 개발 호스트로 사용하였고 사용 OS 및 프로그램은 Redhat Linux 8.0 (커널 ARM 패치), GNU Tool, Minicom 등이다.

#### 3.1 웹기반 감시시스템

웹 기반 감시 시스템 구현을 위하여 TCP/IP 소켓프로그램으로 소켓을 열어 Xscale 프로세서에 연결된 다축 힘 센서의 출력을 LAN 망을 통해 LabVIEW에 의하여 감시하는 웹 기반 감시 시스템을 구현하였다[6-8].

#### 3.2 내장형 시스템

내장형 시스템이라는 것은 통상적으로 일반적인 시스템과는 달리 특정한 작업만을 수행하도록 설계되었다. 이전의 내장형 시스템들은 주로 간단하고 단순한 순차적인 작업에 관련되었기 때문에 굳이 운영체제를 사용하는 것은 낭비가 되었고 그럴 필요조차 없었다. 하지만 최근의 내장형 시스템 분야에서는 그 시스템 자체가 상당히 커지게 되고, 네트워크나 멀티미디어가 시스템에 기본적으로 자리 잡으면서 내장형 시스템이 해야 할 일들도 많아지고 복잡해졌기 때문에 순차적인 프로그램을 작성하기가 어렵게 되었다. 따라서 내장형 시스템에서는 운영체제의 개념이 필요하게 되었고 내장형 시스템의 특성상 실시간이라는 요소를 만족해야 하였으므로 실시간 운영체제가 내장형 시스템에 도입되었다.

#### 3.2 내장형 웹 서버 구성

내장형 웹 서버는 정보를 교환할 수 있는 HTTP와 TCP/IP 지원 가능한 웹 서버 그리고 외부 기기를 제어하기 위한 디바이스 드라이버와 디바이스 드라이버를 제어하여 웹 서버에게 정보를 전달하는 TCP/IP 프로그램 등으로 구성된다. 본 논문에서는 Goahead 웹 서버를 사용하였다. 그림 3은 내장형 웹 서버의 구성을 보여준다.

### 3.3 웹기반 감시시스템을 위한 TCP/IP

인터넷 프로토콜(IP)은 데이터를 통신상에서 전달하는 저수준의 프로토콜이다. IP는 데이터를 잘게 나누는데 이것을 데이터그램(Datagram)이라한다. 데이터그램은 데이터와 헤더에 목적지의 주소와 소스의 주소를 담고 있다. 또한 IP는 네트워크나 인터넷상에서 데이터를 보내기 위해 적당한 경로를 설정한다. 원래의 호스트는 데이터가 보내질 완전한 경로를 모른다. 하지만 헤더를 이용하여 네트워크상의 호스트는 데이터 경로를 설정하여 직접 보내거나 다른 호스트를 경유하여 데이터그램을 보낸다. 그리고 목적지에 도착한 데이터그램은 필요에 따라 작게 분리되어 사용되거나, 또는 데이터를 자동으로 원래의 형태로 조립하기도 한다. IP는 데이터를 보내기 위한 프로토콜이며, 데이터 전달의 확실성은 보장하지 못한다. 또한 IP는 데이터의 순서를 모르기 때문에 순서가 바뀌어 보내지는 수도 있고, 중복되게 보내는 수도 있다. TCP는 네트워크 상에서 데이터의 전달에 대한 신뢰성을 사용자에게 제공한다. 데이터를 에러 없이 순서대로, 그리고 데이터의 손실 없이 보낼 수 있는 것이다. TCP를 사용하면 IP와 같은 데이터를 추가하고 데이터그램 형태로 만들어 전달한다. 이러한 순서는 수신측에서 받전되어 데이터그램을 분리하고 추가된 정보를 추출하고 에러를 체크한다[5].

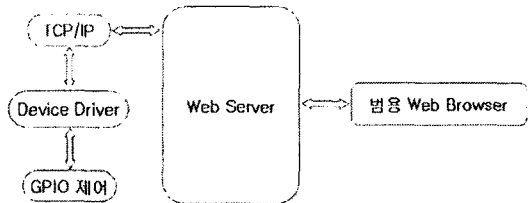


그림 3. 내장형 웹 서버 구성

### 4. 원격제어를 위한 웹 기반 감시 시스템

인터넷을 통한 원격 제어 시스템이란 인터넷을 통하여 인터넷에 연결되어 있는 다른 여러 개의 시스템에 쉽게 접속하여 제어함과 동시에 감시하는 것이라 할 수 있다. 이것은 대부분 PC에서 구현되었다. 웹 서버란 웹 페이지가 들어있는 파일을 사용자에게 제공하는 프로그램이나 이 프로그램이 설치되어 있는 기기를 말하기도 한다.

원격 제어 시스템을 위한 내장형 웹 서버는 주로 시스템에 연결된 특정 하드웨어를 제어하거나 그 정보를 전달하여 주는 역할을 하게 된다. 웹 서버는 사용자의 웹 브라우저 요청에 따라 HTTP를 통해 정보를 전달한다. 이를 위해 TCP/IP 소켓 프로그램을 사용하여 타겟 보드의 IP로 접근하여 데이터를 받아 올 수 있다.

#### 4.1 디바이스 드라이버

범용 웹 브라우저를 통하여 프로세서의 포트를 제어하고 감시하기 위해서는 우선 포트를 제어할 수 있어야 하는데 리눅스 운영 시스템에서 하드웨어 주변장치를 제어하기 위해서는 디바이스 드라이버가 필요하다. 리눅스 운영 시스템은 하나의 장치들 파일로 인식하여 접근하며, 디바이스 드라이버를 통하여 응용 프로그램에서 하드웨어를 제어하게 된다.[8].

본 논문에서는 캐릭터 디바이스 드라이버 방식으로 프로그램 하였으며 커널 모듈(kernel module: 커널의 일부 분을 통적으로 로드 또는 언로드할 수 있는 커널의 구성 요소) 프로그램에 의하여 모듈로 사용할 수 있게 하였다. 커널 모드에서 실행될 때는 메모리에 상주하고 이것을 통해 하드웨어 장치를 파일처럼 다룰 수 있게 된다. 커널 모듈을 만들 때는 반드시 초기화 함수와 삭제 함수, 커널 버전이 정의되어야 한다[7,8,9].

본 논문에서는 힘정보를 LabVIEW로 감시하기 위한 디바이스 드라이버를 작성하였다. 함수의 내용을 살펴보면 init\_module()에서는 장치 등록번호(major number)를 동적으로 받아서 ADC의 이름으로 디바이스와 IRQ를 등록하고 cleanup\_module()에서 등록된 디바이스와 IRQ를 해제한다. adc\_interrupt()는 응용프로그램에서 디바이스를 호출할 때 쓰이는 함수이다. adc\_open()은 응용프로그램에서 디바이스를 open 하였을 때 호출되는 함수로써 모듈의 사용자수를 증가시키고 adc\_release()는 adc\_open() 함수와 반대의 기능을 하며 모듈의 사용자수를 감소시키고 사용자 수가 0이 되면 디바이스를 사용하지 못하게 한다. adc\_read()는 데이터를 저장하고 응용프로그램으로 데이터를 넘겨주는 역할을 하고 adc\_write()는 adc\_read()와 반대의 역할을 한다. 제작된 디바이스 드라이버를 램디스크(RAM disk)에 모듈화하여 실행함으로써 제어하고자 하는 프로그램에서 모듈을 열어 디바이스 드라이버를 사용할 수 있게 된다.

### 4.2 TCP/IP 소켓 프로그래밍

소켓(Socket)은 이를 통해 애플리케이션이 데이터를 보내고 받을 수 있는 추상적 개념으로서, 애플리케이션이 저장 장소에 데이터를 읽고 쓰는 과정을 open된 파일을 통해 처리할 수 있는 것과 유사하다. 소켓은 애플리케이션이 네트워크에 "플러그 인(plug in)"하여 동일한 네트워크에 마찬가지로 플러그 인된 다른 애플리케이션과 통신할 수 있도록 한다. 한 기계에서 애플리케이션에 의해 소켓에 써진 데이터는 다른 기계에서 동작하는 애플리케이션에 의해 읽혀질 수 있으며, 그 역도 성립한다 [10,11].

본 논문에서는 C 언어를 사용하여 서버를 구축하였다. 먼저, socket() 함수를 호출하여 통신에 사용될 소켓을 개설하고 이때 리턴된 소켓번호와 자신의 소켓주소를 bind() 함수를 호출하여 서로 연결시켜 둔다. 서버에서 bind() 함수가 필요한 이유는, 소켓번호는 응용 프로그램 내에서만 알고 있는 통신 창구이고, 소켓주소는 네트워크 시스템(즉, TCP/IP)이 알고 있는 주소이므로 이들의 관계를 묶어 두어야 응용 프로그램과 네트워크 시스템간의 데이터 전달이 가능하기 때문이다.

다음에 서버는 listen() 함수를 호출하여 수동 대기모드로 들어가며 클라이언트로부터 오는 연결요청을 처리할 수 있게 된다. 서버는 이제 연결설정 결과를 얻기 위해 accept() 함수를 호출하고 기다린다. 클라이언트와 연결이 성공적으로 이루어지면 accept() 함수가 새로운 소켓을 하나 리턴하는데 서버와 클라이언트는 이 소켓을 통해서 데이터를 송수신할 수 있게 된다. 그림 5.4는 TCP/IP 소켓 프로그래밍 절차를 보여준다.

### 4.3 LabVIEW를 사용한 Client

LabVIEW는 Datasocket, TCP/IP, UDP, IrDA의 통신을 제공함으로써 네트워크 기반의 제어를 가능하게 해주었다. 본 논문에서는 LabVIEW의 TCP/IP 기능을 이용한 Client를 구축하였다. 이 Client를 이용하여 데이터의 송신과 수신된 데이터의 분석과 컴플라이언스 행렬의 사용으로 힘정보 추출을 할 수 있다.

LabVIEW의 장점으로 사용자 편의성을 둘 수가 있는데 편리하게 프로그램을 작성하고 디스플레이 기능을 통하여 현재의 힘/토크 상태를 그래프도 디스플레이 할 수가 있다. 또 받은 데이터를 이용하여 다른 기기의 제어에 이용할 수 있고 인터넷을 이용한 외부에서의 컨트롤도 가능케 한다 [12].

소켓 프로그램을 이용한 Server에서 TCP 포트를 열면 Client에서 TCP/IP를 받아서 데이터를 Client 컴퓨터에 디스플레이 하게 된다. LabVIEW Block 채널 구분 방법으로는 Formula Node를 사용하였다. Formula Node는 C언어 작성하듯이 작성할 수 있게 구성되어 있다. 구분

한 각 채널의 전압값을 바탕으로 컴플라이언스 행렬을 생성한다. Formula Node에서 컴플라이언스 행렬을 적용하는 프로그램을 작성하여 각 채널의 데이터를 통과시킨 뒤 그 값이 바로 힘과 토크의 값이 된다.

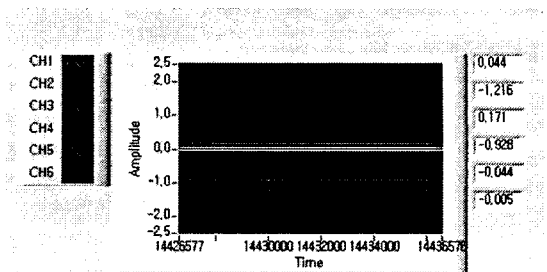


그림 5. 힘 정보 그래프

## 3. 결 론

본 연구에서는 힘/토크센서의 디지털화를 하였고 내장형 시스템의 운영시스템 중 여러 면에서 장점을 가지고 있어 최근 대두되고 있는 리눅스 운영시스템을 기반으로 한 Xscale 프로세서에 소켓프로그램을 이용해 서버를 구축하고 Xscale 프로세서에 연결된 힘센서의 데이터를 받기 위하여 디바이스 드라이버를 구현하였다. 감시만 하는 단방향 통신이 아니라 LabVIEW를 이용해 서버와 클라이언트간의 양방향 통신을 할 수 있는 웹기반 감시시스템을 구현하였다. 이로써 컴플라이언스행렬을 구해 탄성구조물에 필연적으로 존재하는 축간의 커플링효과를 제거할 수 있다.

본 연구는 한국과학재단 목적기초연구(R01-2002-000-00124-0(2002))지원으로 수행되었음.

### [참 고 문 헌]

- [1] 강철구, "힘/토크 센서의 힘감지 오차에 대한 해석," 대한기계학회 논문집, 22권 제7호, pp.1250-1257, 1998.
- [2] 김주용, 강철구, "원구멍이 있는 십자형 탄성체를 가진 6축 힘/토크 센서의 변형률 해석," 한국정밀공학학회지, 16권 2호, pp.5-14, 1999.
- [3] 이재호, "스트레인 게이지를 이용한 6축 힘/토크 센서의 신호처리에 관한 연구" 건국대학교 석사학위논문, 2000.
- [4] 김용찬, "마이크로 컨트롤러와 Labview를 이용한 다축 힘센서의 신호처리," 건국대학교 석사학위논문, 2003.
- [5] 임홍식, "임베디드 시스템을 이용한 웹기반 감시시스템 구현," 단국대학교 석사학위논문, 2004.
- [6] Michael Barr, Programming Embedded System :In C And C++, O'Reilly, 1999.
- [7] 박영환, 임베디드 시스템 임베디드 리눅스, 사이텍 미디어, 2002.
- [8] 송태훈, Intel PXA255와 임베디드 리눅스 응용, 홍릉과학출판사, 2004.
- [9] Alessandro Runibi, Linux Device Driver, O'Reilly, 2000.
- [10] 박준철, TCP/IP 소켓 프로그래밍, 사이텍 미디어, 2001
- [11] 김화중, 컴퓨터 네트워크 프로그래밍, 홍릉과학출판사, 2004.
- [12] 박두영, LabVIEW 컴퓨터 기반의 제어와 계측 Solution, Ohm사, 2004.