

## 휴머노이드 로봇 팔의 위치 추종을 위한 FPGA 기반의 신경회로망 제어기 구현

김 정 섭, 정 슬

충남대학교 메카트로닉스공학과 지능로봇시스템

### FPGA Implementation of Neural Network Controller for Position control of Humanoid Robot Arm

Jeong Seob, Kim and Seul Jung

Department of Mechatronics Engineering, Chungnam National University

**Abstract** - 본 논문은 FPGA 기반에서 실수형 프로세서의 설계 및 구현에 대한 내용과 이를 이용하여 휴머노이드 로봇 팔의 위치제어를 위한 PD 제어기 기반의 신경회로망 제어기의 구현에 대한 내용이다. 설계된 프로세서는 명령어 기반의 처리를 통해 산술 연산 뿐만 아니라 로봇의 제어에 사용되는 외부 모듈의 사용이 가능하도록 설계하였으며, 신경회로망 구현에 사용되는 지수함수를 효율적으로 근사화하기 위한 Taylor series를 이용한 알고리즘을 하드웨어 레벨에서 구현하였다. 휴머노이드 로봇 팔의 위치 추종을 위해 고선적인 PD 제어기를 설계하고 PD 기반의 신경회로망 제어기를 설계하였다. 로봇 팔의 6축 제어를 위한 신경회로망 제어기에 요구되는 많은 연산을 감당하도록 하기 위해 설계된 프로세서를 통해 정의된 프로그래밍 언어로 제어 프로그램을 작성하였다. PD 제어기와 PD 기반의 신경회로망 제어기를 하드웨어에 설계하여 로봇팔의 위치 추종을 실험하였으며 성능을 비교 검증하였다. 프로세서는 Altera의 Stratrix II EP2S180 DSP development board에 구현되었으며 실험적으로 25MIPS의 성능을 가지는 것으로 나타났다.

#### 1. 서 론

데이터 처리량이 많은 이미지 프로세싱을 FPGA 상에서 구현하는 방법에 관한 많은 연구가 이루어지고 있으며, 제어 분야에서도 FPGA를 이용하여 다양한 제어기를 구현하고자 하는 연구가 있다[1]. 하지만 대부분의 경우 구현의 복잡성으로 인해 고정소수점 기반으로 연구가 이루어지며 비선형 함수의 연산이 필요할 경우 Look-Up Table(LUT)을 많이 사용하고 있다 [2]. 그로 인해 부동소수점 기반의 연산에 비해 정확도가 낮아지며 표현 가능한 수의 범위에도 한계가 있을 수밖에 없다.

FPGA를 이용한 제어기 설계의 경우 기본적인 제어 알고리즘의 계산 이외에도 기구학이나 동역학 연산이 필요하기도 하다. 따라서 로봇 제어에 필요한 제어기의 설계에 있어서 덧셈기나 곱셈기, 비교기나 쉬프트 등의 고정된 모듈로 모든 수식을 표현하는 것은 전체적인 연산 속도는 빠를 수 있을지 모르나 제어 알고리즘이 복잡하고 동역학 연산량이 많으며 제어해야 할 축이 많다면 너무 많은 FPGA 용량이 필요하게 된다. 따라서 제어 주기에 영향을 미치지 않는 범위에서 제한된 연산 모듈을 통해 순차적으로 처리하는 것이 보다 더 효율적이다. 최근에는 SIMD(Single Instruction Multiple Data) 방식을 이용하여 순차처리 기법과 병렬처리 기법을 동시에 사용하는 연구도 행해지고 있다[3].

본 논문에서는 부동소수점 기반의 덧셈기, 뺄셈기, 곱셈기 그리고 나눗셈기를 포함한 수치 연산 모듈을 설계하여 로봇 제어에 필요한 비선형 함수를 효율적으로 전개 가능하도록 하였다. 각각의 연산을 위한 제어 코드인 명령어(instruction)를 자체적으로 설계하였으며, 이의 해석 및 시스템의 전체적인 제어를 순차적으로 처리하기 위한 파이프라인 구조를 설계하였다. 또한 로봇 제어를 위한 하드웨어 모듈인 엔코더, PWM 생성기, 그리고 통신 모듈에 대한 명령어 기반의 처리가 가능하도록 하여 사용자가 작성한 프로그램을 통해서 원하는 제어 알고리즘을 적용이 가능하고 설계 하드웨어로 적용이 가능하도록 하는 시스템을 구현하였다.

제안된 부동소수점 프로세서를 FPGA에 구현하였으며 신경회로망 기반의 PD 제어기에 대한 알고리즘을 프로그래밍하였다. 작성된 프로그램을 통해 6축의 휴머노이드 로봇팔의 위치를 추종하는 실험을 진행하였다. 위치 제어의 성능을 비교하기 위해 고선적인 PD 제어기를 설계하여 실험하고 신경회로망 기반의 PD 제어기를 설계하여 실험한 결과를 서로 비교하여 FPGA 기반으로 설계된 신경회로망 제어기의 제어 성능을 검증하였다.

#### 2. 프로세서 설계

##### 2.1 코어 아키텍처 설계

현재 버전의 코어 구조는 이전 버전의 구조에 비해서 범용 레지스터 그룹을 하나로 통합함에 따라 데이터 라인의 복잡도가 감소하는 결과를 얻게 되었다[4]. 범용 레지스터 그룹 모듈의 구조를 Multi-Read/Single-Write 구조로 만들어 Functional Unit에서의 연산에 필요한 오퍼랜드 2개를 하나의 레지스터 그룹에서 읽을 수 있도록 개선하였다.

이전 버전과의 가장 큰 차이점 중 하나는 부동소수점 기반의 구조에서 메모리에 저장되어 있는 변수의 배열 참조 방식을 사용하는 것이 가능해졌다는 것이다. 특정 메모리 번지에는 특정 배열 변수의 주소 값이 IEEE754 구조의 부동소수점 포맷 형태로 저장되어 있으며, 이 값이 배열 구조를 가지는 그 변수를 참조할 수 있도록 unsigned integer 형태로 변환되어 데이터 메모리에 액세스가 가능하다는 것을 의미하며, 이를 위한 별도의 변환

모듈을 설계하였다.

이는 FPGA 기반의 신경회로망 구현에 대한 문제에 있어서 시간적인 요소가 허용하는 범위 내에서 입력층, 은닉층, 그리고 출력층의 뉴런을 쉽게 늘리는 것이 가능하다는 것을 보여준다.

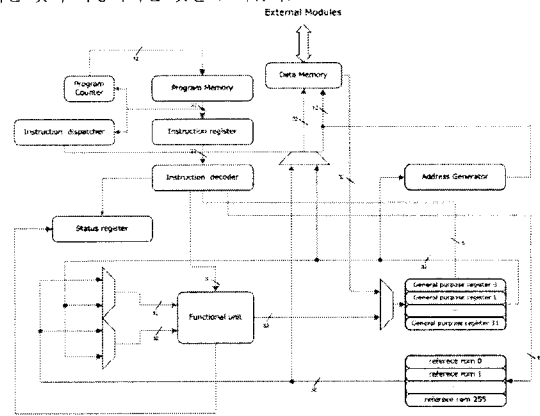


그림 1. 설계된 프로세서

그림 1은 설계된 프로세서 내부의 구조를 간단하게 보여주어 주고 있다. 전체적인 구성은 Program memory, Data memory, Instruction Register, Instruction Decoder, Instruction Dispatcher, Functional Unit, Register Group, Reference ROM, Status Register, Program Counter, Address Generator 그리고 몇 개의 Multiplexer로 이루어져 있다. 각각 모듈에 대한 설명은 아래와 같다.

- 1) Program Memory: instruction code를 담고 있는 이 모듈은 ROM 형태로 이루어져 있으며 32bit 길이의 word가 총 4,096개로 총 128bit의 address를 가진다. 따라서 Program Memory는 16Kbytes의 용량을 가진다.
- 2) Data Memory: 연산에 필요한 데이터가 저장되는 RAM 타입의 모듈이다. Program memory와 마찬가지로 1 word가 32bit로 구성되어 있으며 12bit의 address를 가지며 16Kbytes의 용량을 가진다.
- 3) Program Counter: 다음에 수행되어야 할 명령어가 들어있는 Program Memory의 주소를 지정한다. Status Register를 참조하여 Jump 명령어이나 Condition Branch 명령어를 수행하기도 한다.
- 4) Instruction Register: instruction을 instruction decoder에 보내기 전에 1cycle을 딜레이 시키는 역할을 한다.
- 5) Instruction Dispatcher: instruction을 Fetch 하는 과정에서 LD 명령과 같은 일부 명령어의 경우 메모리와 프로세서 파이프라인 동작과의 동기화를 위해 Decode 단계 이전에 명령어 해석이 이루어지게 되며 Instruction Dispatcher에서 이러한 역할을 담당한다.
- 6) Instruction Decoder: 정의되어 있는 32bit instruction code를 opcode와 operand의 주소 등으로 구분하여 해석하고 명령어 수행을 위한 코어 내부의 각 모듈을 제어하기 위한 컨트롤 시그널을 내보낸다.
- 7) Status Register: arithmetic unit의 연산 결과에 따른 결과나 logical unit의 명령어 수행 결과 또는 코어 내부의 상태를 나타내기 위한 시스템 내부 상태 레지스터이다.
- 8) Register Group: Register Group은 32비트 레지스터로 구성된 32개의 레지스터가 존재하며, Multi-Read/Single-Write 구조를 가진다.
- 9) Reference ROM: 연산에서 자주 참조되는 상수들, 예를 들어  $\pi$ 나 자연 상수 e 또는 비선형 함수의 수치해석적 연산을 위해 필요한 계수 값들이 저장되어 있다.
- 10) Functional Unit: Single precision 기반의 수치 연산 및 논리 연산 그리고 포맷 변환이 이루어지는 모듈이다. 수치 연산으로는 덧셈, 뺄셈, 곱셈 그리고 나눗셈의 사칙연산을 비롯한 절대값과 같은 부가적인 연산이 수행된다. 논리 연산은 operand1과 operand2의 크기를 비교하는 명령어를 수행한다.
- 11) Address Generator: IEEE 754 부동소수점 형식으로 저장되어 있는 값을 메모리의 어드레스를 위해 unsigned integer 형태로 변환해준다.

## 2.2 외부 모듈

로봇 제어에 있어서 가장 많이 사용되는 외부 모듈 중에는 엔코더 카운터, PWM 생성기, 그리고 통신 모듈이 존재한다. 이전 버전의 프로세서에서는 이러한 모듈을 사용하고자 할 때, VHDL로 하드코딩을 하여 구현하였기 때문에 프로그램이 변경되면 이에 맞게 해당되는 모듈의 구조도 변경할 필요가 있었다. 예를 들면, 데이터 메모리 내에 존재하는 특정 데이터를 외부로 전송하고자 할 때, 통신 모듈을 컨트롤하는 모듈을 따로 두어 이 컨트롤 모듈이 해당 데이터 메모리의 주소를 액세스하여 데이터를 가져온 후, 이를 통신 모듈을 통해 전송하였지만, 현재 버전의 프로세서에서는 이를 처리할 수 있는 전용 명령어를 설계 및 구현하여, 사용자가 작성한 프로그램 내에서 특정 데이터를 통신모듈을 통해 외부로 전송할 수 있는 구조로 변경하였다.

통신 모듈 이외에도 엔코더 카운터와 PWM 생성기 마찬가지로 특정 엔코더 카운트 값을 읽어 올 수 있도록 하는 전용 명령어와 특정 PWM 생성기에 데이터를 보내 PWM을 생성할 수 있도록 하는 전용 명령어를 설계 및 구현하여 사용자 프로그램에서 처리가 가능하도록 설계하였다.

## 3. 신경회로망 제어기 설계

위치 제어를 위한 고전적인 PD 제어기의 단점을 해결하기 위해 PD 제어기에 신경회로망을 적용하는 신경회로망 제어기를 설계하였다.

조인트 공간에서 로봇의 동역학은 아래와 같이 나타낸다.

$$D(q)\ddot{q} + H(q, \dot{q}) = \tau \quad (1)$$

$D(q)$ 는  $n \times n$  inertia matrix,  $H(q, \dot{q})$ 는  $n \times 1$  Coriolis와 centrifugal forces,  $q, \dot{q}, \ddot{q}$ 는  $n \times 1$  joint angle vector, joint velocity vector, joint acceleration vector 그리고  $\tau$ 는  $n \times 1$  torque vector이다.

선형 PD 제어기는 다음과 같다.

$$\tau = K_p e + K_D \dot{e} \quad (2)$$

여기서  $e = q_d - q$ 이며,  $K_p$ 와  $K_D$ 는  $n \times n$  제어 이득 행렬이다. (1)은 비선형이고 coupled 되어 있어 (2)의 제어기로는 원하는 출력 응답을 얻기 어렵다. 따라서 각 축에 신경회로망으로 보상을 해주면 로봇의 비선형성을 해결할 수 있다.

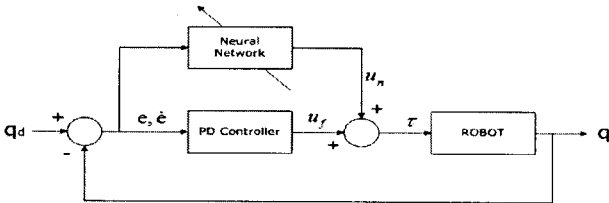


그림 2. PD 기반의 신경회로망 제어기의 블록도

신경 회로망 제어기는 다음과 같고 그림2에 나타나있다.

$$\tau = u_f + u_n \quad (3)$$

여기서  $u_f = K_p e + K_D \dot{e}$ 이며,  $u_n$ 은 신경회로망의 보상 입력이다.

사용된 신경회로망은 입력층, 은닉층 그리고 출력층에 각각 12개, 12개, 그리고 6개의 뉴런을 가지는 구조로 되어있다. 신경회로망에 사용되는 목적 함수는 PD제어기의 출력으로 구성한다.

$$E = \frac{1}{2} u_f^T u_f, \quad u_f = f(e, \dot{e}) \quad (4)$$

신경회로망에 사용되는 비선형 함수는 Gaussian Function이 사용되었으며, 아래와 같이 표현된다.

$$\phi_j(X) = \exp\left(-\frac{\|X - \mu_j\|^2}{\sigma_j^2}\right) \quad (5)$$

여기서  $X$ 는 입력 벡터  $X = [x_1, x_2, \dots, x_n]^T$ 이며,  $\mu_j$ 는  $j$ 번째 은닉층의 mean value 그리고  $\sigma_j$ 는  $j$ 번째 은닉층의 covariance이다.

## 4. 실험

아래 그림 3은 실험을 위한 휴머노이드 로봇팔이다. 각 팔은 6축의 자유도를 가지며, 로봇 머리의 팬-틸트 제어를 위한 2자유도의 모터가 장착되어 있다. 그림 4는 로봇 팔의 위치 제어 결과 중 일부를 나타내고 있다. 위의 그래프는 로봇 팔의 3번째 축의 조인트 공간에서의 제어되고 있는 움직임

을 보여주고 있다. PD 기반의 제어기를 통해 제어되고 있는 결과와 NN-PD 제어 기반의 제어기를 통해 제어되어 있는 모습을 나타낸다.

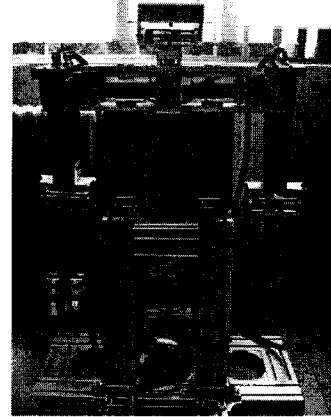


그림 3. 휴머노이드 로봇

아래 그림은 추종오차를 보여주고 있다. PD 제어에 의한 에러는 최대 약 0.08rad 정도의 오차가 발생하는 것에 비해 NN-PD 제어에 의한 오차는 약 0.025rad 이내인 것을 확인하였다.

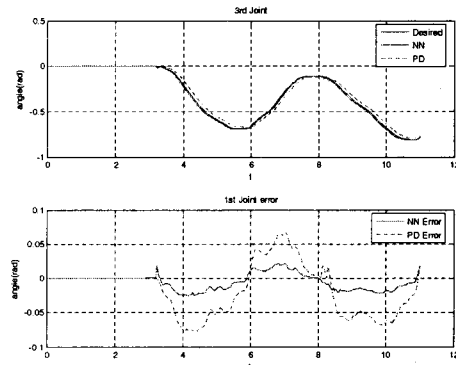


그림 4. 3번째 축의 PD 제어기와 NN-PD 제어기의 위치 추종 결과

## 5. 결 론

본 논문을 통해 FPGA 기반에서 부동소수점 프로세서를 설계 및 구현하여 이를 휴머노이드 로봇의 제어에 응용이 가능함을 보였다. 또한 많은 연산량을 요구하는 신경회로망을 구현하였으며, 설계된 프로세서에 기반한 신경회로망 제어기가 실제 로봇 팔의 위치를 제어하기에 충분함을 증명하였다. 현재까지의 연구는 조인트 공간에서의 위치제어에 응용을 했지만, 앞으로는 카테시안 공간에서의 위치제어를 위한 역기하학 연산과정을 포함하여 로봇의 위치제어에 대한 성능을 향상시키는 연구가 진행하고자 한다.

## 사 사

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과이며 지원에 감사드립니다.

## [참고 문헌]

- [1] Seul Jung and Sung su Kim, "Hardware Implementation of a Real-Time Neural Network Controller With a DSP and and FPGA for Nonlinear Systems", IEEE Transaction on Industrial Electronics, vol.54, No.1, pp.265-271, 2007
- [2] S. Himavathi, D. Antitha, and A. Muthuramalingam, "Feedforward Neural Network Implementation in FPGA Using Layer Multiplexing for Effective Resource Utilization", IEEE Transactions on Neural Network, vol.18, No.3, pp.880-888, 2007
- [3] Martin J. Pearson, A. G. Pipe, B. Mitchinson, K. Gurney, C. Melhuish, I. Gilhespy, and M. Nibouche, "Implementing Spiking Neural Networks for Real-Time Signal-Processing and Control Applications: A Model-Validated FPGA Approach", IEEE Transactions on Neural Network, vol.18, No.5, pp.1472-1487, 2007
- [4] Jeong-Seob Kim and Seul Jung, "Implementation of Neural Network Hardware Based on a Floating Point Operation in an FPGA", ICMIT2007, Dec. 6-7, 2007, Gifu, Japan