2003 정보 및 제어 학술회의 (2003.11.21-22)

FPGA를 이용한 2축 보간기의 설계

Design of a 2-axis interpolator using FPGA

여 수 진*, 김 종 은**, 원 종 백***, 박 종 식****

*경북대학교 전자공학과(전화:(053)940-8839, E-mail : mulsil@hotmail.com)
**경북대학교 전자공학과(전화(053)940-8839, E-mail : jekim@ajinextek.com)
***경북대학교 전자공학과(전화(053)940-8839, E-mail : wjb@aginextek.com)
****경북대학교 전자공학과(전화(053)940-8839, E-mail : jspark@elecscom.com)

Abstract : In this paper, we designed the digital pulse motor control chip including a circular interpolation function. The proposed algorithm in this paper is a nonparametric curve generation algorithm (Jordan's algorith) and a very simple algorithm. So the design for this algorithm used a small number of gates. Also an average error is fairly low. The max output speed is 4Mpps(Pulse per second), the max input frequency is 16MHz and the chip is useful for the stepping and servo motors.

The software contains one or two, and many axes linear interpolation algorithm and two axes circular interpolation algorithm.

Keywords : VHDL, interpolation, Nonparametric curve generation, FPGA, Motor control chip

I. 서론

산업용 로봇, CNC 선반, 반도체 장비 등 고속 정밀 제어 메카트로닉스 장비에 사용되는 모터의 등, 가감속 및 센서 신호 입력에 의한 실시간 제어를 위해 정확한 속도로 펄스를 출력하고 외부 센서와 인터페이스가 가능한 전용 모터 제어칩의 사용이 요구된다. 이러한 전용 칩의 사용으로 인하여 여러 가지 모터 드라이브를 구동하게 될 경우 칩 자체 내에 다양한 펄스의 출력 사항을 가지고 있다면 다른 회로를 첨가하지 않고 간단한 사용자 프로그램 만으로도 여러 가지 모터 드라이브에 알맞은 펄스를 제공할 수 있다.

또한 현대 산업계에서 사용하고 있는 메카트로닉스 장비는 마이크로프로세서 기술의 발달과 ASIC제품의 고속화에 발맞추어 점점 더 고속 정밀한 제어를 요구하고 있다. 특히 고속으로 원호 및 직선보간 기능을 포함하는 모션제어를 하기 위해서는 마이크로프로세서만으로는 모든 복잡한 계산을 할 수 없으므로 모션 제어 전용 ASIC을 사용하여야 한다. 따라서 원호 및 직선보간 전용의 모션 제어칩을 개발하여 FA 산업계 전반에 걸쳐 사용되고 있는 위치제어에 해결책을 제시할 수 있다.

본 논문에서는 기존에 제시되었던 Nonparametric curve generation (Jordan) 알고리즘을 이용하여 경로 오차를 최소화한 원호 및 직선 보간기능을 갖는 모션제어기를 구현하였으며, 경로를 이동하는 속도적용을 용이하게 하고 실제 기구부의 적용 오차를 최소화하기 위한 방안들은 제시하였으며 설계된 FPGA칩과 2축 기구부 연동을 통한 확인 실험 및 연속 보간, 선속 일정등에 관한 추가 기능등을 검증하였다.

II. 모션 제어기의 구조도 및 기능

본 장에서는 Jordan 알고리즘을 이용하여 구현한 모션 제어시스템의 전체적인 구조도와 각각의 블록에 대해 기술한다. 설계된 모션 제어기는 크게 4가지 드라이브로 구현이 가능하며, 각각의 드라이브는 모터를 제어하기 위해 발생되는 펄스를 기본으로 하여 가감속률 (rate1/2/3) 에 따라 대칭 또는 비대칭의 사다리꼴, 혹은 S자 곡선으로 표현되는 속도 프로파일을 가지게 된다.

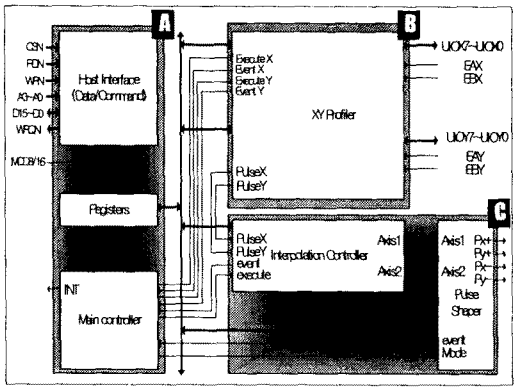

그림 1. 모션제어기의 전체적인 구조도

Fig 1. Block diagram of a motor control chip

그림 1은 모션 제어기의 전체적인 구조도이다. 그림 1의 A블록은 외부 HOST와의 접속, 내부 레지스터 파일 및 각종 제어신호를 생성하는 블록이다. 특히 이 블록은 연속보간을 위한 보간 상태 레지스터 및 다음 보간 데이터 및 명령을 제어하는 기능을 담당하고 있다. B블록은 X축과 Y축의 속도 정보 및 가감속 정보를 가지는 출력 펄스를 생성한다. 또한 외부 2상 인코터 신호를 입력받아 외부 모터의 현재 위치를 감시할 수 있게 하고 외부 접점을 제어, 감시 하기 위한 범용 입출력 8개에 대한 제어 기능을 한다. 다음으로 C블록은 본 논문에서 제시한 보간 알고리즘 기능을 구현한 것으로 내부의 속도 정보를 가지는 펄스의 입력을 받아 설정된 보간 데이터를 바탕으로 직선 및 원호 보간을 하여 최종 출력 펄스를 출력한다. 최종 출력 펄스는 단상 또는 2상 모드를 지원하며 active level을 가변할 수 있다.

III. 모션 제어 알고리즘

3.1 직선 및 원호 보간 알고리즘

Nonparametric curve generation algorithm 은 기존의 analog 적인 접근 방식이었던 digital differential analyzer 와는 다른 digital 적인 접근 방식이다.

Jordan 알고리즘은 2차원 상에서 Nonparametric 방법을 사용한 보간으로 다음의 <식1>의 형태로 수학적으로 정의된 함수에 대하여 적용된다.

$$f(x,y) = ax^2 + by^2 + cx + dy + exy + f = 0 \tag{1}$$

우선 two-dimension curve의 함수를 $f(x,y)=0$로 두었을 경우, 이 함수가 연속적인 도함수를 가지고 있다고 가정 하였을 때 식(1)의 f의 도함수들은 다음의 식(2)와 같이 표현할 수 있다.

$$f_x = \frac{\delta f}{\delta x} \qquad f_y = \frac{\delta f}{\delta y}$$
$$f_{xx} = \frac{\delta^2 f}{\delta x} \qquad f_{xy} = \frac{\delta^2 f}{\delta_i \delta x} \qquad f_{yy} = \frac{\delta^2 f}{\delta y^2} \tag{2}$$

한정된 grid상에서 식(1)에 대하여 보간을 수행하기 위해서 시작점은 f위의 한 점에 있다고 가정할 수 있으며, 이때 x와 y의 변화치를 1이라고 가정한다. 즉 보간의 시작점은 항상 보간 함수가 그리는 선위에 있으며 이후의 보간 결과로 생성되는 좌표는 그렇지 않을 수도 있으며, 연속 적으로 출력되는 보간 결과는 보간 함수와 한정된 grid상에서 가장 가까운 좌표가 선택된다는 것이다. 이러한 가정 하에서 다음의 그림 2와 같은 상황을 적용한다.

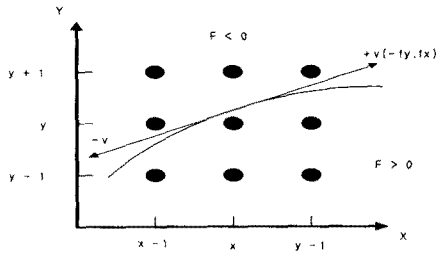

그림 2. 보간함수 및 step, 방향

Fig 2. interpolation function and step, direction

그림 2는 현재 좌표(x,y)에서 보간 함수 f를 +v 방향 또는 -v 방향으로 보간하는 것을 보여 준다. 현재 좌표(x,y)에서 보간 동작으로 진행할 수 있는 좌표는 (x+dx, y), (x+dx, y+dy) 그리고 (x, y+dy) 이며 dx, dy는 ±1이므로 8가지가 된다. 이 때 +v 및 -v에 따라서 현재 좌표에 대한 도함수의 부호와 관계하여 dx,dy를 정해 줄 수 있고 정해진 dx와 dy를 사용하면 현재의 점에서 다음 보간 위치로의 진행방향은 3가지로 좁혀진다.

즉 dx와 dy가 정해지기 때문에 (x+dx, y), (x+dx, y+dy) 그리고 (x, y+dy)로 보간 진행 방향의 경우의 수가 줄어들게 된다. 정해진 3가지 경우에 대하여 보간 함수와 가장 가까운 점을 선택해야 하는데 그 방법은 다음의 계산 과정을 따른다. 먼저 각 진행좌표에 따른 함수의 값을 다음과 같이 정의한다.

$$f^x = f(x+\Delta x, y)$$
$$f^y = f(x, y+\Delta y) \quad (3)$$
$$f^{xy} = f(x+\Delta x, y+\Delta y)$$

위의 세 가지의 값들 중에서 하나를 선택하는 방법은 바로 error criteria를 이용하는 것인데, 보간을 시작하는 시작점은 함수f의 커브 상에 존재하는 좌표이므로 |f|의 값은 항상 0의 값을 가지게 된다. 따라서 |f|의 값은 original curve상에서 얼마만큼 떨어져 있느냐를 판단할 수 있는 값이 된다. 즉 |f|의 값을 에러 값이라고 볼 수 있다. 그러므로, 위 세가지 함수의 $|f^x|, |f^y|, |f^{xy}|$값을 서로 비교하여 가장 작은 값을 가지는 함수를 선택하여 다음 스텝에 이동하게 될 좌표로 삼는다. 보간 수행 시 가지게 되는 에러 값을 수식으로 표현하면 다음과 같다.

$$\begin{aligned} f &= x^2 + y^2 - r^2 \\ f^x &= (x+\Delta x)^2 + y^2 - r^2 \\ &= x^2 + 2x\Delta x + \Delta x^2 + y^2 - r^2 \\ &= (x^2 + y^2 - r^2) + 2x\Delta x + \Delta x^2 \\ &= f + 2x\Delta x + \Delta x^2 \\ &= f + f_x\Delta x + \Delta x^2 \quad (\text{단, } \Delta x = \pm 1) \end{aligned} \quad (4)$$

$$|f^x| = |f + f_x\Delta x + \Delta x^2| \quad (5)$$

위의 수식에서 볼 수 있듯이 모든 연산이 덧셈과 shift연산으로 해결가능한다. 또한 Δx의 값이 ± 1의 값을 가지므로 에러 값이 증가했다 감소했다 하기 때문에 에러가 쌓이거나 포화상태가 되는 경우가 없음을 알 수 있다.

위의 과정을 통해 구해진 진행 좌표의 함수 값들 중 제일 작은 값 즉 보간 함수와 가장 가까운 좌표로 다음의 보간 좌표를 갱신한다. 예를 들어 |f^x|의 절대치가 |f^y| 및 |f^xy|보다 작으면 현재의 좌표 x는 x + dx로 좌표 y 는 현재 값을 유지하게 된다. 또한 연속적인 보간을 위해 다음좌표에서 필요한 변수 값을 현재의 좌표정보로부터 갱신해 주어야 하는데 표 1에서 사용되는 fx 및 fy에 대한 계산은 다음의 식 (6)와 같이 Taylor 전개를 사용하여 쉽게 얻을 수 있다.

$$\begin{aligned} f_x &\leftarrow f_x + f_{xx}\Delta x + f_{xy}\Delta y + \frac{1}{2}f_{x^3}(\Delta x)^2 \\ &\quad + 2f_{x^2y}(\Delta x)(\Delta y) + f_{xy^2}(\Delta y)^2) + \cdots\cdots \\ f_y &\leftarrow f_y + f_{yy}\Delta y + f_{yx}\Delta x + \frac{1}{2}f_{y^3}(\Delta x)^2 \\ &\quad + 2f_{y^2x}(\Delta y)(\Delta x) + f_{yx^2}(\Delta x)^2) + \cdots\cdots \end{aligned} \quad (6)$$

다음의 식 (7)는 f^x 및 f^y, f^xy에 대한 갱신 계산이다.

$$\begin{aligned} f^x &\leftarrow f^\alpha + f_x\Delta x + \frac{1}{2}f_{xx}(\Delta x)^2 + \cdots \\ f^y &\leftarrow f^\alpha + f_y\Delta y + \frac{1}{2}f_{yy}(\Delta y)^2 + \cdots \\ f^{xy} &\leftarrow f^\alpha + f_x\Delta x + f_y\Delta y + \frac{1}{2}f_{xx}(\Delta x)^2 \\ &\quad + 2f_{xy}(\Delta x)(\Delta y) + f_{yy}(\Delta y)^2 + \cdots \end{aligned} \quad (7)$$

식 (7)에서 f^alpha는 현재 위치 좌표의 보간 함수 값을 의미하며 초기 시작점에서는 0을 가지며 보간 진행 중 f^x, f^y 그리고 f^xy중 그 절대값이 가장 작은 값으로 갱신된다. 다음은 제시한 알고리즘의 flow chart 이다.

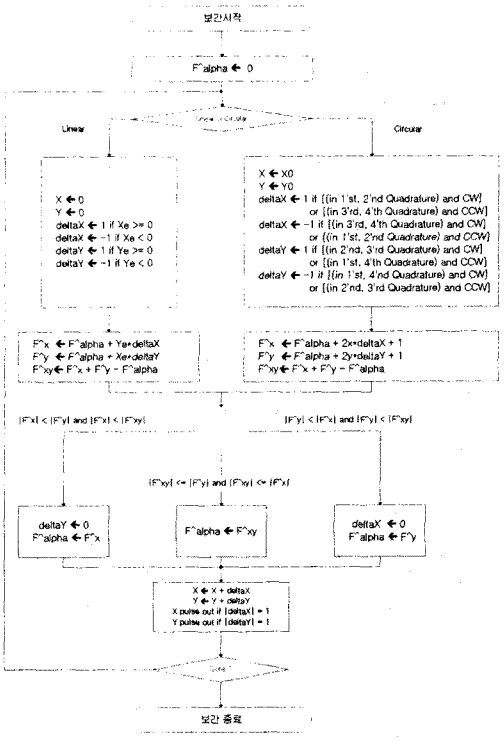

그림 3. 보간 알고리즘의 flow chart
Fig 3. Flow chart of the algorithm

3.2 보간 알고리즘의 검증

다음의 그림 4는 MATLAB을 사용한 직선보간 중 시뮬레이션 출력의 경로 오차 발생정도를 보여준다. 즉 현재 위치를 시작점(0,0)으로 보았을 때 종점(213,91)으로 보간 동작 수행시 기울기 91/213의 선과 그에 대한 ±0.5LSB 오차 직선을 그었을 때 실제 직선 보간 경로는 ±0.5 LSB직선 안에 위치함을 알 수 있다.

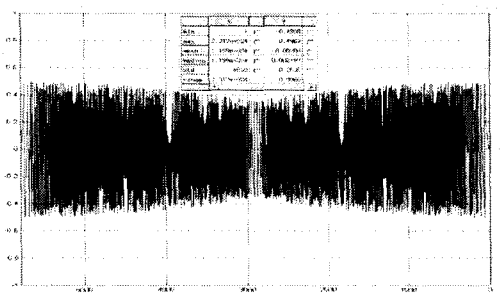

그림 4.. Jordan 원호보간 경로오차
Fig 4. average error of circular interpolation

IV. 모션 제어기의 설계와 구현

4.1 보간 기능 블록의 설계

III장 1절에서 기술한 Jordan 알고리즘을 이용하여 직선 및 원호보간 기능 블록을 설계하였다. 다음의 그림 5는 Jordan 알고리즘을 이용한 원호 및 직선 보간 기능 블록의 구조도이다.

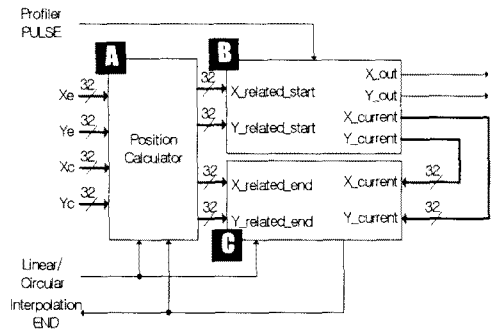

그림 5 . 원호보간 및 직선 보간 기능 블록의 구조도
Fig 5. Block diagram of the interpolation block

그림 5의 A 부분은 보간 변수들의 계산을 수행하며 B 블록은 보간 변수들로부터 다음 단계의 보간 데이터를 출력하는 기능을 담당하며 C 블록은 입력된 종점좌표를 바탕으로 보간 완료판단을 한다. 특히 직선보간의 경우 x와 y축의 종점 좌표에 정확하게 도달 후 완료할 수 있으나 원호보간에서는 원의 중심 좌표로 결정된 반지름을 가지는 보간 함수의 보간 오차가 +-0.5LSB가 되도록 종점 좌표를 설정해 줄 필요가 있다.

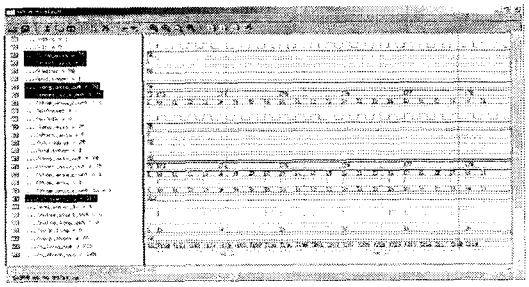

그림 6. 원호보간 시뮬레이션 결과
(R=78, Y:장축, X:단축)
Fig 6. Output result of a simulation

4.2 성능 검증

제작된 응용보드 구동을 위한 소프트웨어를 Visual C를 이용하여 테스트 프로그램을 만들었으며 성능 평가에 적용하였다. 다음의 그림 7은 개발 칩의 성능측정 소프트웨어의 실행 화면이다.

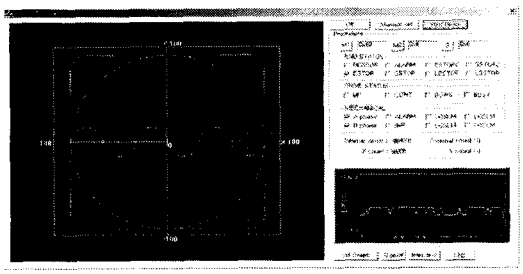

그림 7. 성능 측정
Fig 7. performance result

V. 결론

본 논문은 다축 보간기의 구현방안을 제시하여 산업계 전반에 걸쳐 요구되는 위치제어에 대한 실용적이고 구체적인 해결책을 제시한다. 본 논문에서는 제안된 보간기를 FPGA로 구현하여 그 신뢰성을 검증하였으며 향후 제안된 보간기는 모션제어 ASIC을 위해 활용될 수 있을 것이다.

참고문헌

[1] Bernard W. Jordan, "An improved algorithm for the generation of nonparametric curves", IEEE Transaction on computers, VOL. C-22. NO. 12 1973

[2]PER E. DANIELSSON, "Incremental Curve Generation," IEEE TRANSACTIONS ON COMPUTER, VOL. C-19, NO. 9. SEPTEMBER 1970.

[3] William E. Wright, "Parallelization of Bresenham's Line and Circle Algorithms," IEEE Computer Graphics & Applications 1990.