1999년도 대한전기학회 정기총회 및 추계학술대회 논문집 1999. 11. 20

이산푸리에변환을 이용한 계전 알고리즘의 마이크로컨트롤러에 적용

안용진 . 강상희 . 이승재 . 최면송
명지대학교

A Relaying Algorithm Based on Discrete Fourier Transform and Its Application to Micro-Controller

Yong-Jin Ahn . Sang-Hee Kang . Seung-Jae Lee . Myeon-Song Choi
Myongji University

Abstract - In view of the importance of DFT(Discrete Fourier Transform) in spectrum analysis, its computation efficiency is a topic. This paper presents calculation time to extract the power frequency at a fault signal using DFT. Furthermore, it is tested a relaying algorithm based on modified DFT and its application to Micro-controller.

1. 서 론

보호 계전기는 계통의 고장을 신속 정확하게 검출하고 제거하여 고장으로 인한 영향을 최소화하고, 계통 설비를 건전 계통으로부터 차단시킴으로써 고장난 설비를 보호하는 것과 함께 건전 계통으로 고장이 파급되는 것을 방지하여 계통의 안정 운전을 도모하는 역할을 한다.

디지털 보호 계전기에서 고장 신호의 샘플과 샘플간의 시간사이에 전력주파수 성분을 추출하고 계전 알고리즘을 수행하여 트립신호를 차단기에 보내기 위해서는 전력주파수 성분의 추출에 사용되는 시간이 제한적이며 정확한 전력주파수 성분을 추출하기 위해서는 많은 샘플링이 필요하다. 이러한 이유 때문에 DFT(Discrete Fourier Transform)를 이용하여 전력주파수 성분을 추출하는 경우에는 연산시간과 연산량이 많기 때문에 전력주파수 성분의 추출이나 계전 알고리즘 수행을 위한 시간을 충분히 확보하기 어려웠다.

본 논문에서는 마이크로컨트롤러의 프로그래밍에서 이미 알고 있는 값을 쉬프트 처리하는 방법을 이용하여 주기당 샘플링은 더 많으면서 연산시간은 적게 소요되는 변형된 DFT를 이용하여 전력주파수 성분을 추출하고, 계전 알고리즘의 수행을 저가(低價)의 마이크로 컨트롤러를 사용하여 구현한다.

2. 이산푸리에변환과 마이크로컨트롤러

2.1 이산푸리에변환[1]

고장 신호를 X(t)를 다음과 같이 M개의 주파수 성분으로 이루어 진 것으로 가정하면 다음과 같이 나타낼 수 있다.

$$\begin{aligned} X(t) &= \sum_{n=0}^{M-1} X_n \sin(n\omega_1 t + \theta_n) \\ &= \sum_{n=0}^{M-1} (X_n \sin\theta_n \cos n\omega_1 t + X_n \cos\theta_n \sin n\omega_1 t) \\ &= \sum_{n=0}^{M-1} \{X_{Real}(n)\cos n\omega_1 t + X_{Imag}(n)\sin n\omega_1 t\} \end{aligned} \quad (1)$$

식 (1)에서 ω_1은 데이터 폭에 해당하는 시간을 한 주기로 가지는 주파수를 의미한다. 신호 X(t)를 샘플간격 T로 샘플링하고 k번째 샘플 값을,

$$X[k] = \sum_{n=0}^{M-1} \{X_{Real}(n)\cos n\omega_1 kT + X_{Imag}(n)\sin n\omega_1 kT\} \quad (2)$$

라고 하면 이 샘플 값들로부터 주파수 $k\omega_1$의 페이서 $X_{Real}(n)$과 $X_{Imag}(n)$는 식 (3)에 의해 구해진다.

$$\begin{aligned} X(n) &= X_{Real}(n) + jX_{Imag}(n) \\ &= \sum_{k=0}^{N-1} X[k] e^{-j(\frac{2\pi nk}{N})} \\ &= \sum_{k=0}^{N-1} X[k](\cos\frac{2\pi nk}{N} - j\sin\frac{2\pi nk}{N}) \end{aligned} \quad (3)$$

식 (3)에서 구해지는 $X_{Real}(1)$과 $X_{Imag}(1)$은 주파수 ω_1에 해당하는 페이서의 실수부와 허수부를 나타내며 $X_{Real}(0)$과 $X_{Imag}(0)$은 직류성분에 대한 페이서인데 직류는 허수부가 없으므로 $X_{Imag}(0)=0$이 된다

2.2 마이크로컨트롤러[2]

마이크로 프로세서는 컴퓨터의 중앙 기억 장치(Central Processing Unit : CPU)를 단일 LSI 칩에 집합시켜 만든 반도체 소자이다. 반면 마이크로컨트롤러는 CPU는 물론이고 I/O 인터페이스, RAM, ROM 등의 메모리까지 하나의 칩에 실현되었으므로 시스템이 매우 간단하고 설계가 용이하여 여러 분야에 널리 사용된다.

원칩 마이크로 프로세서는 8비트, 16비트 그리고 32비트 계열 등이 있다. 32비트는 16비트에 비해 연산속도는 빠르나 입출력 기능이 현저히 부족하므로 마이크로컨트롤러로써 응용성이 떨어졌다. 따라서 간단한 제어 시스템에서는 8비트 계열이 사용되지만, 복잡하고 정밀한 연산이 필요한 경우는 16비트 계열을 사용했다. 최근에는 32비트의 응용성이 향상되어 16비트를 대치하고 있다. 본 논문에서는 16비트 계열의 80C196KC를 사용하였다.

고장 입력 신호에서 전력주파수 성분을 추출하기 위한 데이터는 80C196KC 내의 A/D 컨버터를 거쳐 일정 메모리에 저장되어 있는 샘플 값을 번지지정 방법에 의해 호출하여 사용한다. 번지지정 방법에는 Resister direct, Immediate Direct, Indirect, Indirect Auto increment, Short Indexed, Long Indexed 등이 있으며, 산술연산에 사용되는 명령어들의 연산시간을 알아보면 다음과 같다.

표 1. 산술명령 연산시간
(단위: state time , 1state time=125[nsec])

구 분	Direct	Immediate	Indirect		Index	
			Normal	AutoInc	Short	Long
덧셈	4	5	6~8	7~9	6~8	7~9
뺄셈	4	5	6~8	7~9	6~8	7~9
곱셈	16	17	18~21	19~22	19~22	20~23
나눗셈	26	27	28~31	29~32	29~32	30~33
쉬프트	6 ~ 7					

표 1에서 덧셈과 뺄셈 명령어가 곱셈과 나눗셈의 명령어보다 약 3~6 배 정도 연산 시간이 적게 소요됨을 알 수 있다.

2.3 이산푸리에변환의 마이크로컨트롤러에 적용

2.3.1 일반적인 DFT 연산

고장 신호 X(n)를 그림 1과 같이 가정하고, 삼각함수의 Basis는 그림 2와 같다.

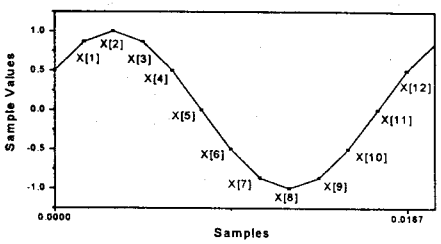

그림 1. $X(n)=\sin(2\pi nk/N+\theta)$

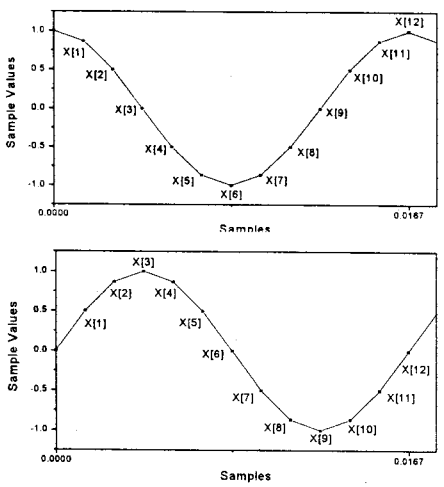

그림 2. Cosine, Sine Basis[12 samples/cycle]

1주기 DFT를 이용한 $X(n)$의 기본파 실효값의 제곱은 다음과 같이 구할 수 있다.

$$X_{Real(12)}=\frac{\sqrt{2}}{12}\sum_{k=1}^{12}X[k]\cos(\frac{2\pi k}{12}) \quad (4)$$

$$=\frac{\sqrt{2}}{12}\{X[1]\cos(2\pi/12)+X[2]\cos(4\pi/12)+\cdots$$
$$\cdots+X[11]\cos(22\pi/12)+X[12]\cos(24\pi/12)\}$$

$$X_{Imag(12)}=\frac{\sqrt{2}}{12}\sum_{k=1}^{12}X[k]\sin(\frac{2\pi k}{12})$$

$$=\frac{\sqrt{2}}{12}\{X[1]\sin(2\pi/12)+X[2]\sin(4\pi/12)+\cdots$$
$$\cdots+X[11]\sin(22\pi/12)+X[12]\sin(24\pi/12)\}$$

$$X_{Rms(12)}{}^2=X_{Real(12)}{}^2+X_{Imag(12)}{}^2 \quad (5)$$

2.3.2 변형된 DFT 연산

그림 1, 2를 참고하여 식 (4)를 다시 표현하면 식 (6)과 같다.

$$X_{Real(12)}=$$
$$(X[1]-X[5]-X[7]+[11])(\sqrt{2}/12)\cos(30^\circ)$$
$$+(X[2]-X[4]-X[8]+[10])(\sqrt{2}/12)\cos(60^\circ) \quad (6)$$
$$+(X[12]+[6])(\sqrt{2}/12)$$

$$X_{Imag(12)}=$$
$$(X[1]+X[5]-X[7]-[11])(\sqrt{2}/12)\sin(30^\circ)$$
$$+(X[2]+X[4]-X[8]-[10])(\sqrt{2}/12)\sin(60^\circ)$$
$$+(X[3]-[9])(\sqrt{2}/12)$$

$(\sqrt{2}/12)\cos(30^\circ)=0.10206\fallingdotseq 2^{-4}+2^{-5}+2^{-7}=0.10156$
$(\sqrt{2}/12)\sin(60^\circ)=0.10206\fallingdotseq 2^{-4}+2^{-5}+2^{-7}=0.10156$
$(\sqrt{2}/12)\cos(60^\circ)=0.05892\fallingdotseq 2^{-5}+2^{-6}+2^{-7}=0.05468$
$(\sqrt{2}/12)\sin(30^\circ)=0.05892\fallingdotseq 2^{-5}+2^{-6}+2^{-7}=0.05468$
$(\sqrt{2}/12)=0.11785\fallingdotseq 2^{-4}+2^{-5}+2^{-6}+2^{-7}=0.11718$

또한, 그림 1과 같은 고장 신호를 1주기에 16번 샘플링 하였을 경우에 신호 X(n)의 기본파의 실수부와 허수부는 식 (7)과 같다.

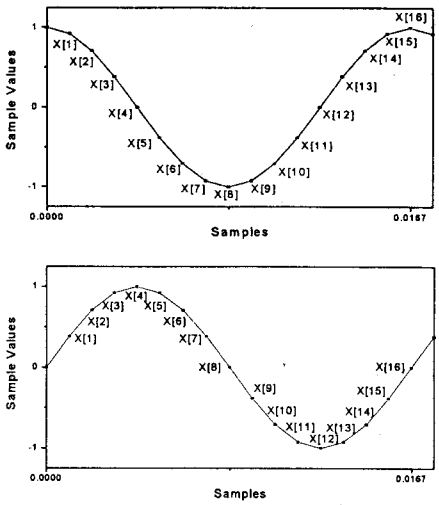

그림 3. Cosine, Sine Basis[16 samples/cycle]

$$X_{Real(16)}=\frac{\sqrt{2}}{16}\sum_{k=1}^{16}X[k]\cos(\frac{2\pi k}{16}) \quad (7)$$
$$=(X[1]-X[7]-X[9]+[15])(\sqrt{2}/16)\cos(22.5^\circ)$$
$$+(X[2]-X[6]-X[10]+[14])(\sqrt{2}/16)\cos(45^\circ)$$
$$+(X[3]-X[5]-X[11]+[13])(\sqrt{2}/16)\cos(67.5^\circ)$$
$$+(X[16]-[8])(\sqrt{2}/16)$$

$$X_{Imag(16)}=\frac{\sqrt{2}}{16}\sum_{k=1}^{16}X[k]\sin(\frac{2\pi k}{16})$$
$$=(X[1]+X[7]-X[9]-[15])(\sqrt{2}/16)\sin(22.5^\circ)$$
$$+(X[2]+X[6]-X[10]-[14])(\sqrt{2}/16)\sin(45^\circ)$$
$$+(X[3]+X[5]-X[11]-[13])(\sqrt{2}/16)\sin(67.5^\circ)$$
$$+(X[4]-[12])(\sqrt{2}/16)$$

$(\sqrt{2}/16)\cos(22.5^\circ)=0.08166\fallingdotseq 2^{-4}+2^{-7}=0.07812$
$(\sqrt{2}/16)\sin(67.5^\circ)=0.08166\fallingdotseq 2^{-4}+2^{-7}=0.07812$
$(\sqrt{2}/16)\cos(45^\circ)=0.0625=2^{-4}=0.0625$
$(\sqrt{2}/16)\sin(45^\circ)=0.0625=2^{-4}=0.0625$
$(\sqrt{2}/16)\cos(67.5^\circ)=0.03382\fallingdotseq 2^{-5}=0.03125$
$(\sqrt{2}/16)\sin(22.5^\circ)=0.03382\fallingdotseq 2^{-5}=0.03125$
$(\sqrt{2}/16)=0.08838\fallingdotseq 2^{-4}+2^{-6}+2^{-7}=0.08593$

일반적인 DFT연산과 변형된 DFT연산과의 차이점은 일반적인 DFT는 샘플별로 연산을 행한 후에 덧셈을 하고, 변형된 DFT는 삼각함수의 부호를 미리 데이터에 반영한 후에 공통된 항을 쉬프트로 처리한 점이다. 이때 2^{-k}는 마이크로컨트롤러의 어셈블러 프로그래밍에 사용되는 쉬프트명령의 계수를 표시한 것으로 DFT계산결과에서의 최대오차는 변형된 12샘플DFT에서는 7.1[%], 변형된 16샘플DFT에서는 7.5[%]이다.

표 2. 산술명령 연산시간 비교

구 분	덧셈(뺄셈)	곱셈	쉬프트	연산시간
12샘플 DFT	23	50		145.1[μsec]
변형 12샘플 DFT	33	2	20	51.37[μsec]
변형 16샘플 DFT	33	2	14	46.12[μsec]

표 2는 식 (5), 기본파 실효값의 제곱을 얻기까지의 명령어 사용횟수이며 일반적인 DFT 연산방법에서 삼각함수 값은 미리 계산한 테이블 값을 사용한다고 하면, 변형된 16샘플 DFT는 일반적인 DFT 보다 약 1/3배 시간이 적게 소요되고, 변형된 12샘플 DFT와는 비슷한 시간이 소요된다. 따라서 단순 수치비교로 변형된 16샘

플 DFT가 변형된 12샘플 DFT와 일반적인 DFT보다 연산시간이 적게 소요되면서, 2^{-k}의 합으로 표시하는 과정에서의 오차는 비슷함을 알 수 있다.

2.3.3 비율차동 계전방식

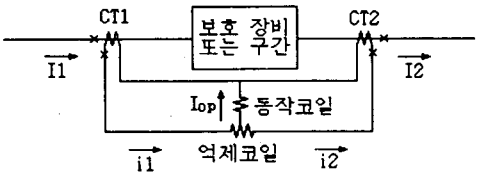

그림 4. 비율차동 계전방식

비율차동 계전방식은 변류기의 특성 차에서 일어나는 불평형 전류가 동작코일로 흐르기 때문에 오 동작을 막기 위하여 통과전류로 억제력을 발생시키도록 한 방식으로, 외부 사고전류 등의 과대 전류가 통과할 때에는 큰 차전류가 동작코일로 흐르지 않으면 계전기는 동작하지 않고, 통과전류가 적을 때는 적은 차전류만으로 동작하는 원리이다. 본 논문에서는 식 (8)을 사용하여 사례연구에 이용하였다.

$$|i1+i2| > (|i1|+|i2|)\cdot I_{top} \qquad (8)$$

3. 사례연구 및 분석

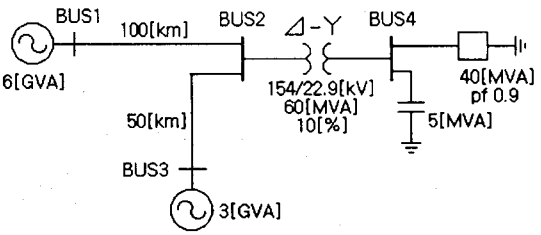

그림 5. 모의 계통

본 논문에서 제안한 DFT 연산방식의 타당성을 검증하기 위하여 80C196KC 2개를 사용한 3상 변압기 보호용 비율차동 계전기를 구성하였다. 계전 알고리즘의 수행은 각 80C196KC에서 계산된 3채널씩의 실효값을 가지고 판단한 후에 트립신호를 출력했다. 비율차동 계전 알고리즘의 시험용 데이터는 전력계통 과도현상 해석 프로그램인 EMTP(Electro-Magnetic Transient Program)를 이용하여 그림 5. 모의 계통의 단상변압기 3대 전류신호 6채널을 출력한 후, D/A 변환기인 PCL-727을 통하여 취득하였다.

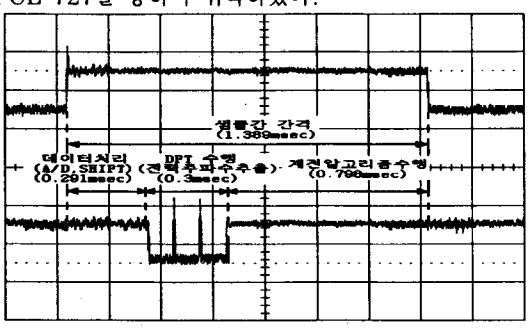

그림 6. 변형된 12샘플 DFT 시간사용

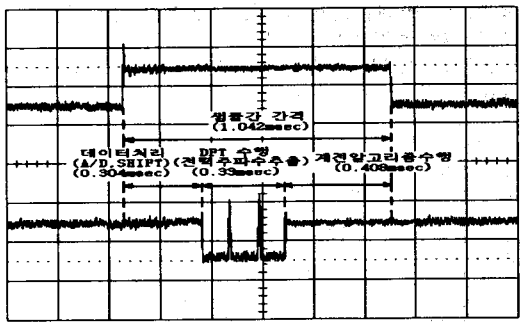

그림 7. 변형된 16샘플 DFT 시간사용

그림 6과 7은 변형된 12샘플 DFT와 16샘플 DFT의 시간사용을 도시한 것으로 DFT수행 시간은 DFT를 수행하기 위해 데이터를 읽어들일 때부터 실효값을 저장할 때까지의 시간이다. 표 2의 단순 수치상의 비교에서는 변형된 12샘플 DFT와 16샘플 DFT가 비슷한 연산시간 소요되었으나, 실제 어셈블러 프로그램 상에서는 DFT 연산과정을 수행하기 전과 후에 사용된 명령어들(데이터를 읽어들이거나 데이터간의 크기 비교, 실효값을 저장)의 시간 사용으로 16샘플 DFT가 약간의 시간이 더 소요되었다. 계전 알고리즘의 수행은 변형된 12샘플 DFT와 16 샘플 DFT가 고장신호 입력 시에 트립신호를 출력하였다. 따라서 변형된 16샘플 DFT가 12샘플 DFT 보다 주기당 샘플링은 많이 하면서도 제한된 시간 내에 사용 가능함을 보였다.

변형된 DFT연산에서 이미 알고있는 값을 쉬프트 처리하기 위하여 2^{-k}의 합으로 표시할 때는, 표 1에서 쉬프트 3번은 곱셈과 비슷한 연산시간을 가지므로 3번 이상 쉬프트의 경우에는 곱셈과의 연산시간 비교 후 선택하는 편이 좋다. 또한 연산소요시간을 더 줄이고 싶다면 식 (6)에서 공통이 되는 2^{-k}를 추출해서 A/D 과정 후에 수행하는 쉬프트 과정에서 처리하면 된다. 쉬프트의 이유는 8비트 A/D 결과 앞에 00000000를 채워 워드 데이터를 만들어 연산 시에 오버플로를 방지하기 위함이다. 따라서 $2^{-(8+k)}$ 쉬프트하면 된다. 이 경우 쉬프트한 결과가 바로 Cosine이나 Sine Basis가 곱해진 값으로 나오는 경우가 생기는데 이런 데이터는 덧셈과 뺄셈 연산에 그냥 사용하면 된다

4. 결 론

고속의 마이크로컨트롤러는 고가(高價)이기에 경제성 있고 널리 사용되고 있는 저가(低價)의 마이크로컨트롤러를 사용한 모의 계전기에서 고장 신호의 샘플과 샘플간의 시간간격 사이에 DFT를 이용해서 전력주파수 성분을 추출하고 계전 알고리즘을 수행할 수 있다는 것을 확인하였다. 일반적인 DFT에서 문제 시 되었던 연산량과 연산시간을 변형된 DFT를 이용해서 연산시간이 많이 소요되는 덧셈과 곱셈의 사용을 줄이고 이미 알고 있는 값을 쉬프트 처리함으로써 연산량과 연산시간을 줄일 수 있었다. 또한 변형된 DFT간의 비교에서 연산에 소요되는 시간은 비슷하면서도 1주기 당 샘플링을 많이 함으로써 정확한 전력주파수 성분을 추출할 수 있었고, 계전 알고리즘을 수행할 수 있었다.

[참 고 문 헌]

[1] 조경래, "송전선의 과도특성을 이용한 고정밀 거리계전 알고리즘", 공학박사학위논문, 서울대, pp 28-29, 1996

[2] 김대근, 정순배, 김재희, "인텔 80C196KC의 모든 것", 도서출판 ohm사, 1988