

Fault Location Identification Using Software Fault Tolerance Technique

金 元 河[†] · 張 湧 元^{*} · 韓 丞 洙^{**}

(Wonha Kim · Yong-Won Jang · Seung-Soo Han)

Abstract - The management of technological systems will become increasingly complex. Safe and reliable software operation is a significant requirement for many types of system. So, with software fault tolerance, we want to prevent failures by tolerating faults whose occurrences are known when errors are detected. This paper presents a fault location algorithm for single-phase-to-ground faults on the teed circuit of a parallel transmission line using software fault tolerance technique. To find the fault location of transmission line, we have to solve the 3rd order transmission line equation. A significant improvement in the identification of the fault location was accomplished using the N-Version Programming (NVP) design paradigm. The delivered new algorithm has been tested with the simulation data obtained from the versatile EMTP simulator.

Key Words : Fault tolerance, NVP, Newton-Raphson, 유전자 알고리즘, EMPT

1. 서 론

산업화의 발전으로 더 많은 전력을 소비하며 안정된 시스템을 요구하게 되었다. 기계의 생명 줄과 같은 전력 에너지는 보다 안정하고 신뢰할 수 있도록 공급되어야 한다. 그럼에도 불구하고 전력계통의 대부분의 고장은 송전선로에서 일어난다. 빠른 수리를 위해 송전선로의 고장 위치를 정확하게 파악할 수 있어야 한다. 송전선로에서 발생하는 각종 계통함수의 정보를 상시 감시하여 고장발생시 신속한 고장 복구가 가능하게 정확한 위치를 찾아주는 고장점 표정 장치는 전력계통에 중요한 항목으로 대두되었다[1][2]. 송전선로 상에서 고장발생시 고장 점까지의 거리를 계산하는 고장점 표정알고리즘은 최근 십여 년 동안 활발히 연구되어왔다. 적용대상 계통은 크게 1회선 계통[2], 병행 2회선 계통[3], 분기점을 갖는 다단자 계통[1][4], 2회선 계통에서 건전회선의 영상분 전류를 이용한 방법으로 구분할 수 있다[5]. 병행 2회선으로 가설된 송전선로의 한 회선이 T분기된 경우 분기되지 않은 회선 1선 지락 사고 발생시 자기단의 정보만을 이용한 방법도 발표되고 있다[6].

이전 연구에서는 다양한 송전선로의 회선 연결에 따른 다양한 고장 점을 찾는 알고리즘을 연구했다. 그래서 본 논문에서는 기존 논문을 이용하여 보다 정확한 고장 점을 얻기

위해 software fault tolerance의 N-Version Programming (NVP) 모델을 적용했다.

Fault tolerance는 시스템의 일부가 고장이 나도 시스템 전체가 영향을 받으면 안 된다는 개념으로 시스템 자체에 오류로부터 강인하게 설계한다[7][8]. Fault tolerance는 크게 software fault tolerance와 hardware fault tolerance로 분류하며 각각의 특성은 fault tolerance의 개념을 소프트웨어적으로 구현 하나 아니면 하드웨어적으로 구현 하나에 따라 구분된다. 본 연구를 수행함에 있어서 Software fault tolerance의 NVP[8], Recovery Block (RcB) [9][10], N-Version Self-Checking Programming (NSCP) [11][12]의 기본 모델의 연구하였고, 보다 정확한 결과를 얻기 위해 NVP와 RcB 모델을 통합한 모델에 대한 연구[13] 및 다수의 version으로부터 얻은 결과 중 최상의 결과를 결정하는 Decision Mechanism (DM)을 연구했다[14][15].

복잡한 시스템에서 소프트웨어적인 계산 중에 작은 오류가 계속 누적되면 시스템 전체에 커다란 영향을 미치는 오류가 될 수 있기 때문에 software fault tolerance system에서는 강인성을 주어 올바른 결과를 산출한다. 전력계통은 산업사회의 가장 중요한 자원요소로 계통의 고장이 발생할 때 정확한 위치를 찾는 것이 매우 중요하며, 이를 위해 software fault tolerance의 NVP 모델을 적용한다. NVP 모델은 forward recovery 방식으로 다수의 개별 알고리즘(version)을 병렬로 계산한 후 얻은 결과 값들을 조합하여 가장 최적의 값을 산출하는 방식이다. 다수의 개별 알고리즘은 병렬로 계산되는 방식으로 실시간 계산을 필요로 하는 시스템에서 사용하기 적합하나 동시에 계산된 다수의 결과를 조합하여 가장 올바른 값을 산출해야 하기 때문에 오류 여부를 검사하는 DM(Decision mechanism)의 구성이 중요하다.

* 교신저자, 學生會員 : 慶熙大學 電子學科 副教授 工博
E-mail : wonha@mju.ac.kr

** 正會員 : 明知大學 情報學科 碩士課程

*** 正會員 : 明知大學 情報學科 副教授 · 工博

接受日字 : 2004년 10월 7일

最終完了 : 2005년 1월 18일

그림1과 같이 NVP 모델의 기본 구성은 n개의 개별 알고리즘(version)을 사용하며 각 version으로부터 얻은 결과를 모아 최상의 결과값을 찾는 DM이 필요하다. 구성할 시스템의 특징에 따라 version의 수와 DM의 방법은 실험적으로 결정된다. NVP 결과 데이터의 정확도를 높이기 위해서는 많은 version을 사용하면 되지만 시스템의 부하를 수반하기 때문에 시스템의 기능에 따라 결정해야 하며 보통 3개 이상의 version을 사용한다.

본 논문에서 적용한 1회선 분기점을 갖는 병행 2회선 송전선로의 고장점 표정 알고리즘으로 유도된 거리 방정식은 미지수가 2개인 3차 복소 비선형방정식이다. 기존에는 비선형 방정식의 해를 구하기 위해 Newton-Raphson 반복 계산법을 이용했다. 그러나 Newton-Raphson 방법은 계산 초기에 미지수의 초기값을 결정해줘야 하는데 이 초기값에 따라 계산 결과의 수렴 여부에 많은 영향을 받게 되는 문제를 가지고 있다. 이러한 Newton-Raphson 반복법에 있는 문제를 해결하고 신뢰할 수 있는 결과를 얻기 위해 본 연구에서는 NVP 모델을 적용했다. NVP 모델에서 사용한 version의 수는 3개이며 각각의 종류는 Newton-Raphson, 유전자 알고리즘, 대수 방정식이다. 또한 Newton-Raphson의 미지수 초기값 결정 문제를 해결하기 위해 유전자 알고리즘으로 얻은 결과를 초기값으로 사용하였다. 대수 방정식은 MATLAB의 solve 함수를 이용하여 계산하였다.

제안한 방법의 실험을 위한 송전선로의 고장 데이터는 EMTP(Electro Magnetic Transients Program)을 사용하여 154[kV], 25[km] 분기된 병행 2회선 송전선로에서 고장지점과 고장저항의 데이터를 얻었으며 구현은 MATLAB을 이용해서 시뮬레이션 했다.

실험결과를 보면 Newton-Raphson의 미지수 초기값에 따른 오류가 발생하지 않았으며 기존 논문에서 얻은 고장점 거리의 결과보다 더 정확한 결과를 얻었고 계산 중간에 발생할 수 있는 오류에 대한 강인성을 부여할 수 있었다.

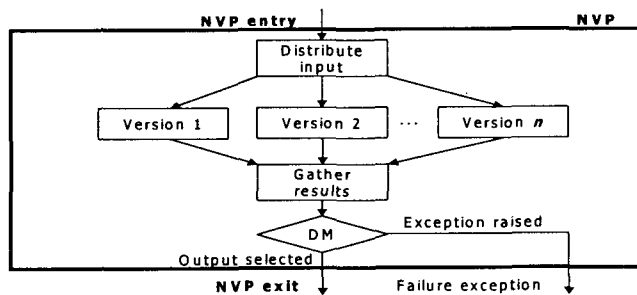


그림 1. NVP의 기본 모델
Fig. 1 Fundamental model of NVP

2. 고장점 표정 알고리즘

본 논문에서는 Sang-Hee Kang등이 제안한 1회선 분기점을 갖는 병행 2회선 송전선로의 고장점 표정 알고리즘에 새로운 NVP 계산 모델을 적용했다[6].

1회선 분기점을 갖는 병행 2회선 송전선로에서의 고장점 표정은 분기점을 기준으로 안쪽과 바깥쪽의 고장 점을 찾는 방정식이 다르다. 안쪽과 바깥쪽에 따른 고장점 표정 방정

식은 다르나 기본 형태는 식 (1)과 같다.

$$a_3p^3 + a_2p^2 + a_1p + s_1RFp + a_0 + s_0RF = 0 \quad (1)$$

식 (1)은 미지수 거리 p와 저항 RF에 관한 3차 비선형 방정식으로 유도된다. 고장 위치에 따라 적용되는 상수 (a[3:0], s[1:0])가 다르게 유도된다. 유도된 방정식은 복소수 방정식으로 안쪽이나 바깥쪽의 고장점 계산은 실수부와 허수부를 나누어지며 이들 방정식을 이용하여 계산한다. 즉 안쪽 고장 점을 구하기 위해서는 실수 안쪽 방정식과 허수 안쪽 방정식등 두개의 방정식을 풀면 된다.

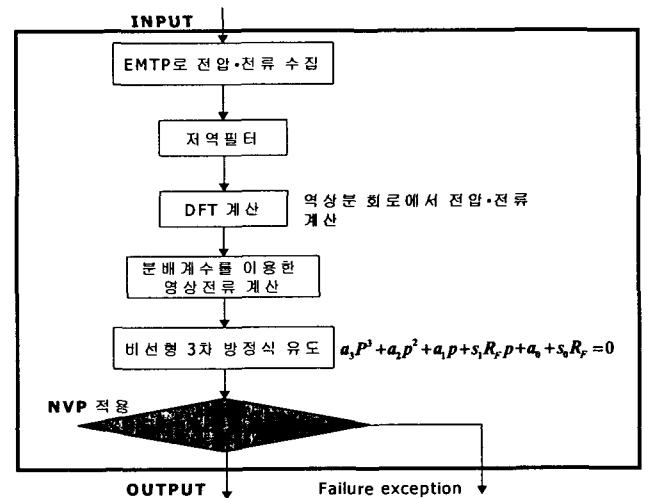


그림 2. 1회선 분기점을 갖는 병행 2회선 송전선로의 계산 과정
Fig. 2 Computation process of parallel 2 wired transmission line with 1teed point

그림 2는 논문[6]에서 제안한 순서를 블록도로 표시한 것이다. 이 블록도는 고장 거리를 계산하기 위해 고장점 표정 알고리즘을 이용하는 방법을 보여준다. 먼저, EMTP를 이용하여 사고 거리와 저항에 따른 고장 데이터를 수집한다. 수집한 데이터를 low pass 필터에 통과한 후 DFT(Discrete Fourier Transform)를 사용하여 역상분 회로에서 전압과 전류를 계산한다. 타 회선의 영상 전류와 고장 전류를 알기 위해 분배계수를 이용하여 유도된 식을 정리하면 식(1)과 같은 미지수가 2개인 3차 복소 비선형 방정식을 얻게 되는데, 이러한 3차 비선형 방정식을 풀기 위해 NVP를 적용한다.

3. 제안한 NVP 알고리즘

계산 순서 중 비선형 3차 방정식을 유도하는 과정까지는 기존 논문에 준하여 구현한다. 제안한 비선형방정식의 계산 모델은 그림 2의 과정 중 유도된 거리방정식 계산 과정에 적용하여 보다 정확한 고장 거리를 찾는데 이용한다. 기존 논문에서는 비선형 방정식을 계산하기 위해 Newton-Raphson 반복 계산법을 사용했으나 미지수의 초기값 설정 문제가 일어날 수 있어 미리 고장 거리의 근사한 값을 찾아 주는 최

적화 알고리즘을 사용하여 보정한다. 또한 시스템 내부의 소프트웨어 오류로 인한 고장 거리의 오차를 줄이기 위해 NVP 모델을 적용한다. NVP 모델은 다수의 version을 이용하여 병렬로 처리된 결과를 모아 최상의 결과 값을 결정해주는 모델이다. 원하는 결과를 찾아주는 DM은 시스템의 특성에 맞게 실험적으로 만들어야 하며 DM의 성능에 따라 결과의 정확도가 결정된다.

본 연구에서 사용한 NVP 모델의 version의 수는 3개이고 각각의 종류는 Newton-Raphson(NR), 유전자 알고리즘(GA), 대수 방정식(AE) 이다. Newton-Raphson의 초기값 결정 문제를 해결하기 위해 GA를 먼저 수행하여 나온 결과 값을 Newton-Raphson의 초기값으로 사용하여 문제를 해결했다. NVP 모델의 결과를 결정해 주는 DM은 version의 결과 중에 식(1)을 0에 가까운 값으로 만드는 결과를 선택하도록 했다.

그림 3은 그림 2에서 유도된 거리 방정식 계산 과정에 NVP 모델을 적용한 새로운 계산 모델이다. 식 (1)에 해당하는 상수들을 계산하기 위해 다음과 같이 계산한다.

- Step 1:** 식 (1)에 해당하는 상수를 계산한다.
 - Step 2:** 입력받은 데이터를 가지고 GA를 적용해서 근사적인 solution 영역의 한 점을 취한다.
 - Step 3:** Step 2의 첫 세대에 구한 GA 결과를 이용해서 Newton-Raphson 반복법을 사용하여 계산한다.
 - Step 4:** Step 2의 결과를 받아 GA를 k 세대만큼 재생성하여 결과를 계산한다.
 - Step 5:** Step 2의 첫 세대에 구한 GA 결과를 이용해서 AE를 구한다.
 - Step 6:** 과정 3, 4, 5에서 구한 각 version의 결과를 모아 각각의 결과가 0과 1 사이에 위치하는지 검사한다. 만약 위치하지 않는 version의 결과가 있으면 제외시킨다.
- $$0 < V_i < 1 \quad i=1,2,3$$
- Step 7:** 각 version의 결과 중에 서로 근접하는 2개의 결과를 찾아 평균값을 취한다.

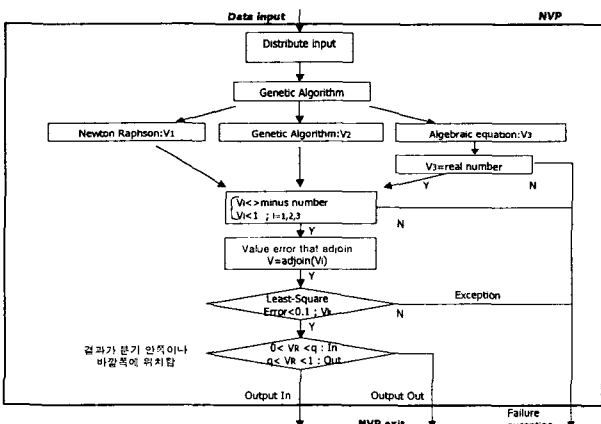


그림 3 거리 방정식을 계산하기 위한 NVP 모델
Fig. 3 NPV model for calculating the distance equation

Step 8: Step 7에서 구한 결과(V)를 36개 모아 최소제곱법을 이용하여 최소 제곱 오차가 0.1보다 작은 V_k를 계산한다 (VR).

Step 9: 과정 8에서 구한 VR의 범위가 분기점의 안쪽인지 아니면 바깥쪽인지를 계산하여 고장 거리를 찾아준다.

4. 실험 및 결과

본문의 알고리즘을 검증하기 위해 EMTP를 사용하여 154[kV] 병행회선길이 25[km]의 계통에서 사고모의를 하였다. 선로전체를 1로 하는 [pu]단위를 사용한다. 그림4는 EMTP를 이용하여 테스트한 시스템의 모델로 S단과 R단의 기울기는 10°이고 S단과 T단은 5°로 기울어 있다. 시스템은 60Hz로 작동하며 한 사이클이 36개의 샘플링을 취하는 시스템으로 전체 샘플링 주파수는 2160Hz 이다.

고장점 표정에 영향을 미치는 요소는 부하전류, 고장점 저항, 분기점의 위치, 사고발생 위치, 전원 임피던스의 변화 등이 있다. 따라서 이 고장 발생 조건을 변화시켜가며 오차의 변화 추이를 살펴보았다. 사고위치는 0.1, 0.5, 0.9[pu] 3단계로 나누었고 분기점의 위치는 0.2, 0.8[pu] 2단계로 나누어 실험 데이터를 얻었으며 자세한 조건은 부록에 첨부했다.

이때 오차는 전체 송전선로 길이를 기준으로 한 것으로 다음과 같다.

$$\text{오차} = \frac{\text{실제 고장거리} - \text{계산된 고장거리}}{\text{전체 송전선로의 길이}} \times 100[\%] \quad (2)$$

제안한 NVP 모델은 Newton-Raphson, 유전자 알고리즘, 대수 방정식의 3개 version을 사용한다. Newton-Raphson의 초기값은 유전자 알고리즘의 1세대 배양으로 계산된 결과를 초기값으로 사용하여 적용한다. 대수 방정식은 MABTAL의 solve 함수를 이용하여 계산하였다. 유전자 알고리즘은 세대수와 배양 조건에 따라 결과에 영향을 준다. 실험에 사용한 조건은, 최대 반복횟수 (maxgen)는 100에서 100단위로 증가하여 300까지 3단계로 나누어 계산했고, 집단 크기 (popsize)는 50에서 50단위로 증가하여 150까지 3단계로 나누어 계산했으며 염색체의 길이 (lchrom)=10으로 고정하여 총 실험은 36가지의 각기 다른 실험 조건의 데이터를 가지고 9번 반복하여 계산했다. 이전 세대의 최적 개체를 다음 세대에 전달해 주기 위해 엘리트 전략(elitist strategy)을 적용했다.

제안한 NVP 모델의 성능을 평가하기 위해 기존의 논문에서 계산한 Newton-Raphson과 제안한 NVP의 결과를 비교했다. 또한 계산 도중에 발생할 수 있는 오류에 강인성을 측정하기 위해 임의로 노이즈를 추가하여 강인성에 대한 테스트를 실험했다. 강인성 테스트의 환경은 사고 위치가 0.5[pu] 지점이고, 분기점의 위치는 0.8[pu] 지점일 때로 하였으며, 유전자 알고리즘의 조건은 maxgen (최대 반복횟수)=100, popsize (집단 크기)=60, lchrom (염색체의 길이)=10 로 하여 알고리즘을 수행하였다.

NVP의 version의 추가는 결과에 대한 정확도를 높이는 반면 계산 프로세서의 부하를 증가시켜 계산 시간의 증가를

수반한다. 그래서 계산 시간에 대한 실험도 병행하여 실험을 했다. 실험의 조건은 고장 위치를 찾는 실험에 포함하여 같은 조건에서 수행했으며 총 36번의 실험 조건에서 일어난 시간의 평균을 계산했다.

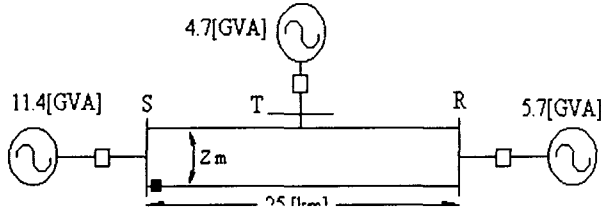


그림 4 EMPT 모의계통
Fig. 4 EMPT Simulated tube

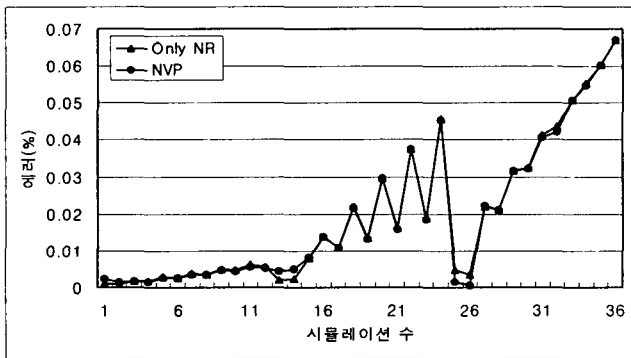


그림 5 NVP 결과와 오직 NR만 사용한 결과에 따른 에러 비교
Fig. 5 Error comparison between NVP results and results used by NR only

부록에 나와 있는 바와 같이 각기 다른 실험 조건으로 36번의 시물레이션을 하였으며, 그림 5는 각 시물레이션 조건에 따라 나오는 NVP 결과와 이전 논문에 결과인 NR 결과를 비교한 그래프이다. 이를 보면 거의 비슷한 결과를 보이나 전체적으로 NVP의 결과가 더 좋게 나온다. 명확한 비교를 위해 36번의 실험 오차를 누적한 누적 오차를 비교하면 $NR=0.693924$ 이고 $NVP=0.68757$ 가 되어 약 0.006354(0.91%)만큼 NVP의 오차가 적게 나온다. 그래프를 보면 분기점의 위치에 따른 오차는 거의 차이가 없으나 고장 저항의 크기가 클수록 오차가 커짐을 알 수 있으며 시물레이션 수가 13에서 24까지 보면 굴곡이 나타나는데 이는 분기점 안쪽에 고장이 발생한 경우와 분기점 바깥쪽에 고장이 발생한 경우 사용된 방정식이 다르기 때문이다.

유전자 알고리즘의 실험조건에 따른 결과는 표 1과 같이 거의 차이가 없다. 그러나 maxgen와 popsize의 변수 값이 클수록 계산시간은 증가한다. 표 1의 계산 시간은 popsize가 50에서 50씩 증가하여 150까지 3 단계로 나누어 계산했을 때 걸린 시간을 모두 더한 값이다. maxgen의 변화에 따른 계산 시간을 보면 maxgen의 값이 클수록 오차는 줄어드나, 계산 시간이 증가하기 때문에 사용 목적에 따라 선택해야 한다. 그러나 maxgen의 증가에 따른 오차의 변화가 거의 없기 때문에 작은 값인 maxgen=100을 사용해도 무방함을 알 수 있다.

표 1. maxgen 변수에 따른 GA의 누적 오차

Table 1. GA accumulated errors according to max generation

maxgen 값	100	200	300
누적 오차(%)	0.687570273	0.687549803	0.687541557
계산 시간(min)	0.548835	1.082784	1.586307

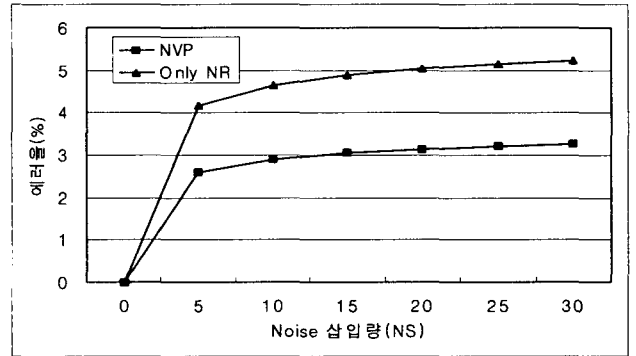


그림 6 강인성 테스트를 위한 노이즈 삽입에 따른 에러 비교 (삽입 조건: $a_2 = a_2 + a_2 \times NS$)

Fig. 6 Noise inserted error comparison for testing robustness (noise condition : $a_2 = a_2 + a_2 \times NS$)

그림 6은 시스템의 소프트웨어가 이상하여 오류가 계산식에 포함되거나 소수점 이하의 작은 자리 값에 반올림에 의한 작은 오차가 누적되어 발생할 수 있는 계산상의 오류로부터 강인성을 테스트한 결과를 보여주는 그래프이다. 임의의 노이즈를 상수 a_2 에 다음 수식을 이용하여 노이즈를 삽입해서 테스트했다.

$$\text{노이즈 삽입 방법: } a_2 \leq a_2 + a_2 \times NS$$

노이즈의 삽입에 영향을 주는 NS는 a_2 의 배수로 증가시켜 고장 거리 방정식의 노이즈를 추가한다. 강인성 결과를 보면 기존 Newton-Raphson만 사용한 경우보다는 NVP를 적용한 경우가 오류에 강인함을 알 수 있다. NVP 모델을 적용한 제안한 방법이 기존 방법보다 약 40% 더 정확한 결과를 나타낸다. 오류에 대한 강인성은 고장위치에 대한 신뢰를 주는 요소로 시스템에 무엇보다 중요한 요소이다.

표 2. NVP 알고리즘의 계산 시간(minute) 비교

Table 2 Comparison of NVP algorithm computation time (minute)

Test environment	AMD1800 - 1.53GHz DDR 133 - 512M	
Newton-Raphson	유전자 알고리즘	대수 방정식
0.0134m	0.5134m	0.5544m

표 2는 NVP 모델의 version에 따른 계산 시간을 나타낸다. 프로세서의 계산 시간은 시스템의 특성에 따라 중요한 선택 요소가 될 수가 있다. 계산 시간을 측정된 컴퓨터의 사양은 AMD 1800이고, 메모리는 DDR133 512M를 사용한다. 유전자 알고리즘의 조건은 maxgen=100, popsize=50, lchrom=10으로 설정하여 테스트를 하였다. 기존 논문은 Newton-Raphson만 사용했기 때문에 고장 거리를 찾는 데 소비한 시간이 0.0134분인 반면 NVP 모델을 사용한 경우는 1.0812(0.0134+0.5134+0.5544)분이 소비된다. 전력계통의 대부분의 고장이 송전선로에서 일어나는 상황에서 정확한 고장 위치를 찾기 위해 약간의 계산 시간의 증가는 감수해야 한다.

4. 결 론

전력 에너지는 안정하고 신뢰할 수 있도록 고장에 대한 빠른 대처가 필요하며 송전선로의 빠른 수리를 위해서는 정확한 고장 위치를 찾는 것이 중요하다. 본 논문에서는 기존 논문의 알고리즘을 이용하여 정확한 고장 위치를 찾는 알고리즘에 NVP 모델을 적용한 새로운 알고리즘을 제안했다.

제안한 방법은 기존 논문보다 약 0.91% 에러가 줄었으며 내부 오류에 대한 강인성은 약 40% 더 강인했다. 그러나 고장 거리를 찾는 계산 시간은 98.8% 더 소모되는 문제가 있다. 그러나 전력계통은 다른 시스템에 비해 중요한 계통으로 안정된 에너지의 보급이 중요하므로 계산 시간의 증가에 의한 손실보다는 정확도의 증가로 인한 이득이 더 크다고 볼 수 있다.

감사의 글

본 연구는 2003년도 산업자원부 전력산업연구개발사업 전력기술기초연구(과제번호 R-2003-B-286)에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다..

참 고 문 헌

- [1] M. Abe, N. Otsuzuki, T. Emura, M. Takeuchi, "Development of a New Fault Location System for Multi-Terminal Single Transmission Lines," IEEE Trans. on PWRD, Vol. 10, No. 7, 1995, pp.159-168
- [2] M. S. Sachdev, R. Agarwal, "A Technique for Estimating Transmission Line Fault Locations from Digital Impedance Relay Measurements", IEEE Trans. on PWRD, Vol. 3, No. 1, pp.121-129, 1988
- [3] A. Wiszniewski, "Accurate Fault Impedance Locating Algorithm", IEE Proceedings, Vol. 130, pt. C, No. 6, pp.311-314, 1983
- [4] R. K. Aggarwal, D. V. Coury, A. T. Johns, A. Kalam, "A Practical Approach to Accurate Fault Location on Extra High Voltage Teed Feeders", IEEE Trans. on PWRD, Vol. 8, No. 3, pp.874-883, 1993
- [5] 박홍규, 이명수, 이재규, 유석구, "송전선에서의 고저항 지락사고시 고장거리 추정에 관한 알고리즘", 대한전기학회 하계학술대회 논문집, pp.1363-1365, 1999
- [6] Sang-Hee Kang, Seung-Jae Lee, Young-Jin Kwon, Yong-Cheol Kang, "A Fault Location Algorithm for Parallel Transmission Line with a Teed Circuit," Power Engineering Society Summer Meeting, 2001. IEEE, Volume: 2, 15-19 July 2001 Pages:921 - 926 vol.2
- [7] Randell, B., "Software Structure for Software Fault Tolerance," IEEE Transactions on Software Engineering, 1,2, June 1975, pp. 220-232.
- [8] Avizienis, A., and Chen, L., "On the Implementation of N-Version Programming for Software Fault Tolerance During Execution," Proceedings 1st IEEE International Computer Science Applications Conference, Chicago, Illinois, November 1977, pp. 149-155.
- [9] Melliar-Smith P.M., B.Randell, "Software reliability: the role of programmed exception handling," SIGPLAN Notices 12(3), 1977, pp. 95-100.
- [10] Kim K.H. "Distributed execution of recovery blocks: an approach to uniform treatment of hardware and software faults, Proc. 4th International Conference on Distributed Computing Systems," IEEE Computer Society Press, 1984, pp. 526-532.
- [11] Yau S.S., R.C.Cheung, "Design of Self-Checking Software," Proc. Int. Conf. on Reliable Software, 1975, pp. 450 - 457.
- [12] Laprie J.C., J.Arlat, C.Beounes, K.Kanoun, "Definition and Analysis of Hardware and Software Fault Tolerant Architectures," IEEE Computer, 23(7), 1990, pp. 39-51.
- [13] Scott R.K., J.W.Gault, D.F. Mc Allister, "Fault tolerant software reliability modeling," IEEE Trans. on Software Engineering, 13(5), 1987, pp. 582-592.
- [14] Avizienis, A., "The N-Version Approach to Fault-Tolerant Software," IEEE Transactions on Software Engineering, Vol. SE-11, No. 12, 1985, pp. 1491-1501.
- [15] Lyu, M. R. (ed.), Handbook of Software Reliability Engineering, New York: McGraw-Hill, 1996.

저 자 소 개



김 원 하(金元河)
1985년 연세대학교 전자공학과 졸업
1988년 Univ. of Wisconsin-madison 전기 및 컴퓨터 공학과 졸업 (석사)
1997년 Univ. of Wisconsin-madison 전기 및 컴퓨터 공학과 졸업 (박사)
1997년~1999년 : 미 Los Alamos 국립연구소 연구원
2000년~2003년 : 명지대학교 통신공학과 조교수
2003년~현재 : 경희대학교 전자정보대학 부교수
관심분야: 멀티미디어 신호처리, 의료영상처리
Tel : (031) 201-2030
E-mail : wonha@mju.ac.kr



장 용 원(張湧元)
2000년 대구효성카톨릭대학교 메카트로닉스 학과 졸업(공학사)
2002년 명지대 정보제어학공과 졸업(공학석사)
現 차세대전력기술연구센터 연구원
관심분야는 Fault Tolerant, 자동제어
Tel : (031) 330-6761
E-Mail : jang122@orgio.net



한 승 수(韓丞洙)
1986년 연세대학교 전기공학과 졸업
1988년 연세대학교 전기공학과 졸업(석사)
1996년 조지아공대 전기 및 컴퓨터 공학과 졸업(박사)
2000년~현재 : 명지대학교 정보공학과 부교수
관심분야: Fault Tolerant, 신경회로망, 유전 알고리즘
Tel : (031) 330-6761
E-mail: shan@mju.ac.kr