

객체지향기반 과도 안정도 해석

Transient Stability Analysis Based on OOP

朴志皓[†]
(Ji-Ho Park)

Abstract - This paper presents the new method of power system transient stability simulation, which combines the desirable features of both the time domain technique based on OOP(Object-oriented Programming) and the direct method of transient stability analysis using detailed generator model. OOP is an alternative to overcome the problems associated with the development, maintenance and update of large software by electrical utilities. Several papers have already evaluated this approach for power system applications in areas such as load flow, security assessment and graphical interface. This paper applied the object-oriented approach to the problem of power system dynamics simulation. The modeling method is that each block of dynamic system block diagram is implemented as an object and connected each other. In the transient energy method, the detailed synchronous generator model is so-called two-axis model. For the excitation model, IEEE type1 model is used. The developed method was successfully applied to New England Test System.

Key Words : OOP, Transient stability, Direct method, Transient energy.

1. 서 론

과도 안정도 해석은 전력계통의 계획 및 운전에서 매우 중요하다. 과도안정도 해석은 시간영역에서의 모의방법에 기반을 두고 있다. 이 방법의 장점은 모델링에 제한이 없고 안정도 현상의 자세한 고찰을 위한 시스템의 모든 변수의 값들을 시간응답으로 볼 수 있다는 것이다. 하지만 시간영역모의 방법은 미분방정식을 풀기 위하여 일정시간 간격으로 적분하기 때문에 매우 느리다. 또한 예 또는 아니요형의 답만 제시한다. 즉 안정도를 유지하기 위한 임계고장 제거시간을 결정하기 위한 정보가 부족하다. 따라서 임계고장 제거 시간을 알기 위해서는 많은 시뮬레이션을 반복적으로 수행해야만 한다. 반면에 과도에너지함수(TEF: Transient Energy Function)에 기초한 과도안정도 직접해석법은 안정도 지수를 만들 수 있는 장점을 가지므로 상대적 안정도를 측정할 수 있다. 하지만 직접해석법은 시스템의 모델링에 많은 제약을 가지고 있고 UEP(unstable Equilibrium Point)[1]의 계산은 뉴턴법과 같은 과중한 수치계산이 필요한 다차원 최적화 문제를 포함하고 그 수렴특성도 좋지 않다. 본 논문은 전력계통의 과도안정도해석에 시간영역모의 방법의 장점과 직접해석법의 장점을 결합한 새로운 방법을 제안한다. 전력계통의 해석은 전통적으로 소프트웨어에 대

한 의존도가 높고 사용되는 소프트웨어 또한 규모가 크고 방대하다. 대규모의 소프트웨어의 관리는 매우 복잡하고 비용도 많이 든다. 객체지향프로그래밍(OOP)[2]은 대규모의 소프트웨어의 개발, 관리 그리고 업데이트 문제에 새로운 접근방법을 제공한다. 전력계통분야에 객체지향기법을 적용한 여러 논문들[3,4,5]이 발표되었는데 주요 적용분야는 전력조류계산, 안전도평가 그리고 그래픽 인터페이스등이다. 본 논문은 동적시스템해석 분야에 객체지향기법을 적용하여 과도안정도를 해석한다. 시간영역모의에 있어서 발전기 및 시스템의 모델링은 매우 방대하고 수식적으로 복잡하기 때문에 기존의 프로그램 방식으로는 모든 시스템에 적용 가능한 일반적인 해법을 찾기에는 한계가 있다. OOP를 이용한 시스템 모델링 방법은 시스템의 동적인 모든 요소들을 클래스(class)로 분해하여 물리적 의미를 구현하여 소프트웨어에서 객체화 시킨다. 이 방법의 가장 큰 장점은 시스템 모델링의 유연성이다. 시스템이 변하면 해당 부분만은 수정교체하여 모델링은 변경할 수 있고 제어장치의 보완 등에 따른 확장에도 유연하게 대처할 수 있다. 동적인 요소들을 객체 단위로 모델링하므로 재사용성이 매우 뛰어나다.

OOP를 이용한 동적시스템의 모델링은 제약을 가지지 않으므로 에너지함수를 이용한 안정도의 직접해석법에도 매우 유리하다. 즉 기존의 직접해석법의 시스템의 모델링 제약을 극복할 수 있다. 객체지향기법으로 모델링한 동적시스템을 시간영역에서 모의하고 그것의 결과를 에너지함수에 적용하여 시스템의 안정도를 유지하기 위한 임계고장제거시간을 결정할 수 있다.

[†] 교신저자, 正會員 : 慶北大 工大 전자전기컴퓨터공학부
계약교수

E-mail : pjh@ee.knu.ac.kr

接受日字 : 2007年 12月 26日

最終完了 : 2008年 2月 1日

2. 동적시스템의 객체지향적 모델링

전력계통은 그림 1과 같이 모선, 선로, 모선과 모선사이의 선로에 설치된 장치들, 부하, 모선에 설치된 장치들의 집합으로 표현할 수 있다. 선로에 설치된 장치는 단로기, 차단기, 변압기등 이고 모선에는 Shunt 장치를 들 수 있다. 전력계통의 구성요소들을 객체로 모델링하여 이들의 연결관계를 계통의 구성에 맞게 맺어주면 전력계통의 표현이 완성되는 것이다.

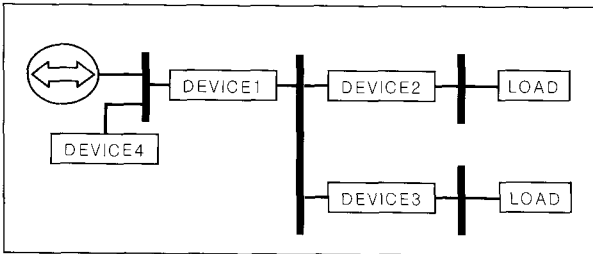


그림 1 전력계통의 구성

Fig. 1 The configuration of power system

전력계통의 동적인 움직임은 비선형 상미분방정식과 비선형 대수방정식으로 표현된다.

$$\frac{dx}{dt} = f(x, V) \tag{2-1}$$

$$I(x, V) = YV \tag{2-2}$$

여기서 x 는 상태벡터, f 는 비선형 벡터 방정식, I 는 주입전류 그리고 Y 는 어드미턴스 행렬이다.

미분방정식은 전력계통의 동적인 구성요소 즉 동기 발전기, 발전기의 제어장치, 터빈, 유도전동기 등의 동적인 움직임을 나타내고, 대수방정식은 발전기를 연결하는 회로망, 변압기 그리고 부하를 나타낸다. 식(2-1)로 표현되는 시스템의 요소는 프로그램에서 객체화 시킬 수 있는 최소단위로 분해하고 이들을 이진트리틀 이용하여 연결한 다음 수치해석 기법을 적용한다. 식(2-2)는 희소행렬(sparse matrix)[6]을 객체지향적으로 모델링하여 기존의 행렬연산보다 향상된 계산속도를 얻을 수 있다. 시스템을 객체 지향적으로 모델링할 때 자료구조는 프로그램의 재사용성을 높이고 유지 및 보수가 쉽다는 OOP의 이론적 틀에 충실해야 된다. 객체단위로 분해할 때 여러 가지 방법을 선택할 수 있지만, 시스템의 물리적 구조를 따르는 것이 가장 좋은 방법이다. 전력계통을 객체지향기법으로 모델링한 논문들이 있지만 이들 논문은 주로 전력조류계산과 소규모 전력계통에 관한 내용이다. 본 논문은 식(2-1),(2-2)로 표현되는 전력계통의 과도안정도에 관한 내용을 다룬다.

2.1 발전시스템 모델링

식(2-1)로 모델링되는 동적방정식의 해를 구하기 위하여 아날로그 컴퓨터[7]를 이용한다면 필요한 소자는 덧셈기, 곱셈기, 이득기 그리고 적분기등이다. 아날로그컴퓨터를 이용하는 경우는 대상 시스템을 미분방정식으로 모델링하고 이를 블록선도로 표현한 다음 각 블록에 해당하는 전자소자를

블록선도의 표현과 일치되게 연결하는 방식이다. 기본소자들은 이미 만들어진 하드웨어이므로 블록선도에 필요한 개수만큼 가져와서 연결하면 동적시스템을 모의할 수 있습니다. 이 방법의 장점은 매우 유연하다는 것이다. 즉 대상시스템이 변경되어 블록선도가 바뀌면 파편 블록선도에 맞게 연결만 새롭게 해주면 모의가 가능하다는 것이다. 이것을 디지털컴퓨터에서 객체지향적인 방법으로 구현하기 위하여 아날로그 컴퓨터의 기본 소자인 덧셈기, 곱셈기, 적분기, 이득기등의 물리적 작용을 그대로 묘사하는 클래스로 정의하고 이외에 시뮬레이션에 필요한 객체들 즉 제한기, 수학함수처리 객체등을 정의한다. 객체란 어떤 상태(state)를 나타내는 데이터의 구조와 동작을 수행하는 연산(operation)으로 이루어진 프로그램의 한 요소이다. 이들 기본 객체들은 한 번의 구성으로 모든 시뮬레이션에 사용할 수 있는 기본소자가 된다. 소자들을 객체로 표현하기 위해 C++언어의 class 키워드를 사용하여 객체(object)로 구성한다. 클래스를 구성할 때 중요한 것은 구성하고자 하는 대상의 물리적 작용을 정확하게 묘사하는 것이다. 그림 2는 덧셈기의 예이다.

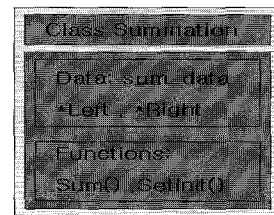


그림 2 덧셈기 클래스

Fig. 2 Class Summation

기본객체들을 사용하여 안정도 해석을 위한 발전기, 여자기, 안정기, 조속기등을 모델링할 수 있다. 각 시스템의 블록선도의 모든 블록은 객체로 표현되고 이들 블록을 서로 연결하여 시스템을 구성할 수 있다. 이와 같이 블록도를 기준으로 구성되어 입력된 객체간의 연결 정보는, 객체간의 실제 해석을 위한 연결을 필요로 하며, 앞에서 언급된바와 같이 이진트리틀 사용하여 구현된다. 그림 3은 동기발전기 이축모델의 과도 유기전압 방정식에 대한 기본객체로의 분해와 이를 이진트리틀로 연결하는 예이다. gain1~gain2는 이득기 객체, sum1은 덧셈기 그리고 Integ1는 적분기 객체이다.

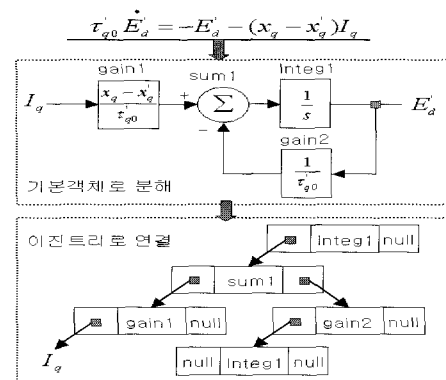


그림 3 이축모델의 객체연결구조

Fig. 3 Object connecting structure of two axis model

그림 3의 트리구조에서 트리의 개수는 적분기의 개수와 같다. 즉 주어진 시스템을 연립 1계미분방정식의 형태로 트리에 저장한다. 횡축전류 I_q 는 외부객체에서 넘어오는 값이므로 외부연결 구조를 마련해 둔다. 외부 연결트리는 여자기, 선로를 나타내는 트리와의 연결을 위한 것이다. 즉 외부의 여자기와 선로의 모델에 상관없이 트리로서 연결할 수 있다. 마찬가지로 발전기 모델에 상관없이 이와 같이 트리를 구성할 수 있다. 하나의 트리는 하나의 1계미분방정식을 표현하는 것이다. 트리에 연결된 객체들은 서로 다른 객체들이므로 바로 연결하는 것은 불가능하고 가상함수를 이용하여 클래스를 설계하면 서로 다른 객체들의 연결이 가능하다. 그림 3의 방정식은 1계미분방정식이므로 생성되는 트리의 개수는 1개이고 계산과정은 트리의 말단에서부터 시작한다. 미분변수 E_d' 는 현재 초기값을 gain2객체에 메시지 넘겨준다. 그러면 gain2객체는 이값에 자신의 이득값을 곱하여 저장한다. gain1객체는 외부객체로부터 전류메시지 I_q 를 받아 자신의 이득값을 곱하여 연산하고 저장한다. 다음으로 sum1객체는 이 두 이득기의 값을 더하여 저장하고 결과를 다시 적분기 객체 Integ1에 넘겨준다. 트리의 연결구조가 곧 적분을 위한 계산순서가 되고 이것은 블록선도로 모델링된 시스템의 구성과 일치한다. 각 객체가 메시지를 주고받을 때는 인간의 언어와 같은 메시지를 주고받는 것이 아니다. 오직 자신의 연산 결과만을 데이터로 넘겨주는 것이다. 이것은 중요한 문제이다. 객체지향기법을 엄격하게 적용하면 메시지의 분석처리를 위한 추가적인 필요과정이 필요하여 오버헤드(overhead)문제를 일으킨다. Smalltalk와 같은 언어는 매우 엄격한 객체지향언어이다. 이것의 장점은 사용자에게 아주 많은 편리를 제공한다는 것이다. 하지만 사용자가 의도하지 않은 많은 추가적인 코드가 발생하여 더 많은 메모리를 차지하고 수행속도도 느려진다. C++은 객체지향언어의 특징을 제공하는 이른바 하이브리드 언어이다. 프로그램의 작성시에 필요한 부분만 선택적으로 객체기법을 구현할 수 있는 것이다. 본 논문은 C++을 이용하였고 가장핵심적인 알고리즘 구현에만 객체지향적인 기법을 이용하여 오버헤드 문제를 발생시키지 않는다.

표 1은 그림 3을 모의하기 위한 객체정보이다. 아날로그 컴퓨터의 모의 방법과 같은 방법으로 설명할 수 있다. 먼저 그림 3에서 필요한 아날로그컴퓨터 소자는 이득기 2개, 덧셈기 1개 그리고 적분기 1개이다. 이들의 연결 관계는 이득기 gain1은 I_q 라는 외부신호와 연결되고 덧셈기 객체는 gain1과 gain2와 연결되고 그리고 적분기는 덧셈기와 연결되어 있다. 이러한 소자의 개수와 연결정보를 입력파일로 작성한다. 파일로 작성된 입력정보에서 필요한 객체를 선언하고 연결정보에서 그림 3과 같이 트리로 구성하면 시스템의 모델링이 완성되는 것이다.

표 1 이축 모델의 객체정보

Table 1 Object information of two axis model

객체번호.	객체이름	입력객체(번호)
0	gain1	I_q (외부연결객체출력)(Ex.)
1	gain2	Integ1(3)
2	sum1	gain1(0), gain2(1)
3	Integ1	sum1(2)

그림 3에서 시스템의 모델링이 변경되어 $\tau_{q0} \dot{E}_d' = -E_d' - (x_q - x_q')I_q + I_d x_q$ 로 되었다고 하면 곱셈기 객체를 추가하여 곱셈을 수행하고 덧셈기에 입력으로 넣어주면 된다. 이것은 트리구조의 변경으로 쉽게 변경된다. 덧셈기는 가변의 입력에 대하여 덧셈이 가능하도록 C++의 기본함수로 구현가능하다. 그림 4는 위에서 설명한 내용을 기초로 한 발전기 안정도 해석을 위한 발전시스템의 객체지향적 모델링 단계를 나타낸 것이다. 시스템을 구성하는 가장 기본적인 요소를 객체로 모델링하고 이들을 이진트리구조로 상호 연결하여 복잡한 시스템을 구성하는 방식이다. 따라서 객체지향적인 모델링 방식은 복잡한 시스템의 모델링에 있어 핵심요소를 먼저 객체로 구현하고, 이를 조립하는 방식을 채택함으로써 시스템의 모델링이 보다 명확하고 복잡성에서 오는 모델링의 오류를 확실하게 줄여준다.

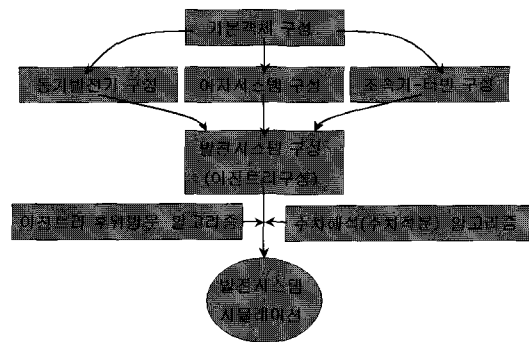


그림 4 발전시스템 모의를 위한 모델링 단계
Fig. 4 Modeling steps for generating system

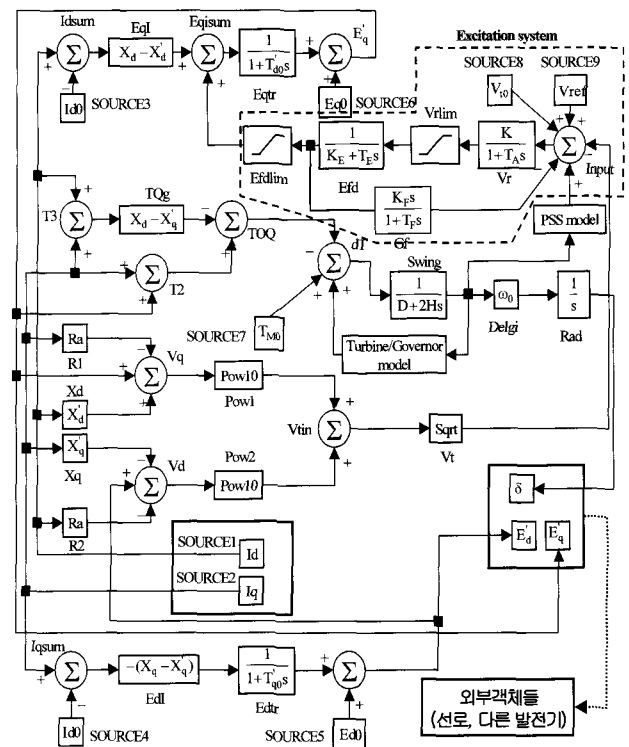


그림 5 발전시스템 객체연결 구성
Fig. 5 Object connecting structure of generating system

그림 5는 설계된 객체들을 상호 연결하여 하나의 발전시스템을 완성한 객체연결구조이다. 실제 프로그램상에서는 이들의 연결구조를 이진트리를 이용하여 구현한다. 그림 5의 구조를 전력계통 과도안정도 해석을 위하여 이미 설계된 발전기 객체가 이진트리 멤버데이터로 가지도록 만들어 주는 것은 OOP에서는 매우 쉬운 일이다. 그림 5에서 각 객체의 이름이 주어졌다. 그림 5는 표 1에 대한 설명과 같은 방법으로 디지털컴퓨터에서 연결트리로 구현된다. 이진트리 후위방문 알고리즘을 이용하면 트리의 말단으로부터 뿌리까지 순차적으로 방문하여 수치해석기법으로 1계미분방정식을 풀 수 있다. 이러한 방식의 가장 큰 장점은 시스템의 변화에 유연하고 모든 시스템을 모델링 할 수 있다는 것이다. 이것은 다양한 제어장치로 구성되는 발전시스템의 모델링에 매우 큰 유연성을 부여한다. 그림 5에서 $\frac{K_R s}{1+\tau_R s}$ 와 같은 전달함수 객체는 각 객체의 파라미터에 따라 그 내부에서 기본객체들의 연결 관계가 그림 6과 같이 자동적으로 성립되도록 구성 하였다. 이미 설명했듯이 모든 각각의 트리는 1계미분방정식을 표현한다.

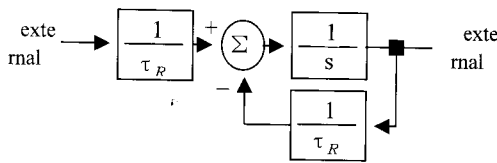


그림 6 전달함수 객체의 내부 구조
Fig. 6 The internal structure of Transfer Object.

최종적으로 발전시스템을 표현하는 객체는 그림 7의 machine 객체이다. 함수 maketree()는 표 1과 같은 객체연결정보로부터 그림 3과 같은 트리를 구성하는 함수이다. 함수 mergetree()는 maketree()에 의해 구성된 트리들이 외부 연결변수를 가지므로 이들을 상호 연결시켜 트리의 뿌리와 말단이 적분기가 되는 새로운 트리를 구성하여 이중포인터로 선언된 전용변수 mtree에 저장한다. 함수 searchtree()는 mtree에 대하여 이진트리 후위방문 알고리즘을 적용하여 계산 순서를 찾는다. mtree에 저장된 N 개의 트리는 하나의 발전시스템에 포함된 모든 1계미분방정식을 나타낸다. 함수 generation은 이 순서를 이용하여 각 트리에 대하여 매 시간격 Δt 마다 계산을 반복하여 미분방정식의 해를 구하게 된다.

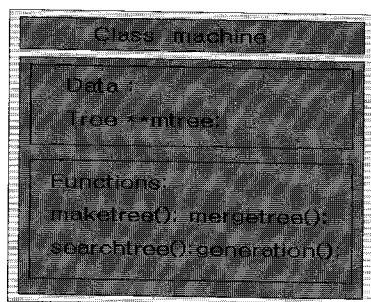


그림 7 Machine 객체
Fig. 7 Class Machine

2.2 선로 모델링

선로는 희소행렬 기법을 이용하여 어드미턴스 행렬 Ybus로 구성한다. Ybus 또한 모선을 연결하는 선로의 임피던스 데이터를 기본객체로 만들고 이들 객체를 연결리스트로 연결하여 구성한다.

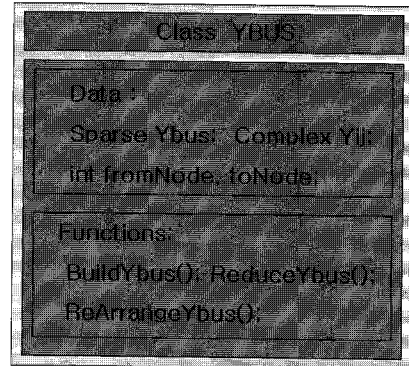


그림 8 YBUS 클래스
Fig. 8 Class YBUS

클래스 YBUS는 모선의 모든 부하를 일정 임피던스 부하로 가정하고 스파스 행렬을 사용하여 Ybus를 구성하고 (BuildYbus), 실제 시뮬레이션시는 발전기 모선으로 축약된다(ReduceYbus)[7]. 과도 안정도 해석을 위한 고장해석시는 Ybus가 재 정렬되도록 한다(ReArrangeYbus).

2.3 전체 시스템 모델링

발전기와 선로와의 상호 연결은 각 객체가 가지는 버퍼를 통한 데이터 교환에 의해 이루어진다. 각 객체는 자신의 작업의 결과를 버퍼에 저장하고 상대 객체에게 넘겨주거나 상대로부터 결과를 얻어 오기만 하면 된다. 즉 객체간의 연결에는 데이터의 교환만 있을 뿐 타 객체의 내부구조까지 알 필요가 없다. 그림 9는 다수기 계통에서 발전시스템과 선로의 상호관계에 대해 설명하고 있다. 그림에서 발전시스템 객체연결은 그림 7의 machine 객체를 의미한다. 각각의 machine 객체는 모든 데이터를 은닉하고 있고 YBUS객체와 메시지교환 즉 데이터 교환만을 수행한다.

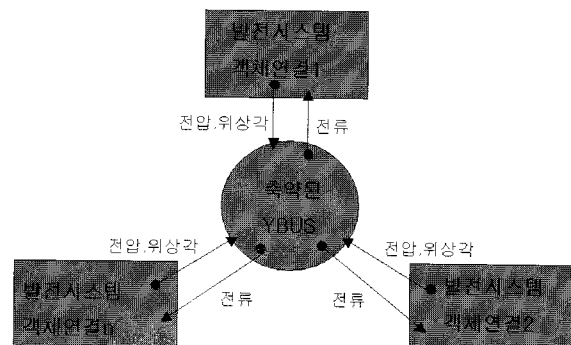


그림 9 다수기 시뮬레이션을 위한 객체연결
Fig. 9 Object connection for multimachine simulation

3. 에너지함수를 이용한 과도안정도 해석

에너지함수법은 과도안정도를 다루는 직접법으로 외란에 의해 시스템이 얻게 되는 과도에너지를 시스템의 임계에너지와 직접 비교하여 안정도를 판별하는 방법이다.

3.1 발전기 고전모델의 에너지함수

과도에너지함수(TEF: Transient Energy Function)는 관성중심 기준축에 대한 발전기 방정식으로부터 유도된다. 시스템 전체에 대한 관성 중심(Center of inertia)은 다음 식으로 정의된다.

$$\delta_0 = \frac{1}{M_T} \sum_{i=1}^N M_i \delta_i, \quad \omega_0 = \frac{1}{M_T} \sum_{i=1}^N M_i \omega_i \quad (3.1-1)$$

여기서 $M_T = \sum_{i=1}^N M_i$ 이고, M_i 는 i 번째 발전기의 관성정수이고, N 은 계통의 총발전기 개수이다.

발전기의 위상각 변수 δ , 각속도 변수 ω 를 관성중심의 변수로 변환하면

$$\theta_i = \delta_i - \delta_0, \quad \tilde{\omega}_i = \omega_i - \omega_0 \quad (3.1-2)$$

이제 관성중심의 발전기 동요방정식은 식(3.1-3)과 같다.

$$M_i \frac{d\tilde{\omega}_i}{dt} = P_i - \sum_{j=1, j \neq i}^N (C_{ij} \sin \theta_{ij} + D_{ij} \cos \theta_{ij}) - \frac{M_i}{M_T} P_{COA} = f_i(\theta) \quad (3.1-3)$$

$$\text{여기서 } \frac{d\theta_i}{dt} = \tilde{\omega}_i, i=1, N, P_i = P_{mi} - E_i^2 G_{ii}$$

, $C_{ij}, D_{ij} = E_i E_j B_{ij}, E_i E_j G_{ij}, P_{COA} = \sum_{i=1}^N P_i - P_{ci}, G_{ij}, B_{ij}$ 는 어드미턴스 행렬의 성분, E_i 는 i 번째 발전기의 내부유기전압이다.

과도에너지함수 V 는 식(3.1-3)을 적분하여 식(3.1-4)과 같이 얻을 수 있다.

$$\begin{aligned} V(\theta, \tilde{\omega}) &= \sum_{i=1}^N \frac{1}{2} M_i \tilde{\omega}_i^2 - \sum_{i=1}^N \left(\int_{\theta_i^*}^{\theta_i} f_i(\theta) d\theta \right) \\ &= \sum_{i=1}^N \frac{1}{2} M_i \tilde{\omega}_i^2 - \sum_{i=1}^N P_i (\theta_i - \theta_i^*) - \sum_{i=1}^{N-1} \sum_{j=i+1}^N [C_{ij} (\cos \theta_{ij} - \cos \theta_{ij}^*) \\ &\quad + \int_{\theta_i^* + \theta_j^*}^{\theta_i + \theta_j} D_{ij} \cos \theta_{ij} d(\theta_i + \theta_j)] = V_{KE}(\tilde{\omega}) + V_{PE}(\theta) \end{aligned} \quad (3.1-4)$$

식(3.1-4)을 다시 쓰면

$$V(\theta, \tilde{\omega}) = V_{KE}(\tilde{\omega}) + V_p(\theta) + V_d(\theta) \quad (3.1-5)$$

$$\text{여기서 } V_{KE}(\omega) = \frac{1}{2} \sum_{i=1}^N M_i \tilde{\omega}_i^2 \quad (3.1-6)$$

$$V_p(\theta) = - \sum_{i=1}^N P_i (\theta_i - \theta_i^*) - \sum_{i=1}^{N-1} \sum_{j=i+1}^N [C_{ij} (\cos \theta_{ij} - \cos \theta_{ij}^*)] \quad (3.1-7)$$

$$V_d(\theta) = - \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left[\int_{\theta_i^* + \theta_j^*}^{\theta_i + \theta_j} D_{ij} \cos \theta_{ij} d(\theta_i + \theta_j) \right] \quad (3.1-8)$$

$$V_{PE}(\theta) = V_p(\theta) + V_d(\theta) \quad (3.1-9)$$

이고, θ^* 는 사고후 시스템의 안정평형점이다.

식(3.1-4)의 TEF의 모든 항들은 안정 평형점을 기준으로 계산된다. 식(3.1-5)의 물리적 의미는 관성중심에 대한 발전기 회전자의 운동에너지의 총 변화를 나타낸다. 식(3.1-6)은 관성중심에 대한 발전기 회전자의 운동에너지의 변화를 나타낸다. 식(3.1-7)의 첫 항은 관성중심에 대한 회전자의 위치에너지의 총 변화를 나타내고, 두 번째 항은 브랜치 ij에 저장되는 자기에너지의 변화를 나타낸다. 식(3.1-8)의 적분은 경로 의존 적분이다. 이것의 적분은 사다리꼴 적분법을 이용하여 시간격 Δt 마다 수치적으로 구할 수 있다. 사다리꼴 적분을 이용하여 식(3.1-8)는 식 (3.1-10)와 같이 나타낼 수 있다.

$$V_d(\theta) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N I_{ij} \quad (3.1-10)$$

여기서 k 번째 스텝에서

$$\begin{aligned} I_{ij}(k) &= I_{ij}(k-1) + \frac{1}{2} D_{ij} [\cos(\theta_i(k) - \theta_j(k)) + \cos(\theta_i(k-1) \\ &\quad - \theta_j(k-1))] \times [\theta_i(k) + \theta_j(k) - \theta_i(k-1) - \theta_j(k-1)] \end{aligned} \quad (3.1-11)$$

3.2 발전기 상세모델의 에너지함수

시스템의 과도상태 동안 더 정확한 과도 에너지를 계산하기 위하여 발전기의 상세 모델링이 필요하고 여자시스템도 필요하다. 여기에 적합한 발전기 모델은 이축모델(two-axis model)이다. 여자시스템은 IEEE type1 모델을 사용한다. 이축 모델의 전기 및 운동방정식은 아래의 식들로 주어진다 [7].

$$\dot{E}_q' = \frac{1}{T_{d0}'} (E_{FD} - E) \quad (3.2-1)$$

$$\dot{E}_d' = \frac{1}{1+sT_{q0}'} (x_q - x_q') I_q \quad (3.2-2)$$

$$E = E_q' + (x_d' - x_d) I_d \quad (3.2-3)$$

$$T_e = E_d' I_d + E_q' I_q - (x_q' - x_d') I_d I_q \quad (3.2-4)$$

$$V_d = -r I_d - x_q' I_q + E_d' \quad (3.2-5)$$

$$V_q = -r I_q + x_d' I_d + E_q' \quad (3.2-6)$$

$$\dot{\omega} = \frac{1}{M} (T_M - D\omega - T_e) \quad (3.2-7)$$

$$\delta = \omega - 1 \quad (3.2-8)$$

여기서 E_q' 는 횡축 과도 유기 기전력, E_{FD} 는 여자전압, E 는 고정자 공극 실효치 전압, V_d 는 단자전압의 직축 성분, V_q 는 단자전압의 횡축 성분, I_d 는 단자전류의 직축 성분, I_q 는 단자전류의 횡축 성분, x_d 는 직축 리액턴스, x_q 는 횡축 리액턴스, x_d' 는 직축 과도 리액턴스, x_q' 는 횡축 과도 리액턴스, T_e 는 전기적 토크, T_{d0}' 는 직축 과도 개방회로 시상수, r 은 고정자 저항, T_{q0}' 는 횡축 과도 개방회로 시상수이다.

발전기 이축모델과 IEEE type1 여자시스템을 이용한 객체연결도는 그림 5에서 보였다. 선로에 대한 어드미턴스 행렬 Y_{BUS} 의 대각 요소를 $Y_{ii} \angle \theta_{ii} = G_{ii} + jB_{ii}$, 비대각 요소를

$Y_{ij} \angle \theta_{ij} = G_{ij} + jB_{ij}$ 라 하면 i 번째 발전기에서 선로로 주입하는 직축 및 횡축전류는 식 (3.2-9)로 주어진다.

$$I_{di} = \sum_{j=1}^N [H_{G-B}(\theta_{ij})E'_{qj} + H_{G+B}(\theta_{ij})E'_{dj}]$$

$$I_{qi} = \sum_{j=1}^N [H_{G+B}(\theta_{ij})E'_{dj} - H_{G-B}(\theta_{ij})E'_{qj}] \quad (3.2-9)$$

여기서 $H_{G+B}(\theta_{ij}) = G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}$ 이고,
 $H_{G-B}(\theta_{ij}) = B_{ij} \cos \theta_{ij} - G_{ij} \sin \theta_{ij}$ 이다.

과도에너지함수 V 는 다음의 적분으로부터 얻는다.

$$V(\theta, \tilde{\omega}) = \int_{\theta_i^*}^{\theta_i} \sum_{i=1}^N [M_i \frac{d\tilde{\omega}_i}{dt} - P_{mi} + P_{ei} - \frac{M_i}{M_T} P_{COA}] d\theta_i$$

$$= V_{KE}(\tilde{\omega}) + V_{PE}(\theta) \quad (3.2-10)$$

식(3.2-10)에서 $P_{ei} = E'_{di} I_{di} + E'_{qi} I_{qi}$ 이고 식(3.2-9)을 대입하여 정리하면

$$P_{ei} = K_{ii} G_{ii} + \sum_{j=1, j \neq i}^N [B_{ij} [K_{ij} \sin \theta_{ij} + L_{ij} \cos \theta_{ij}] + G_{ij} [K_{ij} \cos \theta_{ij} - L_{ij} \sin \theta_{ij}]] \quad (3.2-11)$$

여기서 $K_{ij} = E'_{di} E'_{dj} + E'_{qi} E'_{qj}$ 이고, $L_{ij} = E'_{di} E'_{qj} - E'_{qi} E'_{dj}$ 이다.

식 (3.2-10)에서 운동 에너지는 $V_{KE}(\omega) = \frac{1}{2} \sum_{i=1}^N M_i \tilde{\omega}_i^2$ 이고 $V_{PE}(\theta)$ 는 식(3.2-12)로 유도된다.

$$V_{PE}(\theta) = - \sum_{i=1}^N \int_{\theta_i^*}^{\theta_i} (P_{mi} - K_{ii} G_{ii}) d\theta_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \int_{\theta_j^*}^{\theta_j} B_{ij} [K_{ij} \sin \theta_{ij} + L_{ij} \cos \theta_{ij}] d\theta_{ij} + \sum_{i=1}^{N-1} \sum_{j=i+1}^N \int_{\theta_j^*}^{\theta_j} G_{ij} [K_{ij} \cos \theta_{ij} - L_{ij} \sin \theta_{ij}] d(\theta_i + \theta_j) \quad (3.2-12)$$

식(3.2-12)의 마지막 항은 경로의존 적분을 포함하므로 사다리꼴 적분법을 이용하여 수치해석적으로 적분값을 구한다. 즉 발전기 상세 모델에 대하여 식(3.2-12)의 마지막 항에 식(3.1-11)을 적용하면 된다. 식(3.1-12)처럼 상세모델 발전기의 TEF에서도 위치에너지를 구하는 함수는 경로적분을 포함하므로 일반적인 Lyapunov 함수[9]가 아니다. 하지만 주어진 특정 사고에 대하여 사고궤도를 따르는 V_{PE} 는 단일 값을 가진다. 그러므로 TEF는 다음과 같이 수정할 수 있다.

$$V(\theta, \tilde{\omega}) = V_{KE}(\tilde{\omega}) + V_{PE}(\theta) - V_{PE}(\theta^0) = V_{KE}(\tilde{\omega}) + \widehat{V}_{PE}(\theta) \quad (3.2-13)$$

$$\widehat{V}_{PE}(\theta) = V_{PE}(\theta) - V_{PE}(\theta^0) = - [\sum_{i=1}^N \int_{\theta_i^*}^{\theta_i} f_i(\theta) d\theta_i - \sum_{i=1}^N \int_{\theta_i^*}^{\theta_i^0} f_i(\theta) d\theta_i]$$

$$= - \sum_{i=1}^N \int_{\theta_i^0}^{\theta_i} f_i(\theta) d\theta_i \quad (3.2-14)$$

식(3.2-14)를 이용하면 사고후 안정 평형점 θ_i^* 을 계산할 필요가 없으므로 과도안정도 모의 속도를 높일 수 있다.

4. PEBS방법을 이용한 임계고장제거시간의 계산

본 논문에서는 임계고장제거시간을 계산하기 위해서 PEBS(Potential Energy Boundary Surface)방법[1]을 사용한다. 시스템의 특정 사고궤적에 대하여 PEBS는 시스템의 안정도 한계의 지역적 근사치를 기술한다. PEBS는 $\sum_{i=1}^N f_i(\theta)(\theta_i - \theta_i^*) = 0$ 을 만족하는 점들 즉 θ 값들의 집합이다. 또는 $f^T(\theta) \cdot (\theta - \theta^*) = 0$ 을 만족하는 벡터 θ 가 PEBS이다. 시스템에 사고가 발생하면 θ 는 안정평형점 θ^* 로부터 점차 멀어지고 $f^T(\theta) \cdot (\theta - \theta^*) < 0$ 이면 시스템은 안정한 영역에 존재하고, $f^T(\theta) \cdot (\theta - \theta^*) > 0$ 가 되면 시스템은 안정한 영역을 벗어나 불안정하게 된다. 벡터의 내적 $f^T(\theta) \cdot (\theta - \theta^*)$ 을 감시함으로써 시스템의 안정과 불안정을 판단할 수 있다. 임계고장제거시간을 구하기 위해서는 먼저 임계에너지 V_{cr} 을 계산해야 한다. 임계에너지 V_{cr} 은 시스템의 사고궤도를 과도에너지함수 V 의 위치에너지 부분이 최고가 될 때 까지 적분함으로써 얻는다. 즉 $V_{cr} = V_{PE}^{max}$ 이다. 임계고장제거 시간을 찾기 위해 본 논문에서 제안한 방법은 다음과 같다

step1: 과도 안정도에 필요한 발전기 데이터를 파일로부터 읽는다. 그림 5와 같이 기본객체들을 연결하여 하나의 발전시스템을 이진트리의 형태로 구성하고 객체를 초기화시킨다. 선로 데이터를 입력 파일로부터 읽어 모션 어드미턴스 행렬 Y_{BUS} 를 최소행렬기법을 이용하여 구성한다. 발전시스템과 Y_{BUS} 를 그림 9의 설명처럼 연결하여 전체시스템을 완성시킨다.

step2: 특정선로의 사고를 가정하고 사고에 의한 어드미턴스 변화를 Y_{BUS} 에 반영한다. step1에서 구성된 시스템에 수치적분을 적용하여 사고궤도를 계산한다.

step3: 수치적분의 모든 단계마다 수정된 에너지함수 식(3.2-13)과 식(3.2-14)를 계산한다. 또한 벡터내적 $f^T(\theta) \cdot (\theta - \theta^0)$ 도 계산한다. 수정된 위치에너지가 최대가 될 때 그 값을 $V_{cr} = \widehat{V}_{PE}^{max}$ 로 둔다. 이때가 PEBS를 통과하는 시점이다.

step4: 사고궤도로부터 $V(\theta, \tilde{\omega}) = V_{cr}$ 가 되는 시간을 찾는다. 이시간이 임계고장제거시간 t_{cr} 이다.

5. 적용사례

그림 10은 발전기 10대이고 모션수가 39인 New England Test System[10]이다. 이 시스템은 10번 발전기를 제외한 나머지 발전기는 조속기, 터빈 그리고 여자기를 가지고 있다. 모션부하는 일정 임피던스부하로 가정한다.

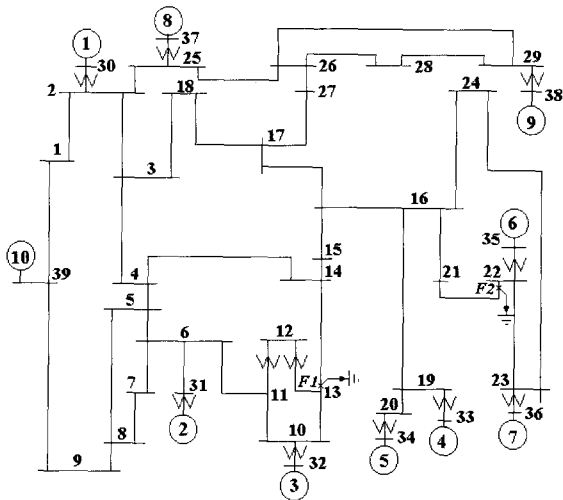


그림 10 New England 시험시스템의 단선도
Fig. 10 Single-Line Diagram for New England Test System

모선 4번 근처의 13-14 선로에 3상 단락사고를 가정하고 시뮬레이션을 하였다. 고장제거시간이 $t_c = 0.2$ 인 경우 그림 11에서 보인 것처럼 모든 발전기는 동기화를 유지한다. 이때 위상각은 관성중심을 기준으로 하여 구한 것이다. 고장제거시간이 $t_c = 0.208$ 인 경우 그림 12에서 가장 먼저 동기화를 잃게 되는 발전기는 3번 발전기이다. 같은 사고지점에 대하여 서로 다른 몇 개의 고장제거시간에 대한 3번 발전기의 응답을 그림 13에 보였다. 시간영역 모의를 이용하여 구한 임계고장시간은 0.206초이다. PEBS방법을 사용한 그림 14에서 수정된 위치에너지가 최대값을 갖는 시간은 0.407초이고, 4절의 step4의 방법으로 구한 임계고장제거시간은 0.206초이다. 이 경우는 두 개의 결과가 일치한다. 그림 14는 객체지향적으로 구현한 유연한 시간모의의 방법과 사고후 안정평형점과 불안정평형점의 계산이 필요 없는 방법과의 결합의 통하여 효과적으로 임계고장제거시간을 구할 수 있음을 보여주는 것이다.

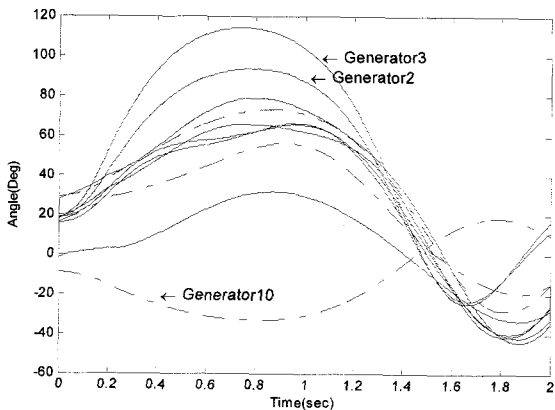


그림 11 사고지점 F1에 대한 발전기 위상각($t_c = 0.2$)
Fig. 11 Generator angles for fault at F1 with $t_c = 0.2$

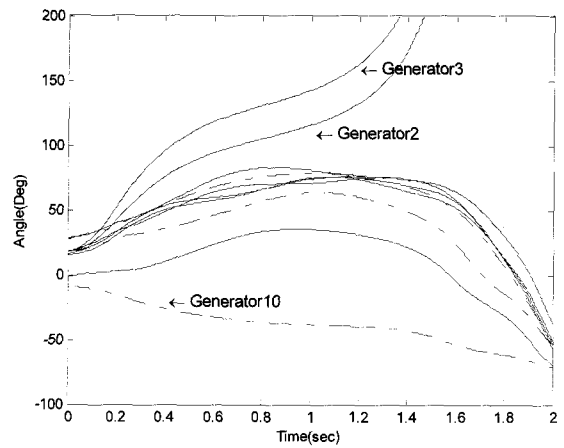


그림 12 사고지점 F1에 대한 발전기 위상각($t_c = 0.208$)
Fig. 12 Generator angles for fault at F1 with $t_c = 0.208$

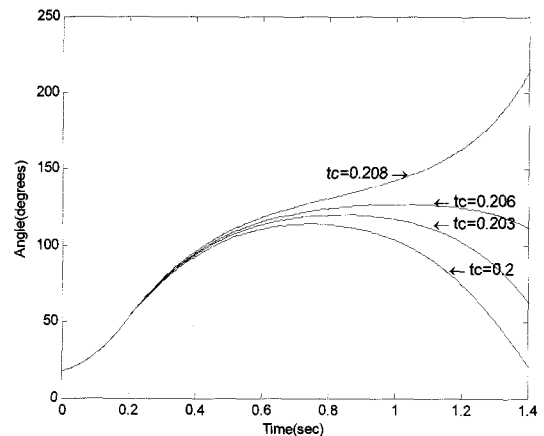


그림 13 3번발전기의 swing 곡선
Fig. 13 Swing curves of generator No. 3

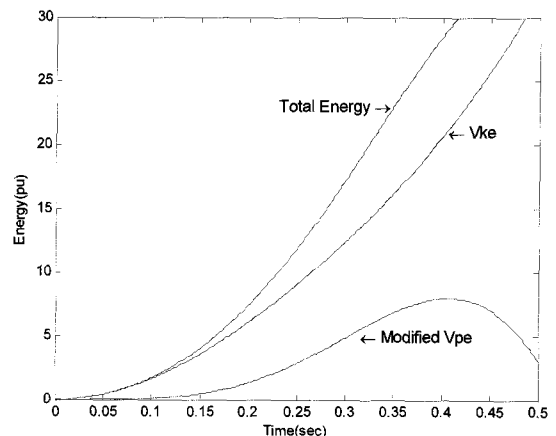


그림 14 사고지점 F1에 대한 총에너지, 운동에너지 그리고 위치에너지
Fig. 14 Total, kinetic and potential energy for fault at F1

모선 22번 근처의 22-21 선로에 3상 단락사고를 가정하고 시뮬레이션한 결과는 그림 15와 16이다. 가장 먼저 동기화를 잃게 되는 발전기는 6번 발전기이다. 수개의 서로 다른 고장제거시간(t_c)에 대하여 시간모의를 행하였다. 그림 15에서 시간영역모의를 통한 임계고장시간의 결과는 0.113초이다. PEBS방법을 사용한 그림 16에서 수정된 위치에너지가 최대값을 갖는 시간은 0.333초이고 4절의 step4의 방법으로 구한 임계고장제거시간은 0.107초이다. 이 경우 PEBS 방법의 오차는 0.004초이다. 모든 발전기를 그림 5와 같이 상세모델링하여 모의시간을 2초, 적분 시간격 $\Delta t = 0.001s$ 로 할 때 펜티엄4(1.8GHz 노트북)의 CPU 시간은 6초정도이다.

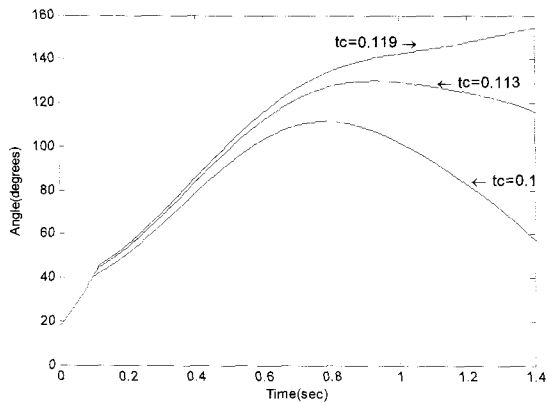


그림 15 6번발전기의 swing 곡선
Fig. 15 Swing curves of generator No. 6

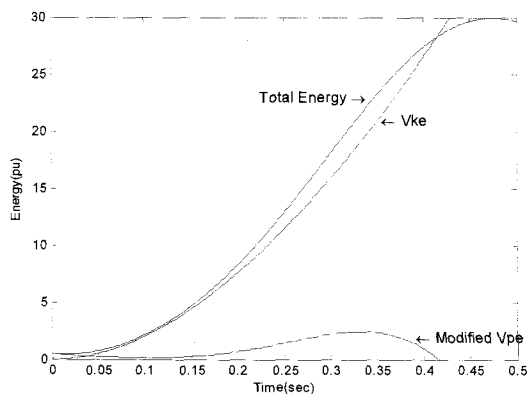


그림 16 사고지점 F2에 대한 총에너지, 운동에너지 그리고 위치에너지
Fig. 16 Total, kinetic and potential energy for fault at F2

6. 결론

본 논문은 과도안정도 해석에 시간영역모의방법의 장점과 직접법의 장점을 결합하여 적용하였다. 과도안정도 해석의 시간영역모의 방법의 장점을 극대화하기 위하여 객체지향기법을 적용하였다. 객체지향기법은 소프트웨어의 개발과 유지보수 측면에서 장점을 제공하지만 전력계통의 과도현상의

모의에 있어서 계산효율은 이 접근법의 잠재력을 평가하는 핵심적인 요소이다. 즉 동적인 시스템의 물리적 의미를 가지는 가장 기본적인 소자를 객체로 모델링하여 이들을 프로그램에서 연결리스트를 구현하여 계산효율을 높였다. 과도안정도 시간영역 모의에 있어서 소프트웨어의 개발의 어려움을 극복했다. 또한 이 방법을 안정도 직접법과 쉽게 결합하여 발전기 상세모델을 사용함으로써 직접법이 가지는 모델링제약의 단점을 극복하였다. PEBS방법을 적용할 때 사고후 안정평형점을 사용하지 않는 방법을 제안하였다. 이 방법을 시험시스템에 적용한 결과 계산된 임계 고장제거시간은 시간영역 모의결과와 아주 작은 오차만을 가졌다. 앞으로 실제계에 온라인으로 적용하기 위해서 계산시간을 개선시키는 연구를 진행할 것이다.

참고 문헌

- [1] Athay, T., R. Podmore and S. Virmani, "Practical method for direct analysis of transient stability," IEEE Trans. on Power Apparatus and Systems, vol. PAS-98, no2, March/April 1979, pp. 171-182
- [2] Scott Robert Ladd, C++ techniques & Applications, M&T Books, 1990.
- [3] A.F. Neyer, F.F. Wu, and K. Inhof, "Object-Oriented Programming for Flexible Software: Example of a Load Flow", IEEE Trans. on Power System, Vol. 5, NO. 3, pp. 689-696, Aug. 1990.
- [4] E. Z. Zhou, "Object-Oriented Programming, C++ and Power System Simulation", IEEE Trans. on Power System, Vol. 11, No.1, Feb. 1996.
- [5] S. Lu, E. Swidenbank and B. W. Hogg, "An Object-oriented Power Plant Adaptive Control System Design Tool," IEEE Trans. on Energy Conversion, Vol. 10, No. 3, September. 1995, pp. 600-605
- [6] B. Hakavik and A.T. Holen, "Power System Modelling and Sparse Matrix Operations Using Object-oriented Programming," IEEE Trans. on Power Systems, vol. 9, No. 2, May 1994.
- [7] P.M. Anderson and A.A. Fouad, Power System Control and Stability, Iowa State University Press, 1977. pp. 327-333, pp. 400-413
- [8] C. F. Gerald and P. O. Wheatley, Applied Numerical Analysis, Addison-Wesley Publishing Company 1984, 303-312.
- [9] Jacques L. Willems, Direct Methods for Transient Stability Studies in Power System Analysis, IEEE Trans. on Automatic Control, Vol. AC-16, no. 4, Aug. 1971
- [10] R. Podmore, A. Germond, Development of dynamic equivalents for transient stability studies, EPRI EL-456(RP-763), Final Report, April 1977.

저 자 소 개



박 지 호 (朴 志 皓)

1991년 2월 경북대학교 전기공학과 졸업.

1996년 8월 동대학원 졸업(석사). 2001년

2월 동대학원 졸업(박사). 현재 경북대학

교 전자전기컴퓨터공학부 BK21계약교수

Tel : (053) 950-7327

Fax : (053) 950-5505

E-mail : pjh@ee.knu.ac.kr