

객체지향기법을 이용한 전력계통 해석을 위한 그래픽 소프트웨어 개발

論 文

55A-10-3

The Development of Graphics Package for Power System Analysis using Object-Oriented Programming

金 在 顯[†]
(Jaehyeon Gim)

Abstract - This paper describes a simulation package for a power system using objected-oriented programming. The package includes four parts which are a power flow, a short circuit calculation, a transient simulation program, and an economic dispatch. The graphical user interface(GUI) is designed as a common platform which allows the user to create one-line diagrams of systems, specify components of power systems as well as simulation parameters, and view the output produced by the chosen application. The paper presents the data structure of the functional modules such as the draw module, power system data module, the power system simulation module, and the utility module using the object oriented programming. This package may be useful for educational and research purposes.

Key Words : Object-Oriented Programming, Power Flow, Short Circuit, Transient Stability, Economic Dispatch

1. 서 론

전력계통의 확장을 계획하고 효율적으로 운영하기 위하여 프로그램을 이용한 시뮬레이션은 필수적이다. 그동안 개발된 프로그램으로는 조류계산, 고장계산, 안정도 및 EMS (Energy Management Systems) 등이 있다. 이 프로그램들은 각 함수의 순서대로 하나씩 수행되는 일련의 프로시저들로 구성하는 절차적인 프로그램으로 프로그램의 유지 및 보수에 많은 비용이 소요된다. 그래서 데이터와 그 데이터를 대상으로 작동하는 프로시저를 단일 객체로 취급할 수 있는 객체지향 프로그램으로 변화되고 있다. 한편, 이 프로그램들은 대형 컴퓨터에서 실행되었던 것이 PC기반으로 변화되었고, DOS 환경에서 실행되던 것이 윈도우의 출현으로 점차로 윈도우 환경에서 실행되는 프로그램으로 변화되어 사용자가 쉽게 사용할 수 있는 프로그램으로 개발되고 있다.

전력계통은 기술의 발달로 새로운 장치들이 개발되어 전력계통에 적용됨에 따라 전력계통을 시뮬레이션하기 위한 프로그램도 이 장치들의 모델링과 알고리즘의 개발을 통하여 지속적으로 유지보수가 필요하게 되었다. 또한 전력산업의 구조조정과 민영화로 전력계통 해석을 쉽게 분석할 수 있는 소프트웨어가 더욱 필요하게 되었다. 따라서 객체지향 기법과 텍스트와 그래픽을 이용한 프로그램의 개발이 필요하게 되었다.

지금까지 개발한 객체지향 기법을 이용한 프로그램으로는

데이터베이스[1,2], GUI[3,4], 전력계통 시뮬레이션[5,6,7], 조류계산[8], 교육용 도구로 개발 등[9,10,11]이 있다.

본 논문은 PC환경의 윈도우 운영체제에서 GUI를 이용하여 전력계통의 단선도를 설계함으로써 전력계통 요소에 관련된 데이터는 단선도 화면에서 입력되고, 또한 그 결과를 객체에서 확인할 수 있는 객체지향 기법을 이용한 GUI와 과도안정도 프로그램을 개발하였다. 이 프로그램은 조류계산, 고장계산, 과도안정도 및 경제급전에 대한 모듈로 구성되며 이들 모듈들의 데이터의 구조는 객체지향기법을 이용하여 설계하였다. 전력계통 시뮬레이션을 위한 모듈들은 공통된 데이터를 사용하면서 독립적으로 실행이 가능하다. 9모선 시스템을 대상으로 모의 실험하였다.

2. 데이터 구조

개발된 프로그램은 C++언어를 사용하여 객체지향 프로그래밍(Object Oriented Programming)개념에 맞추어 설계되었다. C++와 같은 OOP언어는 클래스 정의, 연산자, 오버로딩, 상속, 캡슐화와 같은 차별 특성을 가지고 있다. 이 프로그램은 사용자에게 편리한 Graphic User Interface(GUI)를 제공하기 위하여 Microsoft Fundamental Classes(MFC)를 이용하였으며 Microsoft Visual C++를 이용하여 컴파일 되었다.

본 프로그램의 전체 데이터구조는 그림 1과 같다. 이 데이터 구조에는 화면처리를 담당하고 전력계통 데이터의 입출력을 담당하는 그래픽 드로우 모듈(Draw), 입력된 전력계통 데이터를 관리하는 모듈(Power Data), 전력계통의 모의실험을 담당하는 모듈(Power Simulation), 결과를 그래프로 나타내는 플롯 모듈(Plot), 및 파일로 저장된 결과를 텍스트로 분석할 수 있는 모듈(Text)로 구성된다.

[†] 교신저자, 正會員 : 順天大學 情報通信工學部 副教授 · 工博

E-mail : jhg@sunchon.ac.kr

接受日字 : 2006年 7月 19日

最終完了 : 2006年 9月 10日

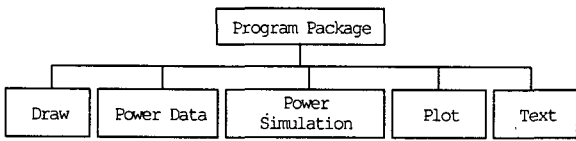


그림 1 프로그램의 데이터 구조
Fig. 1 The Data Structure of the Package

2.1 그래픽 드로우 모듈

그래픽 드로우 모듈은 화면에 전력계통의 단선도를 그리고 편집하는 기능을 담당하는 모듈이다. 이 모듈은 그림 2와 같이 화면에 전력계통의 요소들을 나타내는 요소클래스와 그림 3과 같이 사용자의 이벤트를 받아 각 객체를 조작하는 도구클래스로 구성되었다. 도구클래스는 화면에서 발생한 이벤트를 전달하여 요소클래스에서 그 이벤트를 실행한다. 그림에서 화살표는 상위 클래스를 상속 받은 클래스이고 화살표가 없는 클래스는 상위클래스에 하위클래스를 멤버 변수로 가지고 있는 것이다.

요소클래스는 전력계통의 기본적인 요소인 모선, 발전기, 부하, 선로, 변압기 및 병렬조상기의 6개 하부클래스를 가지고 있다. 이 각각의 요소클래스(DrawBus, DrawGen 등)는 DrawObj클래스를 상속받고, DrawObj는 MFC의 CObject클래스에서 파생된다. DrawObj는 추상 클래스로서 전력계통요소들의 화면의 위치, 선의 굵기, 색상 등의 기본적인 변수와 멤버 함수를 가지고 있다. 요소클래스는 전력계통의 각 요소들을 나타내며 전력계통을 해석하기 위해 필요한 데이터를 멤버 변수로 가지고 있다. 예를 들면 DrawBus클래스는 PowerBus클래스를 가지고 있어 데이터의 이동에 소요되는 시간을 단축할 수 있다.

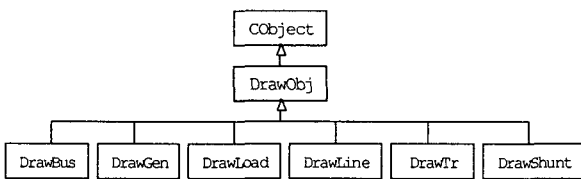


그림 2 드로우 모듈의 데이터 구조
Fig. 2 The Data Structure of Draw Module

화면에 나타난 전력계통의 단선도를 편집할 때 사용자의 이벤트를 받아 각 객체를 조작하는 도구클래스에는 전력계통의 기본적인 요소인 모선, 발전기, 부하, 선로, 변압기 및 병렬조상기의 하부클래스 뿐만 아니라 SelectTool클래스와 LinkTool클래스가 있다. 상위 클래스인 DrawTool은 화면에서 발생하는 이벤트에 관련된 정보를 가지고 처리한다. SelectTool클래스는 여러 객체를 선택하여 이동, 복사, 삭제 등의 기능을 하며, LinkTool클래스는 모선에 연결되는 발전기, 부하, 선로 및 변압기들의 연결을 나타내는 클래스이다. LinkTool클래스는 화면에 나타나는 전력계통의 단선도에서 연결 상태를 나타낼 뿐만 아니라 전력계통 데이터클래스들도 단선도에서 연결된 것과 같이 각각의 클래스를 서로 연결시

킨다. 그리고 각 객체의 전력계통해석에 필요한 데이터클래스는 속성 이벤트를 이용하여 입력하고, 해석결과를 확인할 수 있다.

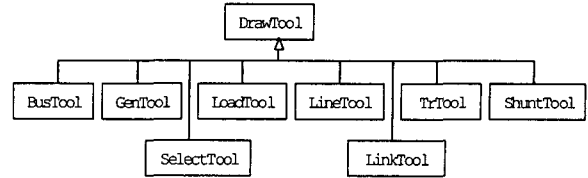


그림 3 화면처리 도구의 데이터 구조
Fig. 3 The Data Structure of Display Tool

2.2 전력계통 데이터 모듈

전력계통 데이터 모듈은 그림 4와 같은 데이터클래스 구조를 가지고 있다. 데이터클래스는 전력계통에 필요한 요소들을 멤버 변수로 가지고 있으며 계통 편집 시 요소들의 추가 삭제를 용이하게 하기 위하여 Double Linked List의 데이터 구조[19]를 사용하였다.

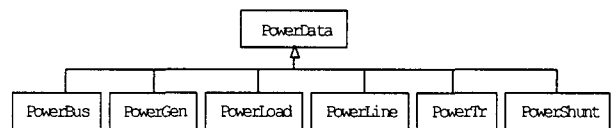


그림 4 전력계통데이터의 데이터 구조
Fig. 4 The Data Structure of Power Data

실제 전력계통의 모선과 같이 PowerBus클래스는 발전기, 부하, 및 선로 등이 그림 5와 같이 연결되어 데이터를 공유한다. 이에 대한 상세한 방법은 참고문헌[16]에 있다. PowerGen, PowerLoad, PowerBranch클래스는 연결된 모선의 PowerBus클래스를 멤버 변수로 선언되어 있어 상호간 데이터를 쉽게 접근할 수 있다. 또한 이 클래스들은 전력계통 해석에 필요한 멤버 함수를 가지고 있다.

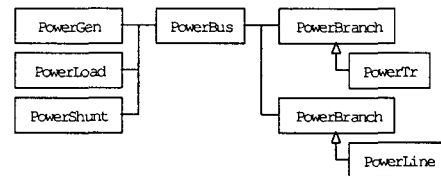


그림 5 모선 연결 관계
Fig. 5 Bus Connection

그림 6과 같은 PowerGen클래스는 발전기의 동적인 특성을 나타내기 위하여 발전기모델을 나타내는 GenModel클래스를 멤버 변수로 가지고 있다. 이 GenModel클래스는 다양한 발전기 모델 클래스를 파생시키고, 여자기와 조속기의 모델을 위한 추상 클래스(AVRModel, GOVModel)를 멤버 변수로 가지고 있어 다양한 모델[12,13,14]을 나타내는 파생클래스를 생성할 수 있다.

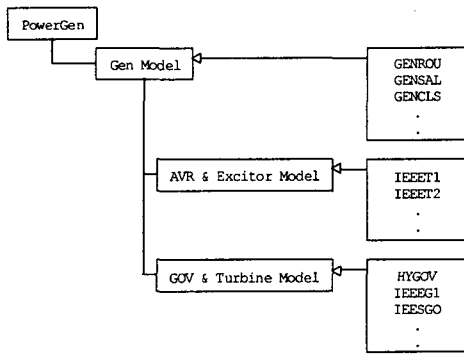


그림 6 발전기 클래스
Fig. 6 Generator Class

2.3 전력계통 시뮬레이션 모듈

개발한 프로그램의 응용 모듈로는 그림 7과 같이 조류계산 클래스(Power Flow), 고장계산 클래스(Short Circuit), 경제급전(Economic Dispatch), 과도안정도 클래스(Transient Stability), JacobianMatrix클래스, YbusMatrix와 ZbusMatrix 클래스가 있다.

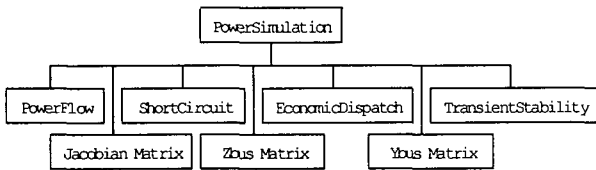


그림 7 시뮬레이션 모듈의 데이터구조
Fig. 7 The Data Structure of Power Simulation

JacobianMatrix클래스는 전력계통의 Jacobian행렬을 계산하는 클래스이다. 이 클래스의 Jacobian행렬 변수는 유틸리티 모듈의 SparseMatrix클래스로 선언하고, 모선 전력Mismatch행렬과 모선 전압 변위 행렬은 Vector클래스로 선언한다. 전력계통의 모선클래스(PowerBus)와 가지 클래스(Power-Line과 PowerTr)에 Jacobian행렬 요소들의 포인터를 멤버 변수로 가지고 있어 모선클래스와 가지클래스에서 행렬의 요소 값을 계산함으로써 데이터를 공유하여 이동에 필요한 소요시간을 단축할 수 있다. YbusMatrix와 ZbusMatrix클래스도 JacobianMatrix클래스와 같은 방법으로 멤버변수 어드미턴스와 임피던스행렬을 SparseMatrix클래스로 선언하고 이들 각 요소는 모선클래스와 가지 클래스에 포인터가 멤버 변수로 선언되어 그 값을 계산한다.

조류계산을 위한 클래스는 PowerFlow클래스이다. 조류계산 기법에는 다양한 방법이 있으나 본 프로그램에서는 Newton-Raphson법을 사용하였다. PowerFlow클래스에는 JacobianMatrix클래스를 멤버 변수로 선언하고, 이 변수에서 Jacobian행렬과 Mismatch행렬이 계산될 뿐만 아니라 전압의 변위 값도 계산된다. 새로운 전압의 갱신은 PowerBus클래스에서 멤버변수로 가지고 있는 전압변위 Vector의 요소를 이용하여 수정하게 되어 행렬의 요소를 찾아가는데 소요되는

시간을 단축할 수 있다. 이를 반복하여 조류계산을 수행하며 수렴판정은 전력편차나 모선 전압편차에 의하여 결정한다. 그리고 수렴판정 방법, 최대 반복횟수, 수렴판정 기준은 조류계산 대화창(그림 11)을 이용하여 사용자가 설정한다.

고장계산을 위한 클래스는 ShortCircuit클래스이다. 이 클래스는 3상 단락, 1선 단락, 선간 단락 및 2선 지락을 모의 실험할 수 있는 멤버 함수를 가지고 각각의 고장전류를 계산한다. 이 클래스에서 사용한 고장계산방법은 불평형 고장을 위한 대칭 좌표법과 임피던스 행렬을 이용하였다. 전력계통의 정상과 역상 데이터는 조류계산에서 입력한 데이터를 사용하고 영상에 대한 데이터는 사용자가 전력계통의 각 객체의 속성 창에서 입력한다. 역상데이터는 정상데이터와 같은 값이 아닐 때는 수정 할 수 있으며 특히 변압기는 결선방식과 중성점 임피던스 값을 입력하여야 한다. 발전기의 대칭 성분 데이터도 같은 방법으로 입력한다.

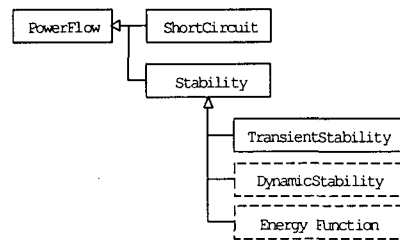


그림 8 시뮬레이션 클래스의 데이터 구조
Fig. 8 The Data Structure of Simulation Class

과도 안정도계산을 위한 클래스는 TransientStability클래스이다. 그림 8과 같이 TransientStability클래스는 Stability클래스에서 파생되었고 Stability클래스는 PowerFlow클래스에서 파생되었다. 이는 전력계통 안정도 계산을 위한 초기조건으로 조류계산의 결과를 이용하기 때문이다. 과도 안정도의 시간영역에서 시뮬레이션은 전력계통의 동특성을 나타내는 비선형 연립 미분방정식과 계통 망을 나타내는 비선형 연립 대수방정식의 해를 구함으로써 발전기의 회전자각, 모선 전압 등의 시간에 따른 변동으로 안정 여부를 판정하게 된다. 미분방정식의 계산 방법은 4차 Runge-Kutta법을 이용하며 이를 위한 클래스는 Kutta클래스이다. 대수방정식의 계산은 어드미턴스 행렬로 해를 구하였다. 과도 안정도 시뮬레이션을 위한 외란 발생은 별도의 대화창을 이용하여 시뮬레이션 절차를 입력한다. 여기서는 다양한 외란(모선사고, 선로사고, 발전기 탈락, 부하 탈락 등)종류, 사고발생 시간, 총 시뮬레이션 시간 등을 Sequence클래스에 입력하여 TransientStability클래스의 멤버변수로 선언하고 있다.

EconomicDispatch클래스는 전력계통의 경제적 운용을 목적으로 발전기의 발전비용을 최소화 시키는 발전력 배분을 위한 경제급전계산을 하는 클래스이다. 이 클래스는 송전손실을 무시할 경우와 고려한 경우를 사용자가 선택하여 발전기의 발전력 배분을 구할 수 있다. 본 논문에서 발전기 비용함수는 2차 방정식으로 가정하였다. 사용한 경제급전 계산법은 Lagrange의 미정계수법을 사용하였으며 송전 손실을 고려할 경우의 손실 방정식의 계수의 계산은 참고문헌[15]에 사용한 방법을 사용하였다.

2.4 유틸리티 모듈

유틸리티 모듈에는 LinkedList클래스, Complex클래스, Vector클래스, SparseMatrix클래스, Ordering클래스, 및 Kutta클래스로 구성된다. Complex클래스는 복소수 계산을 위한 클래스로서 크기와 위상 또는 실수와 허수 값 등을 계산하는 멤버함수와 계산 연산자 함수를 가지고 있다.

LinkedList클래스는 알맞은 크기로 설계된 여러 개의 작은 컨테이너들이 필요에 따라 링크되도록 구성된다. 이 컨테이너는 전력계통의 각종 데이터의 추가와 삭제가 용이하고 배열의 크기는 고정되어 있지 않아 컴퓨터의 메모리 한계까지 크게 링크시킬 수 있다. 실행 시 전력계통의 각종 요소들은 LinkedList클래스를 이용한 각종 전력계통 요소들을 PowerData클래스에 멤버 변수로 가지고 있다. 그러므로 전력계통의 크기에 관계없이 해석할 수 있다.

Vector클래스는 열 행렬과 행 행렬을 나타낸다. 이 행렬은 Linked List구조로 되어 있어 영(0)인 값을 가진 행렬의 요소는 노드가 없고 값을 가지고 있는 요소의 노드만으로 구성된다. SparseMatrix클래스는 희소행렬을 나타내며 단일 Linked List로 구성된다. 행렬의 덧셈, 뺄셈, 곱셈 및 역행렬을 계산할 수 있는 연산자를 가지고 있어 행렬을 이용한 대수함수 계산은 일반 수식과 같은 표현으로 계산할 수 있다. 또한 Vector클래스와 SparseMatrix클래스 사이에서도 연산이 가능하다. 역행렬 계산은 L/U분해방법으로 계산시 행렬의 인덱스가 동적 연결리스트로 연결된 순서대로 이동할 수 있게 알고리즘이 되어 있어 효율적으로 행렬의 계산을 수행할 수 있다. 상세한 내용은 참고문헌[16]에 있다.

Ordering클래스는 희소행렬의 계산 수행 중에 값을 갖는 행렬 요소의 수를 최소화하기 위한 순서를 결정하는 클래스로서 PowerBranch클래스를 이용하여 PowerBus클래스의 순서를 재배열한다. 본 논문에서 사용한 Ordering기법은 순서를 정하는 각 단계에서 남아 있는 모선에 연결된 가지가 최소가 되는 모선 순으로 정하는 Tinney-II방법을 사용하였다.[17,18]

Kutta클래스는 미분방정식의 계산을 위한 클래스로서 한 개의 상태변수에 3개의 멤버변수를 가지고 있다. 4차 Runge-Kutta법을 이용하므로 4단계를 거쳐 해를 구한다. 그러므로 전력계통의 모델의 모든 상태 변수는 Kutta클래스로 선언한다.

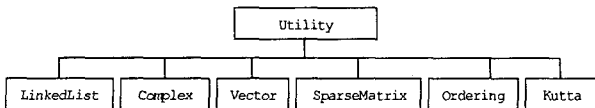


그림 9 유틸리티 클래스
Fig. 9 Utility Classes

2.5 기타 모듈

본 프로그램에 추가된 기능을 위한 클래스로는 Plot클래스와 Text클래스가 있다. Plot클래스는 과도 안정도 프로그램에서 시뮬레이션 결과를 그래프로 분석하기 위한 클래스로서 모선, 발전기 및 선로의 각종 결과 데이터를 이용하여 그래

프를 그릴 수 있다. Text클래스는 시뮬레이션 결과를 텍스트로 확인하기 위한 텍스트 편집기 기능을 한다.

3. 사례 연구

3.1 단선도 구성

단선도 구성과정은 다음과 같다. 주 화면의 "File"메뉴에서 "New"를 마우스로 클릭하면 Draw Power System 또는 Text File을 선택할 수 있는 대화창이 나타난다. 단선도를 그리기 위한 Draw Power System를, 텍스트 파일을 열기 위하여 Text File을 선택한다. Draw Power System을 선택 후 "OK"를 클릭하면 단선도를 그리기 위한 도면모드로 변환되면서 새로운 창이 생성된다. 메뉴나 도구바에서 우선 모선을 선택하고 원하는 위치에 클릭하면 모선이 생성되고 더블 클릭하면 모선 선택이 해제되면서 화살표가 선택된다. 원하는 위치에 원하는 수만큼 모선을 생성한 후 모선에 연결되는 발전기, 부하와 같은 전력계통의 요소들을 선택한 후 연결하고자 하는 모선에 마우스를 이동하여 클릭하면 각각의 요소들이 생성된다. 특히 선로는 모선과 모선 사이에 연결됨으로

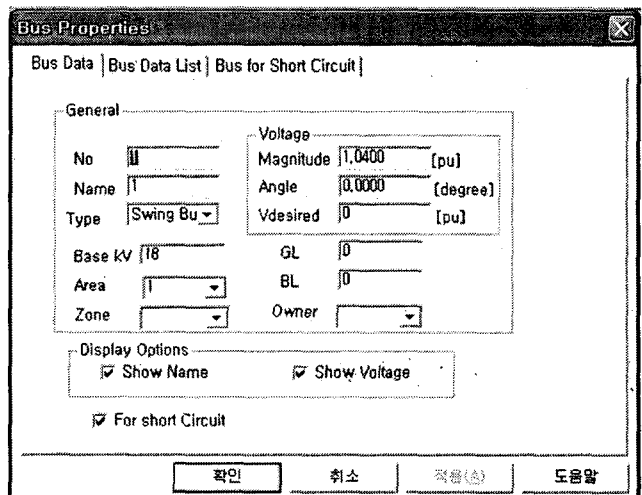


그림 10 모선 데이터 대화창
Fig. 10 Dialog Window for Bus Data

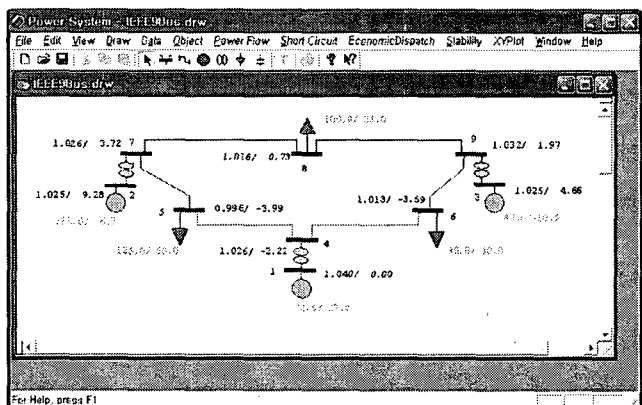


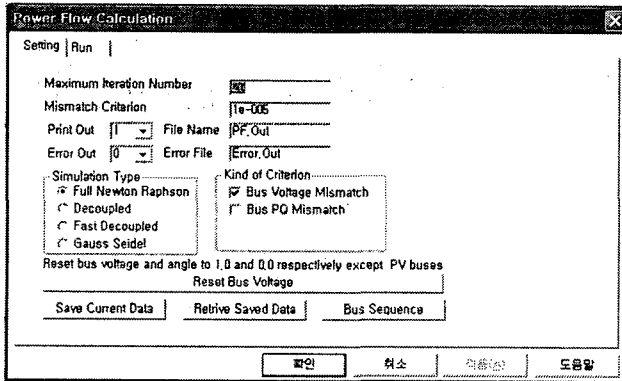
그림 11 단선도와 조류계산 결과
Fig. 11 One-line Diagram and Power Flow Result

한 모선에 클릭 후 다른 모선에 클릭하면 자동적으로 생성된다. 각 객체의 데이터 입력을 위하여 그 객체를 더블 클릭하면 데이터 입력을 위한 대화창이 생성되어 여기서 데이터를 입력하면 된다. 그림 10은 모선을 위한 대화창으로 모선번호, 이름, 모선형태 등을 입력한다.

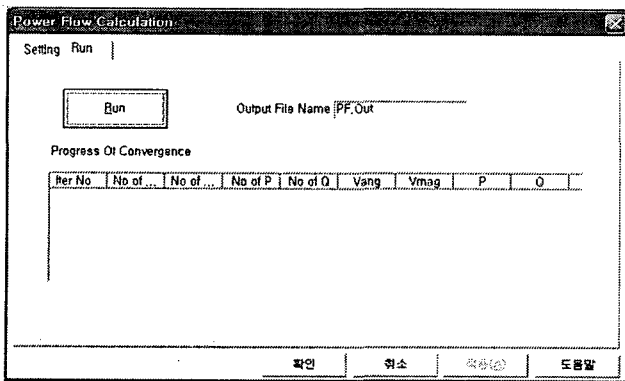
본 연구에서 개발한 전력계통 해석용 프로그램의 특성과 기능들을 확인하기 위하여 IEEE 9모선 계통을 선정하여 사례연구를 수행하였다. 이 전력시스템은 3대의 발전기, 3개의 부하, 9개의 모선으로 그림 11과 같이 구성되어 있다.

3.2 조류계산

메뉴에서 "Power Flow"를 클릭하면 그림 12와 같은 대화창이 생성된다. 이 창은 탭 형태로 "Setting"과 "Run"이 있는데 그림 12(a)와 같이 "Setting" 탭은 조류계산에 필요한 최대반복횟수, 미스매치 기준 값 등의 데이터를 입력하고 그림 12(b)와 같이 "Run" 탭은 "Run" 버튼으로 실행하면 수렴되는 과정이 리스트 박스에 기록된다.



(a)



(b)

그림 12 조류계산 대화창
Fig. 12 Dialog Window for Power Flow

그림 11은 1번 모선을 Swing모선으로 하고, 총 부하는 유효전력이 315[MW] 무효전력 115[MVAR]일 때 조류계산을 수행한 결과를 화면의 단선도에 나타낸 것이다. 각 모선의 전압은 전압의 크기와 위상으로 화면에 나타나고 선로와 변압기선로는 화면에 보이지 않지만 선로와 변압기의 조류도 나타낼 수 있고 선로의 색상으로 선로의 부하 상태를 나타내고 있다. 모든 결과를 더욱 상세히 검토하기 위해서는 각 객

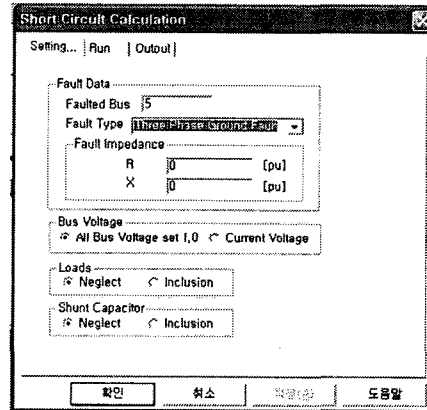
체의 속성 창에서 그 결과를 확인하면 된다.

그림 13은 조류계산 결과를 Text클래스의 텍스트 편집기를 이용하여 다른 프로그램에서와 같은 형태로 확인 할 수도 있다. 이 시스템의 모든 모선의 전압은 0.95~1.05사이를 유지하고 있으며 선로5-7의 용량이 80[MVA]인데 실제 흐르는 전력은 86.6[MW]/-8.38[MVAR]이므로 과부하가 되어 선로의 색이 붉은 색으로 나타나고 있다.

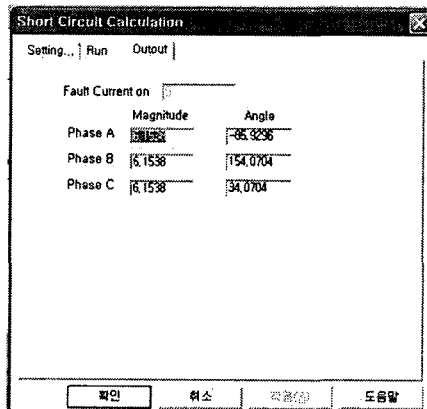
From Bus Name	To Bus Name	Line Flow MW	Line Flow MVAR	Tap Ratio	Angle	Rating X MVA
1	4	71.6	27.0	1.0500M	0.0	1.0
2	4	71.6	27.0	1.0500M	0.0	1.0
3	7	163.0	5.7	1.0500M	0.0	1.0
4	5	86.6	-8.38	1.0500M	0.0	1.0
5	6	39.7	1.0	1.0500M	0.0	1.0
6	4	-24.1	-24.9	1.0500M	0.0	1.0
7	8	76.4	-2.8	1.0500M	0.0	1.0
8	5	56.6	-3.4	1.0500M	0.0	1.0
9	5	-59.5	-12.5	1.0500M	0.0	1.0
9	4	-20.5	-16.5	1.0500M	0.0	1.0
8	2	-169.0	7.2	1.0500M	0.0	1.0
7	2	-15.9	-10.7	1.0500M	0.0	1.0
6	2	-24.1	-24.9	1.0500M	0.0	1.0
5	2	15.0	15.0	1.0500M	0.0	1.0

그림 13 조류계산 결과
Fig. 13 Power Flow Results

3.3 고장 계산



(a)



(b)

그림 14 고장계산 대화창
Fig. 14 Dialog Window for Short Circuit

메뉴에서 "Short Circuit"를 클릭하면 그림 14와 같은 대화 창이 생성된다. 그림 14(a)와 같이 "Setting" 창은 고장 계산에 필요한 고장위치, 고장종류, 고장 임피던스 등을 선택 후, "Run" 창에서 실행하고 그림 14(b)와 같이 "Output" 창에서 그 결과를 확인한다.

"Output" 창에서 모선5에 3상 단락고장이 발생하였을 때 고장 점에서의 각 상의 고장전류이다. 고장 점(모선5)의 고장 전류의 크기는 6.1538[p.u.]이며, 평형고장이므로 위상은 평형을 유지하고 있다. 그림15은 같은 고장발생 시 모선4번의 전압으로 크기는 0.3114[p.u.]이고 위상은 평형을 유지하고 있다.

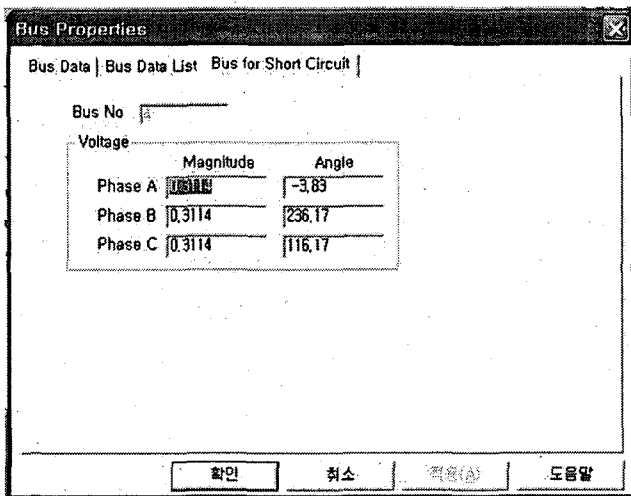


그림 15 모선4의 상 전압
Fig. 15 Bus 4 Phase Voltage

3.3 과도안정도 계산

메인 메뉴의 "Stability"에서 "Transient Stability"를 클릭하면 그림 16과 같은 과도안정도 실행을 위한 대화 창이 나타난다. 과도 안정도의 시뮬레이션을 위하여 실행순서를 위한 데이터 입력 탭을 선택하면 그림 16과 같다. 첫 번째 그룹박스(Read the Sequence Data From File)는 실행순서 데이터가 파일로 저장하거나 파일로부터 불러 오는 박스이고, 저장파일명은 STAB.SEQ로 되어 있는데 변경할 수 있다. Generator Angle Reference는 기준 위상각 결정 방법을 선택하는 콤보박스로 COD(Center Of Degree)를 기본 값으로 되어있다. 두 번째 그룹박스(Sequence Data)는 고장종류, 고장점, 고장 지속시간, 적분 간격시간, 고장 임피던스 등을 입력하여 "Add" 버튼을 클릭하면 아래 리스트 박스에 입력되고 리스트박스에 있는 데이터를 삭제하려면 리스트를 선택하고 "Delete" 버튼을 클릭하면 된다. 그림 16의 리스트 박스에 입력된 데이터는 초기 0.5초 동안 정상상태를 유지하고, 0.5초에서 선로 5-4의 모선5에 3상 단락고장을 발생시켜 0.1초 동안 지속하고 고장 선로의 고장을 제거 후부터 4초 동안 시뮬레이션을 진행하는 절차를 나타내고 있다. 적분간격시간은 0.005초이고, 고장임피던스는 영(0)이다. 그림 16에서 "Out" 탭에서 시뮬레이션 결과를 확인하고자하는 모선, 발전기 및 선로를 선정하고 "Run" 탭으로 이동하여 실행한다.

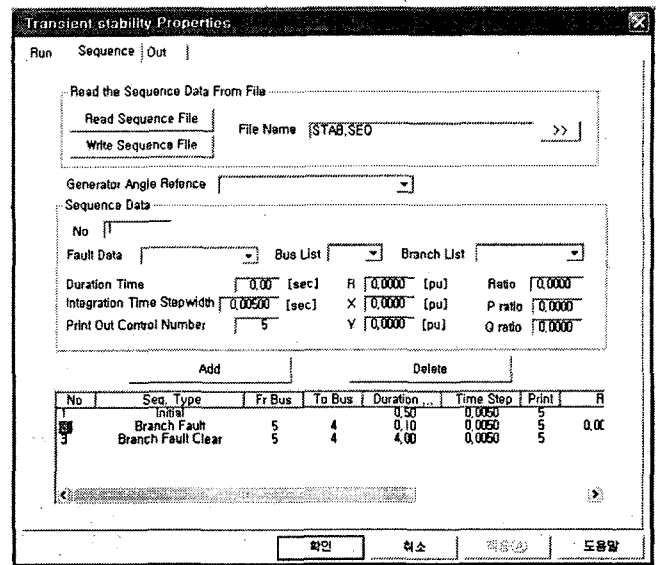


그림 16 시뮬레이션 순서 데이터 대화창
Fig. 16 Sequence Data Dialog Window

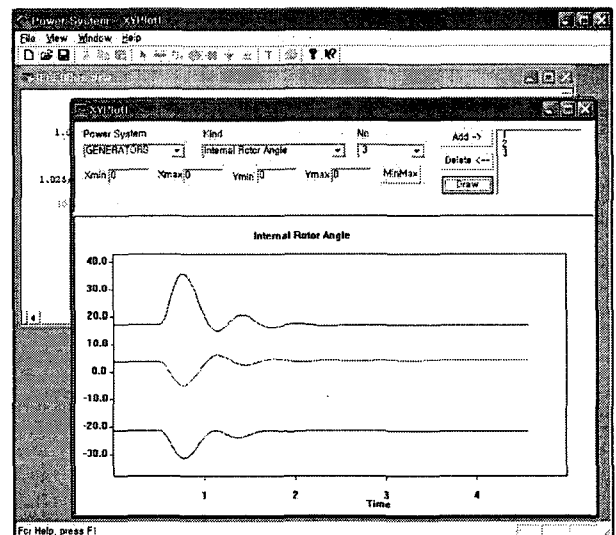


그림 17 발전기 회전자 위상각
Fig. 17 Internal Rotor Angle

메인 메뉴에서 XYPlot를 클릭하면 그림 17과 같은 창이 생성된다. 이 창은 Plot클래스를 이용하여 그림 16의 "Output"에서 선정된 모선, 발전기, 선로들 중 결과를 확인하고자 하는 항목을 선택한 후 "Draw" 버튼을 클릭하면 그래프가 그려진다. 그림 17은 발전기의 내부 위상각의 시뮬레이션 결과를 그래프로 그린 것으로 초기 발전기의 위상은 1번 발전기가 -21.2도 2번 발전기가 17.3도 3번 발전기는 3.9도이다. 0.5초에 고장 발생 후 2번 발전기의 위상각은 35.7도까지 상승하고서 진동이 감쇄되고, 1번과 3번 발전기는 위상이 반대방향으로 진동하고 있다. 그림 18은 모선의 전압특성을 나타낸 것으로 고장 중에 고장점인 모선5의 전압의 크기는 영(0)이고 다른 모선의 전압도 떨어진다. 고장 제거후의 모선 전압은 서서히 회복됨을 보이고 있어 이 전력 시스템이 안정함을 보이고 있다.

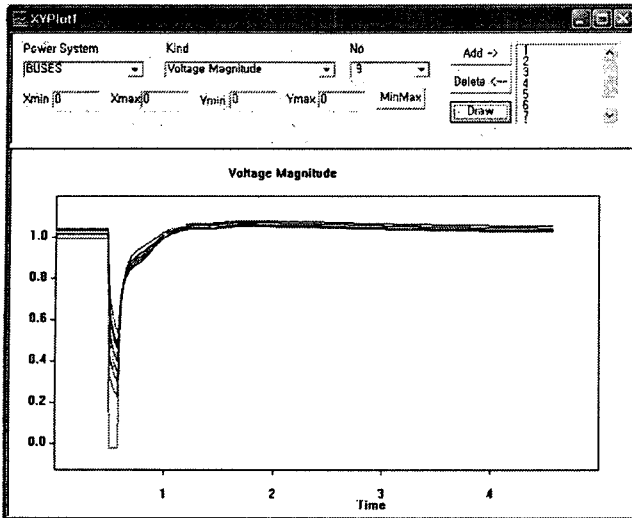


그림 18 모선 전압
Fig. 18 Bus Voltage

3.4 경제급전

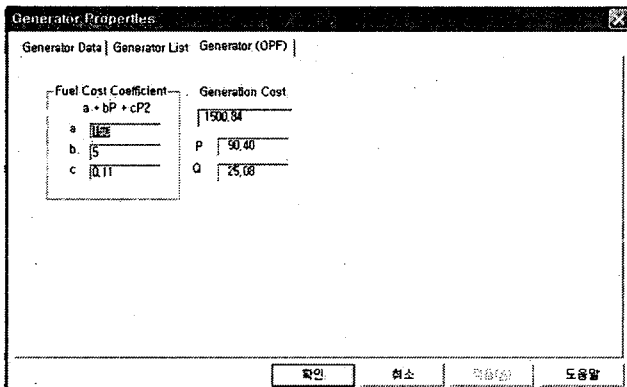


그림 19 경제급전 결과 대화창
Fig. 19 The Result Dialog for Economic Dispatch

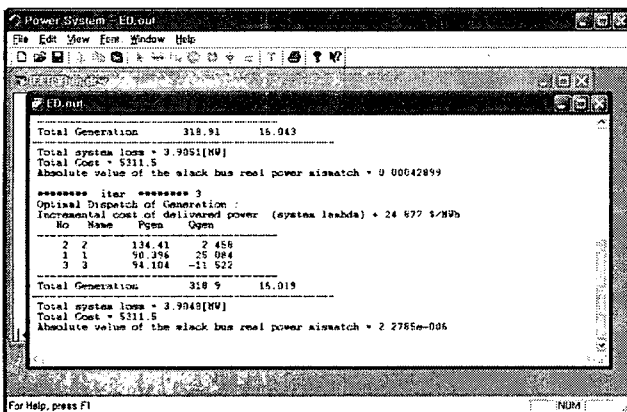


그림 20 경제급전 대화창
Fig. 20 The Result of Economic Dispatch

경제급전 계산을 위한 데이터는 발전기의 발전비용 함수의 계수를 입력하는 것이다. 그림 19는 1번 발전기의 계수

입력을 나타낸 것으로 비용함수 $150 + 5P + 0.11P^2$ 이다. 여기서 P 는 발전기의 유효전력이다. 2번 발전기는 $600 + 1.2P + 0.085P^2$ 이고, 3번 발전기는 $335 + P + 0.1225P^2$ 이다. 이 경우의 경제급전을 하였을 때의 발전량은 그림 20과 같이 1번 발전기는 90.4[MW]/25.084[MVAR], 2번 발전기는 134.4[MW]/2.458[MVAR], 3번 발전기는 94.1[MW]/-11.522[MVAR]이다. 각 발전기의 발전비용은 그림 19의 대화창에서도 알 수 있다. (1번 발전기의 발전비용은 1500.84이다.)

4. 결 론

본 논문은 전력계통의 시뮬레이션을 통하여 분석을 할 수 있는 객체지향기법을 이용한 윈도우 그래픽 프로그램을 위한 클래스를 개발하여 실제 프로그램에 적용하였다. 이 프로그램은 객체지향 기법의 장점인 실제 전력계통 모형을 유지하고, 개발의 편리성과 유지보수의 용이성을 확보하고 있다. 또한 그래픽 편집기를 이용하여 전력계통의 단선도를 작성하고 전력계통 해석에 필요한 데이터를 입력하여 그 결과를 확인 할 수 있다. 이 프로그램은 객체지향적인 최소행렬기법과 연결리스트를 사용하여 전력 계통의 크기에 제한이 없이 사용할 수 있으며, 경제급전 및 안정도 분석을 위한 연구 및 교육용으로 활용 할 수 있다.

참 고 문 헌

- [1] D. J. Flinn, R.C. Dugan, " A Database for Diverse Power System Simulation Applications", IEEE Trans. on PS, Vol. 7, No. 2, May 1992, pp784-790
- [2] G. S. Martire, D. J. H. Nuttall, " Open System and Databases", IEEE Trans. on PS Vol. 8, No. 2, May 1993, pp 434-400
- [3] M. Foley, A. Bose, W. Mitchell and A. Faustini, "An Object-Oriented Based Graphical User Interface for Power Systems", IEEE Trans. on PS, Vol.8, No.1 Feb. 1993 pp94-104
- [4] S. Li and S. M. Shahidehpour, "An Object-Oriented Power System Graphics Package for Personal Computer Environment", IEEE Trans. on PS, Vol.8, No.3 Aug. 1993 pp1054-1060
- [5] M. Foley, and A. Bose, "Object-Oriented on-line Network Analysis", IEEE Trans. on PS Vol. 10, No. 1, Feb. 1995, pp125-132
- [6] E. Z. Zhou, "Object-Oriented Programming C++ and Power System Simulation", IEEE Trans. on PS Vol. 11, No. 1, Feb. 1996, pp206-215
- [7] A. Manzoni, A. S. e Silva, and I. C. Decker, "Power System Dynamic Simulation Using Object-Oriented Programming", IEEE Trans. on PS, Vol. 14, No. 1 Feb. 1999 pp.249-255
- [8] A. F. Neyer, F. F. Wu, and K. Inhof, "Object-Oriented Programming for Flexible Software: Example of a Load Flow", IEEE Trans. on PS, vol. 5, No. 3, Aug. 1990, pp 689-696

- [9] Mats Larsson, "ObjectStab-An educational Tool for Power System Stability Studies", IEEE Trans. on PS, Vol.19, No.1, Feb. 2004
- [10] Jun Zhu, David L. Lubkeman, "Object-Oriented Development of Software Systems for Power System Simulations", IEEE Trans. on PS, Vol. 12, No. 2, May 1997, pp1002-1007
- [11] Jian Yang, Max D. Anderson, "PowerGraf: An Educational Software Package for Power System Analysis and Design", IEEE Trans. on PS, Vol. 13, No. 4, Nov. 1998, pp1205-1210
- [12] IEEE Committee Report, "Computer Representation of Excitation Systems", IEEE Trans on PAS, Vol.87, No.6 June 1968, pp1460-1463
- [13] IEEE Committee Report, "Excitation System Models for Power System Stability Studies", IEEE Trans. on PAS, vol. 100, no.2 Feb. 1981, pp494-509
- [14] IEEE Committee Report, "Dynamic Models for Steam and Hydro Turbines in Power System Studies", IEEE Trans. on PAS Vol. 92, no 6, nov/dec, 1973 pp1904-1915
- [15] Hadi Saadat, "Power System Analysis", McGraw-Hill, 1999, pp289
- [16] Jaehyeon Gim, "The Efficient Method of Power Flow Calculation using Object-Oriented Programming", Trans. KIEE Vol. No. April 2001 pp166-171
- [17] W. F. Tinney, John W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization", Proceedings of the IEEE vol.55, no.11 Nov. 1967, pp1801-1809
- [18] W. F. Tinney, V. Brandwajn, S.M. Chan, "Sparse Vector Methods", IEEE Trans. on PAS vol. 104, No. 2, Feb. 1985 pp295-301
- [19] Sartaj Sahni, "Data Structures, Algorithms, and Applications in C++", McGraw-Hill, 1998

저 자 소 개



김재현 (金在顯)

1956년 1월 25일 생. 1977년 홍익대학교 전기공학과 졸업. 1989년 University of Texas at Arlington 졸업(석사). 1993년 동 대학교 졸업(박사). 현재 국립 순천대학교 정보통신공학부 부교수.

Tel : 061-750-3545
 Fax : 061-750-3508
 E-mail : jhg@sunchon.ac.kr