Power Systems Software Package with Graphical User Interface and Scripting Capabilities for Research and Education

Jovan Ilic

Interactive Power Systems Simulator IPSYS/GIPSYS/MIPSYS

- Flexible computing environment geared towards power systems
- Scripting, GUI, Matlab, and C/C++ access to network data
- Expandable through scripting, Matlab, C/C++, or Fortran 77
- Simultaneous command line and GUI

1

Features

- IPSYS: command line scripting interface
- GIPSYS: IPSYS with a GUI editor and interface
- MIPSYS: IPSYS algorithms exported to Matlab environment
- Data structures suitable for linear algebra and power systems networks

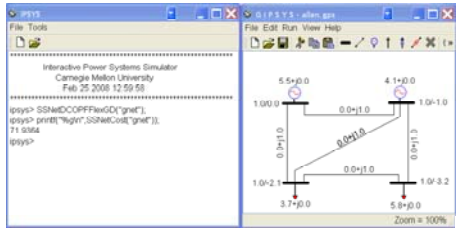
2

User Interfaces

Command Line (IPSYS):

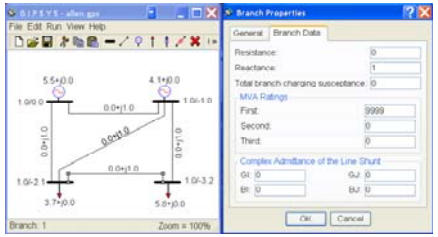
3

Graphical User Interface (GIPSYS):



4

Network Editing



5

Algorithms and Net Access

- Linear algebra data structures and algorithms
- Programmatically read/write network parameters
- Program control flow and scripting functions
- Simultaneous command line execution and GUI display

6

- AC power flow – parallel lines, tap setting transformers, phase shifters, shunts
- Sensitivity calculations
- DCOPF – flexible generation, demand, and/or both
- Distribution factors and DCPF
- ACOPF – in development
- Dynamic programming UC – in development

7

General Idea

- Development of a general framework to integrate new dynamic and or steady state algorithms with the existing ones
- Transparent addition of new C/C++/F77 functions into the scripting language
- Possibility of use parallel architectures for very large systems
- Possibility of real time operations

8

Possible Additions - Modules

- Time domain simulations
- Stability analysis
- Steady state market clearing and interaction
- Real time interface to ISO web sites
- Central database design
- Distributed computation of large systems

9

Design Benefits

- Modular design
- Modules development can proceed in parallel
- Once a module is developed and tested, it can be incorporated automatically
- Modules can be used by each other either through scripting or native code
- Modules mix and match

10

Design difficulties

- Defining a unifying data format standard describing a system in sufficient detail for a range of applications
- Developing a generic graphical user interface for editing and displaying system information for the applications of interest
- Educating students and researchers how to use this tool to its fool potential

11

Conclusion

- It is possible to develop modular computer software for different power systems problems in way that can unify these different areas of interest
- Modules can be added independently of each other
- Usual problem of standardizing input/output data format

12

Electrical & Computer ENGINEERING

Note: This work has been supported by NFS grant EEC-0343760 - Educating 21st Century Power Engineers