SIEMENS
Ingenuity for life

PSS®E Advanced Contingency Analysis and RAS Module

Simulate complex remedial action schemes, and customize the PSS®E power flow and contingency solution

At a glance

In response to a growing need to simulate complex Remedial Action Schemes, and to provide more customization around the PSS®E solutions, Siemens PTI is providing a new add-on module for Advanced Contingency Analysis and Remedial Action Schemes. This module provides:

- Simple, powerful RAS support in PSS®E contingency analysis using high-level Python commands.
- An interface for customization of the PSS®E power flow and contingency solutions in order to define custom power flow control models and custom reporting.

The challenge

Although grid codes require the testing of contingency levels up to N-2 (and sometimes beyond), these requirements cannot be met simply by putting more equipment in the field. Practical systems planning is a combination of capital investments and the design of strategic re-configurations of the system based on real-time system conditions to prevent overloads and collapses.

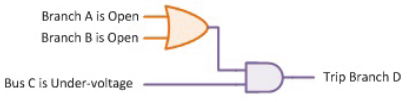
These re-configurations can be manual plans (often referred to as remedial action schemes, or RAS) or they can be fully automated systems (usually called special protection systems, or SPS). Either way, these plans need to be designed and tested in conjunction with contingency analysis in PSS®E. The user needs a convenient yet comprehensive method to define the behavior of their RAS and SPS systems in PSS®E.

Also, beyond RAS modeling, many users desire more control over the powerflow and contingency solution process. For example, customizations are needed for modeling complex HVDC control systems, custom non-linear load models, supervisory control systems, and custom reporting routines.

Our solution

The PSS®E solution for these needs is the Advanced Contingency Analysis and RAS module. This module consists of two parts: a RAS Modeling Interface, and a Powerflow Customization Interface.

RAS Modeling Interface



Example RAS with Nested Logic

The PSS®E approach to modeling RAS combines a straight-forward process for defining the RAS behavior, without compromising on the ability to model even the most complex schemes imaginable. Using this module, RAS definitions are defined by the user as simple Python functions that tell PSS®E what to check for after each contingency (condition functions), and functions that tell PSS®E what actions to take if a RAS is triggered (action functions). For example, the definition of the RAS described in the following logic diagram would look something like this:

```
def  condition():
    a = branch_is_open(151, 152, 1)
    b = branch_is_open(151, 152, 2)
    c = bus_voltage(151) < 0.96
    return (a or b) and c

def action():
    open_branch(152, 202, 1)

define_ras("RAS-1", condition, action)
```

The above RAS definition, along with others, would be added to a **ras.py** file located in the Python Path of your PSS®E application. The definitions would automatically be read and simulated during contingency analysis.

Although RAS and SPS definitions can sometimes be very simple (such as tripping a line under a certain load), they can also be very complex, involving interface flows, complex logic, lookup tables, and arithmetic expressions. Although PSS®E has plans to continue adding additional commands to the TRP (tripping) description language, these TRP commands will always need to be supplemented with the Python RAS interface to cover all of the various RAS schemes used in the industry now and in the future.

Some of the benefits of using Python for the RAS interface are:

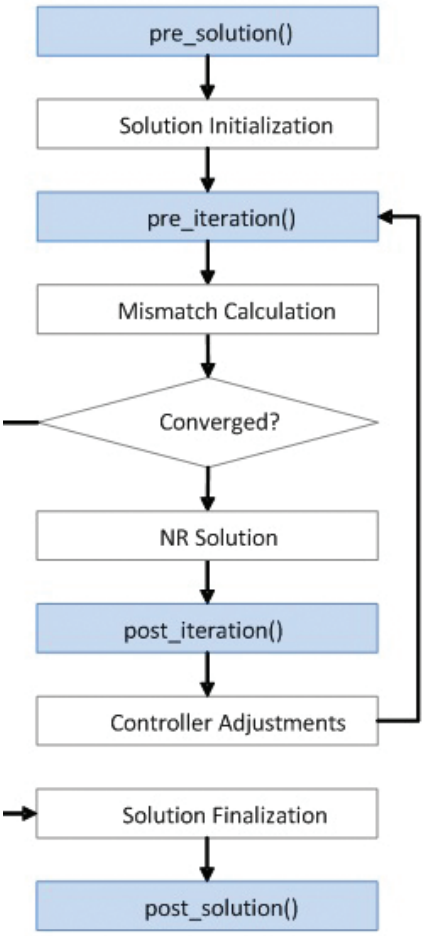
- The amount of Python knowledge needed is very low. The learning curve is comparable to learning the basics of the PSS®E sub, mon, or con formats.
- The RAS and Advanced Contingency Analysis module comes with many example RAS definitions of various complexity that can easily be adapted for your specific needs,
- Almost any data access you need into PSS®E during contingency analysis is available to you. This includes complex use cases, like referencing 2D lookup tables stored in files or writing directly to Excel from within the contingency solution.

Powerflow Customization Interface (PCI)

The RAS modeling interface is just one of the new ways available for the engineer to interact with and customize the behavior of the power flow and contingency solutions. In order to make the RAS modeling possible, we had to add hooks into the solution loop for which the user can provide their own Python functions, effectively allowing arbitrary customizations to the inner-workings of the PSS®E power flow and contingency solutions. Some of the applications of this interface are:

- The implementation of custom power flow control models, such as advanced HVDC control schemes, custom switched-shunt control, or supervisory control models (custom AGC, for example)
- The implementation of custom non-linear load models
- Custom monitoring and reporting. PSS®E does not have built in monitoring for all quantities in the network model, especially quantities based on expressions (such as angle differences). These can easily be monitored and reported on using the PCI.

To define these custom models and routines, a special user-defined Python module, named pssuserpf.py must be provided to PSS®E that implements the interface functions. Several examples of uses of the PCI are included with the module example files to demonstrate the use of this interface.



The PSS®E power flow loop, showing the locations where user-provided functions can be provided for solution customization (shown in blue).

How to get started

For further information or to purchase the PSS®E Advanced Contingency Analysis and RAS Module, please contact Siemens PTI software sales at ***pti-software-sales.ptd@siemens.com*** or +1 518 395 5000.

Siemens Industry, Inc.
Siemens Power Technologies International
400 State Street
P.O. Box 1058
Schenectady, NY 12301-1058 USA