**Technote: OpenDSS and Distribution State Estimation**

The following Tech Note was prepared to address several questions about the DSE capabilities of OpenDSS.

== OpenDSS and Distribution State Estimation ==

As with many OpenDSS functions, there are some relatively simple and straightforward algorithms built in. If the user wishes to do something more complicated, or simply different, the COM interface is available to allow implementation of nearly any algorithm the user desires.

Distribution State Estimation (DSE) with the OpenDSS is done a little differently than, perhaps, one might find in other power system analysis tools. The OpenDSS maintains a physically-based model of the electrical power distribution system complete with detailed models of voltage regulator controls, capacitor controls, etc. The user relies on the OpenDSS to correctly simulate the behavior of the distribution system. There is no need to construct a new set of equations outside of the program. The model within the program is generally sufficient. While many network state estimation algorithms start with voltage estimation, the estimation algorithm presently programmed into OpenDSS first strives to estimate the currents or powers. As Baran, et. al. have found (see references below), it is generally better on North American distribution systems to start with matching the measured currents or powers. The myriad unbalances and numerous discrete voltage regulation devices make this the preferred approach. Then the voltage regulation devices are adjusted to achieve a better match of the voltages.

In the OpenDSS, one cannot arbitrarily force the branch currents to a particular value. In fact, the solution algorithm computes only the node voltages and branch currents must be computed later. The only currents appearing the main set of equations are the currents injected into the network by loads, generators, etc. In the OpenDSS the currents are varied by adjusting the power consumption or production of the Power Conversion elements (PC elements) such as Loads and Generators. In the built-in algorithm, the powers are adjusted until a satisfactory match of the currents at sensor locations is achieved. Sensors represent any element on the distribution line that can measure voltage and current. They can also be fictitious The two classes within OpenDSS that do this are the EnergyMeter and Sensor elements. Sensor elements are responsible for making adjustments to downstream loads. Smart meters with AMI data are defined with appropriate Loadshape object that corresponds to actual measurements. When the kW/kvar or kW/PF are specified for a particular time period, that particular load is no longer adjusted.

Once the PC elements are estimated the normal solution process with automatically adjust tap changer values and capacitor states according to their defined control settings.

== How Well Does it Work? ==

This relatively simple procedure works remarkably well for most radial distribution feeders. Once the current magnitudes or power flows match well at sensor locations, the voltages will generally have only 1-1.5% error. This is within the band of regulator and LTC settings. For any given loading condition there are often three valid tap changer positions that satisfy the control specifications.

Matching active power flow is generally easier than reactive power flow. The expected reactive power flow estimation can be complicated by:

\* Lack of var measurements or power factor measurements at the load

\* Seasonally switched capacitors

\* Single-phase switching of 3-phase capacitors or blown capacitor fuses

The OpenDSS Estimation algorithm always adjusts loads reasonably with physical constraints as defined in the circuit model. For example, 3-phase loads are adjusted as 3-phase elements not three single-phase elements. Therefore, it is ideal if the loads are modeled in a manner that represents the physical makeup. Not all distribution models contain this kind of detail. To have the freedom to adjust each phase individually, 3-phase loads would be modeled as a bank of single-phase loads.

== Custom Estimation Algorithms ==

There are a number of reasons why users may wish to code supplemental estimation algorithms in MATLAB or some other language and perform their own customized estimation. We recognized when the OpenDSS was first written that we would never be able to code enough optimization algorithms, etc., to satisfy every user and circuit condition. That is why the COM interface was provided. Users are able to add their own custom algorithms while letting the OpenDSS handle all the simulation duties.

=== Voltage Estimate Improvement ===

One example would be an algorithm to tighten the voltage estimate. The user may desire a tighter estimate for the voltage than the standard estimation algorithm achieves simply by adjusting the LTC taps at the substation. One thing that can generally be done without violating the spirit of having a physically-based model is to simply raise or lower the source voltage. In actual practice the transmission system voltage will vary significantly in many cases. Another "trick" is to put a Load or a Generator at the substation bus on the source side of the key Energymeter involved in the estimation to be used as a var source to raise of lower the apparent system voltage. In either case, some intelligence outside of the circuit model will be necessary to adjust the source voltage. This would be a good function for some user-written code driving the COM interface.

=== Monte Carlo Estimation Improvement ===

Sometimes there are too many unknowns to get anything approaching an exact estimate. One approach is to used the built-in estimation function to get close and then attempt a number of random changes to variables to see if the error can be reduced. This is a potentially good application for a monte carlo type algorithm. We have employed a simular method for estimating fault currents. Particle swarm methods and genetic algorithms are two other means one might employ to improve the estimated.

=== Custom algorithms with special knowledge of the system ===

State estimation can often benefit from special knowledge of the system being modeled. For example, if there were a large factory load that always took a lunch break at noon, one could take advantage of that knowledge to improve the estimate. Of course, there's no way the built-in algorithm would know that fact, but a supervisory algorithm controlling the software via the COM interface could do that. This would be similar to simulating a DMS system.

== Issues with AMI Data ==

You might think matching loading with AMI data would be a piece of cake -- and it is in many cases. Simply plug in the measured P and Q for each load (meter) and with any luck the computed P and Q at the substation will match what was measured. This would mean losses are properly accounted for. However, there will be missing data in other cases for reasons ranging from hardware problems to communications drop outs. Also, there may be a few large loads that are metered differently and the AMI data will not add up to what is measured at the substation.

It would also be nice if all AMI data contained both P and Q (kW and kvar), but sometimes you only have P. This is usually kWh over the demand interval, which is typically 15 min to 1 h. One problem that may be encountered is that only integer kWh values are recorded. This may make it difficult to get an accurate estimate of the loading for small loads or short intervals.

== References ==

# Baran and McDermott, "State Estimation for Real Time Monitoring of Distribution Feeders", IEEE PES General Meeting, 2009.

--[[User:Rdugan|Rdugan]] 20:24, 16 August 2011 (UTC)