

Powerflow

From GridLAB-D Wiki (UVic shadow)

Introduction

This documentation explains the technical aspects of the distribution power flow computations as implemented for GridLAB-D. The power flow module provides electrical distribution system modeling for power flow solutions. A power flow calculation is performed to determine what the steady state node voltages and line currents are at each point of the system, given the system model, electrical loads connected at each node, and voltage at the substation. The power flow problem is solved using a three-phase unbalanced flow solver. When the topology is strictly radial from the feeder to the loads, the forward-back sweep method is used, which is the default method. When the topology is non-radial a three-phase unbalanced Gauss-Seidel or Newton-Raphson methods are used. The specific methodology and equations of the forward-back sweep method are described in Kersting (2007). The Gauss-Seidel implementation is described in Grainger and Stevenson (1994). The Newton-Raphson method is described in Garcia, et al. (2000).

Algorithm Selection

Contents

- 1 Introduction
 - 1.1 Algorithm Selection
 - 1.1.1 Forward-back sweep
 - 1.1.2 Gauss-Seidel
 - 1.1.3 Newton-Raphson
- 2 Component Modeling
 - 2.1 Node
 - 2.2 Load
 - 2.3 Meter
 - 2.4 Overhead and Underground Lines
 - 2.4.1 Line Shunt Admittance
 - 2.5 Secondary Lines
 - 2.6 Transformers
 - 2.6.1 Delta-Grounded Wye (Step-down)
 - 2.6.2 Delta-Grounded Wye (Step-up)
 - 2.6.3 Ungrounded Y- Δ (Step-down)
 - 2.6.4 Ungrounded Y- Δ (Step-up)
 - 2.6.5 Grounded Y-Grounded Y (Step-up and Step-down)
 - 2.6.6 Δ - Δ (Step-up and Step-down)
 - 2.6.7 Open Y-Open Δ (Step-down)
 - 2.6.8 Single-Phase (Step-down)
 - 2.6.9 Single-Phase Center-Tapped (Step-down)
 - 2.6.9.1 A-Phase Connected Primary
 - 2.6.9.2 B-Phase Connected Primary
 - 2.6.9.3 C-Phase Connected Primary
 - 2.7 Regulators
 - 2.8 Switch
 - 2.9 Fuse
 - 2.10 Substation
 - 2.10.1 Equations
 - 2.11 Capacitor
 - 2.12 Coordinated Volt-Var Control
- 3 References
- 4 See also

By default, the forward-back sweep algorithm is used to solve powerflow models in GridLAB-D. To change the solver method, a `solver_method` switch must be passed in your .GLM file. When calling the powerflow module, specify `FBS` for the forward-back sweep method, `GS` for the Gauss-Seidel method, or `NR` for the Newton-Raphson method. For example,

```
module powerflow{
  solver_method GS;
}
```

would make the powerflow module objects available to your model file and solve the resultant powerflow using the Gauss-Seidel method.

Forward-back sweep

The basic equations are in the following forms:

Backward sweep: $[I_{abc}]_n = [c] \times [V_{abc}]_m + [d] \times [I_{abc}]_n$

Forward sweep: $[V_{abc}]_m = [A] \times [V_{abc}]_m + [B] \times [I_{abc}]_n$

where the $[c]$, $[d]$, $[A]$, and $[B]$ matrices represent individual characteristics of each link section as described in Kersting (2007).

Gauss-Seidel

The basic Gauss-Seidel equation is:

$$V_i^{(k)} = \frac{1}{Y_{ii}} \left[\frac{P_{i,sch} - jQ_{i,sch}}{V_i^{(k-1)*}} - \sum_{j=1}^{i-1} Y_{ij} V_j^{(k)} - \sum_{j=i+1}^N Y_{ij} V_j^{(k-1)} \right]$$

where

- V is the node voltage;
- i is the current node;
- k is the current iteration number;
- Y_{ii} is the self admittance matrix;
- $P_{i,sch}$ is the scheduled real power injection;
- $Q_{i,sch}$ is the schedule reactive power injection; and
- Y_{ij} is the admittance of the line connecting node i with the j node.

As a requirement of the Gauss-Seidel method, one node in the system must be designated as a swing bus. This node represents an infinite bus and provides a fixed voltage reference for Gauss-Seidel iterations.

The Gauss-Seidel solver is an ongoing implementation. Most objects in the powerflow library are available for use. Objects that still need to be implemented include switches, relays, fuses, split-phase transformers, and regulators.

Newton-Raphson

Newton-Raphson is currently under development. Some features in the powerflow solver may work at this time,

but the implementation is not complete.

The Newton-Raphson solver is comprised of two principle sets of equations. The first set of equations describes the current injection from loads into the system. They are split into a real and imaginary component and are given as:

$$\Delta I_{rk}^s = \frac{(P_k^{sp})^s V_{rk}^s + (Q_k^{sp})^s V_{mk}^s}{(V_{rk}^s)^2 + (V_{mk}^s)^2} - \sum_{i=1}^n \sum_t (G_{ki}^{st} V_{r_i}^t - B_{ki}^{st} V_{m_i}^t)$$

$$\Delta I_{mk}^s = \frac{(P_k^{sp})^s V_{mk}^s + (Q_k^{sp})^s V_{rk}^s}{(V_{rk}^s)^2 + (V_{mk}^s)^2} - \sum_{i=1}^n \sum_t (G_{ki}^{st} V_{m_i}^t - B_{ki}^{st} V_{r_i}^t)$$

where

- ΔI is the current injection at that bus
- k is the bus number
- s is the current phase of interest
- t represents all of the phases connected to the bus
- P is the real power component of a load on the bus
- Q is the reactive power component of a load on the bus
- E_k represents the voltage of the bus such that
 - V_{rk} is the real portion of the voltage
 - V_{mk} is the imaginary portion of the voltage, so
 - $E_k = V_{rk} + jV_{mk}$

With the current injections calculated, the voltage updates are computed via

$$\begin{bmatrix} \Delta I_{mk} \\ \Delta I_{rk} \end{bmatrix} = -\mathbf{J}^{-1} \begin{bmatrix} \Delta V_{mk} \\ \Delta V_{rk} \end{bmatrix}$$

where \mathbf{J}^{-1} represents the inverse Jacobian given by $\mathbf{J} = \begin{bmatrix} \frac{\delta \Delta I_{mk}}{\delta V_{rk}} & \frac{\delta I_{mk}}{\delta V_{mk}} \\ \frac{\delta \Delta I_{rk}}{\delta V_{rk}} & \frac{\delta I_{rk}}{\delta V_{mk}} \end{bmatrix}$.

As with the Gauss-Seidel method, one node must be designated a swing or slack bus. This node will represent an infinite bus and provides the fixed voltage reference for the solver iterations.

The Newton-Raphson solver utilizes the superLU sparse matrix solver package by default. An external matrix solving package can be utilized by following the specifications outlined on the Powerflow External LU Solver Interface page.

The Newton-Raphson solver is currently complete, except for some specific regulator models. All other

objects are functional at this time.

Component Modeling

The method used for modeling components in the power flow module is consistent with Kersting (2007). Some minor manipulations were required for Gauss-Seidel implementation. The following sections are included to address any differences between our method and those described in Kersting (2007).

Node

Nodes represent busses or junctions in the distribution topology. The unbalanced three-phase voltage is calculated for each node and is available via the various output methods of GridLAB-D. The different solvers utilize nodes differently and support different node types.

The Forward-Back Sweep method (Kersting's method) treats all nodes as the same. Every node is treated as a connection point for lines and provides voltage levels at various points in the system. Loads on the system are handled explicitly in the load object.

The Gauss-Seidel and Newton-Raphson methods have more explicitly defined bus types. Three bus types are available in both solvers. They are:

- **SWING** - the infinite bus, voltage reference of a particular system
- **PQ** - a standard bus providing voltage levels (loads are handled explicitly in load objects)
- **PV** - a fixed voltage magnitude bus

and are specified with the input parameter `bustype`. All busses are `PQ` by default, so if a swing bus needed to be designated, use the parameter

```
bustype SWING;
```

in the input file.

Despite the solver method selected, nodes provide a voltage point at a specific point in the system. The values for each of the three phases of the distribution system are contained in the `voltage_A`, `voltage_B`, and `voltage_C` parameters. Furthermore, for bus types like `SWING` and `PV`, the voltage parameters are used to specify the fixed specifications of that node.

Along with line-to-neutral voltages, the line-to-line voltages are also available from nodes. These values are available from the `voltage_AB`, `voltage_BC`, and `voltage_CA` parameters.

Load

Loads are very similar to nodes and are handled the same way within the solver. Loads can inherit any properties of a node, so for the Gauss-Seidel and Newton-Raphson solvers a load could be specified as a `PV` bus with negative load and represent a generator.

Loads can explicitly model three different types of loads connected to the distribution system. These loads are:

- constant power
- constant current
- constant impedance

The different loads are specified through the `constant_power_x`, `constant_current_x` and `constant_impedance_x` input parameters respectively. In each of the loading types, the x value is substituted by the appropriate phase. For example, if a constant current load of $1.044 - j0.98$ Amps were needed on phase A, the load object would have a parameter of

```
constant_current_A 1.044-j0.98;
```

Loads are primarily connected in one of two forms: wye or delta. The connection of the different aspects of the load are determined by the `phases` parameter of the load. If a `D` appears anywhere in the `phases` parameter, a delta connection is assumed. For example,

```
phases ABD;
```

would indicate that a delta connected load exists on the line-to-line connection AB.

Many IEEE test systems and feeders have all of the loads defined in terms of power, but subclassed as either a power, current, or impedance load. Due to GridLAB-D's ability to explicitly model these load types, the constant current and impedance load values must be translated from their power rating into an Amperage or impedance value.

To calculate the constant current load from a system, use the equation

$$I_{load} = \left(\frac{P}{V} \right)^*$$

where

- I_{load} is the constant current load,
- P is the specified power value of the load,
- V is the nominal voltage at the load bus, and
- $*$ is the complex conjugate operator.

It is important to point out that V is not a magnitude value. It is the nominal voltage on the particular phase of interest. For example, phase C in a balanced system with a nominal voltage magnitude of 2400 Volts would be represented as $V = 2400 \angle 120^\circ$. If the proper angle is not used, the current load value will be incorrect and produce incorrect answers in your system.

Constant impedance loads are calculated using the equation

$$Z_{load} = \left(\frac{V \cdot V^*}{P} \right)^*$$

where Z_{load} represents the constant impedance load.

Unlike the current calculations, the impedance calculation does not require as much care with the nominal voltage. In the numerator of the equation is $V \cdot V^*$ which is the same as $|V|^2$. However, the outside conjugation must still be considered to obtain the correct impedance value.

Meter

Meters are also very similar to node objects. Meters provide a method for measuring the instantaneous power or energy over time that is flowing through a node. This is useful for time-varying simulations with `recorder` objects attached. GridLAB-D's normal output methods (.XML, .TXT) will only record the final timestep of a system, so for time varying systems a `recorder` and `meter` are needed to track power and energy, as well as voltage and current.

Due to the nature of GridLAB-D's solvers, current passing through a node or load is not directly measurable. Rather, a meter must be used to examine this current flow. The complex current flowing through each of the three phases in a system is available via the `measured_current_A`, `measured_current_B`, and `measured_current_C` variables. The A, B, and C letters represent the value obtained from each of the three phases of the system.

All power measurements in the meter are based on the fundamental power equation given by

$$S = P + jQ = V \cdot I^*$$

where V and I represent the complex voltage and current respectively. Three different power measurements are available from a meter and are defined as:

- `measured_power` or power magnitude as $|S| = |S_A| + |S_B| + |S_C|$,
- `measured_real_power` as $P = P_A + P_B + P_C$, and
- `measured_reactive_power` as $Q = Q_A + Q_B + Q_C$

where A , B , and C represent the three different phases of the distribution system.

Energy passing through a metered point in the system is also available. The energy is obtained using the simple equation

$$E = |S| \cdot t$$

with $|S|$ representing the sum of the magnitudes of power of the three phases and t represents the time in hours. The result is a measure of energy passing through the meter in Watt-hours available on the variable `measured_energy`.

Overhead and Underground Lines

Overhead and underground (concentric neutral and tape-shield) lines are both supported. Single-phase and three-phase lines with a neutral conductor are supported. Systems with more than 3 phases, i.e. n-phase configurations, are not currently supported. Equations used are consistent with Kersting (2007).

Line Shunt Admittance

By default, the `powerflow` module does not model the shunt admittance of underground or overhead lines. This feature can be enabled by setting `line_capacitance` to `true`, similar to:

```
module powerflow {
    line_capacitance true;
};
```

If `line_capacitance` is enabled and relevant parameters are missing from the individual lines, shunt admittance portions of those devices will be excluded.

Secondary Lines

Single-phase triplex secondary cable is supported. Equations used are consistent with Kersting (2007).

Transformers

For three-phase Delta-connected secondary transformers, the impedance within the Delta windings must be calculated. With standard transformer per unit calculations, the secondary line impedance is calculated. Equivalent impedance within the Delta is calculated by multiplying the calculated transformer secondary impedance by 3. The equations for Delta secondary in the following subsections use the impedance within the Delta.

For single-phase transformers connected in a three-phase Delta configuration, the impedance of each individual transformer is equivalent to the impedance within the Delta. Therefore, calculation of the secondary impedance is straightforward.

These equations assume that the transformers modeled are consistent with C57.12.00 (2006): "*The angular displacement between high-voltage and low-voltage phase voltages of three-phase transformers with Y-Δ or Δ-Y connections shall be 30°, with the low voltage lagging the high voltage...*"

Equations for Y-Y or Δ-Δ are the same for step-up or step-down transformers. Equations for step-up and step-down cases are different for the Wye-Delta and Delta-Wye connections because of the "American Standard Thirty Degree" connection, as described in Kersting (2007) and C57.12.00 (2006).

Supported transformers and are discussed in the following subsections.

Delta-Grounded Wye (Step-down)

Equations for the Delta-Grounded Wye step-down transformer are consistent with Kersting (2007).

Delta-Grounded Wye (Step-up)

The equations for the step-up and step-down transformer are not identical because of the use of the "American Standard Thirty Degree" connection, as described in Kersting (2007) and IEEE C57.12.00 (2006). In order to obtain the equations for the step-up transformer from the step-down transformer equations the $[c]$ and $[d]$ matrices must be multiplied by:

$$S = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Matrix $[A]$ must be multiplied by:

$$S^{-1} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$

Ungrounded Y-Δ (Step-down)

Equations for the Ungrounded Y-Δ step-down transformer are consistent with Kersting (2007).

Ungrounded Y-Δ (Step-up)

The equations for the step-up and step-down transformer are not identical because of the use of the "American Standard Thirty Degree" connection, as described in Kersting (2007) and C57.12.00 (2006). In order to obtain the equations for the step-up transformer from the step-down transformer equations the c and d matrices must be multiplied by:

$$S = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Matrix A must be multiplied by:

$$S^{-1} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$

Grounded Y-Grounded Y (Step-up and Step-down)

Equations for the Grounded Y-Grounded Y step-up and step-down transformers are consistent with Kersting (2007).

Δ-Δ (Step-up and Step-down)

Equations for the Δ-Δ step-up and step-down transformers are consistent with Kersting (2007).

Open Y-Open Δ (Step-down)

Equations for the Open Wye-Open Δ step-down transformers are consistent with Kersting (2007).

Single-Phase (Step-down)

Not yet implemented.

Single-Phase Center-Tapped (Step-down)

Single-phase center-tapped transformers mark the transition point between the primary and secondary

distribution system. These are the transformers that step the voltage down from the primary distribution system voltage (for example, 12.47 kV) to the residential voltage (120 V and 240 V).

Transformers are modeled using an interlaced design. The representative equations are created using the method described in Kersting (2007), with the exception that 3 x 3 matrices are in used in lieu of 2 x 2 matrices and shunt impedances have been incorporated into the equations. The specific formats are shown below.

A-Phase Connected Primary

$$[a] = \begin{bmatrix} Z_{eq}n_t & 0 & 0 \\ Z_{eq}n_t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[b] = \begin{bmatrix} Z_{eq}n_t Z_1 + \frac{Z_0}{n_t} & -\frac{Z_0}{n_t} & 0 \\ \frac{Z_0}{n_t} & -\left(Z_{eq}n_t Z_2 + \frac{Z_0}{n_t}\right) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[c] = \begin{bmatrix} \frac{n_t}{Z_c} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[d] = \begin{bmatrix} \frac{Z_1 n_t}{Z_c} + \frac{1}{n_t} & \frac{-1}{n_t} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[A] = \begin{bmatrix} \frac{1}{n_t} & 0 & 0 \\ \frac{1}{n_t} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} Z_1 + \frac{Z_0}{Z_{eq}n_t^2} & -\frac{Z_0}{Z_{eq}n_t^2} & 0 \\ \frac{Z_0}{Z_{eq}n_t^2} & -\left(Z_2 + \frac{Z_0}{Z_{eq}n_t^2}\right) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

B-Phase Connected Primary

$$[a] = \begin{bmatrix} 0 & Z_{eq}n_t & 0 \\ 0 & Z_{eq}n_t & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[b] = \begin{bmatrix} Z_{eq}n_tZ_1 + \frac{Z_0}{n_t} & -\frac{Z_0}{n_t} & 0 \\ \frac{Z_0}{n_t} & -(Z_{eq}n_tZ_2 + \frac{Z_0}{n_t}) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[c] = \begin{bmatrix} 0 & 0 & 0 \\ \frac{n_t}{Z_c} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[d] = \begin{bmatrix} 0 & 0 & 0 \\ \frac{Z_1n_t}{Z_c} + \frac{1}{n_t} & \frac{-1}{n_t} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[A] = \begin{bmatrix} 0 & \frac{1}{n_t} & 0 \\ 0 & \frac{1}{n_t} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} Z_1 + \frac{Z_0}{Z_{eq}n_t^2} & -\frac{Z_0}{Z_{eq}n_t^2} & 0 \\ \frac{Z_0}{Z_{eq}n_t^2} & -(Z_2 + \frac{Z_0}{Z_{eq}n_t^2}) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

C-Phase Connected Primary

$$[a] = \begin{bmatrix} 0 & 0 & Z_{eq}n_t \\ 0 & 0 & Z_{eq}n_t \\ 0 & 0 & 0 \end{bmatrix}$$

$$[b] = \begin{bmatrix} Z_{eq}n_t Z_1 + \frac{Z_0}{n_t} & -\frac{Z_0}{n_t} & 0 \\ \frac{Z_0}{n_t} & -\left(Z_{eq}n_t Z_2 + \frac{Z_0}{n_t}\right) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[c] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{n_t}{Z_c} & 0 & 0 \end{bmatrix}$$

$$[d] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{Z_1 n_t}{Z_c} + \frac{1}{n_t} & \frac{-1}{n_t} & 0 \end{bmatrix}$$

$$[A] = \begin{bmatrix} 0 & 0 & \frac{1}{n_t} \\ 0 & 0 & \frac{1}{n_t} \\ 0 & 0 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} Z_1 + \frac{Z_0}{Z_{eq}n_t^2} & -\frac{Z_0}{Z_{eq}n_t^2} & 0 \\ \frac{Z_0}{Z_{eq}n_t^2} & -\left(Z_2 + \frac{Z_0}{Z_{eq}n_t^2}\right) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $Z_{eq} = \frac{Z_0 + Z_c}{Z_c}$

and

$Z_c = \frac{R_c jX_c}{R_c + jX_c}$ is the core shunt impedance.

Voltage and current values used for the secondary system sweeps are shown below.

Backward sweep: $[I_{abc}] = [c] \times [V_{12n}] + [d] \times [I_{12n}]$

Forward sweep: $V_{12n} = [A] \times [V_{abc}] - [B] \times [I_{12n}]$

where

- I_{abc} is the primary current on lines a , b , and c
- I_{12n} is secondary current on lines 1, 2, and n

- V_{abc} is primary voltage on lines a , b , and c
- V_{12n} is secondary voltage on lines 1, 2, and n

Regulators

Single-Phase Y-connected, Single-Phase Δ connected, and Single-Phase Open Δ regulators all are supported in MANUAL control modes.

Single- and Three-phase Y-connected regulators are supported in multiple automatic control modes, including REMOTE_NODE, OUTPUT_VOLTAGE, and LINE_DROP_COMP.

Regulators use a supporting regulator configuration object to define the properties. The regulator object itself defines standard link parameters, such as to, from, phases, and configuration.

REMOTE_NODE uses a remote sensing nodes, named sense_node to control voltage at a given node in the system. Voltage is not corrected (PT & CT ratio are not used), meaning band_center and band_width are given by the actual voltage desired at that node.

OUTPUT_VOLTAGE looks at the actual output voltage of the regulator at the node it is connected to. Similar to REMOTE_NODE control mode, it looks at the actual voltages at the node.

Finally, LINE_DROP_COMP is modeled according to Kersting (2007), using a model for a line drop compensator, where current out of the regulator is used to estimate the voltage at a specific point on the system determined by the compensator settings (current_transducer_ratio, power_transducer_ratio, and r & x settings for each individual phase).

Mechanical and dwelling time changes have also been implemented. Mechanical delays, called by time_delay, indicates the physical amount of time it takes a tap to actually change. Dwelling time, called by dwell_time, is the amount of time the regulator waits between state changes to determine whether the voltage change is a fluctuation or long-term change.

Matrix equations used are consistent with Kersting (2007), for both Type A & Type B regulators.

Example of regulator properties:

```
object regulator_configuration:1 {
  connect_type      WYE_WYE
                    (or OPEN_DELTA_ABBC);
  band_center       120.0 V;
  band_width        2.0 V;
  time_delay        15 s;           //mechanical time delay
  dwell_time        45 s;           //wait time
  raise_taps        16;             //upper and lower tap limits
  lower_taps        16;
  current_transducer_ratio 600;      //only used for LINE_DROP_COMP
  power_transducer_ratio 60;         //  --each phase can be set individually
  compensator_r_setting_A 6.0;       //
  compensator_r_setting_B 6.0;       //
  compensator_r_setting_C 6.0;       //
  compensator_x_setting_A 16.0;      //
  compensator_x_setting_B 16.0;      //
  compensator_x_setting_C 16.0;      // -
  regulation        0.1;
  Control            LINE_DROP_COMP; //defines control mode
                    (or REMOTE_NODE, MANUAL, OUTPUT_VOLTAGE)
  Type              A;
                    (or B)
  tap_pos_A         0;
```

```

tap_pos_B          0;                                //initial tap positions
tap_pos_C          0;
}
}

object regulator {
  name      test_regulator;
  from      node_1;
  to        node_2;
  phases    ABCN;
  configuration regulator_configuration:1;
  (sense_node node_5000;)                                //used when in REMOTE_NODE control mode
}

```

Switch

System line switches are modeled as shown below.

If in service, $Z = 0.0001$

If out of service, $Z = \infty$

$$[a] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} [b] = [Z_{abc}] [c] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} [d] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} [B] = [Z_{abc}]$$

Fuse

System line fuses have been modified as a simple over-current device. The equations are shown below.

If in service, $Z = 0.0001$

If out of service, $Z = \infty$

$$[a] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} [b] = [Z_{abc}] [c] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} [d] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} [B] = [Z_{abc}]$$

Substation

The substation object in the GridLAB-D powerflow module performs two objectives. The substation reads the `load_voltage` property from the `pw_load` parent, if present, and converts this positive sequence value to the equivalent balanced three-phase voltages to act as the swing bus voltages for the powerflow solution. The substation takes the three phase unbalanced power solution seen at the substation node, calculates the average power on the phases, and writes this average to the `load_power` property in the `pw_load` parent, if present. The substation node also passes positive sequence ZIP components, explicitly set at the substation, to the `pw_load` parent. In addition, there is a property that allows the user to specify which phase at the substation is the reference phase for the GridLAB-D powerflow solution. The substation object is updated to keep track of the three phase power solution.

Substation is a child class of the node object inside the powerflow module.

Equations

Substation uses the following equation to convert the positive sequence value from its `pw_load` connection (`load_voltage`) to the three phase balanced voltages used as the swing bus voltage solution.

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} * \begin{bmatrix} 0 \\ V_{positivesequence} \\ 0 \end{bmatrix} * b.$$

Where b is conditional upon which phase is chosen as the reference phase, a is the complex number $1\angle 120^\circ$, and all other variables are complex. The values of b for each of the possible reference phases are shown below.

| Reference Phase | b |
|-----------------|-------|
| Phase A | 1 |
| Phase B | a |
| Phase C | a^2 |

The average phase load is determined by the below equation. All variables are complex values not the real power loads.

$$P_{load} = \frac{P_A + P_B + P_C}{3}$$

A typical substation implementation is

```
object substation {
    name SubS;
    bustype SWING;
    parent network_node;
    reference_PHASE_A;
    phase ABCN;
    nominal_voltage 7199.558;
}
```

Capacitor

The capacitor object provides a method to attach reactive compensation into the distribution system. Two principle modes are available to the capacitor operation: manual and automatic. Under the manual mode, capacitors are switched on by the system modeler explicitly. In one of the automatic methods, the capacitor object will switch capacitors on and off based on criteria specified.

The physical configuration of the capacitors is dictated by the input `phases_connected`. Using a syntax identical to the `phases` keyword in all powerflow objects, the configuration of the attached capacitors can be defined. For example, if

```
phases_connected A;
```

were passed as the argument in the input, a single wye-connected capacitor would be connected to phase A of the system. To obtain a delta connection, a `D` value must be specified in the `phases_connected` property. If a delta-connected capacitor bank on phases AB and BC were desired, the input file would specify

```
phases_connected ABCD;
```

It is important to note that in this case, a phase connection between CA is also implied. To obtain the open-CA configuration, the `capacitor_c` value would need to be set to 0 VAr.

Capacitance values are defined as the nominal reactive power the capacitor can provide. This will be expressed in terms of Volt-Amps reactive or VARs. As alluded to, these values are specified on inputs `capacitor_A`, `capacitor_B`, and `capacitor_C` on the system. To convert this value into an equivalent reactance, this nominal power must be converted into a nominal impedance value. This is accomplished in one of two ways. The first is to specify the nominal voltage rating of the capacitor via the `cap_nominal_voltage` variable. This variable is useful for connections other than the default powerflow connection (e.g. a delta-connected capacitor bank on a wye-connected system) or where the nominal voltage rating of the capacitor is different than the surrounding powerflow objects. In the absence of this variable, the standard nominal voltage of the system at that point, `nominal_voltage`, is used.

Regardless of the nominal voltage used, the impedance value is calculated in the same manner. Using the constant impedance calculation for the load object, the capacitor impedance is determined by

$$Z_{cap} = \left(\frac{V \cdot V^*}{P} \right)^*$$

where V is the nominal voltage, P is the nominal power rating, and $*$ again represents the complex conjugate operator. Once obtained, this impedance value is applied as a load to the system any time the appropriate switch of the capacitor is closed. During open periods, an impedance of $Z_{cap} = \infty$ is used.

As mentioned, the capacitor has two primary modes of operation. When the `control_method` variable is set to `MANUAL`, the capacitor switches are controlled explicitly using the `switchA`, `switchB`, and `switchC` inputs to the object. Simple `OPEN` or `CLOSED` values indicate whether the switch should be open or closed during the analysis.

The capacitor has three automatic methods of determining the switching point. The first of these is a voltage control method set using `control_method VOLT`. Two set-points are defined on the input parameters of the capacitor. When the voltage measured falls below the value specified in `voltage_set_low`, the capacitor for that

phase is switched in. When the voltage exceeds the value of `voltage_set_high`, the capacitor switches off if it was already on. Proper spacing of the two set-points provides a deadband of operation and prevents the capacitor bank from constantly switching off and on.

A second automatic method is determined using the reactive power value at a measurement point. With `control_method` specified as `VAR`, the reactive power is monitored against specific threshold points. Under the VAR control method, if the VAR value goes below the `var_set_low` value, the capacitor will switch off. If the VAR value exceeds the `var_set_high` parameter, the corresponding capacitor will be switched on. Once again, set point spacing is important to prevent excessive capacitor switching.

A third automatic method is a hybrid method of the first two. When `control_method` is set to `VARVOLT`, both set point pairs will be evaluated. The capacitor behaves basically like the VAR control scheme. Normally, the capacitor is switched in and out based off the measured reactive power and its relation to the `var_set_low` and `var_set_high` parameters. However once a switching action occurs, the system checks an additional safety threshold. The capacitor examines the voltage magnitude to see if it exceeds `voltage_set_high`. If this voltage limit is exceeded, that particular phase of the capacitor is switched off (or prevented from switching on) and locked out for a duration specified in `lockout_time`. After `lockout_time` has cleared, the capacitor tries to resume switching operations on that phase.

For all of the automatic control schemes, control is provided to only monitor certain phases of the system. The input parameter `pt_phase` is used to specify which phase to monitor for switching operations. This parameter takes the same format as `phases_connected` with values of A, B, C, and D. For example, if voltage control was being used as an input of

```
pt_phase BCD;
```

was provided, the BC line-line voltage would be monitored to determine the switching point of the capacitors.

When the `VAR` or `VARVOLT` control scheme is selected, a remote line must be specified for VAR monitoring. This is accomplished by specifying the `remote_sense` property with the name of the line. For example, if we wanted to monitor a line called `Bus401to402` we would include

```
remote_sense Bus401to402;
```

in the input parameter list to the capacitor. Once this line is specified, all VAR-related switching operations will be performed based on the quantity measured from this line.

Under default operation, the `VOLT` and `VARVOLT` control schemes only monitor the local voltage of the node to which the capacitor is connected. To offer further flexibility and reliability, it is often desirable to switch the capacitor based on voltage conditions elsewhere in the system. Often times, this will be a node downstream in the feeder. To utilize a remote node's values instead of the attachment point of the capacitor, two different parameters are utilized.

Under the `VOLT` control scheme, the `remote_sense` parameter can be used to specify the remote node by its name. Similar to the VAR monitoring method, if a bus with the name `Bus754` was the desired reference point for capacitor switching decisions,

```
remote_sense Bus754;
```


would be specified on the input parameter list to the capacitor.

Given the nature of the VARVOLT control scheme, a remote line to monitor must always be specified. Therefore, if it is desired to monitor the voltage on a remote node under this scheme, the `remote_sense_B` parameter must be specified. This merely provides a second remote measurement point for the capacitor to monitor. For the VARVOLT control method, the remote line and node specifications could be interchanged without issue. If the examples above are combined, a VARVOLT controlled capacitor that monitors the reactive power on line Bus401to402 and the voltage on node Bus754 would have

```
remote_sense Bus401to402;
remote_sense_B Bus754;
```

in its parameter list. It is important to note that these two parameters are only BOTH active for the VARVOLT case. Furthermore, if both are used, one must specify a line and the other must specify a node. The `remote_sense_B` parameter can not be used to monitor a second line of interest.

The capacitor object handles two variations on the capacitor operation. The input parameter `control_level` lets the modeler specify if all capacitors will be operated individually or as a bank. With

```
control_level INDIVIDUAL;
```

each capacitor switch is operated independently. However, if

```
control_level BANK;
```

is specified, the capacitors are either all switched in or all switched off based on one phase of interest. Coordinating with the `pt_phase` variable, if the monitored phase of interest on phase C indicated a switching condition, the switches to capacitors A, B, and C would all close. When C requested an open condition, all three phases would then open.

To provide further functionality in the capacitor, two time delays are also available. The first of these time delays represents a mechanical switching delay. The value of `time_delay` will specify how many seconds after a change is requested on the capacitors that the switch actually responds.

The second delay, `dwell_time`, represents a "required period of consistency" before switching. For example, if

```
dwell_time 2.0;
```

were specified, the system would need to request the same capacitor action over two seconds before the switch would be actuated (which would then be subject to the mechanical delay specified in `time_delay`). This is useful to prevent large, single second long "transient" spikes from erroneously switching the capacitors.

The `dwell_time` and `time_delay` values are both utilized by the automatic control schemes. However, only `time_delay` is factored into manual operation of the capacitors. Any system "consistency" intervals are left to the modeler to determine and monitor.

Coordinated Volt-Var Control

The Coordinated Volt-VAR Control (CVVC) object provides a means to coordinate regulators and capacitors on a distribution feeder, or group of feeders. Using the logic from Borozan et al. (2001), a desired voltage level and reactive power compensation for a group of feeders is controlled by a central entity.

The CVVC object works by examining two collections of measurements. The first set of these are voltage measurement points throughout the system. These points are typically node objects specified by name. Utilizing a combination of these points, and which regulator they are attached to, the CVVC object adjusts regulator tap positions to maintain a desired voltage.

The secondary set of measurements comes from a `link`-based object somewhere in the system. This link is typically at the top of all the feeders, and may link the transmission network to the distribution (via a subtransmission network). Using the logic outlined in Borozan et al. (2001), the capacitors in the system are switched in an out based on size and distance from the feeder. The capacitor changes only occur during times when no regulator change is present. If a regulator is still changing, the reactive power logic is deferred as small reactive power changes can occur from the voltage adjustments.

The CVVC object supports activation and deactivation during a simulation. By changing the `control_method` variable, the system can transition between an active and standby states. This attempts to provide insight into how the system may behave if only operated during certain periods of the day. When in standby mode, capacitors and regulators are returned to their original control modes (whether that be automatic or not).

References

- Kersting W. H. 2007. *Distribution System Modeling and Analysis*. Second Edition. CRC Press, Boca Raton, Florida.
- Grainger J. and Stevenson, Jr., W. 1994. *Power System Analysis*. McGraw Hill Inc., New York, New York.
- Garcia P., Periera J., Carneiro, Jr., S., de Costa V., and Martins N. 2000. *Three-Phase Power Flow Calculations Using the Current Injection Method*. IEEE Transactions on Power Systems, vol. 15, no. 2, pp. 508-514, May 2000.
- IEEE C57.12.00-2006. IEEE Standard for Standard General Requirements for Liquid Immersed Distribution, Power, and Regulating Transformers. Available online at: www.ieee.org
- Borozan, V., M.E. Baran, and D. Novosel, *Integrated Volt/Var Control in Distribution Systems*, in Proceedings of the 2001 Power Engineering Society Winter Meeting, vol. 3, Jan. 28-Feb. 01, 2001, Columbus, OH, USA, pp. 1485-1490.

See also

[Power_Flow_User_Guide](#)

Retrieved from "<http://gridlabd.me.uvic.ca/wiki/index.php?title=Powerflow&oldid=6191>"

-
- This page was last modified on 19 March 2013, at 12:14.