Coordinated Initialization of the Load Distribution Equivalent, Load Characteristic, and Load Distributed Generation Models

Date : January 7, 2016

Prepared by : James Weber, Ph.D.
Director of Software Development
PowerWorld Corporation
(217) 384-6330 ext. 13
weber@powerworld.com

PowerWorld Corporation

2001 South First Street
Champaign, IL 61820
(217) 384-6330
www.powerworld.com

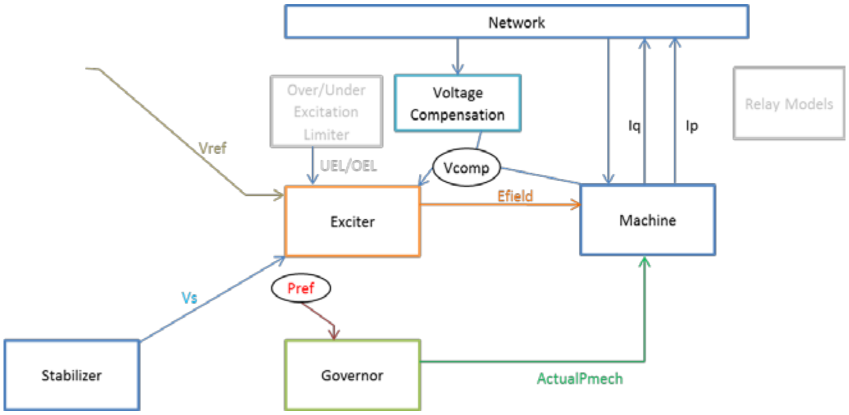
Table of Contents

1 Modular Structure of Transient Stability

Generators in transient stability have always had a very modular structure with the following 8 modules existing for decades in transient stability software.

- Machine (Generator/Converter Model)
- Exciter (P and Q controller)
- Governor (Drive Train)
- Stabilizer (Pitch Control)
- Under Excitation Limiter
- Over Excitation Limiter
- Compensator Model
- Relay Model

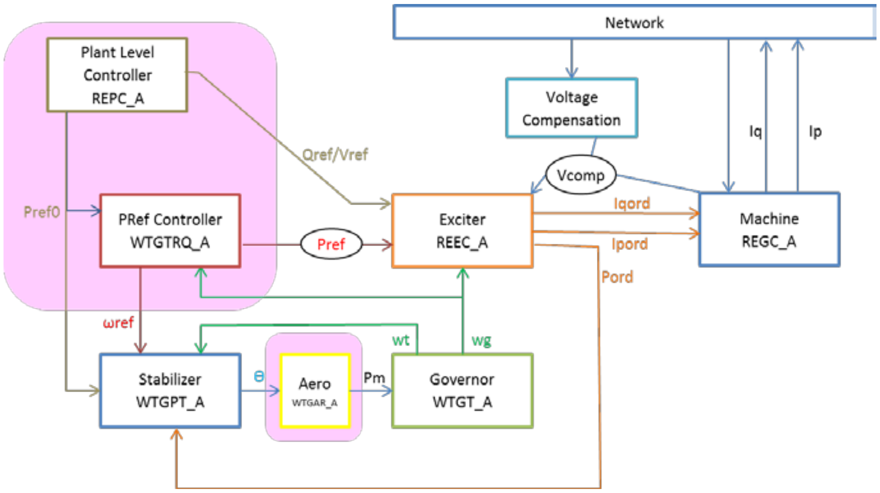
The modules then communicate with one another throw interface variables depending on their complexity. An example of a synchronous machine model and its interface variables are as follows



In the past several years additional modules have been added for generators

- Aerodynamic Model (recent addition for Type 3 Wind)
- Pref Controller (recent addition for Type 3 wind and LCFB1)
- Plant Controller (recent addition for renewable models)
- AGC Controller (Implemented in Version 19 of PowerWorld Simulator)

An example for a Type 3 wind turbine model with a few of these new modules highlighted in pink is shown in the following figure.



Load models have been largely stuck in the 1980s with really only a load characteristic module and a separate relay model. The MOTORW model in the 1990s was a step toward modularizing as the MOTORW model of GE PSLF applied to only a percentage of the load model. In this way, you could have 2 modules: Dynamic Model (MOTORW) and Algebraic model (WSCC). This made it so you were not required to split the load record in the power flow input data if you wanted a percentage of the load to be a dynamic model such as a motor and the rest default back to an algebraic model. Thus the state of the art as of 2010 included three modules for a load.

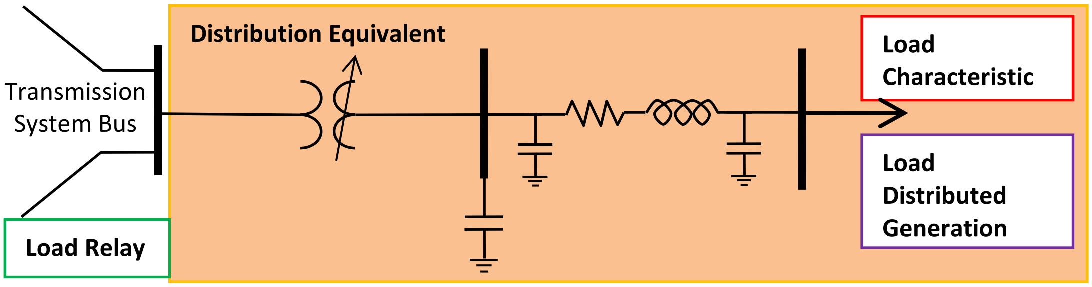
- Algebraic Model
- Dynamic Model
- Relay Model

These modules are very simple for software implementation however because these three modules really do not communicate with each other. The split between the algebraic and dynamic model occurs at the initialization of the stability model and no more communication occurs. The relay model doesn't really communicate with the other load models except to simply open them.

With the development of the Composite Load Model (CMPLDW) in WECC the distribution equivalent model was stuck inside the Dynamic model and a separate algebraic model was not done. This in part has led to the complication of this model as the only way to add additional features to a load model that appears at the end of a distribution equivalent was to shoe-horn that feature into CMPLDW. This was what is now occurring as the need to model distributed generation (Rooftop Solar PV) is realized.

It is now well past time to modularize the load model. At the very least we need to add two additional modules for each load.

- Distribution Equivalent Model
- Distributed Generation Model



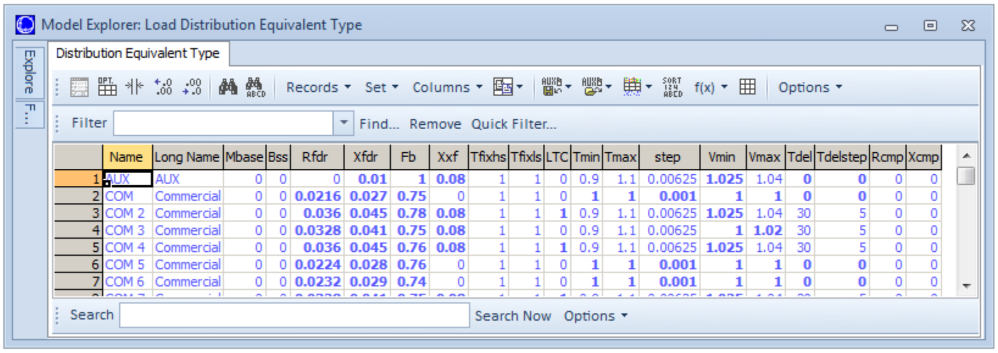
The trick with making this change is that the Distribution Equivalent presently designed for CMPLDW requires that you coordinate the initialization of the Distribution Equivalent Model, the Load Characteristic and the Load Distributed Generation Model.

To demonstrate how this should be done this document fully describes how loads are initialized in transient stability in PowerWorld Simulator. The Distribution Equivalent Model has been available since January 2013 in Version 17 of PowerWorld Simulator. The Distributed Generation Model module has been available since November 2015 in Version 19 of PowerWorld Simulator.

1.1 Load Distribution Equivalents

A Load Distribution Equivalent Types are special supplementary models that define an equivalent of the distribution system's transformer, capacitors, and feeder which connect the load to the transmission system. A Load Distribution Equivalent Type object in Simulator can be created *independently* of the transient load characteristic models. Each load record can optionally be assigned to a particular Load Distribution Equivalent Type and inherit its model. The design assumes that there will be a small number of Load Distribution Equivalent Types and that many different loads will be assigned to each Load Distribution Equivalent Type. Note that the Load Distribution Equivalent Type may be used with any load characteristic model.

The equivalent model and the parameters used for the Load Distribution Equivalent Types are the same as the first 17 parameters of the CMPLDW load characteristic model, along with an MVA base parameter. This is shown in the following figure.



	Name	Long Name	Mbase	Bss	Rfdr	Xfdr	Fb	Xxf	Tfixhs	Tfixls	LTC	Tmin	Tmax	step	Vmin	Vmax	Tdel	Tdelstep	Rcmp	Xcmp
1	AUX	AUX	0	0	0	0.01	1	0.08	1	1	0	0.9	1.1	0.00625	1.025	1.04	0	0	0	0
2	COM	Commercial	0	0	0.0216	0.027	0.75	0	1	1	0	1	1	0.001	1	1	0	0	0	0
3	COM 2	Commercial	0	0	0.036	0.045	0.78	0.08	1	1	1	0.9	1.1	0.00625	1.025	1.04	30	5	0	0
4	COM 3	Commercial	0	0	0.0328	0.041	0.75	0.08	1	1	0	0.9	1.1	0.00625	1	1.02	30	5	0	0
5	COM 4	Commercial	0	0	0.036	0.045	0.76	0.08	1	1	1	0.9	1.1	0.00625	1.025	1.04	30	5	0	0
6	COM 5	Commercial	0	0	0.0224	0.028	0.76	0	1	1	0	1	1	0.001	1	1	0	0	0	0
7	COM 6	Commercial	0	0	0.0232	0.029	0.74	0	1	1	0	1	1	0.001	1	1	0	0	0	0

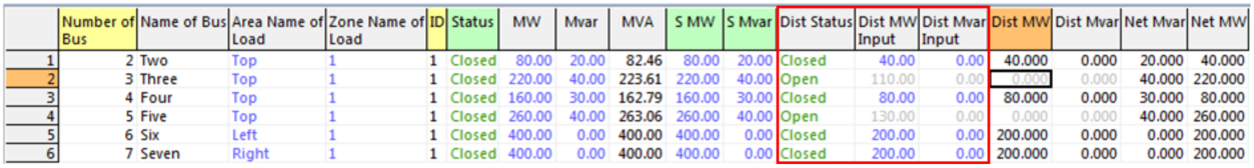
Essentially the CMPLDW load characteristic model has an embedded Load Distribution Equivalent Type specified as part of the model. When a CMPLDW load model is used within Simulator, then any Load Distribution Equivalent Type assigned to the respective load will be ignored and replaced with the embedded model of the CMPLDW.

When PowerWorld Simulator performs the transient stability numerical simulation including the load distribution equivalent, the load is moved internal to the numerical simulation. The actual power system model is NOT changed and the user-interface of Simulator never reflects this change. It only impacts the numerical simulation. The details of this initialization is described later.

1.2 Load Distributed Generation Model

The modeling of distributed generation as part of the load model effects more than just the transient stability simulation. It is also import for book-keeping purposed to allow the software user to specify the amount of distributed generation at each power flow data load record. This is accomplished by adding three additional fields to the power flow load record.

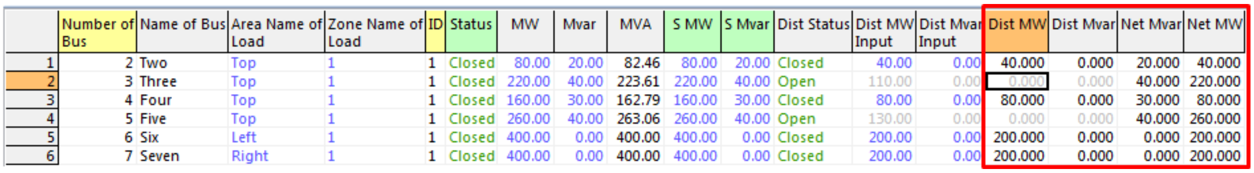
- **Dist MW Input**: the user entered MWs of distributed generation at the load
- **Dist Mvar Input**: the user entered Mvars of distributed generation at the load
- **Dist Status**: The status of the distributed generation (Open or Closed)



	Number of Bus	Name of Bus	Area Name of Load	Zone Name of Load	ID	Status	MW	Mvar	MVA	S MW	S Mvar	Dist Status	Dist MW Input	Dist Mvar Input	Dist MW	Dist Mvar	Net Mvar	Net MW
1	2	Two	Top	1	1	Closed	80.00	20.00	82.46	80.00	20.00	Closed	40.00	0.00	40.000	0.000	20.000	40.000
2	3	Three	Top	1	1	Closed	220.00	40.00	223.61	220.00	40.00	Open	110.00	0.00	0.000	0.000	40.000	220.000
3	4	Four	Top	1	1	Closed	160.00	30.00	162.79	160.00	30.00	Closed	80.00	0.00	80.000	0.000	30.000	80.000
4	5	Five	Top	1	1	Closed	260.00	40.00	263.06	260.00	40.00	Open	130.00	0.00	0.000	0.000	40.000	260.000
5	6	Six	Left	1	1	Closed	400.00	0.00	400.00	400.00	0.00	Closed	200.00	0.00	200.000	0.000	0.000	200.000
6	7	Seven	Right	1	1	Closed	400.00	0.00	400.00	400.00	0.00	Closed	200.00	0.00	200.000	0.000	0.000	200.000

Other informational fields should also then available to show as follows.

- **Dist MW**, **Dist Mvar** show the actual MWs and Mvars seen by the power flow solution. If **DistStatus** = Open, then these will show 0.0 for example. In addition, software tools have various methods for reducing constant power loads as the voltage falls in the power flow solution. The **Dist MW** and **Dist Mvar** follow these same rules.
- **Net MW** is equal to the subtraction of the Load's **MW** and **Dist MW** values. This represents what will be seen by the power flow solution algorithm.
- Similarly, **Net Mvar** = **Mvar** – **Dist Mvar**



	Number of Bus	Name of Bus	Area Name of Load	Zone Name of Load	ID	Status	MW	Mvar	MVA	S MW	S Mvar	Dist Status	Dist MW Input	Dist Mvar Input	Dist MW	Dist Mvar	Net Mvar	Net MW
1	2	Two	Top	1	1	Closed	80.00	20.00	82.46	80.00	20.00	Closed	40.00	0.00	40.000	0.000	20.000	40.000
2	3	Three	Top	1	1	Closed	220.00	40.00	223.61	220.00	40.00	Open	110.00	0.00	0.000	0.000	40.000	220.000
3	4	Four	Top	1	1	Closed	160.00	30.00	162.79	160.00	30.00	Closed	80.00	0.00	80.000	0.000	30.000	80.000
4	5	Five	Top	1	1	Closed	260.00	40.00	263.06	260.00	40.00	Open	130.00	0.00	0.000	0.000	40.000	260.000
5	6	Six	Left	1	1	Closed	400.00	0.00	400.00	400.00	0.00	Closed	200.00	0.00	200.000	0.000	0.000	200.000
6	7	Seven	Right	1	1	Closed	400.00	0.00	400.00	400.00	0.00	Closed	200.00	0.00	200.000	0.000	0.000	200.000

The treatment of the Load record's distributed generation MW and Mvar during a transient stability simulation is handled as follows.

- If a model exists that has distributed generation, then the portion of the load defined as distributed generation will be converted to a MW and Mvar value at the initial state and this will be used for treatment of the dynamic model for distributed generation in transient stability.
- Otherwise, if no dynamic model for distributed generation exists, then the entire **NetMW** and **NetMvar** will be treated according to the dynamic load characteristic models specified. Treatment of the distributed generation values as the normal dynamic load model would not be an error condition preventing a transient stability run, but it would be appropriate for the software to provide validation warnings to the user that a distributed generation dynamic model was expected but not found.

The details now become important regarding how to coordinate the initialization three separate models: of the load characteristic, the distributed generation, and the distribution equivalent. This is described in detail next.

2 Initialization Rules for Particular Load Models

2.1 LD1PAC Initialization

The compressor motor model is divided into two parts:

- Motor A – Those compressors that will *not* restart after stalling
- Motor B – Those compressors that will restart after stalling

Initialization of the MVABase and CompLF values are as follows.

- If Mbase > 0 then *MVAbase* = Mbase and the *CompLF* = 1.0
- If Mbase = 0 then *MVABase* = *Pinit* * *Pul* and *CompLF* = 1.0
- If Mbase < 0 then *CompLF* = abs(*Mbase*) and *MVABase* = *Pinit***Pul*/*CompLF*

The values of *Vstall* and *Vbrk* are adjusted according to the value of *LFAdj*.

- $Vstall = Vstall[1 + LFAdj(CompLF - 1)]$
- $Vbrk = Vbrk[1 + LFAdj(CompLF - 1)]$

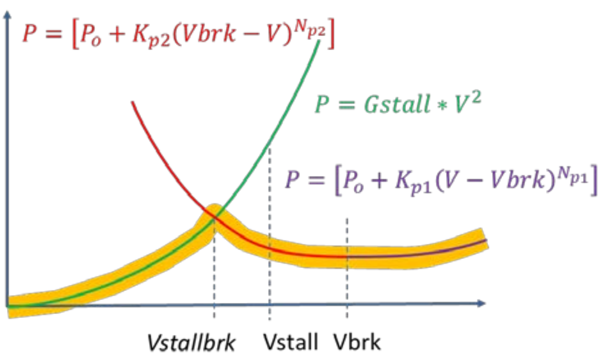
If V < *Vstall* for more than *Tstall* seconds, then the motor will enter the stall state and remain on the P and Q curves defined by $P = Gstall * V^2$ and $Q = Bstall * V^2$. For only the B motor, if V > *Vrst* for more than *Trst* second, then the motor will restart and automatically flip back to the curves as follows.

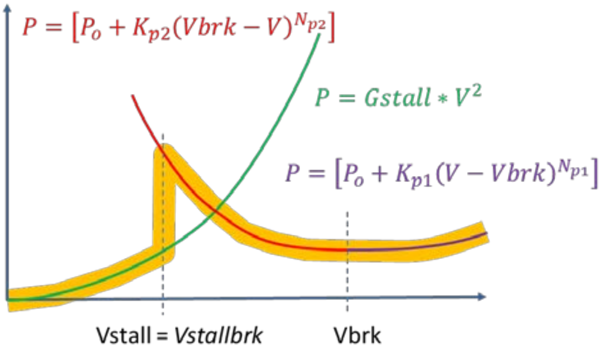
2.1.1 Algebraic Performance Model of LD1PAC

As long as the stall timer (Tstall) has not been exceeded, the motors are represented by algebraic equations, as follows:

- If *V* > *Vbrk* then
  - $P = \left[P_o + K_{p1}(V - Vbrk)^{N_{p1}}\right][1 + CmpKpf * \Delta f]$
  - $Q = \left[Q_o + K_{q1}(V - Vbrk)^{N_{q1}}\right][1 + CmpKqf * \Delta f]$
- If *V* < *Vbrk* and *V* > *Vstallbrk* then
  - $P = \left[P_o + K_{p2}(Vbrk - V)^{N_{p2}}\right][1 + CmpKpf * \Delta f]$
  - $Q = \left[Q_o + K_{q2}(Vbrk - V)^{N_{q2}}\right][1 + CmpKqf * \Delta f]$
- If *V* < *Vstallbrk* then
  - $P = Gstall * V^2$
  - $Q = Bstall * V^2$

This is generally depicted in the figure the figures below where *Vstallbrk*, *Po*, and *Qo* are calculated during initialization.





2.1.2 Calculation of P_o, Q_o, and $Vstallbrk$ for LD1PAC

From the initial terminal voltage, power and reactive power ($V_{init}, P_{init}, Q_{init}$), the following steps are taken to initialize this model

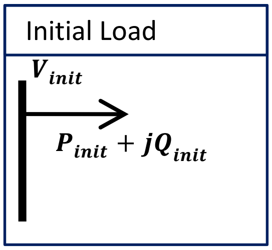
1. $P_o = P_{init} - K_{p1}(V_{init} - Vbrk)^{N_{p1}}$
2. *Vstallbrk* is automatically calculated by determing the intersection of the power stall curve and the power curve defined by *Kp2* and *Np2*. (PowerWorld determines this to a tolerance of 0.0001 per unit voltage). However, if this intersection is calculated as higher than *Vstall*, then instead set *Vstallbrk* = *Vstall*
3. $Q_o = P_{init}\left(\frac{\sqrt{1-CompPF^2}}{CompPF}\right) - K_{p1}(1.0 - Vbrk)^{N_{p1}}$

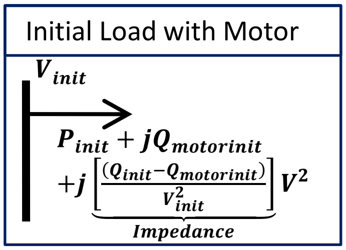
When initializing reactive power that is not accounted for by this initialization will be treated using the static load model assigned to the load.

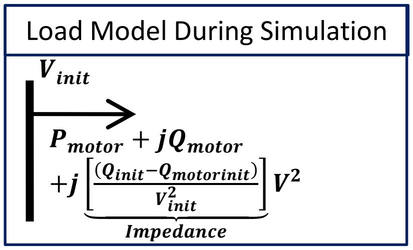
Another note is that during a particular time-step of numerical integration this value of P and Q is treated as constant impedance based on the previous time-steps value of P and Q.

2.2 Induction Motor Initialization

When initializing an induction motor model, the Mvar demand of the induction motor is dependent on the initial terminal voltage and MW demand. Thus, at initialization there will almost always be a mismatch between the induction motor's Mvar and the power flow initial condition Mvar. This mismatch is handled by adding shunt admittance to the bus to match the initial condition.



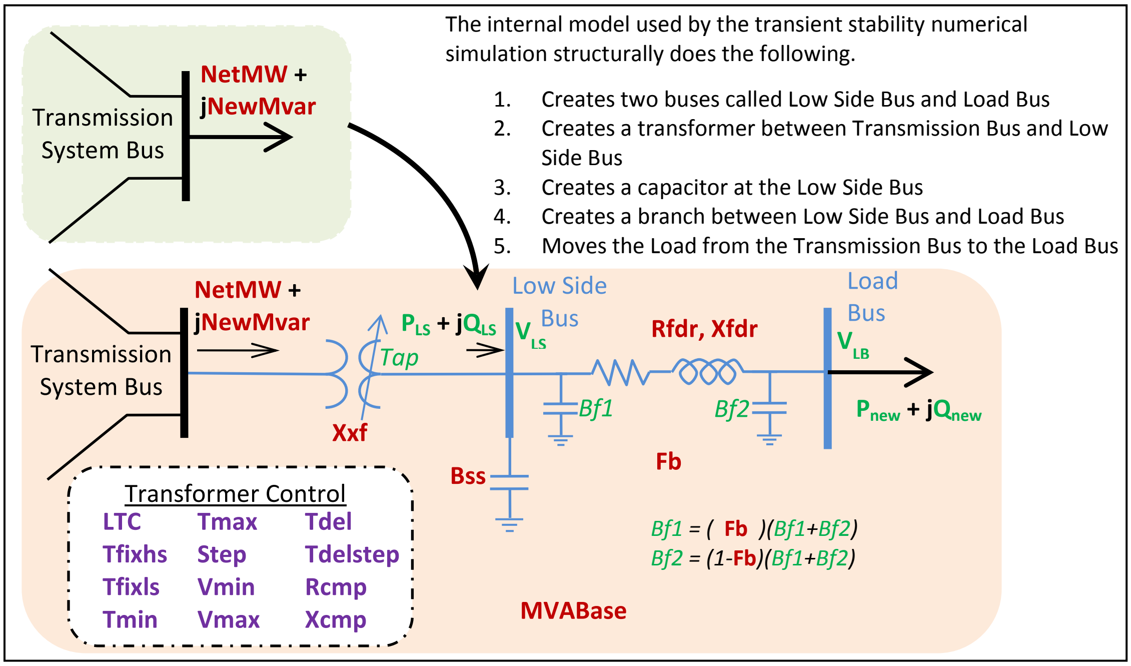

Initial Load with Motor
V_{init}
$P_{init} + jQ_{motorinit}$
$+j\underbrace{\left[\frac{(Q_{init}-Q_{motorinit})}{V_{init}^2}\right]}_{Impedance}V^2$


Load Model During Simulation
V_{init}
$P_{motor} + jQ_{motor}$
$+j\underbrace{\left[\frac{(Q_{init}-Q_{motorinit})}{V_{init}^2}\right]}_{Impedance}V^2$

Another note to be aware of with the CMPLDW model, the MOTORW model for double-cage motors is slightly different than the CIM5, CIM6, and MOTOR1 models. Thus trying to get them to match will not be successful.

3 Coordinated Initialization with Distribution Equivalent

Below is a picture of how the original load is moved to the end of the distribution equivalent without considering the distributed generation model. This will be considered at the end of this section.



The internal model used by the transient stability numerical simulation structurally does the following.

1. Creates two buses called Low Side Bus and Load Bus
2. Creates a transformer between Transmission Bus and Low Side Bus
3. Creates a capacitor at the Low Side Bus
4. Creates a branch between Low Side Bus and Load Bus
5. Moves the Load from the Transmission Bus to the Load Bus

3.1 Input Data Validation Check

Before initializing, validation is done to ensure that the input data represents a reasonable situation. The impedance parameters of the distribution equivalent can be given using an MVABase <= 0 indicating that the MVABase for these impedances are a multiple of the initial power (MW) of the load. This can cause problems if the initial load has a very bad power factor. For example, a load which is 1.2 MW and 30 Mvars is going to cause trouble because the impedances will be based on an MVABase proportional to 1.2, but the 30 Mvars may then exceed the maximum steady state power transformer across the distribution equivalent. PowerWorld Simulator does a validation check and will not permit the simulation to run if it fails. The check starts by determining the DistEquivMVABase as follows

If (**MVABase** > 0) then *DistEquivMVABase* = **MVABase**
Else If (**MVABase** < 0) then *DistEquivMVABase* = abs(**NetMW/MVABase)**
Else If (**MVABase** = 0) then *DistEquivMVABase* = **NetMW**/0.8;

The validation check simply estimates the losses assuming the current magnitude of the load is equal to the current magnitude that will be seen on the feeder, which is a reasonable approximation. If the I^2R losses exceed the value of the load then clearly something is wrong and an appropriate error message is shown requiring that the input data Q/P ratio be fixed.

$$\underbrace{\left[\frac{NewMW^2 + NetMvar^2}{Vpu^2}\right]}_{Current\ Squared} * \underbrace{\left[Rfdr\frac{SystemMVABase}{DistEquivMVABase}\right]}_{R\ on\ system\ MVABase} > NetMW$$

3.2 Initialization Process Overview

Initializing the distribution equivalent in coordination with the load models and load distributed generation models performs the following calculation to determine Tap, Bf1, Bf2, VLS, PLS, QLS, VLB, Pnew, and Qnew. Also, **Bss** may be modified as well.

1. A Distribution Equivalent MVA Base is determined based on the parameter **MVABase**.

   If (**MVABase** > 0) then *DistEquivMVABase* = **MVABase**
   If (**MVABase** < 0) then *DistEquivMVABase* = abs(**NetMW** / **MVABase)**
   If (**MVABase** = 0) then *DistEquivMVABase* = **NetMW** / 0.8;

2. The six impedance parameters (**Bss**, **Rfdr**, **Xfdr**, **Xxf**, **Rcmp**, **Xcmp**) of the Distribution Equivalent Type are assumed to be on this *DistEquivMVABase* and are converted the to the System MVA Base.

   Xxf = **Xxf** * *SystemMVABase*/ *DistEquivMVABase*.
   Rfdr = **Rfdr** * *SystemMVABase*/ *DistEquivMVABase*.
   Xfdr = **Xfdr** * *SystemMVABase*/ *DistEquivMVABase*.
   Rcmp = **Rcmp** * *SystemMVABase*/ *DistEquivMVABase*.
   Xcmp = **Xcmp** * *SystemMVABase*/ *DistEquivMVABase*.
   Bss = **Bss** / *SystemMVABase** *DistEquivMVABase*.

3. Transformer Taps and impedances are converted to the System MVA Base based on the fixed taps. (Note that the variable tab is assumed to be at the Low Side Bus).

   Xxf = **Xxf** * (**Tfixhs**)^2
   Step = **Step/Tfixhs**
   Tmin = (**Tmin** + **Tfixls** - 1)/**Tfixhs**
   Tmax = (**Tmax** + **Tfixls** - 1)/**Tfixhs**

4. The transformer tap ratio (Tap) is assumed to be set such that the voltage at the Low Side Bus is equal to **(Vmin + Vmax) /2**. (See Section 3.2.1). Tap is then rounded to the nearest discrete step and brought back within the **Tmin** - **Tmax** range if necessary.

5. Using the impedances and tap ratio calculated in the previous step, the Low Side Bus voltage (VLS) is calculated exactly and the resulting flow on the Low Side Bus is calculated exactly (P_{LS} + jQ_{LS}). This is done by using equations in Section 3.2.2 with the transformer branch admittances and results of

   a. VLS = $e_m + jf_m$
   b. P_{LS} + jQ_{LS} $= -(P_{mk} + jQ_{mk})$

6. We now initialize *Bf1* and *Bf2* both as zero with the remainder of this routine necessary to determine these values.

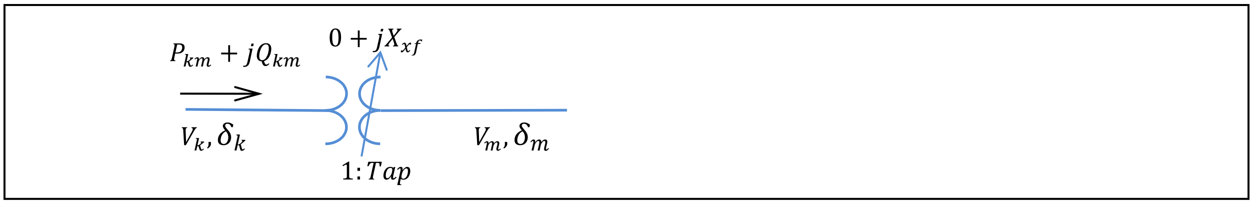
7. Using present values of *Bf1* and *Bf2* estimate both the Load Bus Voltage and the resulting flow reaching the Load Bus is calculated as P_{new} + jQ_{new}. Again this is done by using equations in Section 3.2.2 with the feeder branch admittances and results of

   a. VLB = $e_m + jf_m$
   b. P_{new} + jQ_{new} $= -(P_{mk} + jQ_{mk})$

8. If the magnitude of Load Bus voltage (VLB) falls below 0.95 per unit, then the feeder impedances **Rfdr** and **Xfdr** are reduced by a factor that results in a Load Bus Voltage of 0.95 per unit (See Section 3.2.3 for details on this calculation).
9. Using the values of VLB and P_{new} + jQ_{new}, initialize the dynamic load characteristic model
    a. For load models such as motors, the Mvar of the load is dependent on the MW of the load. This means that at initialization there will almost always be a mismatch between what the stability load model Mvar initializes to and what the power flow initial condition Mvar is specified as. This mismatch is handled by adding a shunt impedance to the bus to match the initial condition. Call these Mvars the ExtraMvars
    b. When including a Load Distributed Generation model, initialize the Load Characteristic Models using values (P_{new} + **DistMW**) + j(Q_{new}+ **DistMvar**)
10. If there is no feeder because **Rfdr** and **Xfdr** were essentially zero, then Exit Initialization and leave load characteristic model initialization as normal
11. Any extra Mvars which are calculated as part of the dynamic load characteristic model described in step 9a are now allocated to feeder shunt values *Bf1* and *Bf2* according to the parameter **Fb**.
    a. If the (ExtraMvars < 1E-4 per unit) OR (**Fb** < 0.001), then stick them all at the Load Bus
        i. *Bf2* = *Bf2* + ExtraMvars/(VLB^2)
        ii. *ExitShortly* = True
    b. Else
        i. *Bf1* = *Bf1* + **Fb***ExtraMvars/(VLS^2)
        ii. *Bf2* = *Bf2* + (1 – **Fb**)*ExtraMvars/(VLB^2)
        iii. *ExitShortly* = False
12. If *Bf1* and *Bf2* are negative and **Bss** > 0 then reduce **Bss** toward zero to cancel out *Bf1* and *Bf2*. This is done with the following logic.
    a. *ExitShortly* = False
    b. Bf1var = *Bf1**VLS^2
    c. Bf2var = *Bf2**VLB^2
    d. Bssvar = **Bss***VLS^2
    e. If abs(Bf1var + Bf2Var) > Bssvar then // Set Bss=0 and reduce magnitude of *Bf1* and *Bf2*
        i. tempVar = BssVar
        ii. **Bss** = 0
    f. If **Fb** = 1 then // Reduce Bss by Bf1Var and reallocate vars to *Bf1* and *Bf2*
        i. tempVar = abs(Bf1Var)
        ii. **Bss** = **Bss** – tempVar / VLS^2
    g. Else // attempt to push *Bf2* toward zero
        i. tempVar = abs(Bf2var) / (1 – **Fb**)
        ii. **Bss** = **Bss** – tempVar / VLS^2
    h. *Bf1* = *Bf1* + **Fb** * tempVar / VLS^2
    i. *Bf2* = *Bf2* + (1 – **Fb**) * tempVar / VLS^2
13. If (ExitShortly) then Exit Initialization
    Else go back to Step 7 and repeat

3.2.1 Calculation of Transformer Tap Ratio

Define a transformer which has the impedances, tap ratio, and the variables defined in the following figure.



The transformer tap ratio on the distribution equivalent is automatically set so that the voltage magnitude at the Low Side Bus is equal to the average of the input parameters **Vmin** and **Vmax**. To do this we use the equations of flow across the lossless transformer which are as follows.

$$P_{km} = \frac{V_k V_m}{Tap X_{xf}} \sin(\delta_k - \delta_m) \text{ and } Q_{km} = \frac{V_k^2}{X_{xf}} - \frac{V_k V_m}{Tap X_{xf}} \cos(\delta_k - \delta_m)$$

From this we can calculate *Tap* exactly as follows.

From P_{km} equation we have

$$\sin(\delta_k - \delta_m) = \frac{Tap X_{xf} P_{km}}{V_k V_m}$$

which means

$$\cos(\delta_k - \delta_m) = \frac{\sqrt{(V_k V_m)^2 - \left(Tap X_{xf} P_{km}\right)^2}}{V_k V_m}$$

Substitute to Q_{km} equation gives

$$Q_{km} = \frac{V_k^2}{X_{xf}} - \frac{V_k V_m}{Tap X_{xf}} \left(\frac{\sqrt{(V_k V_m)^2 - \left(Tap X_{xf} P_{km}\right)^2}}{V_k V_m} \right)$$

Which can be simplified to

$$Q_{km} = \frac{V_k^2}{X_{xf}} - \left(\frac{\sqrt{(V_k V_m)^2 - \left(Tap X_{xf} P_{km}\right)^2}}{Tap X_{xf}} \right)$$

Multiply by $Tap X_{xf}$ to get

$$Q_{km} Tap X_{xf} - V_k^2 Tap = -\sqrt{(V_k V_m)^2 - \left(Tap X_{xf} P_{km}\right)^2}$$

Square both side

$$(Tap)^2 \left(Q_{km} X_{xf} - V_k^2\right)^2 = (V_k V_m)^2 - \left(Tap X_{xf} P_{km}\right)^2$$

Group $(Tap)^2$ terms

$$(Tap)^2 \left[\left(Q_{km} X_{xf} - V_k^2\right)^2 + \left(X_{xf} P_{km}\right)^2 \right] = (V_k V_m)^2$$

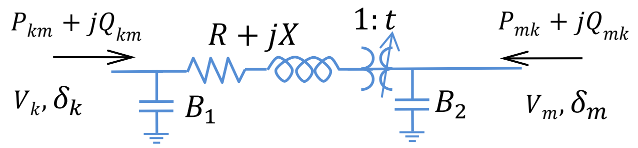
The solution for Tap is then given precisely b the following.

$$Tap = \sqrt{\frac{(V_k V_m)^2}{\left(Q_{km} X_{xf} - V_k^2\right)^2 + \left(X_{xf} P_{km}\right)^2}}$$

We know from the initial condition and the input parameters P_{km}, Q_{km}, V_k, and X_{xf}. Then we use $V_m = \frac{V_{min} + V_{max}}{2}$ to calculate the required Tap.

3.2.2 Calculation of Far Bus Complex Voltage and PQ Flows

Define a branch which has the impedances, admittances, and tap ratio and the variables defined in the following figure.



$$g_{kk} = \frac{R}{(R^2+X^2)} \qquad b_{kk} = \frac{-X}{(R^2+X^2)} + B_1 \qquad g_{km} = g_{mk} = \frac{-R}{(R^2+X^2)t}$$

$$g_{mm} = \frac{R}{(R^2+X^2)t^2} \qquad b_{mm} = \frac{-X}{(R^2+X^2)t^2} + B_2 \qquad b_{km} = b_{mk} = \frac{X}{(R^2+X^2)t}$$

Given the branch impedance/admittances and tap ratio and the initial flow on the near end (k) of the branch we can calculate the far end complex voltage exactly using the equation for P and Q on the branch as follows.

$$P_{km} = +e_k^2 g_{kk} + e_k e_m g_{km} - e_k f_m b_{km} + f_k^2 g_{kk} + f_k f_m g_{km} + f_k e_m b_{km}$$
$$Q_{km} = -f_k^2 b_{kk} + f_k e_m g_{km} - f_k f_m b_{km} - e_k^2 b_{kk} - e_k f_m g_{km} - e_k e_m b_{km}$$

In these calculations we will also know the P_{km} and Q_{km}, and the voltage at the sending bus $(e_k + jf_k)$ so we have two equations and the two unknowns (e_m, f_m) which we can group together terms as

$$e_m[e_k g_{km} + f_k b_{km}] + f_m[-e_k b_{km} + f_k g_{km}] = P_{km} - e_k^2 g_{kk} - f_k^2 g_{kk}$$
$$e_m[f_k g_{km} - e_k b_{km}] + f_m[-f_k b_{km} - e_k g_{km}] = Q_{km} + f_k^2 b_{kk} + e_k^2 b_{kk}$$

These are linearly equations which can and solved in the customary way using 2x2 matrices.

$$\begin{bmatrix} e_k g_{km} + f_k b_{km} & | & -e_k b_{km} + f_k g_{km} \\ f_k g_{km} - e_k b_{km} & | & -f_k b_{km} - e_k g_{km} \end{bmatrix} \begin{bmatrix} e_m \\ f_m \end{bmatrix} = \begin{bmatrix} P_{km} - e_k^2 g_{kk} - f_k^2 g_{kk} \\ Q_{km} + f_k^2 b_{kk} + e_k^2 b_{kk} \end{bmatrix}$$

With the voltage e_m and f_m found, the flows can then be calculated using the following.

$$P_{mk} = +e_m^2 g_{mm} + e_m e_k g_{mk} - e_m f_k b_{mk} + f_m^2 g_{mm} + f_m f_k g_{mk} + f_m e_k b_{mk}$$
$$Q_{mk} = -f_{km}^2 b_{mm} + f_m e_k g_{mk} - f_m f_k b_{mk} - e_m^2 b_{mm} - e_m f_k g_{mk} - e_m e_k b_{mk}$$

For our Transformer branch the admittance parameters are as follows.

$$g_{kk} = 0 \qquad b_{kk} = -\frac{1}{X_{xf}} \qquad g_{km} = g_{mk} = 0$$

$$g_{mm} = 0 \qquad b_{mm} = -\frac{1}{X_{xf}Tap^2} \qquad b_{km} = b_{mk} = \frac{1}{X_{xf}Tap}$$

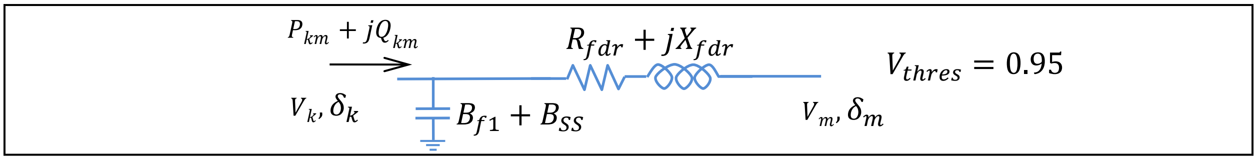
For our Feeder branch the admittance parameters are as follows. (Note: the capacitors are lumped into the feeder admittances for calculation purposes here.)

$$g_{kk} = \frac{R_{dfr}}{R_{fdr}^2 + X_{fdr}^2} \qquad b_{kk} = \frac{-X_{dfr}}{R_{fdr}^2 + X_{fdr}^2} + B_{f1} + B_{ss} \qquad g_{km} = g_{mk} = \frac{-R_{dfr}}{R_{fdr}^2 + X_{fdr}^2}$$

$$g_{mm} = \frac{R_{dfr}}{R_{fdr}^2 + X_{fdr}^2} \qquad b_{mm} = \frac{-X_{dfr}}{R_{fdr}^2 + X_{fdr}^2} + B_{f2} \qquad b_{km} = b_{mk} = \frac{X_{dfr}}{R_{fdr}^2 + X_{fdr}^2}$$

3.2.3 Calculation of Far Bus Complex Voltage and PQ Flows with Voltage Constraint

Define a branch which has the impedances, admittances, and the variables defined in the following figure.



A further constraint is added when calculating the voltage at the load bus end of the feeder branch. In this calculation we require that the far end voltage magnitude must be at least $V_{thres} = 0.95$. If the voltage falls below this minimum, we will assume that the feeder impedance is reduced by a factor of $MULT$ in order to bring the voltage magnitude up. To calculate this we lump the shunt terms (B_{SS} and B_{f1}) at the Low Side Bus into the Q flow on that side. This means that the admittance parameters in the equation are simplified to the following. (Note: B_{f2} is not used in this calculation).

$$g_{kk} = \frac{R_{dfr}}{R_{fdr}^2+X_{fdr}^2} \qquad b_{kk} = \frac{-X_{dfr}}{R_{fdr}^2+X_{fdr}^2} \qquad g_{km} = \frac{-R_{dfr}}{R_{fdr}^2+X_{fdr}^2} \qquad b_{km} = \frac{X_{dfr}}{R_{fdr}^2+X_{fdr}^2}$$

If we divide both R_{dfr} and X_{dfr} by a factor $MULT$, then this means that all the admittance terms above are multiplied by $MULT$. This gives us the following equations for the line flows to which we add the additional equation that enforces the minimum voltage.

$$P_{km} - MULT\begin{pmatrix} +e_k^2 g_{kk} + e_k e_m g_{km} - e_k f_m b_{km} \\ +f_k^2 g_{kk} + f_k f_m g_{km} + f_k e_m b_{km} \end{pmatrix} = 0$$

$$\left[Q_{km} + (B_{SS} + B_{f1})V_k^2\right] - MULT\begin{pmatrix} -f_k^2 b_{kk} + f_k e_m g_{km} - f_k f_m b_{km} \\ -e_k^2 b_{kk} - e_k f_m g_{km} - e_k e_m b_{km} \end{pmatrix} = 0$$

$$e_m^2 + f_m^2 - V_{thres}^2 = 0$$

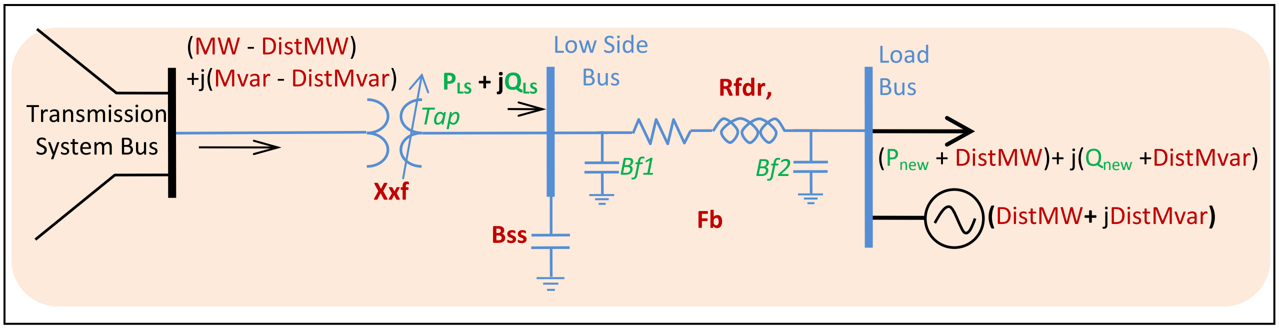
This gives us three equations and three unknown variables of e_m , f_m, and $MULT$. These are non-linear equations which can be solved using Newton's method. The Jacobian matrix for this use of Newton's method is as follows.

$$\begin{bmatrix} -MULT\begin{pmatrix} e_k g_{km} \\ +f_k b_{km} \end{pmatrix} & -MULT\begin{pmatrix} -e_k b_{km} \\ +f_k g_{km} \end{pmatrix} & -\begin{pmatrix} +e_k^2 g_{kk} + e_k e_m g_{km} - e_k f_m b_{km} \\ +f_k^2 g_{kk} + f_k f_m g_{km} + f_k e_m b_{km} \end{pmatrix} \\ -MULT\begin{pmatrix} f_k g_{km} \\ -e_k b_{km} \end{pmatrix} & -MULT\begin{pmatrix} -f_k b_{km} \\ -e g_{km} \end{pmatrix} & -\begin{pmatrix} -f_k^2 b_{kk} + f_k e_m g_{km} - f_k f_m b_{km} \\ -e_k^2 b_{kk} - e_k f_m g_{km} - e_k e_m b_{km} \end{pmatrix} \\ 2e_m & 2f_m & 0 \end{bmatrix}$$

We iterate these equations using Newton’s method until the mismatch of the P and Q equations are less than 1E-5 per unit and the voltage mismatch equation is less than 1E-8.

3.2.4 Treatment of Load Distributed Generation MOdel

To perform this initialization it is assumed that the portion applying to the distributed generation is moved directly to the Load Bus without being modified by losses. Thus a better illustration is as follows.



The initialization of the distribution equivalent is performed the same way as described earlier but net load at the transmission bus is Load – DistGen instead. The value of P_{new} and Q_{new} is done the same, however during the processing when the loads characteristic models are initialized, the MW and Mvar at the Load bus is split between the load characteristic and the load distributed generation model as follows.

- (P_{new} + DistMW)+ j(Q_{new} +DistMvar) is applied to the load characteristic model
- DistMW + jDistMvar is applied to the distributed generation model