

Transient Stability Analysis and Software Design



PowerWorld Corporation
2001 S. First St, Suite 203
Champaign, IL 61820
<http://www.powerworld.com>
info@powerworld.com
217 384 6330



PowerWorld
Corporation

Transient Stability

Multimachine Simulations



- How do we put these all together
 - Generators: (Machines, Exciters, Governors)
 - Loads: Algebraic and Dynamic
- Later we'll add in new model types such as stabilizers, loads and wind turbines
- By way of history, prior to digital computers, network analyzers were used for system stability studies as far back as the 1930's (perhaps earlier)
 - For example see, J.D. Holm, "Stability Study of A-C Power Transmission Systems," AIEE Transactions, vol. 61, 1942, pp. 893-905
- Digital approaches started appearing in the 1960's

Transient Stability

Multimachine Simulations



- The general structure is as a set of differential-algebraic equations
 - Differential equations describe the behavior of the machines (and the loads and other dynamic devices)
 - Algebraic equations representing the network constraints

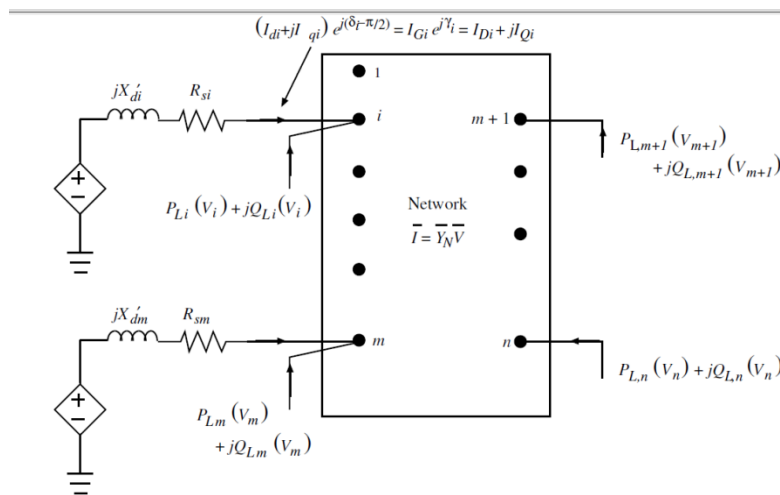


Figure 7.2: Interconnection of synchronous machine dynamic circuit and the rest of the network

In EMTP applications the transmission line delays decouple the machines; in transient stability they are assumed to be coupled together by the algebraic network equations

Power Flow vs. Transient Stability



- Transient stability is used to determine whether following a contingency the power system returns to a steady-state operating point
 - Goal is to solve a set of differential and algebraic equations,
 - $dx/dt = f(x,y,u)$ [y variables are bus voltage and angle, u = setpoints]
 - $g(x,y) = 0$ [x variables are dynamic state variables]
 - Starts in steady-state, and hopefully returns to a new steady-state.
 - Models reflect the transient stability time frame (from about 10 milliseconds up to dozens of seconds)
 - Slow Values → Treat as constants
 - Some values assumed to be slow enough to hold constant (LTC tap changing)
 - Ultra Fast States → Treat as algebraic relationships
 - Synchronous machine stator current dynamics, voltage source converter dynamics (DC transmission, portions of wind turbine models)

Algebraic Constraints



- The $\mathbf{g}$ vector of algebraic constraints is similar to the power flow equations, but usually rather than formulating the problem like in the power flow as real and reactive power balance equations, it is formulated in the current balance form

$$\mathbf{I}(\mathbf{x}, \mathbf{V}) = \mathbf{Y} \mathbf{V} \quad \text{or} \quad \mathbf{Y} \mathbf{V} - \mathbf{I}(\mathbf{x}, \mathbf{V}) = \mathbf{0}$$

where $\mathbf{Y}$ is the $n \times n$ bus admittance matrix ($\mathbf{Y} = \mathbf{G} + j\mathbf{B}$), $\mathbf{V}$ is the complex vector of the bus voltages, and $\mathbf{I}$ is the complex vector of the bus current injections

Simplest cases can have $\mathbf{I}$ independent of $\mathbf{x}$ and $\mathbf{V}$, allowing a direct solution; otherwise we need to iterate

Why Not Use the Power Flow Equations?

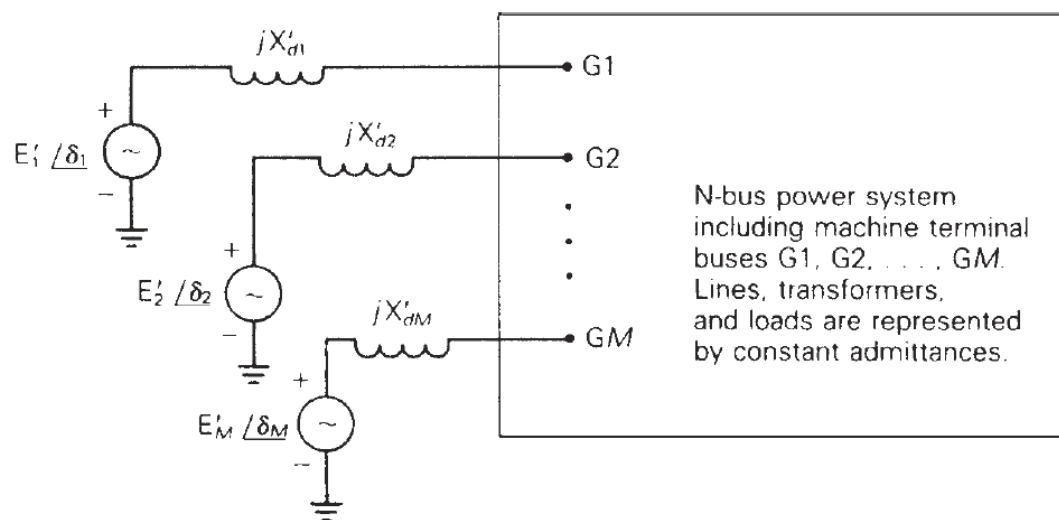


- The power flow equations were ultimately derived from $\mathbf{I}(\mathbf{x}, \mathbf{V}) = \mathbf{Y} \mathbf{V}$
- However, the power form was used in the power flow primarily because
 - For the generators the real power output is known and either the voltage setpoint (i.e., if a PV bus) or the reactive power output
 - In the quasi-steady state power flow time frame the loads can often be well approximated as constant power
 - The constant frequency assumption requires a slack bus
- These assumptions do not hold for transient stability

Algebraic Equations for Classical Model



- Coupling between the synchronous machine models and the network constraints we discussed earlier



We'll discuss network reference frame only here

Initially assume constant admittance loads

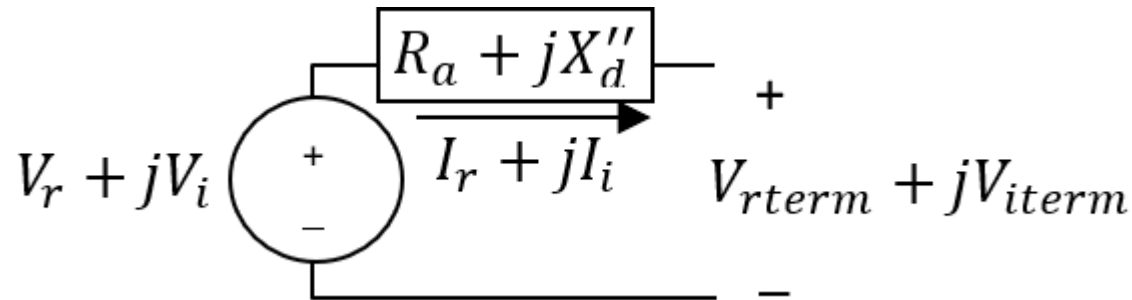
Must convert to that

Network Interface

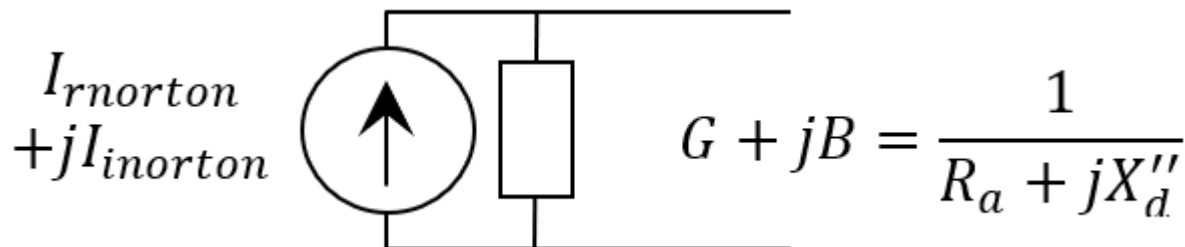


- Synchronous Machine Model we a Thevenin circuit (at least for simple models like GENROU and GENSAL)

$$(V_r + jV_i) = (1 + \Delta\omega_{pu})(E_d'' + jE_q'')e^{+j(\delta - \frac{\pi}{2})}$$



- Must convert this to a Norton Equivalent instead



Algebraic Equations for Classical Model



- Replace the internal voltages and their impedances by their Norton Equivalent

$$\bar{I}_i = \frac{V_r + jV_i}{R_{s,i} + jX'_{d,i}}, \quad Y_i = \frac{1}{R_{s,i} + jX''_{d,i}}$$

- Current injections at the non-generator buses are zero since the constant impedance loads are included in Y
 - We'll modify this later when we talk about dynamic loads
- The algebraic constraints are then $I - Y V = 0$

Numerical Solution



- There are two main approaches for solving

$$\frac{dx}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u})$$
$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y})$$

- Partitioned-explicit

- Solve the differential and algebraic equations separately (alternating between the two) using an explicit integration approach

- Simultaneous-implicit

- Solve the differential and algebraic equations together using an implicit integration approach

Transient Stability Solution Process



- Commercial Software typically uses explicit integration
 - Start with solved Power Flow (similar to $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$)
 - The “Initial” or “Boundary” Conditions
 - Gives $\mathbf{y}_0 \rightarrow$ initial values for all the $\mathbf{y}$ variables (Voltage, Angle)
 - Derivatives at “steady state” are zero by definition, thus use the equation $\frac{d\mathbf{x}}{dt} = \mathbf{0} = \mathbf{f}(\mathbf{x}_0, \mathbf{y}_0)$ and solve for $\mathbf{x}_0$
 - Use numerical integration to simulate going forward in time
 - Many ways to do this (half of an entire class in graduate school was dedicated to this), but let’s just consider the Forward Euler’s method because it’s simple to explain
 - Important user choice is the Time Step Δt that will be used in numerical integration

Transient Stability Numerical Integration



- Already have initial x_0 and y_0
- Initialize $t = 0$ and $k = 0$
- Choose a Time Step Δt that will be used in numerical integration
- Calculate new x : $x_{k+1} = x_k + \Delta t * f(x_k, y_k)$
 - called the “integration time step”
 - Many ways to do this. This is a “Forward Euler”
- Calculate new y : Solve $g(x_{k+1}, y_{k+1}) = 0$
 - Called solving the “network boundary equations”
 - Very similar to power flow equation, but use complex currents instead
- Increment time t : $t_{k+1} = t_k + \Delta t$
- Increment k : $k = k + 1$
- Go back to 1 repeat

Explicit Integration Methods



- Wide variety of explicit integration methods
 - We considered Forward Euler, Runge-Kutta, Adams-Bashforth
- Here we will just consider Euler's, which is easy to explain, and a second order Runge-Kutta, which is commonly used

Forward Euler



- Recall the Forward Euler approach is approximate

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \frac{d\mathbf{x}}{dt} \text{ as } \frac{\Delta\mathbf{x}}{\Delta t}$$

Then

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \mathbf{f}(\mathbf{x}(t))$$

- Error with Euler's varies with the square of the time step

Forward Euler



- Recall the Forward Euler approach is approximate

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) = \frac{d\mathbf{x}}{dt} \text{ as } \frac{\Delta\mathbf{x}}{\Delta t}$$

Then

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \mathbf{f}(\mathbf{x}(t))$$

- Error with Euler's varies with the square of the time step

Runga-Kutta 2nd Order



- With RK2 the first part of the time step is the same as Euler's, that is solving the network equations with

$$\mathbf{x}(t + \Delta t)^{(1)} = \mathbf{x}(t) + \mathbf{k}_1 = \mathbf{x}(t) + \Delta t \mathbf{f}(\mathbf{x}(t))$$

- Then calculate k2 and get a final value for $\mathbf{x}(t + \Delta t)$

$$\mathbf{k}_2 = \Delta t \mathbf{f}(\mathbf{x}(t) + \mathbf{k}_1)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2)$$

- Finally solve the network equations using the final value for $\mathbf{x}(t + \Delta t)$

Angle Reference

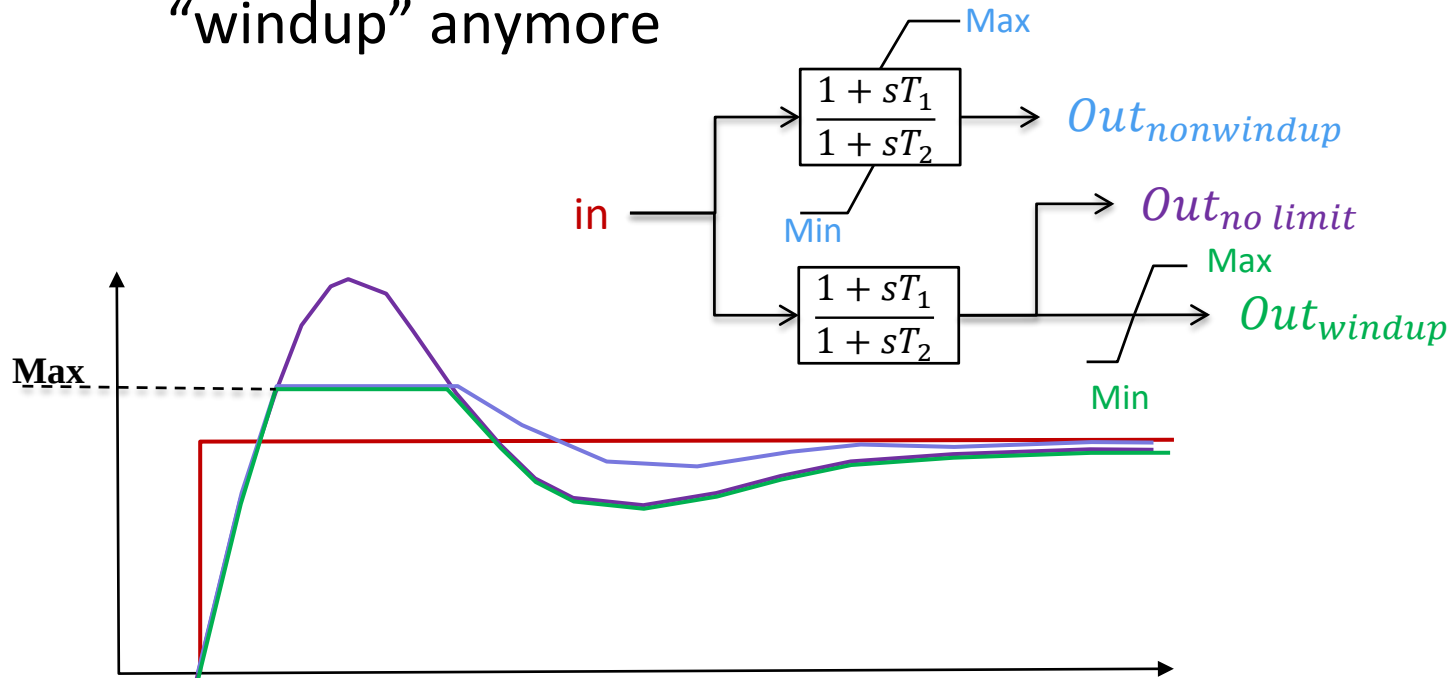


- The initial angles are given by the angles from the power flow, which are based on the slack bus's angle
- As presented the transient stability angles are with respect to a synchronous reference frame
 - Sometimes this is fine, such as for either shorter studies, or ones in which there is little speed variation
 - Oftentimes this is not best since when the frequencies are not nominal, the angles shift from the reference frame
- Other reference frames can be used, such as with respect to a particular generator's value, which mimics the power flow approach; the selected reference has no impact on the solution

Other Tricks in Integration



- Non-windup versus Windup Limits
 - Think an old “windup” clock with a spring. Eventually, the spring reaches a point where it can no longer “windup” anymore

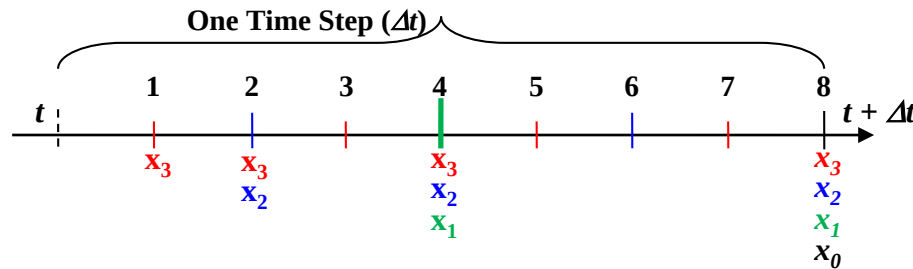


- A “non-windup” limit must be enforced between step 1 and 2 so that the state never goes past its limit

Sub-Interval Integration (Also called “multi-rate”)



- Chop-up the time step into pieces so that you can use a smaller time step only for states that require this.



Subinterval	x_3 updates	x_2 updates	x_1 updates	x_0 updates
1	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	Assume constant	Assume constant	Assume constant
2	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	$x_2 = x_2 + \frac{\Delta t}{4} * f(x, y)$	Assume constant	Assume constant
3	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	Assume constant	Assume constant	Assume constant
4	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	$x_2 = x_2 + \frac{\Delta t}{4} * f(x, y)$	$x_1 = x_1 + \frac{\Delta t}{2} * f(x, y)$	Assume constant
5	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	Assume constant	Assume constant	Assume constant
6	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	$x_2 = x_2 + \frac{\Delta t}{4} * f(x, y)$	Assume constant	Assume constant
7	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	Assume constant	Assume constant	Assume constant
8	$x_3 = x_3 + \frac{\Delta t}{8} * f(x, y)$	$x_2 = x_2 + \frac{\Delta t}{4} * f(x, y)$	$x_1 = x_1 + \frac{\Delta t}{2} * f(x, y)$	$x_0 = x_0 + \frac{\Delta t}{1} * f(x, y)$

Network Equation Solution



- If you read text-books you'll see a lot of constant impedance loads!
 - Works great! You reduce the entire network down to ONLY the generator buses with an equivalent impedance network between these buses
 - Problem though: That's not how our loads behave!
- Hence a nonlinear solution with Newton's method is used
- We'll generalize the dependence on the algebraic variables, replacing V by y since they may include other values beyond just the bus voltages

Nonlinear Network Equations



- Just like in the power flow, the complex equations are rewritten, here as a real current and a reactive current

$$YV - I(x, y) = 0$$

- The values for bus i are

$$g_{Di}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^n (G_{ik} V_{Dk} - B_{ik} V_{Qk}) - I_{NDi} = 0$$

$$g_{Qi}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^n (G_{ik} V_{Qk} + B_{ik} V_{Dk}) - I_{NQi} = 0$$

This is a rectangular formulation; we also could have written the equations in polar form

- For each bus we add two new variables and two new equations
- If an infinite bus is modeled then its variables and equations are omitted since its voltage is fixed

Nonlinear Network Equations



- The network variables and equations are then

$$\mathbf{y} = \begin{bmatrix} V_{D1} \\ V_{Q1} \\ V_{D2} \\ \vdots \\ V_{Dn} \\ V_{Qn} \end{bmatrix} \quad \mathbf{g}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \sum_{k=1}^n (G_{1k} V_{Dk} - B_{1k} V_{QK}) - I_{ND1}(\mathbf{x}, \mathbf{y}) = 0 \\ \sum_{k=1}^n (G_{1k} V_{Qk} + B_{1k} V_{DK}) - I_{NQ1}(\mathbf{x}, \mathbf{y}) = 0 \\ \sum_{k=1}^n (G_{2k} V_{Dk} - B_{2k} V_{QK}) - I_{ND2}(\mathbf{x}, \mathbf{y}) = 0 \\ \vdots \\ \sum_{k=1}^n (G_{nk} V_{Dk} - B_{nk} V_{QK}) - I_{NDn}(\mathbf{x}, \mathbf{y}) = 0 \\ \sum_{k=1}^n (G_{nk} V_{Qk} + B_{nk} V_{DK}) - I_{NQn}(\mathbf{x}, \mathbf{y}) = 0 \end{bmatrix}$$

Nonlinear Network Equation

Newton Solution



The network equations are solved using a similar procedure to that of the Newton–Raphson power flow

Set $v = 0$; make an initial guess of $\mathbf{y}$, $\mathbf{y}^{(v)}$

While $\|\mathbf{g}(\mathbf{y}^{(v)})\| > \varepsilon$ Do

$$\mathbf{y}^{(v+1)} = \mathbf{y}^{(v)} - \mathbf{J}(\mathbf{y}^{(v)})^{-1} \mathbf{g}(\mathbf{y}^{(v)})$$

$$v = v + 1$$

End While

Network Equation Jacobian Matrix



- The most computationally intensive part of the algorithm is determining and factoring the Jacobian matrix, $J(\mathbf{y})$

$$J(\mathbf{y}) = \begin{bmatrix} \frac{\partial g_{D1}(\mathbf{x}, \mathbf{y})}{\partial V_{D1}} & \frac{\partial g_{D1}(\mathbf{x}, \mathbf{y})}{\partial V_{Q1}} & \dots & \frac{\partial g_{D1}(\mathbf{x}, \mathbf{y})}{\partial V_{Qn}} \\ \frac{\partial g_{Q1}(\mathbf{x}, \mathbf{y})}{\partial V_{D1}} & \frac{\partial g_{Q1}(\mathbf{x}, \mathbf{y})}{\partial V_{Q1}} & \dots & \frac{\partial g_{Q1}(\mathbf{x}, \mathbf{y})}{\partial V_{Qn}} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial g_{Qn}(\mathbf{x}, \mathbf{y})}{\partial V_{D1}} & \frac{\partial g_{Qn}(\mathbf{x}, \mathbf{y})}{\partial V_{Q1}} & \dots & \frac{\partial g_{Qn}(\mathbf{x}, \mathbf{y})}{\partial V_{Qn}} \end{bmatrix}$$

Network Jacobian Matrix



- The Jacobian matrix can be stored and computed using a 2 by 2 block matrix structure
- The portion of the 2 by 2 entries just from the Ybus are

$$\begin{bmatrix} \frac{\partial \hat{g}_{Di}(\mathbf{x}, \mathbf{y})}{\partial V_{Dj}} & \frac{\partial \hat{g}_{Di}(\mathbf{x}, \mathbf{y})}{\partial V_{Qj}} \\ \frac{\partial \hat{g}_{Qi}(\mathbf{x}, \mathbf{y})}{\partial V_{Dj}} & \frac{\partial \hat{g}_{Qi}(\mathbf{x}, \mathbf{y})}{\partial V_{Qj}} \end{bmatrix} = \begin{bmatrix} G_{ij} & -B_{ij} \\ B_{ij} & G_{ij} \end{bmatrix}$$

The "hat" was added to the g functions to indicate it is just the portion from the Y_{bus}

- The major source of the current vector voltage sensitivity comes from non-constant impedance loads; also dc transmission lines

Example: Constant Current and Constant Power Load



- As an example, assume the load at bus k is represented with a ZIP model

$$P_{Load,k} = P_{BaseLoad,k} (P_{z,k} |\bar{V}_k|^2 + P_{i,k} |\bar{V}_k| + P_{p,k})$$
$$Q_{Load,k} = Q_{BaseLoad,k} (Q_{z,k} |\bar{V}_k|^2 + Q_{i,k} |\bar{V}_k| + Q_{p,k})$$

The base load values are set from the power flow

- The constant impedance portion is embedded in Y_{bus}

$$\hat{P}_{Load,k} = P_{BaseLoad,k} (P_{i,k} |\bar{V}_k| + P_{p,k}) = (P_{BL,i,k} |\bar{V}_k| + P_{BL,p,k})$$
$$\hat{Q}_{Load,k} = Q_{BaseLoad,k} (Q_{i,k} |\bar{V}_k| + Q_{p,k}) = (Q_{BL,i,k} |\bar{V}_k| + Q_{BL,p,k})$$

- Usually solved in per unit on network MVA base

Example: Constant Current and Constant Power Load



- The current is then

$$\begin{aligned}\bar{I}_{Load,k} &= I_{D,Load,k} + jI_{Q,Load,k} = \left(\frac{\hat{P}_{Load,k} + j\hat{Q}_{Load,k}}{\bar{V}_k} \right)^* \\ &= \left(\frac{\left(P_{BL,i,k} \sqrt{V_{DK}^2 + V_{QK}^2} + P_{BL,p,k} \right) - j \left(Q_{BL,i,k} \sqrt{V_{DK}^2 + V_{QK}^2} + Q_{BL,p,k} \right)}{V_{DK} - jV_{QK}} \right)\end{aligned}$$

- Multiply the numerator and denominator by $V_{DK} + jV_{QK}$ to write as the real current and the reactive current

Example: Constant Current and Constant Power Load



$$I_{D,Load,k} = \frac{V_{DK}P_{BL,p,k} + V_{QK}Q_{BL,p,k}}{V_{DK}^2 + V_{QK}^2} + \frac{V_{DK}P_{BL,i,k} + V_{QK}Q_{BL,i,k}}{\sqrt{V_{DK}^2 + V_{QK}^2}}$$
$$I_{Q,Load,k} = \frac{V_{QK}P_{BL,p,k} - V_{DK}Q_{BL,p,k}}{V_{DK}^2 + V_{QK}^2} + \frac{V_{QK}P_{BL,i,k} - V_{DK}Q_{BL,i,k}}{\sqrt{V_{DK}^2 + V_{QK}^2}}$$

- The Jacobian entries are then found by differentiating with respect to VDK and VQK
 - Only affect the 2 by 2 block diagonal values

Where does Transient Stability Simulation Go Wrong?



- Already have initial x_0 and y_0
- Initialize $t = 0$ and $k = 0$
- Choose a Time Step Δt that will be used in numerical integration

If Δt too large there are numerical troubles

- Calculate new x : $x_{k+1} = x_k + \boxed{\Delta t} * f(x_k, y_k)$
 - called the “integration time step”
 - Many ways to do this. This is a “Forward Euler”
- Calculate new y : Solve $\boxed{g(x_{k+1}, y_{k+1}) = 0}$ **Can have numerical problems here too**
 - Called solving the “network boundary equations”
 - Very similar to power flow equation, but use complex currents instead
- Increment time t : $t_{k+1} = t_k + \Delta t$
- Increment k : $k = k + 1$
- Go back to 1 repeat

Numerical Troubles



- I find most users understand that a large step size can cause numeric problems
 - We have some tricks to get around these problems with particular models too (sub-interval integration)
- The network boundary equations can also be difficult to solve for some models
 - This is where load models present trouble
 - Constant Currents/Power
 - LD1PAC (single-phase air conditioner)
 - Also renewable (wind/solar) energy generator models
 - Also simple DC line models

Network Boundary Equations



- Similar to the “Power Flow”, but we use current equations instead power
 - Synchronous Machines fit nicely into current equations
 - $0 = YV - I_{gen} + I_{load}$
- Traditional Modeling
 - Generator Models
 - Synchronous Machines
 - Load Models
 - Induction Motor Models
 - Constant Impedance Models
 - Algebraic (Polynomial) Load Models
 - This is where most books end their discussion

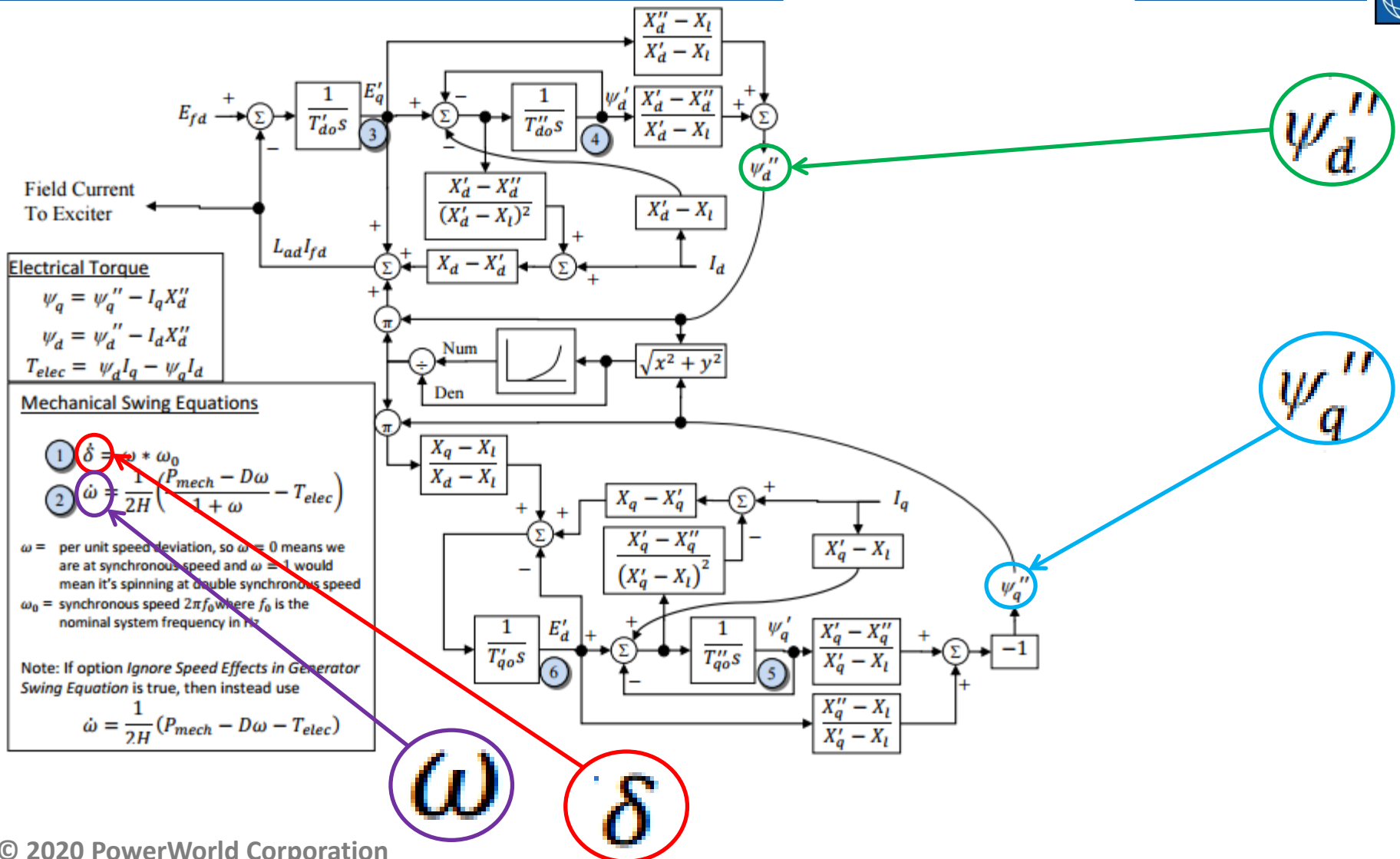
Synchronous Machine Model



- Documentation

- <http://www.powerworld.com/knowledge-base/derivation-of-the-gentpjgentpf-model-from-the-genrougensal>
- GENROU: http://www.powerworld.com/WebHelp/#TransientModels_HTML/Machine_Model_GENROU.htm
- GENTPF: http://www.powerworld.com/WebHelp/#TransientModels_HTML/Machine_Model_GENTPF.htm

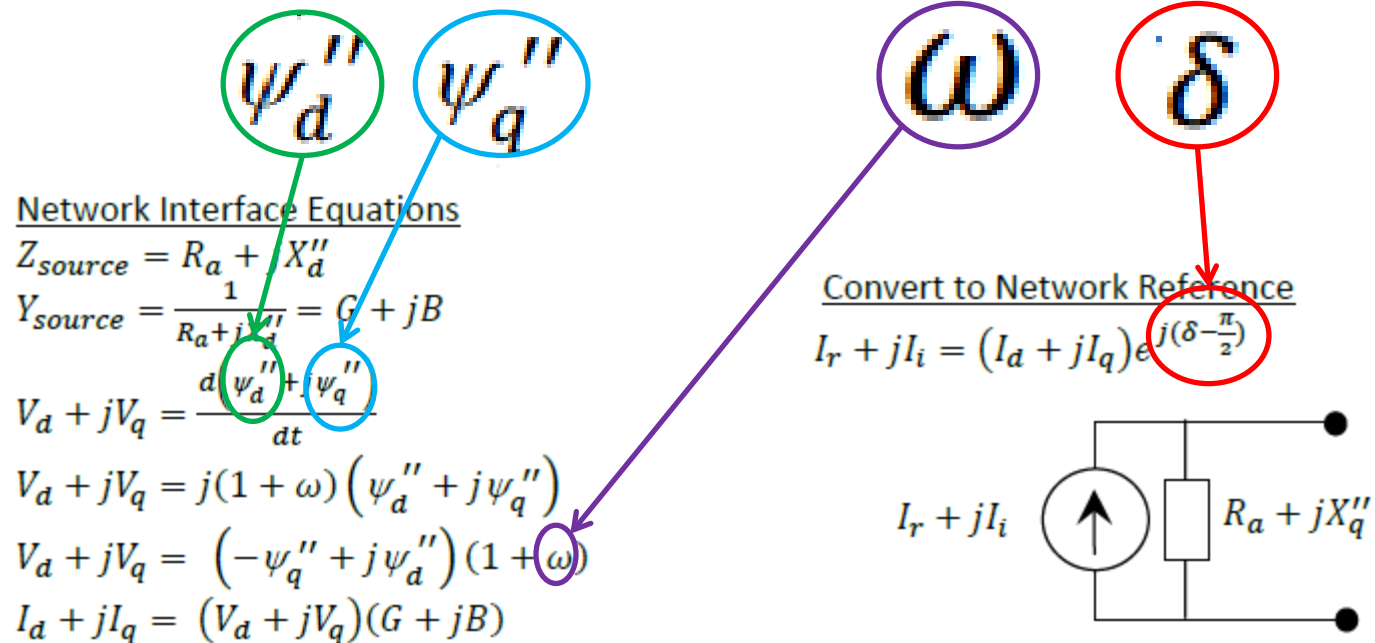
GENROU Model Dynamic Model



GENROU/GENSAL Network Boundary Equation Model



- Older models (GENROU/GENSAL) are modeled as a circuit equation

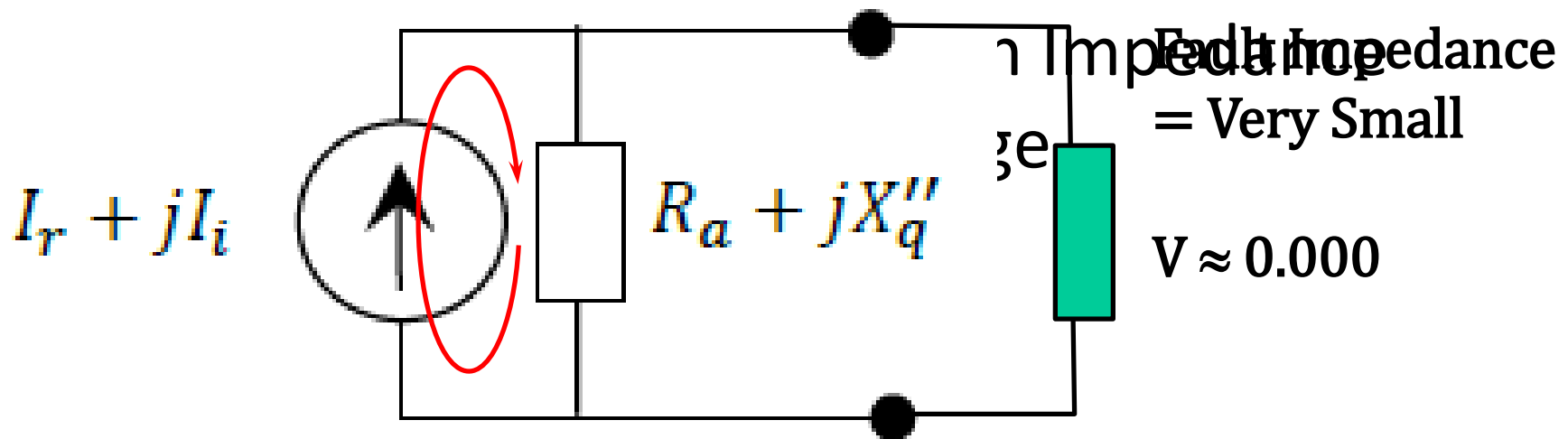


- GENTPF/GENTPJ are more complicated, but the same good behavior during a fault exists

Network Equation = Norton Equivalent Circuit



- Difficulty in Transient Stability Software occurs when modeling during the fault or just generally when the voltage is very low
- No trouble for Synchronous Machine because we have another path for current



Induction Motor Models



- Equation are very well behaved for network boundary equations too
- For a network equation solution they are represented by a
 - Complex voltage
 - Behind a Thevenin impedance
- This is the same as for a synchronous machines

Algebraic Load Models



- Algebraic functions of voltage and frequency

- IEEL load models

- WSCC model same

but $n_1=n_4 = 1$, $n_2=n_5=2$,
and $n_3=n_6=2$

$$P = P_{load} (a_1 v^{n_1} + a_2 v^{n_2} + a_3 v^{n_3}) (1 + a_7 \Delta f)$$

$$Q = Q_{load} (a_4 v^{n_4} + a_5 v^{n_5} + a_6 v^{n_6}) (1 + a_8 \Delta f)$$

- Power System Stability and Control, Kundar page 274

A comprehensive static model which offers the flexibility of accommodating several forms of load representation is as follows [5]:

$$P = P_0 [P_{ZIP} + P_{EX1} + P_{EX2}] \quad (7.5)$$

where

$$P_{ZIP} = p_1 \bar{V}^2 + p_2 \bar{V} + p_3$$

$$P_{EX1} = p_4 (\bar{V})^{a1} (1 + K_{pf1} \Delta f) \quad (7.6)$$

$$P_{EX2} = p_5 (\bar{V})^{a2} (1 + K_{pf2} \Delta f)$$

Algebraic Load Models



- What does Kundar say about these models
 - “Static models given by Equation 7.1 to 7.6 are not realistic at low voltages, and may lead to computational problems.”
 - Software vendors know this too!

The expression for the reactive component of the load has a similar structure. The reactive power compensation associated with the load is represented separately.

The static models given by Equations 7.1 to 7.6 are not realistic at low voltages, and may lead to computational problems. Therefore, stability programs usually make provisions for switching the load characteristic to the constant impedance model when the bus voltage falls below a specified value. In the load model used in the EPRI Extended Transient/Midterm Stability Program (ETMSP) [5], the exponents a_1 , a_2 , b_1 , and b_2 are varied as a function of voltage below a threshold value of bus voltage, and the constant power and constant current components are switched to constant impedance representation.

Why aren't they realistic



- Algebraic Models are aggregate models meant to represent the behavior of 1000s of individual loads (if not millions!)
 - These models are not calculated by measuring how loads behave at voltages near zero!
 - Real loads have protection that would act very quickly at very low voltages

What about Numerical Behavior



- Constant Impedance Loads (Exponent = 2)
 - Behave very well
 - Modeled the same as the transmission network which is also passive impedances
- Constant Current Loads (Exponent = 1)
 - Will have trouble when expressing equations in rectangular coordinates
 - Generally on the edge of numerical stability
- Constant Power Loads (Exponent = 0)
 - Numerical Problems → division by zero as you approach zero voltage

Constant Impedance Loads



- $S = VI^*$ (* mean complex conjugate)
 - $I = (S/V)^* = (S^*)/(V^*)$
- Define $V = e + jf$
- Consider constant current load given as P_{il} and Q_{il} which scale linearly with voltage

- $I_r + jI_i = V^2 \frac{(P_{il} - jQ_{il})}{e - jf}$
- $= [eP_{il} + fQ_{il}] + [fP_{il} - eQ_{il}]$
- $= P_{il}V \cos(\theta) + Q_{il}V \sin(\theta) + j[P_{il}V \sin(\theta) - Q_{il}V \cos(\theta)]$

No trouble at all! You can just model same as impedances from transmission system

Constant Current Loads



- $S = VI^*$ (* mean complex conjugate)
 - $I = (S/V)^* = (S^*)/(V^*)$
- Define $V = e + jf$
- Consider Impedance current load given as P_{il} and Q_{il} which scale linearly with voltage
 - $I_r + jI_i = V \frac{(P_{il} - jQ_{il})}{e - jf} = \frac{eP_{il} + fQ_{il}}{\sqrt{e^2 + f^2}} + j \frac{fP_{il} - eQ_{il}}{\sqrt{e^2 + f^2}}$
 - $= P_{il} \cos(\theta) + Q_{il} \sin(\theta) + j[P_{il} \sin(\theta) - Q_{il} \cos(\theta)]$

**Divide by Zero for rectangular equations.
On the edge of numerical stability**

Constant Power Loads



- $S = VI^*$ (* mean complex conjugate)
 - $I = (S/V)^* = (S^*)/(V^*)$
- Define $V = e + jf$
- Consider constant power load given as P_{sl} and Q_{sl}

$$- I_r + jI_i = \frac{P_{sl} - jQ_{sl}}{e - jf}$$

$$- = \frac{eP_{sl} + fQ_{sl}}{e^2 + f^2} + j \frac{fP_{sl} - eQ_{sl}}{e^2 + f^2}$$

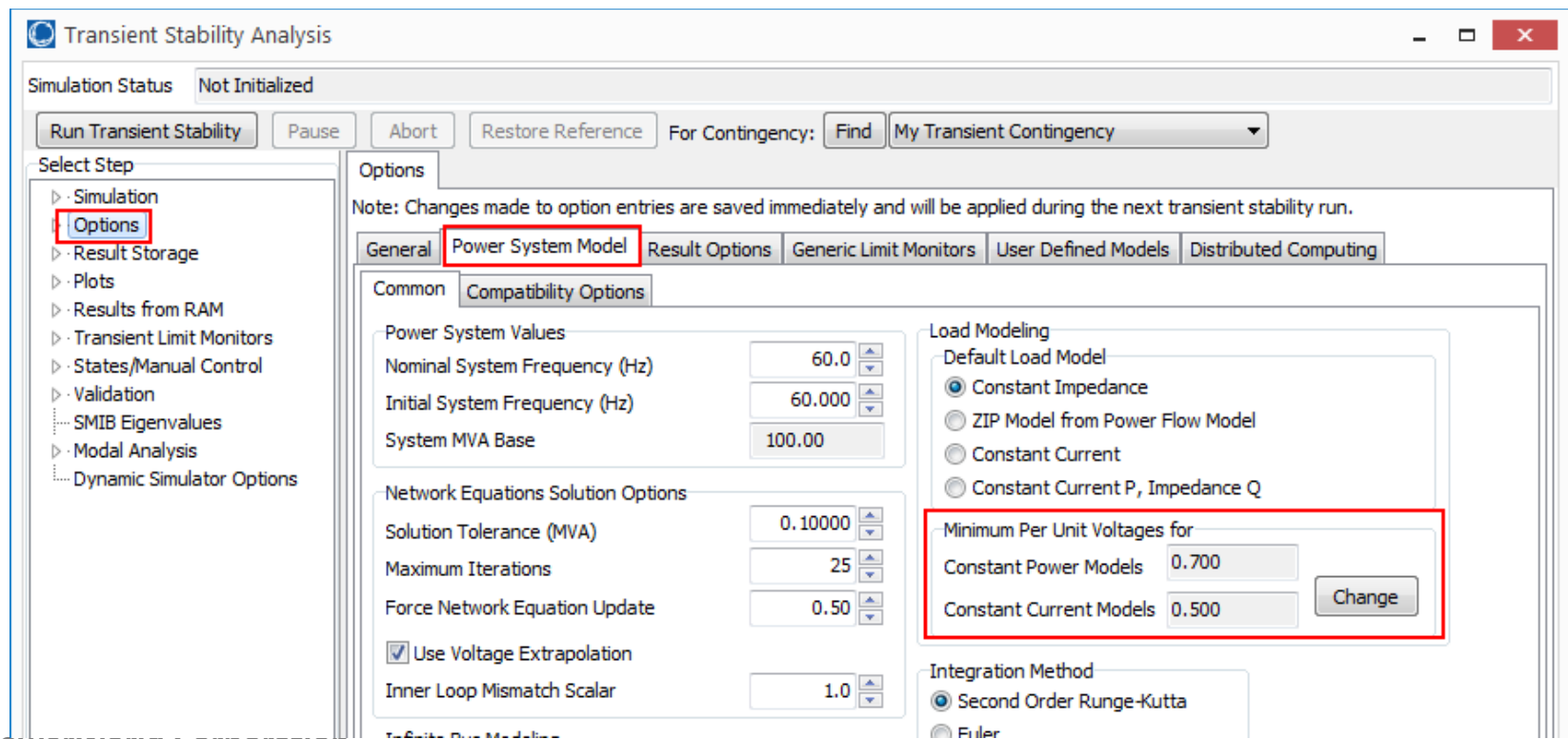
$$- = \frac{P_{sl} \cos(\theta) + Q_{sl} \sin(\theta)}{V} + j \left[\frac{P_{sl} \sin(\theta) - Q_{sl} \cos(\theta)}{V} \right]$$

**Divide by Zero
Impossible**

How does software handle this?



- Answer: user thresholds below which the constant current or power loads transition to something slightly lower



Minimum Per Unit Voltage for Constant Power and Current Loads



- Minimum Per Unit Voltage for Constant Power

$$MW(V_{pu}) = \frac{S_{MW}}{2} \left(1 - \cos \left(\frac{\pi V_{pu}}{V_{S \min pu}} \right) \right)$$

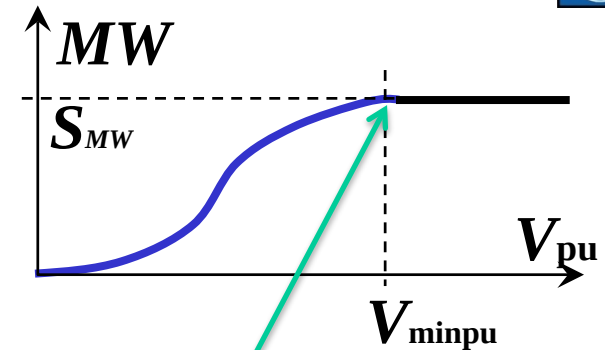
(Default Values are 0.7 and 0.5)

Minimum Per Unit Voltage for

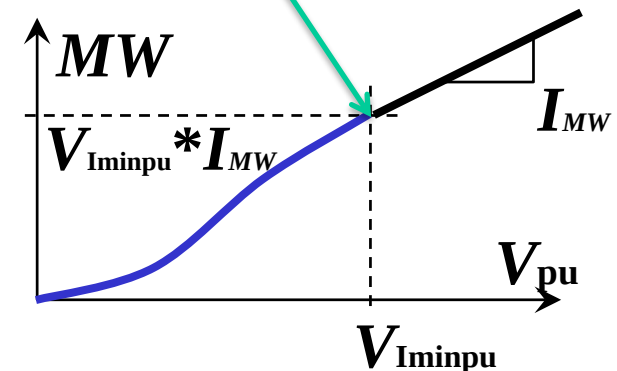
Constant Power Loads	0.700
Constant Current Loads	0.500

- Minimum Per Unit Voltage for Constant Current

$$MW(V_{pu}) = I_{MW} V_{pu} \sin \left(\frac{\pi}{2} \frac{V_{pu}}{V_{I \min pu}} \right)$$



Derivatives with respect to voltage remains continuous at min per unit value

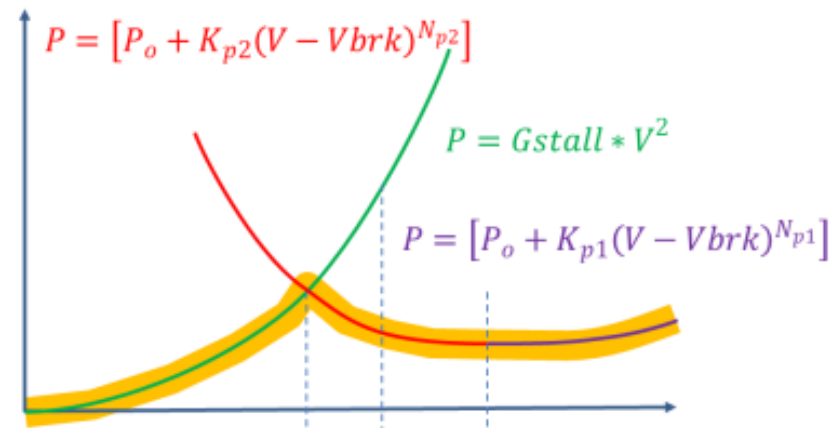


LD1PAC Load Model

(Single Phase Air-Conditioner)



- Performance Model is an algebraic function of Power versus voltage
 - Should P/Q relationship represents the steady-state behavior
 - However, if we're operating on the RED portion of this model and implemented that directly into the network boundary equations things go badly!
 - We don't do this!
 - Model as mostly impedance during network
 - Gives very different for large voltage swings (immediately after the fault and immediately after fault clears)

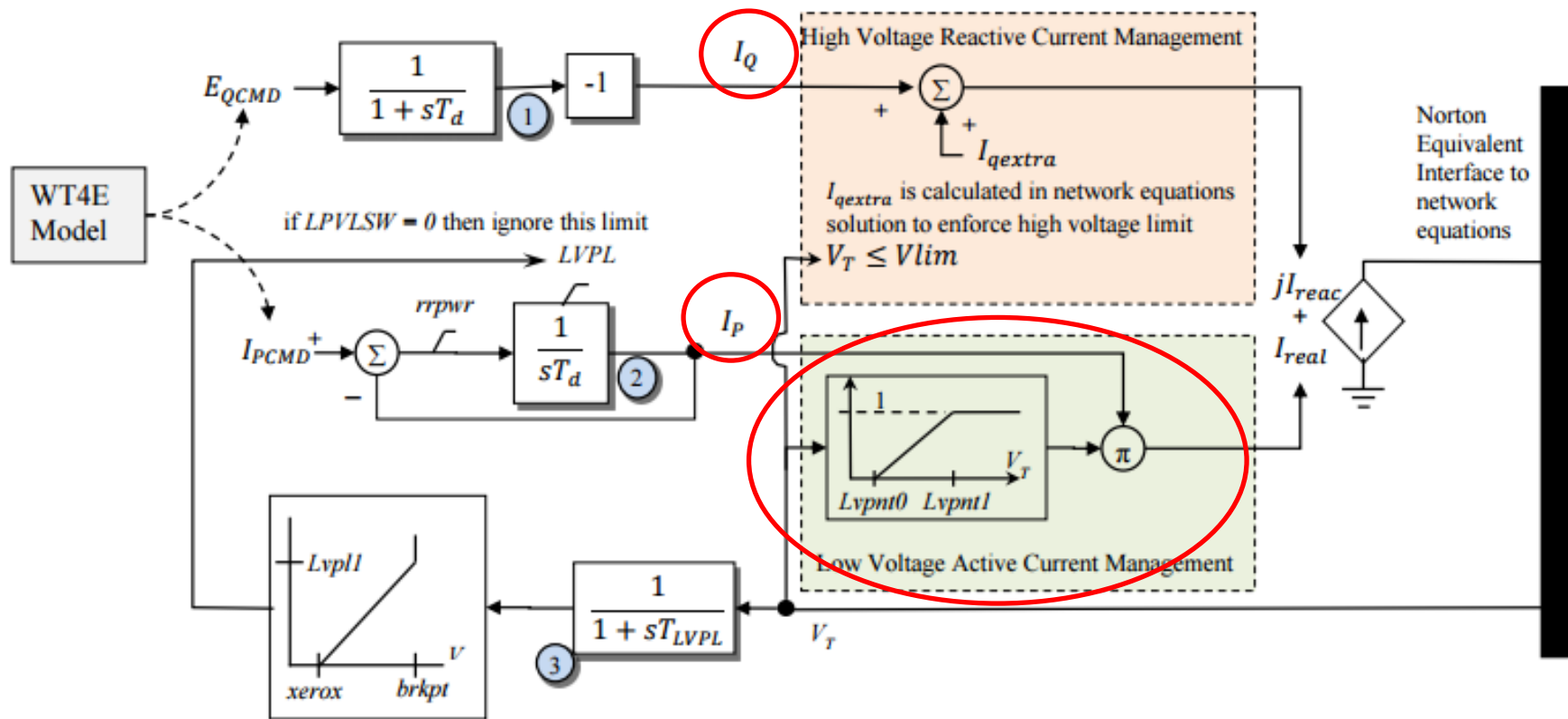


Rewearable Generation Models

REGC_A, WT3G, WT4G



- Constant Current Injection!

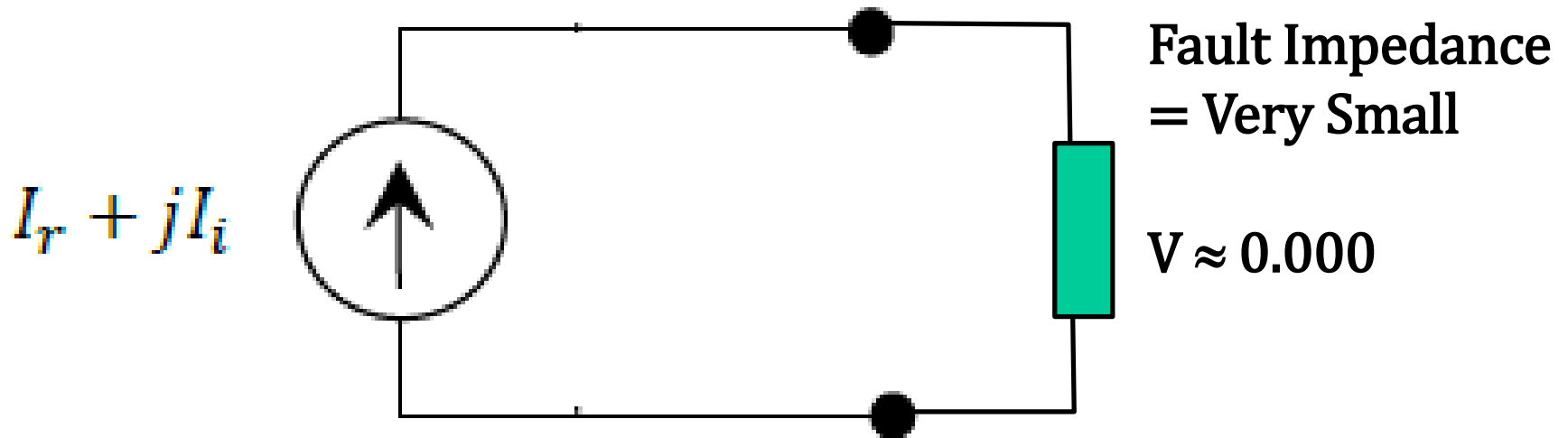


Could be problem, but you can model low voltage algebra inside network boundary equation solution

Constant Current Injection Model



- Injecting current directly into fault
- This is same equations as for constant current load
- Causes convergence issues in network



What is Frequency?



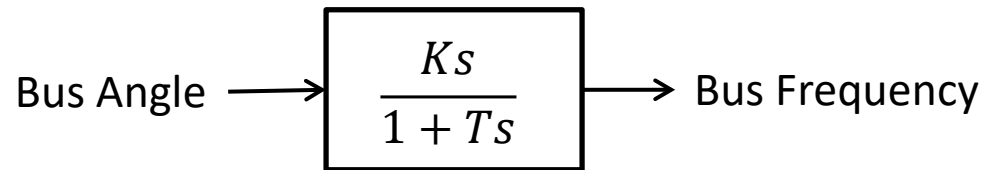
- Seems a weird question, but...
 - Frequency is NOT a state variable or solution variable in any equations in the fundamental numerical equations of transient stability
 - One of the fundamental outputs of transient stability, is a bit a trick in software tools
- In hardware, they count zero crossings of the AC waveforms or use other signal processing techniques
 - Transient stability is using positive sequence equations which assuming a frequency of 60 Hz
- How do we even calculated frequency then?

Bus Frequency Calculation in Transient Stability Software



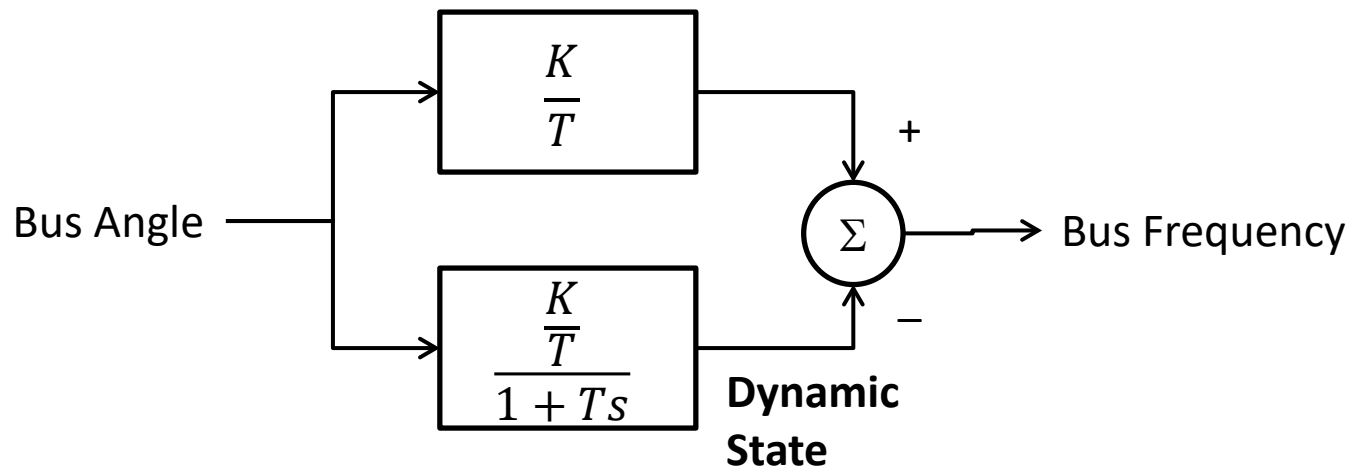
*Frequency is calculated
derivative of Bus Angle*

K represents unit
conversions between
radians to Hz



T represents a time
delay for calculating
the derivative

You can write this (or code it) as below.

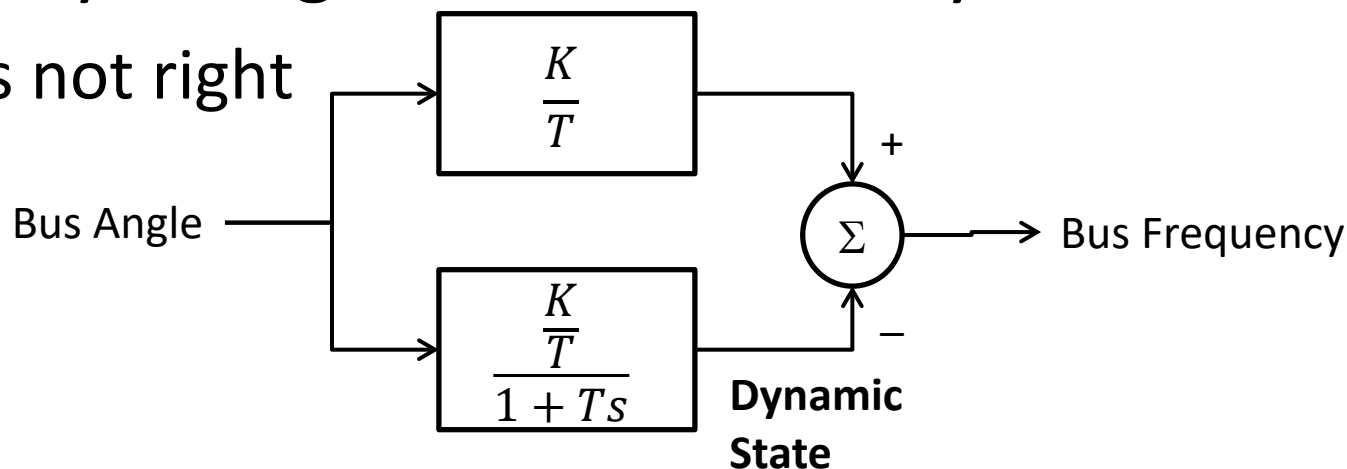


What happens when a network event (fault, open branch) occurs

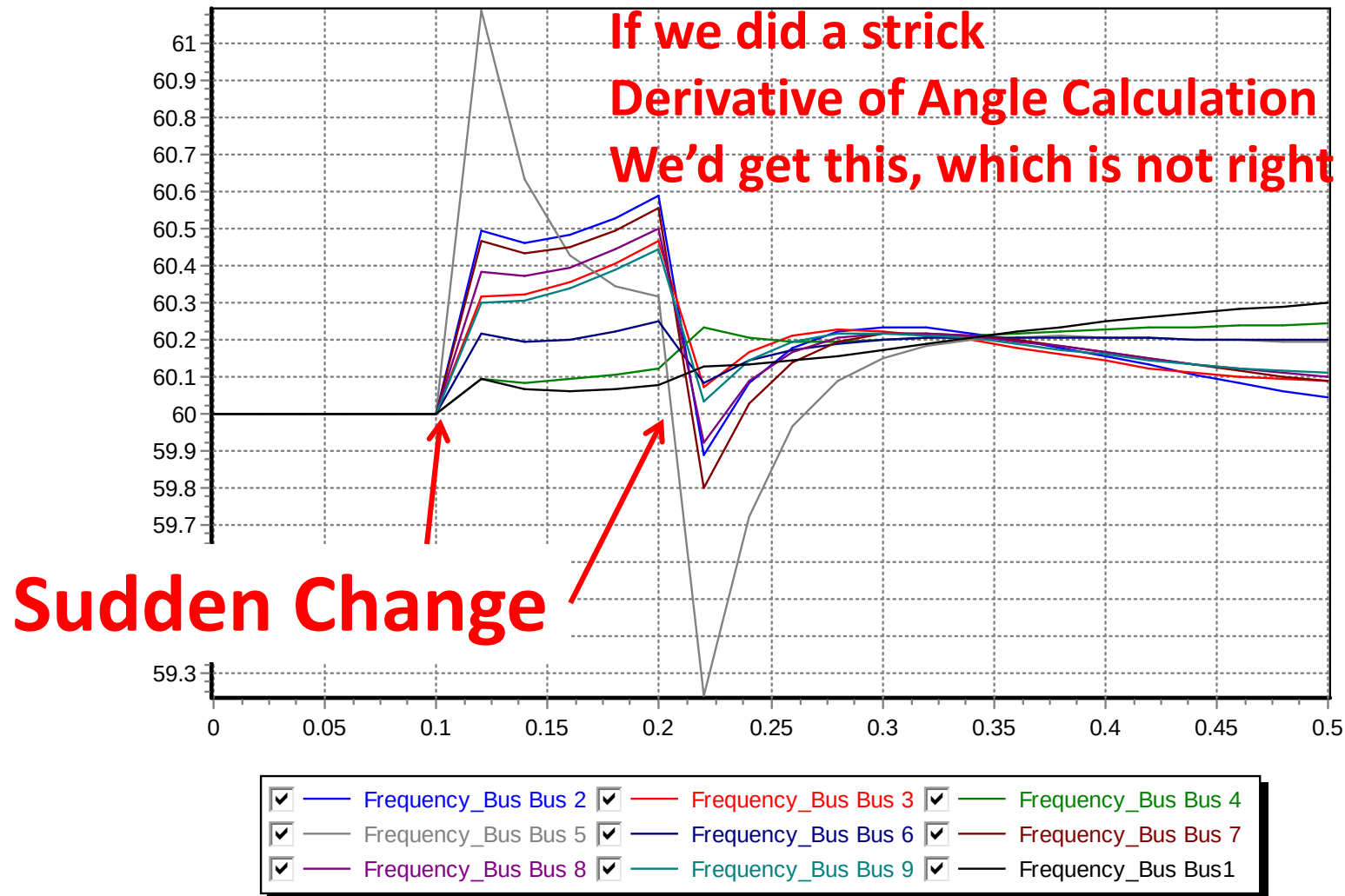


- Bus Angle will change instantaneously
- Proportional gain path will change instantaneously → Frequency will change instantaneously
- Frequency changes instantaneously!

– That's not right



Sudden change in frequency at fault 0.1 and clear at 0.2 seconds

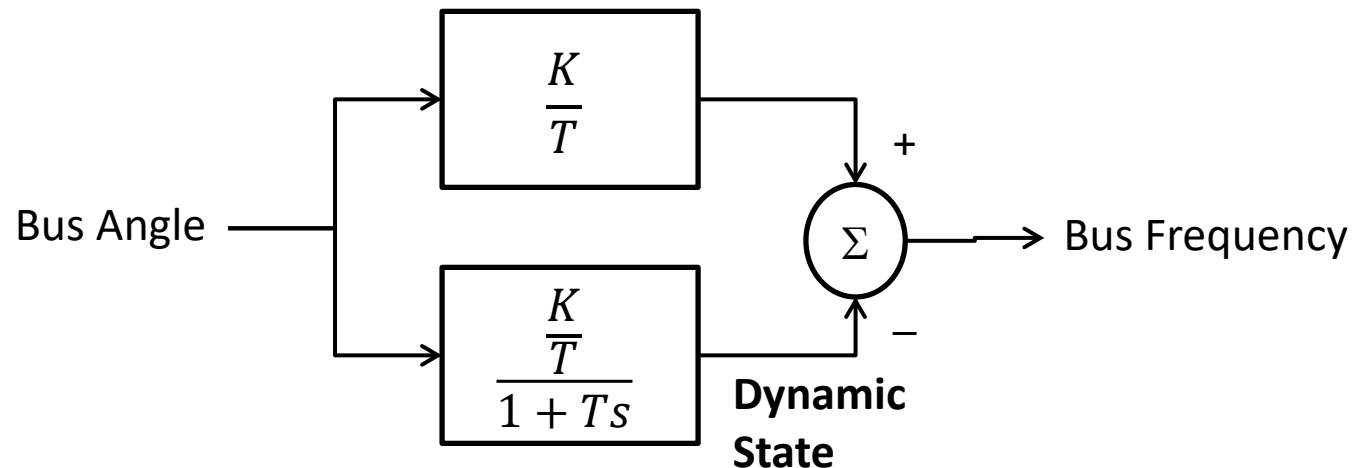


What is the software do for this?



- After the network solution immediately after a network event (Fault, Line Open, etc...), automatically go in and change the Dynamic State so that the Bus Frequency stays constant

$$\text{DynamicState} = \text{BusAngle} * \frac{K}{T} - \text{Bus FrequencyOld}$$



Software Tools



- This “Bus Frequency Measurement Time Constant” will be an input parameter somewhere

Transient Stability Analysis

Simulation Status: Not Initialized

Run Transient Stability Pause Abort Restore Reference For Contingency: Find My Transient Contingency

Select Step

- > Simulation
- Options
 - General
 - Power System Model
 - Common
 - Load Modeling
 - Compatibility Options
 - Remedial Actions
 - Result Options
 - Generic Limit Monitors
 - User Defined Models
 - Distributed Computing
 - Transfer to Power Flow
- > Result Storage
- > Plots
- > Results from RAM
- > Transient Limit Monitors
- > States/Manual Control
- > Validation
 - SMIB Eigenvalues
 - Modal Analysis
 - Dynamic Simulator Options

Note: Changes made to option entries are saved immediately and will be applied during the next transient stability run.

General Power System Model Remedial Actions Result Options Generic Limit Monitors User Defined Models Distributed Computing

Common Load Modeling Compatibility Options

Power System Values

Nominal System Frequency (Hz) 60.000

Initial System Frequency (Hz) 60.000

☒ When Using Playin Models Set Initial Hz to First Value

System MVA Base 100.00

Integration Method

☒ Second Order Runge-Kutta

☐ Euler

Infinite Bus Modeling

☒ No infinite buses (recommended setting)

☐ Model power flow slack buses as infinite buses

Frequency Measurement Options

Bus Frequency Measurement Time Constant (Sec.) 0.050

Minimum PU voltage for relay frequency measurement 0.300

☐ Calculate Bus ROCOF (Rate of Change of Freq)

☐ Use Parallel Code

Network Equations Solution Options

Solution Tolerance (MVA) 0.01000

Maximum Iterations 25

Abort after number of failed solutions 10

Force Network Equation Update 0.00

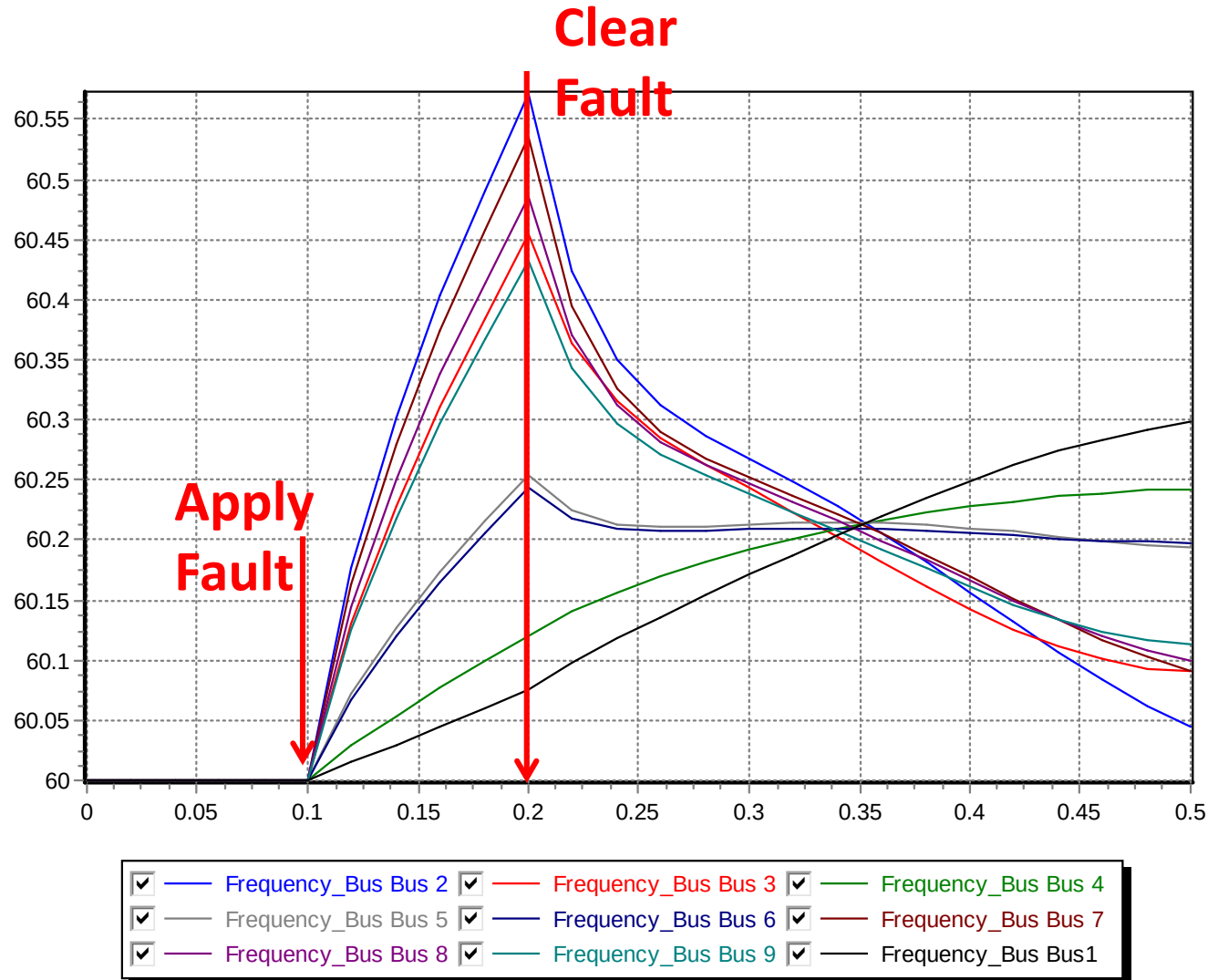
☒ Use Voltage Extrapolation

Inner Loop Mismatch Scalar 1.0

Handling of Initial Limit Violations

Island Angle ☒ S ☐ S ☐ N Degree Geom Ignor ☐ Ju GIC X

With this change, notice frequency doesn't suddenly change



Other Implications of Frequency Calculation

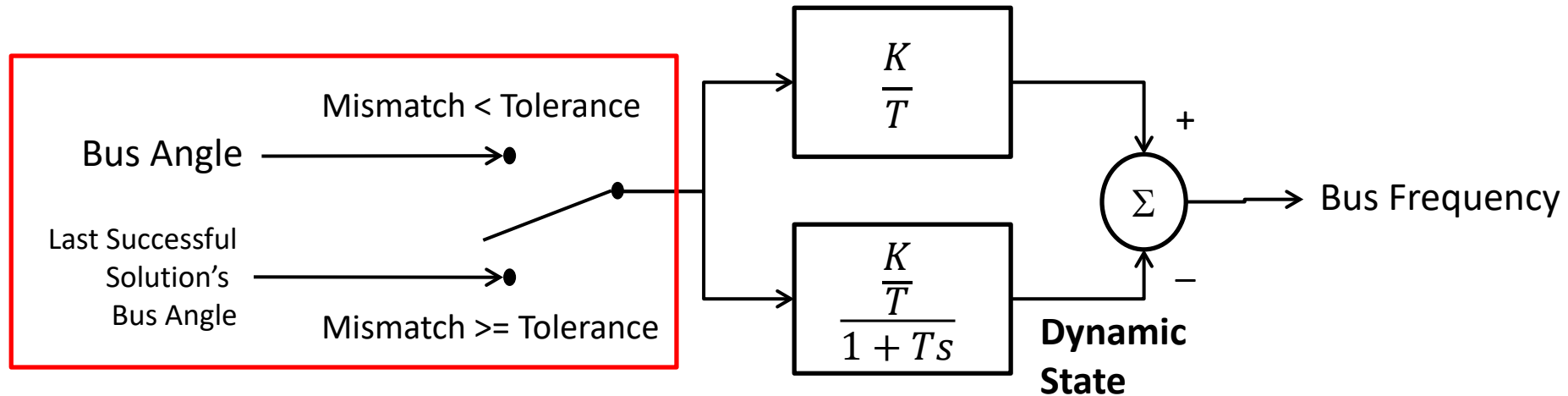


- The Input to the frequency calculation
 - Bus angle of the algebraic network boundary conditions
 - If network boundary conditions are not being successfully solved, then this angle is very noisy
 - Noisy angle = weird spikes in frequency
- One solution might be to add some logic to the frequency calculation
 - If large mismatch at bus (network not solved) then the frequency calculation at that bus should assume that bus angle hasn't changed from previous update?

Frequency Calculation at Buses which have not converged



- Add logic on Frequency Calculation outlined in red below



Use of Frequency in Relaying

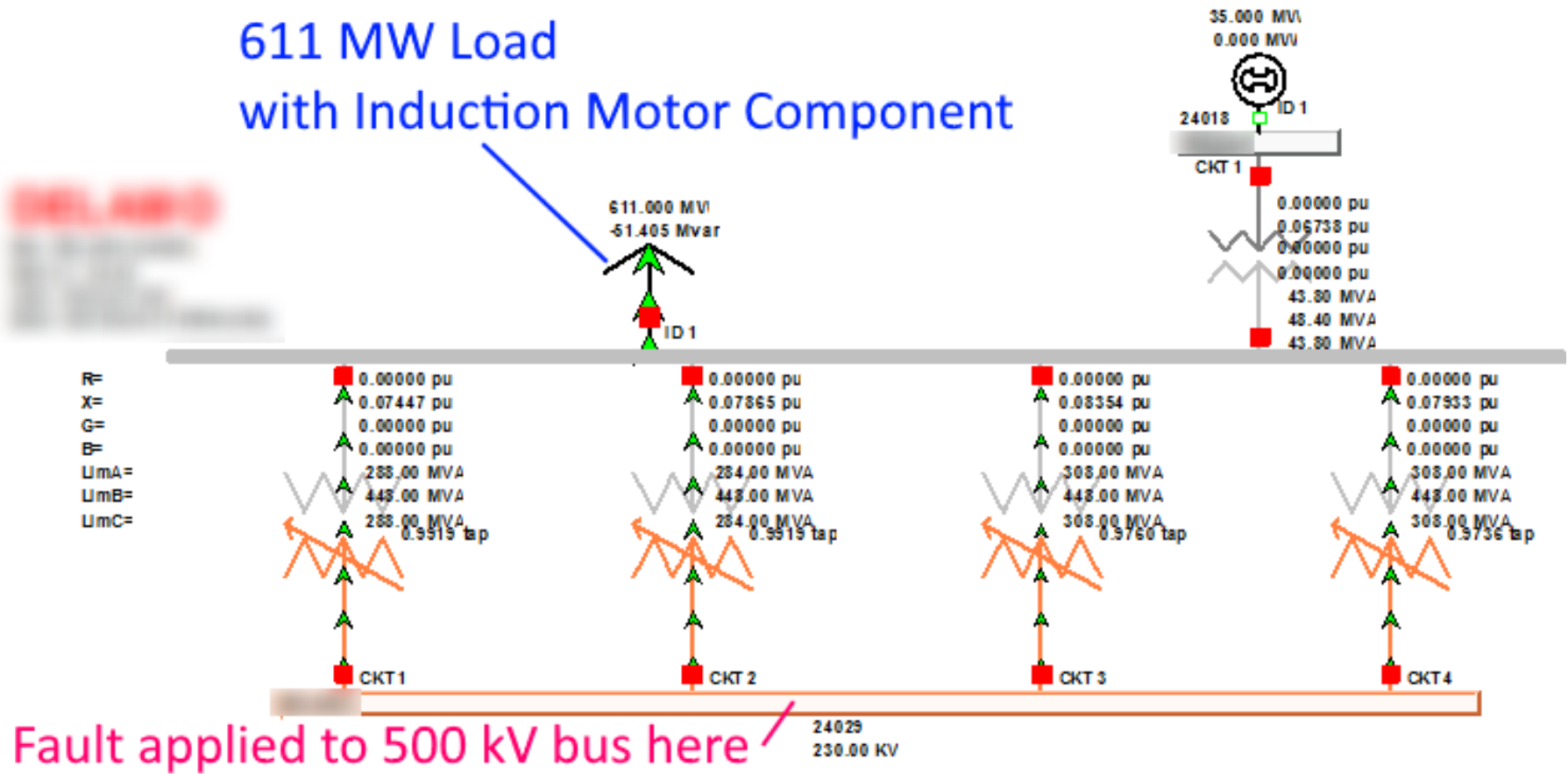


- PowerWorld introduced to this problem in support email on November 10, 2015
- Bug Report from Customer
 - 6 cycle 3-phase fault at a 500 kV bus was resulting in load tripping due to under-frequency
 - Multiple software tools were giving this same result
 - Customer didn't care.
 - Confirmation of a wrong answer wasn't making them happy
 - The load is huge (611 MW)
 - Customer also reported if they added small amount of impedance (small impedance away from 500 kV bus, say part way down the 500 kV line), then this didn't happen

611 MW Load



611 MW Load
with Induction Motor Component



First Line of Thinking: It's a fault impedance problem



- I like to blame the choice of fault impedance because it's often the problem (this wasn't really the problem)
- So I tried running the analysis with different fault impedances
- Fault applied at 0.1 seconds, cleared at 0.2 sec

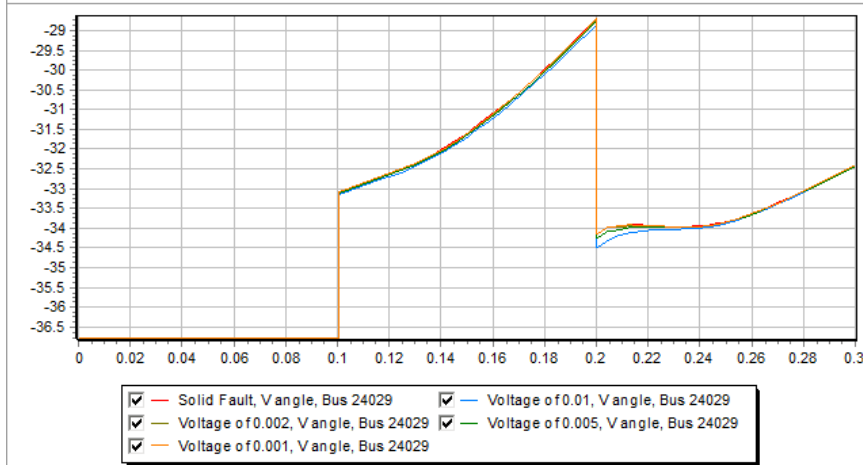
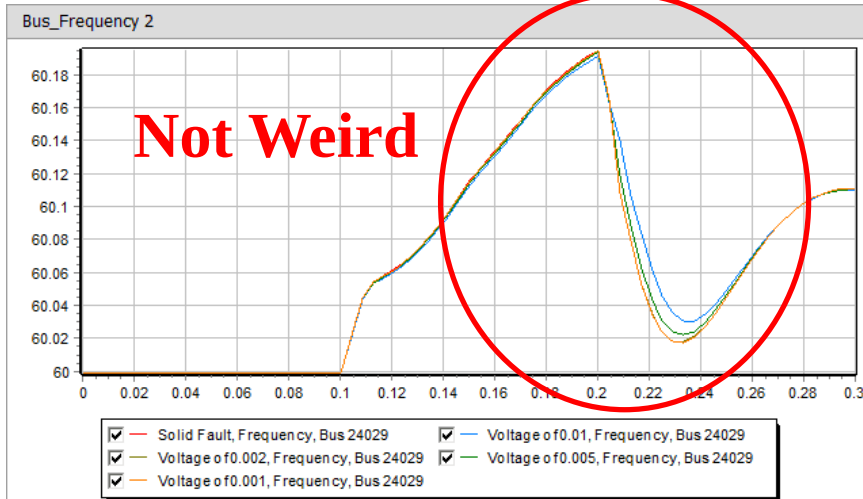
Post Fault Voltage	Fault X in per unit	Color on Plots
0.01	5.586E-5	Blue
0.005	2.778E-5	Green
0.002	1.108E-5	Olive
0.001	0.553E-5	Orange
Solid (Vpu=0.0000018)	1E-8	Red

Bus Frequency (Top)

Bus Voltage Angle (Bottom)

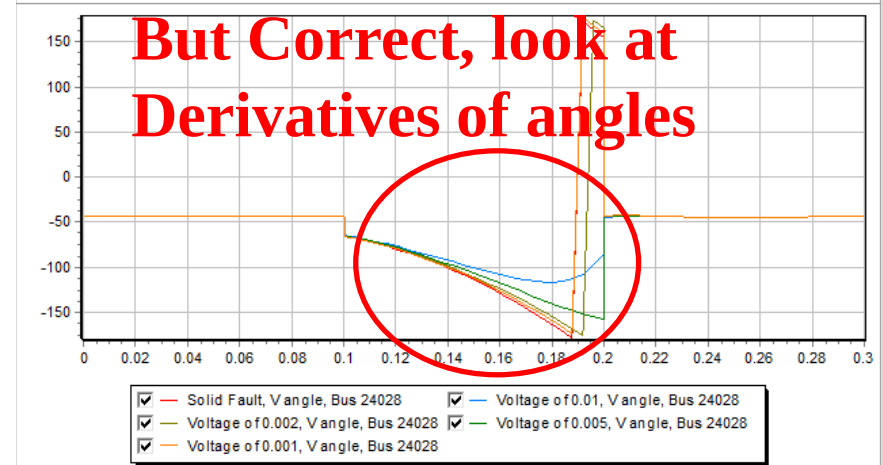
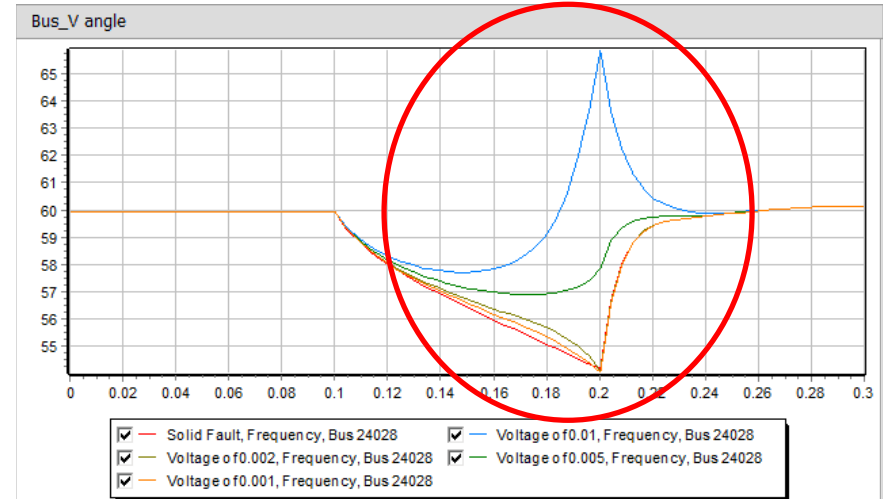


500 kV bus (the fault point)



Load bus

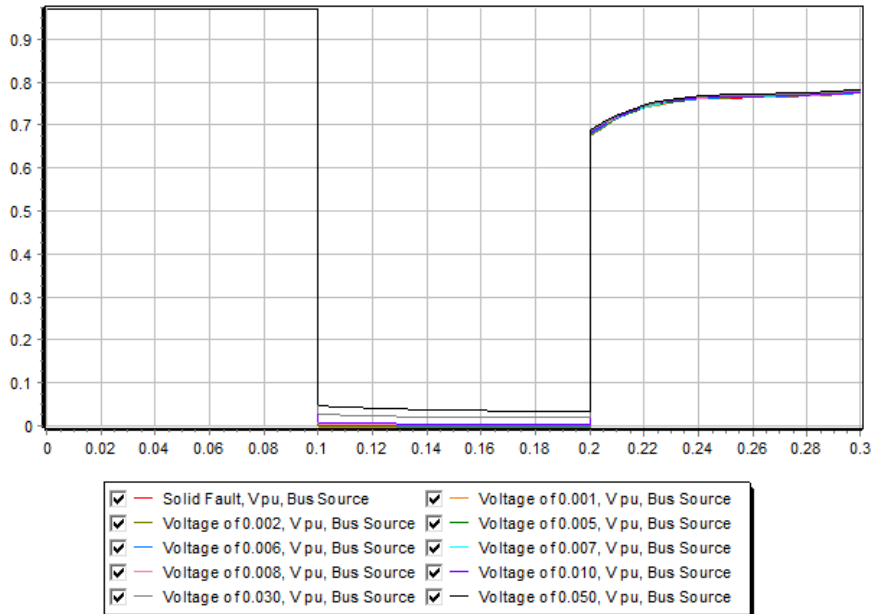
Weird



Bus PU Voltages

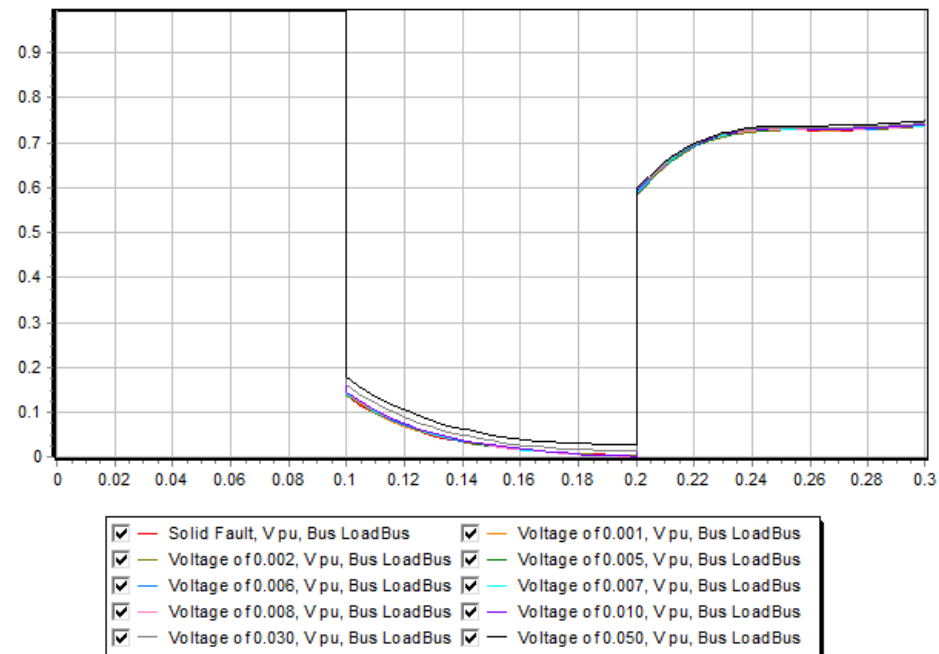


Source Voltage



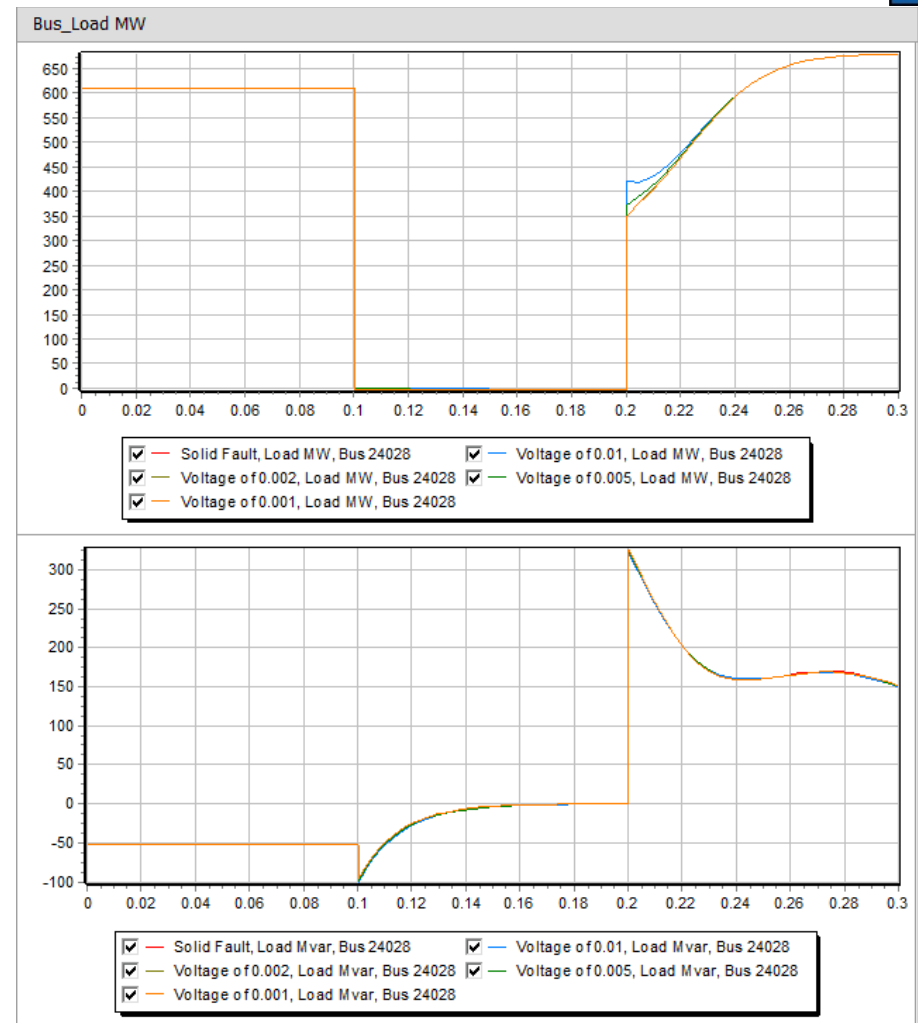
Load Bus Voltages < 0.2

Load Bus Voltage



What's going on with Load MW and Mvar

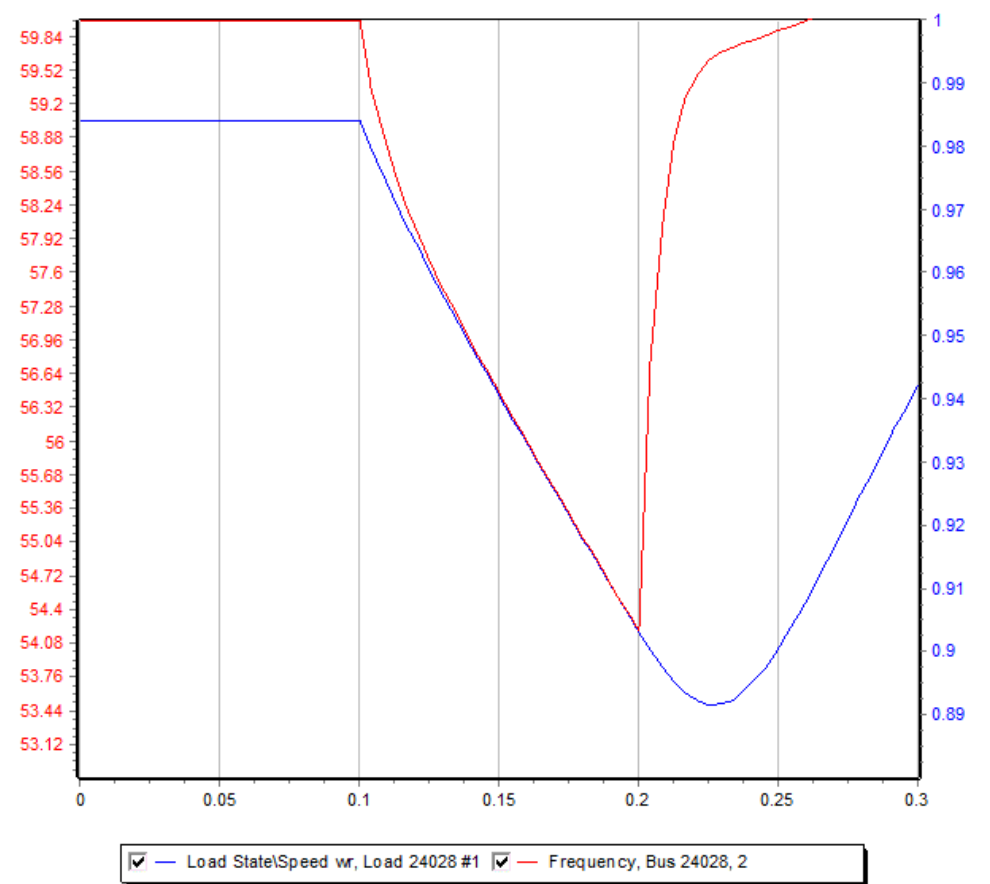
- MOTORW load model has induction motor.
- Below is the Load MW and Mvar at this load
- Varying fault impedance is changing results



Load Bus Frequency: It comes from induction motor



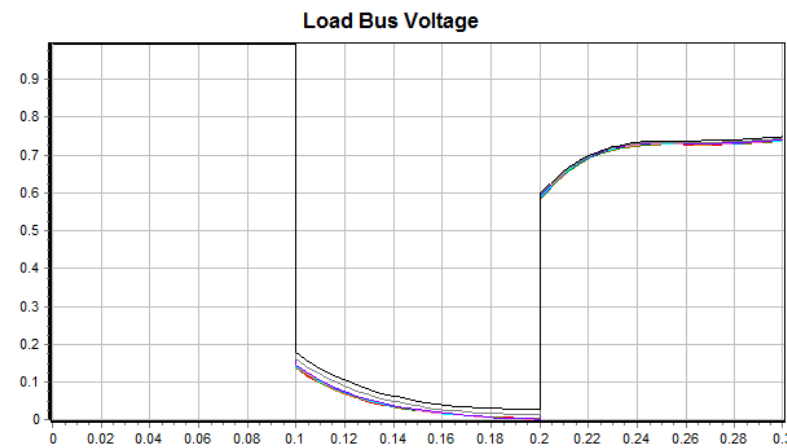
- Being driven by the induction motor
- 0.9 pu speed = 54 Hz bus freq



So what now?



- Frequency is being calculated “correctly” (to the best of a positive sequence tool’s ability)
- Problem is not with the calculation, its with the use of the frequency as an input to the relay
- What was the bus voltage here is extremely low (< 0.2 per unit!)
- My next step was to talk with engineers at Schweitzer Engineering Labs



Contact with Relay Engineers



- Took a week to get an engineering contact at Schweitzer
- November 17, 2015: corresponded with Dale Finney from SEL in Nova Scotia, Canada
- Indicated to me that under-frequency relays would not respond when voltage was so low
- He sent me the following excerpt from C37.117

Most microprocessor-based underfrequency relays have an internal voltage-blocking element that supervises the underfrequency element. The undervoltage element can be set up to block either the frequency element or output of the timer associated with the frequency element (Figure 10-2).

The undervoltage inhibit setting should be as low as possible to ensure that load is shed when necessary, even during voltage collapse incidents, but not so low that the underfrequency relay operates during faults or normal shutdowns, hindering restoration. A typical setting is between 50% and 70% of normal voltage.

- Under frequency relaying stops and assume that under voltage protection will take care of things
- Also indicated it's more than just an inhibit and that it impacts frequency calculation

Resolution: November 18, 2015

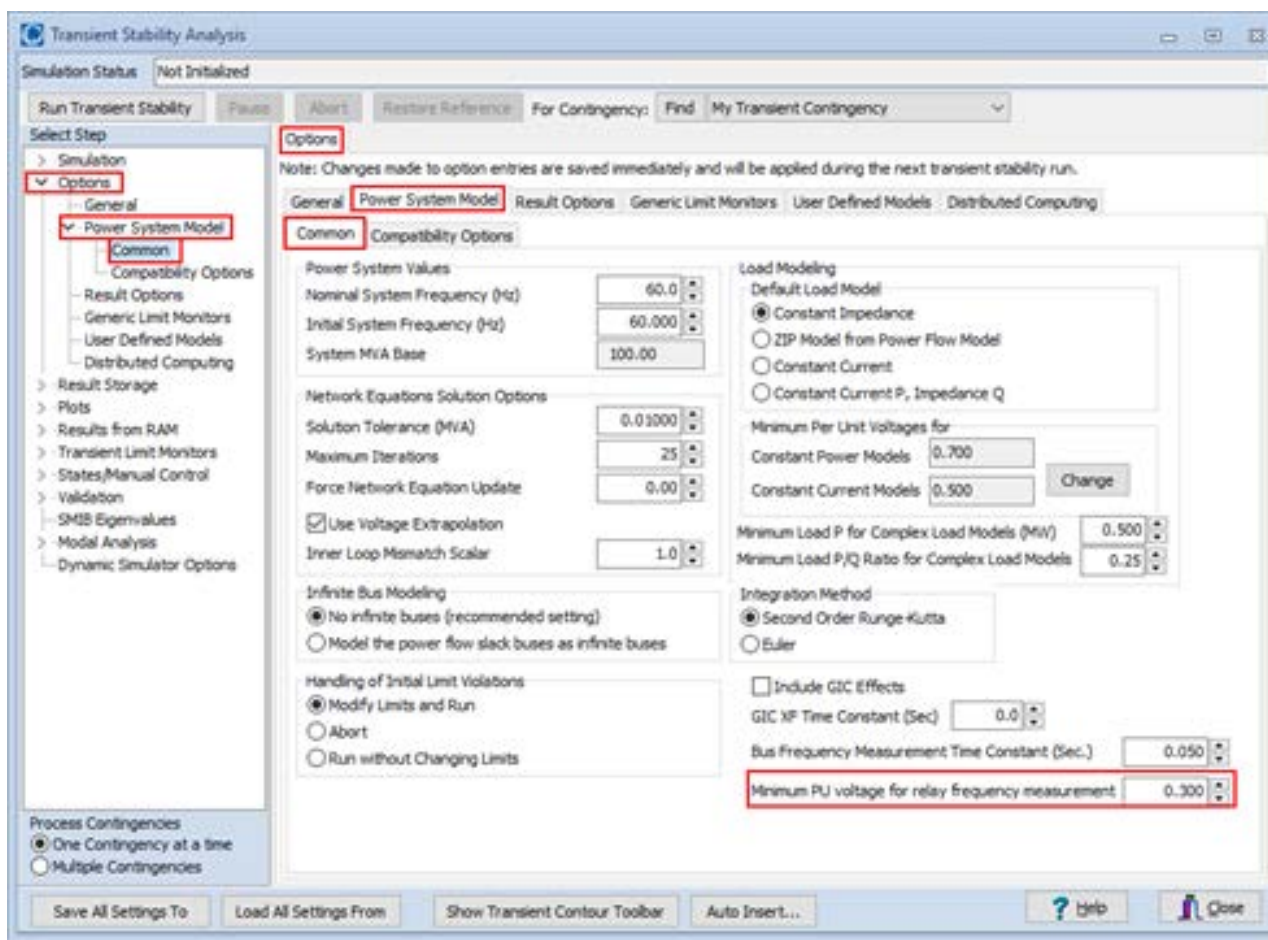


- Added a new global feature that applies to all frequency relays. (relays only)
 - Branch Relay: TLIN1
 - Load Relays: DLSH, LDSH, LDS3, LDST
 - Load Relays: LSDT1, LSDT8, LSDT9
- Minimum PU voltage for relay frequency measurement

Minimum PU voltage for relay frequency measurement



- Default value of 0.3 per unit



PowerWorld Help Documentation



https://www.powerworld.com/WebHelp/#MainDocumentation_HTML/Transient_Stability_Dialog_Options_Power_System_Model.htm

The screenshot shows a web browser window displaying the PowerWorld Corporation help page. The browser's address bar shows the URL: https://www.powerworld.com/WebHelp/#MainDocumentation_HTML/Transient_Stability_Dialog_Options_Power_System_Model.htm. The page features the PowerWorld Corporation logo and a search bar containing the text "minimum pu voltage". A left-hand navigation pane titled "Contents" lists various topics, with "Power System Model" selected under the "Options" category. The main content area is titled "Common: Frequency Measurement Options" and contains the following sections:

- Bus Frequency Measurement Time Constant (Sec.)**

Bus frequency in a transient stability simulation is a value which is calculated by performing a special calculation similar to taking the derivative of the bus angle. This means there is a time-constant associated with calculating this frequency and this value can be specified here. Also note that Bus frequency is calculated at all buses during a transient stability simulation.
- Minimum PU voltage for relay frequency measurement Added in Version 19, November 24, 2015**

Some transient stability models involve using bus frequency as signal that determines whether to trip a device. Load relays or generator relays for instance. In some extreme situations at very low voltages, the calculation of bus frequency using a time constant can give very low or very high frequency values which would not be seen by a real relay. In a real relay device there is always a voltage threshold below which these types of relays be blocked from operating. This is because frequency in relays is calculated by looking at the zero crossing of a the AC wave form. The counting of zero crossings becomes unreliable at low voltage. This option (which defaults to 0.3 per unit) will change how relays perceive frequency when the per unit voltage is below this threshold. If the voltage falls below this threshold the relay will behave as though it is seeing the nominal frequency.
- Calculate Bus ROCOF (Rate of Change of Frequency) Added in Version 20**

Check this box to also calculate the derivative of the bus frequency.
- Common: Island Synchronization Added in Version 20**

Transient stability simulations do not fully model the closing of a transmission line that connects two separately synchronized electrical islands. In order to properly model this numerically, one must assume that the presently the phase angles on either side of the AC branch being closed in are nearly matched and that the frequencies of both

PowerWorld Simulator 18 Patch Website



← → ↻ PowerWorld Corporation [US] | https://www.powerworld.com/patch-releases/simulator-18?filter_by_date=November+2015 ☆ ⓘ 🔍 📄 📱 📺 ⋮

 **PowerWorld**
CORPORATION

[Try Simulator for Free](#) 🔍

[Home](#) [Products](#) [Services](#) [Solutions](#) [Download & Purchase](#) [Online Support](#) [Training & Events](#) [Company](#)

Simulator 18 Patches

	New Features	Bug Fixes	Date
Demo Software How to Purchase Request a Quote Software License Files (Keys) Software and Patches ▪ Simulator 20 ▪ Retriever 20 ▪ Simulator 19 ▪ Retriever 19 ▪ Simulator 18 ▪ Retriever 18	➤ Auxiliary Files (SCRIPT and DATA Sections)	➤ Auxiliary Files (SCRIPT and DATA Sections)	➤ July 2016
	➤ Available Transfer Capability (ATC) Tool	➤ Available Transfer Capability (ATC) Tool	➤ June 2016
	➤ Case Information Displays	➤ Bus View and Substation View Onlines	➤ May 2016
			➤ April 2016

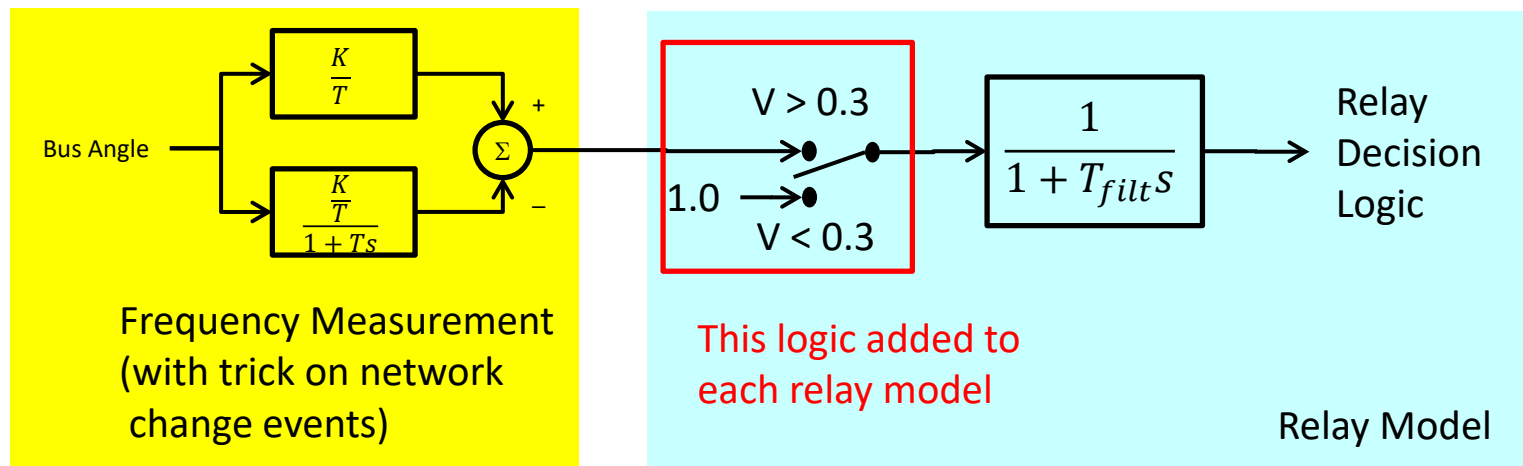
November 24, 2015

➤ **User Interface Dialogs:** Adding a new global transient stability option for "Minimum Voltage For Frequency Relay". When the per unit voltage is below this threshold then all frequency relays will treat their measured frequency as nominal frequency. This was done to mimic the voltage inhibit flag of real relay models and also to prevent motors stuck behind a fault from tripping on low frequency when the measured frequency behind the fault is suspect.

What does this do?



- Added simple logic in red box below inside each relay model
- More than just an “inhibit flag”
 - You must change the calculation of the frequency which is being used for the relay decision logic
 - Must not accumulate the “weird” frequency seen during the fault
 - Otherwise you’ll trip as soon as the voltage recovers because the relay frequency state will have accumulated the frequency



Implications



- First, everyone should add this global flag for use in frequency relays
 - Have I mentioned this only applied to relays
 - This should not change the frequency calculation itself
 - This should not be done on the frequency signal fed into governors for example
- Second, Education of Industry
 - Software users must not configure frequency relays to respond with a time delay of 0.0 seconds (instantaneous)
 - There is always a delay in the calculation of frequency of a few cycles (they're counting zero crossings!)
 - Even if it was possible to use some special signal processing techniques, positive sequence simulation tools can not do this
 - Our frequency calculation is just not precise enough to do that