

# ECEN 667

## Power System Stability

### Lecture 14: Governors, Time-Domain Simulation Solutions (Transient Stability)

---

Prof. Tom Overbye

Dept. of Electrical and Computer Engineering

Texas A&M University

[overbye@tamu.edu](mailto:overbye@tamu.edu)



TEXAS A&M  
UNIVERSITY

# Announcements

---



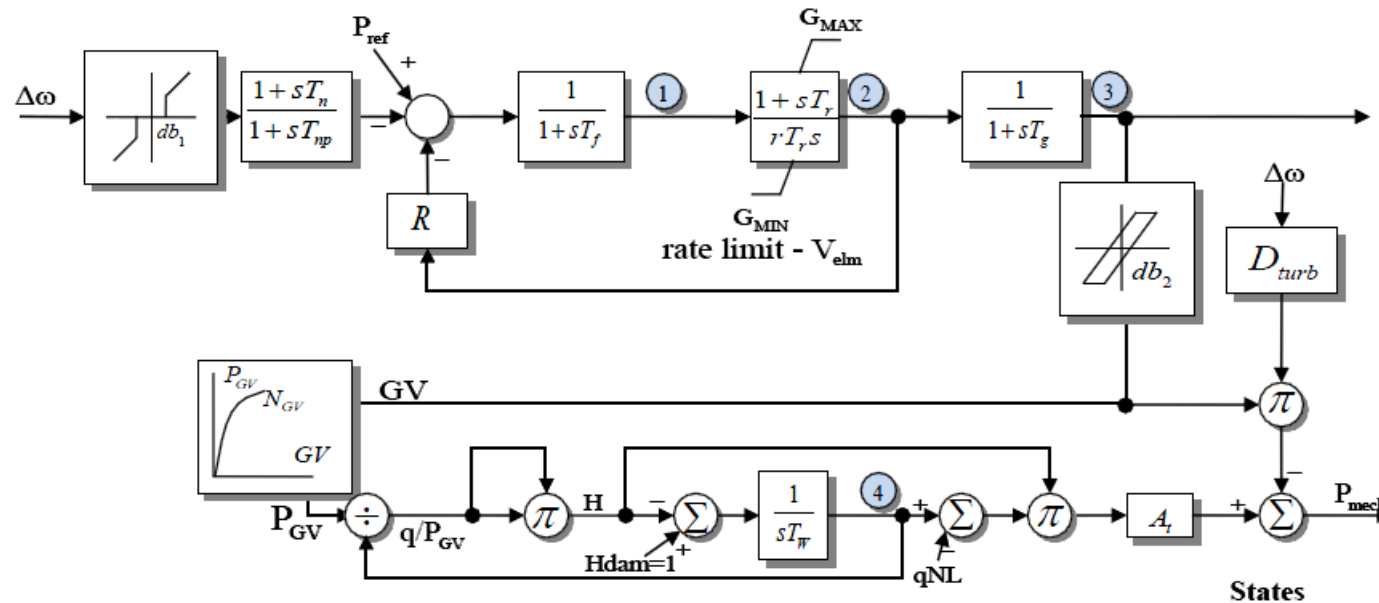
- Read Chapters 4 and 7
- Homework 4 is Thursday October 19
- Exam 1 average 86.9
- A classic paper on stability simulations is B. Stott, “Power System Dynamic Response Calculations,” *Proc. IEEE*, February 1979, pp. 219-241
  - The purpose of IEEE Proceedings is to provide state-of-the-art coverage of current hot topics in the broad areas covered by IEEE; the goal is to help people who are not specialists in a field keep up with important developments
  - Its impact factor is 20.6 (impact factor tells how often articles are cited in a year); the value for IEEE Trans. on Power Systems is 7.3; (highest values are with medical journals, going above 100; the value for Nature is 65)
  - You can see the recent issue of IEEE Proceedings at [proceedingsoftheieee.ieee.org/](http://proceedingsoftheieee.ieee.org/)

# Model HYGOV



- This simple model, combined with a governor, is implemented in HYGOV

About 6% of WECC governors use this model; average  $T_w$  is two seconds

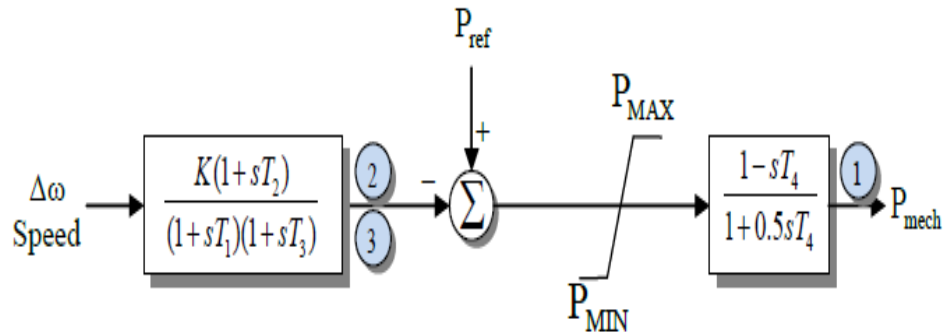


The gate position (GV) to gate power ( $P_{GV}$ ) is sometimes represented with a nonlinear curve

$H_{loss}$  is assumed small and not included

# Linearized Model Derivation

- The previously mentioned linearized model can now be derived as



$$\frac{dq}{dt} = \frac{(1 - h(c))_{gate}}{T_w}$$

$$\frac{d\Delta q}{dt} = -\frac{\Delta h(c)_{gate}}{T_w} \rightarrow \Delta q = \frac{\partial q}{\partial c} \Delta c + \frac{\partial q}{\partial h} \Delta h$$

And for the linearized power

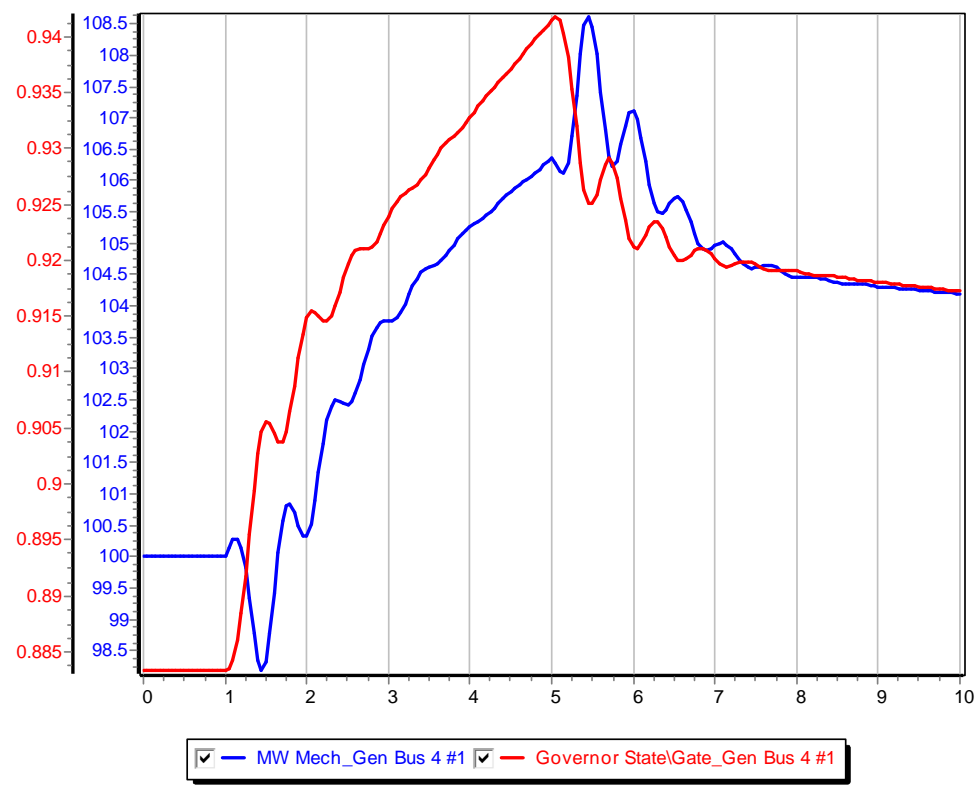
$$\Delta P_m = \frac{\partial P_m}{\partial h} \Delta h + \frac{\partial P_m}{\partial q} \Delta q$$

$$\text{Then } \frac{\Delta P_m}{\Delta c} = \frac{\left[ \frac{\partial q}{\partial c} \frac{\partial P_m}{\partial q} - sT_w \frac{\partial P_m}{\partial h} \frac{\partial q}{\partial c} \right]}{1 + sT_w \frac{\partial q}{\partial h}}$$

# Four Bus Case with HYGOV



- The below graph plots the gate position and the power output for the bus 2 signal generator decreasing the speed then increasing it



Note that just like in the linearized model, opening the gate initially decreases the power output

Case name: **B4\_SignalGen\_HYGOV**

# PID Controllers

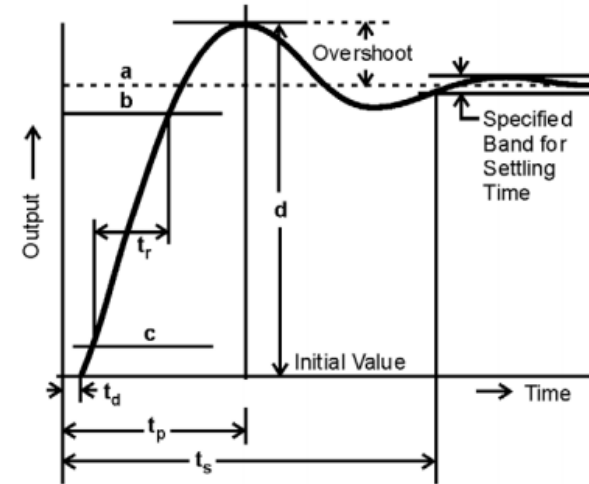
---



- Governors and exciters often use proportional-integral-derivative (PID) controllers
  - Developed in 1890's for automatic ship steering by observing the behavior of experienced helmsman
- PIDs combine
  - Proportional gain, which produces an output value that is proportional to the current error
  - Integral gain, which produces an output value that varies with the integral of the error, eventually driving the error to zero
  - Derivative gain, which acts to predict the system behavior. This can enhance system stability, but it can be quite susceptible to noise

# PID Controller Characteristics

- Four key characteristics of control response are
  - 1) rise time, 2) overshoot,
  - 3) settling time and
  - 4) steady-state errors



a - Steady-state value  
 b - 90% of steady-state value  
 c - 10% of steady-state value  
 d - peak value  
 $t_d$  - Delay time  
 $t_p$  - Time to reach peak value  
 $t_s$  - Settling time  
 $t_r$  - Rise time

Figure F.1—Typical dynamic response of a turbine governing system to a step change

Increasing Gain	Rise Time	Overshoot	Setting Time	Steady-State Error
$K_P$	Decreases	Increases	Little impact	Decreases
$K_I$	Decreases	Increases	Increases	Zero
$K_D$	Little impact	Decreases	Decreases	Little Impact

# PID Example: Car Cruise Control

---



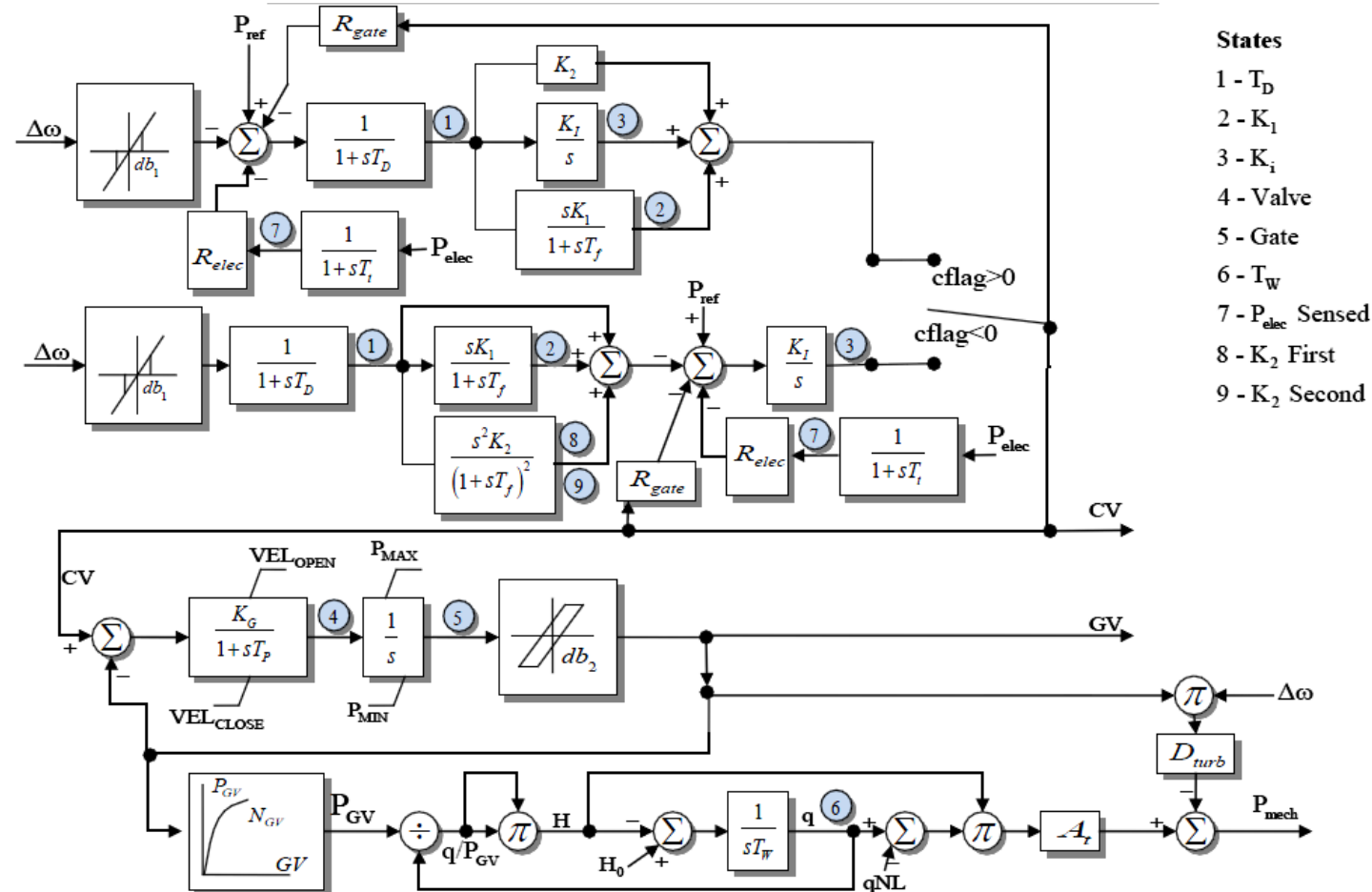
- Say we wish to implement cruise control on a car by controlling the throttle position
  - Assume force is proportional to throttle position
  - Error is difference between actual speed and desired speed
- With just proportional control we would never achieve the desired speed because with zero error the throttle position would be at zero
- The integral term will make sure we stay at the desired point
- With derivative control we can improve control, but as noted it can be sensitive to noise



# HYG3



- The HYG3 models has a PID or a double derivative



- States
- 1 -  $T_D$
  - 2 -  $K_1$
  - 3 -  $K_i$
  - 4 - Valve
  - 5 - Gate
  - 6 -  $T_w$
  - 7 -  $P_{elec}$  Sensed
  - 8 -  $K_2$  First
  - 9 -  $K_2$  Second

Looks more complicated than it is since depending on cflag only one of the upper paths is used

About 15% of current WECC governors at HYG3

# Tuning PID Controllers

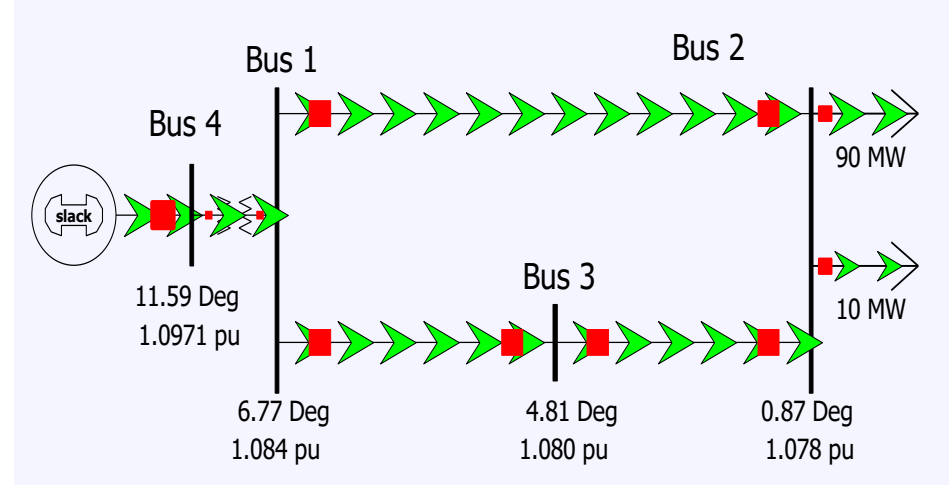
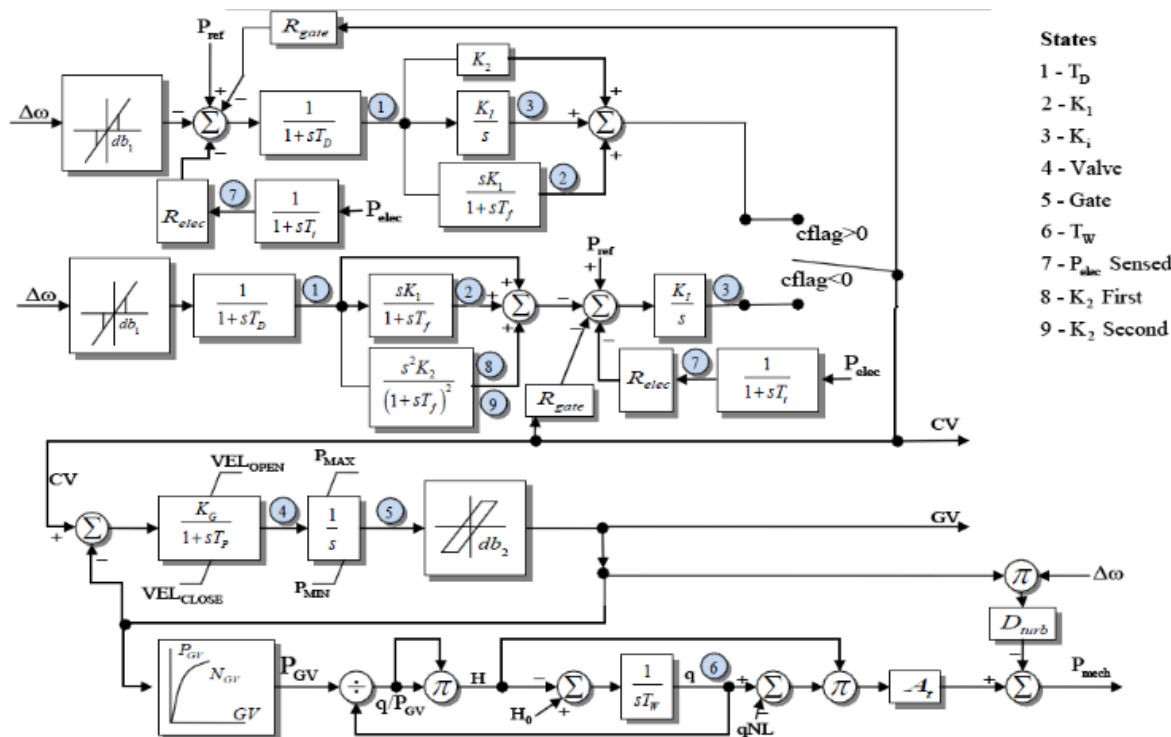


- Tuning PID controllers can be difficult, and there is no single best method
  - Conceptually simple since there are just three parameters, but there can be conflicting objectives (rise time, overshoot, setting time, error)
- One common approach is the Ziegler-Nichols method
  - First set  $K_I$  and  $K_D$  to zero, and increase  $K_P$  until the response to a unit step starts to oscillate (marginally stable); define this value as  $K_u$  and the oscillation period at  $T_u$
  - For a P controller set  $K_p = 0.5K_u$
  - For a PI set  $K_p = 0.45 K_u$  and  $K_I = 1.2 * K_p / T_u$
  - For a PID set  $K_p = 0.6 K_u$ ,  $K_I = 2 * K_p / T_u$ ,  $K_D = K_p T_u / 8$

# Tuning PID Controller Example



- Use the four bus case with infinite bus replaced by load, and gen 4 has a HYG3 governor with  $cflag > 0$ ; tune  $K_P$ ,  $K_I$  and  $K_D$  for full load to respond to a 10% drop in load ( $K_2$ ,  $K_I$ ,  $K_1$  in the model; assume  $T_f=0.1$ )

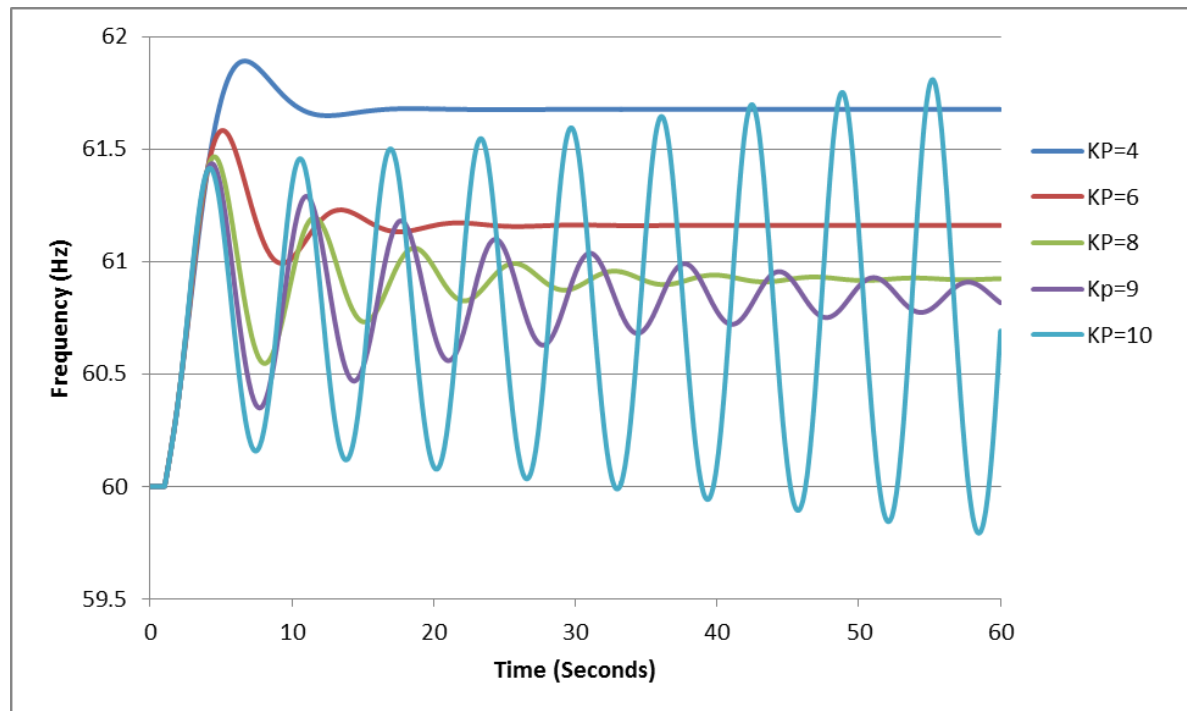


Case name: **B4\_PIDTuning**

# Tuning PID Controller Example



- Based on testing,  $K_u$  is about 9.5 and  $T_u$  is 6.4 seconds
- Using Ziegler-Nichols a good P value 4.75, is good PI values are  $K_P = 4.3$  and  $K_I = 0.8$ , while good PID values are  $K_P = 5.7$ ,  $K_I = 1.78$ ,  $K_D = 4.56$

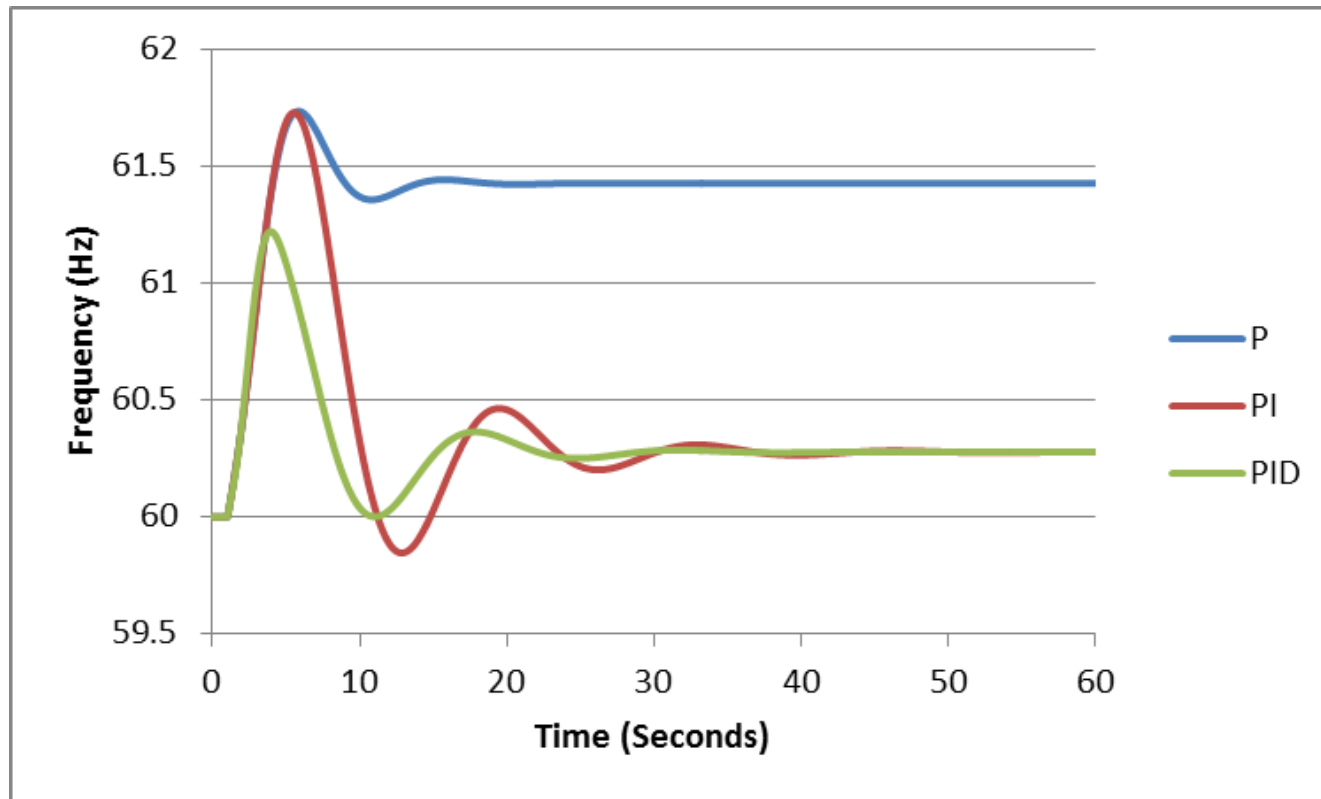


Further details on tuning are covered in IEEE Std. 1207-2011

# Tuning PID Controller Example



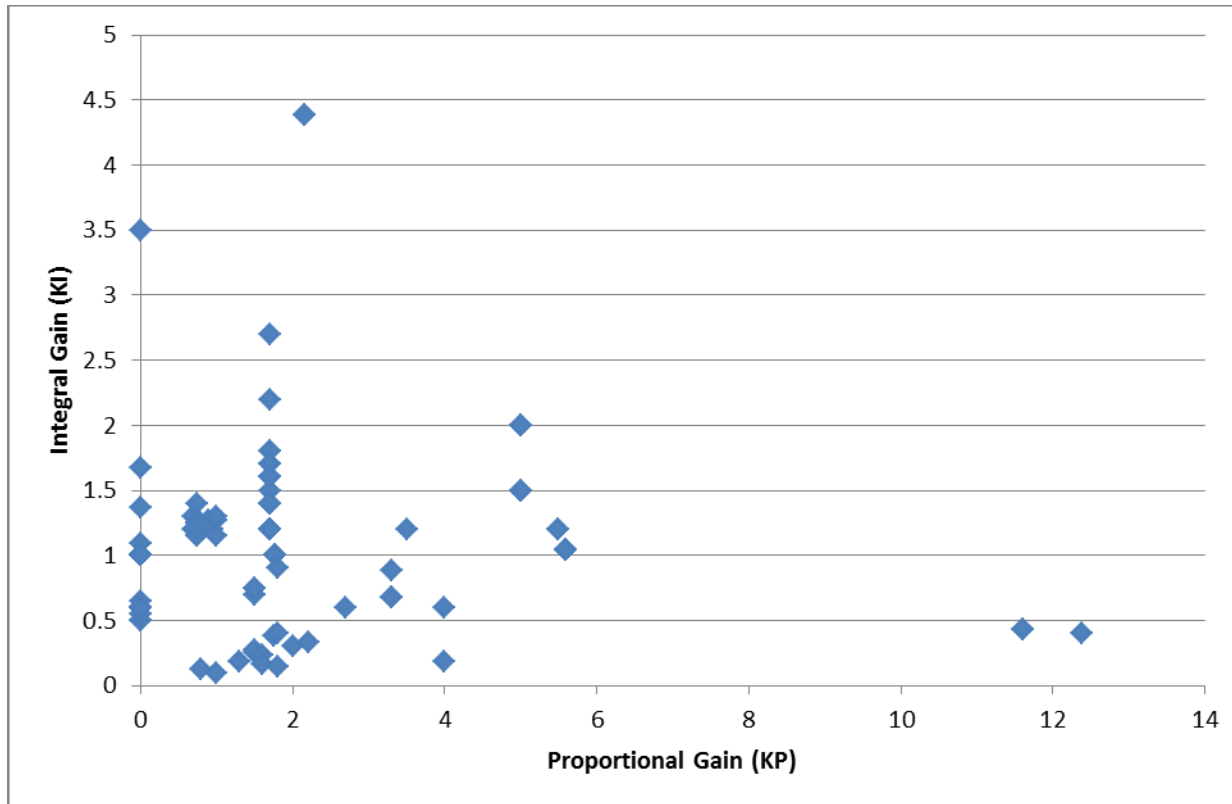
- Figure shows the Ziegler-Nichols for a P, PI and PID controls. Note, this is for stand-alone, not interconnected operation



# Example $K_I$ and $K_P$ Values



- Figure shows example  $K_I$  and  $K_P$  values from an actual system case



About 60% of the models also had a derivative term with an average value of 2.8, and an average  $T_D$  of 0.04 sec

# Non-windup Limits

---



- An important open question is whether the governor PI controllers should be modeled with non-windup limits
  - Currently models show no limit, but transient stability verification seems to indicate limits are being enforced
- This could be an issue if frequency goes low, causing governor PI to "windup" and then goes high (such as in an islanding situation)
  - How fast governor backs down depends on whether the limit winds up

# PI Non-windup Limits

- There is not a unique way to handle PI non-windup limits; the below shows two approaches from IEEE Std 421.5

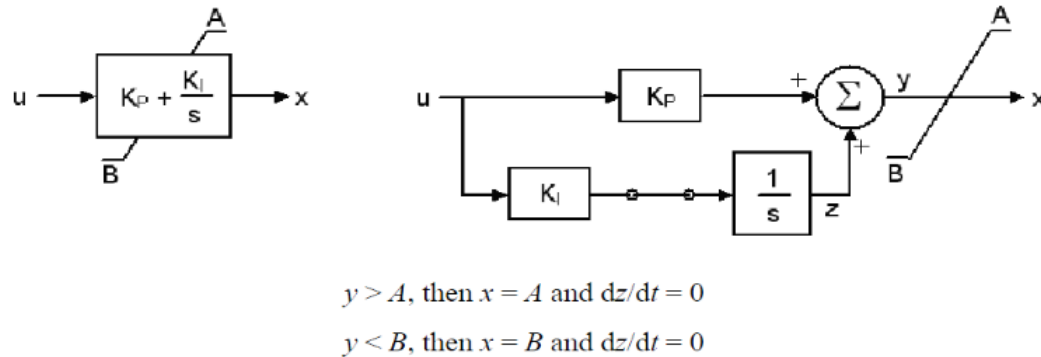


Figure E.7—Non-windup proportional-integral block

Another common approach is to cap the output and put a non-windup limit on the integrator

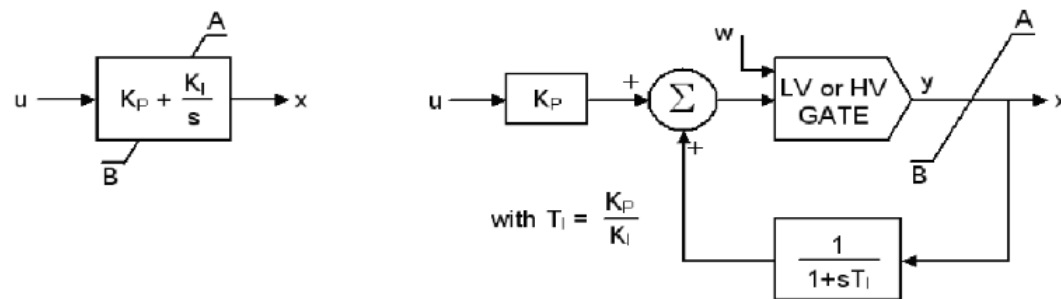


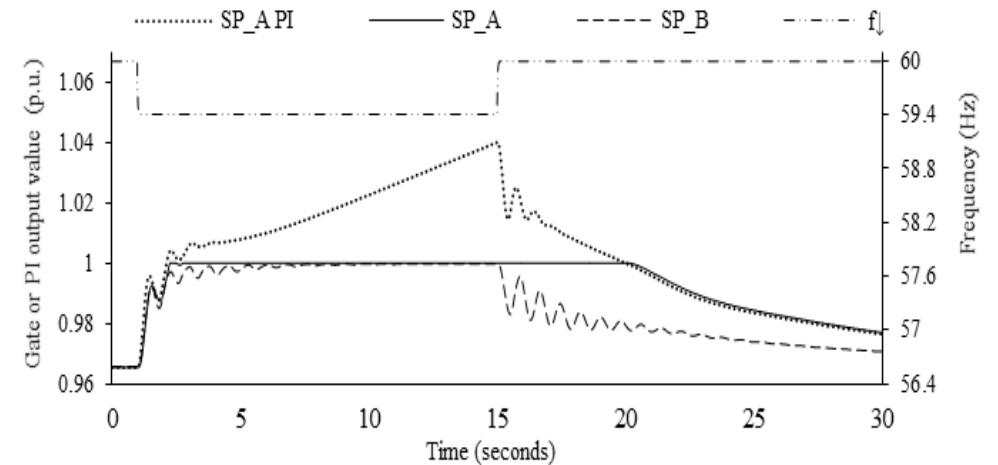
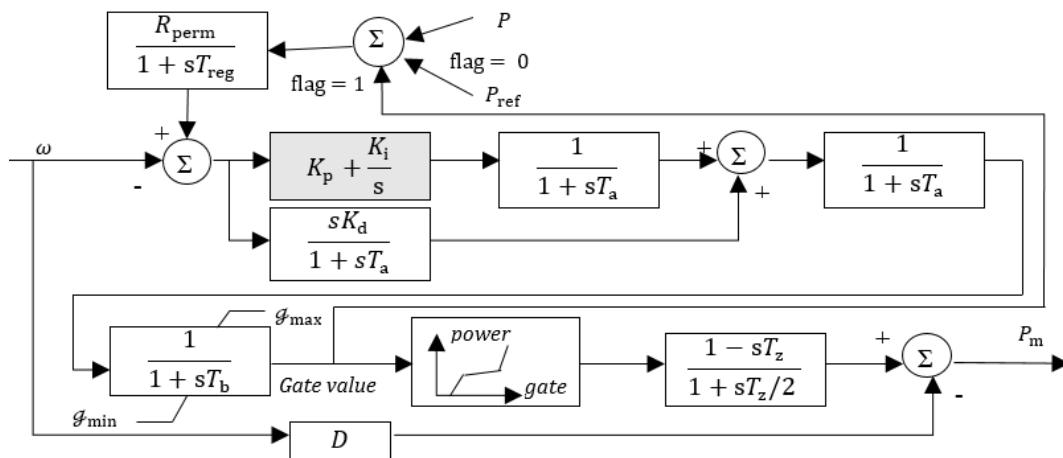
Figure E.8—Non-windup proportional-integral block



# PI Limit Problems with Actual Hydro Models



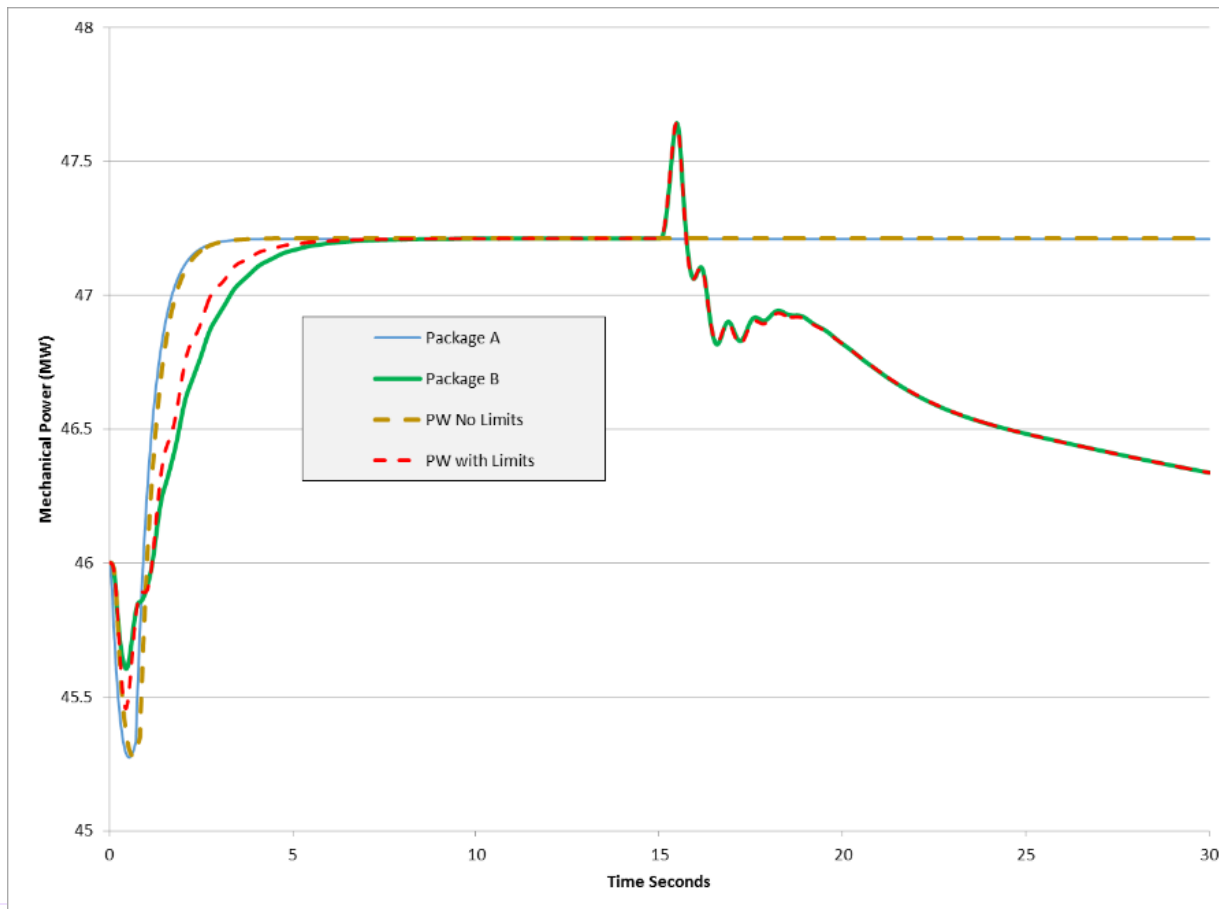
- A previous research project comparing transient stability packages found there were significant differences between hydro model implementations with respect to how PI limits were modeled
  - One package modeled limits but did not document them, another did not model them; limits were recommended at WECC MVWG in May 2014



# PIDGOV Model Results



- Below graph compares the  $P_{mech}$  response for a two bus system for a frequency change, between three transient stability packages



Packages A and B both say they have no governor limits, but B seems to; PowerWorld can do either approach

- GGOV1 is a relatively newer governor model, introduced in early 2000's by WECC for modeling thermal plants
  - Existing models greatly under-estimated the frequency drop
  - GGOV1 is now the most common WECC governor, used with about 40% of the units
- A useful reference is L. Pereira, J. Undrill, D. Kosterev, D. Davies, and S. Patterson, "A New Thermal Governor Modeling Approach in the WECC," *IEEE Transactions on Power Systems*, May 2003, pp. 819-829

# GGOV1: Selected Figures from 2003 Paper

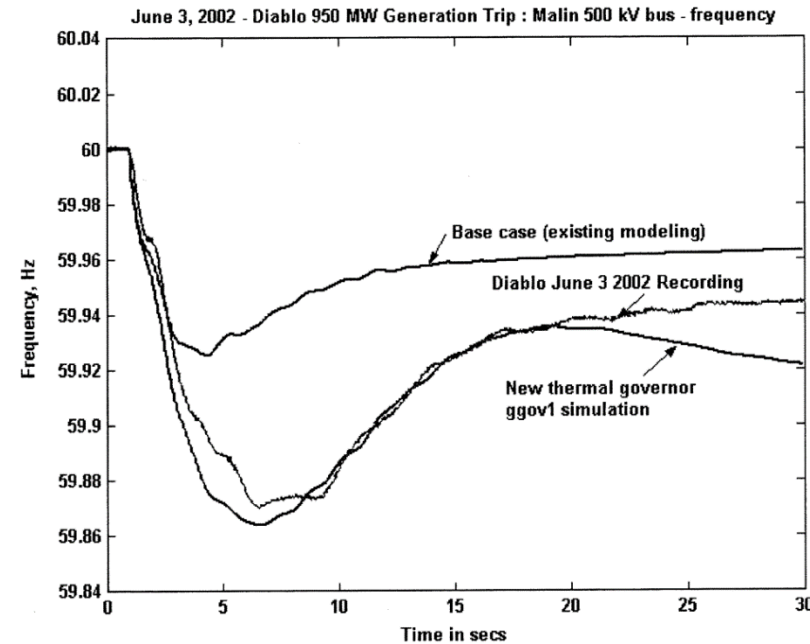
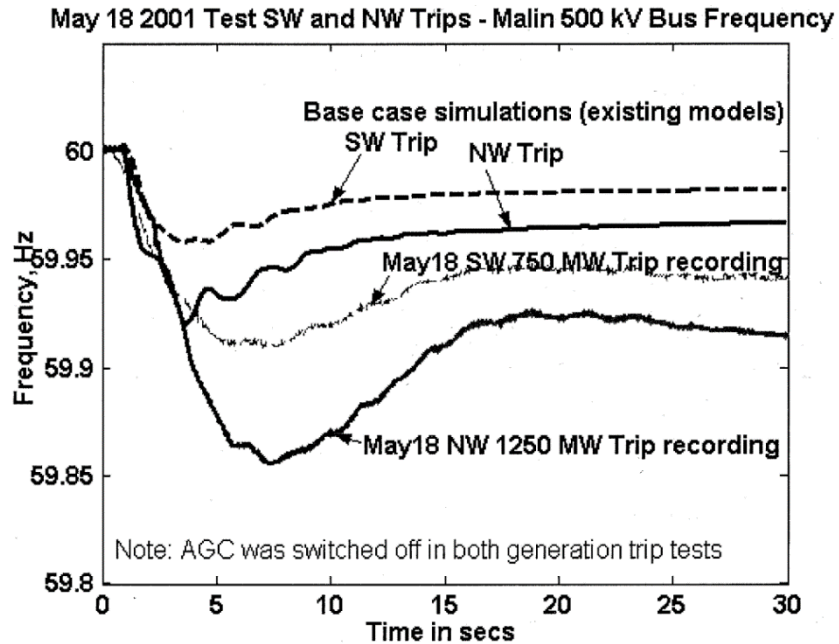


Fig. 1. Frequency recordings of the SW and NW trips on May 18, 2001. Also shown are simulations with existing modeling (base case).

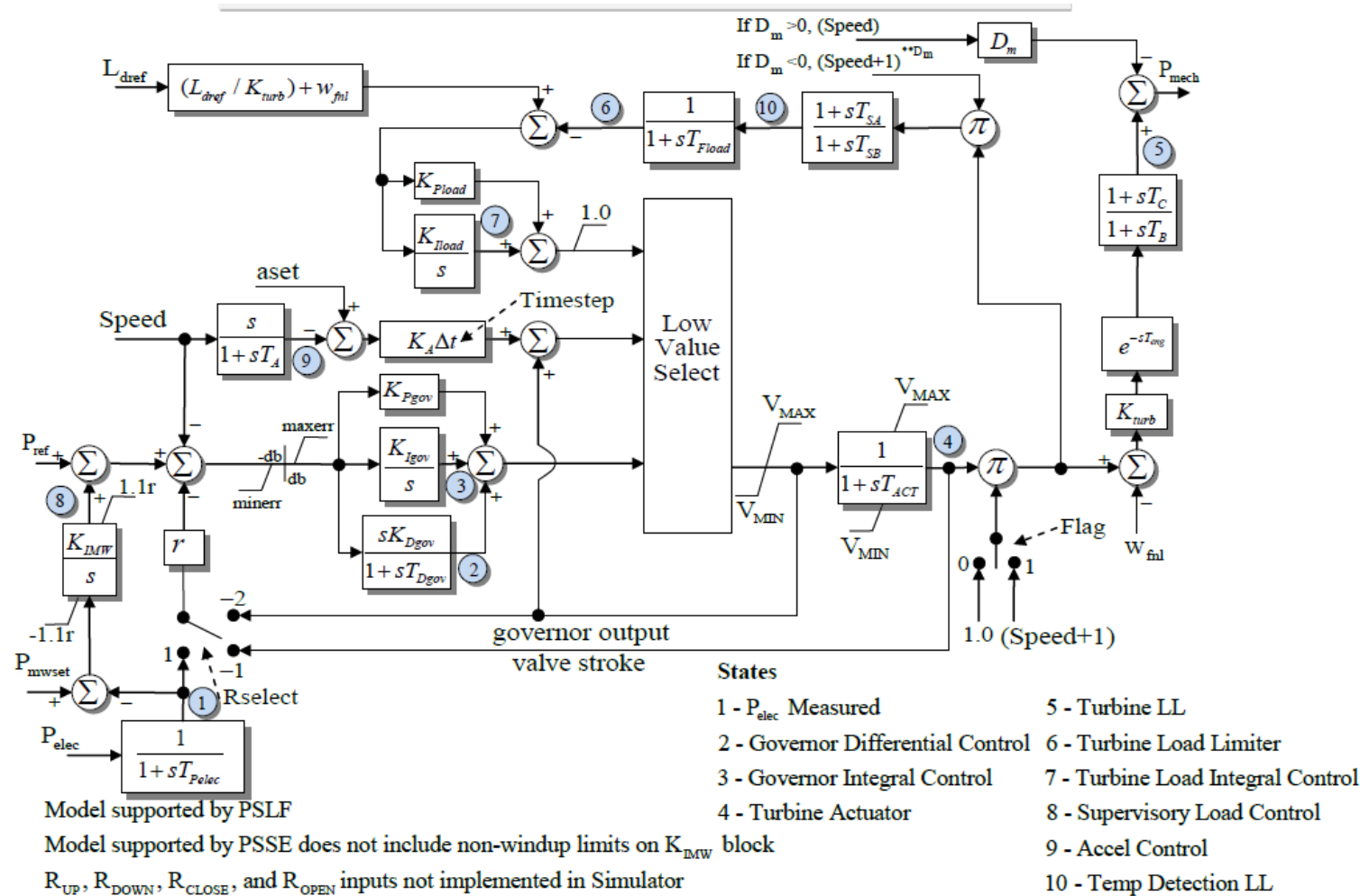
Governor model verification—950-MW Diablo generation trip on June 3, 2002.

Diablo Canyon is California's last nuclear plant; Unit 1 had been scheduled to shutdown in 2024 and Unit 2 in 2025. However, in March 2023 the NRC provided an exemption to PGE to operate both units past these dates (reference is California Senate Bill No. 846, which was signed on 9/2/2022)

# GGOV1 Block Diagram



GGOV1 and the related GGOV3 are the most common governors in WECC, with more than 40% in 2019



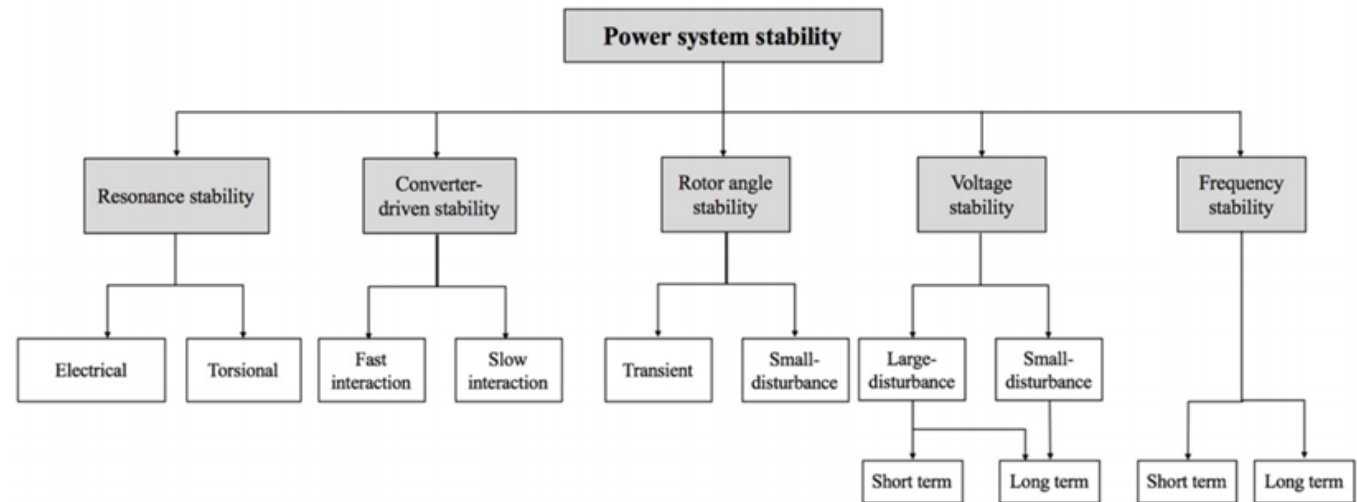
# Power System Stability Assessment



- As given in [1] the formal definition of power system stability is
  - “Power system stability is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact”
- The previously developed models will help in power system stability assessment
- Different techniques are used including time-domain simulations, eigenvalue analysis and power flow

# Power System Stability Terms

- The below image (from Figure 4 of [1]), and also shown lecture 2, helps define the terms



**Fig. 4. Classification of power system stability**

The two main time scales are the electromagnetic (left two branches) and the electromechanical (right three branches). The focus in 667 is mostly on the electromechanical time scale with ECEN 616 focusing on the electromagnetic.

# Transient Stability Multimachine Simulations

---



- Next, we'll be putting the models we've covered so far together
- 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 late 1950's



# A Little History

- A nice early reference is
  - Dyrkacz, Young, Maginniss, “A Digital Transient Stability Program Including the Effects of Regular, Exciter and Governor Response,” Proc. AIEE, Part 3, February 1961, pp. 1245-1254

This 1961 demonstrates results on a 96 bus system, shown below; note that the simulation is quite short, less than one second

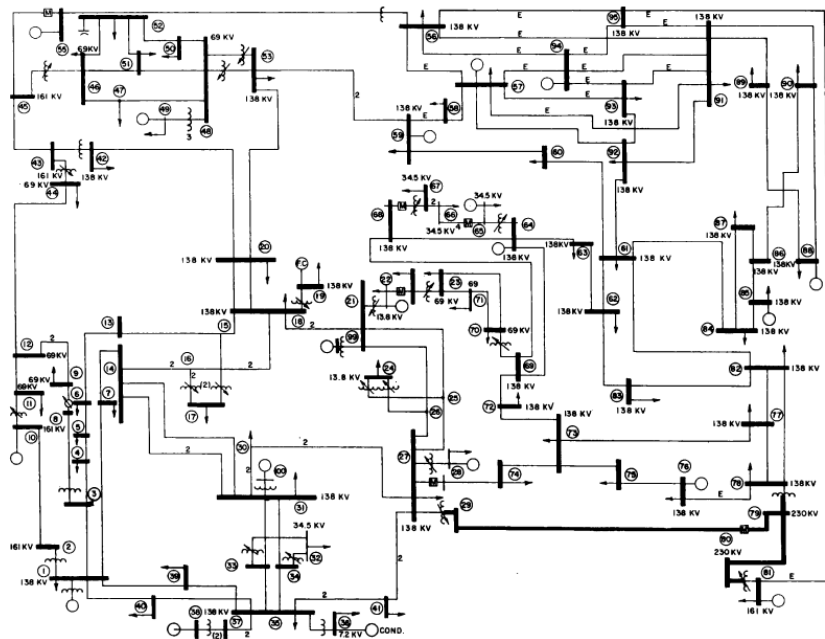
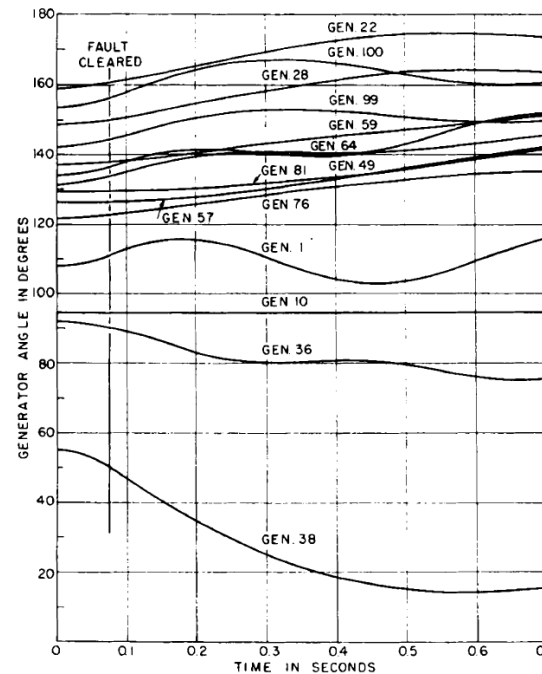


Fig. 4. One-line diagram of Union Electric Company system



# Power System 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

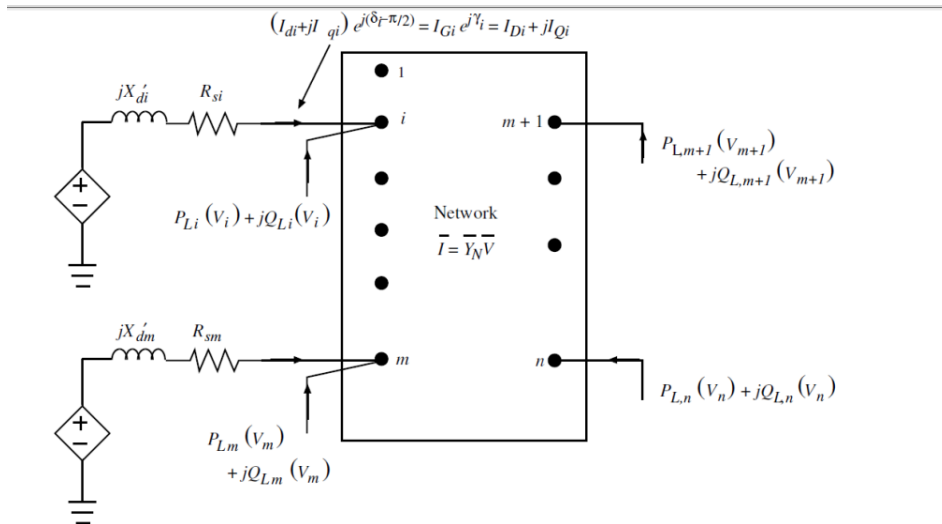


Figure 7.2: Interconnection of synchronous machine dynamic circuit and the rest of the network

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

# General Form



- The general form of the problem is solving

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u})$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y})$$

where  $\mathbf{x}$  is the vector of the state variables (such as the generator  $\delta$ 's),  $\mathbf{y}$  is the vector of the algebraic variables (primarily the bus complex voltages), and  $\mathbf{u}$  is the vector of controls (such as the exciter voltage setpoints)

# Stability Simulations General Solution



- General solution approach is
  - Solve power flow to determine initial conditions
  - Back solve to get initial states, starting with machine models, then exciters, governors, stabilizers, loads, etc
  - Set  $t = t_{\text{start}}$ , time step =  $\Delta t$ , abort = false
  - While ( $t \leq t_{\text{end}}$ ) and (not abort) Do Begin
    - Apply any contingency event
    - Solve differential and algebraic equations
    - If desired store time step results and check other conditions (that might cause the simulation to abort)
    - $t = t + \Delta t$
  - End while

# 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?

---



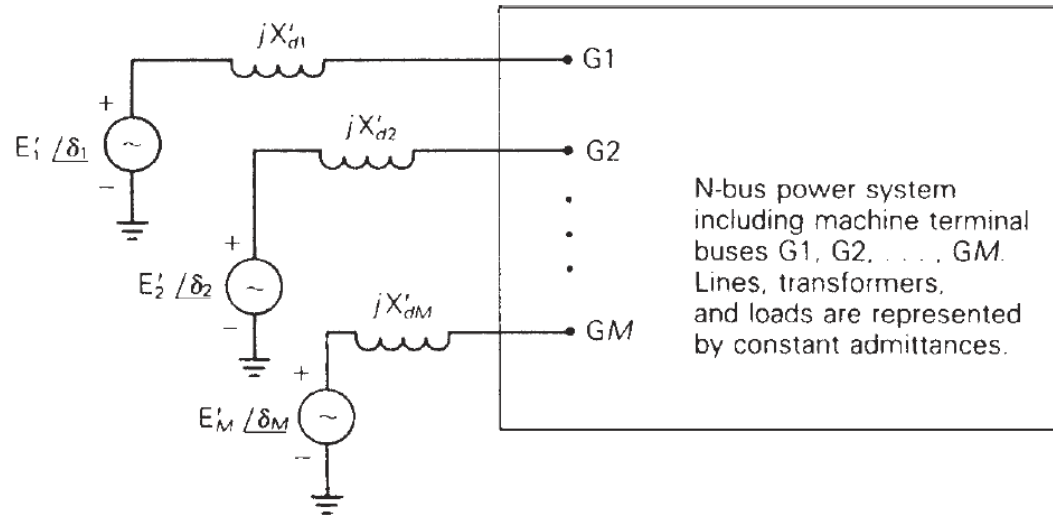
- The power flow equations were ultimately derived from

$$\mathbf{I}(\mathbf{x}, \mathbf{V}) = \mathbf{Y} \mathbf{V}$$

- 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

- To introduce the coupling between the machine models and the network constraints, consider a system modeled with just classical generators and impedance loads



In this example because we are using the classical model all values are on the network reference frame

We'll extend the figure slightly to include stator resistances,  $R_{s,i}$

# Algebraic Equations for Classical Model



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

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

Review Norton and Thevenin equivalents if you are rusty on them

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



# Swing Equation



- The first two differential equations for any synchronous machine correspond to the swing equation

$$\frac{d\delta_i}{dt} = \omega_i - \omega_s = \Delta\omega_i$$

$$\frac{2H_i}{\omega_s} \frac{d\omega_i}{dt} = \frac{2H_i}{\omega_s} \frac{d\Delta\omega_i}{dt} = T_{Mi} - T_{Ei} - D_i (\Delta\omega_i)$$

$$\text{with } T_{Ei} = \psi_{de,i} i_{qi} - \psi_{qe,i} i_{di}$$

# Swing Equation Speed Effects



- There is often confusion about these equations because of whether speed effects are included
  - Recognizing that often  $\omega \approx \omega_s$  (which is one per unit), some stability books have neglected speed effects
- For a rotating machine with a radial torque, power = torque times speed
- For a subtransient model

$$\bar{E}'' = \bar{V} + (R_s + jX'')\bar{I}, \quad E_d'' + jE_q'' = (-\psi_q'' + j\psi_d'')\omega$$

$$T_E = \psi_d'' I_q - \psi_q'' I_d \quad \text{and}$$

$$P_E = T_E \omega = (E_d'' + jE_q'')(I_d - jI_q) = E_d'' I_d + E_q'' I_q$$

# Classical Swing Equation



- Often in an introductory coverage of transient stability with the classical model the assumption is  $\omega \approx \omega_s$  so the swing equation for the classical model is given as

$$\frac{d\delta_i}{dt} = \omega_i - \omega_s = \Delta\omega_i$$

$$\frac{2H_i}{\omega_s} \frac{d\Delta\omega_i}{dt} = P_{Mi} - P_{Ei} - D_i (\Delta\omega_i)$$

$$\text{with } P_{Ei} = (E'_i \angle \delta_i) (E'_i \angle \delta_i - \bar{V}_i) Y_i$$

- We'll use this simplification for our initial example

As an example of this initial approach see Anderson and Fouad, *Power System Control and Stability*, 2<sup>nd</sup> Edition, Chapter 2 (with a newer version third edition of this book now available adding Vijay Vittal and Jim McCalley as authors).

# Numerical Solution

---



- There are two main approaches for solving

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u})$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y})$$

- 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

# Outline of the Solution Process



- The next group of slides will provide basic coverage of the solution process, partitioned explicit, then the simultaneous-implicit approach
- We'll start out with a classical model supplying an infinite bus, which can be solved by embedded the algebraic constraint into the differential equations

We'll start out just solving  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$

and then will extend to solving the full problem of

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u})$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y})$$

# Classical Swing Equation with Power Balance



- With a classical generator at bus  $i$  supplying an infinite bus with voltage magnitude  $V_{\text{inf}}$ , we can write the problem without algebraic constraints as

$$\frac{d\delta_i}{dt} = \omega_i - \omega_s = \Delta\omega_i = \Delta\omega_{i,pu} \omega_s$$

$$\frac{d\Delta\omega_{i,pu}}{dt} = \frac{1}{2H_i} \left( P_{Mi} - \frac{E'_i V_{\text{inf}}}{X_{th}} \sin \delta_i - D_i (\Delta\omega_{i,pu}) \right)$$

$$\text{with } P_{Ei} = \frac{E'_i V_{\text{inf}}}{X_{th}} \sin \delta_i$$

Note we are using the per unit speed approach

# Explicit Integration Methods

---



- As covered on the first day of class, there are a 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 but not too useful, 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

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

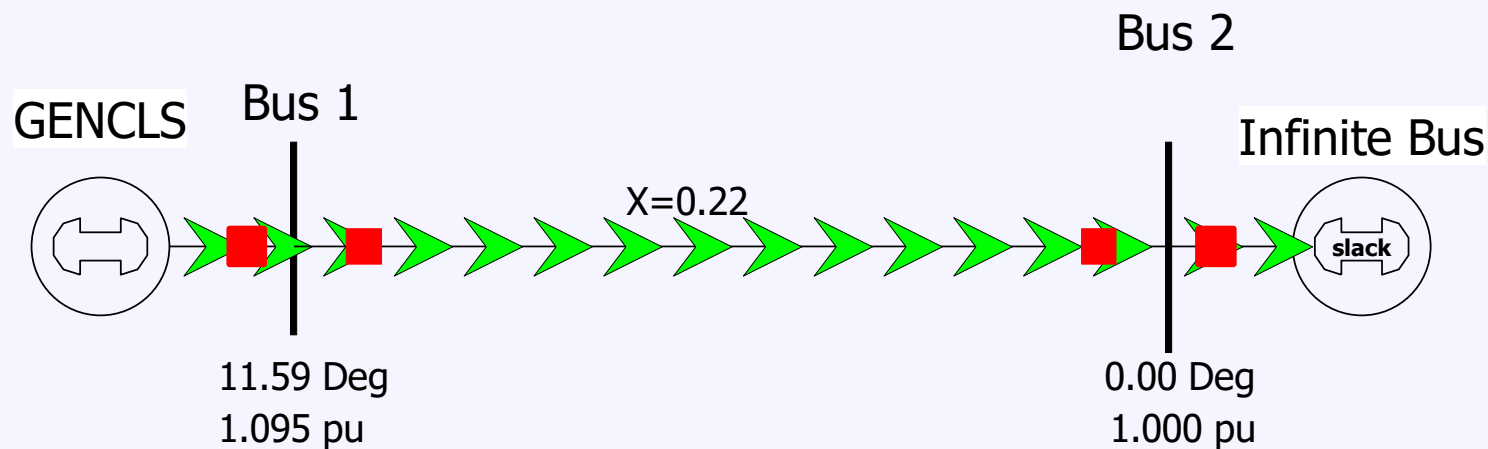
- Error with Euler's varies with the square of the time step



# Infinite Bus GENCLS Example using the Forward Euler's Method



- Use the four bus system from before, except now gen 4 is modeled with a classical model with  $X_d'=0.3$ ,  $H=3$  and  $D=0$ ; also we'll reduce to two buses with equivalent line reactance, moving the gen from bus 4 to 1



In this example  $X_{th} = (0.22 + 0.3)$ , with the internal voltage  $\bar{E}'_1 = 1.281 \angle 23.95^\circ$  giving  $E'_1 = 1.281$  and  $\delta_1 = 23.95^\circ$

# Infinite Bus GENCLS Example



- The associated differential equations for the bus 1 generator are

$$\frac{d\delta_1}{dt} = \Delta\omega_{1,pu} \omega_s$$

$$\frac{d\Delta\omega_{1,pu}}{dt} = \frac{1}{2 \times 3} \left( 1 - \frac{1.281}{0.52} \sin \delta_1 \right)$$

- The value of  $P_{M1} = 1$  is determined from the initial conditions, and would stay constant in this simple example without a governor
- The value  $\delta_1 = 23.95^\circ$  is readily verified as an equilibrium point (which is 0.418 radians)

# Infinite Bus GENCLS Example



- Assume a solid three phase fault is applied at the generator terminal, reducing  $P_{E1}$  to zero during the fault, and then the fault is self-cleared at time  $T^{\text{clear}}$ , resulting in the post-fault system being identical to the pre-fault system
  - During the fault-on time the equations reduce to

$$\frac{d\delta_1}{dt} = \Delta\omega_{1,pu} \omega_s$$

$$\frac{d\Delta\omega_{1,pu}}{dt} = \frac{1}{2 \times 3} (1 - 0)$$

That is, with a solid fault on the terminal of the generator, during the fault  $P_{E1} = 0$

# Euler's Solution



- The initial value of  $\mathbf{x}$  is

$$\mathbf{x}(0) = \begin{bmatrix} \delta_1(0) \\ \Delta\omega_{1,pu}(0) \end{bmatrix} = \begin{bmatrix} 0.418 \\ 0 \end{bmatrix}$$

- Assuming a time step  $\Delta t = 0.02$  seconds, and a  $T^{\text{clear}}$  of 0.1 seconds, then using Euler's

$$\mathbf{x}(0.02) = \begin{bmatrix} 0.418 \\ 0 \end{bmatrix} + 0.02 \begin{bmatrix} 0 \\ 0.1667 \end{bmatrix} = \begin{bmatrix} 0.418 \\ 0.00333 \end{bmatrix}$$

Note Euler's assumes  $\delta$  stays constant during the first time step

- Iteration continues until  $t = T^{\text{clear}}$

# Euler's Solution



- At  $t = T^{\text{clear}}$  the fault is self-cleared, with the equations changing to

$$\frac{d\delta}{dt} = \Delta\omega_{pu} \omega_s$$

$$\frac{d\Delta\omega_{pu}}{dt} = \frac{1}{6} \left( 1 - \frac{1.281}{0.52} \sin \delta \right)$$

- The integration continues using the new equations

# Euler's Solution Results ( $\Delta t=0.02$ )



- The below table gives the results using  $\Delta t = 0.02$  for the beginning time steps

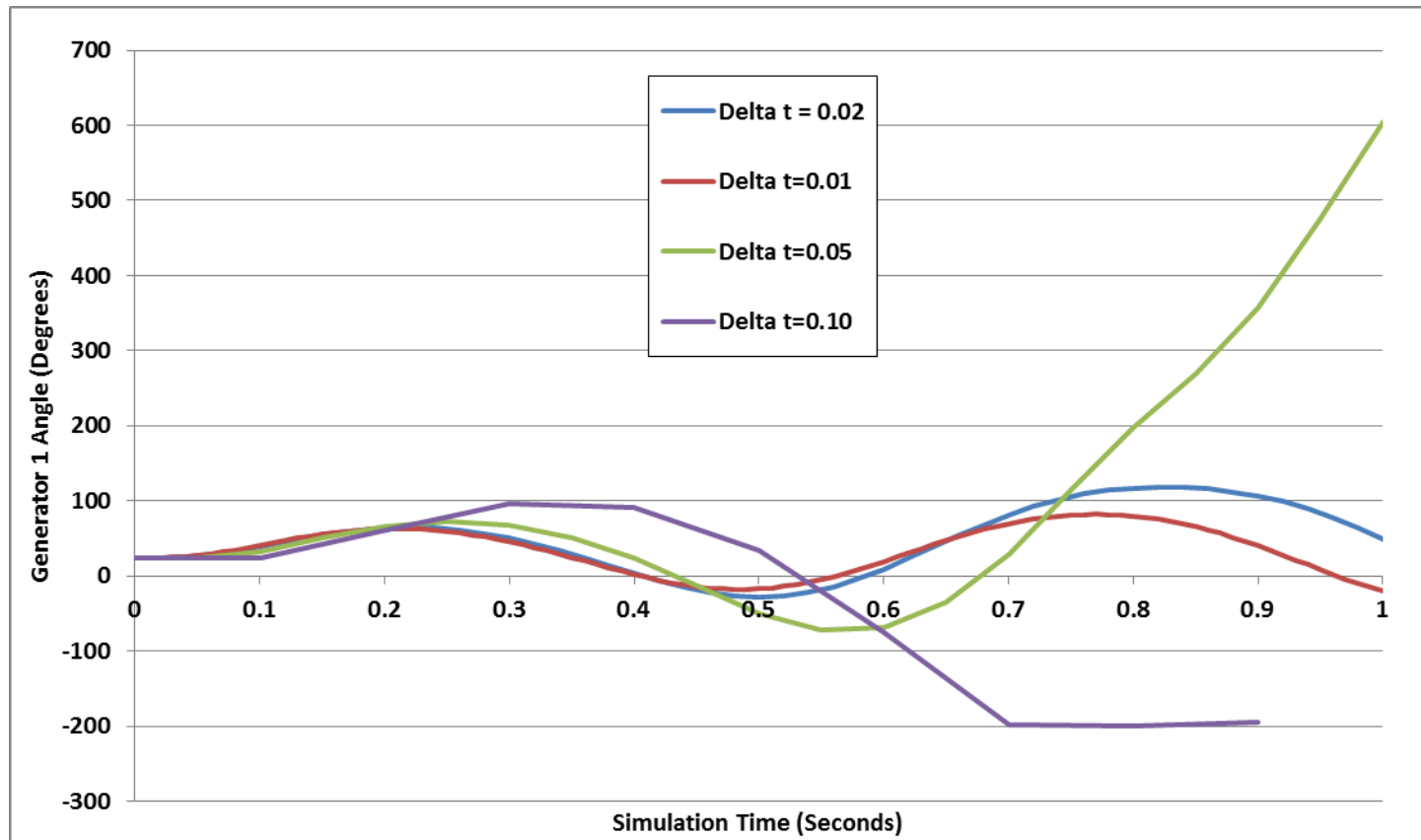
Time	Gen 1 Rotor Angle, Degrees	Gen 1 Speed (Hz)
0	23.9462	60
0.02	23.9462	60.2
0.04	25.3862	60.4
0.06	28.2662	60.6
0.08	32.5862	60.8
0.1	38.3462	61
0.1	38.3462	61
0.12	45.5462	60.8943
0.14	51.9851	60.7425
0.16	57.3314	60.5543
0.18	61.3226	60.3395
0.2	63.7672	60.1072
0.22	64.5391	59.8652
0.24	63.5686	59.6203
0.26	60.8348	59.3791
0.28	56.3641	59.1488

This is saved as PowerWorld case **B2\_CLS\_Infinite**. The integration method is set to Euler's on the **Transient Stability, Options, Power System Model** page

# Generator 1 Delta: Euler's



- The below graph shows the generator angle for varying values of  $\Delta t$ ; numerical instability is clearly seen



# Second Order Runge-Kutta



- Runge-Kutta methods improve on Euler's method by evaluating  $\mathbf{f}(\mathbf{x})$  at selected points over the time step
- One approach is a second order method (RK2) in which

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

where

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

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

This is also known as Heun's method or as the Improved Euler's or Modified Euler's Method

- That is,  $\mathbf{k}_1$  is what we get from Euler's;  $\mathbf{k}_2$  improves on this by reevaluating at the estimated end of the time step
- Error varies with the cubic of the time step



# Second Order Runge-Kutta (RK2)



- Again assuming a time step  $\Delta t = 0.02$  seconds, and a  $T^{\text{clear}}$  of 0.1 seconds, then using Heun's approach

$$\mathbf{x}(0) = \begin{bmatrix} \delta(0) \\ \Delta\omega_{pu}(0) \end{bmatrix} = \begin{bmatrix} 0.418 \\ 0 \end{bmatrix}$$

$$\mathbf{k}_1 = 0.02 \begin{bmatrix} 0 \\ 0.1667 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.00333 \end{bmatrix}, \quad \mathbf{x}(0) + \mathbf{k}_1 = \begin{bmatrix} 0.418 \\ 0.00333 \end{bmatrix}$$

$$\mathbf{k}_2 = 0.02 \begin{bmatrix} 1.257 \\ 0.1667 \end{bmatrix} = \begin{bmatrix} 0.0251 \\ 0.00333 \end{bmatrix}$$

$$\mathbf{x}(0.020) = \begin{bmatrix} 0.418 \\ 0 \end{bmatrix} + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2) = \begin{bmatrix} 0.431 \\ 0.00333 \end{bmatrix}$$

# RK2 Solution Results ( $\Delta t=0.02$ )



- The below table gives the results using  $\Delta t = 0.02$  for the beginning time steps

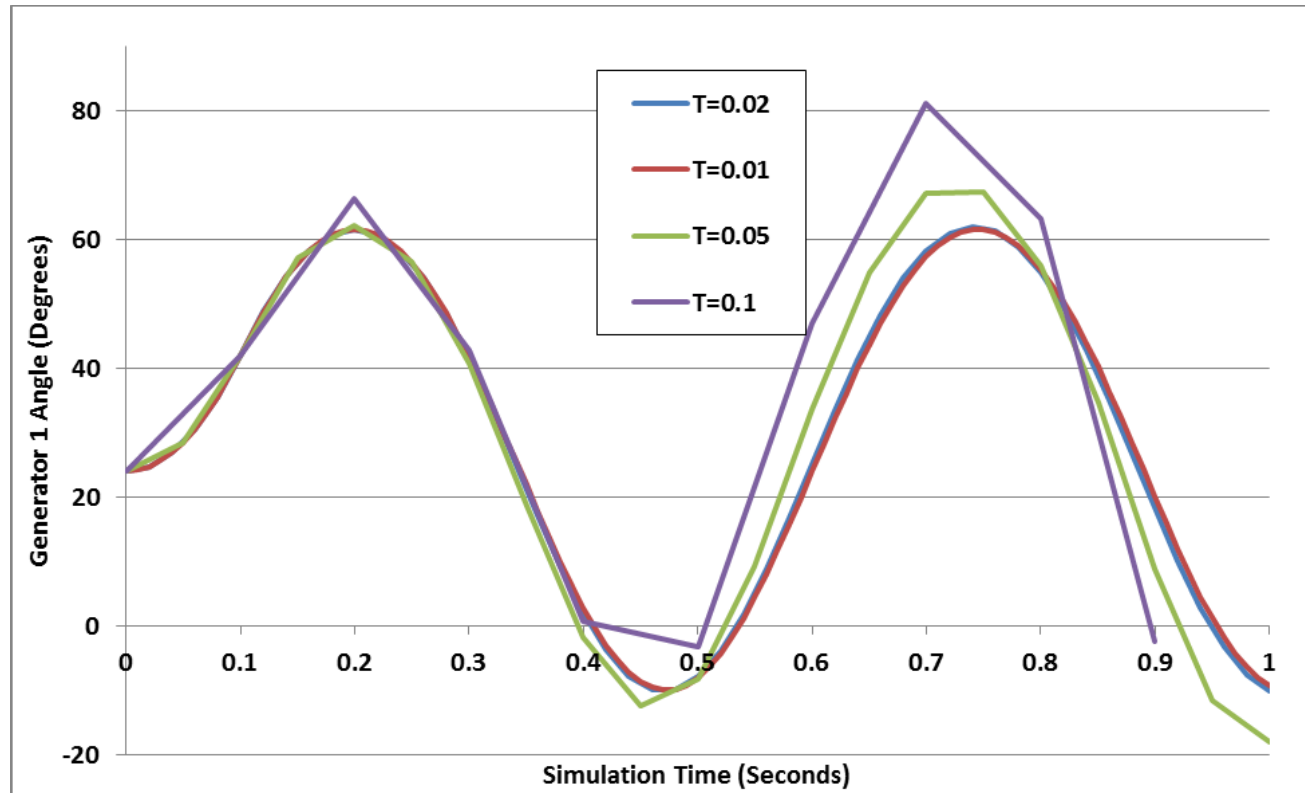
Time	Gen 1 Rotor Angle, Degrees	Gen 1 Speed (Hz)
0	23.9462	60
0.02	24.6662	60.2
0.04	26.8262	60.4
0.06	30.4262	60.6
0.08	35.4662	60.8
0.1	41.9462	61
0.1	41.9462	61
0.12	48.6805	60.849
0.14	54.1807	60.6626
0.16	58.233	60.4517
0.18	60.6974	60.2258
0.2	61.4961	59.9927
0.22	60.605	59.7598
0.24	58.0502	59.5343
0.26	53.9116	59.3241
0.28	48.3318	59.139

This is saved as PowerWorld case B2\_CLS\_Infinite. The integration method should be changed to Second Order Runge-Kutta on the **Transient Stability, Options, Power System Model** page

# Generator 1 Delta: RK2



- The below graph shows the generator angle for varying values of  $\Delta t$ ; much better than Euler's but still the beginning of numerical instability with larger values of  $\Delta t$



# Adding Network Equations



- Previous slides with the network equations embedded in the differential equations were a special case
- In general with the explicit approach we'll be alternating between solving the differential equations and solving the algebraic equations
- Voltages and currents in the network reference frame can be expressed using either polar or rectangular coordinates
- In rectangular with the book's notation we have

$$\bar{V}_i = V_{Di} + jV_{Qi}, \quad \bar{I}_i = I_{Di} + jI_{Qi}$$

# Adding Network Equations

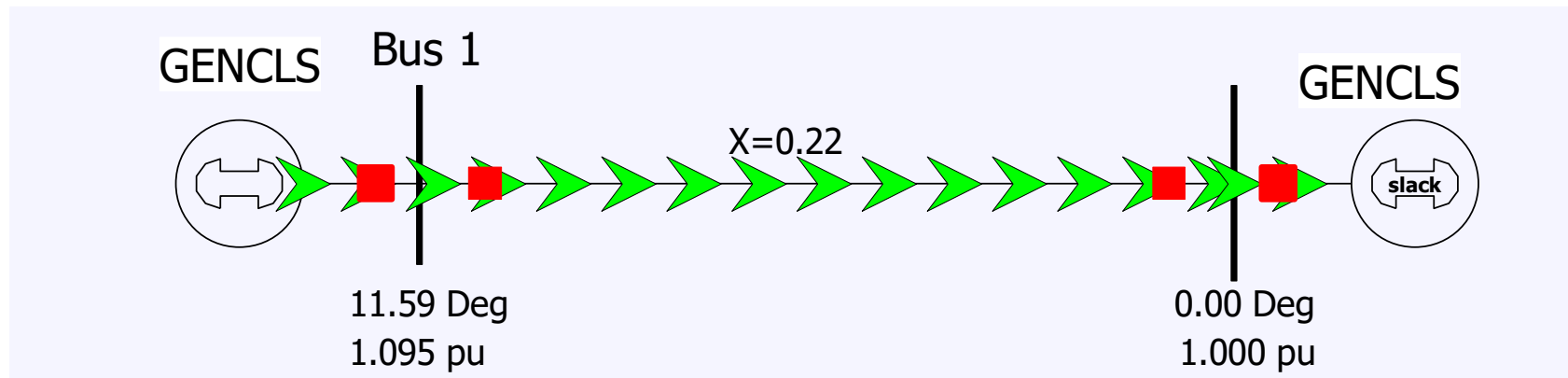


- Network equations will be written as  $\mathbf{Y} \mathbf{V} - \mathbf{I}(\mathbf{x}, \mathbf{V}) = \mathbf{0}$ 
  - Here  $\mathbf{Y}$  is as from the power flow, except augmented to include the impact of the generator's internal impedance
  - Constant impedance loads are also embedded in  $\mathbf{Y}$ ; non-constant impedance loads are included in  $\mathbf{I}(\mathbf{x}, \mathbf{V})$
- If  $\mathbf{I}$  is independent of  $\mathbf{V}$  then this can be solved directly:  $\mathbf{V} = \mathbf{Y}^{-1} \mathbf{I}(\mathbf{x})$
- In general an iterative solution is required, which we'll cover shortly, but initially we'll go with just the direct solution

# Two Bus Example, Except with No Infinite Bus



- To introduce the inclusion of the network equations, the previous example is extended by replacing the infinite bus at bus 2 with a classical model with  $X_{d2}'=0.2$ ,  $H_2=6.0$



PowerWorld Case is **B2\_CLS\_2Gen**

# Bus Admittance Matrix



- The network admittance matrix is

$$\mathbf{Y}_N = \begin{bmatrix} -j4.545 & j4.545 \\ j4.545 & -j4.545 \end{bmatrix}$$

- This is augmented to represent the Norton admittances associated with the generator models ( $X_{d1}'=0.3$ ,  $X_{d2}'=0.2$ )

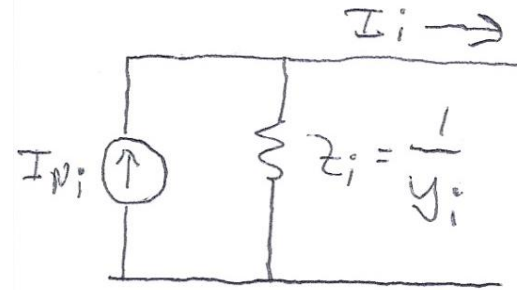
$$\mathbf{Y} = \mathbf{Y}_N + \begin{bmatrix} \frac{1}{j0.3} & 0 \\ 0 & \frac{1}{j0.2} \end{bmatrix} = \begin{bmatrix} -j7.879 & j4.545 \\ j4.545 & -j9.545 \end{bmatrix}$$

In PowerWorld you can see this matrix by selecting **Transient Stability, States/Manual Control, Transient Stability Ybus**

# Current Vector

- For the classical model the Norton currents are given by

$$\bar{I}_{Ni} = \frac{E'_i \angle \delta_i}{R_{s,i} + jX'_{d,i}}, \quad Y_i = \frac{1}{R_{s,i} + jX'_{d,i}}$$



- The initial values of the currents come from the power flow solution
- As the states change ( $\delta_i$  for the classical model), the Norton current injections also change



# B2\_CLS\_Gen Initial Values



- The internal voltage for generator 1 is as before

$$\bar{I} = 1 - j0.3286$$

$$\bar{E}_1 = 1.0 + (j0.22 + j0.3)\bar{I} = 1.1709 + j0.52 = 1.281 \angle 23.95^\circ \quad 0.4179 \text{ radians}$$

- We likewise solve for the generator 2 internal voltage

$$\bar{E}_2 = 1.0 - (j0.2)\bar{I} = 0.9343 - j0.2 = 0.9554 \angle -12.08 \quad 0.2108 \text{ radians}$$

- The Norton current injections are then

$$\bar{I}_{N1} = \frac{1.1709 + j0.52}{j0.3} = 1.733 - j3.903$$

$$\bar{I}_{N2} = \frac{0.9343 - j0.2}{j0.2} = -1 - j4.6714$$

Keep in mind the Norton current injections are not the current out of the generator

# B2\_CLS\_Gen Initial Values



- To check the values, solve for the voltages, with the values matching the power flow values

$$\mathbf{V} = \begin{bmatrix} -j7.879 & j4.545 \\ j4.545 & -j9.545 \end{bmatrix}^{-1} \begin{bmatrix} 1.733 - j3.903 \\ -1 - j4.671 \end{bmatrix}$$
$$= \begin{bmatrix} 1.072 + j0.22 \\ 1.0 \end{bmatrix}$$

# Swing Equations



- With the network constraints modeled, the swing equations are modified to represent the electrical power in terms of the generator's state and current values

$$P_{Ei} = E_{Di} I_{Di} + E_{Qi} I_{Qi}$$

$I_{Di} + jI_{Qi}$  is the current being injected into the network by the generator

- Then swing equation is then

$$\frac{d\delta_i}{dt} = \Delta\omega_{i,pu} \omega_s$$

$$\frac{d\Delta\omega_{i,pu}}{dt} = \frac{1}{2H_i} \left( P_{Mi} - \left( E_{Di} I_{Di} + E_{Qi} I_{Qi} \right) - D_i \left( \Delta\omega_{i,pu} \right) \right)$$

# Two Bus, Two Generator Differential Equations



- The differential equations for the two generators are

$$\frac{d\delta_1}{dt} = \Delta\omega_{1,pu} \omega_s$$

$$\frac{d\Delta\omega_{1,pu}}{dt} = \frac{1}{2H_1} \left( P_{M1} - (E_{D1}I_{D1} + E_{Q1}I_{Q1}) \right)$$

$$\frac{d\delta_2}{dt} = \Delta\omega_{2,pu} \omega_s$$

$$\frac{d\Delta\omega_{2,pu}}{dt} = \frac{1}{2H_2} \left( P_{M2} - (E_{D2}I_{D2} + E_{Q2}I_{Q2}) \right)$$

In this example

$$P_{M1} = 1 \text{ and } P_{M2} = -1$$

# PowerWorld GENCLS Initial States



Transient Stability Analysis - Case: B2\_CLS\_21

File Case Information Draw Onelines Tools Options Add Ons Window

Edit Mode Abort Primal LP SCOPF... OPF Case Info OPF Options and Results... PV... QV... Refine Model ATC... Transient Stability... Stability Case Info GIC... Scheduled Actions... Topology Processing

Run Mode Log Script Log Optimal Power Flow (OPF) PV and QV Curves (PVQV) ATC Transient Stability (TS) GIC Schedule Topology Process

Simulation Status Initialized

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

Select Step

- Simulation
- Options
- Result Storage
- Plots
- Results from RAM
- Transient Limit Monitors
- States/Manual Control
  - All States
  - State Limit Violations
  - Generators
  - Buses
  - Transient Stability YBus
  - GIC GMatrix
  - Two Bus Equivalents
  - Detailed Performance Resu
- Validation
- SMTR Finenvalues

States/Manual Control

Reset to Start Time Transfer Present State to Power Flow Save Case in P

Run Until Specified Time 0.000000 Run Until Time Restore Reference Power Flow Model

Do Specified Number of Timestep(s) 1 Number of Timesteps to Do Save Time Snapshot

All States State Limit Violations Generators Buses Transient Stability YBus GIC GMatrix Two Bus Equivalents Detailed Performance Results

	Model Class	Model Type	Object Name	At Limit	State Ignored	State Name	Value	Derivative	Delta X K1
1	Gen Synch. Ma	TXGENCLS	1 (Bus 1) #1		NO	Angle	0.4179	0.0000000	0.0000000
2	Gen Synch. Ma	TXGENCLS	1 (Bus 1) #1		NO	Speed w	0.0000	0.0000000	0.0000000
3	Gen Synch. Ma	TXGENCLS	2 (Bus 2) #1		NO	Angle	-0.2109	0.0000000	0.0000000
4	Gen Synch. Ma	TXGENCLS	2 (Bus 2) #1		NO	Speed w	0.0000	0.0000000	0.0000000