SANDIA REPORT

SAND2015-5233
Unlimited Release
Printed June 2015

Application Note: Power Grid Modeling With *Xyce*™

Peter Sholander

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-5233
Unlimited Release
Printed June 2015

Application Note:
Power Grid Modeling With *Xyce*™

Peter Sholander and Who Else?
Electrical Models and Simulation
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1177

Abstract

This application note describes how to model steady-state power flows and transient events in electric power grids with the SPICE-compatible *Xyce*™ Parallel Electronic Simulator developed at Sandia National Labs. This application notes provides a brief tutorial on the basic devices (branches, bus shunts, transformers and generators) found in power grids. The focus is on the features supported and assumptions made by the **Xyce** models for power grid elements. It then provides a detailed explanation, including working **Xyce** netlists, for simulating some simple power grid examples such as the IEEE 14-bus test case.

Contents

List of Figures

1. Introduction

This application note describes how to model steady-state power flows and transient events in electric power grids with the ***Xyce***™ Parallel Electronic Simulator developed at Sandia National Labs, where the models assume a balanced system and positive sequence components. This application note provides a brief tutorial on the basic devices (branches, bus shunts, transformers and generators) found in power grids. The focus is on the features supported and assumptions made by the models for power grid elements in **Xyce** 6.3. It then provides a detailed explanation, including working **Xyce** netlists, for simulating some simple power grid examples such as the IEEE 14-bus test case [1]. A tutorial on the basics of Xyce usage for circuit simulation can be found in the Xyce Users' Guide [2].

As background, Xyce is an open-source circuit simulation tool that was developed at Sandia National Labs, starting in 1999, in support of the Advanced Supercomputing (ASC) program. It is "SPICE-compatible" [3], with additional extensions to support the unique requirements of Sandia's customers. The software is coded primarily in C++, with additional support from external math and solver libraries such as Trilinos [4]. It supports serial execution on Windows, OSX and Linux and parallel execution on OSX and Linux. It runs on multi-core desktops, commodity clusters and capacity machines. Recently, Sandia has extended **Xyce** to model steady-state power flows and transient events in electric power grids.

1.1 Target Audience

This application note is intended for both new and existing users of **Xyce** who wish to use it to simulate steady-state power flows and transient events in electric power grids.

1.2 Prerequisites

This application note assumes that you have already downloaded and compiled ***Xyce***™ according to its documentation, that you have installed it in a manner that allows you to run it directly by typing "Xyce" [1] in the command line, and that you are able to run a basic netlist using that installed copy of **Xyce**.

For external open-source users, source code for **Xyce** can be obtained from our website at xyce.sandia.gov. Pre-compiled binaries for Linux, OSX and Windows are available for Sandia

[1] Or perhaps "runxyce" if you are a Sandia user who has installed one of our precompiled binary installations.

internal users at info.sandia.gov/xyce.

The **Xyce** Reference Guide [5] and Users' Guide [2] provide more detail on **Xyce** syntax and usage for circuit simulation. Readers who are not familiar with SPICE or **Xyce** are encouraged to work through the tutorial examples in Chapters 2 and 3 of the Users' Guide before trying to run the examples given in this application note. Those two chapters explain how to run both steady-state[2] (.DC) and transient (.TRAN) simulations in **Xyce**, using a simple Diode Clipper circuit as an example. Reference [6] contains a brief explanation of the mathematical foundations of parallel circuit simulation in **Xyce**. All of these documents are available on our website at xyce.sandia.gov.

The **Xyce** Users' Guide may be useful for power engineers who are not familiar with SPICE-compatible circuit simulators. The mathematical foundations document may be useful for power engineers who are familiar with other solution techniques for simulating power flows and transient events in electric power grids. More extensive details on electric power grids and power grid simulation can be found in [7] and [8].

[2]In SPICE-compatible circuit simulators, a ".DC analysis" or ".DC simulation" calculates the initial conditions (or "DC Operating Point" (DCOP)) for the circuit. It is the analog of a steady-state power flow calculation in a power grid analysis. This circuit simulator terminology may be confusing for power engineers since "DC analysis" has a completely different meaning in that domain. A ".TRAN analysis" in a circuit simulator then models the subsequent time-dependent behavior of the circuit.

2. Overview of Power Grid Devices

This chapter provides a brief overview of the devices found in electric power grids. It focuses on the idealized devices found in the IEEE 14-bus test case [1]. The goal is to provide a brief tutorial for existing **Xyce** users who are not familiar with electric power grids.

For power engineers, this chapter outlines the features supported and assumptions used by Version 6.3 of the **Xyce** device models for steady-state and transient modeling of power grids. This is the initial release of the power grid device models in **Xyce**, so the expectation is that additional features will be supported in later **Xyce** releases. The current release schedule for **Xyce** is approximately every six months.

The IEEE 14-bus test case requires the following idealized device models for a steady-state power flow calculation, where the models assume a balanced system and positive sequence components. The remainder of this chapter describes how these devices can be realized in **Xyce**. It some cases, this can be done with standard SPICE (and **Xyce**) models. In other cases, a **Xyce** "device model" for a power grid element was coded in C++, and the details of that model are described in more detail. The next two chapters then provide detailed examples of how these models can be used for steady-state and transient simulations of power grids. These examples include working netlists that can be run in Version 6.3 of **Xyce**.

- Slack bus, where the Voltage Magnitude ($|V|$ or VM) and Phase (Θ or `Th`) are held fixed and supply the voltage reference for the grid. The Real Power (P) and the Reactive Power (Q) are unconstrained. For this device, the convention is that "injected power" into the grid is positive. In many cases, the voltage at the slack bus is arbitrarily set to ($|V|$,Θ)=(1,0) in "per unit". However, this is not always done in all of the IEEE test cases. (Note: For a description of the "per unit" measurement system, please consult [7].) Finally, the slack bus is also called the "swing bus".
- Branch Model, using a Π-model (e.g., Figure 11.1 on pg. 264 of [8] and figure 2.1 of this document) with parameters for resistance (R), reactance (X) and shunt susceptance (B). The **Xyce** branch model omits the shunt conductance.
- Bus Shunt Model, with parameters for shunt conductance (G) and shunt susceptance (B).
- Generator Bus, where the Voltage Magnitude ($|V|$) and the Real Power (P) injection are held fixed. For this device, the convention is that "injected power" into the grid is positive. Ideally, minimum and maximum limits on the Reactive Power (Q) injection should also be enforced, although this is not required for the IEEE 14-bus test-case [1]. If one of the Q-limits is being enforced, then $|V|$ becomes unconstrained and the generator bus then holds both P and Q fixed. (Note: For a more complete discussion of Q-limits, see the discussion on pp. 250-254

of [8].) While reactive power limits are not supported in **Xyce** 6.3, as discussed in Section 2.6, this is an important feature that should be supported in future releases for both generator buses and synchronous condensers.

- Transformer model, with at least parameters for resistance (R), reactance (X) and a fixed turns ratio (TR). (For example circuit diagrams, see Figure 11.5 on pg. 273 of [8] and figure 2.3 in this document.) For the IEEE 14-bus test case [1], modeling of the magnetizing susceptance (B) or tap-shifting (variable turns ratio) is not required. So, these last two features are not supported in **Xyce** 6.3. A "complex turns ratio" with a fixed phase shift is supported.
- Constant Power Load, where P and Q are held fixed. Even though this is a "load", P and Q can be either positive or negative. For this device, power flow out of the grid is considered "positive". (Note: The other common loads models are constant impedance, constant current and ZIP loads. These loads can be modeled in **Xyce** but are not discussed in this application note.)
- Synchronous Condenser, where the Real Power (P) flow into the network is zero (or nearly zero), and the Voltage Magnitude ($|V|$) is held fixed. Ideally, there should be maximum/minimum limits on the reactive power flow (Q), although this is not required for the IEEE 14-bus test case. For this device, power flow into the grid is considered "positive".

Other requirements and desirements were:

- For research purposes, the simulations should support all three formulations of the power flow equations, which are called I=YV, PQ Polar and PQ Rectangular formats in this application note. This was done because each solution format may have different convergence and stability properties for different grid configurations. In addition, one format (e.g., PQ Polar since the source and load models are simple) may be better for calculating the initial steady-state conditions while another format (e.g., $I = YV$ since the branch equations are linear rather than quadratic or transcendental) may be more optimal for the subsequent transient simulations with more realistic time-domain models for the generating units. The power flow equations for PQ Rectangular are then quadratic with simpler source and load models than $I = YV$ format. The equations used for each **Xyce** device model for each solution format are given in the **Xyce** Reference Guide [5]. However, for convenience those equations are also included in this application note. For I=YV format, the solution variables are the real and imaginary parts of the voltages (`VR` and `VI`) and currents (`IR` and `II`). For the PQ Rectangular format, the solution variables are the real and imaginary parts of the voltages (`VR` and `VI`) and the real and reactive power flows (`P` and `Q`). For the PQ Polar format, the solution variables are the voltage magnitude ($|V|$) and phase (Θ) and the real and reactive power flows (`P` and `Q`).
- There must be an easy way to measure the voltage, phase, real power and reactive power at each grid node. This did not mean that support for lead currents had to be in the initial implementation in **Xyce** 6.3.
- It should be possible to auto-generate a Xyce .cir file (also called a "netlist") from the formats commonly used to describe power grids (e.g., the IEEE Common Data Format (.cdf file), which is described at www.ee.washington.edu/research/pstca/formats/cdf.txt).

- The **Xyce** results should be validated against theoretical results, other open source tools (such as Power System Toolbox (PST)) and/or other results given in textbooks [7].

2.1 Slack Bus

The slack bus does not require a **Xyce** device model, for simple examples like the IEEE 14-bus test case [1]. It can be modeled by two voltage sources (either the real and imaginary voltages (`VR` and `VI`) or the voltage magnitude ($|V|$) and phase (Θ or `Th`)) applied to the inputs of the slack bus.

Zero-voltage ammeters can then be used to measure the current and/or power flows, since the power grid devices do not support the branch current accessors (`I()`) in **Xyce** 6.3. In all three cases, the ammeters are wired so that positive power (or current) flows into the grid. Note that for **Xyce** voltage sources, positive current flows from the positive node through the source to the negative node. For Power Grid slack buses and generator buses, that convention is reversed. Positive current/power flows into the grid from the slack bus or generator bus.

For PQ Polar format, an example is as follows where ($|V|$,Θ) at the slack bus is fixed at (1.06,0) in "per unit" (pu), as in the IEEE 14-bus test case. The zero-voltage ammeters are measuring the real and reactive power flow (P and Q) in this case. In **Xyce**, the ground node is denoted by either `GND` or node 0. The subsequent `.PRINT` line would use `I(Vamm1P)`, `I(Vamm1Q)` to print out the real and reactive power flows and `V(Bus1VM)` and `V(Bus1Th)` to print out the voltage magnitude and phase.

```
V1Th Bus1Th ammBus1P 0V
V1VM Bus1VM ammBus1Q 1.06V
Vamm1P 0 ammBus1P 0V
Vamm1Q 0 ammBus1Q 0V
```

For PQ Rectangular format, an example is as follows. The zero-voltage ammeters are measuring real and reactive power flow (P and Q) in this case.

```
V1R Bus1R ammBus1P 1.06V
V1I Bus1I ammBus1Q 0V
Vamm1P 0 ammBus1P 0V
Vamm1Q 0 ammBus1Q 0V
```

For the I=YV format, an example is as follows. The ammeters are measuring real and imaginary current flow (`IR` and `II`) in this case. As detailed in Section 3.3.2 an "expression" would then be used to print out the real and reactive power flows on the subsequent `.PRINT` line.

```
V1R Bus1R ammBus1R 1.06V
V1I Bus1I ammBus1I 0V
Vamm1R 0 ammBus1R 0V
Vamm1I 0 ammBus1I 0V
```

In all three cases, the zero-voltage ammeters are not strictly necessary and they do introduce additional solution variables. Their advantage is that they simplify the print lines and subsequent data interpretation. The voltage sources could be wired directly between the bus terminals (e.g., Bus1Th) and ground (node 0). However, in that case, the user would have to remember the current conventions in **Xyce** and, in order to show a positive power injection into the grid as a positive quantity in the PQ Polar example, the `.PRINT` statement would either have to use `-I(V1TH)` or the terminals on the voltage source `V1TH` would have to be reversed.

As another note, this application note uses a consistent naming convention throughout for bus and device names. For example, `Bus1Th` is the voltage angle at Bus 1. This convention greatly simplifies the automated generation of the **Xyce** netlists from industry standard formats such as IEEE Common Data Format (.cdf) files.

2.2 Constant Power Load

The constant power load also did not initially need a **Xyce** device model for simple power grid examples such as the IEEE 14-bus test case [1]. For the PQ Rectangular and PQ Polar formats, a constant power load can be modeled with two current sources. The convention is that a "load" has positive power flowing out of the grid. (Note: The other common loads models are constant impedance, constant current and ZIP loads. These loads can be modeled in **Xyce** but are not discussed in this application note.)

An example of a P=0.2 and Q=0.1 load at Bus 2 for PQ Polar (where the loads are given in per unit) is as follows. The real and reactive power dissipated can then be printed out as `I(ILoad2P)` and `I(ILoad2Q)`.

```
ILoad2P bus2Th 0 0.2
ILoad2Q bus2VM 0 0.1
```

For PQ Rectangular format, it is as follows. Again, the real and reactive power dissipated can be printed out as `I(ILoad2P)` and `I(ILoad2Q)`.

```
ILoad2P bus2R 0 0.2
ILoad2Q bus2I 0 0.1
```

For I=YV format, a subcircuit can be defined using the **Xyce** non-linear dependent source (B-device). The constant power loads (CPL) can then be repetitively defined via that subcircuit definition. In this example, upper and lower bounds of 1000 and -1000 (in per unit) are imposed on the real and imaginary currents (`I(BloadR)` and `I(BloadI)`) since $P/|V|$ goes to infinity for small values of $|V|$. This has been found to work empirically but the value of `CurrLim` might have to be adjusted in some cases to ensure that the DC Operating Point (DCOP) calculation converges. (Note: Consult the **Xyce** Users' Guide [2] and Reference Guide [5] for more details on the syntaxes for subcircuit definitions and **Xyce** expressions.)

```
XLoad2 bus2R bus2I CPL PARAMS: P=0.2 Q=0.1
```

```
.SUBCKT CPL RNode INode PARAMS: P=0.5 Q=0.0 CurrLim=1000
* Ammeter at load is defined so that positive power flows into the
* load from the bus
VammR ammR 0 0V
VammI ammI 0 0V
BloadR RNode ammR
+ I={limit((P*V(RNode)+Q*V(INode))/(V(RNode)*V(RNode)+V(INode)*V(INode)),
+ -CurrLim,CurrLim)}
BloadI INode ammI
+ I={limit((P*V(INode)-Q*V(RNode))/(V(RNode)*V(RNode)+V(INode)*V(INode)),
+ -CurrLim,CurrLim)}
.ENDS
```

Printing the real and reactive power flows for the CPL subcircuit is slightly more complicated than for the PQ Polar and PQ Rectangular cases, since the solution variables are now voltages and currents. An example `.PRINT` line to just print out the voltages and power flows at Bus 2 would be as follows (where $+$ as the first character on a line indicates a "continuation" of the previous line in **Xyce** or SPICE):

```
.PRINT DC width=10 precision=6 V(bus2R) V(bus2I)
+ {V(bus2R)*I(Xload2:VammR)+V(bus2I)*I(Xload2:VammI)}
+ {V(bus2I)*I(Xload2:VammR)-V(bus2R)*I(Xload2:VammI)}
```

2.3 Power Grid Branch and Bus Shunt

In theory, it is possible to model power grid branches and bus shunts with standard **Xyce** non-linear dependent sources and sub-circuit definitions. However, that approach was found to be somewhat unstable in larger grid models. So, the **Xyce** developers decided to code "device models" for those two devices and the generator bus and transformer models described below.

It is also possible to combine the branch and bus shunt models into one device model. However, they were split out into separate models for two reasons. First, the bus and branch devices are separate entities in industry formats like IEEE Common Data Format (.cdf) files. So, this approach matched that particular grid-specification format. The second, and more practical reason, is that one bus might have one bus shunt and several branches attached to it. The separation of the branch and bus shunt models made it easier to auto-generate the **Xyce** .cir files for that configuration.

Equivalent Real Forms

In circuit simulators, the Jacobian matrix can be assembled in parallel via each device's "element stamp", which describes its voltage-current relationship (for example, see [9] and pp. 257-258 of

[7]). For a simple resistor $I = gV$, and the element stamp is:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} g & -g \\ -g & g \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{2.1}$$

Power grid elements (e.g., non-generative elements like transmission line branches) can be modeled as generalized 4-port impedances. However, an "Equivalent Real Form" (ERF) must then be used to make the complex-valued voltage-current and power-flow equations compatible with the real-valued solvers used by **Xyce**. The equations given below use a K1 ERF [10], which solves the complex-valued system of equations $I = YV$ as follows. Let $Y = (g + jb)$, $V = (V_R + jV_I)$ and $I = (I_R + jI_I)$. Then the equivalent set of real-valued equations for the generalized 4-port impedance is:

$$\begin{bmatrix} I_{1R} \\ I_{2R} \\ I_{1I} \\ I_{2I} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & -b_{11} & -b_{12} \\ g_{21} & g_{22} & -b_{21} & -b_{22} \\ b_{11} & b_{12} & g_{11} & g_{12} \\ b_{21} & b_{22} & g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} V_{1R} \\ V_{2R} \\ V_{1I} \\ V_{2I} \end{bmatrix} \tag{2.2}$$

Similar 4-port models can be developed for the source/generator devices. Those generalized element stamps can then be used to form a Jacobian matrix that can be solved in parallel by **Xyce**'s existing solution techniques [6].

Y Matrices for Power Grid Branch and Bus Shunt

Figure 2.1 shows a lumped Π model for a power grid branch. The Y-Matrix for the `PowerGridBranch` device can then be expressed as follows where $A = (R + jX)^{-1}$, R is the branch resistance, X is the branch reactance and B is the branch shunt susceptance given on the device's instance line:

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} g_{11} + jb_{11} & g_{12} + jb_{12} \\ g_{21} + jb_{21} & g_{22} + jb_{22} \end{bmatrix} = \begin{bmatrix} A & -A + 0.5j * B \\ -A + 0.5j * B & A \end{bmatrix} \tag{2.3}$$

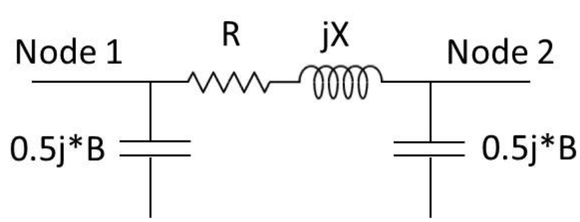

Figure 2.1. Lumped Π Model for PowerGridBranch.

Figure 2.2 shows an equivalent circuit for a power grid bus shunt. The Y-Matrix for the `PowerGridBusShunt` device can then be expressed as follows where G is the bus shunt conductance and B is the bus shunt susceptance given on the device's instance line:

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} g_{11} + jb_{11} & g_{12} + jb_{12} \\ g_{21} + jb_{21} & g_{22} + jb_{22} \end{bmatrix} = \begin{bmatrix} G + jB & -G - jB \\ -G - jB & G + jB \end{bmatrix} \tag{2.4}$$

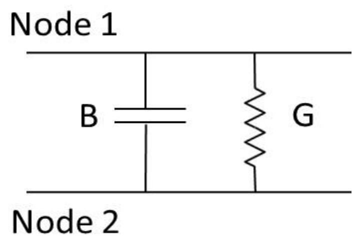

Figure 2.2. Equivalent Circuit for PowerGridBusShunt.

Equations Common to Power Grid Branch and Bus Shunt

The `PowerGridBranch` and `PowerGridBusShunt` devices use the same basic equations to model voltage and current flow or voltage and power flow. The differences are in the exact formulation of the Y-Matrixes described above. There are three options for the equations used, namely I=YV, PQ Polar and PQ Rectangular.

For the I=YV format, the device equations for the `PowerGridBranch` and `PowerGridBusShunt` devices are as follows, where the g_{ij} and b_{ij} terms are given above. Also, V_{R1} and V_{I1} are the real and imaginary parts of the voltage at terminal 1. I_{R1} and I_{I1} are the real and imaginary parts of the current at terminal 1.

$$\begin{bmatrix} I_{R1} \\ I_{R2} \\ I_{I1} \\ I_{I2} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & -b_{11} & -b_{12} \\ g_{21} & g_{22} & -b_{21} & -b_{22} \\ b_{11} & b_{12} & g_{11} & g_{12} \\ b_{21} & b_{22} & g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} V_{R1} \\ V_{R2} \\ V_{I1} \\ V_{I2} \end{bmatrix} \tag{2.5}$$

For the PQ Rectangular format, the device equations are non-linear [8]. Define P_1 and Q_1 as the real and reactive power flows into the grid at terminal 1.

$$P_1 = g_{11}(V_{R1}^2 + V_{I1}^2) + V_{R1}(g_{12} * V_{R2} - b_{12} * V_{I2}) + V_{I1}(b_{12} * V_{R2} + g_{12} * V_{I2}) \tag{2.6}$$
$$P_2 = g_{22}(V_{R2}^2 + V_{I2}^2) + V_{R2}(g_{21} * V_{R1} - b_{21} * V_{I1}) + V_{I2}(b_{21} * V_{R1} + g_{21} * V_{I1}) \tag{2.7}$$
$$Q_1 = -b_{11}(V_{R1}^2 + V_{I1}^2) + V_{I1}(g_{12} * V_{R2} - b_{12} * V_{I2}) + V_{R1}(b_{12} * V_{R2} + g_{12} * V_{I2}) \tag{2.8}$$
$$Q_2 = -b_{22}(V_{R2}^2 + V_{I2}^2) + V_{I2}(g_{21} * V_{R1} - b_{21} * V_{I1}) + V_{R2}(b_{21} * V_{R1} + g_{21} * V_{I1}) \tag{2.9}$$

For the PQ Polar format, the device equations are also non-linear [8]. Define $|V_1|$ as the voltage magnitude at terminal 1 and Θ_1 as the voltage angle at terminal 1.

$$P_1 = g_{11} * |V_1|^2 + |V_1| * |V_2| * (g_{12} * cos(\Theta_1 - \Theta_2) + b_{12} * sin(\Theta_1 - \Theta_2)) \quad (2.10)$$
$$P_2 = g_{22} * |V_2|^2 + |V_2| * |V_1| * (g_{21} * cos(\Theta_2 - \Theta_1) + b_{21} * sin(\Theta_2 - \Theta_1)) \quad (2.11)$$
$$Q_1 = -b_{11} * |V_1|^2 + |V_1| * |V_2| * (g_{12} * sin(\Theta_1 - \Theta_2) - b_{12} * cos(\Theta_1 - \Theta_2)) \quad (2.12)$$
$$Q_2 = -b_{22} * |V_2|^2 + |V_2| * |V_1| * (g_{21} * sin(\Theta_2 - \Theta_1) - b_{21} * cos(\Theta_2 - \Theta_1)) \quad (2.13)$$

2.4 Power Grid Transformer

Equations for Power Grid Transformer

The equations for the `PowerGridTransformer` device are similar to those used by the `PowerGridBranch` and `PowerGridBusShunt` devices. The circuit diagram for a power grid transformer is shown in Figure 2.3. It implements a fixed, complex turns ratio. Tap changing is not supported in **Xyce** 6.3.

For the I=YV and PQ Rectangular formats, the equations are the same as for the `PowerGridBranch` and `PowerBusBusShunt` devices. However, the Y-Matrix is no longer symmetric [7]. Let $A = (R+jX)^{-1}$, R be the resistance, X be the reactance, n be the turns ratio (which is the `TR` instance parameter) and ϕ be the phase shift in radians (which is the `PS` instance parameter). Also, let the effective complex turns ratio be $r = m + jp = n * (cos(\phi) + j * sin(\phi))$. Then the Y-Matrix is given by:

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} g_{11} + jb_{11} & g_{12} + jb_{12} \\ g_{21} + jb_{21} & g_{22} + jb_{22} \end{bmatrix} = \begin{bmatrix} A * (m^2 + p^2)^{-1} & -A * (m - jp)^{-1} \\ -A * (m + jp)^{-1} & A \end{bmatrix} \quad (2.14)$$

The voltage-current and power flow equations for the I=YV and PQ Rectangular formats are then the same as for the `PowerGridBranch` and `PowerGridBusShunt` devices, with the modified Y-matrix parameters given above.

For the PQ Polar format, the Y matrix is also not symmetric and is given by [8]:

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} g_{11} + jb_{11} & g_{12} + jb_{12} \\ g_{21} + jb_{21} & g_{22} + jb_{22} \end{bmatrix} = \begin{bmatrix} A * n^{-2} & -A * n^{-1} \\ -A * n^{-1} & A \end{bmatrix} \quad (2.15)$$

The power flow equations for PQ Polar format are then:

$$P_1 = g_{11} * |V_1|^2 + |V_1| * |V_2| * (g_{12} * cos(\Theta_1 - \Theta_2 - \phi) + b_{12} * sin(\Theta_1 - \Theta_2 - \phi)) \quad (2.16)$$
$$P_2 = g_{22} * |V_2|^2 + |V_2| * |V_1| * (g_{21} * cos(\Theta_2 - \Theta_1 + \phi) + b_{21} * sin(\Theta_2 - \Theta_1 + \phi)) \quad (2.17)$$
$$Q_1 = -b_{11} * |V_1|^2 + |V_1| * |V_2| * (g_{12} * sin(\Theta_1 - \Theta_2 - \phi) - b_{12} * cos(\Theta_1 - \Theta_2 - \phi)) \quad (2.18)$$
$$Q_2 = -b_{22} * |V_2|^2 + |V_2| * |V_1| * (g_{21} * sin(\Theta_2 - \Theta_1 + \phi) - b_{21} * cos(\Theta_2 - \Theta_1 + \phi)) \quad (2.19)$$

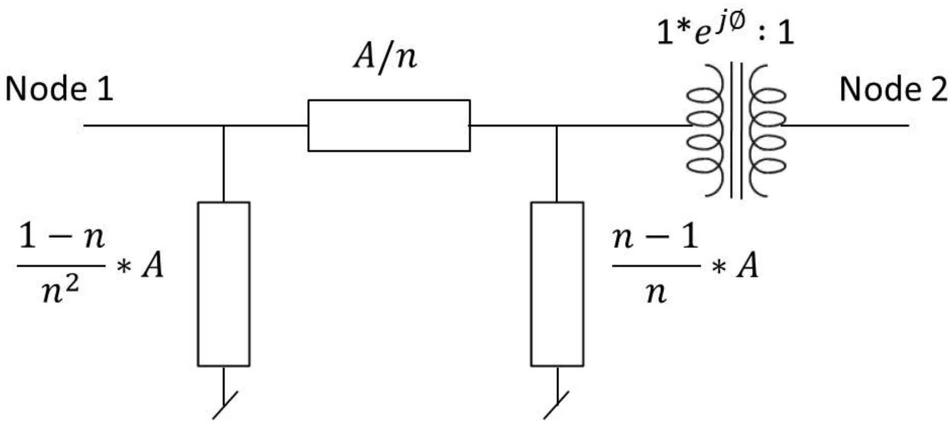

Figure 2.3. Equivalent Circuit for PowerGridTransformer.

2.5 Power Grid Generator Bus

Equations for Power Grid Gen Bus

The `PowerGridGenBus` is an active device that functions as an ideal generator with a fixed power output (P) and voltage magnitude (VM). The device equations for the PQ Polar format are as follows [8]. The other solution formulations are not supported in **Xyce** 6.3. Reactive power limits (`QMax` and `QMin`) are also not supported in **Xyce** 6.3.

$$P_1 = P \tag{2.20}$$

$$|V_1| = VM \tag{2.21}$$

The convention for power grids is that positive power from generators is injected into the grid. So, positive real (`P`) and reactive power (`Q`) flows out of the positive terminals. This is reversed from the normal convention for current direction for voltage and current sources in either **Xyce** or SPICE.

2.6 Summary of Power Grid Model Limitations in Xyce 6.3

The following features are not supported by the existing **Xyce** device models and the 6.3 release of the **Xyce** Power Grid device models.

- Reactive Power (`QMax` and `QMin`) limits in the Generator Bus device model.
- Tap-changing transformers with a variable turns ratio and/or a variable phase shift. In IEEE CDF files, these transformers are termed “Variable turns ratio for voltage control, Type 2”,

"Variable turns ratio for MVAR control, Type 3" and "Variable phase angle for MW control, Type 4".

- Magnetizing susceptance for transformers.

Additional limitations are:

- The Generator Bus device model only supports the PQ Polar format.
- A stable way of modelling remote controlled buses in larger grids, with standard **Xyce** device models rather than custom device models, is still under study.

This chapter has described how the elements of the IEEE 14-bus test case [1] can be modeled with standard **Xyce** current and voltage sources, in conjunction with the **Xyce** device models for the power grid branches, bus shunts, transformers and generator buses. The next chapter will show how to combine these models into a working simulation for the steady-state power flow in the IEEE 14-bus test case, using the PQ Polar solution format. Simple three-bus examples are also given for the I=YV and PQ Rectangular solution formats.

3. Steady-State Power Flow

This chapter provides an annotated **Xyce** netlist for the IEEE 14-bus test case [1], using the PQ Polar format for the power-flow equations. For the PQ Polar case, the solution variables are the voltage angle (Θ or `Th`), voltage magnitude ($|V|$ or `VM`), real power (`P`) and reactive power (`Q`) at each bus. The chapter concludes with simple three-bus examples in the I=YV and PQ Rectangular solution formats.

The netlists given in this chapter are not necessarily the most concise possible syntaxes for each example. Instead, they illustrate key points in translating between industry-standard formats such as IEEE Common Data Format (.cdf) files and a **Xyce** netlist. Sandia is currently developing Python- and Matlab-based tools that can auto-generate a .cir file from various industry standard formats used for describing power grids. Those conversion tools can be provided tor Sandia-internal users at this point.

3.1 General Comments

This subsection provides general comments on **Xyce** usage for simulating electric power grids. It expands on the comments in the annotated netlists given later in this chapter.

3.1.1 Node and Device Naming Conventions

The user is free to use any naming convention for the device and node names, consistent with the **Xyce** requirements on legal characters in those names. In addition, device names must be unique for each device type, while node names must be unique within the entire netlist.

For the purposes of auto-generating the .cir files, this application note uses the following conventions for device and node names:

- They are in the form `pg<fromBusID>_<toBusID>`, or `pg1_2`, for the branch between buses 1 and 2.
- If duplicate branches exist between two buses then the netlist must account for that by (for example) denoting one device as `pg<fromBusID>_<toBusID>_line1` and the other as `pg<fromBusID>_<toBusID>_line2`.
- Nodes names follow a similar convention. They are of the form `Bus<BusID>Th` and `Bus<BusID>VM`, with specific examples of `bus1Th` and `bus1VM` for the Θ and $|V|$ terminals at bus 1.

3.1.2 Device Syntax

The Xyce Reference Guide [5] gives a detailed description of the device syntax for each power grid device. However, an overview is as follows. In **Xyce**, the Y-devices represent various custom device models that are not found in SPICE. In this case, `YPowerGridBranch` is a Xyce model for a "power grid branch" that uses the power flow equations given in Section 2.3. The device name is then `pg1_2` to denote the branch between buses 1 and 2. So, the four terminals are given by bus1Th, bus2Th, bus1VM and bus2VM, since the PQ Polar solution format uses voltage magnitude and phase as the solution variables. The "analysis type" is given by the AT instance parameter, which is set to PQP (which stands for PQ Polar) in this example. Finally, in SPICE and **Xyce**, a $+$ character is used to indicate a continuation of the previous line. The instance parameters for the branch resistance (R), branch reactance (X) and branch shunt susceptance (B) are listed last in this example, and are given in "per unit". However, the ordering of the instance parameters does not actually matter, as long as they come after the terminal names.

```
YPowerGridBranch pg1_2 bus1Th bus2Th bus1VM bus2VM AT=PQP R=0.01938
+X=0.05917 B=0.0528
```

As noted above, the device name and terminal names can be any character string that is legal in **Xyce**. However, a consistent naming convention aids the auto-generation of large netlists from other industry-standard formats. Auto-generation of the **Xyce** netlists is important (for the "maintenance of user sanity") for larger examples.

3.1.3 .NODESET and Flat Start

Xyce allows for the use of both initial conditions (`.IC` statements) and initial guesses (`.NODESET` statements). The netlist may contain `.IC` statements or `.NODESET` statements, but not both. Initial conditions are enforced as part of the steady-state power flow solution. For initial guesses, **Xyce** actually does two DC Operating Point (DCOP)[1] calculations. The first calculation enforces the initial guesses as constraints on the solution. The second calculation then starts from that constrained solution, but does not enforce those guesses as constraints on the final solution. This is different than how SPICE handles the `.NODESET` command, but it has been found empirically to work in **Xyce** simulations of large power grids.

Since the power flow equations for PQ Rectangular and PQ Polar format are non-linear, they do not have a unique solution for a given grid configuration. So, it is essential that a `.NODESET` statement be used in conjunction with those solution formats. In the absence of a better guess, a "flat start" is often used. In that case, the voltage magnitude ($|V|$) at each bus is set to 1, while the voltage angle (Θ) is set equal to zero. This approach is illustrated in the annotated netlists given below. The steady-state solution can then be used as a known starting point (e.g., enforced with `.IC` statements) at the start of a transient simulation in **Xyce**.

[1] In SPICE-compatible circuit simulators, a ".DC analysis" or ".DC simulation" calculates the initial conditions (or "DC Operating Point" (DCOP)) for the circuit. It is the analog of a steady-state power flow calculation in a power grid analysis. This circuit simulator terminology may be confusing for power engineers since "DC analysis" has a completely different meaning in that domain.

3.1.4 Sign Conventions on Power Flows

The user should take care with the sign conventions for power-flow simulations. The normal convention in **Xyce** and SPICE for voltage and current sources is that positive current flows from the positive node through the source to the negative node. For power grids, loads cause power to flow out of the grid while generator buses and the slack bus inject power into the grid. The netlists given below account for this by reversing the polarity of the ammeters in series with the slack bus and generator buses. In that case, `I(Vamm1P)` in a `.PRINT` statement yields a positive value if the slack bus is providing positive real power to the grid. Alternate ways of handling the sign conventions for generators and loads are noted in the discussion of the slack bus models in Section 2.1.

3.1.5 Default Values

Each device model has default values for its instance parameters, which are listed in the Xyce Reference Guide [5]. However, those defaults are typically set at 1. So, it is safest (as with the branch `pg4_5` between buses 4 and 5 in the IEEE 14-bus test case [1]) to explicitly set all of the instance parameters, including ones that are 0. The instance parameters are typically given in the "per unit" system-of-measure, except for angles which are in radians. That system-of-measure is standard in power engineering but may be unfamiliar to the circuit simulation community. For a description of "per unit", please consult [7].

3.1.6 Caveats on Netlist Cut-n-Paste

The netlists given in this document will run in version 6.3 of **Xyce**. However, some caveats must be noted if the user tries to directly cut-n-paste these netlists from this document. So, it is recommended that the user download the associated archive files that contain all of the netlist and library files referenced in this document.

- For Windows users, it is important to copy (or edit) netlists from this document into a text editor (e.g., Notepad) rather than into MSWord, since MSWord adds extra control characters.
- A copy operation from Adobe Acrobat menu might include the document page numbers in the netlist. If this happens then **Xyce** will flag those lines in the netlist as having an invalid syntax, and they can be removed by the user.
- On some operating systems, the "{" and "}" characters are turned into "f" and "g" characters when cutting-n-pasting from Adobe Acrobat. Those lines will be flagged as having invalid **Xyce** syntax. On some Windows systems, this can be obviated by using the Notepad editor and "Copy with Formatting" within Acrobat.
- On some operating systems (e.g., some OSX systems), some of the new lines are not properly preserved when cutting-n-pasting from Adobe Acrobat. Those lines will be flagged as having invalid **Xyce** syntax.

3.2 Annotated Netlist for 14-Bus Test Case

This netlist runs in version 6.3 of **Xyce**. It calculates the DC Operating Point (DCOP) for the IEEE 14-bus test case [1] using the PQ Polar solution format. All lines that begin with * are comments. Slightly different versions of this netlist are found in the **Xyce** Regression Test Suite in the Xyce_Regression\Netlists\POWER_GRID subdirectory.

The units used for each device parameter are explained in detail in the Xyce Reference Guide [5]. In general though, all device parameters and solution variables are in "per unit", with the exception of angles which are in radians.

```
* Xyce netlist for the IEEE 14 Bus Test Case

********************************************************
* Simulation control statements.                       *
* This does one DC Operating Point (DCOP) calculation  *
* for the simulation with the voltage magnitude at     *
* the slack bus set to 1.06 (per unit)                 *
********************************************************
.DC V1VM 1.06 1.06 1

**********************************************************
* Bus 1 is the slack bus.  Ammeters are wired so that    *
* positive power (P or Q) flows into Bus 1, since the    *
* slack bus is a generator. The slack bus has VM=1.06    *
* and V1TH=0 (in per unit) per the IEEE test case.  This *
* is the reference voltage for the grid model.           *
**********************************************************
V1Th Bus1Th ammBus1P 0V
V1VM Bus1VM ammBus1Q 1.06V
Vamm1P 0 ammBus1P 0V
Vamm1Q 0 ammBus1Q 0V

****************************************************
* Branch Definitions.                              *
* Xyce has a verbose (PowerGridBranch) and a       *
* shortened (PGBR) form for the device names.      *
* It is best to explicitly set all instance        *
* parameters (e.g., B=0 for device pg4_5).         *
****************************************************
YPowerGridBranch pg1_2 bus1Th bus2Th bus1VM bus2VM AT=PQP R=0.01938
+X=0.05917 B=0.0528
YPGBR pg1_5 bus1Th bus5Th bus1VM bus5VM AT=PQP R=0.05403 X=0.22304 B=0.0492
YPGBR pg2_3 bus2Th bus3Th bus2VM bus3VM AT=PQP R=0.04699 X=0.19797 B=0.0438
YPGBR pg2_4 bus2Th bus4Th bus2VM bus4VM AT=PQP R=0.05811 X=0.17632 B=0.034
YPGBR pg2_5 bus2Th bus5Th bus2VM bus5VM AT=PQP R=0.05695 X=0.17388 B=0.0346
YPGBR pg3_4 bus3Th bus4Th bus3VM bus4VM AT=PQP R=0.06701 X=0.17103 B=0.0128
```

```
YPGBR pg4_5 bus4Th bus5Th bus4VM bus5VM AT=PQP R=0.01335 X=0.04211 B=0
YPGBR pg6_11 bus6Th bus11Th bus6VM bus11VM AT=PQP R=0.09498 X=0.1989 B=0
YPGBR pg6_12 bus6Th bus12Th bus6VM bus12VM AT=PQP R=0.12291 X=0.25581 B=0
YPGBR pg6_13 bus6Th bus13Th bus6VM bus13VM AT=PQP R=0.06615 X=0.13027 B=0
YPGBR pg7_8 bus7Th bus8Th bus7VM bus8VM AT=PQP R=0 X=0.17615 B=0
YPGBR pg7_9 bus7Th bus9Th bus7VM bus9VM AT=PQP R=0 X=0.11001 B=0
YPGBR pg9_10 bus9Th bus10Th bus9VM bus10VM AT=PQP R=0.03181 X=0.0845 B=0
YPGBR pg9_14 bus9Th bus14Th bus9VM bus14VM AT=PQP R=0.12711 X=0.27038 B=0
YPGBR pg10_11 bus10Th bus11Th bus10VM bus11VM AT=PQP R=0.08205 X=0.19207 B=0
YPGBR pg12_13 bus12Th bus13Th bus12VM bus13VM AT=PQP R=0.22092 X=0.19988 B=0
YPGBR pg13_14 bus13Th bus14Th bus13VM bus14VM AT=PQP R=0.17093 X=0.34802 B=0

*****************************************************
* Transformer Definitions.                          *
* Xyce has a verbose (PowerGridTransformer)         *
* and a shortened (PGTR) form for the device names. *
*****************************************************
YPowerGridTransformer pg4_7 bus4Th bus7Th bus4VM bus7VM R=0 X=0.20912
+ TR=0.978
YPGTR pg4_9 bus4Th bus9Th bus4VM bus9VM AT=PQP R=0 X=0.55618 TR=0.969
YPGTR pg5_6 bus5Th bus6Th bus5VM bus6VM AT=PQP R=0 X=0.25202 TR=0.932

****************************************************************
* Bus Shunt Definitions.                                       *
* Terminal 2 is ground, which is either node 0 or GND in Xyce. *
****************************************************************
YPowerGridBusShunt pg9_GND bus9Th 0 bus9VM 0 G=0 B=0.19

**************************************************************
* Generator Bus Definitions.                                 *
* The generators at buses 2 and 3 use Xyce device models.    *
* The generators at buses 6 and 8 use voltage and current    *
* sources, as an example of both approaches.  The            *
* ammeters are wired so that positive power (P or Q) flows   *
* into the buses. VM and P are given in per unit.            *
**************************************************************
YPowerGridGenBus pgb2 bus2Th ammbus2P bus2VM ammBus2Q AT=PQP
+ VM=1.045 P=0.4
Vamm2P 0 ammBus2P 0V
Vamm2Q 0 ammBus2Q 0V

YPGGB pgb3 bus3Th ammbus3P bus3VM ammBus3Q AT=PQP VM=1.01 P=0
Vamm3P 0 ammBus3P 0V
Vamm3Q 0 ammBus3Q 0V

IGen6P ammBus6P bus6Th 0
Vamm6P 0 ammBus6P 0V
VGen6VM bus6VM ammBus6Q 1.07
```

```
Vamm6Q 0 ammBus6Q 0V

IGen8P ammBus8P bus8Th 0
Vamm8P 0 ammBus8P 0V
VGen8VM bus8VM ammBus8Q 1.09
Vamm8Q 0 ammBus8Q 0V

************************************************************
* Load Definitions.                                        *
* For loads, positive power flows out of the grid.         *
* A separate current source is used to set P and Q at each *
* at each bus.  The zero-current sources could be omitted, *
* but are included for use in the .PRINT statement. The    *
* load values are in per unit.                             *
************************************************************
ILoad2P bus2Th 0 0.217
ILoad2Q bus2VM 0 0.127
ILoad3P bus3Th 0 0.942
ILoad3Q bus3VM 0 0.19
ILoad4P bus4Th 0 0.478
ILoad4Q bus4VM 0 -0.039
ILoad5P bus5Th 0 0.076
ILoad5Q bus5VM 0 0.016
ILoad6P bus6Th 0 0.112
ILoad6Q bus6VM 0 0.075
ILoad7P bus7Th 0 0
ILoad7Q bus7VM 0 0
ILoad9P bus9Th 0 0.295
ILoad9Q bus9VM 0 0.166
ILoad10P bus10Th 0 0.09
ILoad10Q bus10VM 0 0.058
ILoad11P bus11Th 0 0.035
ILoad11Q bus11VM 0 0.018
ILoad12P bus12Th 0 0.061
ILoad12Q bus12VM 0 0.016
ILoad13P bus13Th 0 0.135
ILoad13Q bus13VM 0 0.058
ILoad14P bus14Th 0 0.149
ILoad14Q bus14VM 0 0.05

*******************************************************************
* .NODESET is used to enforce a "flat start" for PQ formulations. *
* The use of .NODESET is mandatory for the PQ formulations.       *
* Otherwise, the DCOP calculation may fail or converge to a valid *
* but undesired solution. VM and Theta are given in per unit.     *
*******************************************************************
.NODESET V(bus1Th)=0 V(bus1VM)=1 V(bus2Th)=0 V(bus2VM)=1
+ V(bus3Th)=0 V(bus3VM)=1 V(bus4Th)=0 V(bus4VM)=1 V(bus5Th)=0 V(bus5VM)=1
```

```
+ V(bus6Th)=0 V(bus6VM)=1 V(bus7Th)=0 V(bus7VM)=1 V(bus8Th)=0 V(bus8VM)=1
+ V(bus9Th)=0 V(bus9VM)=1 V(bus10Th)=0 V(bus10VM)=1
+ V(bus11Th)=0 V(bus11VM)=1 V(bus12Th)=0 V(bus12VM)=1
+ V(bus13Th)=0 V(bus13VM)=1 V(bus14Th)=0 V(bus14VM)=1

*****************************************************************
* Output Statements.                                            *
* V() gives the voltage angle and magnitude at each bus.        *
* I() gives the real (P) and reactive (Q) power at each bus.    *
* Ammeters (Vamm1P) were defined so that positive power flows   *
* into the grid from a generator and out of the grid for a load. *
* The values printed out are in per unit for |V|, P and Q.  The  *
* values are in radians for the voltage angles.                 *
*****************************************************************
.PRINT DC width=10 precision=6
+ V(bus1Th) V(bus1VM) V(bus2Th) V(bus2VM) V(bus3Th) V(bus3VM)
+ V(bus4Th) V(bus4VM) V(bus5Th) V(bus5VM) V(bus6Th) V(bus6VM)
+ V(bus7Th) V(bus7VM) V(bus8Th) V(bus8VM) V(bus9Th) V(bus9VM)
+ V(bus10Th) V(bus10VM) V(bus11Th) V(bus11VM) V(bus12Th) V(bus12VM)
+ V(bus13Th) V(bus13VM) V(bus14Th) V(bus14VM)
+ I(Vamm1P) I(Vamm1Q) I(Vamm2P) I(Vamm2Q) I(Vamm3P) I(Vamm3Q)
+ I(ILoad4P) I(ILoad4Q) I(ILoad5P) I(ILoad5Q) I(Vamm6P) I(Vamm6Q)
+ I(ILoad7P) I(ILoad7Q) I(Vamm8P) I(Vamm8Q) I(ILoad9P) I(ILoad9Q)
+ I(ILoad10P) I(ILoad10Q) I(ILoad11P) I(ILoad11Q) I(ILoad12P)
+ I(ILoad12Q) I(ILoad13P) I(ILoad13Q) I(ILoad14P) I(ILoad14Q)

.END
```

The **Xyce** results for the steady state power flow are shown in Table 3.1. These results match other open source tools, such as Power System Toolbox (PST), to within less than 1e-4.

Table 3.1: Steady State Results for IEEE 14-Bus Test Case

Bus	$\|V\|$ (per unit)	Θ (radians)	P (per unit)	Q (per unit)
1	1.0600	0.0000	2.3239	-0.1655
2	1.0450	-0.0870	0.4000	0.4356
3	1.0100	-0.2221	0.0000	0.2508
4	1.0177	-0.1800	0.4780	-0.0390

Table 3.1: Steady State Results for IEEE 14-Bus Test Case

Bus	$\lvert V\rvert$ (per unit)	Θ (radians)	P (per unit)	Q (per unit)
5	1.0195	-0.1531	0.0760	0.0160
6	1.0700	-0.2482	0.0000	0.1273
7	1.0615	-0.2332	0.0000	0.0000
8	1.0900	-0.2332	0.0000	0.1762
9	1.0559	-0.2607	0.2950	0.1660
10	1.0510	-0.2635	0.0900	0.0580
11	1.0569	-0.2581	0.0350	0.0180
12	1.0552	-0.2631	0.0610	0.0160
13	1.0504	-0.2645	0.1350	0.0580
14	1.0355	-0.2798	0.1490	0.0500

3.3 Annotated Netlists for 3-Bus Examples

The generator bus model in **Xyce** 6.3 only works for the PQ Polar solution format. So, this section gives simple 3-bus examples, with only a slack bus, for the I=YV and PQ Rectangular solution formats. These examples do include all of the other devices though as shown in the single line diagram in Figure 3.1.

3.3.1 PQ Rectangular Solution Format

For the PQ Rectangular solution format, the solution variables are the real and imaginary parts of the voltages (`VR` and `VI`) and the real and reactive power flows (`P` and `Q`). This changes the terminal names for the devices. In addition, the `.PRINT` statement is more complicated since the

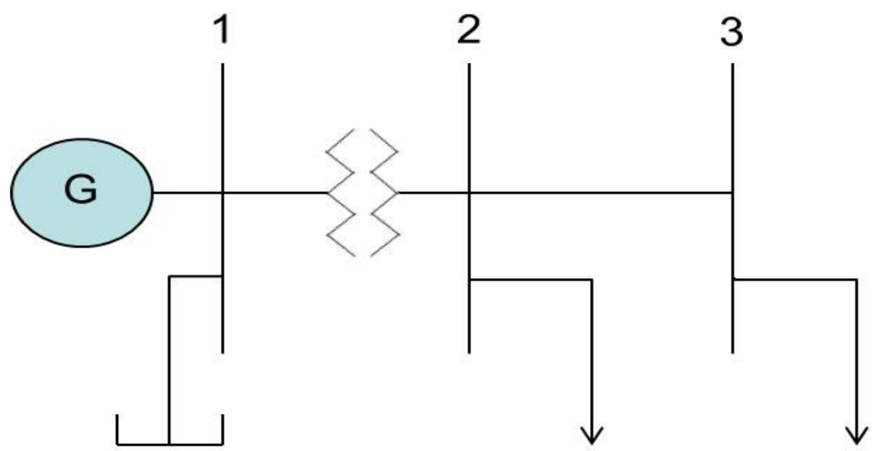

Figure 3.1. Single Line Diagram for Three Bus Example.

voltage magnitudes and phases must now be calculated with expressions. (Note: See Section 2 for a discussion of why the **Xyce** team decided to implement all three solution formats.)

```
* Xyce netlist for the Three-Bus Test Case, using PQR
* solution format.

********************************************************
* Simulation control statements.                       *
* This does one DC Operating Point (DCOP) calculation  *
* for the simulation with the real voltage at          *
* the slack bus set to 1 (per unit).                   *
********************************************************
.DC V1R 1 1 1

***********************************************************
* Bus 1 is the slack bus.  Ammeters are  wired so that    *
* positive current flows into Bus 1, since the slack bus  *
* is a generator.  Terminals are the real and imaginary   *
* voltages at buses 1 and 2.  Values are in per unit.     *
***********************************************************
V1R Bus1R ammBus1R 1V
V1I Bus1I ammBus1I 0V
Vamm1R 0 ammBus1R 0V
Vamm1I 0 ammBus1I 0V

*************************************************
* Transformer, Branch and Bus Shunt Definitions  *
*************************************************
YPGTR pg1_2 bus1R bus2R bus1I bus2I AT=PQR R=0.05 X=0.2 TR=0.978
```

```
YPGBR pg2_3 bus2R bus3R bus2I bus3I AT=PQR R=0.05 X=0.1 B=0.05
YPGBS pg1_GND bus1R 0 bus1I 0 AT=PQR G=0.2 B=0.1

*************************************************************
* Load Definitions                                          *
* Similar to PQ Polar definitions, but P is associated with *
* the VR terminal and Q is associated with the VI terminal. *
* The load values are given in per unit.                    *
*************************************************************
ILoad2P bus2R 0 0.25
ILoad2Q bus2I 0 0.2
ILoad3P bus3R 0 0.2
ILoad3Q bus3I 0 0.1

*******************************************************************
* .NODESET is used to enforce a "flat start" for PQ formulations. *
* The use of .NODESET is mandatory for the PQ formulations.       *
* Otherwise, the DCOP calculation may fail or converge to a valid *
* but undesired solution. Voltage values are given in per unit.   *
*******************************************************************
.NODESET V(bus1R)=1 V(bus1I)=0 V(bus2R)=1 V(bus2I)=0
+ V(bus3R)=1 V(bus3I)=0

******************************************************************
* Output Statements.                                             *
* The expressions sqrt(VR*VR + VI*VI) and atan(VI/VR) give the   *
* voltage magnitude and angle at each bus.  I() gives  the       *
* power flow, just as in the PQ Polar format .                   *
* Ammeters (Vamm1R) were defined so that positive power flows    *
* into the grid from a generator and out of the grid for a load. *
* |V|, P and Q are given in per unit.  Theta is in radians.      *
******************************************************************
.PRINT DC width=10 precision=6
+ {sqrt(V(bus1R)*V(bus1R)+V(bus1I)*V(bus1I))}
+ {atan(V(bus1I)/V(bus1R))}
+ {sqrt(V(bus2R)*V(bus2R)+V(bus2I)*V(bus2I))}
+ {atan(V(bus2I)/V(bus2R))}
+ {sqrt(V(bus3R)*V(bus3R)+V(bus3I)*V(bus3I))}
+ {atan(V(bus3I)/V(bus3R))}
+ I(Vamm1R) I(Vamm1I) I(ILoad2P) I(ILoad2Q) I(ILoad3P) I(ILoad3Q)

.END
```

The results calculated by **Xyce** are shown in Table 3.2. They apply to both the PQ Rectangular and I=YV solution formats. These results match other open source tools, such as Power System Toolbox (PST), to within less than 1e-4.

Table 3.2: Steady State Results for Three Bus Test Case

Bus	$\|V\|$ (per unit)	Θ (radians)	P (per unit)	Q (per unit)
1	1.0000	0.0000	0.6682	0.2243
2	0.9392	-0.0807	0.2500	0.2000
3	0.9196	-0.0993	0.2000	0.1000

3.3.2 I=YV Solution Format

For the I=YV format, the solution variables are the real and imaginary parts of the voltages and currents, which are `VR`, `VI`, `IR` and `II` respectively. So, the netlist given below is similar to the netlist for the PQ Rectangular format with differences in the `.PRINT` statement and the Constant Power Load (CPL) definitions. In this format, the real and reactive power flows are calculated from the real and imaginary current flows with expressions. The CPL is defined using a subcircuit, as outlined in Section 2.2. Finally, since the power-flow equations for the I=YV format are linear, a `.NODESET` statement is not required to enforce a "flat start". (Note: The I=YV solution format is denoted by an analysis type (`AT`) of `IV` on the device instance lines in the **Xyce** netlists given below.)

```
* Xyce netlist for the Three-Bus Test Case, using I=YV
* solution format.

********************************************************
* Simulation control statements.                       *
* This does one DC Operating Point (DCOP) calculation  *
* for the simulation with the real voltage at          *
* the slack bus set to 1 (per unit).                   *
********************************************************
.DC V1R 1 1 1

***********************************************************
* Bus 1 is the slack bus.  Ammeters are  wired so that    *
* positive current flows into Bus 1, since the slack bus  *
* is a generator.  Terminals are the real and imaginary   *
* voltages at buses 1 and 2.  Values are in per unit.     *
***********************************************************
V1R Bus1R ammBus1R 1V
V1I Bus1I ammBus1I 0V
Vamm1R 0 ammBus1R 0V
Vamm1I 0 ammBus1I 0V
```

```
************************************************
* Transformer, Branch and Bus Shunt Definitions  *
************************************************
YPGTR pg1_2 bus1R bus2R bus1I bus2I AT=IV R=0.05 X=0.2 TR=0.978
YPGBR pg2_3 bus2R bus3R bus2I bus3I AT=IV R=0.05 X=0.1 B=0.05
YPGBS pg1_GND bus1R 0 bus1I 0 AT=IV G=0.2 B=0.1

***************************************************
* Load Definitions use the CPL subcircuit definition  *
* given below.  Values are in per unit.              *
***************************************************
XLoad2 bus2R bus2I CPL PARAMS: P=0.25 Q=0.2
XLoad3 bus3R bus3I CPL PARAMS: P=0.2 Q=0.1

*******************************************************************
* Output Statements.                                              *
* The Xyce expressions sqrt(VR*VR + VI*VI) and atan(VI/VR) give the  *
* voltage magnitude and angle at each bus. The real and reactive     *
* power at each bus are then (VR*IR + VI*II) and (VR*II - VI*IR).    *
* Ammeters (Vamm1R) were defined so that positive power flows        *
* into the grid from a generator and out of the grid for a load.     *
* |V|, P and Q are in per unit.  Theta is in radians.                *
*******************************************************************
.PRINT DC width=10 precision=6
+ {sqrt(V(bus1R)*V(bus1R)+V(bus1I)*V(bus1I))}
+ {atan(V(bus1I)/V(bus1R))}
+ {sqrt(V(bus2R)*V(bus2R)+V(bus2I)*V(bus2I))}
+ {atan(V(bus2I)/V(bus2R))}
+ {sqrt(V(bus3R)*V(bus3R)+V(bus3I)*V(bus3I))}
+ {atan(V(bus3I)/V(bus3R))}
+ {V(bus1R)*I(Vamm1R)+V(bus1I)*I(Vamm1I)}
+ {V(bus1I)*I(Vamm1R)-V(bus1R)*I(Vamm1I)}
+ {V(bus2R)*I(Xload2:VammR)+V(bus2I)*I(Xload2:VammI)}
+ {V(bus2I)*I(Xload2:VammR)-V(bus2R)*I(Xload2:VammI)}
+ {V(bus3R)*I(Xload3:VammR)+V(bus3I)*I(Xload3:VammI)}
+ {V(bus3I)*I(Xload3:VammR)-V(bus3R)*I(Xload3:VammI)}

*************************************************************
* Subcircuit definition of a constant power load for the I=YV  *
* solution format. Complex power S = P+jQ = (VR+jVI)*(IR-jII)  *
*************************************************************
.SUBCKT CPL RNode INode PARAMS: P=0.5 Q=0.0 CurrLim=1000

* Ammeter at load is defined so that power flows into the load from the bus
VammR ammR 0 0V
VammI ammI 0 0V
* Use limit function in B-source, since S/V becomes large for small values of V
```

```
BloadR RNode ammR
+ I={limit((P*V(RNode)+Q*V(INode))/(V(RNode)*V(RNode)+V(INode)*V(INode)),
+ -CurrLim,CurrLim)}
BloadI INode ammI
+ I={limit((P*V(INode)-Q*V(RNode))/(V(RNode)*V(RNode)+V(INode)*V(INode)),
+ -CurrLim,CurrLim)}
.ENDS

.END
```

3.4 Additional Examples

Additional examples can be found in the regression test suite that is distributed with the open-source **Xyce** source code. The netlists and results are in the Xyce_Regression\Netlists\POWER_GRID and Xyce_Regression\OutputData\POWER_GRID subdirectories of the open-source distribution, respectively. The results for each regression test were validated against theoretical results (e.g., pp. 250-254 of [7]) or another open-source simulation tool such as Power System Toolbox (PST).

4. Transient Modeling

Xyce 6.3 supports basic capabilities for the simulation of transient events in power grids. It does so by leveraging existing **Xyce** device models to construct time-domain models for generating units such as the steam turbine without re-heat shown in Figure 4.1 below. The interconnecting power grid between the generating units is then modeled by the steady-state power flow at each time step in the **Xyce** simulation, using the models described in the previous chapter. This approach assumes that the 60 HZ electrical transients in the power grid decay much faster than the $\approx$ 1HZ mechanical time constants of the generating units. If additional accuracy is required then a more exact time-domain simulation of the power grid could be added. Indeed, for other projects, Xyce has been used to model sub-60 Hz transients in power grids. However, that level of resolution greatly increases the simulation time and is typically not required for this application. Finally, the **Xyce** models for the power grid assume a balanced system with positive sequence components.

4.1 Generating Unit Models

If a generating-unit submodel (e.g., turbine, governor, exciter, etc.) can be represented by a linearized model[1] then there is a simple way for an end-user to describe that subsystem in **Xyce** (or any SPICE-compatible simulator) as a set of low-order RLC filters, adders and gain blocks. As an example, the turbine/governor model for a steam turbine without reheat (see pp. 598-600 of [7]) has the block diagram shown in Figure 4.1, while a hydroelectric generating unit has the block diagram shown in Figure 4.2. A **Xyce** netlist can be built up from library (`.lib` files) as illustrated in the next two subsections. The first library file (controlSystemModels.lib) contains definitions of basic control-system blocks. The second library file (powerGridModels.lib) then contains subcircuit definitions for the two generating unit models, based on those control system blocks. This encapsulation allows for more compact netlists that are easier to auto-generate for transient simulations of large power grids.

Xyce took this approach because end-users can easily implement new models, from block diagram descriptions, with an elementary knowledge of RLC filter design. More complicated, nonlinear models for power-grid devices might require the development of additional custom models within Xyce's C++ software framework. In addition, a custom C++ **Xyce** device model for the linearized models might have fewer solution variables, and hence run slightly faster.

This solution approach has been shown to converge in large grids that contain approximately 30,000 bus, 40,000 branches and 2500 generators. (Note: The number of solution variables is

[1] Xyce also supports a wide range of non-linear elements via the "non-linear dependent source", or B-source. The **Xyce** Reference Guide [5] provides more details.

approximately 4x the number of buses, with additional solution variables for the generator models.) However, there is ongoing work on results and model validation. In addition, models for more complex synchronous machine models (such as “One d- and One q- Axis” and Sauer-Pai models given on pp. 328-336 of [8]) are needed.

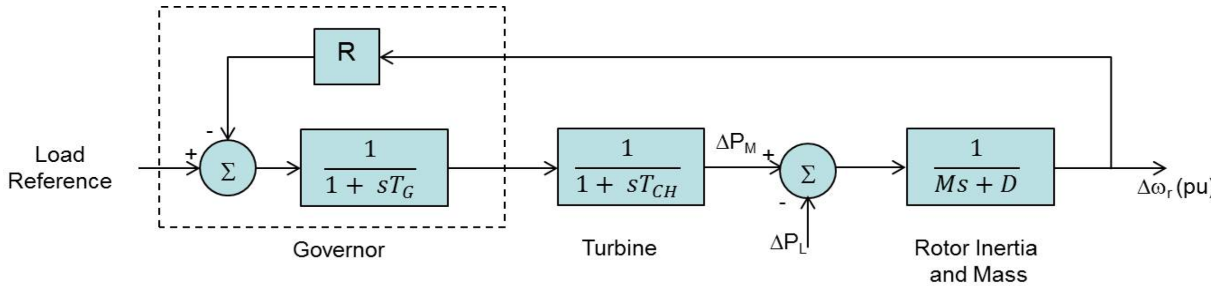

Figure 4.1. Block Diagram for Steam Turbine without Re-Heat.

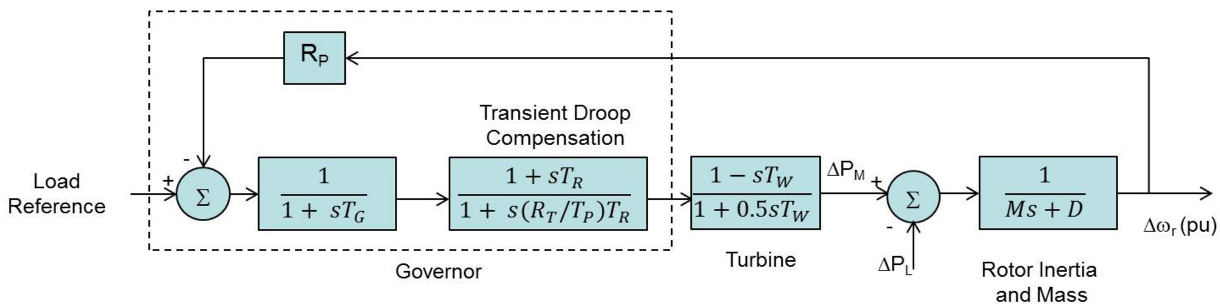

Figure 4.2. Block Diagram for Hydroelectric Generating Unit.

For Figures 4.1 and 4.2, the following definitions apply:

s = Laplace operator

ΔP_m = change in mechanical power input to the generator (per unit)

ΔP_L = change in electrical power output from the generator (per unit)

$\Delta \omega_R$ = rotor speed deviation (per unit)

$M = 2H$, where H = generator inertia constant (MW-sec / MVA)

D = generator load-damping constant (% change in load for a 1% change in frequency)

T_G = governor (or gate) time constant (seconds)

For Figure 4.1, these additional definitions apply (see pg. 426 of [7]):

T_{CH} = time constant of main inlet volumes and steam chest (seconds)

R = speed droop

For Figure 4.2, these additional definitions apply (see pg. 398 of [7]):

R_T = transient droop

R_P = permanent droop

T_R = reset time (seconds)

T_P = pilot valve and servomotor time constant (seconds)

4.2 Library File for Control System Blocks

Xyce supports the SPICE concept of "library" files that contain subcircuit definitions that can be included in netlists. As an example, the subcircuits for the low-pass, high-pass and lead-lag filters used in the block diagrams for the steam turbine and hydro generator can be implemented as RC filters as follows. (Note: These examples may not be the most concise syntax for these particular filters. In addition, some of the filter values (such as C=1) are not "physically sensible" for an electrical circuit. However, they are correct for this application where the time constants for the mechanical elements are typically on the order of 0.1 to 1 Hz.)

```
*controlSystemModels.lib file

*********************************************
* low-pass filter with response = 1/(1+sTc) *
*********************************************
.SUBCKT lowPassFilter lpf_in lpf_out PARAMS: Tc=1
Rlpf lpf_in lpf_out {Tc}
Clpf lpf_out 0 1
.ENDS

*************************************************************
* 2nd order low-pass filter with response = 1/((1+aS)*(1+bS)) *
*************************************************************
.SUBCKT lowPassFilter2ndOrder lpf_in lpf_out PARAMS: a=1 b=1
Rlpf lpf_in ind_in {a+b}
Llpf ind_in lpf_out {a*b}
Clpf lpf_out 0 1
.ENDS

**********************************************
* high-pass filter with response sTc/(1+sTc) *
**********************************************
.SUBCKT highPassFilter hpf_in hpf_out PARAMS: Tc=1
Chpf hpf_in hpf_out 1
```

```
Rhpf hpf_out 0 {Tc}
.ENDS

**********************************************
* lead-lag filter with response (1+aS)/(1+bS) *
**********************************************
.SUBCKT leadLagFilter llf_in llf_out PARAMS: a=1 b=1
* Implement as parallel HPF and LPF.  Note: This is
* not the most efficient implementation.
Xhpf llf_in hpf_out highPassFilter PARAMS: Tc={b}
Xlpf llf_in lpf_out lowPassFilter PARAMS: Tc={b}
Xout lpf_out hpf_out llf_out adder PARAMS: gain1=1 gain2={a/b}
.ENDS

**********************
* adder (with gains) *
**********************
.SUBCKT adder in_1 in_2 out PARAMS: gain1=1 gain2=1
Badd out 0 V={gain1*V(in_1) + gain2*V(in_2)}
.ENDS

**************
* gain block *
**************
.SUBCKT gainBlock in out PARAMS: gain=1
Bgainr out 0 V={gain*V(in)}
.ENDS
```

4.3 Steam Turbine and Hydro Generator Examples

This section gives annotated subcircuit definitions for a steam turbine without re-heat and a hydro generator. They are based on the examples given in pp. 589-600 of [7], and use the control-system blocks defined in the previous section. The subcircuit definitions are now fairly concise for each generator type, and can be written down based on the S-domain block diagrams given in Figures 4.1 and 4.2. Since the generator acceleration equations are common to both block diagrams, they are defined as a subcircuit in this netlist. Finally, in this library file, the `lrp`, `load` and `rotor_out` terminals in the subcircuit definitions correspond to the "Load Reference" and ΔP_L inputs and the $\Delta\omega_R$ output in the respective block diagrams.

```
*powerGridModels.lib file

**************************************************************
* Subcircuit definition for a Steam Generator without Reheat *
```

```
***********************************************************
.SUBCKT steamGenWoReheat lrp load rotor_out
+ PARAMS: D=1 M=10 TG=0.2 TCH=0.3 R=0.05
* adder block in governor and feedback path
XgovIn lrp rotor_out governor_in adder PARAMS: gain1=1 gain2={-1/R}

*low-pass filter for governor
Xgate governor_in governor_out lowPassFilter PARAMS: Tc={TG}

*low-pass filter for non-reheat turbine
XturbOut governor_out turbine_out lowPassFilter PARAMS: Tc={TCH}

*adder block between turbine and rotor
Xtadd turbine_out load rotor_in adder PARAMS: gain1=1 gain2=-1

*generator acceleration equation
Xgae rotor_in rotor_out genAccelEqn PARAMS: mgae={M} dgae={D}
.ENDS

************************************************
*  Subcircuit definition for a Hydro Generator *
************************************************
.SUBCKT hydroGen lrp load rotor_out
+PARAMS: D=1 M=6 TG=0.2 TW=1 RP=0.05 RT=0.38 TR=5

* adder block in governor and feedback path
XgovIn lrp rotor_out governor_in adder PARAMS: gain1=1 gain2={-1/RP}

* low-pass filter for governor
Xgate governor_in glpf_out lowPassFilter PARAMS: Tc={TG}

* transient droop compensator in governor
* response = (1+aS)/(1+bS).  Implement as parallel HPF and LPF
Xghpf glpf_out gtdchpf_out highPassFilter PARAMS: Tc={RT*TR/RP}
Xglpf glpf_out gtdclpf_out lowPassFilter PARAMS: Tc={RT*TR/RP}
Xgtdc gtdclpf_out gtdchpf_out governor_out adder PARAMS: gain1=1 gain2={RP/RT}

* hydro turbine, transfer function = (1-s*2*Tw)/(1+sTw)
* implement as parallel HPF and LPF
Xthpf governor_out thpf_out highPassFilter PARAMS: Tc={0.5*Tw}
Xtlpf governor_out tlpf_out lowPassFilter PARAMS: Tc={0.5*Tw}
Xtout tlpf_out thpf_out turbine_out adder PARAMS: gain1=1 gain2=-2

* adder block between turbine and rotor
Xtadd turbine_out load rotor_in adder PARAMS: gain1=1 gain2=-1

* generator acceleration equation
Xgae rotor_in rotor_out genAccelEqn PARAMS: mgae={M} dgae={D}
```

```
.ENDS

***********************************
*  Generator Acceleration Equations *
***********************************
.SUBCKT genAccelEqn rotor_in rotor_out PARAMS: mgae=10 dgae=1
* set gain K = 1/D and T=M/D
Bgain filter_in 0 V={V(rotor_in)/dgae}

*low-pass filter, set T=M/D
XgaeLpf filter_in rotor_out lowPassFilter PARAMS: Tc={mgae/dgae}
.ENDS
```

Based on these subcircuit definitions, a comparison of the step response of a steam turbine and a hydro generator can be done as follows in **Xyce** by including the library files in the netlist. For both generator instances, the `lrp`, `load` and `rotor_out` terminals correspond to the "Load Reference" and ΔP_L inputs and the $\Delta\omega_R$ output in the respective block diagrams.

```
* Comparison of step-response of steam generating unit without re-heat
* and hydro generating unit.  Use previously defined library files.
.INCLUDE powerGridModels.lib
.INCLUDE controlSystemModels.lib

****************************
* Simulate 20s of operation *
****************************
.tran 1ms 20s

*******************************************************************
* lrsVal is the nominal set point of 50% full power, in per unit.      *
* deltapl a is 1% change in the load.  So, the electrical load changes *
* to 50.5%.  D1 and D2 are the D values for generators 1 and 2.  Etc.  *
*******************************************************************
.param lrsVal=0.5 deltapl=0.01
.param D1=1 M1=10 TG1=0.2 TCH1=0.3 R1=0.05
.param D2=1 M2=6 TG2=0.2 TW2=1 RP2=0.05 RT2=0.38 TR2=5

*load reference setpoint (lrs)
Vloadref lrp 0 {lrsVal}

*step change in load at time = 0.001S.  Use piece-wise linear (PWL) source
*since transitions with infinite slopes cause convergence problems in Xyce.
Vload load 0 PWL(0S {lrsVal} 0.001S {lrsVal*(1+deltapl)})

**************************************************************************
* Definitions of the two generators, using subcircuits in powerGridModels.lib *
**************************************************************************
```

```
Xgen1 lrp load rotor_out1 steamGenWoReheat
+ PARAMS: D=D1 M=M1 TG=TG1 TCH=TCH1 R=R1

Xgen2 lrp load rotor_out2 hydroGen
+ PARAMS: D=D2 M=M2 TG=TG2 TW=TW2 RP=RP2 RT=RT2 TR=TR2

*************************************************
* Output values normalized to change in output power, *
* which is (lrsVal*deltapl).                          *
*************************************************
.print tran V(lrp) v(load)
+ {V(rotor_out1)/(lrsVal*deltapl)}
+ {V(rotor_out2)/(lrsVal*deltapl)}
+ {(V(Xgen1:governor_out)-lrsVal)/(lrsVal*deltapl)}
+ {(V(Xgen2:governor_out)-lrsVal)/(lrsVal*deltapl)}
+ {(V(Xgen1:turbine_out)-lrsVal)/(lrsVal*deltapl)}
+ {(V(Xgen2:turbine_out)-lrsVal)/(lrsVal*deltapl)}

.end
```

The resultant comparison is shown in Figure 4.3. The first pane shows the rotor's speed deviation (`V(rotor_out)`), while the second pane shows the change in mechanical power at the output of the turbine (`V(turbine_out)`). The values are normalized to the step-change size, as indicated in the netlist's `.PRINT` statement. So, they have the same scales as Figures 11.21 (b) and (c) on pg. 600 of Reference [7], and they are a qualitative match to those two figures.

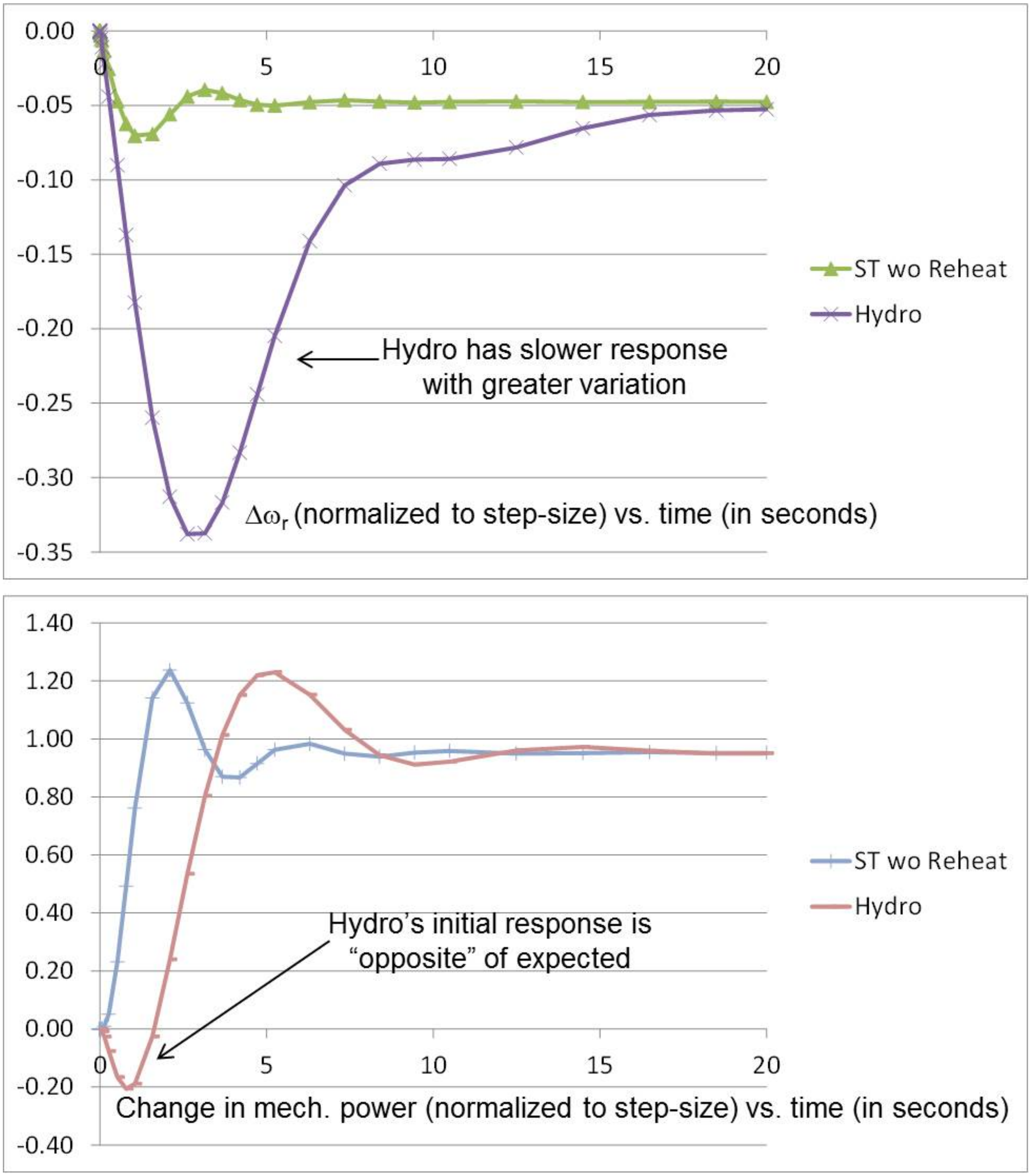

Figure 4.3. Comparison of Step Response of Steam and Hydro Turbines.

4.4 IEEE Classical Generator Model

The previous subsection showed how to turn a block diagram for a generating unit into a **Xyce** netlist. This section shows how to attach a generating unit model to a grid model in **Xyce**. The example given below is the simplest possible case of an IEEE classical generator (e.g., a fixed voltage-magnitude source behind an effective reactance) attached to an "infinite bus" by its transient reactance. Figure 4.4 shows the block diagram for the classical generator's input into the Θ_1 terminal of the branch between bus 1 and bus 2. The $|V_1|$ terminal is held fixed at $|V|$ = 0.95 in this example. (Note: The netlist given below assumes that the damping, D, is non-zero.)

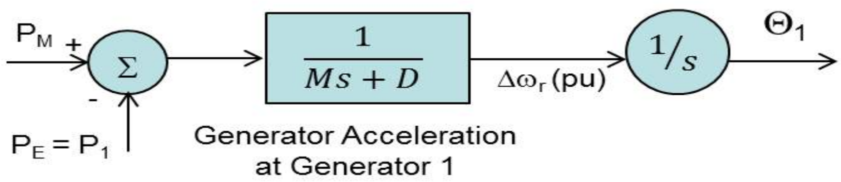

Figure 4.4. Block Diagram of Classical Generator.

```
*Classical Generator Transient Simulation

*********************************************
* Simulation control statements             *
*********************************************
.tran 1ms 100s

**************************************************************************
* Bus 1 is the IEEE Classical Generator.                                 *
* Theta (0.0873) and VM (0.95) values at time=0 were chosen arbitrarily. *
* bto_out comes from the subcircuit definition, and is the integral      *
* of the rotor's speed deviation from its steady-state value.            *
**************************************************************************
.param theta_zero=0.0873
Vamm1P bto_out bus1Th 0V
VM1 bus1VM ammBus1Q 0.95
Vamm1Q 0 ammBus1Q 0V

*********************************************************************
* Step change in generator mechanical power input as in examples in *
* previous section.  Initial value for mechanical power (0.828)     *
* comes from a separate Xyce .DC simulation.                        *
*********************************************************************
.PARAM lrsVal=0.828 deltapl=0.01
VMP1 nodeMP1 0 PWL(0S {lrsVal} 0.001S {lrsVal*(1+deltapl)})
```

```
****************************************************************************
* Model for IEEE Classical Generator.  Difference between mechanical and    *
* electrical power is filtered through the generator acceleration equation. *
* Transient value of theta is the integral of the speed deviation.          *
****************************************************************************
.PARAM D=1 M=10
.PARAM K={1/D}

*Calculate difference between mechanical power input and electrical power
Btadd rotor_gb_in 0 V={V(nodeMP1) - I(Vamm1P)}
*gain block and low-pass filter for rotor acceleration
*T=Rfilter*CFilter, so set Rfilter=M/D
Bgain rotor_in 0 V={K*V(rotor_gb_in)}
Rfilter rotor_in rotor_out {M/D}
Cfilter rotor_out 0 1

* Use B-Source to integrate rotor_out (speed deviation) into
* an angle deviation.
BTO bto_out 0 V={theta_zero + SDT(V(rotor_out))}

************************************************************
* Bus 2 is the slack bus.  Ammeter is wired so that current *
* flows into Bus 2. Values are in per unit.                 *
************************************************************
V2Th Bus2Th ammBus2P 0V
V2VM Bus2VM ammBus2Q 1V
Vamm2P 0 ammBus2P 0V
Vamm2Q 0 ammBus2Q 0V

*************************************************************************
* Branch Definition is simple model for generator's transient reactance. *
*************************************************************************
YPowerGridBranch pg1_2 bus1Th bus2Th bus1VM bus2VM AT=PQP R=0 X=0.1 B=0

*********************************************************************
* Start the grid at DC Operating Point calculated from theory.      *
*********************************************************************
.IC V(bus1Th)=0.0873 V(bus1VM)=0.95 V(bus2Th)=0 V(bus2VM)=1

*****************************************************
* Output Statements.                                *
* VM, P and Q are in per unit.  Theta is in radians. *
*****************************************************
.PRINT TRAN V(bus1Th) V(bus1VM) V(bus2Th) V(bus2VM)
+ I(Vamm1P) I(Vamm1Q) I(Vamm2P) I(Vamm2Q)

.end
```

As noted in the netlist given above, the starting value for P for the transient simulation was calculated from a separate **Xyce** `.DC` simulation. This is an important caveat for the simulation of transient events in power grids with **Xyce** 6.3. For most devices, **Xyce** can do a combined `.DC` and `.TRAN` simulation with one netlist. However, the power grid simulations use different models (and hence netlists) for the generating units for the steady-state and transient simulations. So, in **Xyce** 6.3, separate simulation runs must be done. The netlist for this power flow calculation is shown below. (Note: For this simple two-bus example, a theoretical answer for the power flow (P) at bus 1 is available. See pp. 250-251 of [7] for more details.)

```
* Steady State Power Flow Calculation for Classical Generator Example

*********************************************
* Simulation control statements             *
*********************************************
.DC V2VM 1 1 1

*******************************************************************
* Bus 1 will be the IEEE Classical Generator during the transient *
* simulation.  Theta and |V| values chosen arbitrarily.           *
*******************************************************************
Vbus1Th bus1Th ammBus1P 0.0873
Vbus1VM bus1VM ammBus1Q 0.95V
Vamm1P 0 ammBus1P 0V
Vamm1Q 0 ammBus1Q 0V

*******************************************************************
* Bus 2 is the slack bus (or infinite bus) during the transient   *
* simulation.  Ammeters are wired so that positive power (P or Q) *
* flows into Bus 2.                                               *
*******************************************************************
V2Th Bus2Th ammBus2P 0V
V2VM Bus2VM ammBus2Q 1V
Vamm2P 0 ammBus2P 0V
Vamm2Q 0 ammBus2Q 0V

*********************************************
* Branch Definition                         *
*********************************************
YPowerGridBranch pg1_2 bus1Th bus2Th bus1VM bus2VM AT=PQP R=0 X=0.1 B=0

***********************************************************
* Start the simulation at the chosen VM and Theta values. *
* Goal is to calculate the steady state P value at Bus 1. *
***********************************************************
.IC V(bus1Th)=0.0873 V(bus1VM)=0.95 V(bus2Th)=0 V(bus2VM)=1

*********************************************
```

```
* Output Statements                          *
*********************************************
.PRINT DC V(bus1Th) V(bus1VM) V(bus2Th) V(bus2VM)
+ I(Vamm1P) I(Vamm1Q) I(Vamm2P) I(Vamm2Q)

.end
```

Figure 4.5 shows the resultant “step response” from t=0 to t=80 seconds to a 1% change in the mechanical input power . The Θ values are plotted in radians in this figure, while the P values are in per unit. Both quanitities exhibit damped oscillations as they converge to their new steady-state values.

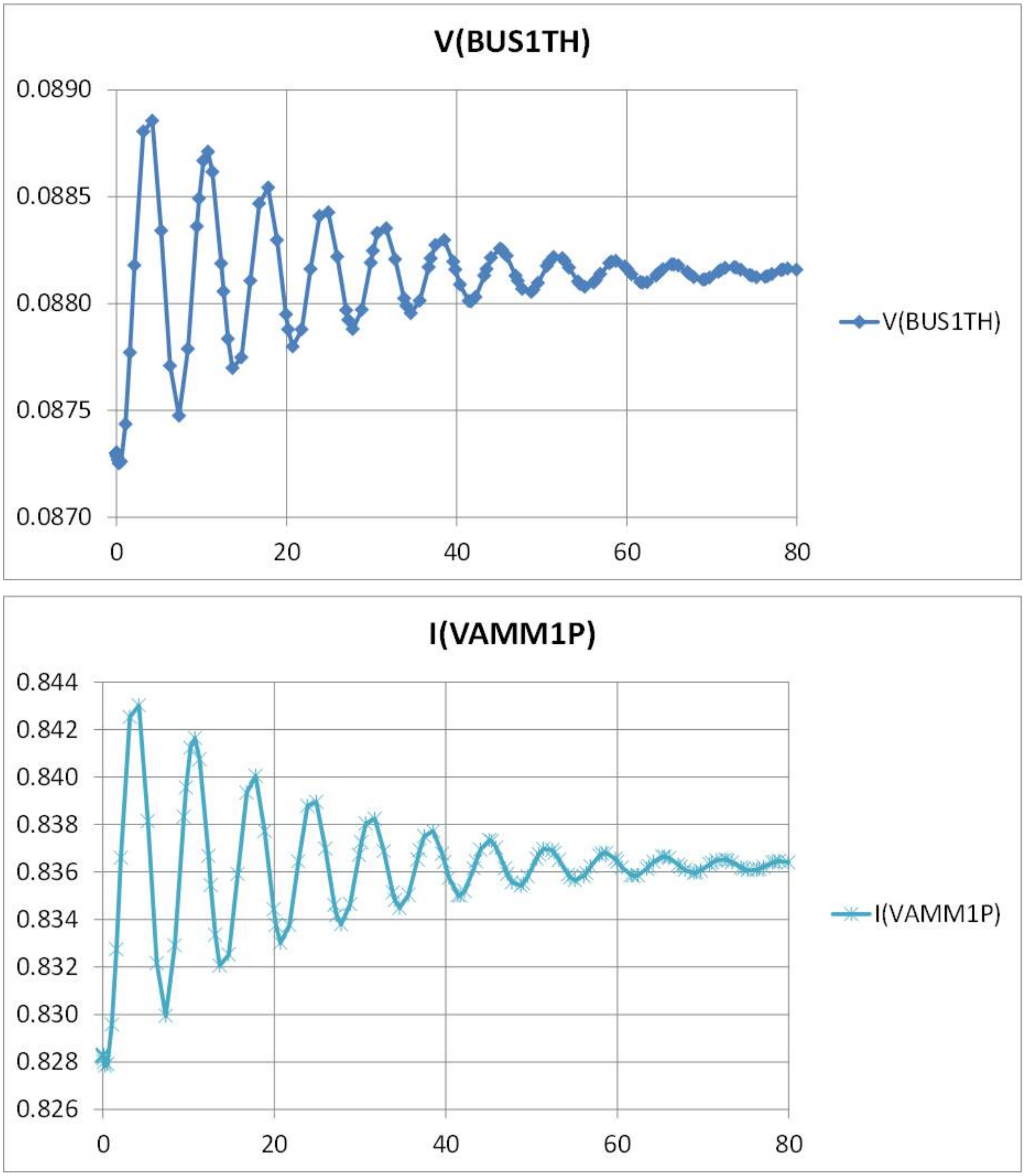

Figure 4.5. Step Response of a Classical Generator Connected to an Infinite Bus.

References

[1] *University of Washington Power Systems Test Case Archive, 14 Bus Power Flow Test Case*. `http://www.ee.washington.edu/research/pstca/pf14/pg_tc`, 1999.

[2] Eric R. Keiter, Ting Mei, Thomas V. Russo, Richard L. Schiek, Peter E. Sholander, Heidi K. Thornquist, Jason C. Verley, and David G. Baur. Xyce Parallel Electronic Simulator: User's Guide, Version 6.3. Technical Report SAND2015-3377, Sandia National Laboratories, Albuquerque, NM, 2015.

[3] Tom Quarles. Spice3f5 Users' Guide. Technical report, University of California-Berkeley, Berkeley, California, 1994.

[4] *The Trilinos Project*. `http://trilinos.sandia.gov`, 2002.

[5] Eric R. Keiter, Ting Mei, Thomas V. Russo, Richard L. Schiek, Peter E. Sholander, Heidi K. Thornquist, Jason C. Verley, and David G. Baur. Xyce Parallel Electronic Simulator: Reference Guide, Version 6.3. Technical Report SAND2015-3381, Sandia National Laboratories, Albuquerque, NM, 2015.

[6] Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra, Lon J. Waters, and Thomas V. Russo. Xyce parallel electronic simulator design: Mathematical formulation, version 2.0. Technical Report SAND2004-2283, Sandia National Laboratories, Albuquerque, NM, June 2004.

[7] Prabha Kundur. *Power System Stability and Control*. McGraw Hill, 1994.

[8] Federico Milano. *Power System Modelling and Scripting*. Springer, 2010.

[9] Farid N. Najm. *Circuit Simulation*. Wiley, 2010.

[10] Abnita Munankarmay and Michael A. Heroux. A comparison of two equivalent real formulations for complex-valued linear systems part 1: Introduction and method. *American Journal of Undergraduate Research*, 1(3), 2002.

DISTRIBUTION:

1 MS 0899 Technical Library, 9536 (electronic copy)

Sandia National Laboratories