Simulation Modelling Practice and Theory 24 (2012) 84–102

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Development and implementation of hydro turbine and governor models in a free and open source software package

Wei Li, Luigi Vanfretti *, Yuwa Chompoobutrgool

KTH Royal Institute of Technology, School of Electrical Engineering, Electric Power Systems, Teknikringen 33, SE-100 44 Stockholm, Sweden

ARTICLE INFO

Article history:
Received 28 October 2011
Received in revised form 13 February 2012
Accepted 14 February 2012
Available online 14 March 2012

Keywords:
Computer simulation
Hydro turbine models
Hydro governor models
Power system modeling
Power system stability
Free and open source software
PSAT
Power system simulation
FOSS

ABSTRACT

This article studies accurate and detailed hydro turbine and governor models, and implements these models in Power System Analysis Toolbox (PSAT)—a free and open source software. The implementation of these models in PSAT is approached using a general methodology, which is described in detail. To evaluate the performances of the developed hydro turbine and governor models, simulation studies are carried out on test systems of different scales, from a single-machine infinite-bus (SMIB) system to a larger system which includes 20 generators. Further more, transient stability analysis and small signal stability analysis are carried out to assess the performance of the implemented models.

1. Introduction

Hydro-power in Sweden accounts for one half net production of electrical power generation [1]. Furthermore, as one of the most important renewable energy sources, the exploitation of hydro-power naturally attracts more and more attention worldwide. In hydro power production systems, the functions of hydro turbine and governors cannot be neglected because they participate in primary frequency control of power systems. To highlight this fact, Fig. 1 shows the relationship of a generator's turbine and governor (indicated by the dashed block) with the overall power system.

Different power sources determine the physical features and categories of hydro turbine and governors, thus each type requires its unique modeling. For example, time delays in hydro turbine and governor models are due to transient droop compensation, pilot valves and gate servomotors. Additionally, the effect of water inertia in hydro turbines significantly influences the governor's requirements and must be compensated by a rate feedback, which must be reflected in the modeling of these devices [2]. In this case, it is important that hydro turbine and governors are adequately modeled so that the models can accurately represent the behaviors and capacities of hydro turbine and governors as close as possible. To examine model performance, the developed models are implemented in test power systems of different scales and their stabilities are evaluated: i.e. if the system remains in equilibrium after being subjected to a disturbance [3]. To this end transient (large signal) stability and small signal (small disturbance) stability analysis are frequently utilized as system stability analysis approaches [4]; in this study both approaches are used.

* Corresponding author. Tel.: +46 87 90 62 25; fax: +46 87 90 65 10.
E-mail addresses: luigiv@kth.se, luigi.vanfretti@ee.kth.se, lvanfretti@ieee.org (L. Vanfretti).

1569-190X/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2012.02.005

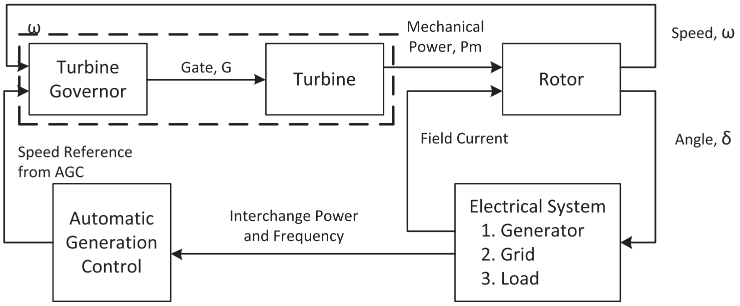


Fig. 1. Functional block diagram showing relationship of the turbine and governor with the overall power system.

On the other hand, computer-based simulation tools are used by engineers for modeling and simulation to assess the stability dynamic and performance of the power system. With advances in computer software, there has been a large number of power system simulation software packages developed, which can be basically divided into three categories: proprietary software, open source software and free software [5]. Proprietary software are conceived by the general public to be well-tested and computationally efficient. However, license agreements restrict their use by imposing different conditions. Users are prohibited and generally have no possibility to modify the source code nor to distribute binaries (i.e. executables). Additionally, they seldom allow to import or export data in different formats [6].

Nevertheless, free and open source software are usually freely (at no cost) distributed on line and are less cumbersome for education and research purposes. More importantly, they allow users to change the source code, add new algorithms, or implement new components. In addition, free software stands on an ethical pillar which aims to warrant intrinsic freedom of computer uses that are jeopardized by proprietary software [7]. Despite theses advantages, there still exist some implementation challenges; particularly for new users. One of these challenges is that it is seldom documented in a detailed fashion the necessary approaches for implementing new components or algorithms in free and open source software. This severely restricts free and open source software to be accepted and popularized by engineers, and also decreases individuals' cognition of the software's flexibility characteristics.

The objective of this paper is to develop hydro turbine and governor models and implement them in MATLAB-based Power System Analysis Toolbox (PSAT)—a free and open source software. To evaluate models' performance, the relevant simulation will be performed on a single-machine infinite-bus (SMIB) system and another larger system, e.g. the KTH-NORDIC 32 system, for both transient and small signal stability analyses.

In view of the description above, the contributions of this paper can be summarized as follows:

- To develop four detailed hydro turbine and governor models which can represent the behaviors of hydro turbine and governor precisely, e.g. water hammer effect.
- To exhibit a systemic, clear and general approach for implementing new components in PSAT—a free and open source software.
- To illustrate a detailed description for how to implement new hydro turbine and governor models in PSAT.
- To perform different system stability analysis approaches and their application in PSAT, e.g. transient stability analysis and small signal stability analysis.

The remainder of this paper is organized as follows. Section 2 describes how to develop models for hydro turbine and hydro governor, respectively, and presents an example model with its structure and realization description. In Section 3, a general approach for implementing components in PSAT is described, as an example, this approach is illustrated with the implementation of a hydro turbine and governor model in PSAT. Next, Section 4 shows simulation results and discusses the performance of the implemented models in the SMIB system as reflected by transient stability analysis and small signal stability analysis. Implementation of turbine and governor models in a large power system is carried out in Section 5. Conclusions are drawn in Section 6 highlighting the contributions reported in this paper.

2. Hydro turbine and governor modeling

In this section the hydro turbine and hydro governor models which are implemented in PSAT are throughly described and their realization diagrams are derived.

2.1. Hydro turbine modeling

As shown in the hydro turbine schematic diagram in Fig. 2, at steady state the net force on the water in the penstock is equal to the rate of momentum change of the water according to Newton's Second Law [3]:

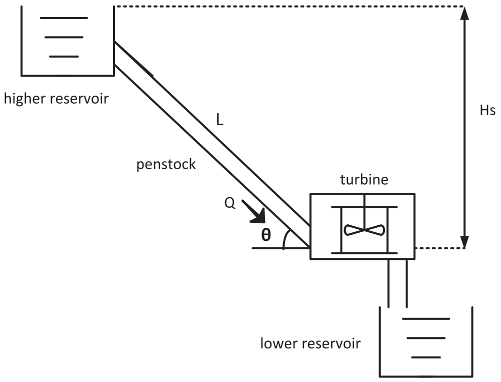


Fig. 2. Schematic diagram of hydro turbine.

$$F_{net} = \rho L \frac{dQ}{dt} \tag{1}$$

where ρ the mass density of water, L the length of penstock, and Q the volumetric flow rate.

The net force on the water in the penstock can be obtained by considering the pressure head [3]:

$$F_{net} = \rho(H_s - H_l - H)Ag \tag{2}$$

where H_s, H_l and H are static head, head loss, and head at turbine gate, respectively. Combining the Eqs. (1) and (2), the rate of change of the volumetric flow rate can be described as:

$$\frac{dQ}{dt} = \frac{(H_s - H_l - H)Ag}{L} \tag{3}$$

Common bases (h_{base} and q_{base}) are used to normalize the above equation, resulting in

$$\frac{dq}{dt} = \frac{(1 - h_l - h)Agh_{base}}{Lq_{base}} = \frac{(1 - h_l - h)}{T_w} \tag{4}$$

where $T_w = \frac{Lq_{base}}{Agh_{base}}$ is the water starting time or water time constant. T_w defines the time period it takes for water at the head h_{base} to obtain the flow rate of q_{base}. h_l is proportional to the flow rate square and penstock friction factor, as described by $h_l = k_f q^2$. The pressure head across the turbine is related to the flowrate as $h = \frac{q^2}{G^2}$, where G describes the gate position from 0 (closed) to 1 (fully open) [3].

The mechanical power developed by the hydro turbine is proportional to pressure head and flow rate. As there is no-load flow q_{nl}, it should be subtracted from the actual flow rate. If damping effects are also taken into account [3], then

$$P_m = A_t h(q - q_{nl}) - DG\Delta\omega \tag{5}$$

where A_t accounts for the difference in per units for both sides of the equation, as follows

$$A_t = \frac{\text{turbine power (MW)}}{\text{generator rating (MVA)}} * \frac{1}{h_r(q_r - q_{nl})} \tag{6}$$

The nonlinear turbine model encompassing the equations above is drawn in Fig. 3.

Linearizing the turbine characteristics around an operating point can aid in simplifying the turbine model. In hydro turbine systems, the mechanical torque/power increment m and the water flow increment q are dependent on the gate opening increment g, the rotational speed increment n, and the head increment h. The following linearized equations are valid to represent the turbine for small perturbations around a steady state condition [2].

$$q = a_{11}h + a_{12}n + a_{13}g; \quad m = a_{21}h + a_{22}n + a_{23}g \tag{7}$$

where

$$a_{11} = \frac{\partial q}{\partial h}, \quad a_{12} = \frac{\partial q}{\partial n}, \quad a_{13} = \frac{\partial q}{\partial g}$$
$$a_{21} = \frac{\partial m}{\partial h}, \quad a_{22} = \frac{\partial m}{\partial n}, \quad a_{23} = \frac{\partial m}{\partial g}$$

When the rotational speed deviation is small ($n \approx 0$), the turbine transfer function which relates mechanical power increment in attaining gate position increment is

$$\frac{m}{g} = \frac{a_{23} + (a_{11}a_{23} - a_{13}a_{21})T_w s}{1 + a_{11}T_w s} \tag{8}$$

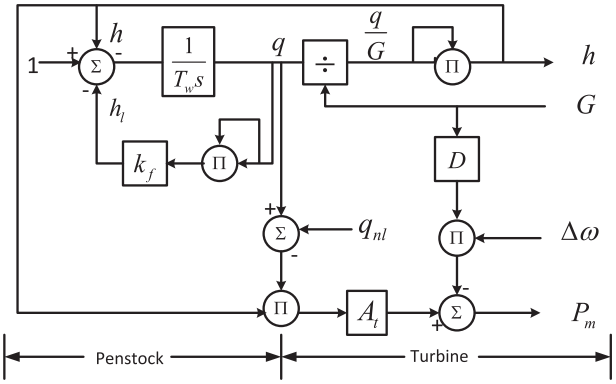


Fig. 3. Nonlinear turbine model (adapted from Ref. [3]).

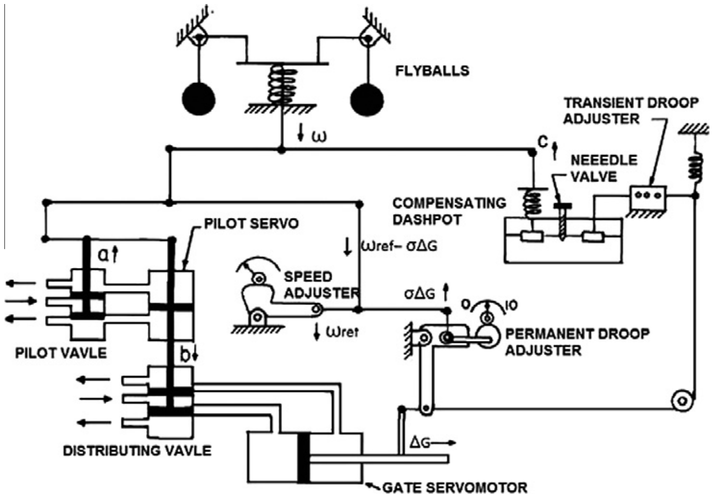


Fig. 4. A simplified schematic of a mechanical hydraulic governor (adapted from Ref. [8]).

2.2. Hydro governor modeling

A simplified schematic of a mechanical hydraulic governor is shown in Fig. 4. In steady state, the rotor speed ω is compared to the reference speed ω_{ref}, and is modified by the permanent droop compensation $\sigma\Delta G$. When the gate position is changing, a transient droop compensation is developed to oppose fast changes in the gate position. These signals are transmitted from mechanical motion to the operation of the pilot valve through the floating levers system [8].

According to the schematic, the derived transfer function of the pilot valve a and pilot servo b is

$$\frac{b}{a} = \frac{K_1}{1 + T_p s} \tag{9}$$

The transfer function between the disturbing valve b and gate position change ΔG is

$$\frac{\Delta G}{b} = \frac{K_2}{s} \tag{10}$$

Combining (9) and (10) yields in

$$\frac{\Delta G}{a} = \frac{K_1 K_2}{(T_p s + 1)s} = \frac{1}{T_g} \frac{1}{(T_p s + 1)s} \tag{11}$$

where $\frac{1}{T_g} = K_1 K_2$. T_g is equal to the time in seconds for a 1 p.u. change in frequency to produce a 1 p.u. change in gate position. Assuming the flow of dashpot fluid through the needle valve is proportional to the dashpot pressure, the compensating dashpot transfer function is

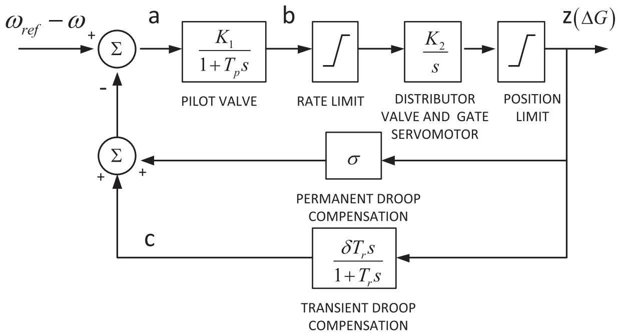

$\omega_{ref} - \omega$
a
b
c
$z(\Delta G)$
$\frac{K_1}{1+T_p s}$
PILOT VALVE
RATE LIMIT
$\frac{K_2}{s}$
DISTRIBUTOR VALVE AND GATE SERVOMOTOR
POSITION LIMIT
σ
PERMANENT DROOP COMPENSATION
$\frac{\delta T_r s}{1+T_r s}$
TRANSIENT DROOP COMPENSATION

Fig. 5. Transfer function block diagram of a typical hydro turbine governor model.

$$\frac{c}{\Delta G} = \frac{\delta T_r s}{1 + T_r s} \tag{12}$$

Through the action of the system's floating levers, the pilot valve input signal a is determined by the reference speed ω_{ref}, rotor speed ω, permanent droop $\sigma\Delta G$, and temporary droop signal c:

$$a = \omega_{ref} - \omega - \sigma\Delta G - c \tag{13}$$

The relationship between the speed difference $\omega_{ref} - \omega$ and the gate position change ΔG can be described as shown in Fig. 5. The rate limit may occur for large or rapid speed deviation, and the position limit may correspond to wide open valves or the setting of a load limiter.

2.3. Hydro turbine and governor models

The hydro turbine and its governor are normally combined together for representation. However sometimes the output of turbine governor is the derivative of gate position, which does not match the input of the turbine. Therefore, a desired gate position reference is applied to add to the gate position derivative.

The report [9] presents four developed hydro turbine and governor models. Since PAST already has two turbine and governor models, Model 1 (thermal turbine and governor model) and Model 2 (simiplied turbine and governor model), the new implemented models will be ordered sequentially afterwards. As shown in Table 1, the first new model, namely, Model 3, consists the typical hydro turbine and governor model. Model 4 contains a simple PI controller in front of Model 3 [10]. A nonlinear turbine model [10,11] is used in Model 5 [12], and the nonlinear hydro turbine and governor model which is available in SPS [13] as HTG [14] and implemented in PSAT is called Model 6. As space is limited, only Model 3 is used as an example in this paper. Detailed structure and model realization for software implementation is described below.

Normally, a general device model can be described by a set of differential and algebraic equations (DAEs) of the form

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_e, \mathbf{y}_e, \mathbf{p}_i, \mathbf{u}_i); \quad 0 = \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_e, \mathbf{y}_e, \mathbf{p}_i, \mathbf{u}_i) \tag{14}$$

where the sub-index i indicates specific device internal variables and the sub-index e indicates external variables from other devices; **f** are the differential equations, **g** are the algebraic equations, **x** are the state variables, **y** are the algebraic variables, **p** are device parameters assigned in the data file, and **u** are controllable variables (such as reference signals in control loops) [15].

For the turbine and governor model (14) can be simplified:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u}); \quad 0 = \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{u}) \tag{15}$$

where the most important state variables **x** are generator rotor angles δ and generator rotor speeds ω. The vector of algebraic variables **y** includes system variables (e.g. bus voltage magnitudes), internal variables (e.g. internal input and output signals) and output variables (e.g. mechanical power); and **u** are controllable and/or specified signals (e.g. speed references) [15].

2.3.1. Model 3—typical hydro turbine and governor

The structural diagram of Model 3 is shown in Fig. 6. It consists of a typical hydro turbine governor model and a linearized hydro turbine model. The output of turbine governor is the gate position derivative (ΔG), while the input of the turbine is the gate position (G). Consequently, a position reference G_{ref}, which is regarded as equal to P_{ref}, is required between them.

To implement models in computer software and perform simulations, a set of state variables from a set of first-order coupled differential equations is necessary. However, it is difficult to map the state variables in the structural diagram shown in Fig. 6. The solution is redrawing the models by using only integrators and gain blocks. In this way, along the signal flow each state variable is located behind each integrator, while the derivative of state variables labeling in front of each, as shown in

Table 1
Summary of governor models and turbine models combinations.

Model	Turbine governor model selection	Turbine model selection
Model 3	Typical model	Linear model
Model 4	Typical mode with PI controller in front	Linear model
Model 5	PI controller combined with servomotor	Nonlinear model
Model 6	PID controller combined with servomotor	Nonlinear model

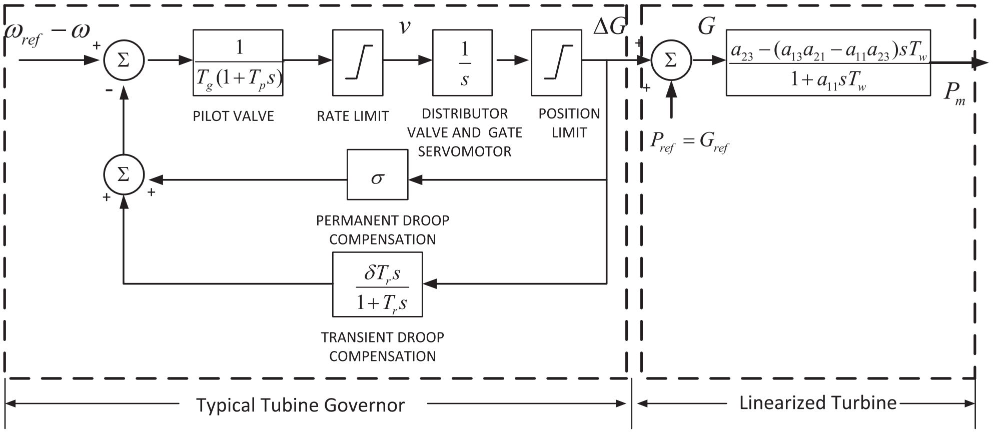

$\omega_{ref}-\omega$
+
-
Σ
$\frac{1}{T_g(1+T_p s)}$
PILOT VALVE
RATE LIMIT
v
$\frac{1}{s}$
DISTRIBUTOR VALVE AND GATE SERVOMOTOR
POSITION LIMIT
ΔG
G
$\frac{a_{23}-(a_{13}a_{21}-a_{11}a_{23})sT_w}{1+a_{11}sT_w}$
P_m
$P_{ref}=G_{ref}$
σ
PERMANENT DROOP COMPENSATION
$\frac{\delta T_r s}{1+T_r s}$
TRANSIENT DROOP COMPENSATION
Typical Tubine Governor
Linearized Turbine

Fig. 6. Diagram showing the general structure of Model 3.

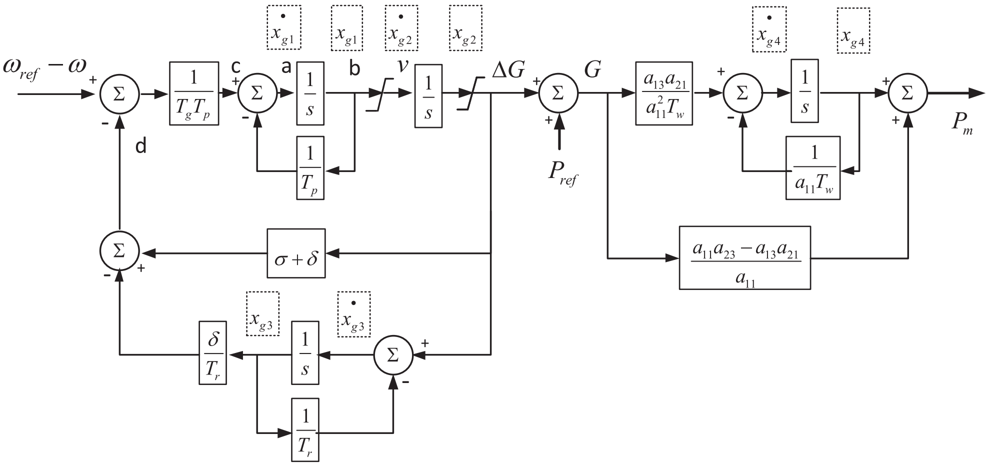


Fig. 7. Block diagram realization of Model 3.

Fig. 7. The number of state variables is equal to the total number of integrators in a system. This is an specific realization of the mathematical model described by the block diagram in Fig. 6 suitable for computer implementation [16].

From the block diagram realization in Fig. 7, it is now possible to derive a set of differential and algebraic equations which can be utilized for implementing the model in PSAT and to calculate the integrators' initial values. The derived DAEs are

$$\dot{x}_{g1}=\frac{1}{T_gT_p}\left[(\omega_{ref}-\omega)-(\sigma+\delta)\Delta G+\frac{\delta}{T_r}x_{g3}\right]-\frac{1}{T_p}x_{g1}$$

$$\dot{x}_{g2}=v=\begin{cases}x_{g1} & \text{if } v_{max}\geqslant x_{g1}\geqslant v_{min}\\ v_{max} & \text{if } v_{max}<x_{g1}\\ v_{min} & \text{if } v_{min}>x_{g1}\end{cases}$$

$$\dot{x}_{g3} = \Delta G - \frac{1}{T_r} x_{g3}$$

$$\dot{x}_{g4} = \frac{a_{13}a_{21}}{a_{11}^2 T_w}(\Delta G + P_{ref}) - \frac{1}{a_{11}T_w} x_{g4}$$

$$\Delta G = \begin{cases} x_{g2} & \text{if } G_{max} - P_{ref} \geqslant x_{g2} \geqslant G_{min} - P_{ref} \\ G_{max} - P_{ref} & \text{if } G_{max} - P_{ref} < x_{g2} \\ G_{min} - P_{ref} & \text{if } G_{min} - P_{ref} > x_{g2} \end{cases}$$

$$P_m = x_{g4} + \frac{a_{11}a_{23} - a_{13}a_{21}}{a_{11}}(\Delta G + P_{ref})$$

When the system is in steady state, $\omega = \omega_{ref}$, the rate of the gate movement $v = 0$, and the gate is fixed as $\Delta G = 0$. The initial values for Model 3 can be obtained by setting the derivatives of the differential equations to zero, as follows

$$x_{g1} = x_{g2} = x_{g3} = 0; \quad x_{g4} = \frac{a_{13}a_{21}}{a_{11}} P_{ref}; \quad P_m = a_{23} P_{ref}$$

This initial values can be also determined by analyzing Fig. 7. When the system is in steady state, $\omega = \omega_{ref}$, all the derivatives of all state variables are zero. Consequently, point a and b are zero. Because of $c - b = a$, c is accordingly equal to zero. Similarly, point d is also zero as $(\omega_{ref} - \omega)$ and c are zero. With this recurrence method, it is possible to obtain all the state variables' initial values that are as same as those obtained from the DAEs above. This approach can be used to check if the calculation using the DAEs is correct.

3. Implementation in free and open source software

PSAT is an open source MATLAB and GNU/Octave-based software package for analysis and design of electrical power systems [5,15,17,18]. It is a very flexible and modular tool for power flow (PF), continuation power flow (CPF), optimal power flow (OPF), small signal stability analysis (SSSA) and time domain simulation (TDS). In addition, a variety of static and dynamic models are provided. Both graphic user interface (GUI) and command line execution can be utilized for calculations and simulations. Since MATLAB is a proprietary software, PSAT can also run on the latest GNU/Octave [19] release to realize its full free software potential. The greatest advantage of PSAT is that it is free and open source, allowing the user to thoroughly know about components' inside structures and power flow calculations, and even to develop their own models.

PSAT is written by using classes and object-oriented programming techniques. Each device is defined by a class with attributes and methods [15]. The advantages of using classes include: easing the maintenance of the code, facilitating modularity and reusing well assessed code. Moreover, class modularity allows one to quickly master the code. Normally, each type device in power system corresponds to one class, and all classes PSAT are documented in the PSAT manual [15].

3.1. A general approach for implementing models in PSAT

To implement a newly developed model in PSAT, firstly, the following common properties regarding the model (such as the values of parameters, maximum amount of parameters) have to be determined [15].

1. *con*: device data in the form of a matrix. Each row defines a new instance of the device, while each column defines a parameter. The content of the matrix *con* varies with each device.
2. *format*: a string containing the format of each row of the *con* matrix. Used for printing PSAT data to files.
3. *n*: total number of devices. This is also the amount of rows in *con* matrix.
4. *ncol*: maximum number of columns of the matrix *con*.
5. *store*: data backup. This is a copy of the *con* matrix. This attribute can be used for automatically modifying some input data and can be useful for command line usage.
6. *u*: vector containing the status of the devices. 1 means on-line, 0 means off-line. Some devices can lack this attribute (i.e. buses).

Then, for the model implementation, some common and particular methods are utilized to describe each model. As shown below, the common methods include Jacobian matrix computation, initialization determination, differential equations implementation and so on [15].

1. *add*: adds one or more instances of the device.
2. *base*: converts device parameters to system power and voltage bases.
3. *block*: defines special operations of the device mask (used only for Simulink models).

4. *display*: prints the class properties in a structure-like format.
5. *dynidx*: assigns indexes to the state variables of the device.
6. *fcall*: computes differential equations $\mathbf{f}$ of the device.
7. *Fxcall*: computes Jacobian matrices $\mathbf{f}_x$, $\mathbf{f}_y$ and $\mathbf{g}_x$ of the device.
8. *gcall*: computes algebraic equations $\mathbf{g}$ of the device.
9. *getxy*: returns indexes of state and algebraic variables of the device.
10. *Gycall*: computes the Jacobian matrix $\mathbf{g}_y$ of the device.
11. *init*: cleans up all device properties.
12. *mask*: coordinates of the black mask (used in Simulink models and for drawing system).
13. *remove*: removes one or more instances of the device.
14. *restore*: restores device properties as given in the original data file.
15. *setup*: initializes the main device properties.
16. *setx0*: computes the initial value of state variables of the device after power flow analysis.
17. *subsasgn*: assigns device properties. Properties that are not listed in this function cannot be assigned from outside the class.
18. *subsref*: returns device properties. Properties that are not listed in this function cannot be get from outside the class.
19. *XXclass*: class constructor. XX is the device specific code.
20. *xfirst*: assigns an initialization value to state and algebraic variables of the device.
21. *warn*: prints some warning messages.
22. *windup*: applies the anti-windup limiter if necessary.

A flowchart illustrating a general approach for implementing models in PSAT and integrating them in power systems for simulation is shown in Fig. 8.

3.2. Hydro turbine and governor models implementation

This section presents a hydro turbine and governor implementation example with Model 3. The parameters arranged in "***Tg.con***" for Model 3 are illustrated in Table 2.

To add new turbine and governor models in the TGclass, the following files need to be modified. Some of these files are particularly necessary only for turbine and governors, rather than common devices:

1. *base*: in PSAT, when defining the TG class data, the model droop (or gain) is given in p.u. with respect to the synchronous machine power rating. During initialization, the droop (or gain) is converted to the system power base in order to keep the same base with other variables in whole system, like electric and mechanical power.

   It is quite easy to see why this conversion is necessary when considering two machines, called A and B. S_A = 1 MVA and S_B = 100 MVA are the power rates of machine A and B, respectively. When assuming the droops are $R_A = R_B$ on system base,

W. Li, L.Vanfretti and Y. Chompoobutrgool / SIMPAT 00 (2012) 1–28

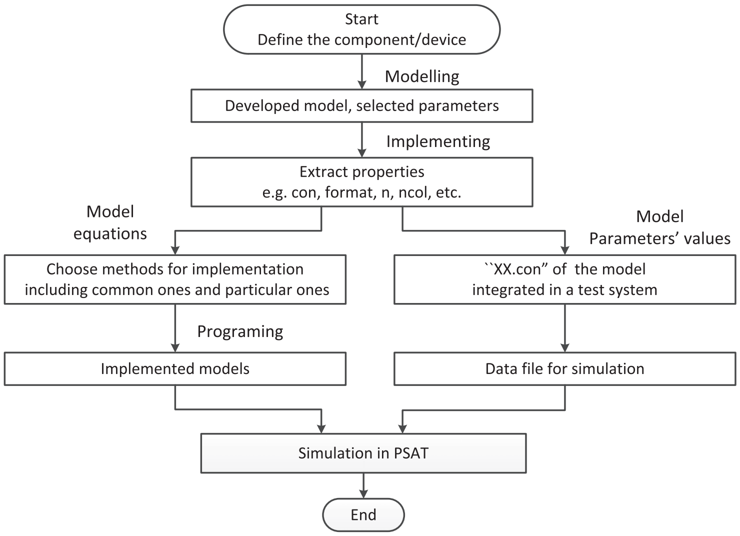

Start
Define the component/device
Modelling
Developed model, selected parameters
Implementing
Extract properties
e.g. con, format, n, ncol, etc.
Model
equations
Model
Parameters' values
Choose methods for implementation
including common ones and particular ones
``XX.con" of the model
integrated in a test system
Programing
Implemented models
Data file for simulation
Simulation in PSAT
End

Fig. 8. A general approach for implementing models in PSAT and system simulation.

Table 2
Turbine and governor Model 3 data format (Tg.con).

Column	Variable	Description	Unit
1	–	Generator number	int
2	3	Turbine governor type	int
3	ω_{ref}	Referenced speed	p.u.
4	T_g	Pilot valve droop	p.u.
5	G_{max}	Maximum gate opening	p.u.
6	G_{min}	Minimum gate opening	p.u.
7	v_{max}	Maximum gate opening rate	p.u.
8	v_{min}	Minimum gate opening rate	p.u.
9	T_p	Pilot valve time constant	s
10	T_r	Dashpot time constant	s
11	σ	Permanent speed droop	p.u.
12	δ	Transient speed droop	p.u.
13	T_w	Water starting time	s
14	a_{11}	$\partial q/\partial h$	p.u.
15	a_{13}	$\partial q/\partial g$	p.u.
16	a_{21}	$\partial m/\partial h$	p.u.
17	a_{23}	$\partial m/\partial g$	p.u.
18	u	Connection status	0.1

the two machines will vary their power production by the same amount following a frequency variation. Clearly, this is not what is requited. What is needed is that, for example, machine B provides 100 times the power variation provided by machine A. Hence, the droop of the two machines, must be $R_A = R_B$ on the machine base, and $R_A = 100R_B$ on the system base. Eq. (16) shows the droop and gain conversions.

$$\frac{1}{R_{system}} = \frac{S_{machine}}{S_{system}} * \frac{1}{R_{machine}}; \quad \text{or} \quad K_{system} = \frac{S_{machine}}{S_{system}} * K_{machine} \tag{16}$$

The gain (droop) is defined as (the inverse of) the limit for $s \Rightarrow 0$ of $F(s)$, where $F(s)$ is the transfer function of hydro turbine governor. In this case, the first mission is to calculate the transfer function and determine the gain (or droop). The transfer function of Model 3 is given by

$$F_3(s) = \frac{\Delta G}{\omega_{ref} - \omega} = \frac{1 + T_r s}{T_g T_r T_p s^3 + T_g(T_r + T_p)s^2 + (T_g + \sigma T_r + \delta T_r)s + \sigma} \tag{17}$$

Parameter σ is the droop of $F_3(s)$ when $s \Rightarrow 0$, which is defined as the 11th parameter in "***Tg.con***" as shown in Table 2. The following code transfers it from machine base to system base according to (16).

```
≫ p.con (i,11) = p.con (i,11).*Settings.mva./getvar (Syn,p.syn (i),'mva');
```

The gate position limits and rate limits also need to be converted into system base because the whole loop of the governor will operate on the system base when performing a simulation.

2. *block*: this file defines the prompt strings in parameter box for turbine and governor models' graphic mask in Simulink environment. All parameters in "*Tg.con*" have to be defined here. The following is an example for Model 3.[1]

```
≫ prompts (idx5) = 'Maximum gate opening rate vmax [p.u.]';
≫                  'Minimum gate opening rate vmin [p.u.]';
≫                  'Pilot valve time constant Tp [s]';...
≫                  'Patical derivative of turbine torque with respect to gate position a23';
≫ enables (idx6) = 'off'; 'off';
```

"*idx5*" and "*idx6*" are indexes indicating the position of each parameter located in the parameter box. For this instance, all the listed parameters are located at positions 6–16 in parameter box, while positions 17 and 18 are vacant for other models utilizing more parameters.

```
≫ idx5 = [6 7 8 9 10 11 12 13 14 15 16];   idx6 = [17 18];
```

The *block.m* file in the TGclass programs the parameter boxes for turbine and governor models' graphic mask in Simulink environment. While *fm_lib* designs more general settings for the whole simulation library, which also contains some turbine and governor models specifications. This *fm_lib* file does not belong to TGclass, but acts as an independent file in the PSAT package. Within this file it is possible to find the "Tg" block definition region.

[1] "..." indicates some code are omitted as space limited. The same cases are applied for the following code fragments.

As the amount of models changes, relevant variables from “MaskStyleString” until “MaskVariables” need to be modified. More strings should be added in “MaskPromptString” to make sure that the maximum parameters amount is the same with that in *block.m*. These strings are only default, and await for specific strings set in the *block.m* file to cover them. “ModelHelp” is used to give a simple description of the model.

3. *dynidx*: indexes for state variables in newly developed models are assigned. For Model 3, there are four state variables, so

```
≫ a.tg1(i) = DAE.n + 1;...; a.tg4(i) = DAE.n + 4; DAE.n = DAE.n + 4;
```

4. *fcall*: differential equations introduced in Section 2 for Model 3 are revealed here.

```
≫% define the names of state variables
≫tg1 = DAE.x (p.dat3(:,1));... tg4 = DAE.x (p.dat3(:,4)); wref = DAE.y (p.wref (p.ty3));
≫% gate position limits...
≫%rate limits...
≫%differential equations
≫ DAE.f (p.dat3(:,1)) = p.u (p.ty3).*(p.dat3(:,5).* (wref-DAE.x (p.dat3(:,17)) -(p.dat3(:,15)+p.-
dat3(:,14)).*delta_G + p.dat3(:,15).*p.dat3(:,8).*tg3)- p.dat3(:,7).*tg1);...
≫ DAE.f (p.dat3(:,4)) = p.u (p.ty3).*(p.dat3(:,11).*p.dat3(:,12).*(p.dat3(:,6)+delta_G)-tg4.*
p.dat3(:,11));
```

p.dat3 is the index for “***Tg.con***” of Model 3 and will be stated in the *setx0* file. DAE.x (p.dat3(:,1)) represents a state variable, which is the first element in the matrix *p.dat3*, namely x_{g1} in Model 3’s description in Section 2 (here is defined the name as t_{g1}). Similarly, other parameters’ index can be found in *p.dat3* as well. While, DAE.f (p.dat3(:,1)) stands for the derivative of state variable x_{g1}, *p.u (p.ty3)* is a scale factor to change the parameters in per unit.

5. *Fxcall*: this file computes the Jacobian matrix $\mathbf{f}_x$, $\mathbf{f}_y$ and $\mathbf{g}_x$ (refer to (18) in Section 4). As (18) indicates, $\mathbf{f}_x$ represents the partial derivatives of function $\mathbf{f}$ with respect to state variables $\mathbf{x}$, and $\mathbf{f}_y$ the partial derivative of function $\mathbf{f}$ with respect to algebraic variables, and $\mathbf{g}_x$ the partial derivative of algebraic function $\mathbf{g}$ with respect to state variables $\mathbf{x}$.

The following matrices indicate the parts of the Jacobian matrix $\mathbf{f}_x$ which are related to the state variables in Model 3. The first element $-\frac{1}{T_p*T_g}$ represents the partial derivative of $\dot{x}_{g1}$ with respect to ω. These matrices are used for writing the “Fxcall” file. The same approach is applicable for writing the Jacobian matrix $\mathbf{f}_y$ and $\mathbf{g}_x$ related to turbine and governor models.

$$\text{Model 3}: \begin{array}{c} \\ \dot{x}_{g1} \\ \dot{x}_{g2} \\ \dot{x}_{g3} \\ \dot{x}_{g4} \end{array} \left(\begin{array}{ccccc} \omega & x_{g1} & x_{g2} & x_{g3} & x_{g4} \\ -\frac{1}{T_p*T_g} & -\frac{1}{T_p} & -\frac{\sigma+\delta}{T_p*T_g} & \frac{\delta}{T_p*T_g*T_r} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{T_r} & 0 \\ 0 & 0 & \frac{a_{13}a_{21}}{a_{11}^2 T_w} & 0 & -\frac{1}{a_{11}T_w} \end{array} \right)$$

The Jacobian matrix is populated primarily with zeros, which coincidences with sparse matrices. Sparse data is by nature easily compressed, and this compression almost always results in significantly less computer data storage usage. PSAT uses sparse matrices for creating the Jacobians, and therefore the Jacobian elements for new devices need to be implemented using sparsity. For example, the following code

```
≫ sparse (tg1, p.dat3 (:,17), u3.*p.dat3 (:,5), DAE.n, DAE.n);
```

determines the partial derivative of $\dot{x}_{g1}$ with respect to state variables ω. The Jacobian matrix $\mathbf{f_x}$ is represented as DAE.Fx, whose size is DAE.n*DAE.n. The value in the x_{g1} row, ω column is u3.*p.dat3 (:,5), which is the partial derivative of $\dot{x}_{g1}$ with respect to state variables ω according to the DAEs of Model 3.

u_G and u_v are logically true (=“1”) only when the gate position and gate rate do not reach limits. For example, when u_v is logically true, the partial derivative $\dot{x}_{g2}$ with respect to x_{g1} is 1, otherwise it is zero; this is implemented with the following code.

```
≫u_G = delta_G < (p.con (p.ty3,5)-p.dat3(:,6)) & delta_G >(p.con (p.ty3,6)-p.dat3(:,6));
≫u_v = v <p.con (p.ty3,7) & v >p.con (p.ty3,8);...
≫DAE.Fx = DAE.Fx- sparse (tg1,tg2,u_G.*u3.*(p.dat3(:,14)+p.dat3(:,15)).* p.dat3(:,5),DAE.n,DAE.n);...
≫DAE.Fx = DAE.Fx + sparse (tg2,tg1,u_v.*u3,DAE.n,DAE.n);
```

6. *init*: initialization properties of newly developed models are added.
7. P_{mec}: mechanical power equations for newly developed models are added.
8. *remove*: remove one or more instances of the newly developed models.
9. *setx0*: the tg.dat defines index of parameters in each model. In the index, all parameters that will be implemented in the model equations are arranged with fixed positions so that they can be called.

```
≫ a.dat3 = [a.tg1(a.ty3), % 1
≫        a.tg2(a.ty3), % 2 ...
≫        Porder3, % 6
≫        ap, % 7 ...
≫        Syn.omega (a.syn (a.ty3)), % 17
≫        zeros (length (a.ty3),1), % 18
≫        zeros (length (a.ty3),1)]; % 19
```

Moreover, the initial values of the state variables and function *f* are set here.

```
≫ DAE.x (a.dat3(:,1)) = 0;...
≫ DAE.x (a.dat3(:,4)) = a.u (a.ty3).*a.dat3(:,12).*a.dat3(:,6);
≫ DAE.f (a.dat3(:,1)) = 0;...
≫ DAE.f (a.dat3(:,4)) = 0;
```

10. *subsasgn*: assign turbine and governor properties for the newly developed models.
11. *subsref*: return turbine and governor properties for the newly developed models.
12. *TGclass*: add the newly developed models in the TGclass constructor. In addition, change *ncol* to be equal to maximum amount of columns in the matrix *con* and modify the corresponding format.

4. Simulation and performance for the implemented models in SMIB system

The single-machine infinite-bus (SMIB) systems, as the name suggests, includes only one generator. It is possible the simplest power system, consequently, is widely used for preliminary evaluation of the implemented models.

Fig. 9 shows a SMIB system consisting of one 991 MVA, 20 kV, 50 Hz generator, one transformer operating 20 kV on the primary and secondary, two lines with reactance of 0.1 p.u. and an infinite generator (or source) with 100,000 MVA (an infinite bus).

For comparison purposes only, the SMIB system with hydro turbine and governor Model 3 to Model 6 are also implemented in SPS. When simulating hydro turbine and governor models in the SMIB system using SPS, the power flow routine cannot automatically set the initial conditions of newly implemented models. In addition there are small oscillations in the beginning of the simulation in SPS as depicted in Fig. 10b from 1 to 5 s. One possible solution is to begin the simulation with a power flow execution file, where the initial values for different integrators are set.

Here is a fragment of the power flow execution file for Model 3. The command "*power_loadflow*" computes the power flow, this is normally realized in the power flow GUI, but not for the case with new models. The command "*set_param*" can be used to set the values of power reference and initial values of four integrators. As long as parameters have valid names in the system, they can be set to desired values by the user. Even though a power flow execution file is used to initialize integrators in the hydro turbine and governor models, the oscillations seen at the beginning of the simulation cannot be mitigated. This reflects one shortcoming of proprietary software: it is not possible for users to access nor to modify the source code by themselves so that this error could be mitigated.

```
≫lfparam (1).set (1).TerminalVolage = 20000;
≫lfparam (1).set (1).ActivePower = 991000000*Pref;
≫%Computes the power flow of the model for the power flow parameters given in LFPARAM struct.
≫lf = power_loadflow (model,lfparam);
≫%Calculate the path of all block of the TG that should be initialized with the ''lf'' variables
≫param_pref_constant=[model'/TG_type3/Pref'];...
≫param_ic_integrator4=[model '/TG_type3/Integrator4'];
≫%Set the parameters of the integrator and constant block
```

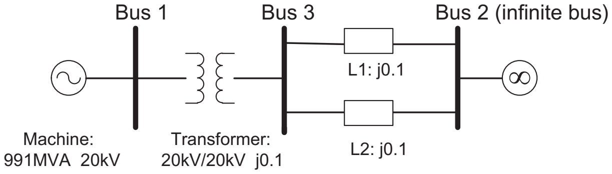
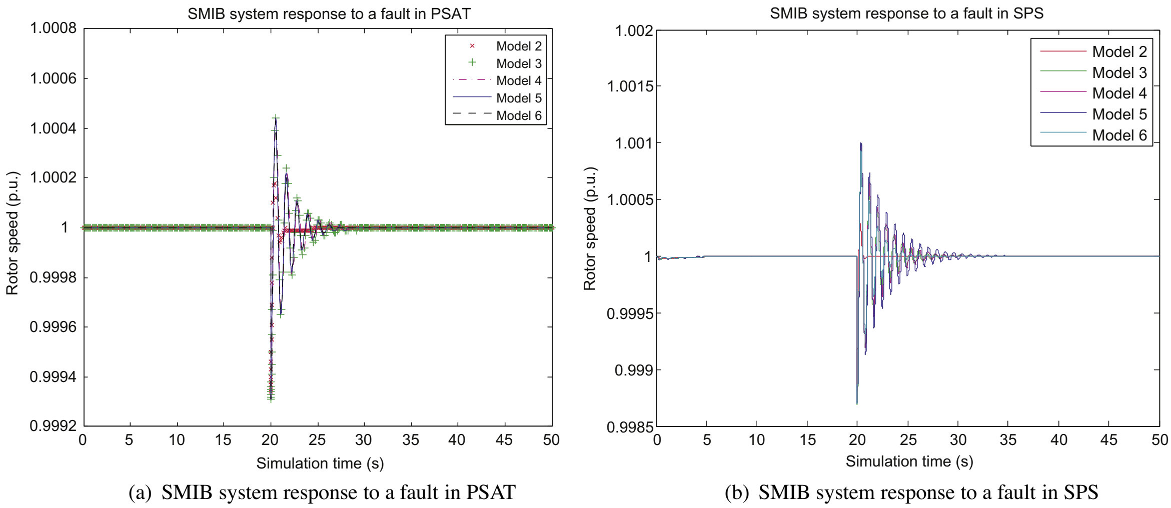


Fig. 9. The SMIB system.



(a) SMIB system response to a fault in PSAT

(b) SMIB system response to a fault in SPS

Fig. 10. SMIB system response to a fault.

```
≫set_param (param_pref_constant, ‘Value’, mat2str (lf (1).Pmec (2)));...
≫set_param (param_ic_integrator4, ‘initialcondition’, mat2str (lf (1).Pmec (2)*3));
≫%Run simulation.
≫sim (’SMIB_test_type3’);
```

4.1. Transient stability analysis

Transient (or large disturbance) stability is the ability of the power system to maintain synchronism when subjected to a severe transient disturbance, such as a fault on transmission line or a load change [4]. In this section we examine the performance of the newly implemented models when used for transient stability simulations.

4.1.1. Response to a fault

A three-phase fault is applied at Bus 3 at $t = 20$ s and removed at $t = 20.02$ s. After computing the power flow, time-domain simulations show the model responses in PSAT. Fig. 10a depicts the generator rotor speed of five models in PSAT and Fig. 10b in SPS for comparison.

4.1.2. Response to a load change

Next, the system response is evaluated for a load change by connecting a load with 0.5 p.u. active power and 0 reactive power on Bus 3 and implementing a 20% increment load change at $t = 2$ s. This simulation is only done in PSAT.

4.1.3. Analysis of results

The time-domain simulations show the models’ performances when subject to a large disturbance. For the responses to a fault, comparing SMIB system transient response in PSAT and SPS, clearly, they are not as close as expected. All parameters and structures for each hydro turbine and governor model in these two software are exactly same, but the settings for other devices, such as transformers, lines, loads, are really hard to get totally coherent. This underlines that the differences are in part due to model exchange barriers between two software. Another phenomenon can prove this statement as well. In SPS, even though Model 6 utilizes a nonlinear turbine model, it behaves faster than Model 3 and Model 4. It is probably because Model 6 is provided by SPS with a more effective compatibility and particular design. Model 5 is supposed to be a bit slower as it is utilizing nonlinear turbine model. Comparatively speaking, Model 3 to Model 6 show relatively same performances to perturbations in PSAT, which is expected for the type of fault implemented.

In the responses to a fault, Model 2 behaves with outstanding stability, which derives from its simple and robust structure. However, this model is not suitable for representing the behavior of hydro turbine and governors, which is indicated in the response to a load change. When the load in a power system increases, the water flow gate has to open wider to meet the power demand. Once the water flow gate suddenly opens, the volume of the water flow will tend to increase which causes water pressure reduction, and the output mechanical power will decrease at first and then turn to increase. This effect is called “water hammer effect” [20,21] as shown in Fig. 12. From Fig. 11a and b, it is shown that Model 2 does not have a water hammer effect, in other words, it cannot represent the performance of hydro turbine and governors.

4.2. Small signal stability analysis

Small signal (or small disturbance) stability is the ability of a power system to maintain steady under small disturbances. The disturbances are considered sufficiently small for linearization of system equations to be permissible; this implies that the system operates in its linear region despite the perturbation.

The linearization of system DAE (15) is as follows:

$$\Delta\dot{x} = f_x\Delta x + f_y\Delta y + f_u\Delta u; \quad 0 = g_x\Delta x + g_y\Delta y + g_u\Delta u \tag{18}$$

Extracting Δy from the second equation above and substituting it into the first one, then we obtain:

$$\Delta\dot{x} = \left(f_x - \frac{f_y g_x}{g_y}\right)\Delta x + \left(f_u - \frac{f_y g_u}{g_y}\right)\Delta u; \quad \Delta y = -\frac{g_x}{g_y}\Delta x - \frac{g_u}{g_y}\Delta u \tag{19}$$

These equations just correspond to the linearization form of a system with input and output matrices **A**, **B**, **C**, **D** as:

$$\Delta\dot{x} = \mathbf{A}\Delta x + \mathbf{B}\Delta u; \quad \Delta y = \mathbf{C}\Delta x + \mathbf{D}\Delta u \tag{20}$$

where

$$\mathbf{A} = f_x - \frac{f_y g_x}{g_y}; \quad \mathbf{B} = f_u - \frac{f_y g_u}{g_y}; \quad \mathbf{C} = -\frac{g_x}{g_y}; \quad \mathbf{D} = -\frac{g_u}{g_y}$$

The equilibrium points are calculated from

$$\dot{x} = f(x_0) = 0$$

The eigenvalues λ_i can be computed from the **A** matrix according to

$$det(\lambda I - \mathbf{A}) = 0 \tag{21}$$

A complex eigenvalue λ is represented by

$$\lambda = \sigma \pm j\omega \tag{22}$$

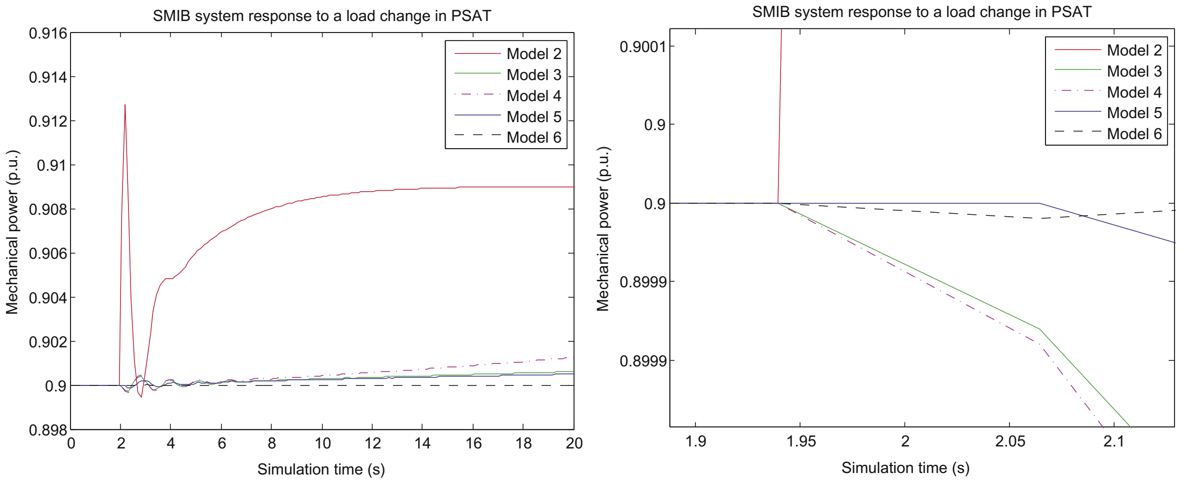
The damping ratio describes the rate of decay the system oscillations after a disturbance. It is given by

$$\zeta = \frac{-\sigma}{\sqrt{\sigma^2 + \omega^2}} \tag{23}$$

and the frequency (which is called "pseudo-frequency" in the PSAT manual [15]) that can be observed during the transient is

$$f = \frac{\omega}{2 * \pi} \tag{24}$$

A lower damping ratio implies a lower decay rate, and influences the dynamic system behavior more than higher damping ratio. In this case, only two lowest damping modes are considered. Table 3 shows the small signal stability analysis results of the two lowest damping modes in SMIB system.



(a) SMIB system response to a load change in PSAT

(b) SMIB system response to a fault in SPS

Fig. 11. SMIB system response to a load change.

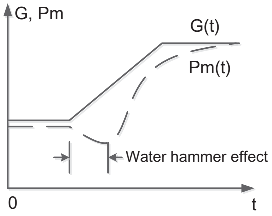


Fig. 12. Water hammer effect.

Table 3
Summary of the linear analysis results of the two lowest damping modes in SMIB system.

Model in system	Frequency (Hz)	Damping ratio	Eigenvalues
Model 2	0.86233	0.48048	−2.9684 ± 5.4182j
	6.6960	0.12865	−5.4581 ± 42,072j
Model 3	0.89782	0.10574	−0.59983 ± 5.6412j
	6.6960	0.12865	−5.4581 ± 42,072j
Model 4	0.89804	0.10394	−0.58970 ± 5.6426j
	6.6960	0.12865	−5.4581 ± 42,072j
Model 5	0.89116	0.11766	−0.66340 ± 5.5993j
	6.6960	0.12865	−5.4581 ± 42,072j
Model 6	0.82586	0.052253	−0.27151 ± 5.1890j
	6.6960	0.12865	−5.4581 ± 42.072j

As Table 3 illustrates, the SMIB system with Model 6 has a significantly small damping ratio for the 0.8 Hz mode, which can probably result in a high risk of instability. This results from its more complex nonlinear structure. Even though its low level damping does not directly affect the transient stability response in SMIB system, it would have a larger effect when implemented it in a more complex network. It is important to highlight that proper modeling of hydro turbine and governors is necessary for small-signal stability analysis; otherwise, simple models not capturing the behavior of hydro mills (such as Model 2) will result in over-optimistic damping calculations.

Fig. 13 depicts zero-pole map for hydro turbine and governor Model 3 and Model 2 in SMIB systems. It is important to note from the zero-pole map of Model 2 that this system is a minimum phase system, as its poles and zeros are all in the left half plane. On the other hand, Model 3 (same cases for Model 4–Model 6) are nonminimum phase systems with one right half plane zero. This is the essential reason which highlights that Model 2 has no water hammer effect. Nonminimum phase system behaves phase lag compared to minimum phase system, which means they are slower at the start of the time-domain response. While this delay is just the property of water flow like shown by Model 3 (same cases for Model 4–Model 6). However Model 2 has no capacity to represent this property of hydro turbine and governor in reality, and should only be used for thermal or steam units.

4.3. Linear and nonlinear model validation comparison

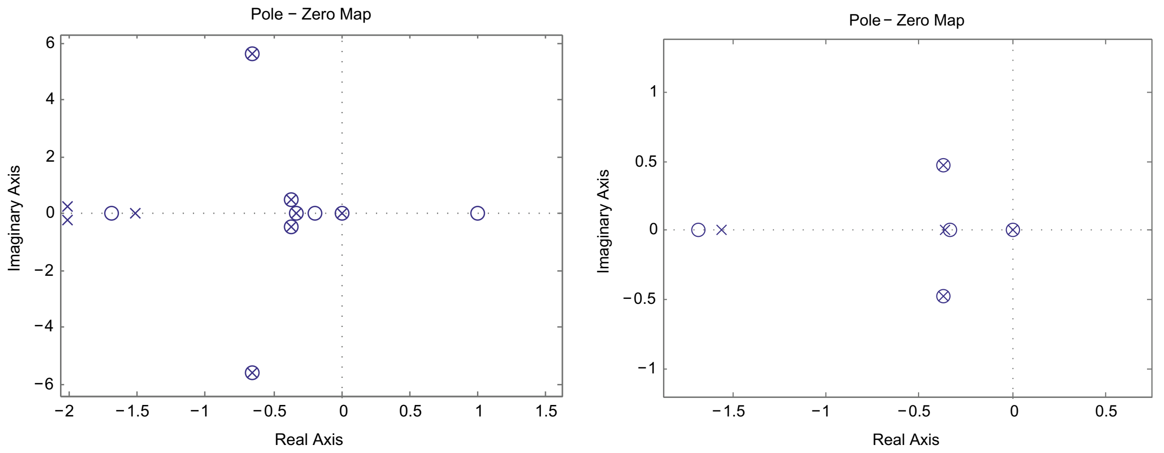
Linear models can be used to simplify power system analysis and allow control design. This section verifies the linear model response to a step change at hydro turbine and governor's speed reference, and finds out how well the linearized model represents the behavior of nonlinear model in the linear operating region where the nonlinear model has been linearized. Both time-domain simulations run in PSAT, and Model 3 serves as hydro turbine and governor in SMIB system.

As shown in Fig. 14, the behavior of the linearized model is considerably close to the nonlinear model. It is demonstrated that the linear model does capture the same dominant modes of the nonlinear model in the time-domain simulation. It is relevant to note that if the Jacobians implemented in Section 3 were incorrect, there would be a discrepancy between the linear and non-linear model responses. Consequently, we can conclude that linearized model is verified to represent nonlinear models. Moreover, linear analyses, such as small signal analysis are therefore accurate enough to provide the stability information for nonlinear model which can be used for control design.

5. Implementation of turbine and governor models in large power systems

5.1. KTH-NORDIC32 system introduction and parameters in PSAT

The KTH-NORDIC32 system is depicted in Fig. 15. The overall topology is longitudinal; two large regions are connected through considerably weak transmission lines. The first region is formed by the north and the equivalent areas located in



(a) Zero-pole map for Model 3 in SMIB (local enlargement) (b) Zero-pole map for Model 2 in SMIB (local enlargement)

Fig. 13. Zero-pole maps for models in SMIB.

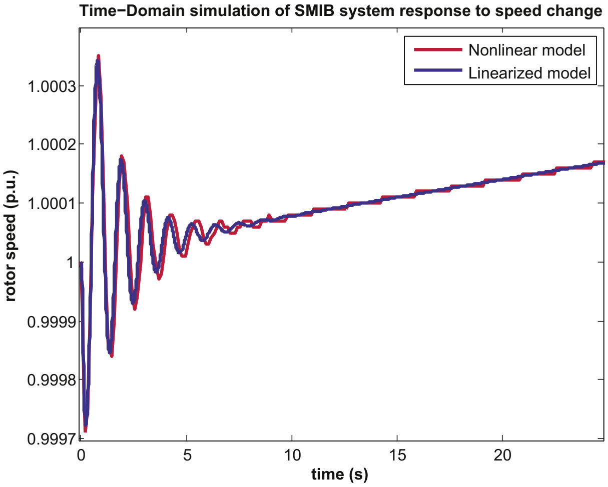


Fig. 14. Time-domain simulation of SMIB system response to speed change.

the upper part, while the second region is formed by the Central and the South areas located in the bottom part. The system has 20 generators, 12 of which are hydro generators located in the North and the Equivalent areas, whereas the rest are thermal generators located in the Central and the South areas. In this paper the 12 hydro turbine and governors will utilize Model 2 or Model 3, while 8 thermal turbine and governors only use Model 1. As discussed in the previous section, Model 2 is unsuitable for representing hydro turbine and governors, and is used here for comparison purpose only. There is more generation in the upper areas while more loads congregate in the bottom areas resulting in a heavy power transfer from the northern area to the southern area, which easily leads to system oscillations [22].

5.2. Transient stability analysis

A three-phase fault is applied at “BUS1011” at t = 5 s and removed at t = 5.02 s. The reason why the fault is set on this bus is to decrease the fault influence on critical generators, and at the same time so that the dominant power flow is not disturbed. This allows for a good comparison of the performances of the turbine and governor models implemented in this paper. Fig. 16a and b depict the response of the generators rotor speeds in the KTH-NORDIC32 System with Model 2 and Model 3, respectively.

Comparing these two simulations, we can easily find out that the system with Model 2 recovers to steady state faster than that with Model 3. Moreover, in the system with Model 3, there are oscillations with particular frequencies swing around a steady value “1”, even when the system already gets back to steady state. In fact this is a normal and common phenomenon in large power system called system oscillations [23–25].

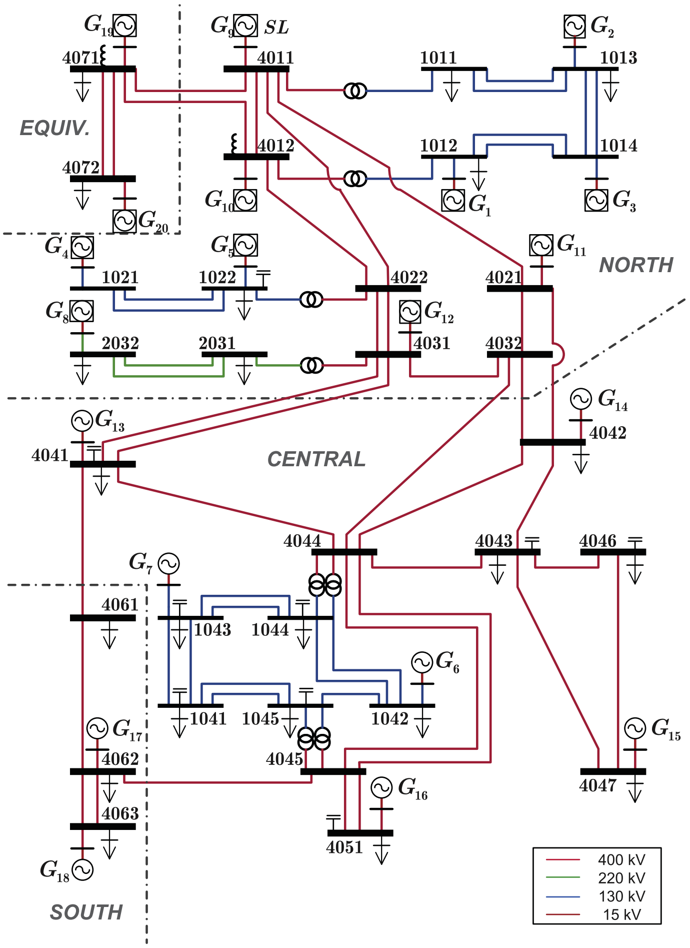
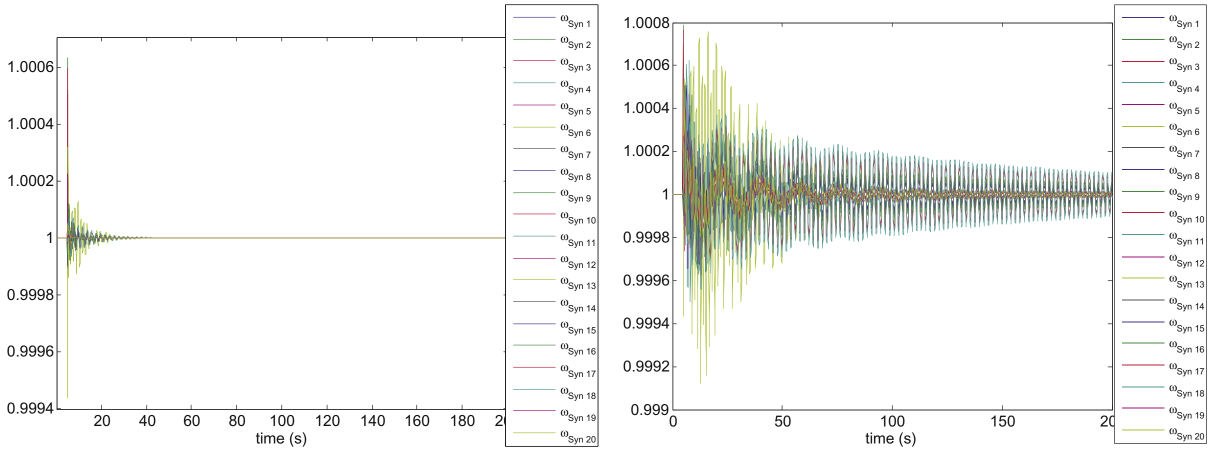


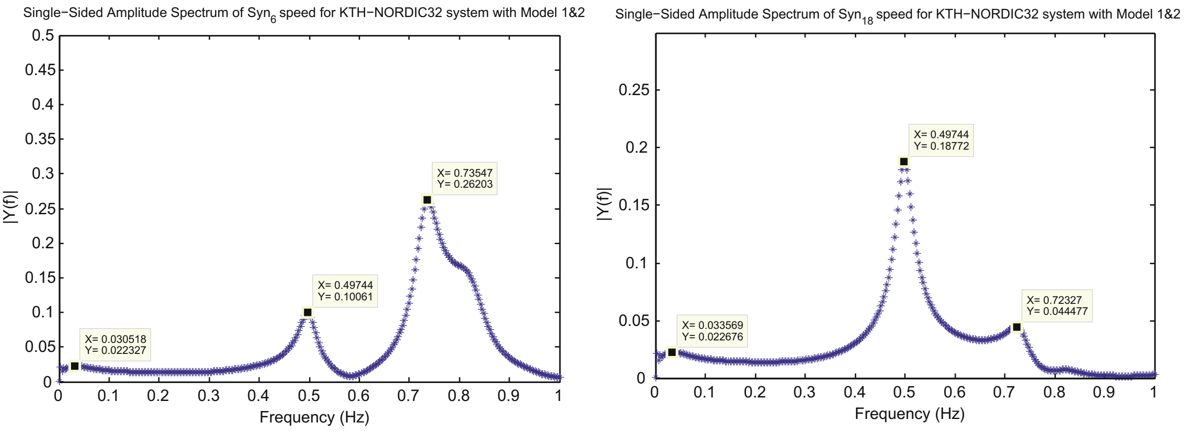
Fig. 15. KTH-NORDIC32 test system (Taken from Ref. [22]). (The generators marked with panes are hydraulic generators, which utilize Model 2 or Model 3 as their hydro turbine and governors in KTH-NORDIC32 test system for simulation; the other generators utilize Model 1 as their steam turbine and governors.)



(a) KTH-NORDIC32 test system with Model 1 and Model 2

(b) KTH-NORDIC32 test system with Model 1 and Model 3

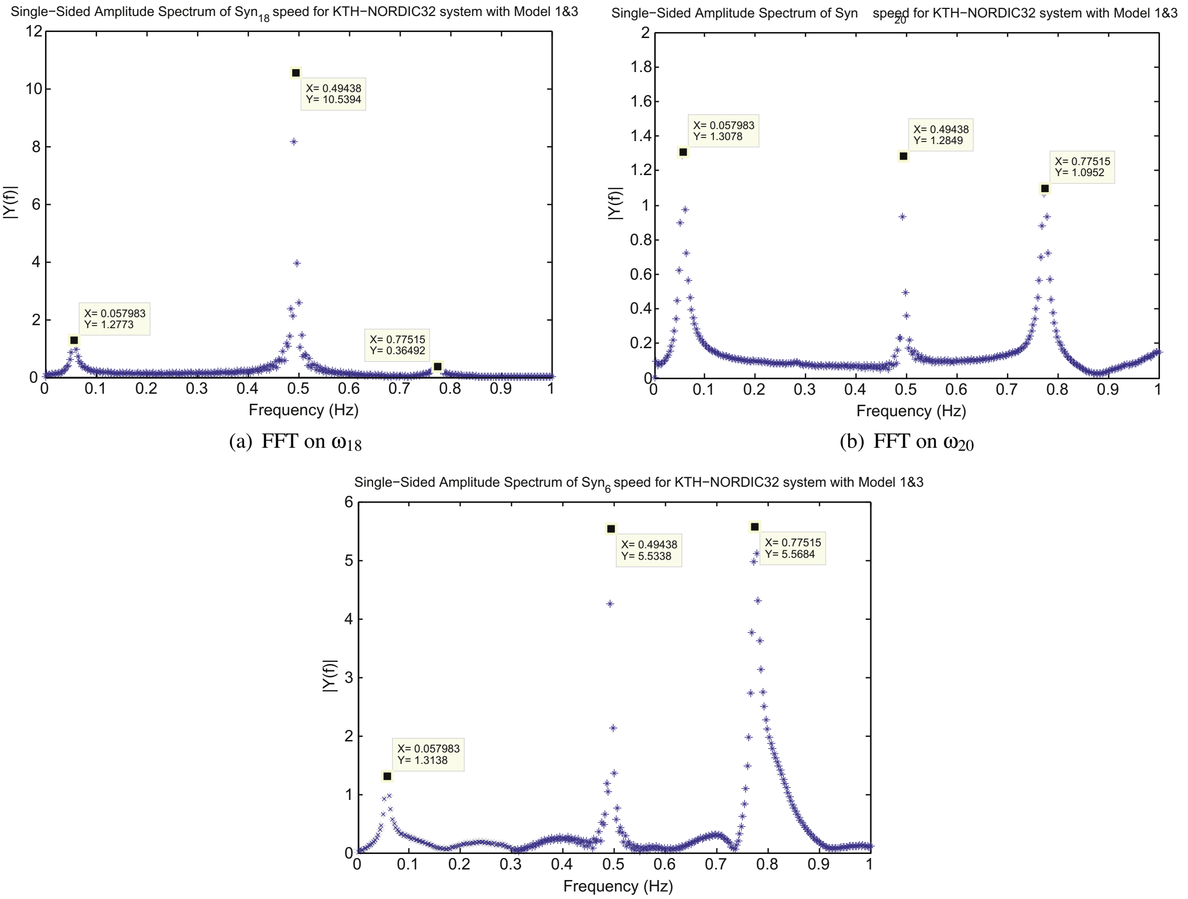
Fig. 16. KTH-NORDIC32 test systems response to a fault.



(a) FFT on ω_6

(b) FFT on ω_{18}

Fig. 17. FFT on rotor speed signals in KTH-NORDIC32 including Model 2.



(a) FFT on ω_{18}

(b) FFT on ω_{20}

(c) FFT on ω_6

Fig. 18. FFT on rotor speed signals in KTH-NORDIC32 including Model 3.

It is difficult to identify the frequencies of system oscillations by looking at the time-domain simulation signals. The fast Fourier transform (FFT) should be used to convert the time-domain signals into frequency domain, so that is possible to determine the particular frequencies in the signals. Making sure that the signal for FFT is from a fixed point time-domain simulation, and that the smaller time step is, the higher resolution it can be obtained. Here a window from 5 to 200 s is

Table 4
Linear analysis results of the two lowest damping modes in KTH-NORDIC32.

Model	Frequency	Damping ratio	Eigenvalues	Most associated states
Model 2	0.49866	0.035223	$-0.11043 \pm 3.1331j$	ω_{18}, δ_{18}
	0.73218	0.031801	$-0.14637 \pm 4.6004j$	ω_6, δ_6
Model 3	0.49362	0.0019950	$-0.0061875 \pm 3.1015j$	ω_{18}, δ_{18}
	0.77442	0.0082036	$-0.039918 \pm 4.8658j$	ω_{20}, δ_{20}

selected form the ω_6,ω_{18} signals in the system which includes Model 2; and 5–200 s from the ω_{18}, ω_{20} and ω_6 signals in the system which includes Model 3 for FFT operation with 0.01 s time step.

Fig. 17 depicts the frequencies of rotor speed signals. There are two primary frequencies around 0.5 Hz and 0.7 Hz, and an inconspicuous frequency about 0.03 Hz for both ω_6 and ω_{18}. As the frequency of oscillations caused by the fault and controlled by hydro turbine and governor (here for short we can call it "turbine/governor dynamics") normally keep below 0.1 Hz, we can assert that the frequency 0.03 Hz in Fig. 17 belongs to the "turbine/governor dynamics". The other frequencies are due to system oscillations, which are extremely weak and disappear quickly, as shown in Fig. 16a. Similarly, Fig. 18 indicates the frequencies of system oscillations and "turbine/governor dynamics". When hydro turbine and governor models are used, the related "turbine/governor dynamics" as shown in Fig. 18 become more prominent, indicated by an increase energy of the related mode at about 0.05 Hz.

Even though the system with Model 2 behaves faster in response to the fault, we cannot assert Model 3 is bad, because Model 3 represent the real behavior and properties of hydro turbine and governors, as we discussed in Sections 4.1 and 4.2.

5.3. Small signal stability analysis

All eigenvalues for the KTH-NORDIC32 test system, either with Model 2 or Model 3, are located in the left half plane, which indicates that the systems are stable. The two lowest damping modes in the KTH-NORDIC32 system, as done for the SMIB system, are provided in Table 4. Moreover, which states are most associated with these two eigenvalues can be directly figured out from the "Eigenvalue Report" in the "Eigenvalue Analysis" routine from PSAT. Looking at these frequencies, they are just the system oscillation frequencies as computed in the FFT operation. This means the two lowest decay rate are just at the frequencies of system oscillations. It can therefore be asserted that the time-domain simulation results coincide with the small-signal stability analysis results, and thus, there is confidence of the implemented models for small-signal stability analysis.

6. Conclusions

This paper presented four developed hydro turbine and governor models and implemented them in PSAT—a free and open source software. Moreover, detailed implementation steps are also provided. To evaluate models features, the performances of the implemented models in two power systems—a single-machine infinite-bus (SMIB) system, and the KTH-NORDIC 32 system were examined. The evaluation of a power system's performance is concerned with the stability of that system by analyzing transient stability and small signal stability. Model validation demonstrates that the developed models have the capacity of representing the behaviors of hydro turbine and governors in reality. Future work will focus on tuning parameters of hydro turbine and governors models in order to reduce the system oscillations [24]. Additionally, models' performances in real time simulation environment [26] will be explored.

Acknowledgment

The authors gratefully acknowledge the aid of Prof. Federico Milano of the University of Castilla-La Mancha for his valuable comments and suggestions for implementing the new models described in this paper.

This Project was supported in part by the EIT InnoEnergy Collocation Center Sweden within the "Smart Power" Thematic Area, Work Package 3.1: Smart Transmission Grids Real-Time Simulation Platform.

Luigi Vanfretti is supported by the STandUP *for* Energy collaboration initiative and the KTH School of Electrical Engineering.

References

[1] International Energy Agency, Electricity/Heat in Sweden in 2008. <http://www.iea.org/stats/electricitydata.asp?COUNTRY_CODE=SE>.
[2] IEEE Committee Report, Dynamic models for steam and hydro turbines in power system studies, IEEE Transactions on Power Apparatus and Systems (1973) 1904–1915.
[3] J. Machowski, J. Bialek, J. Bumby, Power System Dynamics: Stability and Control, second ed., John Wiley & Sons, 1997.
[4] P. Kundur, Power System Stability and Control, McGraw-Hill, 1993.
[5] F. Milano, An open source power system analysis toolbox, IEEE Transactions on Power Systems 20 (2004) 1199–1206.
[6] F. Milano, M. Zhou, G. Hou, Open model for exchanging power system data, Power & Energy Society General Meeting, 2009. PES '09. IEEE, 2009, pp. 1–7.

[7] R.M. Stallman, Free Software, Free Society: Selected Essays of Richard M. Stallman, second ed., Free Software Foundation, Inc., 2010.
[8] D. Ramey, J. Skooglund, Detailed hydrogovernor representation for system stability studies, Power Apparatus and Systems, IEEE Transactions on PAS-89 (1970) 106–112.
[9] W. Li, Hydro Turbine and Governor Modeling and Scripting for Small-signal and Transient Stability Analysis of Power Systems, Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, September, 2011.
[10] I.C. Report, Hydraulic turbine and turbine control models for system dynamic studies, Power Systems, IEEE Transactions on 7 (1992) 167–179.
[11] L.N. Hannett, J.W. Feltes, B. Fardanesh, Field tests to validate hydro turbine-governor model structure and parameters, Power Systems, IEEE Transactions on 9 (1994) 1744–1751.
[12] T.V. Cutsem, Description, Modeling and Simulation Results of a Test System for Voltage Stability Analysis, Version 2, November, 2010.
[13] I. The MathWorks, SimPowerSystems 5.4, 2011. <http://www.mathworks.com/products/simpower/index.html?ref=productlink>.
[14] The MathWorks, Inc., Hydraulic Turbine and Governor. <http://www.mathworks.com/help/toolbox/physmod/powersys/ref/hydraulicturbineandgovernor.html>.
[15] F. Milano, Power System Analysis Toolbox Documentation for PSAT, Version 2.1.6, May, 2010.
[16] P.M. DeRusso, R.J. Roy, C.M. Close, A.A. Desrochers, State Variables for Engineers, second ed., John Wiley & Sons, 1997.
[17] F. Milano, L. Vanfretti, J. Morataya, An open source power system virtual laboratory: the PSAT case and experience, Education, IEEE Transactions on 51 (2008) 17–23.
[18] L. Vanfretti, F. Milano, Application of the PSAT, an open source software for educational and research purposes, in: Power Engineering Society General Meeting, 2007, IEEE, 2007.
[19] J.W. Eaton, GNU Octave Manual, Network Theory Ltd., Bristol, UK, 1997.
[20] M.H. Chaudhry, D. Arsenie, P. Vlase, Applied Hydraulic Transients, second ed., Taylor & Francis, 1987.
[21] B.B. Sharp, D.B. Sharp, Water Hammer: Practical Solutions, Butterworth-Heinemann, 1996.
[22] Y. Chompoobutrgool, L. Vanfretti, Linear Analysis of the *k*th-nordic32 System. Internal Report, Tech. Rep., KTH Royal Institute of Technology, March, 2011. <http://www.vanfretti.com/Dr._Luigi_Vanfrettis_Website/Welcome.html>.
[23] F. Schleif, G. Martin, R. Angell, Damping of system oscillations with a hydrogenerating unit, Power Apparatus and Systems, IEEE Transactions on PAS-86 (1967) 438–442.
[24] F. Schleif, A. Wilbor, The coordination of hydraulic turbine governors for power system operation, Power Apparatus and Systems, IEEE Transactions on PAS-85 (1966) 750–758.
[25] D.N. Kosterev, C.W. Taylor, Model validation for the august 10,1996 WSCC system outage, Power Systems, IEEE Transactions on 14 (1999) 967–979.
[26] C. Dufour, J. Belanger, A pc-based real-time parallel simulator of electric systems and drives, in: International Conference on Parallel Computing in Electrical Engineering, 2004. PARELEC 2004, pp. 105–113.