

A HIGH-SPEED TRANSIENT SIMULATOR FOR ON-LINE AND OFF-LINE ANALYSIS

G. H. Benson¹, R. W. Dunn¹

1. University of Bath, UK

ABSTRACT

Accurate assessment of transient and dynamic stability allows power systems to be operated closer to their stability limits, and the resulting gains in efficiency are beneficial to all.

Modern interconnected power systems form a highly complex, nonlinear problem space which resists analysis by traditional exact analytical methods, and it is widely regarded that time domain simulation is the only way to model and analyse the system at any more than a superficial level.

The University of Bath is well known for its time-domain power system simulators: the current version can simulate power systems such as the English and Welsh system (in excess of 200 machines and 1100 busbars) at 10ms resolution several times faster than real-time on a standard desktop computer.

This paper describes a new method for more accurately handling discontinuous nonlinearities such as limits by using multiple dynamic matrices, and proposes a technique to automatically optimise the integration algorithm for a given power system.

INTRODUCTION

The ability to simulate a power system's behaviour is a highly useful tool for stability analysis. Due to the computational load imposed by time-domain simulation, this has traditionally been done at the planning stage (*off-line* studies), however it is now feasible to use time-domain simulation in on-line stability assessment[1, 2].

The University of Bath has been developing power system simulation software since 1986. The current version of the simulator, the power system simulation engine (PSS-ENG), can simulate the full England and Wales power system (approximately 200 generators each between 7th and 30th order, 1100 busbars and 1800 lines) at about 3 times faster than real time on a single 650MHz PC.

To put this into an engineering perspective, the power system outlined above has roughly 3000 contingencies which must be analysed to assess its stability. For these to be assessed on a 15 minute timescale, a cluster of roughly 24 650MHz 'processing nodes', basically stripped down PCs, would be required at a cost of about £9000 (as of June 2000) for the hardware and operating system. Compared with the cost of even the most modest item of transmission system plant, such an on-line stability assessment system is not only feasible but ridiculously cheap!

The ability to quantify the stability of a power system within 15 minutes is incredibly useful for power system operators. Day-ahead planning can only be a forecast of what will really happen: such an on-line stability assessor can either confirm that the forecast was correct or, more importantly, highlight potential stability problems early enough for them to be rectified.

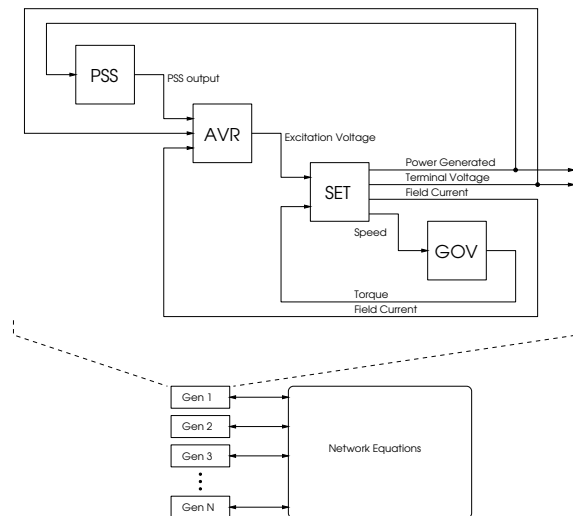


Figure 1: The overall hierarchy within PSS-ENG.

High-speed simulation need not be confined to on-line applications: a fast simulator could be employed as part of a planning system where its speed allows greater flexibility in the methods of analysis used. Applications have been developed to use PSS-ENG to optimise a power system for stability and economy[3], which is only possible due to the speed and accuracy which PSS-ENG offers.

Overview of PSS-ENG

To simplify the problem of simulating power systems, it is traditional to separate the model of the transmission system and the models of the generators. Each generator

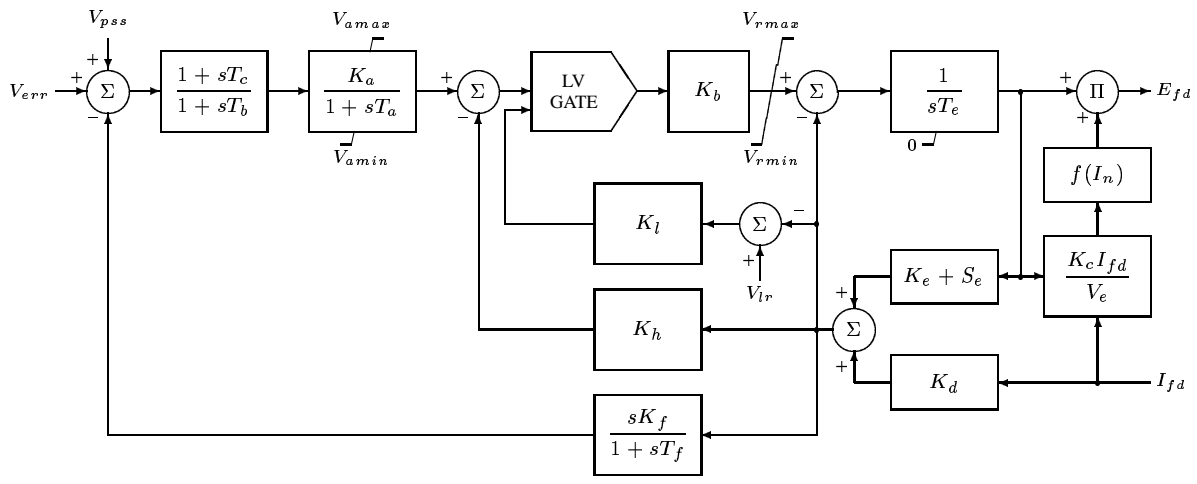


Figure 2: A typical AVR model: the IEEE AC2 standard AVR model[4].

is treated as an individual system whose inputs and outputs link it to the network and from there on to the other generators.

A simulation, therefore, consists of N generators and the network of transmission lines that connect them together, as shown in figure 1. The network is in principle the easier of the two to model, since it is described by simple linear algebra. The parameters which describe the network are formed into an *admittance matrix* \mathbf{Y} and the busbar voltages may be found by solving the vector equation $\underline{V} = \mathbf{Y}^{-1}\underline{I}$, where \underline{V} is the vector of voltages and \underline{I} is the vector of currents. Although simple to describe, in practice this is a non-trivial equation to solve, since the dimension of the matrix \mathbf{Y} is the number of busbars in the network: 1100 for the England and Wales system described above.

The generators are more complex to model since they are high-order systems, often with many nonlinearities. The core of a generator model is the *machine set*: in PSS-SENG, this is modelled using a 6th order *voltage behind subtransient reactance* model, based on work by Pal[5], Dale[6] and Berry[7]. Each machine set is accompanied by a number of items of control equipment, such as a speed governor to regulate the frequency, an automatic voltage regulator (AVR) to regulate the terminal voltage and a power system stabiliser (PSS) to suppress oscillations within the transmission network.

The authors have developed a method to isolate sections of complex simulations and analyse them in detail[8]. This has shown that the principal cause of inaccuracy and speed loss occurs within the simulation of the generation sets, which is therefore the focus for the work described in this paper.

This paper presents a series of modifications to PSS-SENG to improve both its speed and accuracy when handling discontinuous nonlinearities, to implement the facility for user-definable models and to facilitate further model optimisation.

GENERATOR SIMULATION

When broken down into their simplest form, generator models consist of nothing but integrators, gains and nonlinearities – figure 2 shows a typical AVR to illustrate the level of complexity being modelled. The outputs of the integrators are known as *integrable states*, \underline{x} , and the outputs of nonlinearities and inputs from the network are known as *non-integrable states*, \underline{u} . The entire generator model can then be expressed as:

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \quad (1)$$

$$\underline{y} = \mathbf{C}\underline{x} + \mathbf{D}\underline{u} \quad (2)$$

$$\underline{u} = f(\underline{x}, \underline{u}) \quad (3)$$

where $\dot{\underline{x}}$ is the derivative of \underline{x} , \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} are matrices which describe the connectivity within the model and $f()$ is a nonlinear function which describes the nonlinearities. Equation 2, which defines the outputs \underline{y} from the model, is relatively unimportant for the purposes of this paper, but has been included for completeness.

The state of a generator model is solely defined by the values of the integrable states and the values of its inputs from the network; all other values can be calculated from these.

To move from the current timestep (\underline{x}_k) to the next (\underline{x}_{k+1}), an *integration algorithm* is required. The original integration algorithm was invented by Euler in the mid 18th century to enable accurate navigation tables to be calculated. For each timestep, values were repeatedly passed through Euler's algorithm until the solution converged.

Algorithms which iterate in this manner are known as indirect, or *explicit*, algorithms; well-known examples of this genre include *Runge-Kutta* and of course Euler. Explicit algorithms can be iterated as many times as required to achieve the required level of accuracy, and they are ideally suited for *variable steplength* applications: if the re-

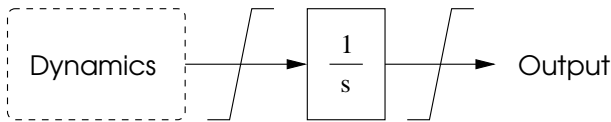


Figure 3: Simplified version of model susceptible to dynamic leakage.

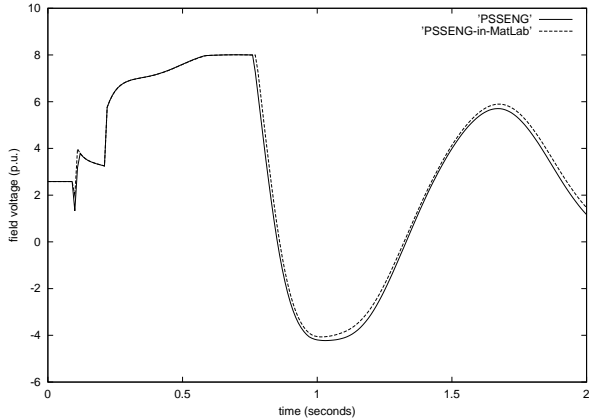


Figure 4: An artefact caused by dynamic leakage.

sult does not converge, then lower the length of time you are trying to advance and start again. Their main disadvantage is that the repetition of calculations often requires a lot of computing power.

The other main grouping of integration algorithms are the direct, or *implicit*, algorithms. These solve without iteration, moving directly from one timestep to the next. The algorithm must be well suited to the application to ensure accuracy, and this is of great importance when simulating generators since these tend to be *stiff* systems, where the ratio of the largest time constant over the smallest is large. When correctly employed, the accuracy of an implicit algorithm can match that of explicit algorithms and require much less computing power.

PSSENG uses a modified implicit algorithm to achieve its speed: the loss in integration accuracy is small when compared with the considerable errors introduced by assumptions and approximations in the models used for transient simulation. What is more important is that the error is not propagated and multiplied from step to step so that it swamps the real response. For this and other reasons, a modified version of the implicit trapezoidal method[9] is used. The modification, by Dale[6], refers to an outer loop which iterates the integration algorithm until the results are consistent; it is a testament to the algorithm's stability, however, that iteration is rarely necessary – consistent results are almost always achieved on the first pass.

Handling limits

The authors' investigation of the causes of inaccuracy (and hence speed loss) within PSSENG have revealed that the principal cause for concern lies in the handling of discontinuous nonlinearities, such as limits and the LV (lowest-value) gate in figure 2. The difficulty arises because of the implicit integration algorithm used: the integrable states advance in time simultaneously, which complicates the situation when there is a limit between two integrators.

As an example, take the V_r limit in figure 2. The simplest way of implementing this would be to calculate the input to the limit, apply the limit if necessary, and then inject the resultant value as a non-integrable state. This has the unfortunate effect of 'decoupling' the dynamics of the model, effectively breaking the model in half at this point, which introduces errors when the limit is not in force.

Another way to implement this would be to calculate the input to the limit and add to the signal a non-integrable variable which modifies the signal such that it is limited. This allows the dynamics to pass through the model uninterrupted when not limiting, but does not decouple them when the limit is in force. This allows small amounts of signal to 'leak' through the limit during steps which ultimately builds up errors on other integrators.

The effects of this leakage can be seen in figure 4; the system being simulated is shown in simplified form in figure 3. Small amounts of signal have leaked through the first limit and been stored on the integrator. The result is that the signal came out of the second limit too early and retained a slight phase and amplitude shift for the duration of the simulation.

An improved method for handling limits should encompass the best features of these two methods: the dynamics should be decoupled whilst limiting, but otherwise allowed to pass.

Matrix switching

It was observed that when a limit is not in force the limit is effectively absent; sections further on in the model are unaffected by it and act as though it did not exist. When the limit is crossed, however, the system's dynamics change abruptly, isolating the section before the limit and supplying a constant signal to the section after the limit. In this condition, the sections after the limit see only a constant signal; sections before the limit can have no effect on sections after the limit.

Remember that the system's dynamics are solely described by the four matrices **A**, **B**, **C** and **D** in equations 1 and 2. Instead of using one set of matrices to describe them, a *pair* of matrices can be used: one containing a unity gain and one containing a signal injection which is set to the value of whichever limit, upper or lower, is in

force.

This method can be extended to other discontinuous nonlinearities – the LV gate, for example, can be modelled with a pair of matrices each representing a connection from one of the inputs to the output.

The disadvantage of this method lies in its memory burden – a model with one limit requires two sets of matrices, whereas a model with two limits requires *four* sets of matrices, and so on. In the authors' experience, the most complex models have up to 16 such nonlinearities, requiring 65536 sets of matrices, which should not pose any significant problems given the large amounts of memory that today's PCs possess.

This multiple matrix method, therefore, should model the limit far more accurately than either of the two other methods and without either of their drawbacks.

Implementation and optimisation

PSSGEN currently has a number of built in models for machine sets, governors, AVRs and PSSs; the exact models and their parameters used for a particular generator are defined by the user. Internally, the different models are implemented individually, which would make the conversion to the multiple matrix method tedious and time consuming. Work is currently in progress to allow models to be defined by the user, ultimately using a graphical user interface with which block diagrams may be drawn. It is expected that the change to user definable models will not reduce the high simulation speed that PSSGEN currently enjoys; indeed, it seems likely that the speed may even be increased.

Apart from the obvious benefits to the user that this will bring, the internal structure of the simulator will be much more generic, allowing the multiple matrix method to be implemented simply and robustly. Furthermore, this internal standardisation will make it easy to test and implement various types of automatic model optimisation, both for speed and accuracy. As an example of the kind of optimisation envisaged, consider the following:

When the model is stable at a given timestep length, using shorter steplengths will take longer to simulate but yield no increase in accuracy. However, lengthening the timestep length can cause numerical instability to occur: in essence the sampling rate being used is too low to express the system dynamics correctly, and aliasing occurs. Numerical instability causes PSSGEN to reiterate over particular timesteps until the required level of accuracy is reached. Thus, increasing the steplength does not yield the expected speed increase.

Clearly, therefore, there is a range of timestep lengths within which the simulator is performing optimally. If this range could be identified on a per-model basis, an optimal timestep length could be found that keeps as many generators as possible within their optimal range. Thus,

the simulator would be running at the best possible steplength, for speed and accuracy, for the particular power system being simulated.

RESULTING BENEFITS

The software modifications are presently nearing completion and initial comparisons with MATLAB are very promising. There are two main components to the modifications: reading the model files and building the matrices, and initialising the models to a specified steady state ready for simulation. At the time of writing, the matrix builder is complete but the model initialiser is about 50% complete, hence this paper will present predicted beneficial results.

- It is expected that the user-definable models will not affect the accuracy of the simulator; a moderate increase in speed is likely due to a simpler simulator core.
- It is expected that the multiple matrix method will remove artifacts such as the one shown in figure 4, increasing accuracy without loss of speed.

CONCLUSION

It is widely regarded that time domain simulation is the only way to model and analyse modern interconnected power systems at any more than a superficial level. The computational load of such analysis is high, and stable and accurate implicit integration algorithms are an excellent way to lower this burden.

The multiple matrix method outlined in this paper will bring an increase in accuracy when simulating discontinuous nonlinearities such as limits using an implicit integration algorithm. It is also possible that the method will have advantages in applications using explicit algorithms.

The user-definable models implemented in PSSGEN in order to implement multiple matrices will herald an enormous increase in flexibility, and the internal standardisation will allow automatic analysis of the models prior to their use in simulation.

Work is now in progress to automate the identification of the onset of numerical instability within models, a task that is easily accomplished by the human eye but far more complex to carry out using a computer. This will allow the steplength optimisation outlined above to be implemented.

ACKNOWLEDGEMENTS

The authors are pleased to acknowledge the continuing support they have received from The National Grid Company for this and other research studies.

REFERENCES

- [1] A.R. Edwards, K.W. Chan, R.W. Dunn, and A.R. Daniels. Interactive online dynamic security assessment of large complex power systems part 2: Contingency screening. *Proceedings of IEEE Power Engineering Society*, Winter 1996.
- [2] M.E. Bradley, A.O. Ekwue, F. Li, K.W. Chan, R.W. Dunn, and A.R. Daniels. On-line stability analysis for an operational tripping scheme monitor. *Proceedings of IEE Power System Control and Management*, IEE-421:71–75, April 1996.
- [3] B.A. Nicholson, R.W. Dunn, K.W. Chan, and A.R. Daniels. A new technique for applying genetic algorithms to power system security optimisation problems. *Proceedings of IEEE Energy Management and Power Delivery*, supp:22–27, March 1998.
- [4] IEEE Committee Report. Excitation system models for power system stability studies. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(2):494–509, February 1981.
- [5] M.K. Pal. *Mathematical methods in power system stability studies*. PhD thesis, University of Aston, 1971.
- [6] L.A. Dale. *Real-time modelling of multimachine power systems*. PhD thesis, University of Bath, 1986.
- [7] T. Berry, L.A. Dale, A.R. Daniels, and R.W. Dunn. Real-time modelling of multimachine power systems. *IEE Proceedings*, 140:241–248, 1993.
- [8] G.H. Benson and R.W. Dunn. Compartmentalised validation of models for complex nonlinear dynamic simulators using a trusted third party generic simulator. *Proceedings of IASTED MS 2000*, pages 148–152, 2000.
- [9] H.W. Dommel and N. Sato. Fast transient stability solutions. *IEEE Transactions on Power Apparatus and Systems*, 91:1643–1650, 1972.

AUTHOR'S ADDRESS

The first author can be contacted at:

Power Systems Modelling and Control
Department of Electronic and Electrical Engineering
University of Bath
Claverton Down
Bath BA2 7AY, UK

email: psmac@ee.bath.ac.uk