

A New Tool for Visualization and Animation of Power Component and System Operation

A. P. Sakis Meliopoulos
 School of Electrical and Computer
 Engineering
 Georgia Institute of Technology
 Atlanta, Georgia 30332
sakis.meliopoulos@ece.gatech.edu

George Cokkinides, Ben Beker, Roger Dougal
 Department of Electrical and Computer
 Engineering
 University of South Carolina
 Columbia, SC 29208
cokk@engr.sc.edu

Abstract

A new tool for computing and visualizing the operation of power system components and systems is presented. One component of the tool is a time domain simulator that operates in the background within a multitasking environment. Visualization and animation objects can retrieve data in real time from the time domain simulator and display a detailed picture of a component or system operation. As the system state evolves, the pictures are updated resulting in an animated visualization of the process. The tool has been implemented in a multitasking environment, thus permitting user interaction in real time with the system parameters and immediate reflection of the results in animations and visualizations. Several examples of visualization and animation are presented, including animation of Mho relay operation, visualization of small signal stability, load dynamics and voltage sensitivity and synchronous machine operation.

1. Introduction

Recent advances in software engineering have made it possible to develop dynamic system simulators that operate in a multitasking environment. The additional graphical user interface tools and hardware accelerated graphics algorithms make the final product an indispensable tool to the understanding of the operation of the system. Many products have been developed along these lines for power system engineering. Most of these are focused on simulating the system under sinusoidal steady state operation. In the past few years, we have undertaken an effort to develop a time domain simulator of systems in a multitasking environment. In this environment, visualization and animation objects have access to the instantaneous conditions of the components and the overall system. Thus the visualizer/ animator can generate detailed 3D pictures/movies of the instant by instant operation of the component or the system.

The paper provides a brief overview of the new tool, the mathematical formulation of the simulator and the

data flow between the time domain simulator and the visualizer/animator. The paper focuses on a number of important power system operational issues. For these problems, visualization and animation tools are presented.

2. Description of the Virtual Test Bed

Part of the internal structure of the Virtual Test Bed is illustrated in Figure 1. This architecture was developed with consideration for the minimal representation of system components and the requirement to perform various tasks of the Virtual Test Bed on multiple processors. In the background of the Virtual Test Bed is the network solver that is a time domain simulation program. The network solver is based on the representation of each system component with its algebraic companion form (ACF) [1].

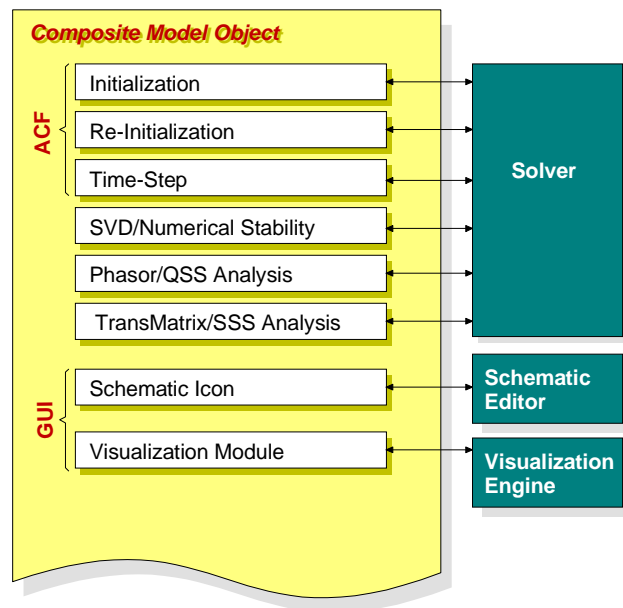


Figure 1. The Virtual Test Bed Architecture

The ACF is developed from the integro-differential equations of a component by numerical integration. Other component models (for example from SPICE, ACSL, etc.) can be accepted via model translators. The ACFs of all components in a system are related via the connectivity constraints. Application of the connectivity constraints yields a quadratic network equation that is solved by the network solver.

The network solver continuously executes, providing the simultaneous solution of the entire system and determining the state of each component of the system. This information is made available to animation and visualization components via a data backplane. The Virtual Test Bed has been developed in a multitasking environment, thus allowing parameter changes and immediate system response observations.

The paper discusses a number of power system visualization and animation examples. Specific examples are:

- Synchronous machine dynamics and internal fluxes.
- Distance and Mho relay operation.
- Visualization of voltage stability margins.
- Small signal stability visualization.

Here we discuss a couple of visualization and animation examples.

3. Animation and Visualization Example 1: Synchronous Generator Operation

Figure 2 illustrates one visualization of the operation of a synchronous generator. The simulation engine provides continuously the state of the generator at discrete time steps, i.e. $\dots t-2h, t-h, t$. At the present time t , the state of the generator is known, $x(t)$. From the state one can compute all the currents and voltages of the generator model. The magnetic fluxes of the generator can be also computed as well as the electromagnetic torque. Figure 2 is a snapshot of this information.

Note that the magnetic fluxes due to various currents in the generator can be reproduced as a function of the location in the air gap. The flux due to the armature current is displayed in the first window. The flux produced by the damper winding D is displayed in the second window. The flux produced by the damper winding Q is displayed in the third window. Finally the torque generated by the interaction of the armature and field currents is displayed in the fourth window together with the torque due to the damper windings D and Q and the position of the direct axis of the generator. Figure 2 is a snapshot. In an actual simulation, Figure 2 evolves as the state of the system evolves. For example under normal operating conditions, the armature flux rotates with synchronous speed. The fluxes due to damper

winding currents are zero and the armature-field torque is constant while the damper winding torque is zero. In case of a fault anywhere in the system, the picture changes and the user can observe the evolution of the magnetic fields and the electromagnetic torque.

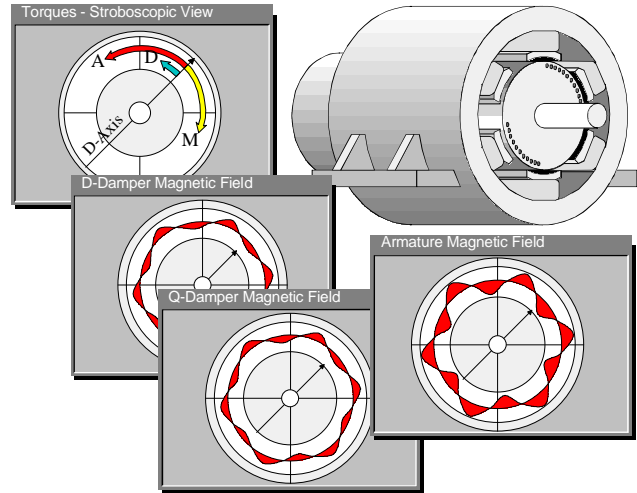


Figure 2. Visualization and Animation of the Operation of a Synchronous Generator

4. Animation and Visualization Example 2: Mho Relay Operation

Figure 3 illustrates another visualization and animation example. This application is concerned with the visualization of the so-called modified mho relay. The operation of this relay is based on the apparent impedance that the relay 'sees' and the trajectory of this impedance.

The visualization associated with this relay illustrates what the relay 'sees' during a disturbance in the system. Note that the disturbance may be at a location far away from the relay as it is illustrated in Figure 3. The relay monitors the three phase voltages and currents at the point of its application. The animation model retrieves the information that the relay monitors from the simulator at each time step. Subsequently, it computes the phasors of the voltages and currents. Part of the visualization displays these phasors (see left part of the visualizer). From this information, the positive sequence voltage and current are constructed and displayed. The apparent impedance is also computed and displayed. As the system state evolves so do the pictures of the visualization providing a pictorial view of the operation of this relay. This animation of the impedance trajectory is superimposed on the relay characteristics and settings.

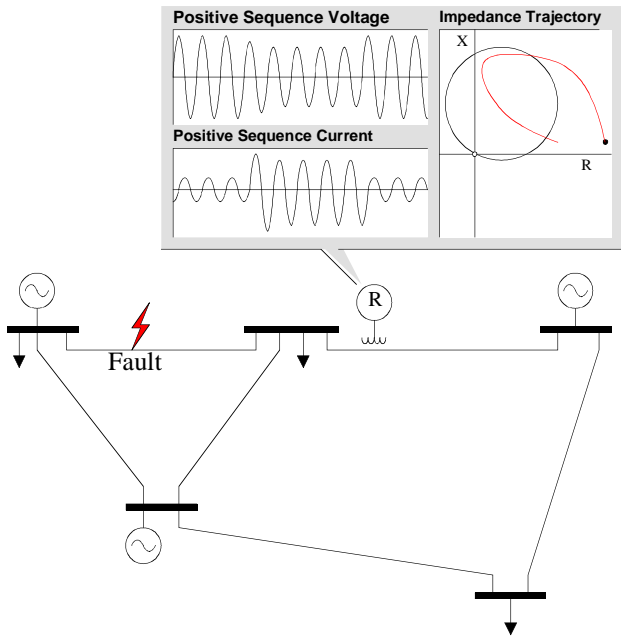


Figure 3. Visualization and Animation of the Operation of a Modified Mho Relay

5. Animation and Visualization Example 3: Induction Motor Animation

Figure 4 illustrates an induction motor simulation example. The simulated system consists of a 15MVA 13.8 kV three phase induction motor fed by a three phase source via a 100 meter cable. In addition to the speed, torque and voltage versus time plots, two animation objects are included which show different aspects of the motor operation.

The first animation object (top right window of Figure 4) shows the location of the rotor and the air gap magnetic field (red area). The window is updated at every step of the simulation, thus showing the relative rotation speeds of the rotor and magnetic field, as the rotor accelerates. Also the changing magnetic field magnitude can be observed during the motor acceleration. Furthermore, effects of motor current transients and phase asymmetries on the air-gap magnetic field can be visualized.

The second animation object (bottom-left window) illustrates the motor dynamics using a dynamically updated torque-speed plot. This plot includes the electrical torque (red trace) the motor mechanical load torque (blue trace) and a cursor (vertical white line) that indicates the present operating point of the motor. To the left of the torque-speed plot, the RMS values of the motor terminal voltages and currents are shown using bar graphs. Note that the electrical torque speed curve of the motor is a function of the RMS voltage at the motor

terminals. During the motor start-up, the motor draws a large current (5.5 pu in this example) which causes the motor terminal voltage to drop below the nominal value (0.7 pu in this example). This animation object dynamically updates the motor electrical torque-speed curve as it is affected by the changing motor terminal voltage, as well as the motor operating point cursor. The motor electrical torque-speed curve is computed as follows:

During the simulation, the RMS values of the phase voltages are continuously calculated over a sliding time window using a recursive Fourier transform method. The sliding time window width is user-selected (typically one power frequency cycle (16.66 ms)) and ends on the latest calculated waveform sample. Using the RMS values of the phase voltages, the motor electrical torque versus speed is computed as follows:

$$T = \frac{pR_r(V_a + V_b + V_c)}{2\omega(1-s)\left(\left(\frac{R_r + R_s}{1-s}\right)^2 + (X_r + X_s)^2\right)}$$

where: s is the motor speed in pu., R_r, X_r are the rotor resistance and reactance R_s, X_s are the stator resistance and reactance p is the number of poles ω is the power frequency.

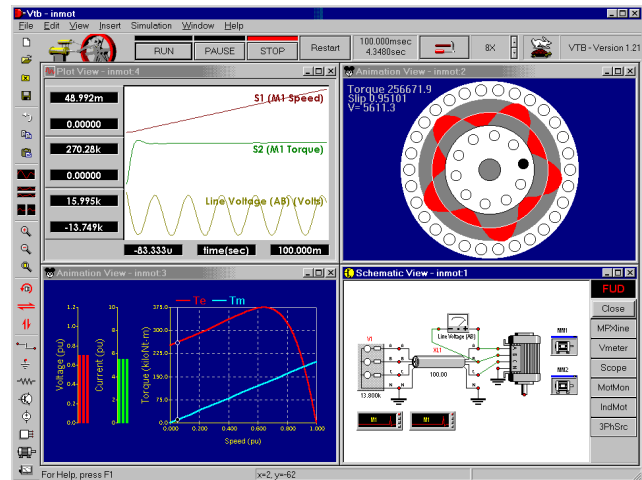


Figure 4. Animation of Induction Motor Operation

The vertical distance between the intersections of the operating point cursor with the electrical and mechanical load torque curves is equal to the accelerating torque. When the system reaches steady state, the operating point cursor overlaps with the intersection between the electrical and mechanical torque curves. Furthermore, the relative shape of the electrical and mechanical torque curves determines the stability characteristics of the

system. Thus this animation object illustrates the voltage sensitivity of systems containing induction motor loads.

6. Animation and Visualization Example 4: System Stability Tracking

An important visualization function is the animation of the system small signal stability properties. This is achieved by providing displays of the eigenvalue map in real time. Specifically, the VTB computes the system transition matrix over a user specified time interval. Subsequently, the eigenvalues of the transition matrix are computed and displayed. The process is illustrated in Figure 5. A concise description of the computational algorithm for the transition matrix is presented in the Appendix.

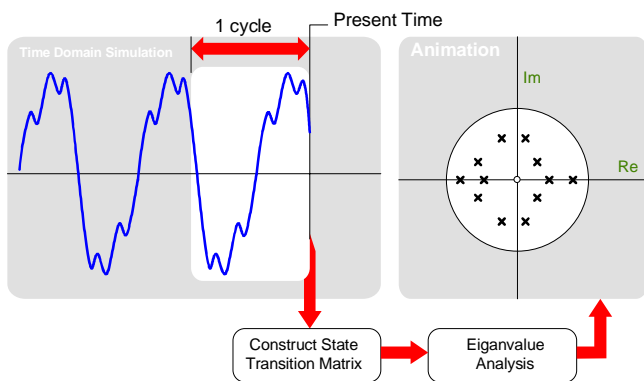


Figure 5. Illustration of Small Signal Stability Tracking

The proposed method has been applied to a power system with generators, loads and motors. A program-generated screen is illustrated in Figure 6. The simulated system is displayed in the lower left window in single line diagram form. It contains four three phase buses labeled 10 20, 30 and 40, connected via three phase transmission lines. Three phase generators are connected at busses 10, 30 and 40. Y-connected balanced loads are connected at busses 10 and 20. The schematic diagram also contains three voltmeters and three current meters, as well as the stability animation object symbol (at the right of the schematic window). The meters monitor the voltage and current at bus 20.

The waveforms captured by the meters are displayed in the top left window of the screen image illustrated in Figure 6. The y-axis units are in kilovolts and kiloamperes. The computed system eigenvalues are displayed in the top right window. Note that all eigenvalues are within the unit-circle indicating a stable system. Both waveform and eigenvalue displays are continuously updated during the simulation, always indicating the present state of the system. For linear

systems the eigenvalue map remains unchanged. For nonlinear systems the eigenvalue map changes with the operating point of the system providing instant feedback of the stability of the system

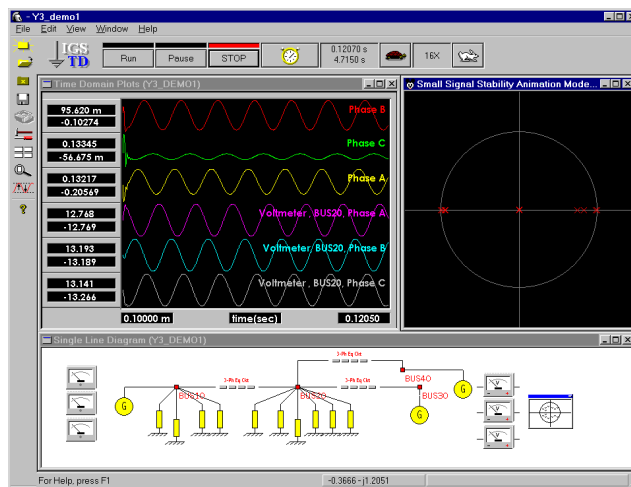


Figure 6. An Example Power System with Loads and Motors

7. Conclusions

A new tool for power system animation and visualization has been presented. The tool is based on a time domain solver that continuously ‘runs’ in the background. The animation/visualization engine retrieves the state of the system or components and appropriately displays the information in a useful form. Several examples of the capabilities of this new tool have been presented. The presentation of the paper will include live demonstration of this powerful tool.

8. Acknowledgments

The work reported in this paper has been partially supported by the ONR Grant No. N00014-96-1-0926. This support is gratefully acknowledged.

9. References

[1] A. P. Sakis Meliopoulos and G. J. Cokkinides, ‘A Time Domain Model for Flicker Analysis’, *Proceedings of the IPST '97*, pp. 365-368, Seattle, WA, June 1997.

[2] Eugene V. Solodovnik, George J. Cokkinides and A. P. Sakis Meliopoulos, ‘Comparison of Implicit and Explicit Integration Techniques on the Non-Ideal Transformer Example’, *Proceedings of the Thirtieth Southeastern Symposium on System Theory*, pp. 32-37, West Virginia, March 1998

[3] Eugene V. Solodovnik, George J. Cokkinides and A. P. Sakis Meliopoulos, "On Stability of Implicit Numerical Methods in Nonlinear Dynamical Systems Simulation", *Proceedings of the Thirtieth Southeastern Symposium on System Theory*, pp. 27-31, West Virginia, March 1998.

[4] Beides, H., Meliopoulos, A. P. and Zhang, F. "Modeling and Analysis of Power System Under Periodic Steady State Controls", *IEEE 35th Midwest Symposium on Circuit and Systems*

[5] A. P. Sakis Meliopoulos, *Power System Grounding and Transients*, Marcel Dekker, Inc., 1988.

[6] A. P. Sakis Meliopoulos, G. J. Cokkinides and A. G. Bakirtzis, "Load-Frequency Control Service in a Deregulated Environment", *Decision Support Systems*, Vol. 24, No. 3-4, pp. 243-250, January 1999.

[7] A. P. Sakis Meliopoulos, Murad Asad and George J. Cokkinides, 'Issues of Reactive Power and Voltage Control Pricing in a Deregulated Environment', *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, p. 113 (pp. 1-7), Wailea, Maui, Hawaii, January 5-8, 1999.

[8] Ben Beker, George J. Cokkinides, Roger Dugal and A. P. Sakis Meliopoulos, 'The Virtual Test Bed for PEBB Based Systems', *Proceedings of the 3rd International Conference on Digital Power System Simulators*, Vasteras, Sweden, May 25-28, 1999.

[9] A. P. Sakis Meliopoulos, David Taylor, George J. Cokkinides and Ben Beker 'Small Signal Stability Analysis in PEBB Based Systems', *Proceedings of the 3rd International Conference on Digital Power System Simulators*, Vasteras, Sweden, May 25-28, 1999.

[10] P. Sakis Meliopoulos and George J. Cokkinides 'Small Signal Stability Analysis in PEBB Driven Motion Systems', *Proceedings of the ELECTROMOTION '99 Symposium*, pp. 273-278, Patras, Greece, July 8-9, 1999

10. Appendix: Calculation of System Transition Matrix

This appendix discusses the algorithm for the computation of the system transition matrix within the Virtual Test Bed. We present first the one time-step transition matrix and then the transition matrix over a user defined time interval.

10.1 One-Time Step Transition Matrix

In this section we consider the problem of calculating the system transition matrix at time $(t+nh+h)$ for the time interval $(t+nh, t+nh+h)$, i.e. one time step interval. This transition matrix is computed by manipulating the algebraic equations of all devices in the system in such a way that we obtain a relationship between the state of the

system at time $(t+nh+h)$ and the state of the system at time $(t+nh)$. Note that at time $t+nh+h$ of the simulation process, the algebraic companion form for all devices in the system are known. The quadratic terms are neglected. Then the general expression of the linearized equation for device i , is:

$$I_i(t+nh+h) = A_{i,t+nh+h} x_i(t+nh+h) + B_{i,t+nh+h} x_i(t+nh) + C_{i,t+nh+h} I_i(t+nh)$$

where x is the state (including external and internal states). Application of interface constraints (Kirchoff's current law), will result in eliminating the terminal currents at time $(t+nh+h)$, resulting in the system wide equation:

$$0 = \sum_{i=1}^N E_i^T A_{i,t+nh+h} E_i x(t+nh+h) + \sum_{i=1}^N E_i^T B_{i,t+nh+h} E_i x(t+nh) + \sum_{i=1}^N E_i^T C_{i,t+nh+h} I_i(t+nh)$$

Note that above equation is of the form:

$$0 = A_{s,t+nh+h} x(t+nh+h) + B_{s,t+nh+h} x(t+nh) + \sum_{i=1}^N E_i^T C_{i,t+nh+h} I_i(t+nh)$$

where:

$$A_{s,t+nh+h} = \sum_{i=1}^N E_i^T A_{i,t+nh+h} E_i, \text{ and}$$

$$B_{s,t+nh+h} = \sum_{i=1}^N E_i^T B_{i,t+nh+h} E_i$$

By solving for the system state at time $t+nh+h$, we recognize that the transition matrix is:

$$\Phi(t+nh+h, t+nh) = -A_{s,t+nh+h}^{-1} B_{s,t+nh+h}$$

Note that the transition matrix is computed as a byproduct of the simulation process. It is also computed for the prevailing operating conditions.

10.2 N-Time Step Transition Matrix

The computation of the n-time step transition matrix follows the same general procedure as for the one-time

step transition matrix. The idea is again the same: manipulate the algebraic companion equations in such a way as to obtain a relationship between the system state at time (t+nh+h) and the system state at time (t+h). These manipulations are tedious and require a substantial computational effort. To limit the computational effort, certain information can be stored and used in a recursive algorithm. This algorithm is presented here.

Assume that the following transition matrices have been computed and stored:

$$\begin{aligned} x(t+h) &= \Phi(t+h,t)x(t) + B_1(t) \\ x(t+2h) &= \Phi(t+2h,t)x(t) + B_2(t) \\ &\dots\dots\dots \\ x(t+nh) &= \Phi(t+nh,t)x(t) + B_n(t) \end{aligned} \tag{A-1}$$

At the same time assume that the linearized algebraic companion form equations for each device for the last n steps have been stored:

$$\begin{aligned} I_i(t+h) &= A_{i,t+h}x_i(t+h) + B_{i,t+h}x_i(t) + C_{i,t+h}I_i(t) \\ I_i(t+2h) &= A_{i,t+2h}x_i(t+2h) + B_{i,t+2h}x_i(t+h) \\ &\quad + C_{i,t+2h}I_i(t+h) \\ &\dots\dots\dots \\ I_i(t+nh) &= A_{i,t+nh}x_i(t+nh) + B_{i,t+nh}x_i(t+nh-h) + \\ &\quad + C_{i,t+nh}I_i(t+nh-h) \end{aligned} \tag{A-2}$$

It is important to realize that as the simulation progresses, above information is known. We simply store the information and then use it for the computation of the transition matrix. The algorithm consists of the following steps:

Step 1: Form matrix A_s using the following constructor:

$$A_s \leftarrow A_s + E_i^T A_{i,t+nh+h} E_i$$

Step 2: Form the matrix B_s using the following constructor:

$$\begin{aligned} B_s &\leftarrow B_s + E_i^T (B_{i,t+nh+h} + C_{i,t+nh+h} A_{i,t+nh}) E_i \\ &\Phi(t+nh, t+h) \\ B_s &\leftarrow B_s + E_i^T C_{i,t+nh+h} (B_{i,t+nh} + C_{i,t+nh} A_{i,t+nh-h}) E_i \\ &\Phi(t+nh-h, t+h) \\ &\dots\dots\dots \\ B_s &\leftarrow B_s + E_i^T C_{i,t+nh+h} C_{i,t+nh} \dots \\ C_{i,t+4h} &(B_{i,t+3h} + C_{i,t+3h} A_{i,t+2h}) E_i \Phi(t+2h, t+h) \end{aligned}$$

Step 3: Compute the transition matrix:

$$\Phi(t+nh+h, t+h) = -A_s^{-1} B_s$$

Step 4: Perform eigenvalue analysis of the transition matrix and display results:

Step 5: Update equations (A-1):

$$\begin{aligned} x(t+2h) &= \Phi(t+2h,t)\Phi(t,t+h)x(t+h) + B_2(t+h) \\ x(t+3h) &= \Phi(t+3h,t)\Phi(t,t+h)x(t+h) + B_3(t+h) \\ &\dots\dots\dots \\ x(t+nh) &= \Phi(t+nh,t)\Phi(t,t+h)x(t+h) + B_n(t+h) \\ &\text{-----} \\ x(t+nh+h) &= \Phi(t+nh+h,t+h)x(t+h) + B_{n+1}(t+h) \end{aligned}$$

Step 6: Update equations (A-2) and proceed to the next time step (t+nh+2h). The updating of equations (A-2) involves the computation of the algebraic companion forms (part of the simulator) and storing the linear part.

It should be apparent that the above algorithm provides the n-time step transition matrix at time (t+nh+h). The computational effort is minimized at a price: equations (A-1) and (A-2) must be stored, this is indeed a large amount of data. The usefulness of this analysis is great. Since eigenvalues are computed at each time step and are displayed in real time, the stability properties of the system are immediately understood. One can observe whether all eigenvalues remain within the unit circle (stable system), or any eigenvalue movement towards the unit circle is observed immediately. The proximity of an eigenvalue to the unit circle also provides a measure of the damping level in each mode of oscillation, etc.

10.3 Modeling of Controls

Existing control schemes can be included in the proposed method for small signal stability analysis. In PEBB based systems the controls are typically based on complex feedback laws. For example the equidistant control of six-pulse based systems, requires the evaluation of the positive sequence voltages that drive the inverter. The positive sequence voltages require the past history voltages of all three phases for at least one period of the fundamental frequency. This type of feedback control and other similar is incorporated into the proposed method as follows. A new state variable is introduced to denote the decision, x_u . This state variable is expressed as a function of the other state variables, i.e.

$$x_u = f_u(x_1, x_2, \dots)$$

The dependence of x_n on x_1, x_2 , etc. may be expressed as a Fourier transform over a time window, etc. This equation is added to the model equations and the method is applied as described in the previous sections. Note that these equations involve the state variables over a time window and therefore the transition matrix must be defined over a time period at least equal to the time window of the control equation.

11. Biographies

A. P. Sakis Meliopoulos (M '76, SM '83, F '93) was born in Katerini, Greece, in 1949. He received the M.E. and E.E. diploma from the National Technical University of Athens, Greece, in 1972; the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology in 1974 and 1976, respectively. In 1971, he worked for Western Electric in Atlanta, Georgia. In 1976, he joined the Faculty of Electrical Engineering, Georgia Institute of Technology, where he is presently a professor. He is active in teaching and research in the general areas of modeling, analysis, and control of power systems. He has made significant contributions to power system grounding, harmonics, and reliability assessment of power systems. He is the author of the books, *Power Systems Grounding and Transients*, Marcel Dekker, June 1988, *Ligthing and Overvoltage Protection*, Section 27, Standard Handbook for Electrical Engineers, McGraw Hill, 1993, and the monograph, *Numerical Solution Methods of Algebraic Equations*, EPRI monograph series. Dr. Meliopoulos is a member of the Hellenic Society of Professional Engineering and the Sigma Xi.

George Cokkinides (M '85) was born in Athens, Greece, in 1955. He obtained the B.S., M.S., and Ph.D. degrees at the Georgia Institute of Technology in 1978, 1980, and 1985, respectively. From 1983 to 1985, he was a research engineering at the Georgia Tech Research Institute. Since 1985, he has been with the University of South Carolina where he is presently an Associate Professor of Electrical Engineering. His research interests include power system modeling and simulation, power electronics applications, power system harmonics, and measurement instrumentation. Dr. Cokkinides is a member of the IEEE/PES and the Sigma Xi.

Benjamin Beker (M '83, SM '94) was born in Vilnius, Lithuania, in 1959. He obtained the BS, MS, and Ph.D. degrees at the University of Illinois at Chicago (UIC) in 1982, 1984, and 1988, respectively. From 1982 to 1988, he was a research assistant in the EECS department at UIC. Since 1988, he has been with the University of South Carolina where he is presently a professor of Electrical Engineering. His research interests include electromagnetic modeling and simulation, microwave integrated circuits, and electronic packaging.

Roger Dougal (M '82, SM '93) joined the faculty at the University of South Carolina in 1983 after earning the Ph.D. degree in electrical engineering at Texas Tech University. He is a professor of Electrical Engineering and is VTB project director. His research interests include physical electronics and modeling and simulation of physical devices. Dr. Dougal has received the *Samuel Litman Distinguished Professor of Engineering* award, and has been honored as a *Carolina Research Professor*.