



AN INTEGRATED PLOTTING TOOL FOR THE VISUALIZATION OF RESULTS PRODUCED BY MULTIPLE POWER SYSTEM ANALYSIS SOFTWARE

Alex de Castro*

Fundação Padre Leonel Franca

Paulo E. M. Quintão

Fundação Padre Leonel Franca

Sergio Gomes Junior

CEPEL

Nelson Martins

CEPEL

Herminio J.C.P. Pinto

Nexant/PCA, (previously with CEPEL)

SUMMARY

This paper describes an integrated platform for the visualization of results produced by various power system analysis software developed by CEPEL. This advanced plotting tool incorporates state-of-art programming techniques such as: object oriented programming (OOP), components development technology and reusability, message passing between applications, etc.

Keywords: graphical user interface, object oriented programming, components technology, power system analysis software.

1. INTRODUCTION

Since the early days of power system computations, there has been a pressing need to better visualize the simulated results. The proper visualization of results requires advanced graphics and tables that could not be obtained in the screens of the old-generation, character-based, user terminals. It was then only possible to obtain hardcopies of the graphics, whose curves were usually comprised of a sequence of alphanumeric characters. The printers had limited graphical capabilities and plotters were very expensive.

Personal computers became increasingly powerful and started being routinely used in electric utilities to run power system analysis software. The fast dissemination of personal computers with continuous advances in hardware and compilers was followed by the development of comprehensive and user-friendly power system analysis software, incorporating graphical user interface.

The Windows environment caused a revolution in man machine interface and a dramatic improvement in

graphics quality. Another important fact was the dissemination of object-oriented languages [1,2,3] among software developers. This ignited yet another revolution in the software development area, especially in subjects like code reusability. Note that code reusability also evolved into components reusability through the object oriented building block – classes.

Over the last two decades, CEPEL has been involved in the development of graphical user interfaces and tools for visualization of results. A current state-of-the-art tool for visualizing power system simulation results is Plot CEPEL.

Three power system analysis software currently utilize Plot CEPEL. PacDyn for small signal stability analysis [4], ANATEM for transient, mid-term stability simulations [5] and ANAREDE for network analysis when calculating continuation power flow [6,7,8,9]. HARMZS is a modern power system harmonic analysis software [26] that visualizes results in three alternative platforms, one of which is Plot CEPEL.

2. MODULARIZATION

Modern software is generally comprised of smaller pieces of software known as components. These components may have been written in different programming languages and are usually organized in Dynamic Linking Libraries (DLLs) [10,11] or ActiveX OLE Components (OCXs) [12,13,14]. The main advantage in this approach is that these components can be reutilized by many applications, through component reusability.

2.1 Components Architecture

Plot CEPEL was developed utilizing Microsoft Visual C++ 6.0 [2,11,15,16] and Microsoft Foundation Classes

* Fundação Padre Leonel Franca, Rio de Janeiro, RJ - Brazil – email: plotcepel@cepel.br

(MFC) [11,15,17]. MFC has many built-in classes to help the development of Windows-compatible applications through a shell over the Windows Application Programming Interface (API) [10]. Plot CEPEL also utilizes the object oriented programming paradigm [1,2,3] and components technology through ActiveX and Component Object Model (COM) [12,13,14].

A block diagram showing the main components of Plot CEPEL is depicted in Figure 1.

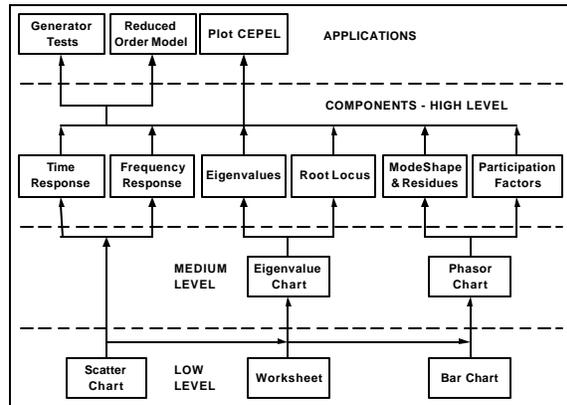


Figure 1 –Plot CEPEL Components Block Diagram.

Note that the components are organized in three levels according to their functionality and specialization. The high level components instantiate the medium level whereas the medium level components instantiate the low level components according to their needs. The arrows point in the direction of increasing specialization level: from low level to high level components. There are six high level, two medium level and three low level components.

Each high level component has a specific visualization task as described in Table 1.

Table 1 – High Level Components Functionality.

Component	Component Functionality
Time Response	Handling and visualization of linear time response simulation generated by PacDyn, time response simulation generated by ANATEM and ANAREDE continuation power flow.
Frequency Response	Handling and visualization of frequency response simulations generated by PacDyn.
Eigenvalues	Handling and visualization of system poles and transfer function zeros calculated by PacDyn.
Root Locus	Handling and visualization of the root locus and root contour simulations produced by PacDyn.

Mode Shapes & Residues	Handling and visualization of Mode Shapes, Residues, Observability and Controllability Factors calculated by PacDyn.
Participation Factors	Handling and visualization of the Participation Factors calculated by PacDyn.

The high level components inherit by default the functionalities of the medium level and low level components to which they are connected.

The low level components are, by design, the interface with all other external graphical components and controls. The low level components functionality is shown in Table 2.

Table 2 – Low Level Components Functionality.

Component	Component Functionality
Scatter Chart	Used by all high level components that display results utilizing scatter plots.
Worksheet	Used by all high level components that display results in tables.
Bar Chart	Used by all high level components that display results in bar charts.

Consider as an example, that the full system eigenvalues have been calculated by PacDyn. In order to visualize the eigensolution results, the Eigenvalues high level component instantiates the Eigenvalue Chart medium level component, which then instantiates the low level components Scatter Chart and Worksheet. These low level components allow visualizing the poles in the complex plane plot (Figure 9) or in a worksheet (Figure 10), respectively.

The Generator Tests and the Reduced Order Model applications, shown in the diagram of Figure 1, are examples of component reusability. They are completely different applications that also use the Time Response and Frequency Response high level components originally developed to be used in Plot CEPEL.

2.2 Data Flow among High Level Components Classes

Figure 2 shows the four classes that can be instantiated by any high level component in order to visualize a specific result. The Control Class is the manager class. The three other classes, File Class, Series Management Class and Plot Class are nested classes. The three nested classes are completely independent and unaware of the existence of each other while the Control Class manages any information exchanges among them. The data flow among these classes is described in the next paragraphs.

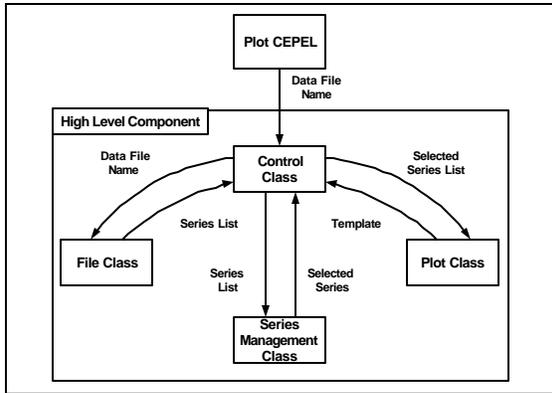


Figure 2 – High Level Components Data Flow.

Plot CEPEL receives the results file name from other application, through direct call or message passing, identifies the file format and instantiates the high level component associated with the file format and the file visualization needs.

Through a high level component instantiation, the Control Class receives from Plot CEPEL the name of the file to be opened. The received file name is then sent to File Class, responsible for reading the results file in the appropriate format and building a series list, which is then sent back to Control Class. The series list contains all those series read in the results file by the File Class.

The Control Class then sends the series list to Series Management Class. The Series Management Class is responsible for the series management through the Graphic Management Module whose user interface is shown in Figure 4. The Series Management Class sends the modified series list back to the Control Class. The Control Class then builds a new list with only the selected series and sends it to the Plot Class.

Building another series list with only the selected series may seem unnecessary at first glance but one should remember that Plot Class plots all the series received. As already mentioned, Control Class is in charge of all information exchanges among the other three classes, and was designed to generate a new series list (containing only the selected series) and sent it to Plot Class.

The Plot Class plots the selected series list, received from the Control Class, utilizing the template information associated with the plotting type. The user can customize this template information as desired. The customized template information may then be sent back to the Control Class to be stored in a template file (see section 3.1).

Figure 3 shows the data flow for the Generator Tests and Reduced Order Model applications. These two applications build selected series lists that are sent to the Control Class while Plot CEPEL sends the results data file name (Figure 2). As these applications have default

(selected) series lists there is no need to instantiate File Class or the Series Management Class.

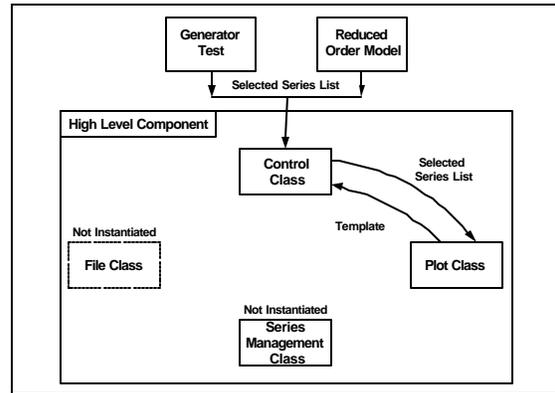


Figure 3 – Generator Test and Reduced Order Model Applications Data Flow.

3. GRAPHICAL USER INTERFACE FEATURES

Plot CEPEL graphical user interface is divided into various modules, for improved productivity in the visualization of results and the elaboration of technical reports. The various modules allow the user to manipulate the high level components classes described in section 2.2. Plot CEPEL is fully compatible with the Windows environment, allowing the use of skills learned from other Windows applications. Advanced features such as drag and drop plots through Object Linking and Embedding (OLE), among others, help making the user rapidly familiar with the software usage.

Hotkeys were also defined to allow faster access to the most utilized graphical user interface features. Another important feature is the ability to export figures from Plot CEPEL in metafile file format. The metafile file format is a vector graphics format employed by a large number of applications developed for Windows-compatible computers. This format reduces the sizes of documents and improves graphical quality. Another program feature is the Online Help, which is quite practical and simple to use.

3.1 Graphical Template Feature

Plot CEPEL incorporates a graphical template feature to store plotting customizations made in one plot for use in other plots of similar type. This feature stores information like pen settings, colors, grids, scales, axis, plot titles, etc., for each plotting type.

The template feature automatically stores in memory all the formatting customizations made for each set of selected series to be plotted. It is possible to store modified templates in separate files for future use (template file *.tpl) or to detailed specify a new default template.

When a plotting file is saved, the template in current use is automatically stored with the file. When a plotting file

is opened, any modified template information that has been stored with the file is automatically loaded, overwriting the default template information.

3.2 Graphic Management Module

The Graphic Management Module is the graphical interface of the Series Management Class with all its functions concentrated in one single dialog.

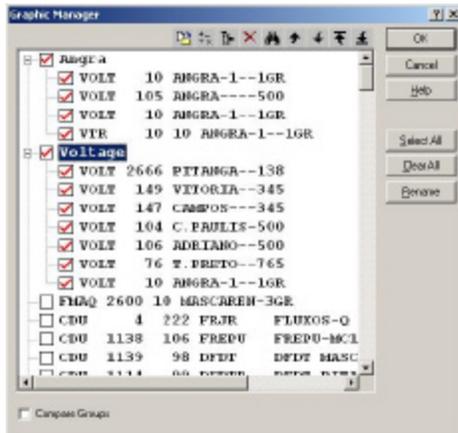


Figure 4 – Graphic Management Module Dialog Showing Various Series Organized into Groups.

The Graphic Management Module allows the user to browse, find, delete or insert new series from other result files, organize similar series into groups, make arithmetic and functional operations with series using the Calculator Module (see section 3.3) and select the desired series for plotting.

3.3 Calculator Module

The Calculator Module allows the user to generate a new series through operations on one or various existing series. The new series is the result of an arithmetic operation involving the original series, a series and a constant or a function applied to one of the series.

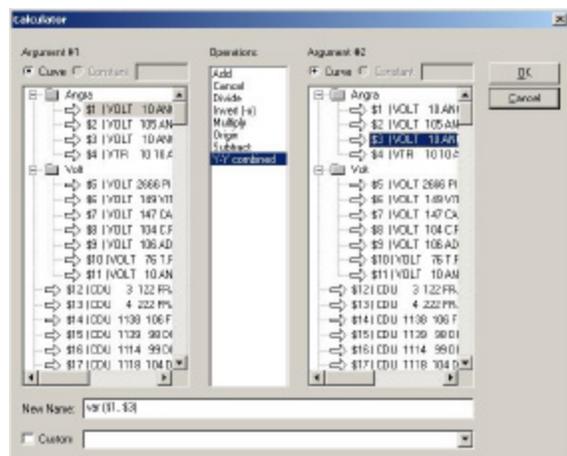


Figure 5 – Calculator Module Dialog.

The functions currently implemented in the Calculator Module are:

- Origin – this function subtracts the value of the initial element in the series from all elements in the entire series; it is routinely used to compare PacDyn and ANATEM simulation results;
- Invert – this function multiplies the entire series by -1;
- Y-Y Combine – this function combines two series discarding the x-values of both series; its main use is to plot a given variable against another.

The user can also generate a new series using the custom option to apply the desired arithmetic operation or function directly. A Fast Fourier Transform function is being implemented.

3.4 Power System Stabilizer Design Module

The Power System Stabilizer Design Module is mostly based on frequency response methods (Nyquist design or GEP design). The module described in this section relates to Nyquist design.

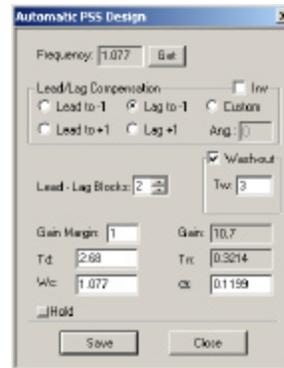


Figure 6 – Power System Stabilizer Design Module Dialog.

Plot CEPEL calculates the Power System Stabilizer (PSS) parameters focusing on pole frequencies selected by the user. Other data such as number of lead blocks, center frequency of these blocks and pole damping ratio (for pole location) may also be specified. The PSS designed is then multiplied to the original transfer function (Figure 16). It is also possible to obtain several power system stabilizer designs to graphically compare their performance.

Once the design phase is completed, the designed PSS can be exported to a PacDyn dynamic data file to be used in future small signal stability studies (see section 3.7).

There are two other design options available in PacDyn: phase compensation design utilizing the widely known GEP(jw) concept and Pole Residue-Phase angle information. These design options are currently being implemented into Plot CEPEL.

3.5 Dominant Pole Spectrum Module

The purpose of this module is to display the dominant pole spectrum of a given transfer function. This module utilizes two result files: one containing the full system pole spectrum and the other containing the full set of zeros for a given transfer function.

Plot CEPTEL shows a dialog box so the user may choose a full system pole spectrum results file and a full system transfer function zeros file.



Figure 7 – Dominant Pole Spectrum Module Dialog.

The visualization comprises three subsequent steps. The pole spectrum (first file) is plotted in the first step and in the second step the transfer function zeros spectrum (second file) is plotted over the pole spectrum. In the last step, Plot CEPTEL performs the pole-zero cancellation according to a given mismatch, displaying then only the dominant pole spectrum (Figure 21).

3.6 File Comparison Module

This module allows the comparison of two or more result files of the same type. The user must first select the files to be compared in the Compare Files dialog box shown below:

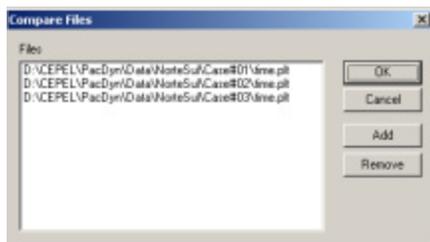


Figure 8 – Compare Files Dialog Box.

The user can then browse and choose the graphics displayed using the Graphics Management dialog (see section 3.2).

3.7 Message Communications with Other Software

Plot CEPTEL directly communicates with various software. For example, when PacDyn calculates a linear time response simulation, it sends a message to the Windows operational system to run Plot CEPTEL and plot the simulation result.

When the Power System Stabilizer Design Module sends a PacDyn dynamic data file with PSS parameters back to PacDyn, Plot CEPTEL automatically sends a message to PacDyn in order to create a new case that

includes the newly created dynamic data file. These message communications increase program automation, greatly improving work productivity in practical engineering studies.

4. RESULTS OF STABILIZATION STUDIES FOR THE BRAZILIAN NORTH-SOUTH-SOUTHEAST INTERCONNECTION

The PacDyn results displayed in this chapter were obtained along the period 1995-1999 in the course of the planning studies of the North-South Brazilian interconnection. These results have been repeatedly obtained using PacDyn and Plot CEPTEL during the CEPTEL software training courses and are reproduced here to show the capabilities of Plot CEPTEL. Details on this study may be found in [18,19].

4.1 System Pole Spectrum and Transfer Function Zeros

The full system pole spectrum can be obtained in PacDyn through the QR algorithm for small and medium size systems. The QZ algorithm is also available in PacDyn for obtaining the system transfer function zeros of small and medium size systems. The results may be displayed either in the complex plane (Figure 9) or in a worksheet (Figure 10).

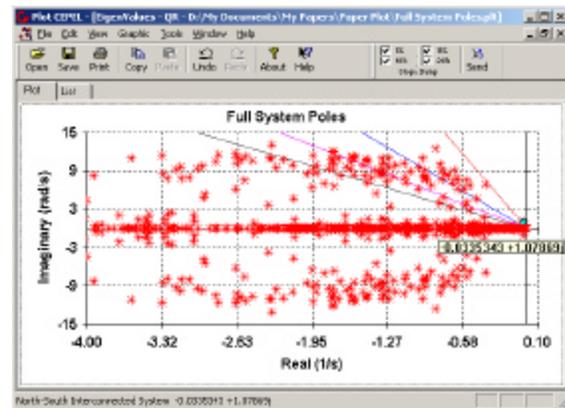


Figure 9 – Full System Pole Spectrum – Complex Plane.

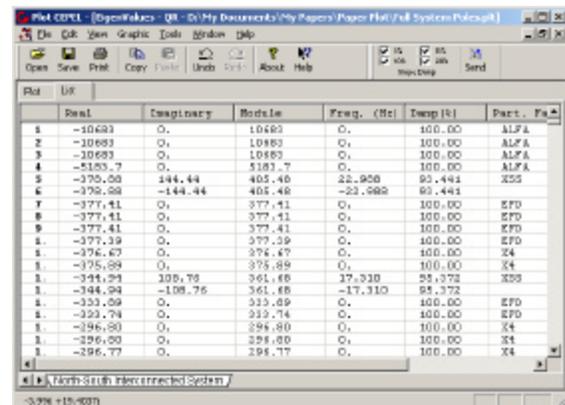


Figure 10 - Full System Pole Spectrum – Worksheet.

4.2 Mode-Shapes

Mode-shapes plots may be visualized in Plot CEPEL by a phasor plot (Figure 11), a bar-chart plot (Figure 12), or as a worksheet (Figure 13).

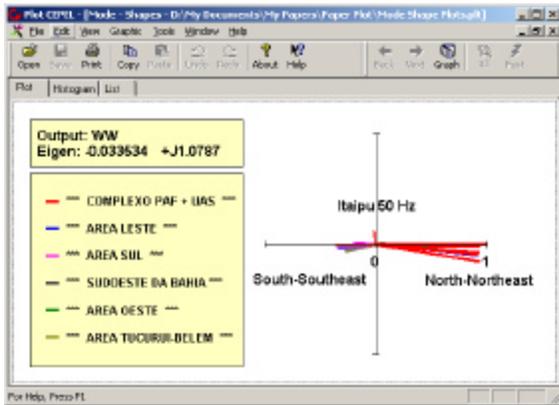


Figure 11 – Mode-Shapes Phasor.

Figure 12 shows the same mode-shape in a bar-chart plot. The positive bars indicate that a given phasor has angle between -90 and 90 degrees (4^{th} and 1^{st} quadrants).

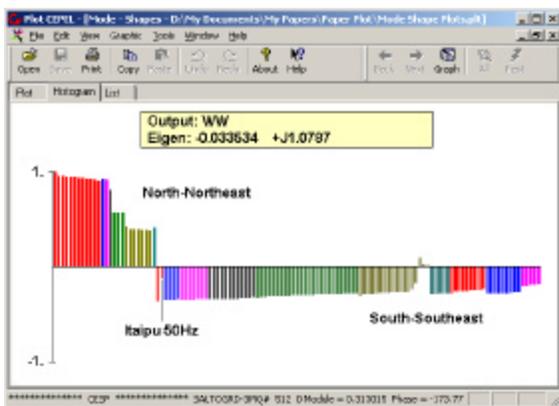


Figure 12 – Mode-Shapes Bar-Chart Plots.

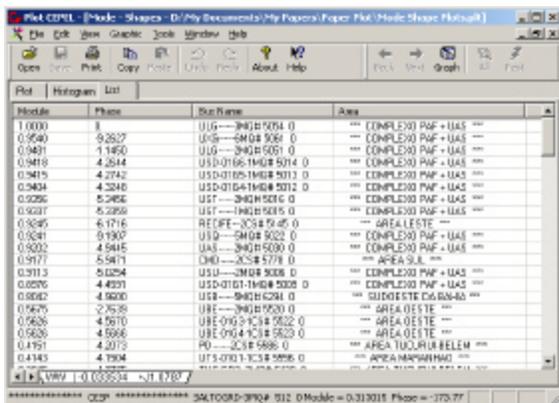


Figure 13 – Mode-Shapes Worksheet.

4.3 Linear Time Response

The linear time response results from PacDyn may be visualized in Plot CEPEL. Terminal power deviations of three major power plants following a specified disturbance [18,19] are depicted in Figure 14.

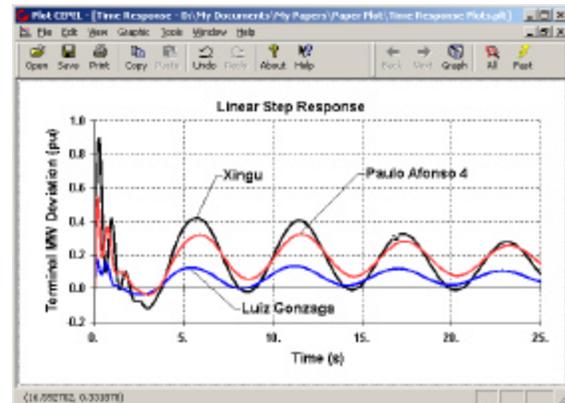


Figure 14 – Linear Step Response Plots.

The non-linear time response simulation results obtained by ANATEM, the transient stability program developed by CEPEL, can also be visualized using Plot CEPEL. Plot CEPEL allows an easy comparison between the results obtained by these two softwares through a drag and drop feature, or the Comparison Module.

4.4 Frequency Response Plots

Frequency response results may be displayed in Nyquist diagrams or Bode plots (both gain and phase plots). Power System Stabilizers (PSS) or Power Oscillation Damping (POD) controllers for FACTS may be designed through use of the Power System Stabilizer Design Module (section 3.4). The uncompensated Nyquist diagram for the North-South TCSC (Thyristor-Controlled Series Compensator) device is shown in Figure 15. The compensated Nyquist curve and the POD parameters for the North-South TCSC device are shown in Figure 16.

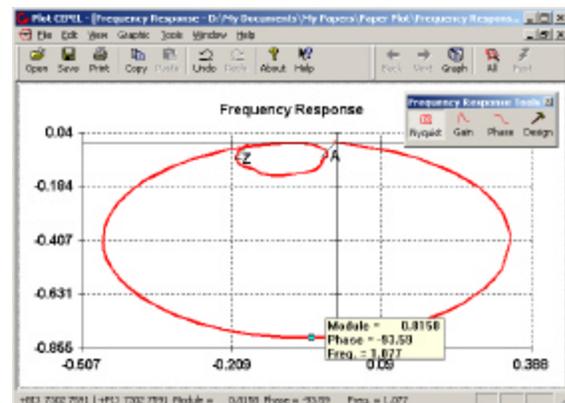


Figure 15 – Frequency Response Plots – Nyquist Diagram.

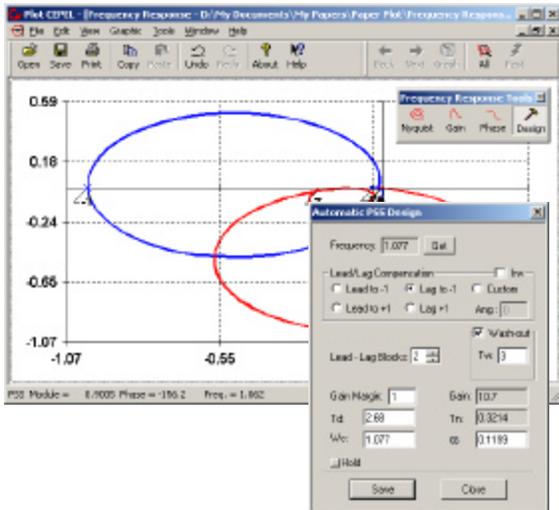


Figure 16 – Nyquist Diagram Showing the Compensated Curve and Power System Design Module with Designed PSS Parameters.

Figure 17 and Figure 18 show the Bode plots, gain and phase respectively, for the same frequency response simulation. The gain axis in the Bode plot of Figure 17 may be displayed in either a linear scale or logarithmic scale (db).

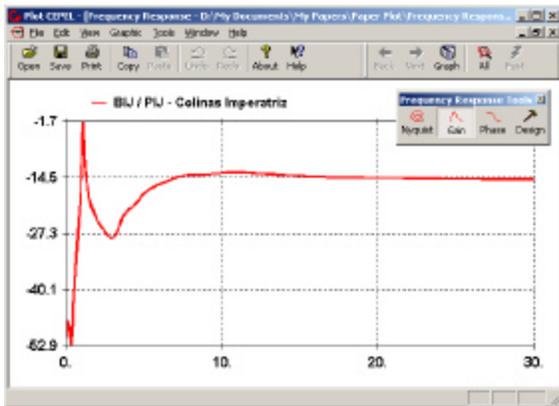


Figure 17 – Frequency Response – Bode Plots – Gain.

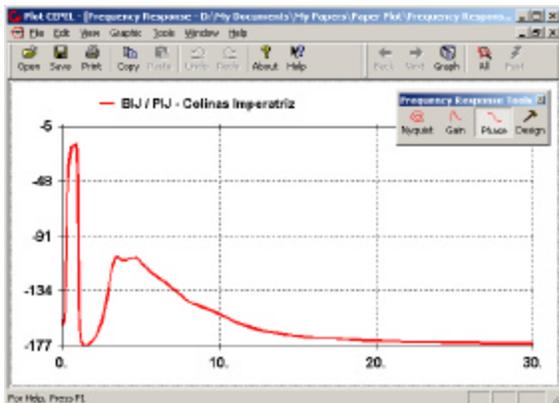


Figure 18 – Frequency Response – Bode Plots – Phase.

4.5 Transfer Function Residues

A set of transfer function residues may be displayed in Plot CEPTEL in a phasor plot (Figure 19) or in a worksheet similar to the mode-shapes worksheet (Figure 11).

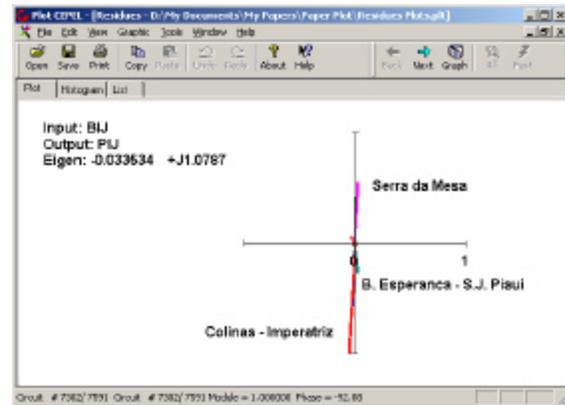


Figure 19 – Residues Phasor Plot.

4.6 Root Locus Plots

Dominant pole and other sparse eigenroutines are employed to automatically obtain root-locus plots for high-order transfer functions, by repeatedly tracking the dominant pole spectrum for a sequence of small changes in specified controller parameters. Figure 20 shows the Root Locus Plot obtained when increasing the gain of the Power Oscillator Damping Controller of one TCSC device in the North-South intertie.

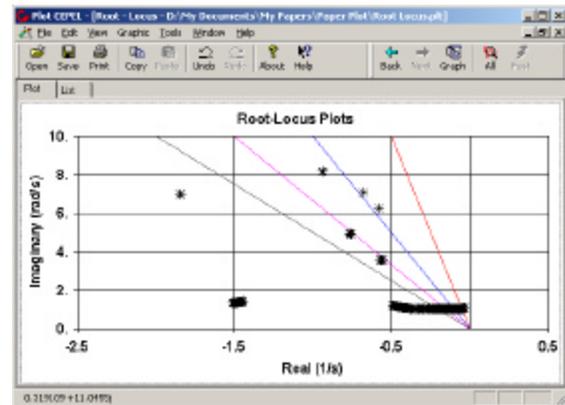


Figure 20 – Root-Locus Plot for POD gain increase.

4.7 Pole-Zero Spectra

The pole-zero spectra plots are obtained using the Dominant Pole Spectrum Module (section 3.5). Figure 21 shows the third and last frame of this module, which displays the dominant pole-zero spectra (see the Full System Pole Spectrum in Figure 9). The chosen transfer function relates to the Itaipu power plant, with the exciter reference voltage as the input variable and the generator rotor speed as the output variable.

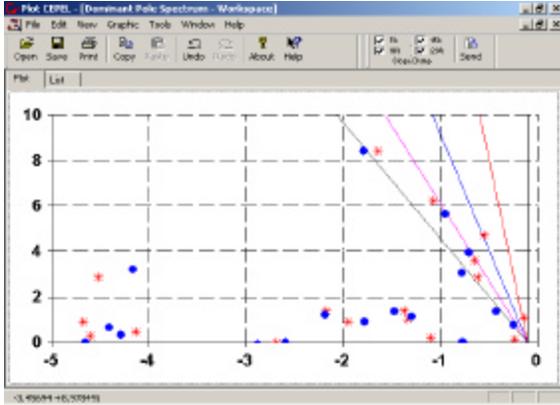


Figure 21 – PoleZero Spectra Plots

4.8 Generator Tests Application

PacDyn generator test allows checking system generators and controls for gross errors in input data. The Generator Tests application displays the results obtained in PacDyn with the use of three main windows, as depicted in Figure 22. The left window shows a list containing several columns with information regarding the study case, the type of generator test, the generator identification, the available generator controls and the damping ratio of the least damped eigenvalue associated with the generator under test. The two windows on the right display the eigensolution list and the step-response plot associated with the single-machine model comprised by the generator and associated controls.

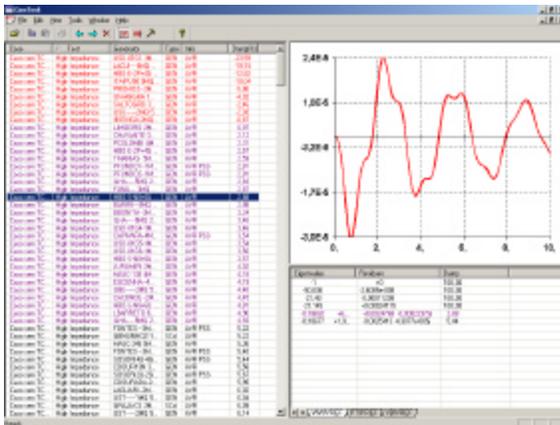


Figure 22 - Generator Tests Application Main Window.

4.9 Model Order Reduction Application

This section illustrates the effectiveness of the model-order reduction techniques available; these techniques rely on the knowledge of the set of dominant poles of a given scalar transfer function for the full-size system. When dealing with large-scale systems, the use of the Dominant Pole Spectrum Eigensolver (DPSE) is recommended. [21].

The time response (Figure 23) is obtained by the Inverse Laplace Transform of a step disturbance applied to a

reduced order model comprised by a selected set of dominant transfer function poles [20,21] and associated residues.

Figure 23 shows the time response of the reduced order model obtained when using three pairs of complex-conjugated poles calculated by the DPSE algorithm [21] in PacDyn. Note that the 6th order model practically reproduces the full system linear step response (1,676 state variables) obtained by step-by-step numerical integration (trapezoidal-rule algorithm).

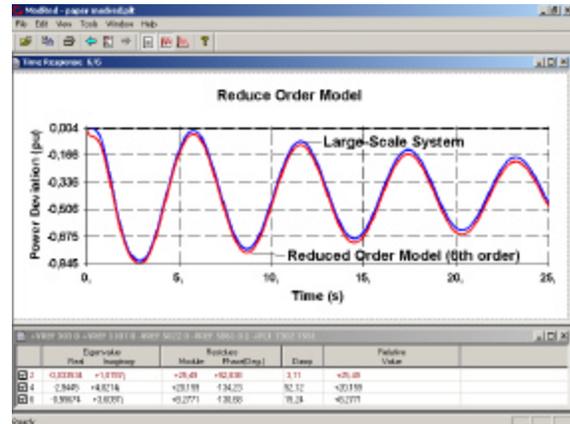


Figure 23 - Time Response of the Reduced Order Model compared to that of the Large Scale System.

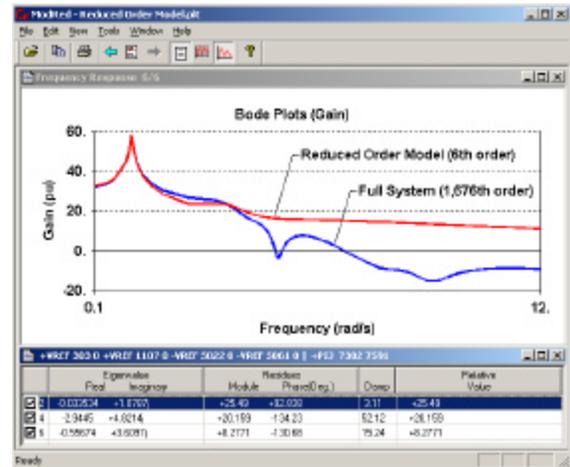


Figure 24 – Frequency Response - Bode Plots (Gain) - Reduced Order Model.

These results are for a Single-Input-Single-Output (SISO) transfer function. Reduced-models for Multi-Input-Multi-Output (MIMO) transfer functions are necessarily of a higher order, but can also be obtained at reasonable cost [23].

Highly reduced models for power system transfer functions have been reported to be cost-effective for the design of damping controllers using multivariable control design techniques.

4.10 Continuation Power Flow

A well-designed continuation power flow program determines quite effectively the system maximum loadability margins for a set of contingencies in the transmission network considering different loading scenarios.

Figure 25 shows the PV curves for two buses in the South region of the Brazilian interconnection. The results were obtained by ANAREDE and visualized by Plot CEPEL. The voltage depressed area is shown to be close to Pelotas 230 kV bus. The Blumenau 230 kV bus is seen to be kept at an adequate voltage level, for the same load and generation changes.

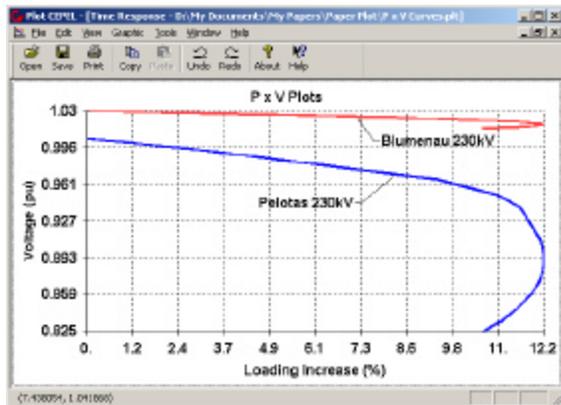


Figure 25 – Continuation Power Flow.

5. FUTURE DEVELOPMENTS

- Development of scripts in order to run Plot CEPEL in “batch mode”. This feature will allow the automatic generation of large sets of figures to be used in technical reports that investigate multi-scenarios conditions.
- Implementation of a more complete module for the design of Power Oscillation Damping Controllers and Power System Stabilizers, incorporating three of the most frequently used techniques.
- Implementation of a new module of Coordinated Design of Damping Controllers, considering multiple scenarios for large scale power systems.
- Addition of new functions to the Calculator Module, such as SQRT, FFT (Fast Fourier Transform), POW, among others.
- Implementation of new high level components for the complete visualization of the HARMZS results.

6. REFERENCES

[1] **F. Montenegro, R. Pacheco**, Object Oriented Programming in C++, Editora Ciência Moderna, Brazil, 1994 (in Portuguese).

[2] **S. Oualline**, Practical C++ Programming, O’Reilly & Associates, Inc., USA, 1995.

[3] **B. Meyer**, Object Oriented Software Construction, 2nd ed., Prentice Hall, USA, 2000.

[4] **N. Martins, P.E.M. Quintão, H.J.C.P. Pinto, A. de Castro, S. Gomes Jr., J.C.R. Ferraz**, “A Small-Signal Stability Program Incorporating Advanced Graphical User Interface”, *Proceedings of the VII SEPOPE*, Curitiba, PR, Brazil, May 2000.

[5] **R.D. Rangel, S. Gomes Jr., C.H.C. Guimarães. N. Martins, A. Castro, H.J.C.P. Pinto, A.R.C.D. Carvalho**, “Recent Developments in ANATEM - A Comprehensive Program for the Analysis of Electromechanical Stability of Large Power Systems”, *Proceedings of the VII SEPOPE*, Curitiba, PR, Brazil, May 2000.

[6] **H.J.C.P. Pinto, J.L.R. Pereira, N. Martins, J.A. Passo Filho, S. Gomes Jr., F.R.M. Alves, J.C.R. Ferraz, R.M. Henriques, V.M. da Costa**, “Needs and Improvements in Power Flow Analysis”, *Proceedings of the VII SEPOPE*, Curitiba, PR, Brazil, 2000.

[7] **J.C.R. Ferraz, N. Martins, H.J.C.P. Pinto, D.M. Falcão**, “Fluxo De Potência Continuado e Análise Modal na Avaliação e Melhoria da Estabilidade de Tensão do Sistema Sul-Sudeste”, *Proceedings of the VII SEPOPE*, Curitiba, PR, Brazil, May 2000, (in Portuguese).

[8] **J.A. Passos Filho, L.C.A. Ferreira, N. Martins, A.C.Z. Souza**, “Benefícios da Representação do Controle Automático e Discreto de Bancos de Capacitores/Reatores e de LTC Sistemáticos em Estudos de Segurança de Tensão”, *Proceedings of the XVII SNPTEE*, Uberlândia, MG, Brazil, Outubro 2003, (in Portuguese).

[9] **N. Martins, R.M. Henriques, A.A. Barbosa, S. Gomes Jr., C.B. Gomes, A.C.B. Martins**, “Impact of Induction Motor Loads in System Loadability Margins and Damping of Inter-Area Modes”, *Load Modelling Panel, IEEE Toronto General Meeting*, USA, July 2003.

[10] **C. Petzold**, Programming Windows, 5th ed., Microsoft Press, USA, 1999.

[11] **D.J. Kruglinski**, Inside Visual C++, Microsoft Press, 4th ed., USA, 1997.

[12] **D. Chappel**, ActiveX and OLE, Microsoft Press, USA, 1996.

[13] **A. Denning**, ActiveX Controls Inside Out, Microsoft Press, USA, 1997.

[14] **D. Rogerson**, Inside COM, Microsoft Press, USA, 1997.

[15] **I. Horton**, Beginning Visual C++ 6, Wrox Press Ltd., UK, 1999.

[16] **H. Schildt**, C++: The Complete Reference, 4th ed., McGraw-Hill, USA, 2003.

- [17] **J. Prorise**, Programming Windows 95 with MFC, Microsoft Press, USA, 1996.
- [18] **N. Martins, A.A. Barbosa, J.C.R. Ferraz, M.G. dos Santos, A.L.B. Bonfim, C.S. Yung, V.R. Oliveira, N.J.P. Macedo**, "Retuning Stabilizers for the North-South Brazilian Interconnection", *Panel Session on System Reliability as Affected by Power System Stabilizers, Proceedings of the 1999 IEEE/PES Summer Meeting*, Edmonton, CA, pp. 58-67.
- [19] **N. Martins, H.J.C.P. Pinto, C.A. Gama, J.A. Cavalcanti, R.L. Leoni, R.V. Souto, N.J.P. Macedo, M.J.X. Eiras**, "Oscillation Damping Analysis and Control Studies of Future Interconnection Between the North-Northeast and South-Southeast Systems", *Proceedings of the V SEPOPE*, Recife, PE, Brazil, 1996.
- [20] **N. Martins, L.T.G. Lima, H.J.C.P. Pinto**, "Computing Dominant Poles of Very High Order Transfer Functions", *IEEE Trans. on Power Systems*, Vol. 11, No. 1, February 1996, pp. 162-170.
- [21] **N. Martins**, "The Dominant Pole Spectrum Eigensolver", *IEEE Trans. on Power Systems*, Vol. 12, No. 1, February 1997, pp. 245-254.
- [22] **S. Gomes Jr., N. Martins, H.J.C.P. Pinto**, "Power System Stabilizer Design by the Nyquist Method, with an Embedded Partial Pole Placement Capability", *Proceedings of the XII Brazilian Automatic Control Conference*, Uberlandia, Brazil, 1998, (in Portuguese).
- [23] **N. Martins, P.E.M. Quintão**, "Computing Dominant Poles of Power System Multivariable Transfer Functions", *IEEE Trans. on Power Systems*, Vol. 18, No. 1, February 2003, pp. 152-159.
- [24] **G. Booch, J. Rumbaugh, I. Jacobson**, The Unified Modeling Language User Guide, Addison-Wesley Object Technology Series, USA, 1999.
- [25] **M. Page-Jones**, Fundamentals of Object-Oriented Design in UML, Addison-Wesley Object Technology Series, USA, 2000.
- [26] **Sergio Luis Varricchio, Sergio Gomes Jr., Nelson Martins, Leandro R. Araújo, Franklin C. Véliz and Cristiano de O. Costa**, "Advanced Tool for Harmonic Analysis of Power Systems", *Proceedings of the IX SEPOPE*, Rio de Janeiro, RJ, Brazil, May, 2004.