Transactions on Power Systems, Vol. 7, No. 4, November 1992

A Toolbox for Power System Dynamics and Control Engineering Education and Research

Joe H. Chow and Kwok W. Cheung
Electric Power Department and
Electrical, Computer and System Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

Abstract This paper discusses the design concept and usage of the Power System Toolbox, a Matlab-based power system dynamics simulation and control design package. The motivation for developing the package is to provide a flexible environment for teaching power system simulation techniques and control design concepts to advanced undergraduate and graduate students, and for graduate students to perform research and development on power systems. The package can be executed without modification on all platforms that Matlab supports, including IBM PC, Macintosh and many workstations.

1. Introduction

This paper presents a prototype power system analysis and design package called Power System Toolbox (PST) for small to medium size power systems. PST is conceptually similar to the simulation package POSSIM/MANSTAB [1] and its main purpose is to allow the user to build simulation programs and state matrices. It is designed for use in an advanced undergraduate and graduate level course on power system computer methods, dynamics and control, and for rapid testing of research and development ideas by graduate students. In light of the increased importance of linear system analysis in power systems [2], PST seems to be a useful contribution.

Although many simulation packages ranging from large scale simulation programs such as the EPRI-Extended Transient/Mid-Term Stability Program (ETMSP) and the PTI-Power System Simulation/E Program (PSS/E) to education software [3] for small systems are available, PST is unique as it provides equipment models (coded as functions) such that the user can readily assemble the available models into a simulation program. As a teaching tool, it allows the students to reinforce their understanding on the steps involved in performing a power simulation learned from textbooks. In addition, new models can be added by following a set of simple rules. The PST environment can also readily incorporate functions such as energy function computation, coherent machine identification and dynamic equivalencing, which are usually external to a simulation package.

PST is coded in Matlab ([4],[5]), a high level scientific language, and can be executed without modification on all platforms that Matlab supports, including IBM PC, Macintosh and many workstations. Although Matlab has only been used in few applications in power system engineering [6], it is one of the high level scientific languages that has gained acceptance in many engineering discipline such as signal processing and control system design. Many universities, including Rensselaer Polytechnic Institute, have Matlab available on workstations. With many built-in mathematical functions such as matrix inversion and integration, the use of Matlab significantly reduces the coding effort over C or PASCAL. For example, network solution using LU decomposition requires a single line code in Matlab. The graphics capability allows the user to display time response in 2D and 3D. For linear system analysis and control design, a Control System Toolbox is available [7].

The remainder of the paper is organized as follows. Section 2 discusses the capabilities of PST and Section 3 the software development philosophy. Sample applications are given in Section 4. Some potential educational usage is suggested in Section 5. Future enhancement to the package is outlined in Section 6.

2. Capabilities

2.1 Simulation

The basic purpose of PST is to provide models for performing power system simulation. As a result, the following functions are provided by PST:

1. Network functions:
   (a) Power system loadflow solution – A function to solve loadflow using a Newton-Raphson algorithm in polar coordinates (bus angles and voltages) is available. The user has the option to require the loadflow jacobian be re-computed at every other iteration to save computation time.
   (b) Reduced Y matrix computation – Constant impedance loads are eliminated in the network solution. The voltage of these eliminated buses can be obtained through a (complex) voltage reconstruction matrix.
   (c) Non-conforming load network solution – For systems with non-conforming loads, including static var control systems (SVC), and FACTS devices (such as controlled phase-shifting transformers), the non-conforming load buses are not eliminated in the reduced Y matrix. A Newton iteration is carried out to solve for the non-conforming load bus solutions.
2. Dynamic models:
   (a) Machine models – Three types of machine models are available: an electromechanical model, a model in-

92 WM 082-8 PWRS A paper recommended and approved by the IEEE Power Engineering Education Committee of the IEEE Power Engineering Society for presentation at the IEEE/PES 1992 Winter Meeting, New York, New York, January 26 - 30, 1992. Manuscript submitted August 21, 1991; made available for printing November 20, 1991.

0885-8950/92$03.00©1992 IEEE

cluding transient effects, and a model including subtransient effects [8].

(b) Excitation system models – IEEE type DC1, DC2 and ST3 models [9] are available.

(c) Static var control system model – a simple SVC model used in [10] is available.

(d) Power system stabilizer (PSS) model – a general purpose single input PSS model [11] is available. The user can specify the input variable and the compensator parameters accordingly.

These functions can then be put together according to the flow diagram in Figure 1 to form a simulation program. First the loadflow of a system is obtained. Then the reduced Y matrices of the pre-fault, fault-on and post-fault systems are computed. In addition, the loadflow solution is used to initialize the state variables of the machine models and their control devices. The trajectory computation consists of a transformation from the individual machine dq-axis to the system reference coordinate for computing the network solution, and the computation of the rate of change of the state variables. PST provides an Euler integration routine. However, the user can set up more accurate and reliable integration routines such as predictor-corrector methods and variable stepsize techniques. As a gauge of the programming effort required, a simulation program using the Euler integration method requires less than 60 lines of Matlab code (including the specification of loadflow parameters and fault sequence) and a trapezoidal corrector method requires less than 80 lines of code.

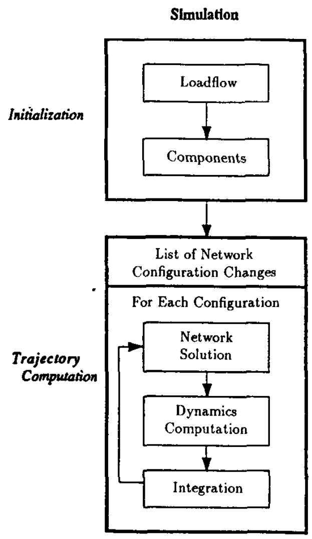

Figure 1: Program Structure for Simulation

2.2 State Matrix Building

The functions used in simulation can also be used to build the state matrix model of a power system

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \tag{1}$$

where x is the vector of state variables, u the vector of input variables, and y the vector of output variables. The structure of a state matrix building program is very similar to a simulation program, except that the integration part is replaced by a sequential perturbation of the state and input variables (Figure 2). The entries of the state matrices are obtained by partial derivatives, which are numerically obtained by dividing the state derivatives and output variable increments by the small perturbations. Once the model (1) is obtained, the built-in Matlab functions for eigenvalue and frequency response computation can be used to investigate the system characteristics, and control system design can be performed.

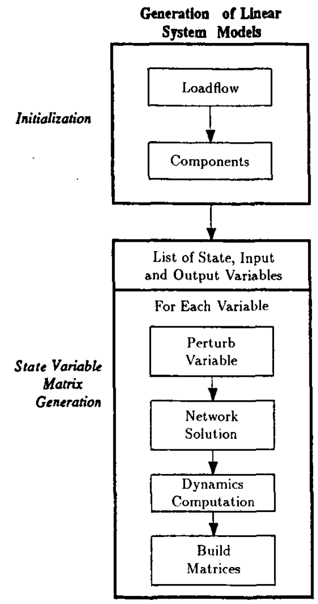

Figure 2: Program Structure for State Matrix Building

2.3 Coherent area identification

The slow coherency method in [12] is based on the modeshapes of the low frequency electromechanical modes. To apply the method, the state matrix the matrix $M^{-1}K$ is constructed, where M is the diagonal matrix of machine inertias, and K the matrix of synchronizing coefficients. The eigenvectors of $M^{-1}K$ are the modeshapes. The matrix V_s is formed by the slow eigenvectors and can be used to identify the coherent groups of machines using a PST function performing the algorithm in [12].

3. Software Development

The design of PST is based on the following objectives:

1. to provide a simulation environment that requires a minimal amount of time to be proficient;

2. to allow straightforward coding of routine simulation programs, which in most cases, requires the editing of an existing simulation program;

3. to allow advanced users to build additional models and functions, which is useful for rapid development of prototype models and testing of research ideas;

4. to promote the use of graphics for the presentation of results.

Some of the features in PST that support the objectives include:

1. The machine and control system variables are declared as

global variables (like COMMON in FORTRAN) such that they are available in all the functions. The declaration is needed to be done only once at the beginning of simulation program. The meaning of the variables are transparent to the users, for example, the variable **mac_ang** is a vector of δ, the machine angle, and **dmac_ang** is $\dot{\delta}$. These variables are always available to the user to display in matrix or graphical form.

2. Each model consists of three parts: the first part for linearization, the second part for machine-network interface, and the third part for dynamics computation. Thus the code on the sequencing of the models used for linearization is repeated at the network interface calculation and the dynamics computation. A flag controls the execution of the appropriate parts of the models. New models added to PST need to follow the same coding convention.

3. To speed up the Matlab computation, a vectorized computation option is available for the machine models. Instead of a do loop to compute the variables of each machine sequentially, the same variable of all the machines is computed simultaneously in a vector form. This reduces the interpretation of machine codes and streamlines the data retrieval process.

4. The usage of the functions is documented in the same style as in [4]. Figure 3 contains an example of the documentation of the reduced Y matrix function. On-line help is also available by typing **help red_Ybus**. To provide this capability, the first part of a function file must contain a brief description of the usage of the function. Note the each function can be designed for several different usage. For example, if seven output variables are requested, the function **red_Ybus** would set up the reduced Y matrices for non-conforming load solution (see Figure 3).

5. Demo files are available to introduce the users to the package. A routine simulation program for a new data set can be readily edited from a demo file by editing the input loadflow and machine data, and the fault sequence.

4. Test Results

The functions of PST are tested using 386Matlab on a 386 25 MHz PC with a 25 MHz 387 co-processor and 4 Mb RAM. For 386Matlab, there is no limitation on the number of matrix arrays and the size of the matrix arrays. However, the total amount of data that can be handled is limited by the size of the RAM. The 4 Mb RAM allows Matlab to handle about 1.5 million double precision numbers in the memory.

4.1 Loadflow

The loadflow performance results for a 9-bus system [12], a 68-bus system [12] and a 145-bus system [13] are shown in Table 1. These loadflow results were obtained with the jacobian recomputed at every other iteration. The voltage profile of the 145-bus system in Figure 4 is obtained by using the Matlab bar chart plotting function **bar**. If the coordinates of the buses were known, it would be relatively straightforward to show the voltage profile in 3D. If a Sun Sparc 1+ workstation were used for the loadflow solutions, a speed up of about three times could be expected.

Function:

red_Ybus

Purpose:

Form the reduced admittance matrix

Synopsis:

```
[red_Y,rec_V] = red_Ybus(bus,line)
[Y11,Y12,Y21,Y22,rec_V1,rec_V2,bus_ord] = red_Ybus(bus,line)
```

Description:

[red_Y,rec_V] = red_Ybus(bus,line) uses the bus data in bus, the line data in line and the machine reactance in mac_con to return the reduced admittance matrix red_Y and the voltage reconstruction matrix rec_V so that

$$I_g = \text{red\_Y} * V_g$$
$$V_b = \text{rec\_V} * V_g$$

where I_g is a column vector of generator current injection, V_g and V_b are column vectors of generator bus voltages and load bus voltages, respectively.

[Y11,Y12,Y21,Y22,rec_V1,rec_V2,bus_ord] = red_Ybus(bus,line) is the reduced admittance matrix in partitioned form required when there are non-conforming load buses in the system. The function uses the bus data in bus, the line data in line, the machine reactance in mac_con and the load data in load_con to return the reduced admittance matrices Y11, Y12, Y21, Y22 and the voltage reconstruction matrix rec_V1, rec_V2 so that

$$I_g = \text{Y11} * V_g + \text{Y12} * V_{nc}$$
$$V_b = \text{rec\_V1} * V_g + \text{rec\_V2} * V_{nc}$$

where V_{nc} is the column vector of the non-conforming load bus voltages. The matrices Y21, Y22 and the bus reordering information contained in the column vector bus_ord are used in nc_load.

Figure 3: Reduced Y Matrix Computation Documentation

Table 1: Loadflow Performance

System	Convergence Tolerance	Number of Iterations	Solution Time
9-bus	10^{-13}	5	3.1 sec
68-bus	10^{-12}	7	90 sec
145-bus	10^{-11}	3	317 sec

4.2 Simulation

The interarea mode case of the 50-machine, 145-bus system [13] was investigated using a simulation program consisting of PST models. Electromechanical models were used for all the machines. According to [13], Buses 93 and 110 were each set at 1580 MW generation, and Buses 104 and 111 were each set at 1775 MW generation. A 0.108 second duration short circuit fault was applied at Bus 7 and cleared by opening the line between Buses 6 and 7. The response of the machine angles is shown in Figures 5 and 6. Note that 29 machines (1-22, 24-27, 33-35) became unstable and separated from the other machines.

4.3 Eigenvalue and Coherency Analysis

The pre-fault steady-state operating condition of the 50-machine, 145-bus system used in the simulation of the interarea mode was used to generate a linearized state space model. Using the eigenvalue computation function in Matlab, the five slowest modes of the system were found to be $\pm j1.8295$, $\pm j1.9822$, $\pm j2.7024$, $\pm j3.0081$, and $\pm j3.7098$.

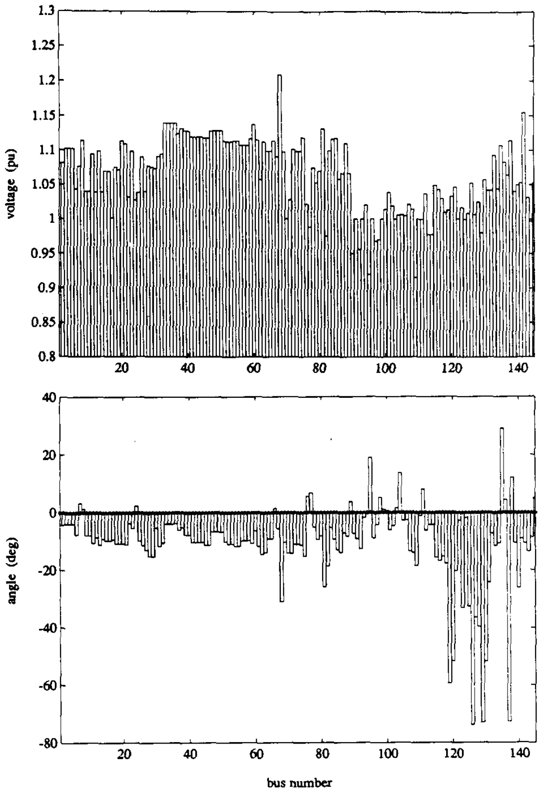

Figure 4: Voltage Profile of 145-Bus System

The mode shapes of the low frequency modes can be used to identify slow coherent areas. One of the means of identifying coherent groups of machines is to form the coherency map matrix

$$C_m(i,j) = v_i v_j^T / (\|v_i\| \|v_j\|), \tag{2}$$

where v_i is the ith row of V_s whose columns are the eigenvectors corresponding to the low frequency modes. A tolerance $tol < 1$ is selected so that machines i and j are said to be coherent if $C_m \geq tol$. A visual effect of coherency can be achieved by displayed the matrix

$$C_{m0} = \max\{(C_m - tol * J), Z\}, \tag{3}$$

where J is a matrix of ones and Z is a matrix of zeros. Thus if the (i,j) entry of C_{m0} is nonzero, then Machines i and j are coherent. The C_{m0} matrix for the 50-machine system using $tol = 0.95$ and the two slowest modes is shown in Figure 7.

The coherency map matrix C_m can also be used to find coherent areas. Using a PST function based on a few rules on tight and loose coherency using $tol = 0.95$, six coherent areas are found (Table 2). Note that 28 out of 29 machines in Area 1 correspond to the unstable machines in the interarea mode simulation. Machine 36 in Area 1 is stable and Machine 16 (not in Area 1) is unstable. Machines 16 and 36 are the so-called boundary machines, that is, they are located close to the boundaries between the areas. The coherency of these machine are typically difficult to classify exactly. When more low frequency modes are used in the identification, Machines 16 and 36 leave their respective areas and form new areas.

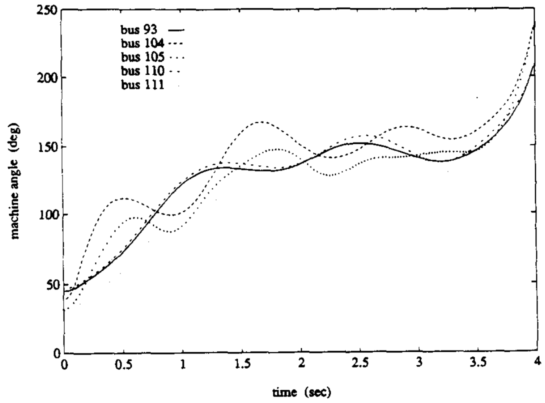

Figure 5: Machine Angles of 50-Machine System

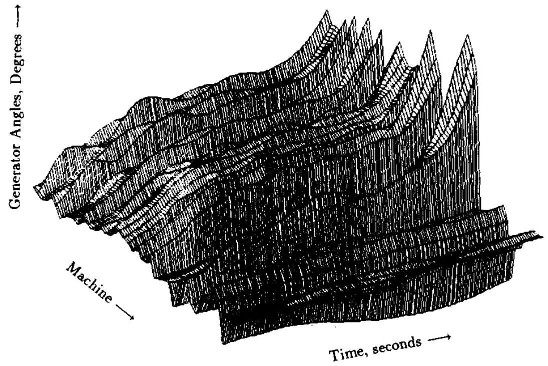

Figure 6: Machine Angles in 3D Display

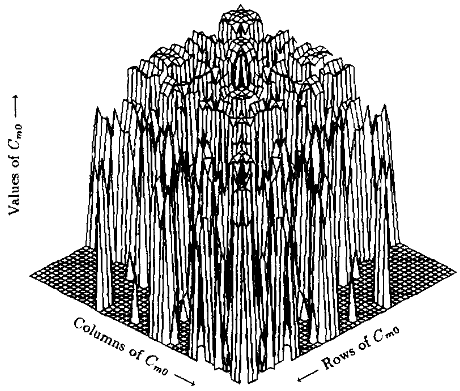

Figure 7: 3D Display of C_{m0}, the (1,1) entry is the farthest point

5. Educational Use

PST can be used at all universities that have license to use Matlab. PST is computer platform independent and can be set up quite readily.

For a course on computer methods in power systems, a student can be required to build simulation programs using the basic functions provided by PST. A typical application is to in-

Table 2: Coherent Areas of 50-Machine System

Area	Machines
1	1-15,17-22,24-27,33-36
2	40-42,44,50
3	16,23,28-32,37-39,47-49
4	46
5	45
6	43

vestigate critical clearing time. Additional exercises can include building new excitation system models, and investigating different loadflow and integration techniques.

For a course on power systems dynamics and control, the following exercises can be performed using PST:

1. Study of the effect of SVC and various types of excitation systems on transient stability.

2. Building state matrices which can then be used for eigenvalue computations, coherency investigation, synchronizing and damping torque calculations, and tuning of power system stabilizers.

3. Voltage collapse study using non-conforming load models.

4. Illustration of energy functions for direct stability computation.

6. Future Development

PST provides a flexible environment such that new models, functions and solution technique can be readily added. A partial list of functions and models to be developed include:

1. a sparse factorization function to be used with loadflow and network solutions;

2. additional excitation system models, HVDC models, turbine-generator models, and FACTS models;

3. a short circuit analysis function;

4. an interactive graphical means of inputting data;

5. network aggregation functions for dynamic equivalencing [14];

6. linearization using analytical formulas instead of the perturbation approach [15].

7. Conclusions

A Power System Toolbox to perform power system simulation and state matrix building has been described. By building PST on Matlab functions, users can be proficient in using PST with a minimal amount of training. Since many universities have Matlab available, PST can be used for design projects for courses on power system dynamics, control and computer methods.

Acknowledgment

The PST loadflow was converted from a FORTRAN Newton-Raphson loadflow program provided by Professor Shep Salon (RPI). The GE PSEC notes by Bill Price were helpful in the development of PST. We express our thanks to them.

References

[1] E. V. Larsen and W. W. Price, "MANSTAB/POSSIM Power system Dynamic Analysis Programs – A New Approach Combining Nonlinear Simulation and Linearized State-Space/Frequency Domain Capabilities," IEEE PICA Conference, Toronto, Canada, 1977.

[2] J. F. Hauer, editor, *Eigenanalysis and Frequency Domain Methods fro System Dynamic Performance,* IEEE Publication 90TH0292-3-PWR, 1990.

[3] D. C. Yu, S.-T. Chen and R. F. Bischke, "A PC Oriented Interactive and Graphical Simulation Package for Power System Study," *IEEE Trans. Power Systems*, vol. 4, pp. 353-360, 1989.

[4] J. N. Little, *Matlab: User's Guide,* The Mathworks, Inc.

[5] L. L. Scharf and R. T. Behrens, *A First Course in Electrical Engineering with MATLAB Programs and Experiments,* Addison-Wesley, 1990.

[6] I. Kamwa, P. Viarouge and J. Dickinson, "Optimal Estimation of the Generalized Operational Impedances of Synchronous Machines from Short-Circuit Tests," *IEEE Trans. Energy Conversion,* vol. 5. pp. 401-407, 1990.

[7] J. N. Little and A. J. Laub, *Control System Toolbox: User's Guide,* The Mathworks, Inc., 1985.

[8] IEEE Committee Report, (P. L. Dandeno, Chairman) "Current Usage and Suggested Practices in Power System Stability Simulations for Synchronous Machines," *IEEE Trans. Energy Conversion,* vol. 1, pp. 77-93, 1986.

[9] IEEE Committee Report, "Excitation System Models for Power System Stability Studies," *IEEE Trans. PAS*, vol. PAS-100, pp.494-507, 1981.

[10] E. V. Larsen and J. H. Chow, "SVC Control Design Concepts for System Dynamics Performance," in *Applications of Static Var Systems for System Dynamics Performance,* IEEE Publication 87TH0187-5-PWR, 1987.

[11] E. V. Larsen and D. A. Swann, "Applying Power System Stabilizers, Parts I-III," *IEEE Trans. PAS*, vol. 100, pp. 3017-3046, 1981.

[12] J. H. Chow (editor), *Time-Scale Modeling of Dynamic Networks with Applications to Power Systems,* Springer-Verlag, 1982.

[13] IEEE Committee Report (V. Vittal, chairman), "Transient Stability Test Systems for Direct Stability Methods," IEEE Winter Power Meeting, Paper 91 WM 224-6 PWRS, 1991.

[14] J. H. Chow, R. A. Date, H. Othman and W. W. Price, "Slow Coherency Aggregation of Large Power Systems," in *Eigenanalysis and Frequency Domain Methods for System Dynamic Performance,* IEEE Publication 90TH0292-3-PWR, 1990.

[15] D. Y. Wong, G. J. Rogers, B. Porreta and P. Kundur, "Eigenvalue Analysis of Very Large Power Systems," *IEEE Trans. on Power Systems,* vol. 3, pp. 472-480, 1988.

About the Authors

Joe H. Chow is an Associate Professor of Electric Power Engineering Department and Electrical, Computer and Systems Engineering Department at Rensselaer Polytechnic Institute. He received his PhD degree from the University of Illinois, Urbana, in 1977, and worked at General Electric Company, Schenectady, as an application engineer before joining RPI in 1987. His research interests include power systems dynamics and control, coherency, singular perturbations and decentralized control.

Kwok W. Cheung received his BS degree from National Cheng Kung University, Taiwan, in 1986 and his MS degree from the University of Texas at Arlington in 1988. Since then he has been a research assistant in the Electrical, Computer and Systems Engineering Department at Rensselaer Polytechnic Institute, where he is pursuing a doctoral degree. His current research interests include singular perturbations, control theory and power systems.