IEEE PES Task Force on Benchmark Systems for Stability Controls

Simplified 14-Generator Model of the South East Australian Power System:

(Including implementations in Mudpack for small-signal analysis and PSS/E for transient-stability analysis)

Tool: AUTimeSeriesTools_Ver01

Version 1 - 18 February 2014

Mike Gibbard & David Vowles

Power Systems Dynamics Group
School of Electrical and Electronic Engineering
The University of Adelaide, South Australia

A large number of small- and large-disturbance time-domain studies have been conducted on the simplified 14-generator model of the South East Australian power system [1]. The results of these studies are provided to allow the researcher to benchmark their own implementation of the model against implementations reported in [1].

This document describes (i) the format of files used to store time-series data from time-response simulations conducted in the PSS/E [2] and Mudpack [3] programs; (ii) simple Matlab [4] tools to allow the researcher to readily extract and/or graphically display the time-responses stored in such files; and (iii) the procedure for installing the tools.

1 Time-series data formats

Time-series data is provided in two formats: (i) native Matlab format; and (ii) comma-separated-value format. For Matlab users, the former is usually more convenient; however, the CSV format is a very common format that can be imported into a number of third party plotting programs.

1.1 Native Matlab format (*.mat files)

The time-series data is stored in a Matlab *.mat file with the following components:

© The University of Adelaide

1. channelNames – Matlab cell array containing a list of variable names in the format <varName>.<devName> for which time-series data is stored; where <varName> is substituted with the name of the variable (e.g. ‘P’ for power, ‘Vt’ for bus voltage magnitude, etc.) and <devName> is substituted with the name of the device at which the variable is monitored (e.g. HPS_1, NPS_5, etc.)
2. channelUnits – Matlab cell array containing the units of the corresponding variable, that is, channelUnits{j} contains the units of the variable channelNames{j} (e.g. ‘pu’, ‘MW’, ‘Mvar’, etc.)
3. TT – Matlab column vector containing the time-points, in seconds, at which the variable values are stored.
4. YY – Matlab matrix containing the responses of the variables: YY(i,j) contains the value of the variable channelNames{j} at time TT(i). The variable is in units of channelUnits{j}.

The following additional optional information fields may also be present:

5. Notes – Matlab cell array of strings containing a description of the study-case and disturbance which produced the time-response.
6. filename – Name of the file from which the time response data was loaded.
7. CaseID – Case identifier string.

1.2 CSV format (*.csv files)

The time-series data is stored in ASCII text files in CSV format as follows:

1. Line 1 – Integer “N” containing the number of lines of “Notes” describing the study case from which the time-series data was derived. (N = 0 means no case notes are provided.)
2. Lines 2 to N+1 – Contain the “Notes” on the study case
3. Line N + 2 – List of comma-separated strings denoting variable names. The first element corresponding to the first column of data is always “TIME”. The second and subsequent elements correspond to the names of the variables whose responses have been recorded in the corresponding columns in the file. The variable names are in the format <varName>.<devName>; where <varName> is substituted with the name of the variable (e.g. ‘P’ for power, ‘Vt’ for bus voltage magnitude, etc.) and <devName> is substituted with the name of the device at which the variable is monitored (e.g. HPS_1, NPS_5, etc.)
4. Line N + 3 – List of comma-separated strings denoting variable units. The first element corresponding to the time vector is always “s” for seconds. The second and subsequent elements are the units of the corresponding variables listed in the previous line.
5. Lines N+4 to end – Matrix of simulation results. The first column in the matrix is the time-vector; columns 2 to the end contain the responses of the corresponding variables listed on line N + 2 in the units listed on line N + 3.

2 Rudimentary Matlab post-processing tools

Several rudimentary Matlab functions are provided to allow the researcher to conveniently extract and/or graphically display time-response data. These are intended to facilitate the researcher in benchmarking their own implementation of the simplified 14-generator model against the PSS/E and/or Mudpack implementations described in [1]. The tools are simple and are provided on an as-is basis as a convenience. They are not intended to be “production quality” and as such error-checking is rudimentary.

A Matlab script ‘AUTRES_Demo.m’ containing working examples of the use of the tools is provided.

2.1 Function AUTRES_Load

timeResponseData = AUTRES_Load(filename,caseID)

Purpose:
Load into Matlab memory the time-series data stored in the specified file.

Input Arguments:

filename	Name of native Matlab (*.mat) or CSV (*.csv) file containing the time series data to be loaded into Matlab. The file must be in one of the formats specified in Section 1.
caseID	(Optional) Case identification code that the user can, for example, use in plot titles or legends, etc.

Output Arguments:

timeResponseData	Matlab structure containing the fields defined in Section 1.1. This structure is referred to as a time-response data structure in the following.

Description:
This function is used to load time-series data stored in one of the formats specified in Section 1 into Matlab memory. If filename has the extension ‘.mat’ it is assumed the file is formatted according to the specification in Section 1.1; if it has the extension ‘.csv’ its is assumed to be in the format specified in Section 1.2. Rudimentary checks are made to verify the file is in the expected format.

If the user does not supply the caseID argument then the base name of the file is used as the case identifier.

The other functions in this tool box operate on time-response data structures.

Example:

```
% Load time response data in two CSV formatted files into Matlab
%
fileList = {'IEEE_Case01_VrefStep_HPS_1_PSSE.csv', ...
            'IEEE_Case01_VrefStep_HPS_1_MP.csv'};

timeResponseList = {};
for i = 1:length(fileList)
   timeResponseList{i} = AUTRES_Load(fileList{i});
end
```


© The University of Adelaide

2.2 Function AUTRES_Contents

str = AUTRES_Contents(timeResponseData)

Purpose:
List the Notes, channelNames and channelUnits for the specified time-response data structure.

Input Arguments:

timeResponseData	Matlab time-response data structure whose contents are to be displayed.

Output Arguments:

str	(Optional) Formatted string containing the Notes for the study case and a list of variables whose time-responses are stored in the structure. If str is not provided then this information is displayed on the Matlab console.

Description:
This function is used to display the contents of the specified time response data structure. It lists the Notes stored in the structure which typically describe the study case and disturbance; and the function lists of the variables whose responses are stored in the structure.

Example:

```
% Display the contents of the time-response file
% 'IEEE_Case01_VrefStep_HPS_1_PSSE.csv'
%
tr = AUTRES_Load('IEEE_Case01_VrefStep_HPS_1_PSSE.csv');

AUTRES_Contents(tr);

... The following is displayed in the Matlab console ...

----------------------------------------------------------------------------
Simplified 14 generator model of the South East Australian Power System
Small-signal step-response simulation results.
CaseID:      Case01_VrefStep_HPS_1
Disturbance: 0.5%% step on Vref.HPS_1
Source:      PSS/E v29.5 simulation,
             Performed by David Vowles and Mike Gibbard
             Power Systems Dynamics Group,
             School of Electrical & Electronic Engineering,
             The University of Adelaide, South Australia
             Email: david.vowles@adelaide.edu.au
Date:        10:36:47 on Friday, 21 February 2014

Channels ...

  1  P.HPS_1 (MW)
  2  W.HPS_1 (pu)
  3  Vt.HPS_1 (pu)
  4  Q.HPS_1 (Mvar)
  5  EFD.HPS_1 (pu)
  6  IFD.HPS_1 (pu)
----------------------------------------------------------------------------
```

2.3 Function AUTRES_Extract

[IsFound,tt,yy,chanName,chanUnits] = AUTRES_Extract(timeResponseData,var)

Purpose:
Extract the time response of one variable from a time response data structure.

Input Arguments:

timeResponseData Time response data structure from which the response of a single variable is to be extracted.

var String containing the name of the variable whose time response is to extracted. (The search for var in the channelNames component of the timeResponseData structure is case insensitive. Thus, if var = ‘p.hps_1’ then it will match ‘P.HPS_1’ stored in the channelNames array.).

Output Arguments:

IsFound Flag to indicate if the response of the specified variable has been found in the timeResponseData structure. IsFound = 1 if the variable is found; = 0 otherwise.

tt Time vector (s).

yy Response of var; yy(i) is the value of chanName in units of chanUnits at time tt(i)

chanName Name of the selected variable (var) in the same case as it stored in the timeResponseData structure.

chanUnits String containing the units of the variable chanName.

Description:
This function can be used to extract the response of a single variable from the time response data structure in a format that is then straight-forward for the user to, say, compare with the corresponding response obtained with some other simulation package.

Example:

```
% Extract the response of the variable 'P.GPS_4' from the tres structure
%
[IsFound,tt,yy,chanName,chanUnits] = AUTRES_Extract(tres, 'p.gps_4');
```


© The University of Adelaide

2.4 Function AUTRES_Plot

AUTRES_Plot(timeResponseData,varList,varargin)

Purpose:
Plot selected time responses in a straight-forward manner with a minimal user interface. Provided as a stand alone plotting program with the simplified 14 generator model of the South East Australian power system.

It is possible to overlay on a single plot responses of multiple variables from multiple simulations.

Input Arguments:

timeResponseList:

(1) a time response data structure, usually obtained using AUTRES_Load; or
(2) a string containing the name of a time response file (the data will be loaded from the file with AUTRES_Load); or
(3) A cell array of items in (1) and/or (2), in which case the time responses of the selected variables in each of the multiple studies will be displayed.

varList: Either:
(1) a string containing the name of the channel to be plotted; or
(2) a cell array of such strings, in which case the response of each selected variable in each of the selected study will be displayed.

varargin Optional arguments are entered in the following key, value pairs. The keys are enclosed in single quotes (e.g. 'title')

title	String or cell array of strings to be displayed as a plot title. Default: no title is displayed.
xlabel	String to be displayed as the x-label on the plot. Default: 'Time (s)'
ylabel	String to be displayed as the y-label on the plot. Default: If varList contains only a single variable then ylabel is the name of the variable and the associated units in brackets; otherwise no ylabel is displayed.
legend	Cell array of strings to be used as a plot legend. legend{i,j} is the legend string for variable varList{j} from the response timeResponseDataList{i} Default: If time response data from multiple studies is displayed the legend for each response includes the caseID of the associated time response. If more than one variable is displayed the legend for each curve includes the name of the variable.
tmin,tmax	Minimum, maximum time (s) displayed on the time axis. Default: Minmum / maximum time in the time-vectors of the displayed responses.
ymin,ymax	Minimum, maximum values of the vertical axis. Default: Minimum / maximum values of the displayed responses within the displayed time-range.

Output Arguments:

None

Description:
AUTRES_Plot is intended to be used in the following ways:

1. Plot the response of a single variable from a single study case;
2. Overlay the responses of two or more variables from a single study case on a single plot;
3. Overlay the responses of a single variable from two or more study cases on a single plot;
4. Overlay the response of multiple variables from multiple study cases on a single plot.

Refer to the following for examples of each of these.

Example 1: (Plot single variable from single study case)
Plot the single variable (P.BPS_2) from a single time response (Case01)

```
Case01 = AUTRES_Load('Case01.csv');
AUTRES_Plot(Case01,'P.BPS_2');
```

Alternatively, instead of explicitly loading the time response pass the AUTRES_Plot function the name of the time-response file:

```
AUTRES_Plot('Case01.csv','P.BPS_2');
```

Example 2: (Overlay the responses of multiple variables from a single study case)
Plot the rotor-speeds (W) of each generator ('HPS_1', 'BPS_2', etc.) from a single time response (Case01).

```
Case01 = AUTRES_Load('Case01.csv');
gens = {'HPS_1', 'BPS_2', 'EPS_2', 'VPS_2', 'MPS_2', 'LPS_3', 'YPS_3', ...
        'TPS_4', 'CPS_4', 'SPS_4', 'GPS_4', 'NPS_5', 'TPS_5', 'PPS_5' };
vars = {};
for i = 1:length(gens)
   vars{i} = ['W.' gens{i}];
end
AUTRES_Plot(Case01,vars, ...
         'title', {['Simplified 14 generator model of the South East', ...
                    'Australian system.'], ...
                   ['Case 1, Rotor-speeds of all 14 generators']});
```


© The University of Adelaide

Example 3: (Overlay the responses of a single variable from multiple study cases)
Plot the perturbation in electrical-power output (P) of generator 'NPS_5' due to a +0.5% step increase in the voltage-reference input to the generator obtained with PSS/E ('IEEE_-Case01_VrefStep_NPS_5_PSSE.csv') and Mudpack ('IEEE_Case01_Vref-Step_NPS_5_MP.csv').

```
mudpack = AUTRES_Load('IEEE_Case01_VrefStep_NPS_5_MP.csv');
psse = AUTRES_Load('IEEE_Case01_VrefStep_NPS_5_PSSE.csv');

AUTRES_Plot({psse, mudpack}, 'P.NPS_5', ...
   'title', {['Simplified 14 generator model of the South East ', ...
              'Australian system.'], ...
             ['Case 1, Benchmark comparison of PSS/E & Mudpack, ', ...
              '0.5% step on Vref.NPS_5']});
```

Example 4: (Overlay the responses of multiple variables from multiple study cases)
Plot the terminal voltages of the five SVCs following two-phase to ground faults applied at buses 507 ('IEEE_NetworkBusFaultStudy_Case01_F507_NoTrip.csv') and 315 ('IEEE_NetworkBusFaultStudy_Case01_F315_NoTrip.csv') in study case 1. The faults are cleared without disconnection of any network elements.

```
filelist = { 'IEEE_NetworkBusFaultStudy_Case01_F507_NoTrip.csv', ...
             'IEEE_NetworkBusFaultStudy_Case01_F315_NoTrip.csv' }
for i = 1:length(filelist)
   studies{i} = AUTRES_Load(filelist{i});
end
svcs = { 'ASVC_2', 'RSVC_3', 'BSVC_4', 'PSVC_5', 'SSVC_5' };
vars = {};
for i = 1:length(svcs)
   vars{i} = ['Vt.' svcs{i}];
end
AUTRES_Plot(studies,vars, ...
   'title', {['Simplified 14 generator model of the South East', ...
              'Australian system.'], ...
             ['Case 1, Compare two-phase to ground faults on bus ', ...
              '315 & 507']});
```

2.5 Script AUTRES_Demo

This script provides working examples showing the application of the above Matlab tools. Refer to the comments in the mfile AUTRES_Demo.m.

To run the examples, install the tools as described in Section 3. Then, in Matlab, change to the folder <roor>\AUTRES and enter AUTRES_Demo at the Matlab command prompt. (Note: <root> is the installation folder.)

3 Installation

The plotting tools are contained in a zip archive file 'AUTimeSeriesTools_Ver01.zip'. Unzip the archive to a folder of your choice, being sure to preserve the folder-structure of the archive. Your installation folder will be referred to as <root> in the following.

In Matlab include the folder <root> on your Matlab path. There are a variety of ways to do this, including the following.

Enter the following sequence of commands at the Matlab prompt:

```
% Prepend the installation folder <root> of the plotting tools to
% the Matlab path.
addpath('<root>\AUTRES');
% Save the updated path so the plotting tools are accessible in future
% Matlab sessions.
savepath
```

If you wish to remove the plotting tools from your Matlab search path then enter

```
rmpath('<root>\AUTRES');
```

4 References

[1] M. J. Gibbard and D. J. Vowles, “Simplified 14-Generator Model of the South East Australian Power System: (Including implementations in Mudpack for small-signal analysis and PSS/E for transient-stability analysis)”, Version 4, Power System Dynamics Group, School of Electrical and Electronic Engineering, The University of Adelaide, 18 February 2014.

[2] *PSS/E Version 32, Program Operation Manual:* Siemens, June 2009.
Siemens-PTI PSS/E web address:
http://w3.usa.siemens.com/smartgrid/us/en/transmission-grid/products/grid-analysis-tools/transmission-system-planning/Pages/transmission-system-planning.aspx

[3] Vowles, D.J. and Gibbard, M.J., “Mudpack - a software package for the analysis of the small-signal dynamic performance and control of large electric power systems”, School of Electrical & Electronic Engineering, The University of Adelaide, South Australia.

[4] The Mathworks, Inc. (2010, 23 August 2010). MATLAB®. Available: www.mathworks.com.

© The University of Adelaide