RAMSES User Guide (v3.37)

Petros Aristidou

July 1, 2016

Contents

Chapter 1

Introduction

RAMSES is a power system dynamic simulation software developed at the University of Liege. The Differential-Algebraic Equation systems deriving from the time-domain simulations of complex power system models, are solved using Domain-Decomposition Methods to accelerate the procedure numerically and computationally.

More information on the algorithms, methodology and performance of can be found in [1–5].

Chapter 2

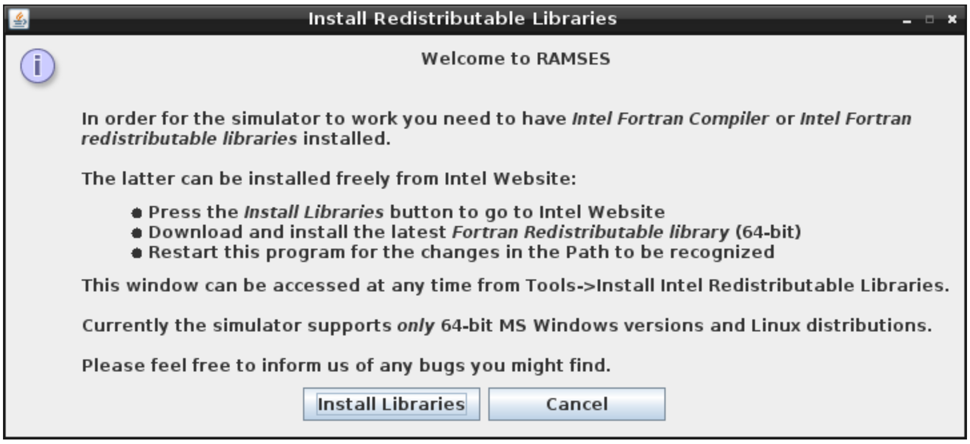
Installation

2.1 Basic Installation

Prerequisites:

- MS Windows **64-bit** OS (XP, VISTA and 7 have been tested but it should work on other versions).
- Linux **64-bit** OS (Debian 6 and 7, Ubuntu 12.04 and scientific linux have been tested but it should work on other distros)
- Java JRE is needed for the graphic user interface. This can be downloaded and installed freely here.

The first time you execute ramses.jar, you will be welcomed by the message to accept the license followed by a message to install some necessary libraries:



If you already have Intel libraries on your computer you can ignore this, otherwise, pressing *Install Libraries* will redirect you to the appropriate website of Intel to download them for free. This window can be accessed later on from *Tools->Install Intel Redistributable Libraries*.

2.1.1 Linux Specific Issues

Under linux, you also need to install:

- *gnuplot*: for the graphic representation of curves
- *wine*: so that notepad++ can be executed to modify the data files
- *gfortran-multilib, blas* and *lapack*: for the numerical libraries needed

Under Debian-like systems, this can be done with:

```
sudo apt-get install gnuplot wine gfortran-multilib libblas3 liblapack3
```

2.2 MATLAB Support

MATLAB is a well known MathWorks software which can be used with RAMSES to provide some *extra* features. MATLAB is not provided with RAMSES. The developers of RAMSES and the University of Liege are not affiliated to MATLAB.

The following of this text assumes you already have MATLAB installed on your computer. It uses matlab engine to establish a connection between MATLAB and RAMSES.

2.2.1 Linux

Matlab should be installed and able to be accessed from command line. To check this, one can open a command-line window and execute *matlab*, this should start a matlab window.

Following, some necessary libraries need to be added to the library path. For example, under Debian you have to modify the *.bashrc* file (i.e. */etc/bash.bashrc*) to include the libraries:

```
export LD_LIBRARY_PATH=/usr/local/MATLAB/R2012b/bin/glnxa64:\
/usr/local/MATLAB/R2012b/sys/os/glnxa64:$LD_LIBRARY_PATH
```

Finally, the matlab scripts that will be accessed, need to be in the path of matlab. Putting them in the same directory as the executable doesn't count. The file pathdef.m has to be modified. For example:

```
p3tris@Odysseus:~$ cat /usr/local/MATLAB/R2012b/toolbox/local/pathdef.m
function p = pathdef %PATHDEF Search path defaults.
% PATHDEF returns a string that can be used as input to MATLABPATH
% in order to set the path.
% Copyright 1984-2007 The MathWorks, Inc.
% $Revision:  1.4.2.2 $
$Date:  2007/06/07 14:45:14 $
% DO NOT MODIFY THIS FILE. IT IS AN AUTOGENERATED FILE.
% EDITING MAY CAUSE THE FILE TO BECOME UNREADABLE TO
% THE PATHTOOL AND THE INSTALLER.
p = [...
%%% BEGIN ENTRIES %%%
'/home/p3tris/Documents/MATLAB:',...
matlabroot,'/toolbox/matlab/codetools:',...
matlabroot,'/toolbox/matlab/datafun:', ...
matlabroot,'/toolbox/matlab/datamanager:', ...
matlabroot,'/toolbox/matlab/datatypes:', ...
matlabroot,'/toolbox/matlab/elfun:', ...
matlabroot,'/toolbox/matlab/elmat:', ...
...
```

Of course, the MATLAB version (here `R2012b) as well as all the paths need to be modified to match your own installation.`

2.2.2 Windows

On Windows systems, you need to do the following:

- **<u>Setting Run-Time Library Path on Windows:</u>**
  Set the Path environment variable to the path string returned by the following MATLAB command:

  ```
  fullfile(matlabroot,'bin',computer('arch'))
  ```

  To set an environment variable on Windows XP, select Start > Settings > Control Panel > System. The System Properties dialog box appears. Click the Advanced tab, and then click the Environment Variables button.
  In the System variables panel scroll down until you find the Path variable. Click this variable to highlight it, and then click the Edit button to open the Edit System Variable dialog box. At the end of the path string, enter a semicolon. Then, enter the path string that MATLAB returns after

evaluating the expression shown above. Click OK in the Edit System Variable dialog box, and in all remaining dialog boxes. Check here for explanations how to add directories to windows path.

- **Registering MATLAB Software as a COM Server:**
  To run the engine application on a Windows operating system, you need to register MATLAB as a COM server. Do this for every session, to ensure that the current version of MATLAB is the registered version. If you run older versions, the registered version could change. If there is a mismatch of version numbers, MATLAB displays Can't start MATLAB engine. To manually register MATLAB as a server, type:

```
cd(fullfile(matlabroot,'bin',computer('arch')))
!matlab /regserver
```

  Close the MATLAB window that appears. Check here for more information.

- **Add the matlab scripts used to the path of matlab:**
  Putting them in the same directory as the executable doesn't count. This can be done by changing the path file in *File->Set Path* (see Fig. 2.1).

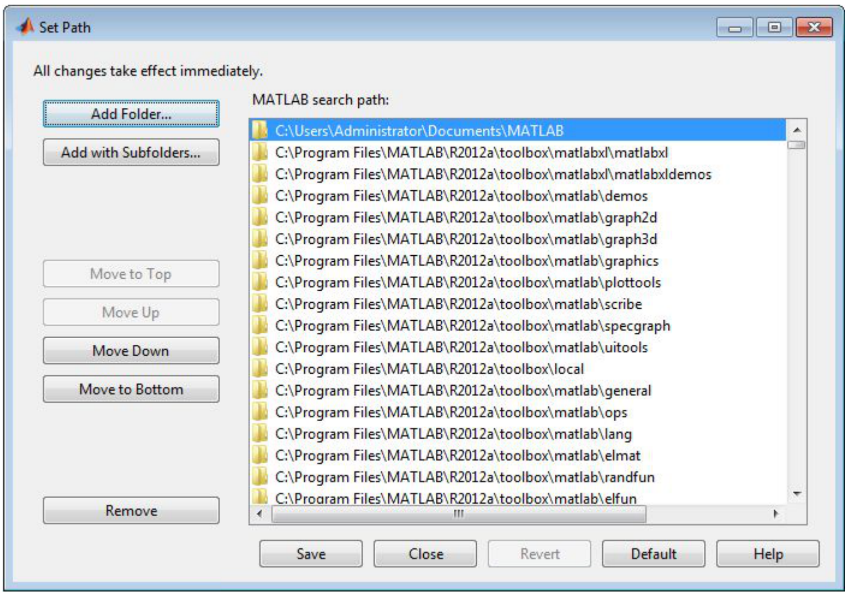


Figure 2.1: Path file in Windows 7

You have to restart your computer before these changes take effect.

Chapter 3

System Data

3.1 Network

3.1.1 AC BUS

```
BUS BUS_NAME NOMINAL_VOLTAGE[kV] ;
```

If more parameters are given, they are ignored. Those extra parameters are used in the load-flow computation.

3.1.2 DC BUS

```
BUSDC BUS_NAME NOMINAL_VOLTAGE[kV] ;
```

3.1.3 AC LINE

```
LINE LINE_NAME BUS_NAME1 BUS_NAME2 R X WC/2 SNOM BR1 BR2 ;
```

See also artere model data.

3.1.4 SWITCH

```
SWITCH SWITCH_NAME BUS1 BUS2 BR ;
```

3.1.5 SHUNT

```
SHUNT SHUNTNAME BUS_NAME X BR ;
```

3.1.6 Transformer (TRANSFO)

```
TRANSFO TRANSFO_NAME BUS_NAME1 BUS_NAME2 R X B1 B2 N PHI SNOM BR ;
```

See also artere model data.

3.1.7 NRPI

3.1.8 Distribution Transformer

```
DIST_TRANSFO MV_BUS_NAME HV_BUS_NAME FP FQ P Q VNOM RES REAC BMAG
PHASE SNOM VDES ;
```

3.1.9 Initial Load Flow Values

```
LFRESV BUS_NAME VOLTAGE(p.u.) PHASE(rad) ;
```

3.1.10 Zones

```
ZONE ZONE_NAME BUS_NAME_1 BUS_NAME_2 ... BUS_NAME_N ;
```

3.2 Discrete Controllers

3.2.1 LTC

```
DCTL LTC CTLNAME TRFONAME BUS_NAME DIR NMIN NMAX NBPOS TOL DELAY1 DE-
LAY2 ;
```

3.2.2 Real-time synchronizer

```
DCTL RT CTLNAME ratio_to_rt ;
```

If ratio_to_rt is set to 1.0, the simulation will be slowed down when it's faster than RT to synchronize. The moments it is slower, nothing will be done. Setting ratio_to_rt to 2.0, means twice faster than RT (if possible), etc.

3.2.3 MATLAB Controller

Check DCTL Matlab manual.

3.3 LOADS

3.3.1 Exponential Load

```
LOAD injname BUS_NAME FP FQ P Q DP A1 alpha1 A2 alpha2 alpha3 DQ B1 beta1 B2 beta2
beta3 ;
```

3.3.2 Impedance Load

```
IMPLOAD loadname BUS_NAME FP FQ P Q ;
```

3.4 Synchronous Machines and Controls

3.4.1 Synchronous Machine

```
SYNC_MACH Sync_Mach_Name BUS_NAME FP FQ P Q SNOM Pnom H D IBRATIO XT/RL
Xl Xd X'd X"d Xq X'q X"q m n Ra T'do T"do T'qo T"qo
EXC GENERIC1 IFLIM d f S K1 K2 L1 L2 G TA TB TE L3 L4 SPEEDIN KPSS Tw T1 T2 T3 T4
DVMIN DVMAX
TOR 1STORDER FHP TR SIGMA ;
or
TOR HYDROGENERIC1 SIGMA TP Qv KP KI TSM LIMZDOT TW ;
```

3.4.2 Exciters

3.4.2.1 CONSTANT

3.4.2.2 1ST_ORDER

3.4.2.3 GENERIC1

3.4.2.4 GENERIC2

3.4.2.5 GENERIC3

3.4.2.6 4LOOPPSS

3.4.2.7 AVR+PSS

3.4.2.8 AVR+PSST

3.4.3 TOR

3.4.3.1 CONSTANT

3.4.3.2 1ST_ORDER

3.4.3.3 HYDRO_GENERIC1

3.4.3.4 GOVHYDR

3.4.3.5 GOVNUC

3.4.3.6 GOVCLASM

3.4.3.7 GASTURBM

3.4.3.8 THERMAL GENERIC1

Check TOR THERMAL GENERIC1.

3.5 Injectors

3.5.1 Induction Machine

3.5.2 Double Cage Induction Machine

```
INJEC INDMACH2 NAME BUS_NAME FP FQ P Q Snom R1 L1 Lm R2 L2 R3 L3 H A B LF ;
```

where LF is "Load Factor".

3.5.3 Restorative Load

3.5.4 LOAD

See 3.3.1.

3.5.5 COMPLD

3.5.6 LINK (or two-port injector)

```
LINK LINK_NAME BUS_NAME1 BUS_NAME2 SYNCHRONIZATION ;
```

3.5.7 Wind Turbine Type 1

```
INJEC wt1 NAME BUS_NAME FP FQ P Q Snom Pnom Rs Lls Lsr Rr Llr Hg Ht Dtg Ktg p Kw
Kdroop Kp Ki pimax pimin Tpe T1 T2 LF ;
```

Check INJ WT1 manual.

3.5.8 Wind Turbine Type 2

```
INJEC wt2 INJEC_NAME BUS_NAME FP FQ P Q Snom Pnom Rs Lls Lsr Rr Llr Hg Ht Dtg Ktg
p Kw Kdroop Kp Ki pimax pimin Tpe T1 T2 Tw Kpe Kip Kpp Rmax Rmin LF ;
```

Check INJ WT2 manual.

3.5.9 Wind Turbine Type 3

```
INJEC wt3 INJEC_NAME BUS_NAME FP FQ P Q Snom nturb Pnom Xeq Hg Ht Dtg Ktg R ratio
p rho beta0 bmin bmax b.min b.max Tp Kpc Kic Kpp Kip Tpc Kptrq Kitrq Tw KVi KQp KQi XIQmin
XIQmax Vmin Vmax Qmin Qmax Tc Tv Tr Kpv Kiv Kpll theta.min theta.max Tcon Rcomp Xcomp
wmin wmax pmin pmax p.min p.max mode ;
```

Check INJ WT3 manual.

3.5.10 Wind Turbine Type 4

```
INJEC wt4 INJEC_NAME BUS_NAME FP FQ P Q Snom nturb Pnom Ht R ratio poles rho beta0
betamin betamax betadotmin betadotmax Tp Kpc Kic Kpp Kip Tpc Kptrq Kitrq Tw KVi KQp KQi
Vmin Vmax Iqhl Iphl Imaxtd pqflag Qmin Qmax Tc Tv Tr Kpv Kiv Kdbr Ebst Kpll thetadotmin
thetadotmax Tcon Rcomp Xcomp wmin wmax Pmin Pmax Pdotmin Pdotmax mode ;
```

Check INJ WT4 manual.

3.5.11 Infinite Bus

```
INJEC THEVEQ INJEC_NAME BUS_NAME FP FQ P Q MVA ;
```

3.6 Two-port Injectors

3.6.1 DC Line

3.6.2 HVDC Link

3.6.3 AC-DC Link

Chapter 4

Disturbance Description

Disturbances need to have a continuity.

4.1 Continue Solver

```
time(s) CONTINUE SOLVER disc_meth max_h(s) min_h(s) latency(pu) upd_over
```

Discretization method (disc_meth):

- TR: Trapezoidal
- BE: Backward Euler
- BD: BDF2

Jacobian update override (upd_over):

- ALL: Update all injectors and network
- NET: Update only network
- ABL: Update only injectors
- IBL: Update all injectors and network
- NOT: Do not override

It is mainly used to modify the settings of the solver and has to exist at the first line. For example:

```
0.000 CONTINUE SOLVER BD 0.0200 0.001 0. ALL
```

4.2 Set stopping criteria for voltage

Define in the data files the following record:

```
DCTL SIM_MINMAXVOLT CTRL_Name VMAX(pu) VMIN(pu) DEADTIME(s)
Stop_Simulation(T/F) ;
```

Check VMIN VMAX.

4.3 Set stopping criteria for machine speed

Define in the data files the following record:

```
DCTL SIM_MINMAXVOLT CTRL_Name MAX_SYNC_SPEED(pu) MIN_SYNC_SPEED(pu)
DEADTIME(s) Stop_Simulation(T/F) ;
```

Check SpeedMIN SpeedMAX.

4.4 Stop

```
time(s) STOP
```

It is used to signal the end of the simulation and has to exist at the last line.

For example:

```
100.000 STOP
```

4.5 Trip line

```
time(s) BREAKER BRANCH name_of_line orig_break(0/1) extrem_break(0/1)
```

Used to open/close the breakers of a line.

For example, opening both ends of a line:

```
10.000 BREAKER BRANCH 1044-4032 0 0
```

4.6 Trip synchronous machine / injector

```
time(s) BREAKER SYNC_MACH/INJ name_of_synch_mach/name_of_injector breaker(0/1)
```

Used to open/close the breaker (trip) of a synchronous machine or injector.

For example:

```
10.000 BREAKER INJ L_11 0
```

4.7 Three phase short-circuit (with use of resistance to ground)

This demands two commands:

```
time(s) FAULT BUS name_of_bus rfault [xfault]
time(s) CLEAR BUS name_of_bus
```

The first line is used to declare the starting of the short-circuit. The fault has a resistance of *rfault+j*xfault* to the ground where the values of rfault and xfault are in Ohm. If xfault is not defined, a fully resistive fault is assumed.

Example of a 100ms short-circuit directly to ground:

```
10.000 FAULT BUS 1044 0. 0.
10.100 CLEAR BUS 1044
```

4.8 Three phase short-circuit (with use of voltage reached after fault)

This demands two commands:

```
time(s) VFAULT BUS name_of_bus Voltage_reached_after_fault
time(s) CLEAR BUS name_of_bus
```

The first line is used to declare the starting of the short-circuit. The fault has an unknown resistance *(j*xfault)* to the ground. Based on the value of the voltage reached after the fault (declared in pu), xfault is computed and used for simulating the fault.

Example of a 100ms short-circuit where the voltage at the faulted bus reached 0.5 pu after the fault:

```
10.000 VFAULT BUS 1044 0.5
10.100 CLEAR BUS 1044
```

Supported: Version 3.13 and above.

4.9 Change parameters

Used to change the parameters of a model.

4.9.1 BRANCH

```
time(s) CHGPRM BRANCH name_of_line MAGN/PHAN ±increment
```

4.9.2 SHUNT

```
time(s) CHGPRM SHUNT name_of_shunt QNOM ±increment
```

The increment should be expressed in MVAr and it is per unitized internally by RAMSES.

4.9.3 EXC

```
time(s) CHGPRM EXC name_of_equipment name_of_parameter ±increment (MVAr/%) duration(s)
```

The units is not obligatory. If nothing is given then the parameter is modified in absolute value. If MVAr is given, then the increment is per unitized using *Snom* of the machine before applying the change. If % is given, then the parameter is changed as a percentage of the original value. If $duration = 0$ then a step change is applied, otherwise the change is applied as a ramp over the given duration (in seconds).

For example:

```
10.000 CHGPRM EXC g1 V0 +10 % 10
```

This means the parameter V0 of the exciter of synchronous machine g1 is ramped by +10% between 10 and 20 seconds.

4.9.4 TOR

```
time(s) CHGPRM TOR name_of_equipment name_of_parameter ±increment (MW/%) duration(s)
```

The units is not obligatory. If nothing is given then the parameter is modified in absolute value. If MW is given, then the increment is per unitized using *Pnom* of the machine before applying the change. If % is given, then the parameter is changed as a percentage of the original value. If $duration = 0$ then a step change is applied, otherwise the change is applied as a ramp over the given duration (in seconds).

For example:

```
10.000 CHGPRM TOR g1 P0 +1 MW 10
```

This means the parameter P0 of the torque controller of synchronous machine g1 is ramped by +1 MW between 10 and 20 seconds.

4.9.5 INJ/TWOP/DCTL

```
time(s) CHGPRM INJ/TWOP/DCTL name_of_equipment name_of_parameter ±increment (MW/MVAr/%/SETP) duration(s)
```

The units is not obligatory. If nothing is given then the parameter is modified in absolute value. If MW or MVAr is given, then the increment is per unitized using system's *Sbase* before applying the change. If % is given, then the parameter is changed as a percentage of the original value. IF SETP is given then the value increment is actually the new setpoint. If $duration = 0$ then a step change is applied, otherwise the change is applied as a ramp over the given duration (in seconds).

For example:

```
10.000 CHGPRM INJ L_11 P0 +50 % 60
10.000 CHGPRM INJ L_11 Q0 +30 % 60
```

This means the parameter P0 (resp. Q0) of the injector L_11 is ramped by +50% (resp. 30%) between 10 and 70 seconds. In this case, if L_11 is a load model, it can be used to simulate a load increase.

4.10 Export Jacobian matrix

```
time(s) JAC 'name_of_filename'
```

Also, make sure to add these to your settings:

```
$OMEGA_REF SYN ;
$SCHEME IN;
```

4.11 Export load flow

```
time(s) LFRESV 'name_of_filename'
```

Chapter 5

Solver Settings

5.1 Sampling time for observed variables

```
$PLOT_STEP time(s) ;
```

5.2 Display Profiling Results

```
$DISP_PROF T/F ;
```

5.3 Run-time observables refresh interval

```
$GP_REFRESH_RATE time_interval(s) ;
```

5.4 Time constant of load restoration

```
$T_LOAD_REST time(s) ;
```

5.5 Omega Reference

```
$OMEGA_REF SYN/COI ;
```

Synchronous reference frame or center of inertia.

5.6 Maximum Fault Value

```
$MAX_FAULT value ;
```

5.7 Base Power

```
$S_BASE BASE(MVA) ;
```

5.8 Nominal Frequency

```
FNOM Frequency(Hz) ;
```

5.9 Newton Tolerance

```
$NEWTON_TOLER NETWORK_TOLERANCE INJ_RELATIVE_TOLERANCE INJ_ABSOLUTE_TOLERANCE ;
```

Set's the solver Newton iteration tolerance for stopping. Default values are: 1e-03, 5e-04, 5e-04.

5.10 Finite Difference Values

```
$FIN_DIFFER proportional_value absolute_value ;
```

Values used to calculate numerically Jacobian matrices of injectors.

5.11 Full Jacobian Update

```
$FULL_UPDATE T/F ;
```

Disable partial Jacobian updates.

5.12 Skip Converged Blocks

```
$SKIP_CONV T/F ;
```

Activate/Deactivate stopping to solve converged injectors.

5.13 Latency Tolerance

```
$LATENCY OBS_TIME_WINDOW(s) EARLY_STOP(T/F) ;
```

5.14 Solution Scheme

```
$SCHEME DE/IN ;
```

Integrated or Decomposed solution scheme.

5.15 Number of Threads for parallel computing

```
$NB_THREADS Number ;
```

5.16 Way of injector distribution over parallel threads

```
$OMP STA/DYN/GUI chunk ;
```

STA is for static assignment (better for NUMA architecture computers), DYN is for dynamic assignment (better for UMA architecture computers) and GUI is for guided. Chunk is the number of consecutive injectors assigned to each thread.

5.17 Update network elements with frequency

```
$NET_FREQ_UPD T/F ;
```

Check Network update with frequency.

Bibliography

[1] P. Aristidou and T. Van Cutsem, "Dynamic simulations of combined transmission and distribution systems using decomposition and localization," in *Proc. of IEEE PES PowerTech Conference*, June 2013. [Online]. Available: http://hdl.handle.net/2268/145092

[2] P. Aristidou, D. Fabozzi, and T. Van Cutsem, "Dynamic simulation of large-scale power systems using a parallel schur-complement-based decomposition method," *IEEE Transactions on Parallel and Distributed Systems*, October 2013. [Online]. Available: http://hdl.handle.net/2268/156230

[3] ——, "Exploiting localization for faster power system dynamic simulations," in *Proc. of IEEE PES PowerTech Conference*, June 2013. [Online]. Available: http://hdl.handle.net/2268/145093

[4] ——, "A schur complement method for dae systems in power system dynamic simulations," in *Lecture Notes in Computational Science and Engineering*. Springer, June 2013. [Online]. Available: http://hdl.handle.net/2268/154312

[5] P. Aristidou, D. Fabozzi, and T. Van Cutsem, "A schur complement method for dae systems in power system simulation (presentation)," in *The Twenty First International Conference on Domain Decomposition Methods*, June 2012. [Online]. Available: http://hdl.handle.net/2268/126725