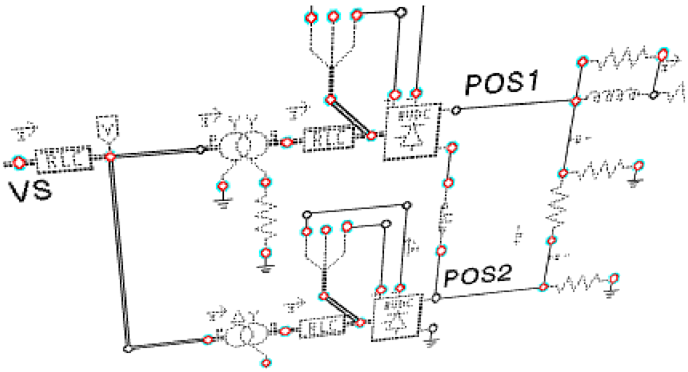
ATPDRAW

version 3.5
for Windows 9x/NT/2000/XP

Users' Manual



László Prikler,

Hans Kristian Høidalen

ATPDraw™
for Windows
3.5

Preliminary Release No. 1.1
October 2002

PREFACE

This Users' Manual documents all main features of ATPDraw version 3.5. The manuscript is prepared by László Prikler at SYSTRAN Engineering Services Ltd. in Budapest, and based on the previous Users' Manual for ATPDraw version 1.0, SINTEF TR A4790, dated November 1998. The Reference Manual gives a summary of menu items and menu options. The Advanced Manual covers the new features Grouping, $Parameter, line/cable-, and transformer modeling. Finally the Application Manual is extended with several new examples.

New ATPDraw users are advised to start with the Installation and Introductory manuals.

ATPDraw is developed by SINTEF Energy Research (SEfAS). Program and documentation development have been financed by Bonneville Power Administration, USA.

For SINTEF Energy Research Trondheim, Norway 7th August 2002

Hans Kr. Høidalen
project manager

SUMMARY

ATPDraw is a graphical, mouse-driven preprocessor to the ATP version of the Electromagnetic Transients Program (EMTP) on the MS-Windows platform. The program is written in Borland Delphi 2.0 and runs under Windows 9x/NT/2000/XP. In ATPDraw the user can construct an electrical circuit using the mouse and selecting components from menus, then ATPDraw generates the ATP input file in the appropriate format based on "what you see is what you get". The simulation program ATP and plotting programs can be integrated with ATPDraw.

ATPDraw supports multiple circuit modeling that makes possible to work on more circuits simultaneously and copy information between the circuits. All kinds of standard circuit editing facilities (copy/paste, grouping, rotate, export/import, undo/redo) are available. In addition, ATPDraw supports the Windows clipboard and metafile export. The circuit is stored on disk in a single project file, which includes all the simulation objects and options needed to run the case. The project file is in zip-compressed format that makes the file sharing with others very simple.

Most of the standard components of ATP (both single and 3-phase), as well as TACS are supported, and in addition the user can create new objects based on MODELS or $Include (Data Base Module). Line/Cable modeling (KCLee, PI-equivalent, Semlyen, JMarti and Noda) is also included in ATPDraw where the user specifies the geometry and material data and has the option to view the cross section graphically and verify the model in the frequency domain. Objects for Harmonic Frequency Scan (HFS) have also been added. Special components support the user in machine and transformer modeling based on the powerful Universal Machine and BCTRAN components in ATP-EMTP.

ATPDraw supports hierarchical modeling by replacing selected group of objects with a single icon in unlimited numbers of layers. POCKET CALCULATOR and $PARAMETER features of ATP is also supported, allowing the user to specify a text string as input in a components' data field, then assign numerical values to these texts strings later.

TABLE OF CONTENTS

Page

SINTEF

1. Introduction . . .

. . . .

ATPDraw™
for Windows
3.5

1.1 What is ATPDraw?

ATPDraw™ for Windows is a graphical, mouse-driven preprocessor to the ATP version of the Electromagnetic Transients Program (EMTP). In ATPDraw the user can construct the digital model of the circuit to be simulated using the mouse and selecting predefined components from an extensive palette, interactively. Then ATPDraw generates the input file for the ATP simulation in the appropriate format based on "what you see is what you get". Circuit node naming is administrated by ATPDraw, thus the user needs to give a name only to nodes having special interest.

ATPDraw has a standard Windows layout and offers a large Windows help file system. All kinds of standard circuit editing facilities (copy/paste, grouping, rotate, export/import, undo/redo) are available. Other facilities in ATPDraw are: built-in editor for ATP-file editing, text viewer for displaying the output LIS-file of ATP, automatic LIS-file checking with special trigger strings to detect simulation errors, support of Windows clipboard and metafile export. ATPDraw supports multiple circuit modeling that makes possible to work on more circuits simultaneously and copy information between the circuits.

Most of the standard components of ATP (both single and 3-phase), as well as TACS are supported, and in addition the user can create new objects based on MODELS or $INCLUDE (Data Base Module). Line/Cable modeling (KCLee, PI-equivalent, Semlyen, JMarti and Noda) is also included in ATPDraw where the user specifies the geometry and material data and has the option to view the cross section graphically and verify the model in the frequency domain. Objects for Harmonic Frequency Scan (HFS) have also been added. Special objects help the user in machine and transformer modeling including the powerful UNIVERSAL MACHINE and BCTRAN features of ATP.

ATPDraw supports hierarchical modeling to replace a selected group of objects with a single icon in unlimited numbers of layers. $PARAMETER feature of ATP is also implemented, allowing the user to specify a text string as input in a components' data field, then assign numerical values to these texts strings later. The circuit is stored on disk in a single project file, which includes all the simulation objects and options needed to run the case. The project file is in zip-compressed format that makes the file sharing with others very simple.

ATPDraw is most valuable to new users of ATP-EMTP and is an excellent tool for educational purposes. However, the possibility of multi-layer modeling makes ATPDraw a powerful front-end processor for professionals in analysis of electric power system transients, as well.

Version 2.0 and above of ATPDraw for 9x/NTx/2000/XP Windows platforms are written in Borland Delphi 2.0. Development of 16-bit versions for Windows 3.1/95 stopped with the launching of ATPDraw 1.0. A version 3.6 compiled with Delphi 6.0 is also available.

ATPDraw™ is a trademark and copyrighted by © 1998-2001 SINTEF Energy Research, Norway. Program developer is Dr. Hans Kristian Høidalen at SINTEF Energy Research in Trondheim, Norway, with Dahl Data Design in Norway as a programming sub-contractor and SYSTRAN Engineering Services in Hungary as a sub-contractor for program documentation. Program development has been financed by Bonneville Power Administration in Portland, Oregon, USA, with Pacific Engineering Corporation as project coordinator.

The ATPDraw program is royalty free and can be downloaded free of charge from several Internet sites. The on-line help of ATPDraw and the present program documentation includes third-party proprietary information of, thus *ATP licensing is mandatory* prior to get permission to download the program and documentation from the Internet, or to receive ATP related materials from others.

1.2 What is ATP?

The Alternative Transients Program (ATP) is considered to be one of the most widely used universal program system for digital simulation of transient phenomena of electromagnetic as well as electromechanical nature in electric power systems. With this digital program, complex networks and control systems of arbitrary structure can be simulated. ATP has extensive modeling capabilities and additional important features besides the computation of transients.

The Electromagnetic Transients Program (EMTP) was developed in the public domain at the Bonneville Power Administration (BPA) of Portland, Oregon prior to the commercial initiative in 1984 by the EMTP Development Coordination Group and the Electric Power Research Institute (EPRI) of Palo Alto, California. The birth of ATP dates to early in 1984, when Drs. Meyer and Liu did not approve of proposed commercialization of BPA's EMTP and Dr. Meyer, using his own personal time, started a new program from a copy of BPA's public-domain EMTP. Since then the ATP program has been continuously developed through international contributions by Drs. W. Scott Meyer and Tsu-huei Liu, the co-Chairmen of the Canadian/American EMTP User Group. Several experts around the world have been contributing to EMTP starting in 1975 and later to ATP in close cooperation with program developers in Portland, USA.

Whereas BPA work on EMTP remains in the public domain by U.S. law, ATP is *not* in the public domain and licensing is required before access to proprietary materials is granted. Licensing is, however, available free of all charge to anyone in the world who has not participated voluntarily in the sale or attempted sale of any electromagnetic transients program, (hereafter called "EMTP commerce").

1.3 Operating priciples and capabilities of ATP[1]

The ATP program predicts variables of interest within electric power networks as functions of time, typically initiated by some disturbances. Basically, trapezoidal rule of integration is used to solve the differential equations of system components in the time domain. Non-zero initial conditions can be determined either automatically by a steady-state phasor solution or they can be entered by the user for simpler components.

ATP has many models including rotating machines, transformers, surge arresters, transmission lines and cables. Interfacing capability to the program modules TACS (Transient Analysis of Control Systems) and MODELS (a simulation language) enables modeling of control systems and components with nonlinear characteristics such as arcs and corona. Dynamic systems without any electrical network can also be simulated using TACS and MODELS control system modeling.

Symmetrical or unsymmetrical disturbances are allowed, such as faults, lightning surges and several kind of switching operations including commutation of valves. Frequency-domain harmonic analysis using harmonic current injection method (HARMONIC FREQUENCY SCAN)

[1] Source: WWW.EMTP.ORG

and calculation of the frequency response of phasor networks using FREQUENCY SCAN feature is also supported. The model-library of ATP at present consists of the following components:

- Uncoupled and coupled linear, lumped R,L,C elements.
- Transmission lines and cables with distributed and frequency-dependent parameters.
- Nonlinear resistances and inductances, hysteretic inductor, time-varying resistance, TACS/MODELS controlled resistance.
- Components with nonlinearities: transformers including saturation and hysteresis, surge arresters (gapless and with gap), arcs.
- Ordinary switches, time-dependent and voltage-dependent switches, statistical switching (Monte-Carlo studies).
- Valves (diodes, thyristors, triacs), TACS/MODELS controlled switches.
- Analytical sources: step, ramp, sinusoidal, exponential surge functions, TACS/MODELS defined sources.
- Rotating machines: 3-phase synchronous machine, universal machine model.
- User-defined electrical components that include MODELS interaction

1.3.1 Integrated simulation modules in ATP

MODELS in ATP is a general-purpose description language supported by an extensive set of simulation tools for the representation and study of time-variant systems.

- The description of each model is enabled using free-format, keyword-driven syntax of local context and that is largely self-documenting.
- MODELS in ATP allows the description of arbitrary user-defined control and circuit components, providing a simple interface for connecting other programs/models to ATP.
- As a general-purpose programmable tool, MODELS can be used for processing simulation results either in the frequency domain or in the time domain.

TACS is a simulation module for time-domain analysis of control systems. It was originally developed for the simulation of HVDC converter controls. For TACS, a block diagram representation of control systems is used. TACS can be used for the simulation of

- HVDC converter controls
- Excitation systems of synchronous machines
- power electronics and drives
- electric arcs (circuit breaker and fault arcs).

Interface between electrical network and TACS is established by exchange of signals such as node voltage, switch current, switch status, time-varying resistance, voltage- and current sources.

Supporting routines are integrated utilities inside the program that support the users in conversion between manufacturers' data format and the one required by the program, or to calculate electrical parameters of lines and cables from geometrical and material data. Supporting modules in ATP are:

- Calculation of electrical parameters of overhead lines and cables using program modules LINE CONSTANTS, CABLE CONSTANTS and CABLE PARAMETERS.
- Generation of frequency-dependent line model input data (Semlyen, J.Marti, Noda line models).
- Calculation of model data for transformers (XFORMER, BCTRAN).
- Saturation and hysteresis curve conversion.
- Data Base Modularization (for $INCLUDE usage).

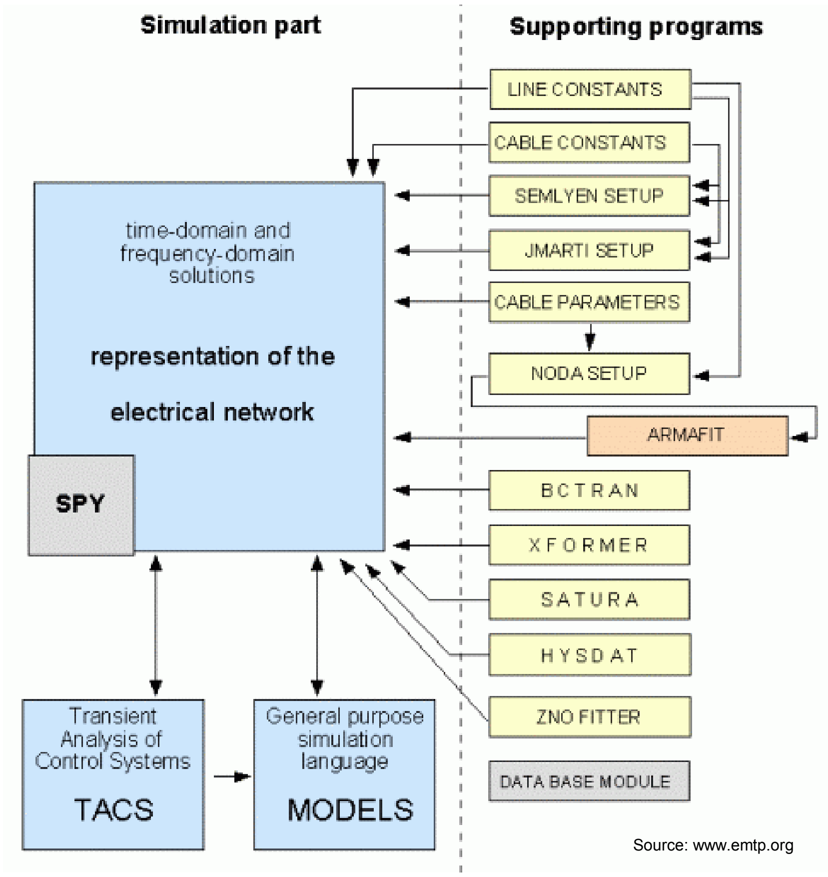


Fig. 1.1 - Supporting routines in ATP.

1.3.2 Program capabilities

ATP-EMTP tables are dimensioned dynamically at the start of execution to satisfy the needs of users and their hardware (e.g., RAM). No absolute limits have ever been observed, and the standard version has limits that average more than 20 times default table sizes. Today, the largest simulations are being performed using Intel-based PC's. The following table shows maximum limits for standard program distribution.

Busses	6000	Sources	900
Branches	10000	Nonlinear elements	2250
Switches	1200	Synchronous machines	90

1.3.3 Main characteristics of plotting programs for ATP

These post-processors are interfaced with ATP via disk files and their main function is to display the results of a time- or frequency domain simulation. ATP simulation data are stored in a file having extension `.pl4`, and it can be processed either off-line, or on-line. The latter (i.e. to display results while the simulation proceeds) is available only if the operating system provides concurrent PL4-file access for ATP and the postprocessor program.

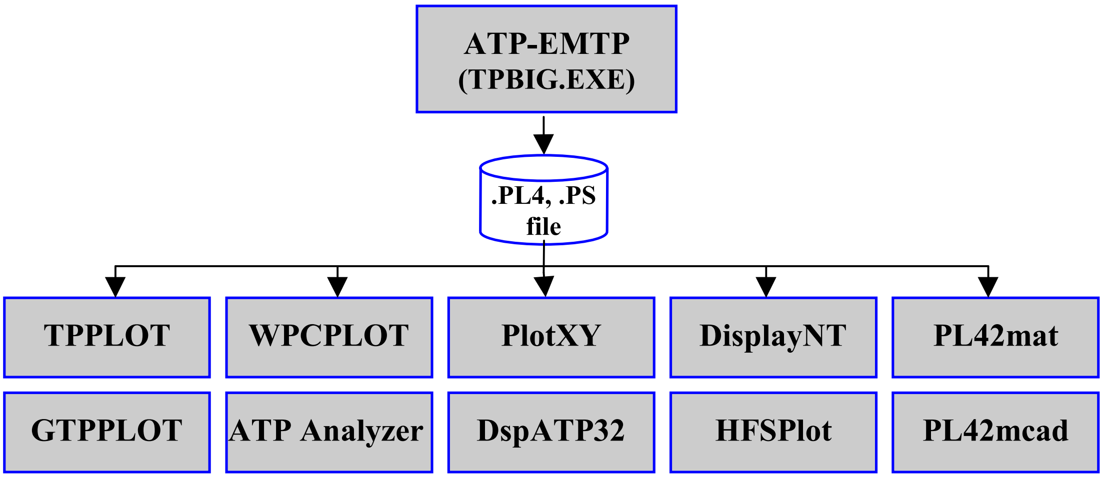


Fig. 1.2 – Plotting programs for ATP.

TPPLOT program has been written under Salford FORTRAN that requires a DOS extender. DBOS is incompatible with NT series of MS-Windows, thus TPPLOT cannot be used under NT, 2000 or XP. TPPLOT detects automatically the PL4-file format and reads almost all file formats created by any PC version of ATP. TPPLOT has a semi-graphical user interface and nearly all commands can be executed by the mouse. The program supports up to 20 curves per plot, plots versus time as well as X-Y plots, factor and offset, automatic or manual axis scaling and labeling. The curves are drawn using solid lines with different colors and user can mark each curve with different characters. Visually redundant data points are eliminated on the screen. Screen plots can be exported in various text-, bitmap or vector graphics file formats including Postscript, HPGL and COMTRADE. The program not only writes but also reads data files in COMTRADE format. Simple mathematical manipulations are also supported, as well as FFT analysis and bar chart plots for harmonics.

Developer: Dr. W. Scott Meyer, canam@emtp.org, USA.
Licensing: Distributed at no cost to the licensed ATP users.
Distribution: EEUG annual CD distribution, EEUG, JAUG secure Web sites.

GTPPLOT is a plotting program for processing PL4 output of ATP. It is compiled with the GNU FORTRAN, and makes use of the graphical package DISLIN. The program is available for DOS-djgpp extender, Windows 32, and Linux. GTPPLOT can read *widenn*, formatted PL4-files (FMTPL4 = 10Fnn.), C-like binary files, unformatted files, COMTRADE and ASCII data files. GTPPLOT is able to process graphics files with up to 1000000 points and up to 1000 variables. The program can plot up to 20 curves end export the graphics in nine different formats: HP-GL, CGM, WMF, PCX, PostScript, PNG, WMF, JAVA and GNUPLOT. For FS and HFS runs the plot can be bar charts. The data can be exported as widenn PL4, COMTRADE, Matlab, MathCad and Mathematica files. Furthermore, the program calculates lot of Power Quality Indexes from data, can be used for FOURIER analysis, turbine shaft loss of life estimation. Various simple math operations with variables, as integration, derivation, RMS, power, energy, I2T are also supported.

GTPPLOT can be used to generate KIZILCAY F-DEPENDENT elements from FREQUENCY SCAN PL4 output, as well. GTPPLOT has no graphical interface, the user must use the keyboard for all the input commands.
Developer: Mr. Orlando P. Hevia, heviaop@ciudad.com.ar, Santa-Fe, Argentina.
Licensing: Distributed at no cost to the licensed ATP users.
Distribution: EEUG annual CD distribution, EEUG, JAUG, MTU secure FTP/Web sites.

PlotXY is a WIN32 plotting program originally designed for ATP-EMTP. The program is mainly designed to make, as easy and fast as possible, line plots in Microsoft Windows environments. It is also able to perform some post-processing on the plotted curves: algebraic operations, computation of the Fourier series coefficients. The program has an easy-to use graphical user interface, and the 32 bit code provides very fast operation. Up to 3 PL4 or ADF files can be simultaneously held in memory for easy comparison of different data and up to 8 curves per plots versus time, or X-Y plots are allowed. The program has a clever automatic axis scaling capability and able to make plots with two independent vertical axes and provides easy tools for factors, offsets and zoom support, and a graphical cursor to see values in numerical format. Screen plots can be exported as Windows Metafile via win32 clipboard.

Developer: Dr. Massimo Ceraolo, ceraolo@dsea.unipi.it, University of Pisa, Italy.
Licensing: "acknowledgeware". Distributed at no cost to the licensed ATP users. If user keeps it beyond the 30-day trial period, he/she must send an acknowledgement letter to the developer.
Distribution: EEUG annual CD distribution, EEUG, JAUG and MTU secure FTP sites.

PCPLOT was steadily developed and improved until 1997 using Borland Turbo Pascal under MS-DOS platforms. The program can read PL4-file types of unformatted, C-like binary and formatted files. PCPLOT can display maximum 4 curves with 16000 plot points per curve. The maximum number of plot variables stored in the plot file is limited up to 100. The program supports three different plot types: time function (results of the simulations), X-Y plot (one variable against another), frequency-response (results of "FREQUENCY SCAN" cases). The values of the plotted variables can be displayed by means of a vertical marker line. Different type of curves (e.g. currents and voltages) can be mixed in the same plot by defining scaling factors and offset. The curves are drawn using solid lines with different colors and user can mark each curve with different characters at the desired positions. Visually redundant data points on plots are eliminated to accelerate the drawing speed. Screen plots can be sent to disk file in HP-GL format.

Developer: Prof. Dr. Mustafa Kizilcay (m.kizilcay@fh-osnabrueck.de), Germany.
Licensing: freely available without separate licensing to all ATP users.
Distribution: EEUG annual CD distribution, EEUG, JAUG secure FTP/Web sites.

WPCPlot is a graphical output program for ATP-EMTP running under Microsoft Windows 95/98/NT/2000. The program is capable of processing PL4-files of C-like and formatted types Maximum 6 variables in the same diagram are allowed. Zooming, redraw features and a readout facility to obtain instantaneous values of plotted curves are provided. Screen plots can be copied to clipboard or save as color or monochrome bitmap image file.

Developer: Prof. Dr. Mustafa Kizilcay, m.kizilcay@fh-osnabrueck.de, Deniz Celikag, dcelikag@aol.com.
Licensing: available only for EEUG members at present.

Main characteristics of other postprocessors for ATP are summarized in [6].

1.3.4 Typical EMTP applications

ATP-EMTP is used world-wide for switching and lightning surge analysis, insulation coordination and shaft torsional oscillation studies, protective relay modeling, harmonic and power quality studies, HVDC and FACTS modeling. Typical EMTP studies are:

- Lightning overvoltage studies
- Switching transients and faults
- Statistical and systematic overvoltage studies
- Very fast transients in GIS and groundings
- Machine modeling
- Transient stability, motor startup
- Shaft torsional oscillations
- Transformer and shunt reactor/capacitor switching
- Ferroresonance
- Power electronic applications
- Circuit breaker duty (electric arc), current chopping
- FACTS devices: STATCOM, SVC, UPFC, TCSC modeling
- Harmonic analysis, network resonances
- Protection device testing

1.3.5 Hardware requirements for ATP

ATP is available for most Intel based PC platforms under DOS, Windows 3.1/9x/NT, OS/2, Linux and for other computers, too (e.g., Digital Unix and VMS, Apple Mac's, etc.). Most users, including program developers, use Intel Pentium-based PCs with MS-Windows 9x/NT. A standard Pentium PC configuration with min. 128 MB RAM, hard disk (20 MB free space) and VGA graphics is sufficient to execute ATP under MS-Windows. Most popular program versions are at present:

- MS-Windows 9x/NT/2000/XP™: 32-bit *GNU-Mingw32* and *Watcom ATP*
- MS-DOS, MS-Windows 3.x/95/98™: 32-bit *Salford ATP* (requires DBOS/486)
- *Linux:* GNU version of ATP

1.4 Contents of this manual

This User's Manual of ATPDraw for Windows 3.5 contains five parts:

INSTALLATION MANUAL

How to obtain the ATP license
How to download ATPDraw
How to install ATPDraw
Hardware requirements
How to configure your system
How to use ATPDraw as operating shell for other ATP simulations
How to communicate with other users and program developers

INTRODUCTORY MANUAL

How to create a circuit in ATPDraw
Operating windows
Your first circuit
Three-phase circuits

REFERENCE MANUAL

Reference of main menu items and program options
Reference of the Component, Node and Group dialog boxes
Reference of ATPDraw circuit objects

ADVANCED MANUAL

How to create new circuit objects in ATPDraw
How to use new features: Grouping and $PARAMETERS
How to use MODELS and $Include in ATPDraw?
How to use the integrated LCC object for line/cable modeling
How to use the integrated BCTRAN object for transformer modeling
Referencing four non-standard Component dialog boxes:

- Saturable 3-phase transformer
- Univeral Machines
- Statistical switches
- Harmonic source

APPLICATION MANUAL

Simple line energization studies
Pulse width modulated induction machine
Creating libraries of circuits and user specified libraries
HVDC station, rectifier/converter modeling
Shunt capacitor bank switching
Line/cable constant application examples
Single-phase to ground fault and fault tripping transients
Electric arc simulation using MODELS controlled switches
Lightning studies, arrester modeling
Transformer energization, inrush currents
Line energization studies with statistical approach

1.5 Manual conventions

The following typographical conventions are used in this manual:

Italic: Menus in ATPDraw
E.g.: Select *Edit* | *Rotate* : Select *Rotate* command in the pop-up menu *Edit.*
`Courier 9 - 10`: Data files.
E.g.: Listing of ATP input files, MODELS code, etc.
Description of menu options in component dialog boxes.
`Courier 11 - 12`: Data code and file names.
E.g.: Give the file the name `HVDC_6.LIB` and store it in the `\USP` directory.
The `\USP` directory is a directory under the main directory of ATPDraw.
`Courier 12` : Commands on the DOS prompt.
E.g.: `C:\TMP>`**`setup`**: Type the command **`setup`** at `C:\TMP>`.

SINTEF

2. Installation Manual . . .

. . . .

ATPDraw™
for Windows
3.5

2.1 ATP licencing policy

ATPDraw and the present documentation includes ATP proprietary information, thus *ATP licensing is mandatory* prior to get permission to download the program from the Internet. ATP license is free of all charge for all who have not engaged in EMTP commerce, and it can be obtained from the Canadian/American EMTP User Group, or an authorized regional users group. In general, organizational licensing is preferred over licensing of individuals. Undergraduate students are not licensed personally. If ATP usage is to be organizational rather than personal (i.e., if ATP materials are to be used by, in, for, or on behalf of, a company, university, etc.), the licensee must certify that the organization has not participated in EMTP commerce -- nor has any employee, contractor, or other agent who would be granted access to ATP materials. Once one is licensed, he/she is authorized to download ATP materials from the secure Internet sites or obtain them from a similarly licensed user, or order these materials from the regional user groups.

At present the Canadian/American, European and the Japanese user groups accepts ATP license applications via the Internet. Interested parties are requested to visit the on-line licensing page at www.emtp.org, fill-in and submit the appropriate web-form. Potential users of other continents must follow the licensing procedure of their regional EMTP user group. Geographical location of ATP-EMTP user groups and contact information details are shown below:

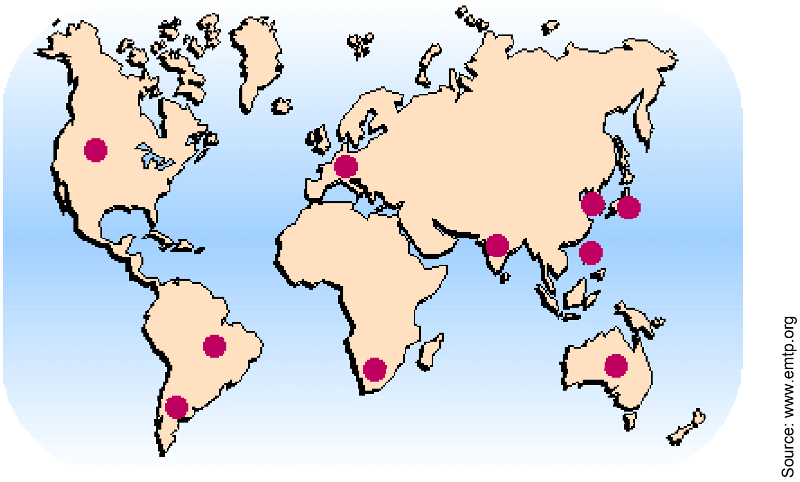


Fig. 2.1 - Location of ATP-EMTP user groups.

Name	E-mail and WWW address
Canadian/American EMTP User Group	canam@emtp.org
European EMTP-ATP Users Group (EEUG)	eeug@emtp.org, www.eeug.org
Japanese ATP User Group (JAUG)	jaug@emtp.org, www.jaug.jp/~atp/index-e.htm
Latin American EMTP User Group (CLAUE)	claue@emtp.org, www.furnas.gov.br/atp
Argentinian EMTP User Group (CAUE)	caue@emtp.org http://iitree.ing.unlp.edu.ar/estudios/caue/caue.html
Australian EMTP User Group (AEUG)	aeug@emtp.org
Korean EMTP User Group (KEUG)	keug@emtp.org
Republic of China EMTPUser Group	d023@taipower.com.tw
Indian EMTP User Group (IEUG)	malathi@bom4.vsnl.net.in
South African ATP User Group (SAAUG)	j.vancoller@ee.wits.ac.za, www.ee.wits.ac.za/~atp

Chapter 2.7.3 of the Installation Manual gives further information about the ATP related Internet resources.

2.2 How to download ATPDraw?

ATP licensing is mandatory prior to receiving any materials. Following the license agreement approval by an authorized user group, you are eligible to use the ATP program and all ATP related tools, like ATPDraw and this manual. There are different sources of obtaining ATPDraw and additional ATP related tools and program manuals:

- Order ATP materials from the Canandian/American EMTP User Group (`http://www.emtp.org/canamfl.html#ger`) in Oregon, USA, or from the European EMTP-ATP Users Group Association (`http://www.eeug.org`).
- Download from secure, password-protected web site of the European EMTP-ATP Users Group Association (`http://www.eeug.org/files/secret`)
- Download from secure, password-protected web sites of the Japanese ATP User Group (`http://alpha.kisarazu.ac.jp/~secure`, or `http://pels.pwr.eng.osaka-u.ac.jp/~atp/restricted`)
- Download from the password-protected FTP file server at Michigan Technological University in Houghton (USA) (`http://www.ee.mtu.edu/atp/ftp.html`).

Please contact the regional user group to acquire passwords to access these sites. Passwords are changed regularly!

2.3 Hardware requirements for ATPDraw

ATPDraw requires moderate CPU power and memory. It runs even on a slow Pentium 100 MHz/32 MB PC with acceptable speed. A standard Pentium PC configuration with min. 128 MB RAM (256 MB under Windows 2000 and XP), 100 MB free hard disk space and XVGA graphics is sufficient to execute ATPDraw and other ATP programs.

2.4 Program installation

The `/atpdraw` subfolder under the above secure servers contains a zip-compressed archive `atpdraw3x.zip`, a short installation guide and the latest patch file (if any). Following a successful download of the distribution kit, perform the next operations:

1) Copy the `atpdraw3x.zip` file into a TEMP directory and unzip it.
2) Run the program **`setup.exe`**. The installation process will be assisted by a standard Install Shield Wizard.
3) Specify a destination directory for ATPDraw when prompted. It is wise to avoid using directory name including "space". E.g. `C:\Program Files` is not recommended. Install the program into a root directory, e.g. `D:\ATPDraw3`. If you are not allowed to install programs outside `Program Files`, let the Wizzard to install ATPDraw into this folder. Note that in such a case special care is needed when setting environmental variables for ATP.
4) The installation process will be completed after creating a new shortcut for ATPDraw under *Start | Programs | ATPDraw*. When you start ATPDraw3.x first time it will create the necessary system sub-folders `/ATP`, `/BCT`, `/GRP`, `/LCC`, `/MOD`, `/Project` under the main program folder. If you install ATPDraw first time skip points 5) and 6).

5) Copy the existing files in the /USP, /MOD and /Project folders of the earlier ATPDraw version into the new folders. Even if you do not have your own USP objects it is strongly advised to create an empty /USP sub-folder. Lack of it, projects with embedded USP objects received from another user will not run properly. Standard objects are stored in a single file called ATPDraw.scl, thus no separate /SUP and /TAC folders are needed in version 3 and above.
6) Copy ATPDraw.ini of the previous version (2.x or 3.x) into the system folder of the new ATPDraw installation. This way you can preserve the *ATP | Edit Commands* and many other program settings if you modified default values earlier.
7) Download the latest patch file called patchxv3.zip (if exists on the server), then unzip it and simply overwrite the existing files in the ATPDraw system folder with the newer ones received in the patch file.

The program installation will create a directory structure as shown next. ATPdraw can be uninstalled in the standard manner using Windows' uninstaller (*Start menu | Settings | Control Panel | Add/Remove programs*).

```
PROJECT        <DIR>         10-22-01  9:54p Project
LCC            <DIR>         10-22-01  9:54p lcc
ATP            <DIR>         10-22-01  9:58p Atp
USP            <DIR>         04-29-02  8:11a Usp
GRP            <DIR>         10-22-01  9:58p Grp
MOD            <DIR>         10-22-01  9:58p Mod
BCT            <DIR>         03-22-02 12:42p Bct
ATPDRAW  CNT         3,091   04-11-02  3:23p ATPDraw.CNT
ATPDRAW  EXE     1,182,208   04-29-02 10:58a Atpdraw.exe
ATPDRAW  HLP       421,824   04-11-02  3:24p ATPDraw.hlp
ATPDRAW  SCL       203,378   04-25-02 10:08p ATPDraw.scl
_ISREG32 DLL        24,576   02-07-96  8.07a _ISREG32.dll
DeIsL1   ISU         2,863   06-08-02 10.11a DeIsL1.isu
RUNAF    BAT            71   10-22-01 10:22p runAF.bat
RUNATP_G BAT            90   10-22-01 10:56p runATP_G.bat
RUNATP_S BAT           108   10-22-01 10:55p runATP_S.bat
RUNATP_W BAT            90   10-22-01 10:54p runATP_W.bat
```

The files _ISREG32.dll and DeIsL1.isu are created by the install shield for uninstall purposes.

2.5 Files and sub-folders in the ATPDraw system folder

To use ATPDraw four files are required: ATPDraw.exe, ATPDraw.scl (standard component library), ATPDraw.cnt (help content), and ATPDraw.hlp (help file). Besides, the user can create his own components (user specified or models) and include files.

Project file: When the user saves a circuit the work is stored in the project file (*.adp = atpdraw project). This file contains the circuit with all data and graphical representation and in addition, all files describing user specified components (support, line/cable, trafo, model and library files). The project file is compressed by a public domain Pkzip routine and can in fact be opened with any version of WinZip. It may occur that *a virus checker inaccurately recognizes the project files* as virus infected and quarantine them when you send or receive such a file in e-mail attachments. If it happens, the local virus filtering database should be modified to allow the exchange of project files. Contact IT staff!

Support file: All components require a support file. This file describes the type of component, the nodes (phases, position, identity) and data (default value, limits, parameter flag, number of digits, identity), the default icon (41x41 pixels) and the help file. The version 3.x of the support file also contains options for specifications of the data format in ATP but this is not used yet. The support files for standard components are zipped together in the file `ATPDraw.scl` (standard component library) and this file is required together with the project file to open and run a project. The support files can be edited inside ATPDraw in the *Objects* menu. The default icon can also be modified by using the built in icon editor. New objects can be created by specifying new support files.

ATP file: This file is produced by ATPDraw and used as input to ATP simulation. The `.atp` files are located in the `\ATP` sub- directory and can be edited with any text-processors, including ATPDraw's own *Text Editor*. It is advised, however only for experts to modify this file manually.

Line/cable file: A line or cable is described by an `.alc` file (atpdraw line/cable). This binary file contains the line-, cable constants or cable parameter data. The alc data are used by ATPDraw to run ATP and create an electrical model of a line or cable.

BCTRAN file: A BCTRAN (Transformer) component is described in a `.bct` file. This binary file contains the input data required for the supporting routine BCTRAN of ATP-EMTP. The `.bct` data are used by ATPDraw to run ATP and create an electrical transformer model.

Model file: A model is described in a model file (`.mod`). This text file starts with MODEL <name> and ends with ENDMODEL. The <name> must be equal to the model file name. The model file is included directly in the final ATP input data file. It is recommended to store the models file in the `\MOD` sub-directory.

Include files: User Specified Components are described in a library file (`.lib`). This text file has a pre-defined format (as specified in by the Data Base Module of ATP) and contains a header describing the positions of the parameters, further the ATP cards and finally a trailer containing the specification of the parameters. The library file is included in the ATP input file with $Include. The line/cable components also have a library file, created by ATPDraw. Some nonlinear components or saturable transformers might also have an include file for the nonlinear characteristic.

2.5.1 Organizing the files

When ATPDraw opens a project the included files are stored on disk preferable where they were located when the project was stored, or if this directory does not exists, in the default directories specified under *Tools | Options/Directories*. In both cases existing files are not overwritten without noticing the user. The time stamp of the disk file and the file in the project is compared and a warning like *File on disk is older/newer than the stored project file* will appear. It is important that the user replies to these messages carefully. To avoid conflicts the user is advised not to use the same name for several different user created models.

When the project is stored the disk files are not deleted (except for group support-files). Thus, as times goes by the number of files on disk grows. It is the user's responsibility to tidy up the directories. Remember: All required files are stored in the project and only the files you modify yourself outside a project need to be kept. All other files (sup, lib, mod, alc, bct) can be deleted.

2.5.2 Configuring ATPDraw

The `ATPDraw.ini` file contains customizable program options. Generally, default settings meet most of the user's requirements. When required, the `.ini` file can either be modified via *Tools* | *Options* menu of the program, or by using a text editor. Default values and supported options are described in the Appendix part of this Manual.

2.6 Interfacing ATPDraw with other programs of the ATP-EMTP package

The ATP-EMTP simulation package consists of various separate programs which are communicating with each other via disk files: i.e. the output of pre-processors are used as input for the main program `TPBIG.EXE`, while the product of the simulation can be used as input for plotting programs. The main program itself is often used as pre-processor (e.g. for LINE CONSTANTS, CABLE CONSTANTS, BCTRAN or DATA BASE MODULE runs), and the punch-file products in that cases can be re-used as input in a subsequent run via $Include. Taking that the structure of the program components is rather difficult, a user shell to supervise the execution of separate programs and input/output flows has a great advantage.

The new *Edit Commands*...feature of ATPDraw supports to extend the command set under the *ATP* menu by integrating optional user commands, such as *Run ATP (file) / Run PlotXY / Run TPPlot / Run PCPlot / / Run ATP_Analyzer / Run ACC / Run PL42mat, etc.* This option makes possible to use the ATPDraw program as a graphical operating environment and execute the other ATP programs in a user friendly way as shown in Fig. 2.2.

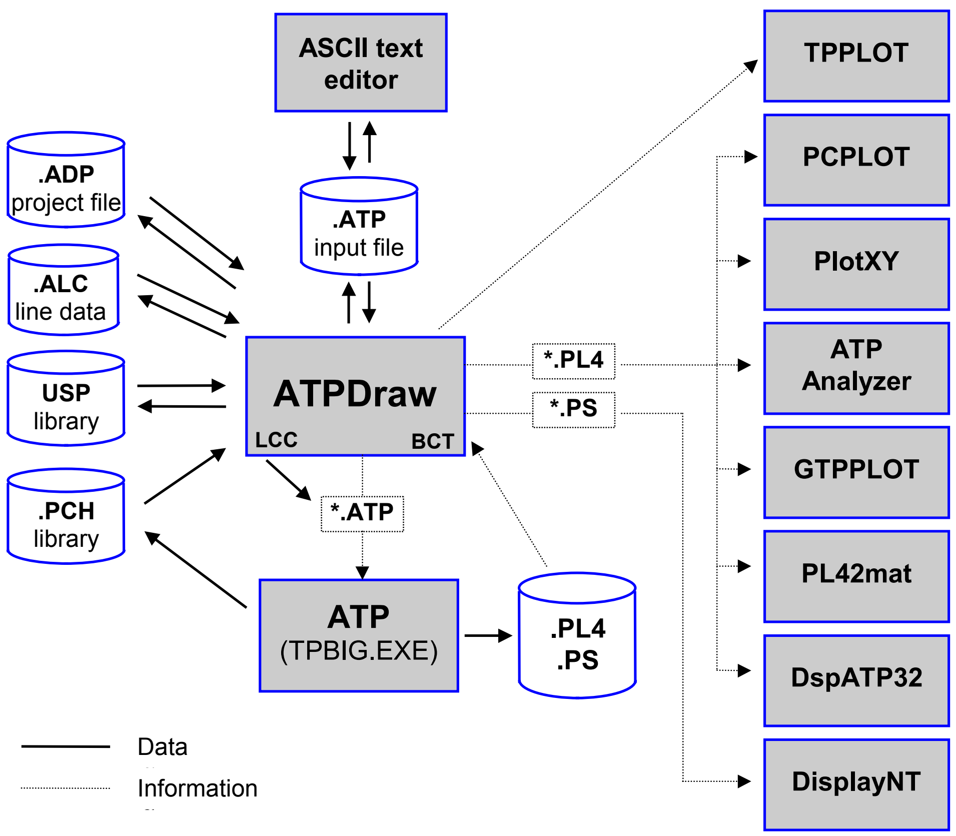


Fig. 2.2 - Interaction between ATPDraw and the other ATP programs.

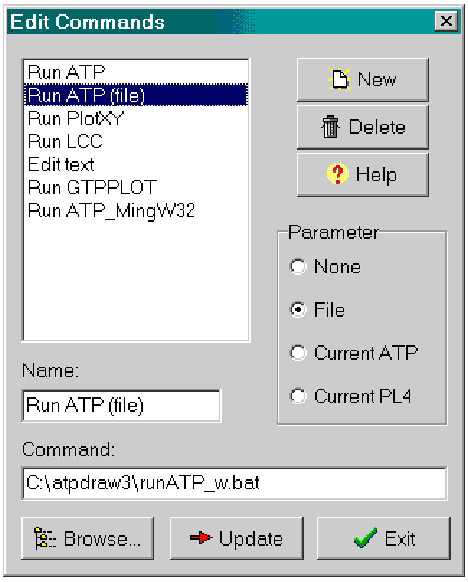


Fig. 2.3 - The Edit Commands dialog box.

Fig. 2.4 - User specified commands.

In the *Edit Commands* dialog box of Fig. 2.3 the user can specify the name of a `.bat` or an `.exe` file and the name of a file, which then will be sent as parameter (e.g. `ATP.bat <current .atp file>` or `PlotXY.exe <current .pl4 file>`) when ATPDraw executes these external programs. The *Name* field specifies the name of the command, while the *Command* and *Parameter* fields specify the name of the file to be executed and the optional parameter. Selecting *Current ATP* radio button, the full name of the ATPDraw project in the current circuit window with extension `.atp` will be sent as parameter. When selecting the *File* button, the ATPDraw performs a file open dialog box before executing the command, where the user can select a file, which is then will be passed as parameter. The commands are inserted in the *ATP* menu dynamically, when the user activates the *Update* button as shown above.

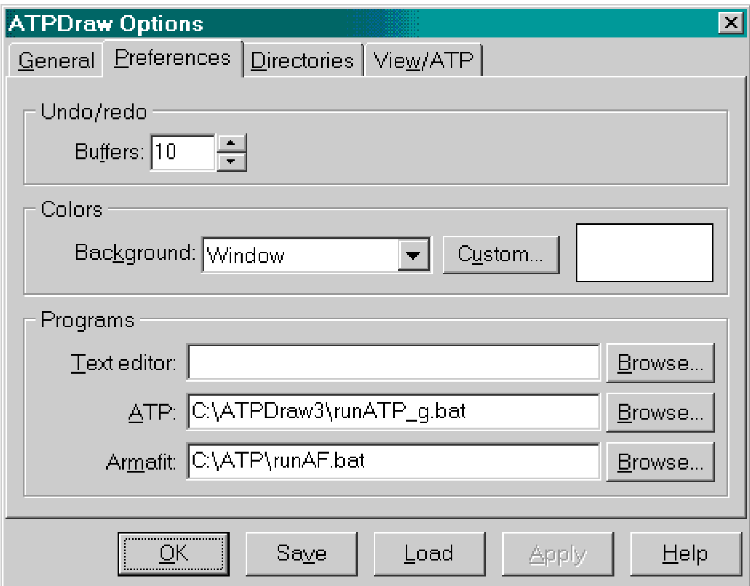


Fig. 2.5 - Default settings to run ATP and Armafit.

The default batch command that is executed when the user selects *run ATP* or (F2) is specified under the *Tools* | *Options /Preferences* tab as shown in Fig. 2.5. Checking the contents of this batch file is very important following the program installation, because ATPDraw needs to be able to execute ATP for several reasons automatically, and this has always performed by activating this command. It must be noted that ATPDraw has no connection with the main program of ATP (`TPBIG.EXE`) at the code level or via DLLs. The *run ATP* menu item simply executes the external commands specified by the user. So it is always the user's responsibility to install ATP properly and provide these external `.bat` files in correct format.

2.6.1 Calling Watcom ATP and GNU MingW32 ATP from ATPDraw

Proper execution of the Watcom and GNU version of ATP requires that environmental variables `WATDIR` or `GNUDIR` be set correctly[1], i.e. `SET WATDIR=Drive:\Path\WatcomATPdir\` in the `AUTOEXEC.BAT` if you use Win9x, or set these parameters under *My Computer | Properties* dialog if Windows NT/2000 or XP is used. The *RunATP_W.BAT* and *RunATP_G.BAT* commands are created by the install program of ATPDraw. These batch files has a single line:

```
%watdir%tpbig.exe both %1 * -r
%gnudir%tpbig.exe both %1 s -r
```

If an additional "`W`" or "`G`" is seen at the end of the ATP executable (`TPBIG.EXE`) in your installation, you have to modify the `RunATP_*.BAT` accordingly. You may find inserting some additional commands into these batch files, as well. E.g.:

```
echo off
%gnudir%tpbigg.exe both %1 s -r
pause             -- waits for user interaction before the DOS box of ATP closed (optional)
del dum*.bin      -- delete temporary files created by ATP (optional)
del *.tmp
del ..\*.tmp
```

2.6.2 Running Salford ATP from inside ATPDraw

Note that Salford ATP can be used under Win9x only! To ensure smooth interaction between ATPDraw, DBOS and Salford ATP, it is recommended to add the DBOS directory to the search PATH and specify an `ATPDIR` environmental variable in the `AUTOEXEC.BAT` file, i.e. `SET ATPDIR=Drive:\Path\SalfordATPdir\`. The *RunATP_S.BAT* is created by the install program of ATPDraw and this batch files has a single line:

```
run77.exe %atpdir%tpbig.exe /PARAMS both %1 * -r
```

If an additional "*S*" is seen at the end of the ATP executable (`TPBIG.EXE`) in your installation, you have to modify the `RunATP_S.BAT` accordingly. You may find inserting some optional commands into the batch file, as well. E.g.:

```
echo off
run77.exe %atpdir%tpbigs.exe /PARAMS both %1 * -r
pause             -- waits for user interaction before the DOS box of ATP closed (optional)
del dum*.bin      -- delete temporary files created by ATP (optional)
del *.tmp
del ..\*.tmp
```

1 The Install Shield wizard of the annual ATP program ditribution for EEUG members makes these settings automatically.

RunATP_S.BAT properties should be set as shown next:

Program:
Cmd line: Drive:\Path\RunATP_S.BAT
Batch file: DBOS_ON.BAT
Run: Normal Window
Close on exit: ☑
Screen Usage: Window

Because Salford ATP is running under DBOS, a single line batch file DBOS_ON.BAT must be executed before TPBIG.EXE is launched. The corresponding single line command of the DBOS_ON.BAT file referenced in the *Properties* | *Batch file* should be as follows:

Drive:\dbos3_5\dbos /page 800000 *{800000 limits DBOS to use 8MB extended memory}*

2.6.3 Calling PlotXY , PCPlot or ATP_Analyzer

After creating the *Run PlotXY* and/or *Run PCPLOT* commands using the *ATP* | *Edit Commands* submenu selet "Current PL4" as *Parameter* and *Browse* to select the name of the executable disk file of the corresponding application. *Update* button adds the new *Run…*command to the *ATP* menu.

2.6.4 Running TPPLOT from ATPDraw

TPPLOT is also a DBOS application and it does not run under Windows NT/2000/XP. Settings are very similar to that of in section 2.6.2 for Salford ATP. When creating the *Run TPPLOT* command using the *ATP* | *Edit Commands* submenu, select "None" as *Parameter* and Drive:\Path\TPPLOT.BAT as *Command*. The TPPLOT.BAT has a single line **run77 %atpdir%tpp.exe.** *Properties* are almost identical as in section 2.6.2, with *Screen* as exception that should be set full-screen. Moreover, it is suggested to modify the settings at the end of the TPPLOT.BEG file as shown below:

C @LAB2 -- *comment it out or copy the LAB2 file into your ATPDraw directory*
FILE ATP*.PL4 -- *add these two commands at the end of the file*
CHOICE

2.6.5 ATPDraw command line options

Command lines are rarely used under Windows operating systems, nevertheless ATPDraw provides an option to load one or more project files at program start. In the example below, the project file my1st.adp and my2nd.adp will be loaded automatically and displayed in separate circuit windows.

C:\ATPDRAW>atpdraw c:\atpdraw\cir\my1st.adp c:\cir\my2nd.adp

In MS-Windows environment you can use this property to create a shortcuts on the desktop for the ATPDraw project files. For instance, click with the right mouse button on an empty space of the desktop and select *New* | *Shortcut*, then browse and select ATPDraw.exe. Click right on the just created icon and select *Properties*. Specify the 'Target:' properties of the new shortcut as the full path of the program including the project file name (e.g. **c:\atpdraw\atpdraw.exe**

mycir.adp), and the 'Start in:' parameter as the project file directory (e.g. **c:\atpdraw\project**).

2.7 How to get help?

ATPDraw offers a standard Windows help file system. This file provides help on all windows and menus in ATPDraw and assists in building up a circuit. Several links between help pages and a relatively large index register for searching text or phrases are also available. A *Help* button is attached to all circuit objects, which shows a brief overview of the meaning of each parameter. Modification and extension of these help files with users' own remarks are also possible using the built in *Help Editor* in the *Tools* menu.

2.7.1 Help from the author of ATPDraw

The author of the program is also available for questions from ATPDraw users, but is only responsible to Bonneville Power Administration and Pacific Engineering Corporation.

Address: Dr. Hans Kr. Høidalen
SINTEF Energy Research
7465 Trondheim - NORWAY
http://www.energy.sintef.no
E-mail: Hans.Hoidalen@elkraft.ntnu.no
Phone: + 47 73594225
Fax: + 47 73594279

The ATPDraw Web page is maintained at address:

http://www.ee.mtu.edu/atp/atpdraw/ATPDraw.html

2.7.2 Help via electronic mail

Electronic mail is the most known feature of the Internet. By this way, anyone who has an account on a computer connected to the Internet can send messages to others. For ATP users this service provides an easy, efficient and very fast way of communication with other users all over the world, including program developers, regional user group representatives, or the author of ATPDraw.

2.7.3 Help via the ATP-EMTP-L mailing list

The listserver is an E-mail remailer program, which rebroadcasts incoming messages to all subscribers to the list. The European EMTP-ATP Users Group Association in cooperation with the German Research Network (DFN) and the University of Applied Sciences of Osnabrück, Germany operates a free electronic mailing list using address atp-emtp-l@listserv.dfn.de. This LISTSERV mailing list is for ATP-related announcements, questions, answers, etc. The ATP-EMTP-L list is *moderated* and only licensed ATP users are entitled to subscribe by means of the authorized persons of the regional ATP-EMTP user groups, who checks first the license status of the applicant, then send a subscription request to the list operator. To learn more about the subscription procedure and the operation rules of this very active mailing list, please visit the www.emtp.org web site.

After your name has been added to the list, you can post messages. To do this, you simply send e-mail to **atp-emtp-l@listserv.dfn.de**. Your message then will be submitted to moderators, who decide whether or not to accept it. The task of moderators is maintenance of the quality of communication and discussion. The language of communication is English. Messages written in any other language are not accepted. The author of each submission must be clearly identified. This includes name, organizational affiliation, and location. Attachments, especially encoded files, are not allowed. They can be forwarded later to interested persons by private e-mail. Any subscriber who sends a message to this mailing list gives up his right to confidentiality. This is regardless of the message's possible declaration in auto-attached legal disclaimers, which are removed by moderators. Subscribers of the ATP-EMTP-L mailing list must fulfill the ATP license requirements. Specifically, they are forbidden to disclose to non-licensed persons ATP information that is received from this mail service.

2.8 Available circuit objects in ATPDraw

At the time of writing of this manual ATPDraw's standard component library contains 194 circuit object support files. These 194 files support more than 100 of ATP's components, i.e. many components have several versions in ATPDraw.

Standard components

Linear branches:

Resistor, Inductor, Capacitor, RLC
RLC 3-phase, symmetric and non symmetric
Inductor and capacitor with initial condition

Non-linear branches:

1-phase nonlinear R and L components
Current dependent resistor, type 99
Type-93, 96 and 98 nonlinear inductors including initial flux linkage conditions
Time dependent resistor, type 97
Single and 3-phase MOV type 92 exponential resistor
TACS controlled resistor

Line models:

Lumped, PI-equivalents (type 1, 2...) and RL coupled components (type 51, 52...)
RL symmetric, sequence input. 3 and 6-phase
Distributed lines of constant parameters, Transposed (Clarke), untransposed (KCLee)
LCC objects: Bergeron, nominal PI, JMarti, Semlyen and Noda models

Switches:

Time controlled. 1 and 3-phase
Voltage controlled
Diode, thyristor, triac (type 11 switches)
Simple TACS controlled switch of type 13
Measuring switches
Statistic and systematic switches, independent and master-slave

Sources:

DC, type 11
Ramp, type 12
Two-slope ramp, type 13
AC source. 1 and 3 phase, type 14
Double-exponential surge source, type 15

Heidler-type source, type 15
Standler-type source, type 15
CIGRÉ-type source, type 15
TACS source, type 60
Ungrounded DC source, type 11+18
Ungrounded AC source, type 14+18

Machines:

Synchronous machine type 59 with no control, or max. 8 TACS controls
Universal machines. Universal machines (type 1, 3, 4, 6, and 8)

Transformers:

Single-phase and 3-phase ideal transformer. Type 18 source
Single-phase saturable transformer
3-phase, 2- or 3 winding saturable transformer
3-phase 2-winding saturable transformer, 3-leg core type of high homopolar reluctance
BCTRAN. 1-3 phases, 2-3 windings. Auto-transformers, Y-, and D- connections

MODELS

Input/output and Data variables of MODELS code are recognized automatically
Corresponding support file for the model is automatically created
Type 94 (Thevenin, Norton, Iterative) objects are supported

TACS

Coupling to circuit object helps in hybrid simulations
Transfer functions: General Laplace transfer function with or without limits
Integral, Derivative, first order Low and High Pass transfer functions
Fortran statements: General Fortran statement (single line expression)
Simplified Math statements or Logical operators
Sources: DC, AC, PULSE, RAMP.
TACS devices. All devices except type-55
Initial condition for TACS objects (type-77)

User specified objects

Users can create new objects using Data Base Modularization and $Include

Frequency components

Harmonic sources for Harmonic Frequency Scan studies
Single and 3-phase frequency dependent loads in CIGRÉ format
Single phase RLC element with frequency dependent parameters

SINTEF

3. Introductory Manual . . .

. . . .

ATPDraw™
for Windows
3.5

This part of the user's manual gives the basic information on how to get started with ATPDraw. The Introductory Manual starts with the explanation of how to operate windows and mouse in ATPDraw. The manual shows how to build a circuit step by step, starting from scratch. Then special considerations concerning three phase circuits are outlined.

3.1 Operating windows

ATPDraw has a standard Windows user interface. This chapter explains some of the basic functionalities of the *Main menu* and the *Component selection menu* of the *Main window*.

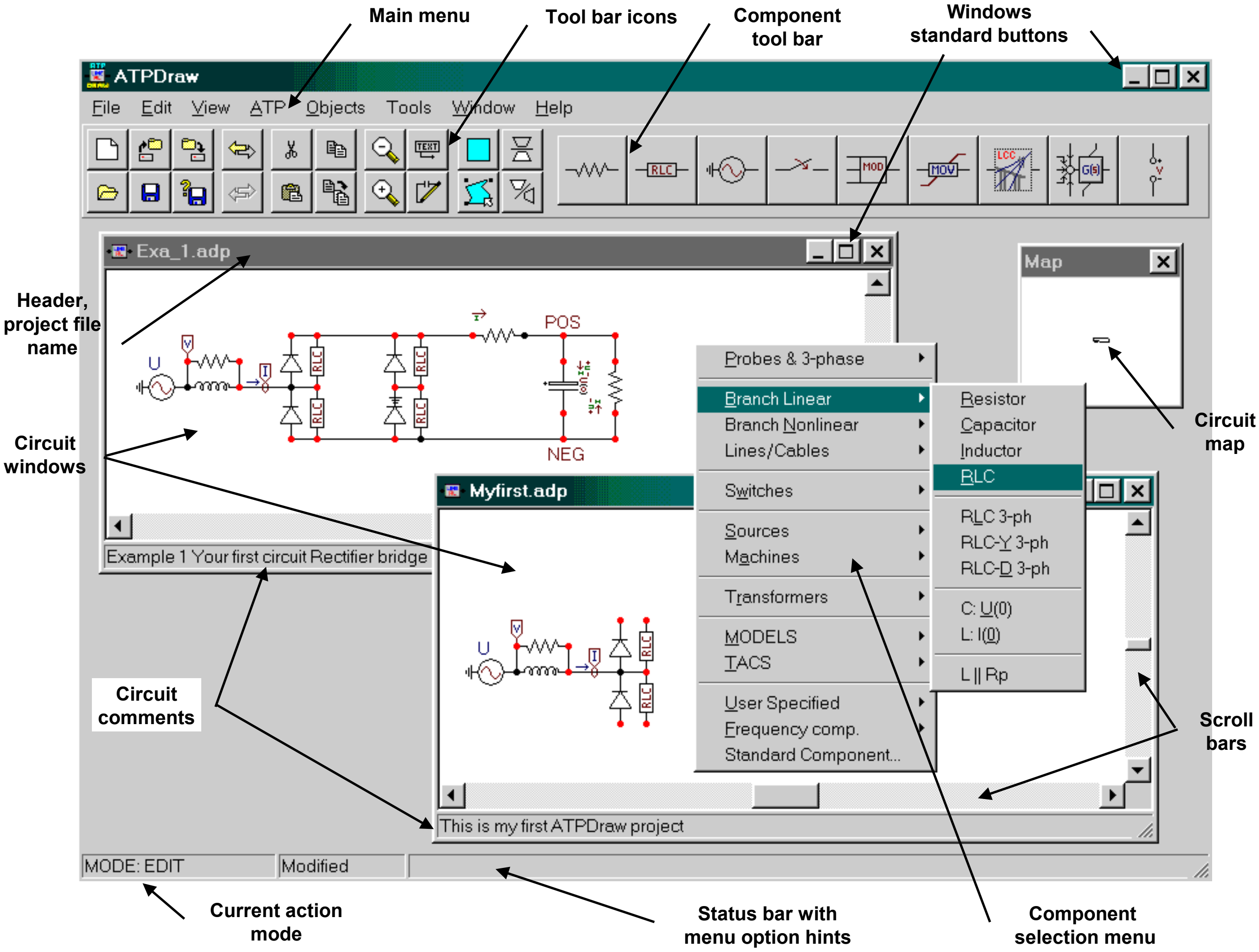


Fig. 3.1 - The Main window and the floating Component selection menu.

The ATPDraw for Windows program is functionally similar to the DOS version [1]. The *Component selection menu* is hidden, however, but appears immediately when you click the right mouse in the open area of the *Circuit window*.

Fig. 3.1 shows the main window of ATPDraw containing two open circuit windows. ATPDraw supports multiple documents and offers the user to work on several circuits simultaneously along with the facility to copy information between the circuits. The size of the circuit window is much larger than the actual screen, as is indicated by the scroll bars of each circuit window. The *Main window* consists of the following parts:

Header + Frame:
As a standard Windows element, it contains the system menu on the left side, a header text and minimize, maximize, exit buttons on the right side. The main window is resizable.

System menu:	Contains possible window actions: Close, Resize, Restore, Move, Minimize, Maximize or Resize and Next. The last one exists only if multiple circuit windows are open.
Header text:	The header text is the program name in case of the main window and the current circuit file name in case of the circuit window(s). To move a window, click in the header text field, hold down and drag.
Minimize button:	A click on this button will iconize the main window.
Maximize button:	A click on this button will maximize the window. The maximize button will then be replaced with a resize button. One more click on this button will bring the window back to its previous size.
Corners:	Click on the corner, hold down and drag to resize the window.

Main menu:
The main menu provides access to all the functions offered by ATPDraw. The menu items are explained in detail in the Reference part of this Manual:

File:	Load and save circuit files, start a new one, import/export circuit files, create postscript and metafile/bitmap files, print the current circuit and exit.
Edit:	Circuit editing: copy/paste/delete/duplicate/flip/rotate, select, move label, copy graphics to clipboard and undo/redo etc.
View:	Tool bar, status bar and comment line on/off, zoom, refresh and view options.
ATP:	Run ATP, make and edit ATP-file, view the LIS-file, make node names, ATP-file settings (miscellaneous, file format, file sorting etc.), assign data to variables for $PARAMETER usage and specify RECORD for MODELS output requests.
Objects:	Edit support files (default values, min/max limits, icon and help file), create new files for MODELS and User Specified Objects.
Tools:	Icon editor, help file editor, text editor, setting of various program options.
Window:	Arrange the circuit windows and show/hide the Map window.
Help:	About box and Windows help file system.

Circuit window:
The circuit is built up in this window. The circuit window is the container of circuit objects. From the *File menu* you can load circuit objects from disk or simply create an empty window to start building a new circuit. Circuit objects include standard ATP components, user specified elements, MODELS and TACS components, connections and relations. To move around in the circuit, you can use the window scrollbars, or drag the view rectangle of the *Map window* to another position.

Component selection menu:
This menu pops-up immediately when you click with the right mouse button in an empty space of the *Circuit window*. In this menu you select the circuit objects. After selecting an object in one of the sub-menus, the object is drawn in the circuit window in marked and moveable mode.

Circuit comments:
A comment line below the circuit window shows a user defined circuit comment text.

MAP window:
This window gives a bird's eye view of the entire circuit. The size of a circuit is 5000x5000 pixels (screen points); much larger than your screen would normally support. Consequently, the *Circuit window* displays only a small portion of the circuit. The actual circuit window is represented by a

rectangle in the *Map window*.

Press and hold down the left mouse button in the map rectangle to move around in the map. When you release the mouse button, the circuit window displays the part of the circuit defined by the new rectangle size and position. The map window is a stay-on-top window, meaning that it will always be displayed on the top of other windows. You can show or hide the map selecting the *Map Window* option in the *Window* menu, or pressing *M* character,

Status bar - Action mode field:
The current action mode of the active circuit window is displayed in the status bar at the bottom of the main window, when the *Status Bar* option is activated in the *View* menu. ATPDraw can be in various action modes. The normal mode of operation is *MODE : EDIT* , in which new objects are selected and data are given to objects. Drawing connections brings ATPDraw into *CONN.END* mode and so on. ATPDraw's possible action modes are:

EDIT	The normal mode.
CONN.END	After a click on a node, the action mode turns into *CONN.END* indicating that the program is waiting for a left mouse click to set the end-point of a new connection. To cancel drawing a connection, click the right mouse button or press the ESC key to return to *MODE : EDIT*.
MOVE LABEL	Indicates a text label move. Clicking the left mouse button on a text label, then holding it down and dragging it enables you to move the label to a new position. If the text label is overlapped by a component icon, the text label can be moved using *Move Label* in the *Edit* menu. Then the action mode turns into *MOVE LABEL*. Releasing the mouse at the new text label location, the action mode returns to *MODE : EDIT*.
GROUP	Indicates region selection. Double clicking the left mouse button in an empty space of the active circuit window enables you to draw a polygon shaped region. To end the selection, click the right mouse button. Any objects within the selected region are marked then for selection. To cancel region selection, press the *Esc* key.
INFO.START	Indicates the start of a relation when *TACS | Draw relation* is activated in the selection menu. Clicking the left mouse button on a component node or on the end-point of another relation will initiate the drawing of a new relation. Relations are used to visualize information flow into FORTRAN statements and are drawn as blue connections, but do not influence the connections of components.
INFO.END	Indicates the end of a relation. The program is waiting for a left mouse button click to set the end-point of the new relation. To cancel drawing relation, click the right mouse button or press the *Esc* key.

Status bar - Modified and Hints field:

The middle field of the status bar is used to display the *Modified* state of the active circuit. As soon as you alter the circuit (moving a label, deleting a connection, inserting a new component, etc.), the text *Modified* appears, indicating that the circuit should be saved before exit. The field will be empty when you save the circuit or undo all modifications. The rightmost field of the status bar displays the menu option hints.

3.2 Operating the mouse

This chapter contains a summary of the various actions taken dependent on mouse operations. The left mouse button is generally used for selecting objects or connecting nodes; the right mouse button is used for specification of object or node properties.

<u>Left simple click:</u>

On object: Selects object or connection.

If the *Shift* key is pressed, the object is added to the current selection group.

On object node: Begins to draw a connection.

Move the mouse to the end node, left click to place, right to cancel.

In open area of the circuit window: Unselects object.

<u>Right simple click:</u>

In open area of the circuit window:

Opens the *Component selection menu*, or

Cancels the connection made if connection draw mode has been activated earlier.

On object node:

Pops-up the *Node data* window.

On unselected object: Opens the *Component* dialog box.

If *Shift* key is pressed simultaneously: opens the circuit window *Shortcut menu*.

On selected object(s): Rotates object(s).

If *Shift* key is pressed simultaneously: opens the circuit window *Shortcut menu*.

<u>Left click and hold:</u>

On object: Moves the object or selected group of objects.

On node: Resizes connection (it is often necessary to select connection first).

In open area of the circuit window: Draws a rectangle for group selection.

Objects inside the rectangle are becoming member of the group when the mouse button is released.

<u>Left double click:</u>

On object node:

Performs the *Node data* window.

On selected or unselected single object:

Performs the *Component* dialog box.

On selected group of objects:

Performs an *Open Group* dialog box.

In open area of the circuit window:

Starts the group selection facility. Click left to create an enclosing polygon, click right to close. Objects inside the polygon become a group.

3.3 Edit operations

ATPDraw offers the most common edit operations like copy, paste, duplicate, rotate and delete. The edit options operate on a single object or on a group of objects. Objects must be selected before any edit operations can be performed. Selected objects can also be exported to a disk file and any circuit files can be imported into another circuit.

Tool	Shortcut key	Equivalent in menus
UNDO	Alt + BkSp	*Edit | Undo*
REDO	Shft+Alt+BkSp	*Edit | Redo*
Cut/Copy	Ctrl+X/Ctrl+C	*Edit | Cut/Copy*
Paste	Ctrl+V	*Edit | Paste*
Duplicate	Ctrl+D	*Edit | Duplicate*
Select/All	Ctrl+A	*Edit | Select All*
Select/Polyg.	Ctrl+P	*Edit | Select Group* (or left double click in open space)
Select/Type	Ctrl+T	*Edit | Select Type/Group number*
Label	Ctrl+L	*Edit | Move Label*
Rotate	Ctrl+R	*Edit | Rotate* (or right click)
Rubber Band	Ctrl+B	*Edit | Rubber Bands*
Edit Group/Circuit	Ctrl+G/Ctrl+H	*Edit | Edit Group/Circuit* (one layer down or up)
Zoom In/Out	+ / -	*View | Zoom In / Out*
Zoom window	Z	*View | Zoom* (pups-up zoom window)
Refresh	R	*View | Refresh* (redraw the circuit)

3.4 Overview of working with ATPDraw

After selecting a component in the *Component selection menu* the new circuit object appears in the middle of the circuit window enclosed by a rectangle. Click on it with the left mouse button to move, or the right button to rotate, finally click in the open space to unselect and place the object.

To select and move an object, simply press and hold down the left mouse button on the object while moving the mouse. Release the button and click in an empty area to unselect and confirm its new position. The object is then moved to the nearest grid point (known as gridsnapping). If two or more components overlap as a consequence of a move operation, you are given a warning message and can choose to proceed or cancel the operation.

Selecting a group of objects for moving can be done in three ways: Holding down the *Shift* key while left clicking on an object. Pressing and holding down the left mouse button in an empty area enables the user to drag a rectangular outline around the objects he wants to select. And finally, double-clicking the left mouse button in an empty area enables the definition of a polygon-shaped region by repeatedly clicking the left mouse button in the circuit window. To close the region, click the right mouse button. Objects that are defined to fall within the indicated region or rectangle are added to the selected objects group. For components this means that the centre point of a component icon must lie within the defined region or rectangle. For connections and relations the region or rectangle must surround both end-points. To move the selected group of objects, press and hold down the left mouse button inside the group while moving the mouse. Unselect and confirm the new position by clicking in an empty area. Any overlapping components will produce a warning. To move objects outside of the visible part of the circuit, use the window scrollbars or the view rectangle in the map window. Any selected objects or group will follow the window to its new position. Objects or a group can be rotated by clicking the right mouse button inside the selected object or group. Other object manipulation functions, such as undo/redo and clipboard options can be found in the *Edit* menu. Additionally, the most frequently used object manipulation functions can be accessed by holding down the *Shift* key while clicking with the right mouse button on an object or on a selected group of objects. This will display and activate the circuit window shortcut menu.

Components and component nodes can be opened for editing by a right-click (or left double-click) on an unselected component or node. Either the *Node data*, *Component* or *Open Probe* dialog box will appear, allowing the user to change component or node attributes and characteristics. The *Component* dialog box has the same layout for most circuit objects. In this window the user must specify the required component data. The number of DATA and NODES menu fields are the only difference between input windows for standard objects. The nonlinear branch components have a *Characteristic* page too, in addition to the normal *Attributes* page, where the nonlinear characteristics and some include file options can be specified.

Selecting a single component and press the *Crtl+F1* key combination, the component specific help is displayed. When double-clicking on a selected group of objects, the *Open Group* dialog box will appear, allowing the user to change attributes common to all components in that group, such as group number, hide and lock state. Default component attributes are stored in support files. Access to create and customize support files is provided by the *Objects menu*.

Components are connected if their nodes overlap or if a connection is drawn between the nodes. To draw a connection between nodes, click on a node with the left mouse button. A line is drawn between that node and the mouse cursor. Click the left mouse button again to place the connection (clicking the right button cancels the operation). The gridsnap facility helps overlapping the nodes. Connected nodes are given the same name by the *Make Names* and *Make File* options in the ATP menu. Nodes can be attached along a connection as well as at connection end-points. A connection should not unintentionally cross other nodes (what you see is what you get). A warning for node naming appears during the ATP-file creation if a connection exists between nodes of different names, or if the same name has been given to unconnected nodes. Connections can be selected as any other objects. To resize a connection, click on its end-point with the left mouse button, hold down and drag. If several connections share the same node, the desired connection to resize must be selected first. Selected connection nodes are marked with squares at both ends of the selection rectangle.

Relations are used to visualize information flow into Fortran statements and are drawn as blue connections, but have no influence on components connectivity. Relations are drawn in the same way as drawing a short circuit connection between nodes, except that you have to select the *TACS | Draw relation* option in the component selection menu to start the relation drawing. You can then draw multiple relations until you click the right mouse button or press *Esc* key.

3.5 Your first circuit *(Exa_1.adp)*

This chapter describes how to use ATPDraw step by step. As an example, composing the circuit file of a single-phase rectifire bridge (see Fig. 3.2) is presented. Reading this tutorial carefully, you will be proficient in the use of the most important ATPDraw functions, such as:

- How to select and assemble components?
- How to perform edit operations and give data to components?
- How to give node names, draw connections and specify grounding?
- How to create the ATP input file and perform the simulation?

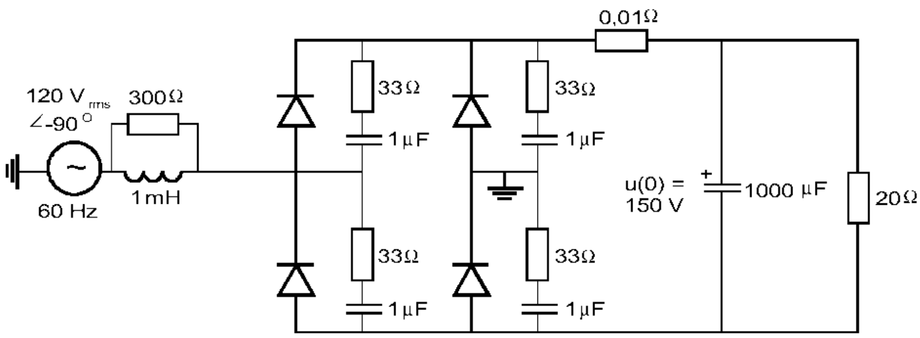


Fig. 3.2 – Single-phase rectifier bridge.

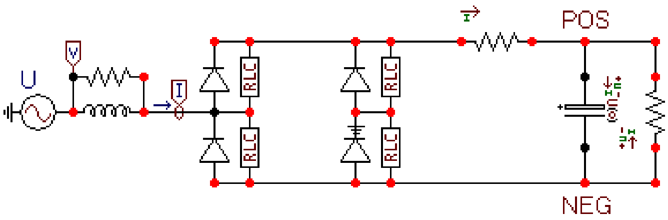


Fig. 3.3 – Your first circuit (`Exa_1.adp`).

The circuit is a single-phase rectifier bridge, supplied by a 120 V_{rms}, 60 Hz source. The source inductance is 1 mH in parallel with a damping resistor of 300 Ω. The snubber circuits across the rectifying diodes have a resistance of 33 Ω and a capacitance of 1 µF. The smoothing capacitor is 1000 µF and the load resistor is 20 Ω. The example has been taken from [2], exercise 1.
The units given in Fig. 3.2 are based on settings of Xopt and Copt equal to zero as will be explained later.

The circuit in Fig. 3.3 has been chosen since its construction involves the most commonly used edit operations.

3.5.1 Building the circuit

Most parts of the building process will be demonstrated in this chapter, along with the explanation of correcting possible drawing errors. The normal mode of operation is *MODE : EDIT*. You must always be in this mode to be able to select and specify data to objects. To return to EDIT from other modes, press *Esc*.

3.5.1.1 Starting to create a new circuit

Selecting the *New* command in the *File menu* or pressing the new (empty) page symbol in the *Component Toolbar,* a new circuit window will be created.

3.5.1.2 Source

First, an AC source is selected from the floating *Component selection menu*, which appears with a right mouse click on open area of the circuit window. Fig. 3.4 shows how to select a 1-phase sinusoidal voltage source (*Sources* | *AC type 14*) using the mouse.

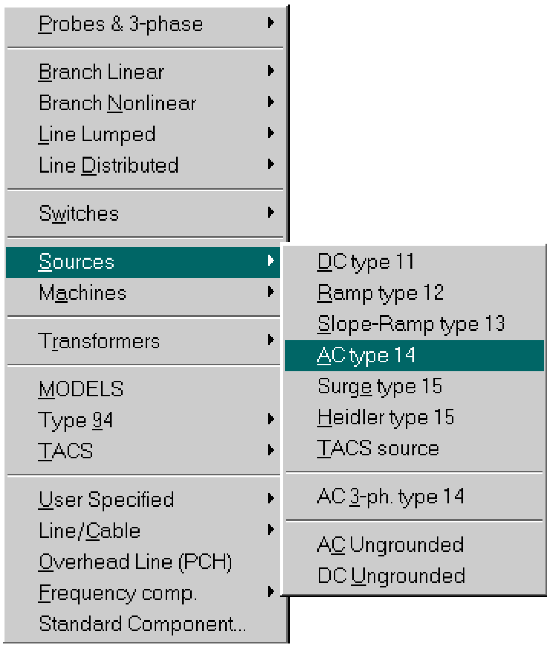
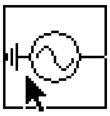


Fig. 3.4 - Selecting an AC source.

After you have clicked in the *AC type 14* field, the selected source appears in the circuit window enclosed by a rectangle.



Click on it with the **left mouse button**, hold down and drag it to a desired position. Then click with the left mouse button on open space to place it. The AC object is redrawn in red color as an indication that no data have been given to the object.

Next select the source inductance as shown in Fig. 3.5:

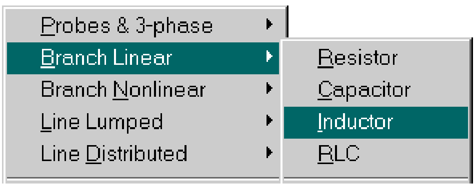


Fig. 3.5 - Selecting an inductor.

After you have clicked in the *Inductor* field, the selected inductor appears in the circuit window enclosed by a rectangle (from version 3.2 an optional, parallel damping resistance is included, specify data `Kp`=0 to remove it). Click on it with the left mouse button, hold down and drag it to a position shown in Fig. 3.6:

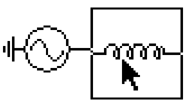
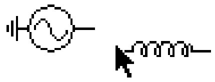


Fig. 3.6

Click on the white space with the left mouse button to place the inductor (the enclosing rectangle disappears). A grid snap facility helps you to place the inductor in the correct position. The component position is rounded to the nearest 10th pixel. (The included parallel resistor is shown in a dotted style.)

The inductor in Fig. 3.6 should be placed so that the node of the inductor touches the source. Objects having overlapping node dots will automatically be connected.

The next figure shows a situation where the inductor has been misplaced. In this situation the objects are disconnected. To correct this, a connection could be drawn between the objects as will be explained later. In this example you are supposed to place the inductor so that its left node overlaps the AC source node. To move the inductor, follow the instructions given in Fig. 3.7.



Click on the object with the left mouse button, hold down and drag it to the proper position, then click on white space.

Fig. 3.7 - Error!

When you have placed the inductor, you can add the damping resistance (really directly included from version 3.2). After you have clicked in the *Resistor* field of the component selection menu, a resistor icon appears enclosed by a rectangle. Click on it with the left mouse button, hold down and drag it to a position shown in Fig. 3.8. Click in open space to place/unselect it.

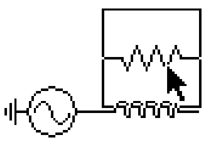


Fig. 3.8

This resistor is supposed to be parallel with the inductor and connections, which ensure this will be drawn later. The resistor in Fig. 3.8 would also be recognized as in parallel with the inductor, if it had been placed in a position completely overlapping the inductor. This tricky way is not recommended however, since the readability of the drawing is strongly reduced.

We want to measure the source current flowing into the diode bridge. To be able to do so, you can add a measuring switch. A special multi-phase current probe is available for such measurements in ATPDraw. When using this object, you are requested to specify the number of phases and in which phases the current should be measured. Select the probe as shown in Fig. 3.9.

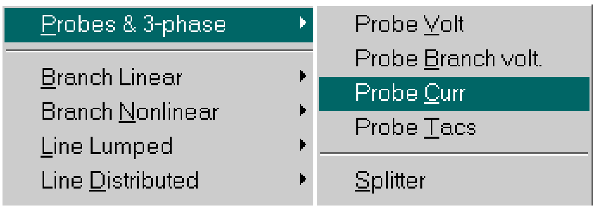
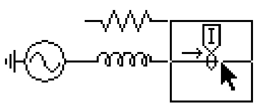


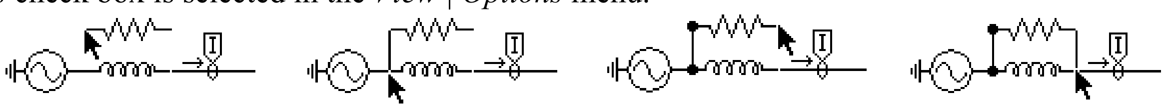
Fig. 3.9 - Selecting a current measuring probe.



After you have clicked in the *Probe Curr.* field, the selected probe appears in the circuit window enclosed by a rectangle. Click on it with the left mouse button, hold down and drag it to a position shown in the figure, then place it.

At this stage of the building process, it is time to draw some connections in the circuit diagram. To draw a connection you just click the left mouse button on a node, release the button and move the mouse. The cursor style now changes to a pointing hand and a line is drawn between the starting position and the current mouse position (the action mode now is *MODE : CONN.END* indicating that the program is waiting for the end point of the connection). Click with the left mouse button again to place the connection or click with the right button to cancel the starting point.

Two connection drawings are required to parallel connecting the source inductance and the damping resistor as shown below. The connections are always drawn with node dots if the *Node dots* check box is selected in the *View* | *Options* menu.



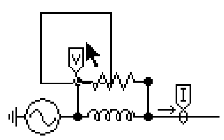
Click left
Release + move

Click left

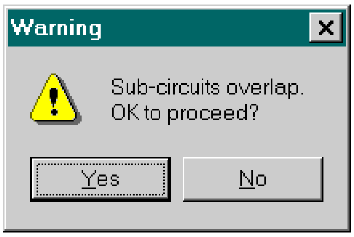
Click left
Release + move

Click left

The last object we want to introduce in the source part of the circuit is a voltage measuring probe, which results in an output request for the node voltage in the ATP input file. The voltage sensor can be selected via the *Probe & 3-phase* | *Probe Volt* in the component selection menu (see Fig. 3.9). The probe is drawn in the middle part of the circuit window in marked and moveable mode.



Use the left mouse button to drag and place the object as shown on the figure to the left. When you place an object by clicking on open area of the circuit window, you will sometimes receive a warning message as shown in Fig. 3.10.



This message appears if a center of one of the permanent objects is inside the enclosing polygon of a marked object (or more general; a group of objects). This is to prevent unintentional object overlap if the left mouse button were pressed while moving the object.

Fig. 3.10 - Prevent object overlap.

If you click on *No*, the object not placed but continues to be selected and you can move it further. Normally it is OK to click on *Yes*. If you change your mind later, the *Edit* | *UNDO* option provides an easy way to return to an earlier version of the circuit.

The circuit objects used in the circuit so far were drawn in red color. This tells you that no data have been given to these objects. You can give data to objects at any time during the building process. We will now give data to the objects in the source part of the rectifier. To do so: simple click with the **right mouse button** (or double click with the left button) on an object. If you right click on the AC source icon, a window as shown in Fig. 3.11 appears.

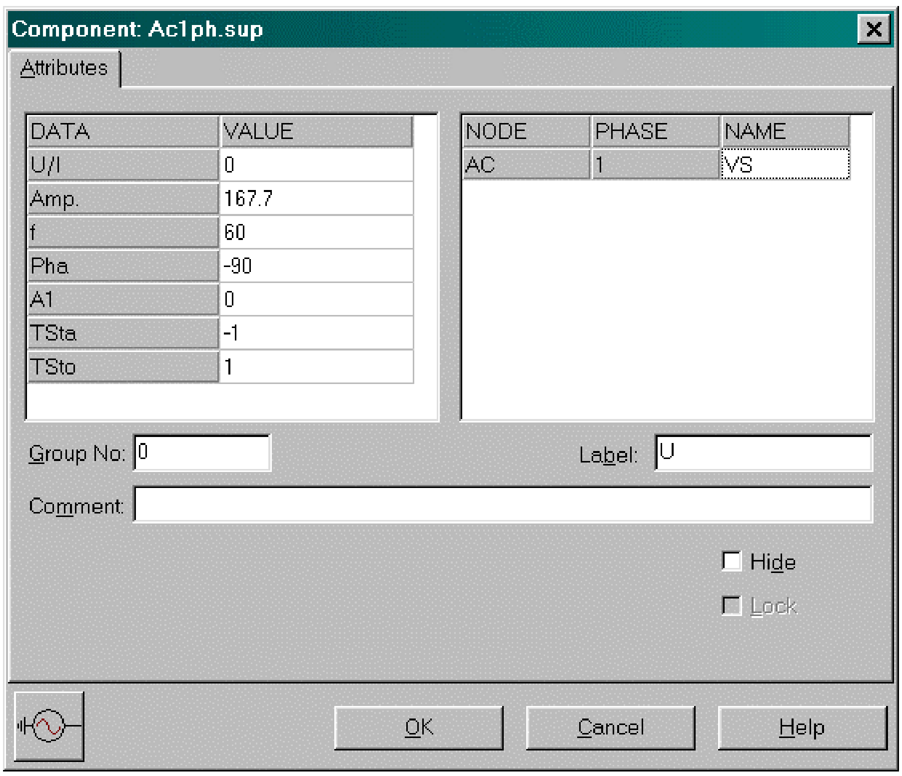


Fig. 3.11 - Component dialog box of the single-phase sinusoidal source.

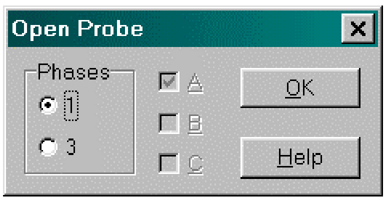
Data values shown in Fig. 3.11 refer to the circuit parameters of Fig. 3.2. The name of the numerical fields is identical with that of used by the ATP Rule Book [3] for an AC source. This AC source has 7 input data and one node. Click on the *HELP* button to load the on-line help if you are unsure about the meaning of parameters. Specify data as shown in Fig. 3.11. Here

U/I = 0 results in voltage source with default label U,
U/I = -1 results in a current source with label I.

The node names can **not** be specified in this window. Click *OK* to close the window and update the object values. Click on *Cancel* to just quit the window.

After you have given data to the AC source and closed the window (note how the object layout changes when you exit the window), proceed to the other objects. Repeat the procedure explained above to give data to the resistor and to the inductor by calling the *Component* dialog box of the objects. To do so: click with the right mouse button on the resistor and inductor icon, respectively.

The probe objects have different input window than other objects.

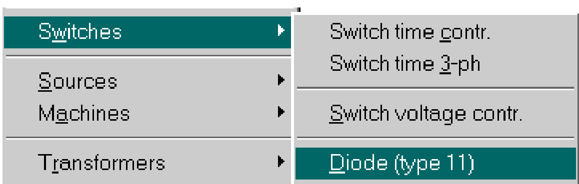


To open the voltage or current probe input window, click on its icon with the right mouse button. In this window, you can select the number of phases of the probe and which phases to monitor. In this single-phase example, default values (no. of phases=1, monitored phase=A) of both voltage and current probes should be selected, as shown in Fig. 3.12

Fig. 3.12 - Open probe dialog box.

3.5.1.3 Diode bridge

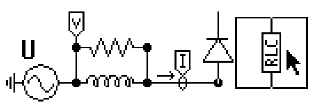
In this process, you will learn how to use some editing options like rotate, group, duplicate and paste. Since the diode bridge consists of four equal branches, you do not need to build all of them from scratch. First, you select a diode from the selection menu as shown in Fig. 3.13. After you have clicked on *Diode (type 11)* the diode appears in the circuit window enclosed by a rectangle.



The diode has to be rotated so click the right mouse button or select *Edit* in the main menu and click on *Rotate*. The diode is now rotated 90 deg. counter clock-wise. Click on the diode with the left mouse button, hold down and drag to the position shown in Fig. 3.14.

Fig. 3.13 - Selecting a diode.

Click with the left mouse button on empty area to place the diode. Remember the grid snap facility and the overlap warning.



Next, you select the snubber circuit across the diode. In this example the snubber circuit is a resistor and a capacitor in series. Select an RLC object from the component selection menu (Fig. 3.5).

Fig. 3.14

Click on the selected RLC branch with the right mouse button to rotate, then click with the left button, hold down and drag the RLC branch to be in parallel with the diode. Click on the left mouse button to place.

The idea is further to copy the diode and the RLC branch, but before doing so, it is wise to give data to them (since the data are kept when copied). A simple click on the RLC or diode icon with the right mouse button activates the component dialog box to give data to objects.

Again, an explanation of the input parameters is given in a help file. Press the *HELP* button to see this help file. The numerical values of the diode are all zero, meaning that the diode is ideal and is open during the steady state. The RLC branch in Fig. 3.14 has been given a resistance of 33 Ω and a capacitance of 1 μF.

You have now given data to the diode and the RLC branch and instead of repeating the process four times, you can use the copy facility. First, you have to select a group of components. This can be done by selecting *Edit* | *Select group* field in the main menu or with a double click with the left mouse button on an empty space of the *Circuit window.* Then cursor style changes to a pointing hand and the action mode is *EDIT : GROUP*. The process is then to click with the left mouse button to create a corner in a fence and to click the right button to enclose the fence (polygon). All components having their center inside the fence are included in the group.

Alternative way of group selection is to draw a rectangle around the objects by a left mouse click and hold at the upper-left corner of the desired rectangle, and moving thereafter to the lower-right corner. Objects inside the rectangle become a group when the mouse button is released.

You can follow the procedure shown in Fig. 3.15.

Fig. 3.15 - Drawing a polygon: First double click on white space, click the left mouse button at each corner of the polygon, then click the right button to enclose the polygon.

The group created in Fig. 3.15 can be copied/rotated etc. like a single object. Now we want to duplicate this group. Click on the main menu *Edit* field and choose *Duplicate* or press the *Ctrl+D* shortcut key. The selected group is copied to the clipboard and pasted in the same operation. The old group is redrawn in normal mode and the copy is drawn in the top of the original.

The fencing polygon is now a rectangle. The pasted group is moveable, so you can click on it with the **left mouse button**, hold down and drag to a desired position. Click the left mouse button on open space to put the group in the position shown in Fig. 3.16.

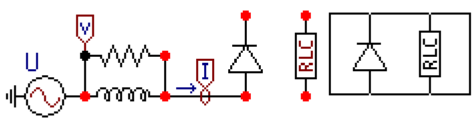


Fig. 3.16 - Move a group.

Only the enclosing polygon is drawn during a move operation. The objects are drawn when the mouse button is released. If you misplaced the group you can mark it again with *Edit* | *Select group*. *Undo* and *Redo* facilities are also available on the main menu *Edit.*

You can now paste a second copy of the diode/RLC group into the circuit. Since the duplicate facility has already copied the group to the clipboard, you can just select the *Paste* option from the *Edit* menu by using the mouse or pressing *Ctrl+V*, or selecting the *Paste* icon from the *Toolbar*. The pasted group is drawn on top of the original one enclosed by a rectangle. Click on this group with the left mouse button, hold down and drag it to a position shown in Fig. 3.17.

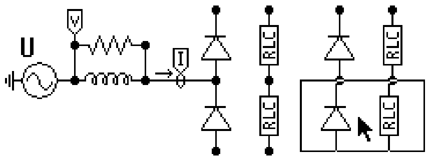


Fig. 3.17

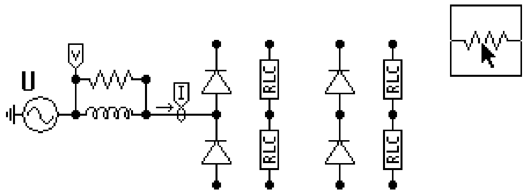


Fig. 3.18

As part of the connection between the rectifier bridge and the load a small resistor is included in Fig. 3.2. The resistor is included to demonstrate the option of using a small resistor for current measurement purposes.

Select a resistor in the component selection menu, then click on the resistor with the left mouse button, hold down and drag it to a desired position as shown in Fig. 3.18. You must place the resistor precisely, because the next step is to connect the top nodes of the diode bridge with the resistor.

Before doing so first, give data to this resistor opening the component dialog box by a right-click on the resistor. Specify data value RES= 0.01 Ω and set *Output* to *1-Current* to get current output in the subsequent ATP run. Having closed the component dialog box a small $\vec{\imath}$ symbol appears on the top-left side of the resistor indicating the current output request (if not check *View* | *Options*).

Now you can start to connect the diode bridge and the resistor together. The procedure is to first click with the left mouse button on a starting node, as shown in Fig. 3.19. The cursor style now changes to a pointing hand and the action mode is *MODE : CONN.START*. Then release the mouse button and move the mouse (a rubber band is drawn from the starting point to the current cursor position). To place a connection, click on the left mouse button again. Click on the right button or press *Esc* to cancel the connection make operation.

The connection draw in Fig. 3.19 picks up intermediate nodes so all the five nodes will be connected together. In this way, ATPDraw suits the requirement: "What you see is what you get" and the amount of required connections are significantly reduced.

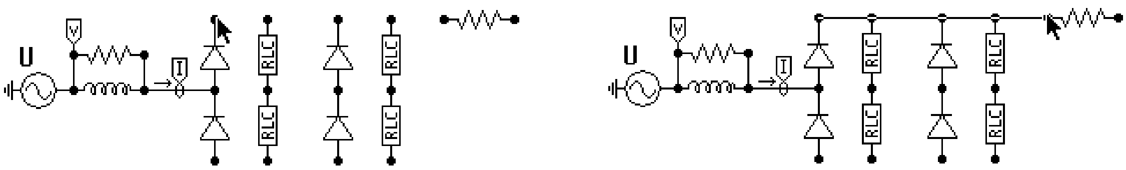


Fig. 3.19 - Click left button. Release + move, then click left button to place the connection.

If you made a mistake in the connection drawing process, you can correct the error easily, because connections are editable (copy/move/rotate) as any other objects. If you would like to correct/modify a misplaced connection, click on it with the left mouse button. After this selection, the connection is enclosed by a rectangle and two squares replace node dots at the end of the line. To move the connection, click on an internal point of it using the left mouse button, then hold down and move, and release the mouse at the correct position. To reposition a connection, click on the node squares with the left button and stretch the connection as illustrated in Fig. 3.20:

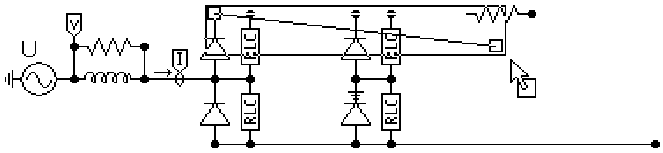


Fig. 3.20 - Edit connection. Click any point of the line then click node squares and stretch.

3.5.1.4 Load

The last part of this example circuit is the load consisting of a smoothing capacitor with initial condition and a load resistor. First, you can select the capacitor as shown in Fig. 3.21:

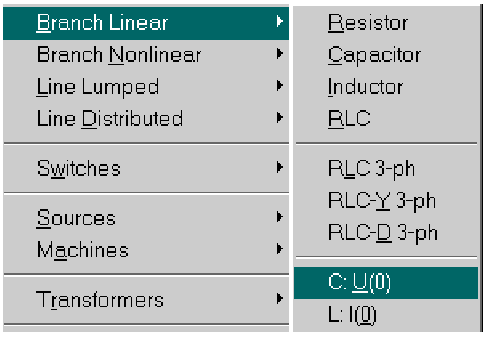
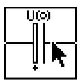


Fig. 3.21 - Select capacitor with initial condition.



After this selection, the capacitor appears in the middle of the circuit window in moveable mode enclosed by a rectangle. Click on the capacitor with the left mouse button, hold down and drag to a desired position, then click the right mouse button 3 times (or press *Ctrl+R*) to orient the capacitor as shown in Fig. 3.22. Finally, click on open space to place the capacitor.

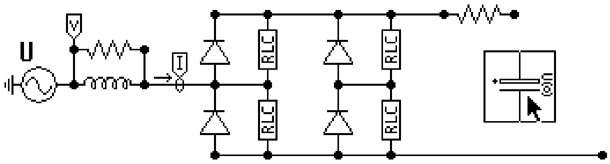


Fig. 3.22 - Placing a capacitor with initial conditions.

Next select the load resistor in the component selection menu *Branch linear + Resistor*. The resistor is drawn in moveable mode in the circuit window. Select *Edit + Rotate* to rotate the resistor. Click on it with the right mouse button to rotate, then click with the left mouse button, hold down and drag it to a desired position and place as shown in Fig. 3.23.

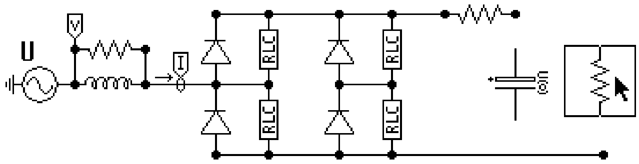


Fig. 3.23 - Place load resistor.

The time has come to connect the load to the rest of the diode bridge. The process has been explained before. Click on the component nodes you wish to connect with the left mouse button, sequentially. A left mouse click on open area while in *MODE: CONN.END* generates a new node dot, which can be used as the starting point of any new connections. This way creating a circuit having only perpendicular connections (recommended for complex circuits, to improve the circuit readability) is a relatively simple task, as shown in Fig. 3.24.

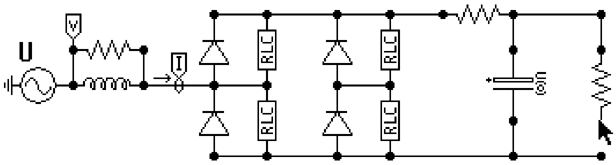


Fig. 3.24 - Your first circuit is almost ready!

After you have finished connecting the source side and the load side of the circuit, you can specify the load data. Click with the right mouse button on the capacitor and specify the parameters shown in Fig. 3.25:

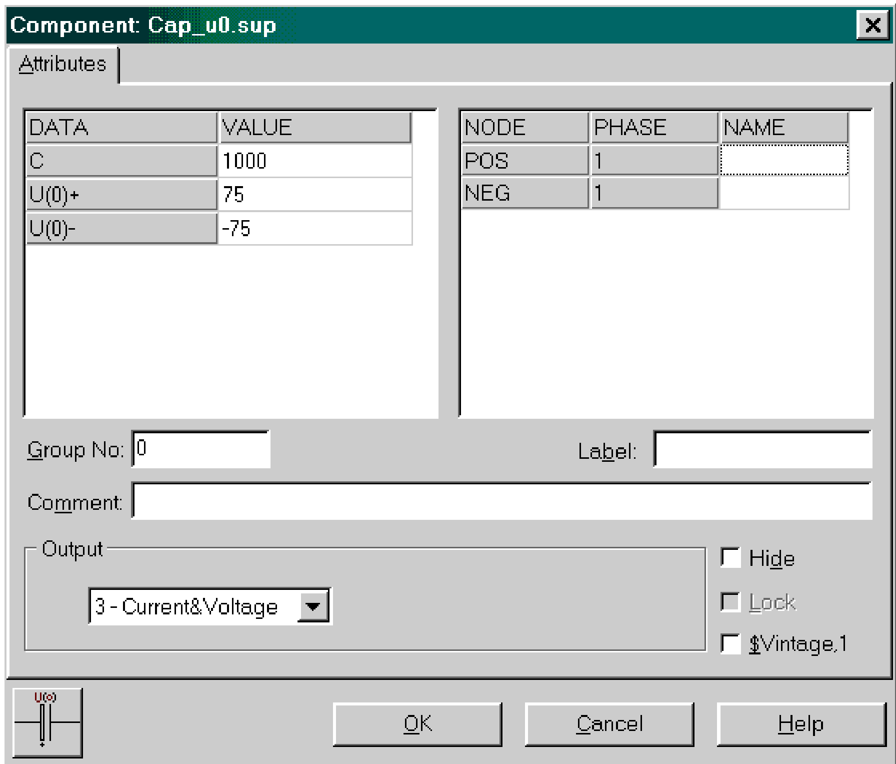


Fig. 3.25 - Capacitor data with initial condition.

The capacitance is 1000 µF (if Copt=0 in *ATP* | *Setting* | *Simulation*). The positive node has an initial voltage of 75 V and the negative -75 V. Both branch current and voltage will be calculated, so the *Current&Voltage* is selected in the *Output* combo box. Following the branch output request, the appearance of the object's icon will change if the *Show branch output* is checked under *View* | *Option*. If this option is enabled, a small $\overset{+U-}{\rightarrow I}$ symbol appears on the top-left side of the capacitor, indicating the branch voltage and the current output requests (see Fig. 3.29).

Next click with the right mouse button on the load resistor to get the input window and specify the load resistance of 20 Ω. Branch current and voltages will be calculated so the small $\overset{+U-}{\rightarrow I}$ symbol appears again on the top-left side of the resistor after leaving the dialog box. Once all the entries in the component dialog box are completed, select the *OK* button to close the window and update the object values or click *Help* to obtain an on-line help.

3.5.1.5 Node names and grounding

The final step of building this circuit is to give data to nodes (node names and grounding). All nodes will automatically receive names from ATPDraw, so the user should normally **give names to nodes of special interest** only. It is advised in general to perform the node naming as the last step in building up a circuit. This is to avoid undesirable multiple node names (which is corrected by ATPDraw automatically, but results in irritating warning messages).

To give data to a node, you simply have to click on this node once with the right mouse button. Fig. 3.26 - Fig. 3.29 shows how to give data to four different nodes

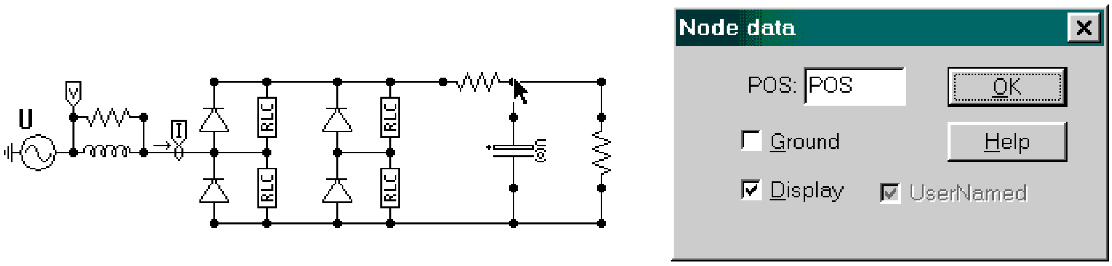


Fig. 3.26 - Click on a node with the right mouse button and specify a name in the dialog box.

When you exit the window in Fig. 3.26 by clicking *OK*, the circuit is updated as shown in Fig. 3.27. All node names are forced left adjusted, and as a general rule in the ATP simulation, capital letters should be used. ATPDraw does accept lower case characters in the node data window, however this “feature” should be avoided, in particular if the node is connected with electrical sources.

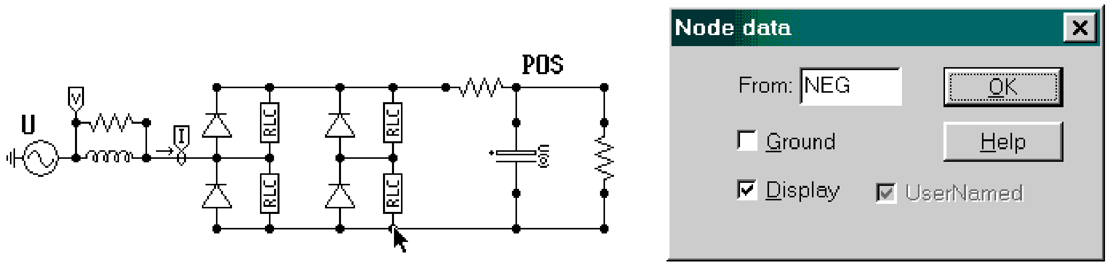


Fig. 3.27 - Click on a node with the right mouse button and specify a name in the node data window. The name ‘NEG’ will be assigned to all nodes visually connected.

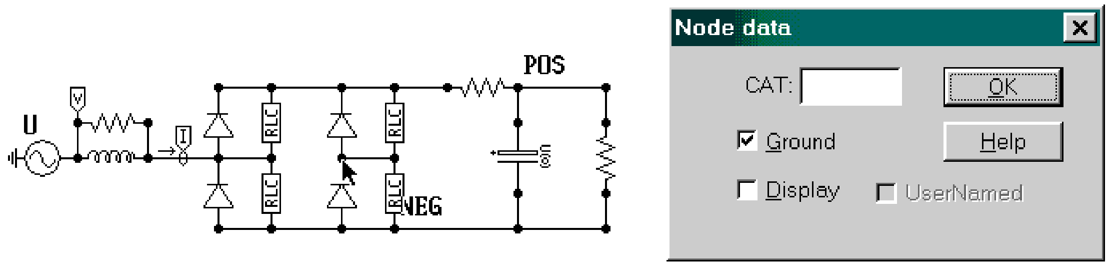
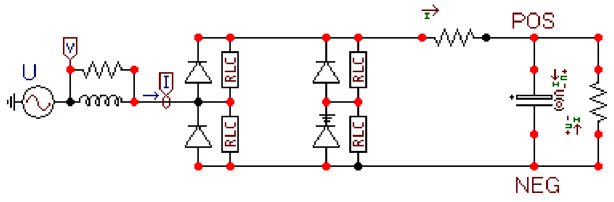


Fig. 3.28 - Click on a node with the right mouse button and check the *Ground* box indicating that the node is connected with the ground reference plane of the circuit.

The ground symbol is drawn at the selected node when you exit the window as Fig. 3.29 shows. The nodes not given a name by the user will automatically be given a name by ATPDraw, starting with `XX` for single phase and `X` for 3-phase nodes followed by a four-digit number. Nodes got the name this way (i.e. from the program) are distinguished by red color from the user specified node names, which are drown black as shown in Fig. 3.29.



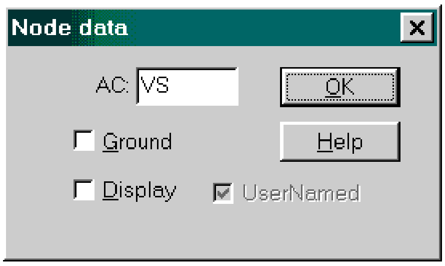


Fig. 3.29 - Click on the voltage source with the right mouse button and specify the node name.

3.5.2 Storing the project file on disk

You can store the project in a disk file whenever you like during the building process. This is done in the main menu with *File* | *Save* (or *Ctrl+S*). If the current project is new, a *Save As* dialog box appears where you can specify the project file name and location on the disk. Two different styles of the *Save As* dialog boxes are available, depending on the *Open/Save dialog* setting in the *Tools* | *Options* | *General* menu: a Windows 9x standard dialog box and a Windows 3.1 style. The default extension is *.adp* in both cases and it is automatically added to the file name you enter.

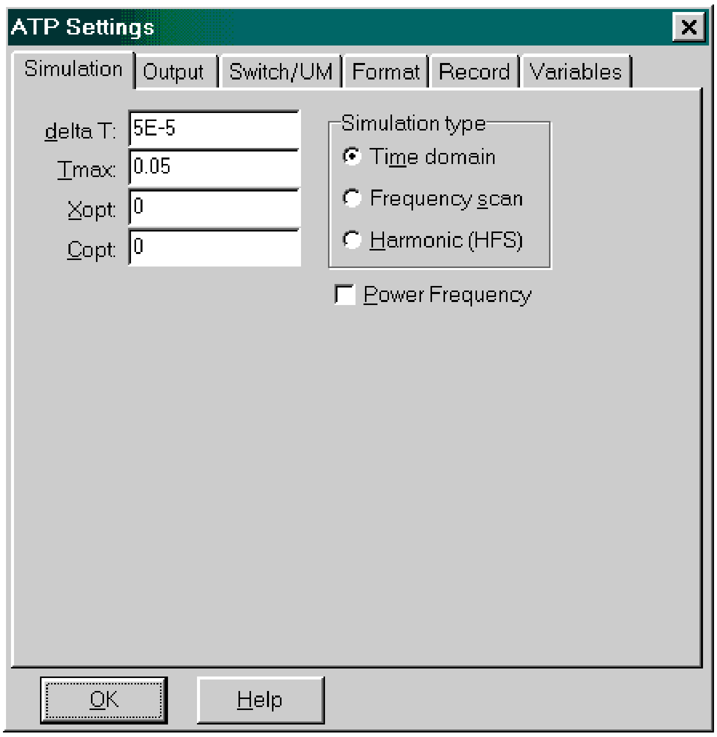
When the circuit once was saved, the name of the disk file appears in the header field of the circuit window. Then if you hit *Ctrl+S* or press the *Save circuit* icon in the Toolbar, the circuit file is updated immediately on the disk and the *Modified* flag in the status bar disappears. The *File + Save As* option or the *Save As* Toolbar icon allows you to save the circuit currently in use under a name other than that already allocated to this project.

3.5.3 Creating the ATP input file

The ATP-file describes the circuit according to the ATP input file syntax. You can create this file by selecting *Make File As* command in the *ATP* main menu. The ATP-file is regenerated whenever you just execute the *run ATP* command (or press *F2*). In the latter case the process is hidden for the user.

However, before you create the ATP input file or run the simulation, you *must not* forget to specify the miscellaneous parameters (i.e. parameters, that are printed to the Misc. Data card(s) of the ATP input file). The default values of these parameters are given in the `ATPDraw.ini` file. Changing these default values can either be done in the *ATP* | *Settings* | *Simulation* sub-menu for the current project, or under the *Tools* | *Options* | *View/ATP* | *Edit settings* | *Simulation* for all new ATPDraw projects created henceforth.

Fig. 3.30 shows an example of the 1st miscellaneous data card settings of an ATP simulation (specifying time step, time scale of the simulation etc.). This window appears if you select the *Simulation* tab of the *ATP* | *Settings* menu.



Select:

- Time step *delta T* in sec.
- End time of simulation *Tmax* in sec.
- Xopt=0: Inductance in mH.
- Copt=0: Capacitance in µF (mikro-F!).

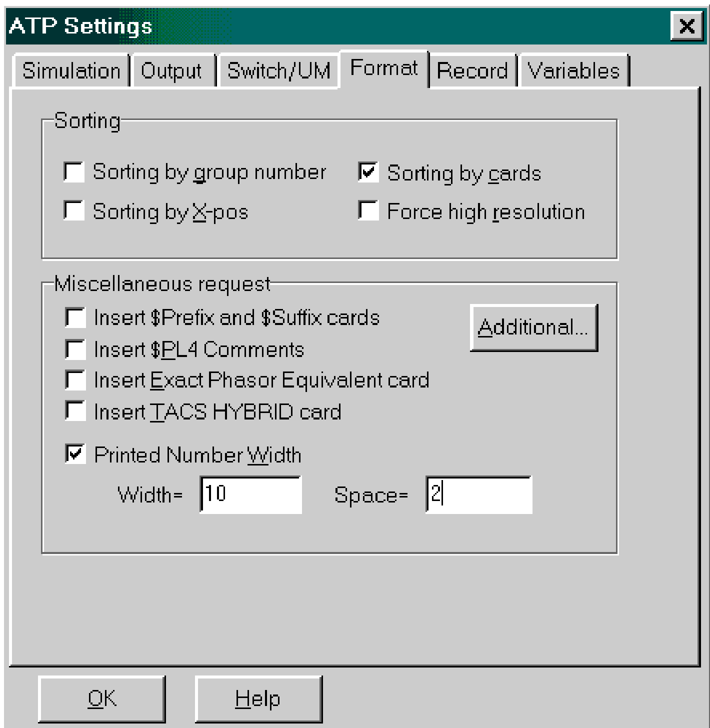
The main characteristic of the simulation (time domain or frequency scan) can also be set on this page.

Press *Help* to get more information or *OK* to close the dialog box.

The simulation settings are stored in the project file, so you should save the file after changing these settings.

Fig. 3.30 Simulation settings.

Values on the first integer miscellaneous data card of ATP can be changed under the *ATP* | *Settings* | *Output* page. The next *ATP* | *Settings* | *Switch/UM* tab is the home of control flags required by statistical switching or universal machine simulations.



Under the *Format* page the user can select precision mode and the ATP-file sorting criteria. If you select the *Format* page, the window shown in Fig. 3.31 appears:

Select:

☑ Sorting by cards: First /BRANCH, then /SWITCH and then /SOURCE.

☑ Printed Number Width request is enabled. *Width* is the total column width of ATP printed output LIS-file, *Space* is the number of blanks between columns.

All other check boxes are unselected

Fig. 3.31 - The ATP-file format menu.

To create an ATP-file without starting the simulation you must select the *Make File As* in the *ATP* menu. This selection will start a procedure, which examines your circuit and gives node names to circuit nodes. Then a standard Windows' *Save As* file window appears, where you can specify the name and path of the ATP-file. The same name as the project file with extension `.atp` is recommended. You can edit this file or just display it by selecting the *ATP | Edit ATP-file* menu. The ATP-file (`Exa_1.atp`) you have just created will look like as follows:

```
BEGIN NEW DATA CASE
C ---------------------------------------------------------
C Generated by ATPDRAW  December, Saturday 1, 2001
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2001
C ---------------------------------------------------------
PRINTED NUMBER WIDTH, 10, 2,
C Example 1
C Your first circuit
C Rectifier bridge
C  dT  >< Tmax >< Xopt >< Copt >
   5.E-5     .05
     500       1       1       1       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
        XX0031               33.          1.                                   0
  NEG                        33.          1.                                   0
  XX0031POS                  .01                                               1
  POS   NEG                             1.E3                                   3
  NEG   POS                  20.                                               3
  VS    XX0021                      1.                                         0
  VS    XX0021              300.                                               0
  NEG   VA                   33.          1.                                   0
  VA    XX0031               33.          1.                                   0
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
11VA    XX0031                                                                 0
11      XX0031                                                                 0
11NEG   VA                                                                     0
11NEG                                                                          0
  XX0021VA                                            MEASURING                1
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14VS     0     167.7       60.      -90.                           -1.        1.
/INITIAL
 2POS               75.
 2NEG              -75.
 3POS   NEG                             150.
/OUTPUT
  VS
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

3.5.4 Running the simulation

Starting the ATP simulation is supported in ATPDraw in a user friendly way. The user just has to press *F2* function key to create an ATP input file with the current project file as input and run the simulation. The default command that is executed when the user selects *run ATP* under the *ATP* menu or when the *F2* is pressed can be specified under the *Tools | Options /Preferences* tab as it has been described in section 2.6 of the Installation Manual.

3.6 Three phase circuits *(Exa_2.adp)*

Both single-phase circuits and three-phase circuits are available in ATPDraw. For 3-phase objects, the number of phases is indicated in the selection menu. The icon border of the 3-phase objects is generally thick or gray shaded and the icon consists of thick lines from the nodes into the object symbol.

All three-phase nodes have only 5 characters available in the *Node data* windows. ATPDraw adds the extension *A*, *B* and *C* at the end of the node name. By default, the phase sequence is *ABC*; the first data card uses *A*, the second *B* and the last *C*. The only way to change the phase sequence is to use the available transposition objects (`Transp1 - Transp4`) selectable under *Probes & 3-phase* in the component selection menu.

The current phase sequence is displayed in the bottom of the node input window after an *ATP | Make Names* or *Make File As* selection has been made. The following example illustrates the usage of three-phase objects:

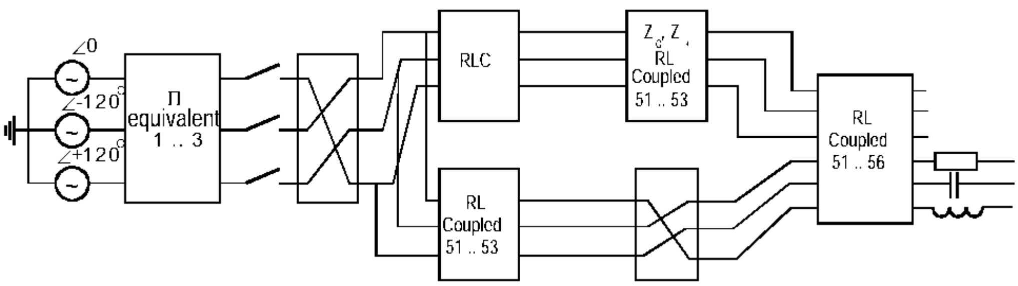


Fig. 3.32/a - Illustrative three-phase circuit.

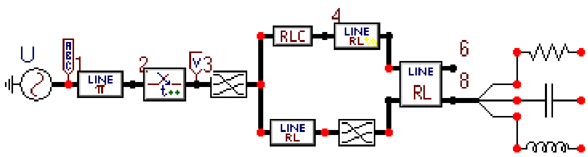


Fig. 3.32/b - Equivalent ATPDraw circuit (`Exa_2.adp`).

The circuit shown in Fig. 3.32 was built up in the same way as your first circuit. You can note that connections between the three phase nodes appear to be thick. The circuit contains 3 special objects, the already mentioned transposition object (in this case from *ABC* to *BCA*), a splitter object, which splits three phase nodes into three single-phase nodes and an *ABC reference* object.

Names 1 - 8 have been given to nodes in the circuit. By selecting *Make Names* under *ATP* in the main menu, ATPDraw examines the circuit and creates unique node names.

If you click with the right mouse button on nodes after selecting *ATP | Make Names* you are able to see the phase sequence in the bottom of the node input window as shown in Fig. 3.33/a and Fig. 3.34/b.

The single phase nodes do not have a phase sequence, but the single phase side of a splitter object has one, as shown in Fig. 3.34/b.

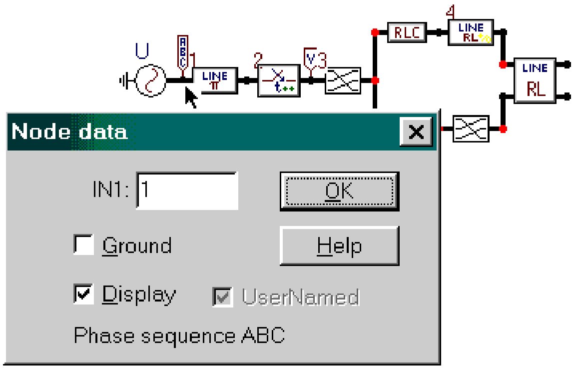


Fig. 3.33/a - Click right button on node *1*.

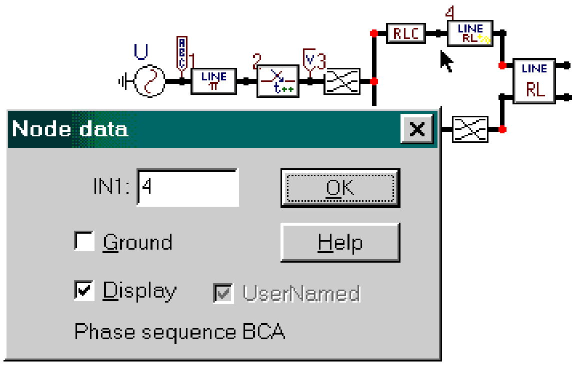


Fig. 3.33/b - Click right button on node *4*.

The node names in Fig. 3.33/a are *1A*, *1B* and *1C*, all left adjusted. The node names in Fig. 3.33/b are *1B*, *1C* and *1A*, all left adjusted. ATPDraw gives the phase sequence *ABC* to the sub-network left of the first transposition object, that can be overridden by connecting an *ABC reference object* to any 3-phase nodes of the circuit.

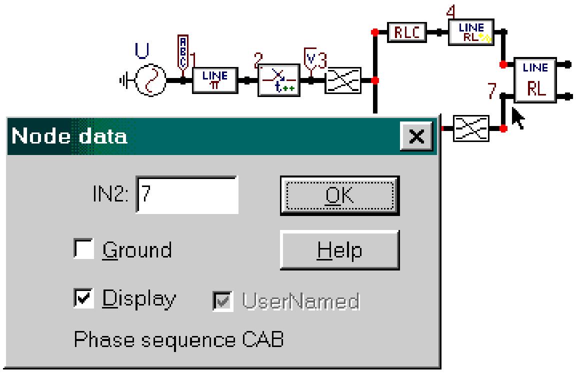


Fig. 3.34/a - Click right button on node *7*.

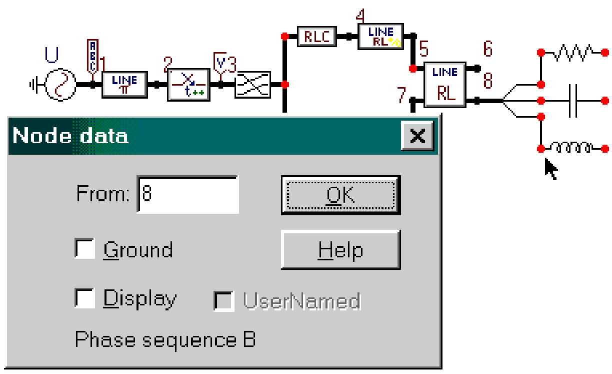


Fig. 3.34/b - Click right on single-phase node *8*.

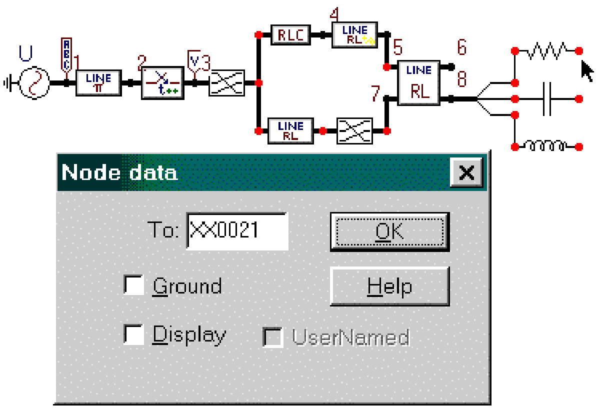


Fig. 3.35 - Click the right mouse button on single-phase right node of the resistor.

A node name is not assigned to the right node of the resistor (Fig. 3.35). ATPDraw thus gives the node a name starting with XX followed by a unique number. This node is a single-phase node with no phase sequence. 3-phase nodes with no user specified names are given a name starting with X, followed by a four digit number and ending with the phase sequence letters *A*, *B* and *C*.

Some special restrictions apply to the splitter objects (found under *Probes & 3-phase* in the component selection menu):

- Connecting splitter objects together on the 3-phase side or with connections on the 1-phase side is permitted (earlier it was illegal), but transposition/disconnection is not allowed.
- If the name *NODEA* is given to what you know is phase *A* on the single phase side, ATPDraw does not accept this and adds its own *A* at the end, creating the node name *NODEAA*. The general rule is that ATPDraw takes care of the phase sequence alone!
- Best solution is to specify a node name on the 3-phase side only.

The ATP data file created by ATPDraw from the circuit in Fig. 3.32/b is shown below:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Tuesday 30, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-6    .001
     500       1       1       1       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
1 1A    2A                  10. .0001    1.
2 1B    2B                      .0001    1.   10. .0001    1.
3 1C    2C                      .0001    1.       .0001    1.   10. .0001    1.
  3B    4B                   1.  .001                                         0
  3C    4C                   1.  .001                                         0
  3A    4A                   1.  .001                                         0
514B    5B                   2.         1.5
524C    5C                   1.          1.
534A    5A
513B    7B                  10.          1.
523C    7C                               2.   10.          1.
533A    7A                               2.                2.   10.          1.
515B    6B                  10.          1.
525C    6C                               2.   10.         10.
535A    6A                               2.                2.   10.          1.
547C    8C                               2.                2.                2.
                            10.        1.
557A    8A                               2.                2.                2.
                                         2.   10.          1.
567B    8B                               2.                2.                2.
                                         2.                2.   10.          1.
  8C    XX0021             1.E3                                               0
  8A    XX0023                           1.                                   0
  8B    XX0025                    10.                                         0
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  2A    3A        .0001      .001                                             0
  2B    3B          -1.      .001                                             0
  2C    3C          -1.      .001                                             0
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
141A     0     1.5E5       60.                                 -1.        1.
141B     0     1.5E5       60.     -120.                       -1.        1.
141C     0     1.5E5       60.      120.                       -1.        1.
/INITIAL
/OUTPUT
  3A    3B    3C
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

4. Reference Manual . . .

. . . .

ATPDraw™
for Windows
3.5

This part of the manual outlines all menu items and program options, and gives an overview about the supported ATP objects, TACS, LCC components and MODELS features.

ATPDraw has a standard Windows user interface. The *Main window* of the program is shown in Fig. 4.1. The *Main menu,* the *Circuit window* and the *Component selection menu* are the most important items of that window. Elements of the *Main menu* and supported ATP components in the *Component selection menu* will be referenced in this part of the manual.

4.1 Main window

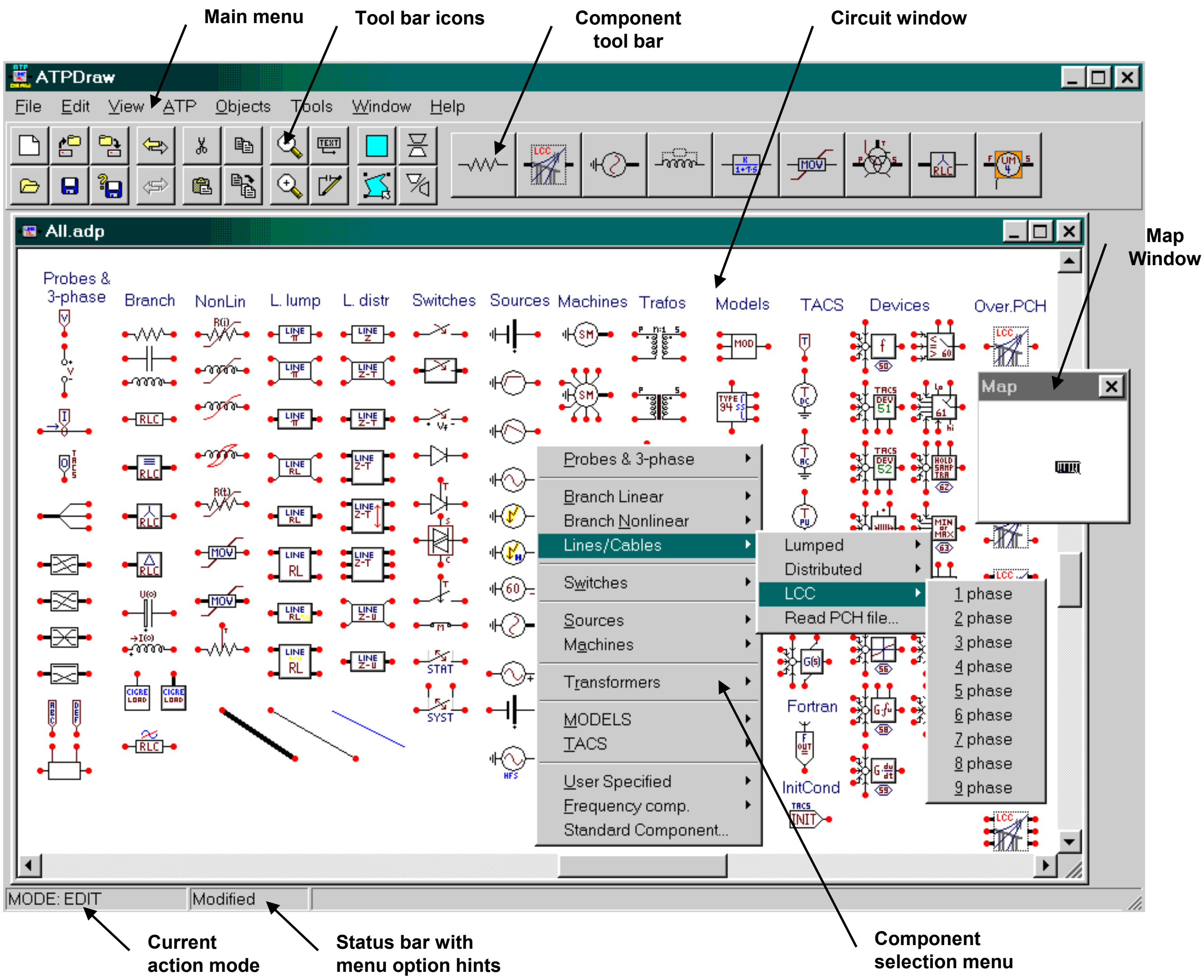
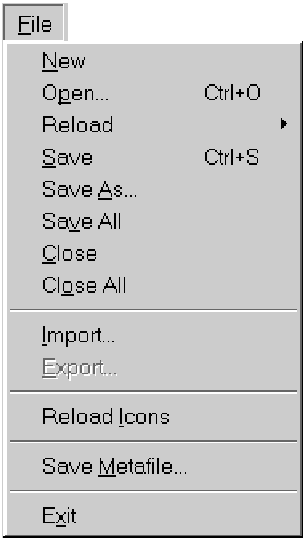


Fig. 4.1 - Components of ATPDraw's main window.

If you are unfamiliar with the use of ATPDraw, read the Introductory Manual to learn how to create a circuit or the Advanced Manual to learn how to create a new object in ATPDraw. The Introductory Manual starts with the explanation of operating windows and the mouse in ATPDraw, and shows how to build up a circuit and how to create an ATP-file to be used as input for a subsequent transient simulation.

4.2 Main menu

4.2.1 File



This field contains actions for input/output of ATPDraw projects. Selecting the *File* item in the main menu will result in a popup menu shown in Fig. 4.2.

Fig. 4.2 - File menu.

4.2.1.1 New

Selecting this menu item will open a new empty *Circuit window*. ATPDraw supports to work on several circuits simultaneously and copy information between the circuits. The number of simultaneous open windows is limited only by the available MS-Windows resources. The circuit window is much larger than the actual screen, as it is indicated by the scroll bars of each circuit windows.

4.2.1.2 Open

This menu performs a Windows standard Open dialog box shown in Fig. 4.3. In this window the user can select a project file and load it into the ATPDraw. Short key: *Ctrl+O*.

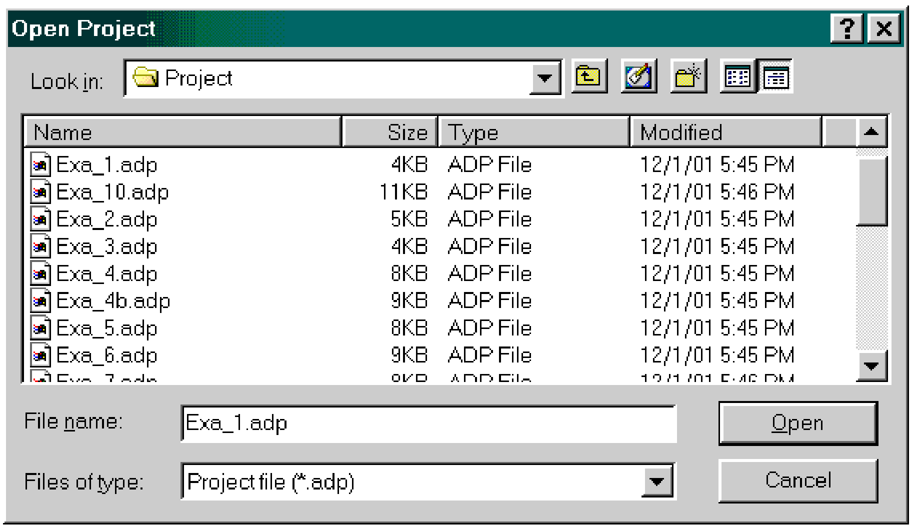


Fig. 4.3 - Open file window (Win9x style).

ATPDraw can read both circuit (*.cir*) files created by an earlier version of the program and project files (*.adp*). When opening a project file ATPDraw first unzips the files (e.g. *.lib*, *.mod, .alc, .bct* or *.sup* files) belong to the project and store them in the specified folders. If a folder name specified in the project file is unknown, the default folder will be used. If a file already exists on the target location with a different creation date/time, ATPDraw lets the user to choose which file to keep.

This Open/Save dialog box is used for several different selections in the main menu. An alternative MS-Windows 3.1 style is also supported as shown in Fig. 4.4. There is a check box in the *Tools* | *Options* | *General* tab to switch between the two supported alternatives.

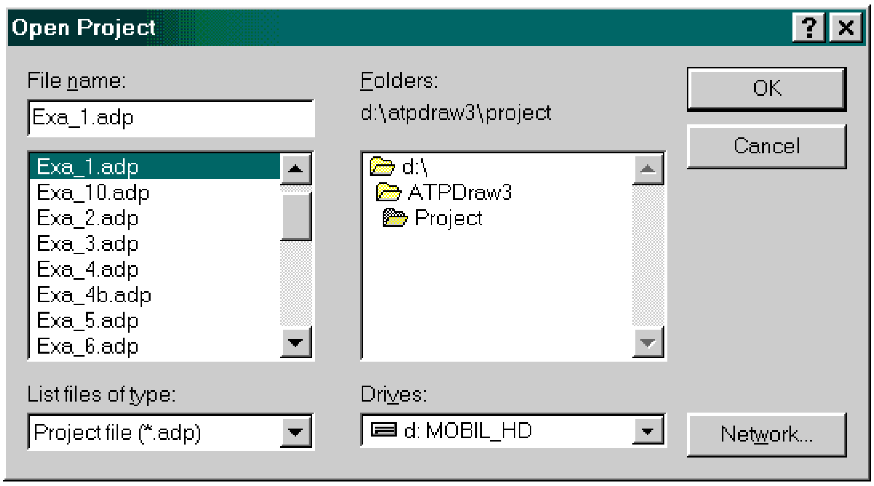


Fig. 4.4 - Open file window (Win 3.1 style).

The existing project files in the `\Project` subfolder of ATPDraw are shown below the *File name:* field. A specific file can be selected either by typing the name directly, or by a left mouse click in the file list. Clicking *OK* will perform the selection made and the file is loaded into a new circuit window. Clicking on *Cancel* will simply close the window.

4.2.1.3 Reload

The name and location of the recently used project files are listed right to this menu item. The user can select and load one of them into the circuit window by a simple mouse click.

4.2.1.4 Save

Activating this menu item will save the project in the active circuit window into a disk file. If the name `Noname.adp` is shown in the circuit window a *Save As* dialog box will be performed, where the user can specify a new name for the current project file name. Short key: *Ctrl+S*.

4.2.1.5 Save As

The project in the active circuit window is saved to disk under a new name. The name of the file can be specified in the *Save As* dialog, which is similar to the *Open Project* dialog in Fig. 4.3 or in Fig. 4.4. This command allows the user to save the project under a name other than that is already used. ATPDraw can read circuit files (*.cir*) created by earlier program versions, but the *Save As* command supports only the new file format. The default extension of the project files on disk is

(*.adp*). The project file contains the circuit description along with all user specified support (*.sup*), model (*.mod*), line/cable (*.alc*) or transformer (*.bct*) data and library (*.lib*) files in Zip-compressed format.

4.2.1.6 Save All

Saves all modified projects to disk under their own project file names. If one or more open projects still have not got a name (`Noname.adp`), it will be requested in a *Save As* dialog boxes successively.

4.2.1.7 Close

Close the active circuit window. If any changes to the circuit have not been saved yet, the user will be warned as shown in Fig. 4.5 to confirm before the circuit is closed. If the project has been modified, the user is given a chance to save it first.

4.2.1.8 Close All

Close all circuit windows. If a project has been modified since the last save operation, a confirmation dialog will be prompted giving a chance for the user to save it first.

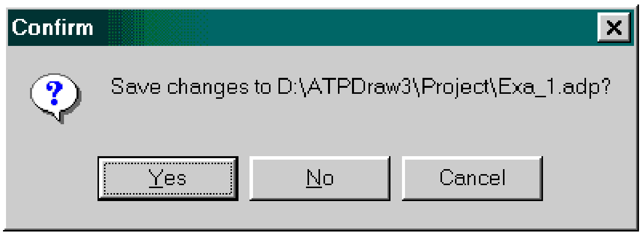


Fig. 4.5 - Confirmation prevents the loss of unsaved project data.

4.2.1.9 Import

This command inserts a circuit from disk file into the *active* circuit window contrary to the Open command, which loads the circuit into a *new* circuit window. Selecting this menu will result in an *Import Project* dialog box where the user can select the file to load. The imported circuit appears in the circuit window as a group in marked moveable mode. Existing node names will be kept or rejected upon the selection of the user.

4.2.1.10 Export

Save the selected objects of the active circuit to a disk file. Same as *Save As*, but only the selected objects (marked by a rectangular or polygon area) of the circuit is written to the disk file.

4.2.1.11 Reload Icons

Reads and displays component icons from their respective support files. This function is useful when the user has redesigned one or more support file icons and wants the changes to be reflected in the circuit window.

4.2.1.12 Save Metafile

Write the selected objects of the active circuit to a disk file in Windows metafile (*.wmf*) format. If no objects are selected, the entire circuit window content is written to disk. This way even graphics of large circuits can be exported to other applications without loss of resolution seen on the screen when the *Zoom* option is used to fit the circuit to the screen size. Metafiles created by this command can be imported as picture into other applications (like MS-Word or WordPerfect) having filter available to this format.

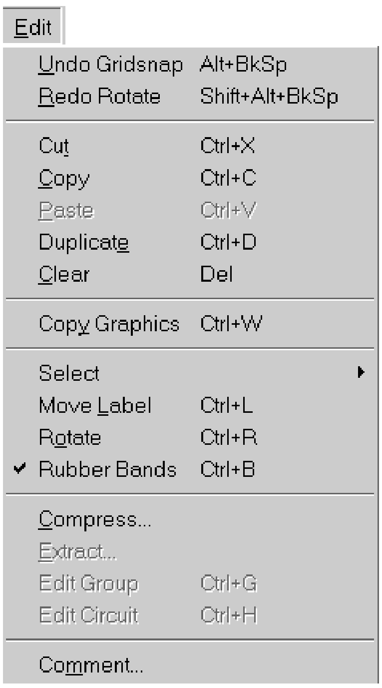
Note! Direct printing is not supported in ATPDraw v3.0 due to a scaling problem that made this feature useless anyway.

4.2.1.13 Exit

This command closes all open circuit windows of ATPDraw. User will be asked to save any modified circuits before the application is terminated.

4.2.2 Edit

This menu contains the various edit facilities of circuit objects in ATPDraw. The *Edit* popup menu is shown in Fig. 4.6.



An object or group of objects must be selected before any edit operation can be performed on them. If the user clicks on an object with the left mouse button in the circuit window the icon of the object will be rounded by a rectangular frame indicating that it is selected.

Fig. 4.6 - Edit menu.

4.2.2.1 Undo/Redo

The *Undo* command cancels the last edit operation. The *Redo* cancels the last undo command. Short key for Undo/Redo: *Alt+Backspace* and *Shift+Alt+Backspace*. The number of undo/redo operations depends on the Undo/redo buffers: setting on the *Preferences* tab of the *Tools* | *Options* menu. Default value is 10. Almost all object manipulation functions (object create, delete, move, rotate, etc.) can be undone (or redone). Changes made to the circuit data in the component dialog box are also supported by the Undo/redo functions. These functions also update the circuit's *Modified* state in the status bar to indicate that the circuit has been modified. During an undo operation, the modified state is reset its previous value.

4.2.2.2 Cut

Copies the selected objects to the Windows clipboard and deletes them from the circuit window. The objects can later be pasted into the same or other circuit windows, or even other instances of ATPDraw. Short key: *Ctrl+X*.

4.2.2.3 Copy

The selected objects are copied to the clipboard. Short key: *Ctrl+C*. A single marked object or a group of objects can be copied to the clipboard. This command unselects the selected objects.

4.2.2.4 Paste

The contents of the clipboard are pasted into the current circuit when this menu item is selected. Short key: *Ctrl+V*. The pasted object or objects appear in the current window in marked moveable mode.

4.2.2.5 Duplicate

Copies the selected object or a group of objects to the clipboard and then duplicates them in the current circuit window. Duplicated objects appear in the current window in marked moveable mode. Short key: *Ctrl+D*.

4.2.2.6 Clear

Selected objects are removed the from the circuit window. Short key: *Del*.

4.2.2.7 Copy Graphics

The selected objects are copied to the clipboard in Windows Metafile format. This way graphics of selected objects can be exported to other Windows applications. Short key: *Ctrl +W*.

4.2.2.8 Select

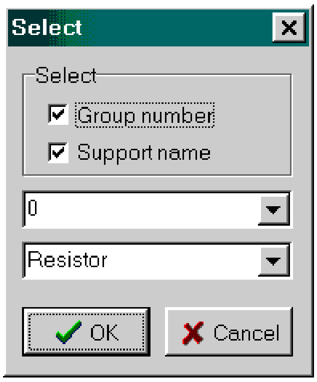
This menu has four sub-menus:

`All`: Select all objects in the current circuit window. Short key: *Ctrl +A*.
`None`: To cancels the object selection. Short key: *Ctrl +N*.
`Polygon`: Enables object selection by a polygon shaped region. Short key: *Ctrl +P* (or double-click with the left button in an empty region of the circuit window).
`Type/Group`: Enables selection by objects' support file name or group number (see below). Short key: *Ctrl +T*.

A selected object or group of objects can be subject of the most editing operations: *Move* (click left button, hold down and drag), *Rotate/Copy/Duplicate/Erase* or *Export* (in the File menu). To unselect a group, select *None*, or just click with the left mouse button in an empty space of the circuit window.

In *Polygon* mode, the mouse cursor icon changes its style to a pointing hand and moves to the middle of the circuit window. The current action mode also changes to *MODE:GROUP* in the status bar. To draw a polygon around a group of objects move the cursor to the starting location and click the left mouse button. Then release the button and a rubber band line will be drawn between the starting point and the current mouse cursor location. And so forth: left click to create corners, right to complete the polygon. All objects with midpoint inside or connections with both endpoints inside the polygon will be included in the selection.

In the *Type/Group* selection mode the group of components can be selected by their type and/or group number. The type here is the name of the support file and the group number is the identifier specified in the component dialog box.



The available component types and support names are listed in two combo boxes as shown in Fig. 4.7. When you click on *OK* the components with the selected group number and/or support file name become selected. Then all kinds of edit operation can be performed on the group (copy/paste, copy graphics, rotate, edit, grouping etc.).

Fig. 4.7 - Selecting objects by name or group no.

4.2.2.9 Move Label

This menu is used to select and move a component or node text label. Short key: *Ctrl+L.*

All circuit objects can have a label. These labels are displayed on the screen in blue (component label) or in red color (node name). Selecting the *Move Label* menu item, the mouse cursor style will change to a pointing hand and move to the middle of the circuit window. The action mode indicator in the status bar will also change to *MODE: MOVE LABEL*. In this operating mode any label can be dragged to a new position by clicking the label with the left mouse button, holding the button down and move. This operation ends and the mouse is unlocked when you finish moving a label and the mouse buttons is released. In most cases, when the component icons do not overlap the labels, it can be moved by a simple left click then holding the left mouse button down and move. Therefore, the *Move Label* command is most frequently used when labels are located close to or behind the component icons and cannot be selected otherwise.

4.2.2.10 Rotate

This command rotates the selected object(s) 90 degrees counter-clockwise. This operation can also be performed by clicking the right mouse button inside the selected group. Short key: *Ctrl + R.*

4.2.2.11 Rubber Bands

If this option is checked, connections with one endpoint inside a selected region and one outside are treated as a rubber band between the selected group and the rest of the circuit. Short key: *Ctrl + B*. This command does not work for short cut single component selections: e.g. left click on several components while the *Shift* key is pressed, because this way no connections are selected.

4.2.2.12 Compress

This command will replace a group of selected objects with a single icon having user selectable external data and nodes. ATPDraw supports real grouping or single icon replacement of sub-groups in unlimited numbers of layers. The process requires a group selection first. The *Compress dialog* box (see Fig. 4.8) appears where the user designs the new group object.

In the Compress dialog box the user can specify the external data and nodes of the compressed circuit. The selected data and nodes appear as input to the group object that replaces the selected circuit and their values are automatically transferred. A nonlinear characteristic common for up to 3 components can also be selected as external data.

Under *Objects:* all the components in the group are listed with their name (support file) followed by their label. When the user clicks on one of the component's name, its data and nodes appear under *Available:* starting with data/node name and followed by their names and values. Here the user can select a parameter and click on the >> button to transfer it to the *Added to group:* list. All data and nodes listed in the *Added to group:* will be an external attribute of the new group object. The position of the external nodes on the icon border (1-12) can be specified in the lower right corner of the dialog box. You can change the *Added to group:* names by double clicking on them. Data with the same name are treated as a single data in the component dialog box (Fig. 4.8/b). Selected data and nodes can also be removed from the *Added to group:* by clicking on the << button.

As all other components, the group object is limited to 36 data and 12 nodes. When later you open the component dialog box of the group-object, the selected data values and node parameters will appear as input possibilities. The values will automatically be transferred to the sub-group as shown in Fig. 4.8/b.

4.2.2.13 Extract

This is the reverse operation of *Compress*. The group is extracted on the current circuit layer. To perform the operation, a compressed group (and only one!) must be selected first.

4.2.2.14 Edit Group

This command shows the group content. Short key: *Ctrl+G*. The group is extracted in a separate window. To perform the operation a compressed group (and only one!) must be selected first. It is possible to edit the group in a normal way, except deletion of the reference components. I.e. components having been referenced in one of the *Added to group:* lists cannot be deleted. If the user tries a *"Marked objects are referenced by compressed group..."* warning message appears.

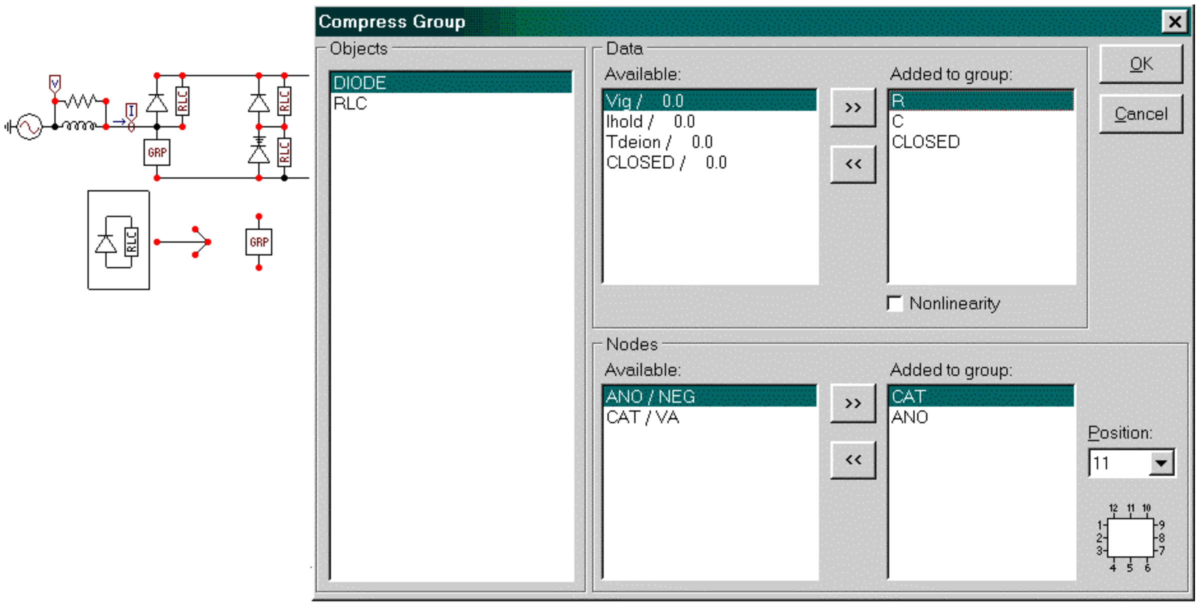


Fig. 4.8/a - The Compress dialog box.

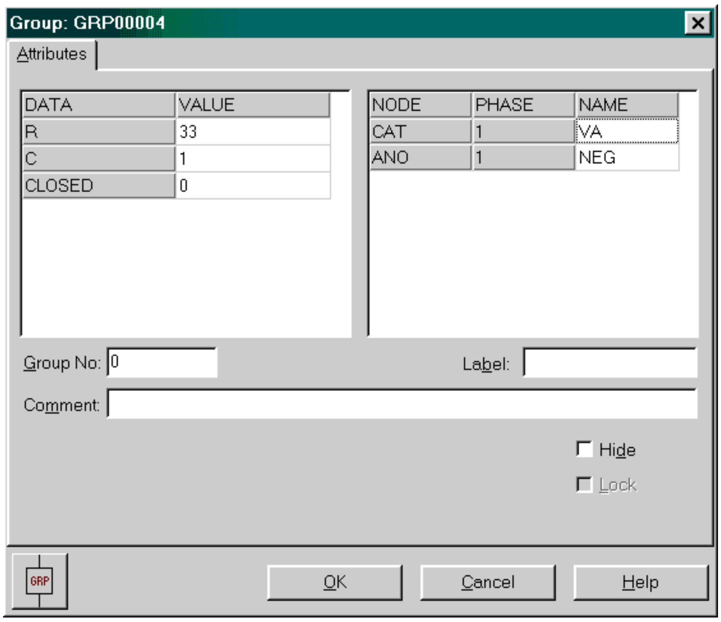


Fig. 4.8/b - Component dialog box for a sub-group object.

4.2.2.15 Edit Circuit

Displays the circuit where the current group belongs. Short key: *Ctrl + H*. Actually the grouping structure can be taken as a multi-layer circuit, where the *Edit Group* brings the user one step down in details, while *Edit Circuit* brings one step back. The group object (single icon replacement of objects) acts as the connection between the layers and transfers data between them.

4.2.2.16 Comment...

Opens a comment dialog box (see Fig. 4.9), where three text lines can be entered. These comments serve as a commentary section for the circuit in the header section of the `.atp` file. Selecting the *Comment Line* option checked in the *View* menu will display these comments at the bottom of the circuit window, as well. This menu also enables the user to change the circuit comment if it already exists.

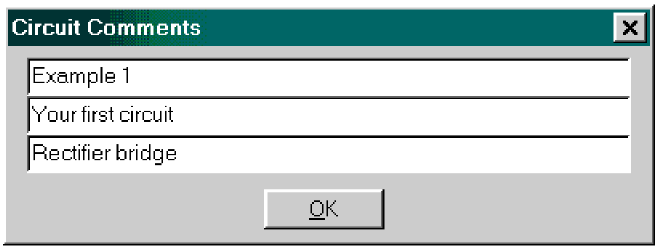
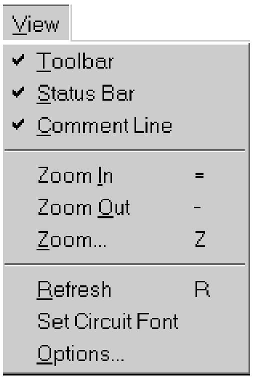


Fig. 4.9 - Circuit comments dialog box.

4.2.3 View



This menu provides options for displaying and controlling the visibility of user interface and circuit window objects. The menu items are shown in Fig. 4.10.

Fig. 4.10 - View menu.

4.2.3.1 Toolbar

Shows or hides the toolbar at the top of the main window. The toolbar contains speed buttons for the most frequently used menu options. Available toolbar icon are as follows:

 Opens an empty circuit window

 Loads a circuit file into a new window

 Saves the objects in the active circuit window to disk

 Saves the objects in the active circuit window to a specified disk file

 Inserts a circuit from file into the active circuit window

 Saves the selected objects of the active circuit to a disk file

 Cancels the last edit operation

 Cancels the last undo command

 Copies the selected objects to the Clipboard and deletes them from the circuit window. The objects can later be pasted into the same or other circuit windows.

 Copies the selected objects to the Clipboard

 Inserts the objects in the Clipboard into the circuit window

 Copies the selected objects to the Clipboard and then inserts them into the circuit.

Enables the user to select and move a component or node text label. The mouse cursor type will change to a pointing hand.

Redraws all objects in the active circuit window

Selects all objects in the active circuit window

Enables the user to select a group of objects by specifying a polygon shaped region in the active circuit window. The mouse cursor style will turn to a pointing hand to indicate this mode of operation. To close the region and unlock the mouse, press the right mouse button.

Enlarges the objects by increasing the current zoom factor by 20 percent

Diminishes the objects by reducing the current zoom factor by 20 percent

Rotates the selected objects 90 degrees counter-clockwise. This operation can also be performed by clicking the right mouse button inside the selected region

Flips the selected objects by rotating 180 degrees

To the right of the speed buttons the nine most recently used component icons are displayed. Selecting one of these shortcut icons inserts a new component into the active circuit window. The leftmost icon represents always the last inserted component.

4.2.3.2 Status Bar

Status bar on/off at the bottom of the main window The status bar displays status information about the active circuit window. The mode field on the left hand side shows which mode of operation is active at present. Possible modes are:

EDIT	Normal mode. Indicates no special type of operation.
CONN.END	Indicates the end of a connection. The program is waiting for a left mouse button click to set the end-point of a new connection. To cancel drawing a connection, click the right mouse button or press the *Esc* key.
MOVE LABEL	Indicates a text label move. Clicking the left mouse button on a text label, then holding down and dragging enables you to move the label to a position of your choice. To cancel moving a label, click the right mouse button or press the *Esc* key.
GROUP	Indicates region selection. Double clicking the left mouse button in an empty space of the active circuit window enables you to draw a polygon shaped region. To finish the selection click the right mouse button. Any object within the selected region is then marked for selection. To cancel region selection, press the *Esc* key.
INFO.START	Indicates the start of relation drawing when the *TACS | Draw relation* was selected in the component selection menu. Clicking the left mouse button on a component node or on the end-point of another relation will initiate the drawing of a new relation. Relations are used to visualize information flow into Fortran statements and are drawn as blue connections, but do not influence the connectivity of components.
INFO.END	Indicates the end of a relation. The program is waiting for a left mouse click to set the end-point of a new relation. To cancel drawing a relation, click the right mouse button or press the *Esc* key.

The field to the right of the mode field displays the modified status of the active circuit. As soon as you alter the circuit (moving a label, deleting a connection, inserting a new component, etc.), the text *Modified* will show up to indicate that the circuit needs saving. The field will be empty when you save the circuit or undo all modifications. Note that the number of available undo buffers is limited (default value is 10, but can be increased on the *Preferences* tab of the *Tools* | *Options* menu). In the default case, if more than 10 modifications are done, the field will indicate a modified status until you save the circuit.

The rightmost field of the status bar displays the menu option hints.

4.2.3.3 Comment Line

Shows or hides the comment line at the bottom of the active circuit window.

4.2.3.4 Zoom In

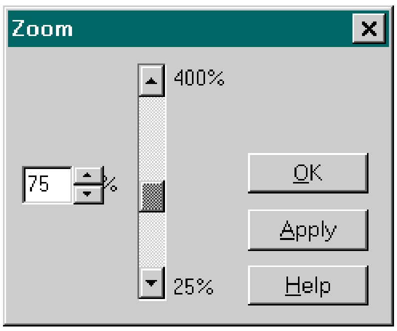
Enlarges the objects in the active circuit window by increasing the current zoom factor by 20 percent. Short key: + *(plus sign on the numeric keypad or "=/+" alphanumeric key).*

4.2.3.5 Zoom Out

Reduces the icon size in the active circuit window by 20 percent. Short key: - *(minus sign on the numeric keypad or the "-/_" alphanumeric key).*

4.2.3.6 Zoom

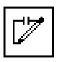
Selecting this field brings up the *Zoom* dialog box shown in Fig. 4.11. Short key: *Z.* In the Zoom dialog the zoom factor of the active circuit window can be specified. The actual zoom factor is given in the input field at left in percent. The normal view has a zoom factor of 100 percent. To zoom in increase the zoom factor, to zoom out and view a larger portion of the circuit, decrease the factor. Upper and lower limits are 400 and 25 percent, respectively.



To accept the new zoom factor and return from the Zoom dialog, select the *OK* button. To set a new zoom factor and view the result without closing the dialog box, select the *Apply* button.

Fig. 4.11 - The Zoom dialog box.

4.2.3.7 Refresh

This command redraws all objects in the active circuit window. Short key: *R*. This command can also be activated by clicking the Toolbar icon: 

4.2.3.8 Set Circuit Font

Enables you to select a font type and size for the node names and labels on the screen (and also for the metafile export). The default font is MS Sans Serif, regular, 8 pt size.

4.2.3.9 Options

Selecting this menu item will bring up the *View Options* dialog box. The *View Options* dialog can be used to control the visibility of the objects in the active circuit window.

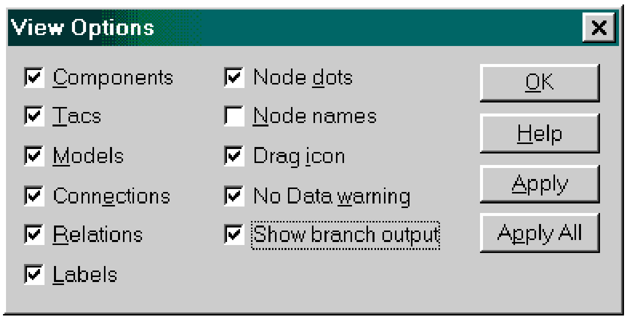


Fig. 4.12 - View Options dialog box.

By default, all objects except node names are visible. The meaning of options assuming the option is selected (☑) are listed below:

Components	All standard and user specified components are displayed.
Tacs	All TACS components are displayed.
Models	All MODELS components are displayed.
Connections	All connections (short circuits between nodes) are displayed.
Relations	All relations (to visualize connections between Fortran statements and other objects) are displayed.
Labels	Component labels are displayed on the screen.
Node dots	Node and connection end-points are displayed as filled circles.
Node names	Node names are visible on the screen (overrides the *Display* attribute of the Node data window). This option is useful after a *Make Names* selection in the *ATP* menu.
Drag icon	The complete icon is seen when a selected object or group of object is moved on the circuit window (in the unchecked state, only an inverted rectangular or polygon outline is drawn).
No Data warning	Components and node dots are drawn with a red color until the component or node is opened and given meaningful data. No such checking is done when this option is unselected.
Show branch output	Small U/I symbols indicate the selected branch output requests. Branch output requests can be specified in most of the component dialog boxes.

To accept the current view options and return from the dialog, select the *OK* button. To set and view new options without returning, select the *Apply* button. If you want the current settings be applied to all current and future circuit windows, select the *Apply All* button before you exit the dialog box.

4.2.4 ATP

The *ATP* menu provides options to create, display and modify the ATP input files and to set circuit specific ATP options (e.g. ΔT, Tmax) before running the case by the *Run ATP* command or the *F2* function key. Other components of the ATP-EMTP package (e.g. pre- and post-processors, supporting programs and utilities) can also be launched from this menu. Besides the default commands, the user can add additional commands (e.g. *Run PlotXY / Run Analyzer / Run PCPlot / Run TPPlot, etc.*) to the existing program items, which are listed immediately below the *Edit commands...*as shown in Fig. 4.13.

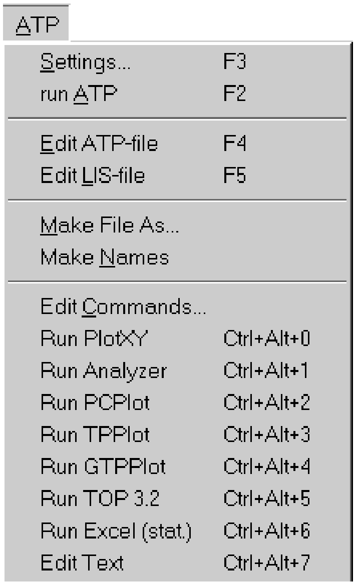
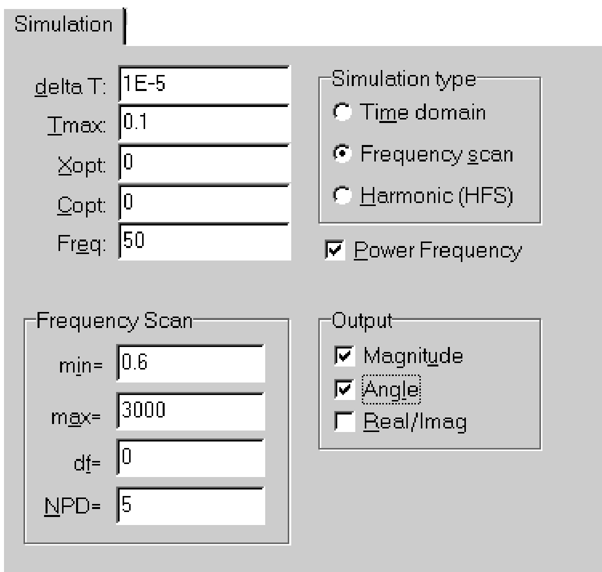


Fig. 4.13 - The ATP menu.

4.2.4.1 Settings

In the *ATP Settings...* dialog box several options for the active circuit window can be specified. These settings are used when ATPDraw generates the ATP input file. Options are sorted in six tabs, such as the *Simulation* and *Output* for the miscellaneous data card settings, *Switch/UM* for statistical and Universal Machine studies, *Format* for specification of data-card sorting options and miscellaneous request, *Record* for MODELS output specification and the new *Variables*.

Simulation settings



Simulation type: Select between the simulation methods supported by ATP:
- o Time domain
- o Frequency Scan
- o Harmonic Frequency Scan (HFS)

Time domain

delta T: Time step of simulation in seconds.

Tmax: End time of the simulation in seconds.

Xopt: Inductances in [mH] if zero; otherwise, inductances in [Ohm] with *Xopt* as frequency

Copt: Capacitances in [micro-F] if zero; otherwise, capacitances in [uMho] with *Copt* as frequency.

Freq: System frequency in Hz

Fig. 4.14 - Simulation settings.

Power Frequency: when checked the SYSTEM FREQUENCY request card is written in the ATP-file.

Frequency scan

If *Frequency scan* is selected the FREQUENCY SCAN option of ATP is enabled.

min: Starting frequency for the frequency scan
max: Ending frequency for the frequency scan
df: Frequency increment. Leave 0 for logarithmic frequency scale
NPD: Number of frequency points per decade in logarithmic scan

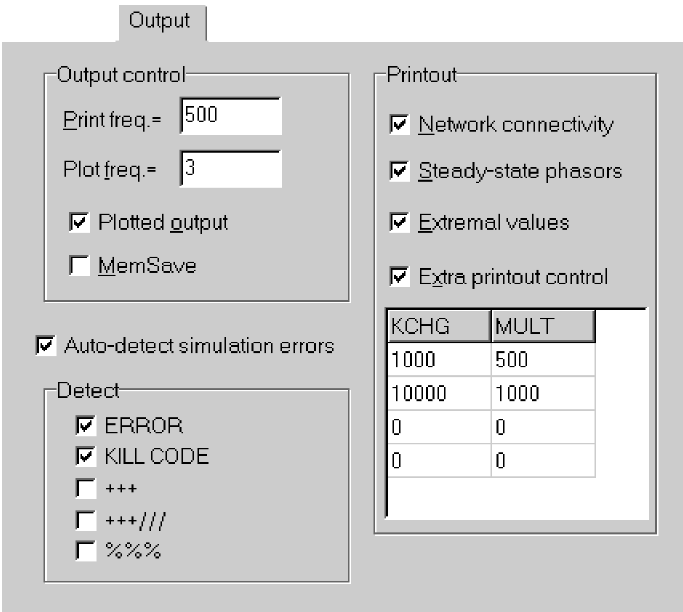
Harmonic Frequency Scan (HFS)

Selecting *HFS* will run the ATP data case so many times as specified in the *Harmonic source* component dialog box (see chapter 4.9.12). The frequency of the harmonic source will for each ATP run be incremented. The basic frequency specification is mandatory for HFS simulations, which can be set as *Freq.* parameter on Fig. 4.14.

If *Frequency scan* or *HFS* is selected the user must specify which component of the solution to print out:

Magnitude only: Default request
Magnitude & Angle: Results are printed in POLAR
Magnitude & Angle & Real/Imag: Both POLAR and RECTANGULAR
Real/Imag: RECTANGULAR output request. Other combinations are illegal and are prevented by button logic.

Output settings



Output control

Print freq.: Frequency of LUNIT6 output within the time-step loop. For example, a value of 500 means that every 500^{th} simulation time step will be printed to the LIS-file. This option controls ATP's 1^{st} misc. data parameter IOUT

Plot freq.: Saving frequency of the simulation data to the *.pl4* output file. A value of 5 means for example, that every fifth time step will be written to the PL4-file. This option controls ATP's 1^{st} misc. data parameter IPLOT

Fig. 4.15 - Output request tab.

Plotted output: If checked ATPDraw sets the 1^{st} misc. data parameter ICAT=1 in the ATP input file which results in a *.pl4* output file.

MemSave: Controls the dumping of EMTP memory to disk at the end of simulation if START AGAIN request is specified. If checked indicates memory saving.

Auto-detect simulation errors: If this option is selected, ATPDraw will analyze the output LIS-file of ATP following the completion of the simulation. If the specified trigger string is found, the corresponding section of the file is displayed in a Notepad window. This feature helps the user to recognize the simulation errors/warnings generated by ATP during the time step loop or input data interpretation. The string or strings, which makes this function work, are user selectable and activating at least "Error" and "Kill code" are highly recommended.

Printout

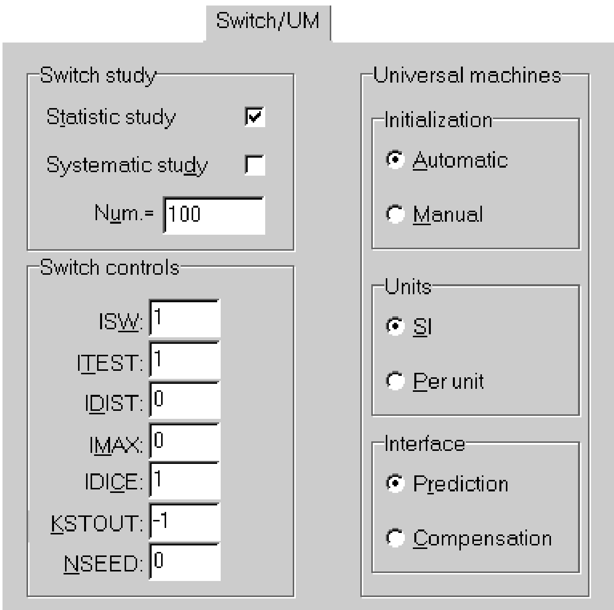
Network connectivity: If checked connectivity table (description of the topology of the circuit) is written to the LUNIT6 output file. This option controls ATP's 1^{st} misc. data parameter IDOUBLE. If unchecked, no such table is written.

Steady-state phasors: If checked complete steady state solution (branch flows, switch flows and source injection) is written to the LUNIT6 output file This option sets ATP's 1^{st} misc. data parameter KSSOUT=1. If unchecked, no such output is produced by ATP.

Extremal values: If checked, extrema of each output variables will be printed at the end of the LIS-file. This option controls ATP's 1^{st} misc. data parameter MAXOUT. If unchecked, no such output is produced by ATP.

Extra printout control: Additional control for the frequency of LUNIT6 output within the time-step loop. If checked, the 1^{st} misc. data parameter IPUN is set to -1 and a 2^{nd} misc. data card will appear in the ATP input file. Parameters KCHG and MULT control the breakpoints and the new *Print freq.* value. If unchecked, IPUN is set to 0 and LUNIT6 printout frequency will be constant throughout the simulation.

Switch/UM settings



Switch study

Statistic study: Study with statistic switches

Systematic study: Study with systematic switches

Num: Number of simulations. This value influences ATP's 1^{st} misc. data parameter NENERG. ATPDraw sets the correct sign of NENERG: i.e. >0 for statistic or <0 for systematic switch studies.

Switch controls

ISW: If 1, printout of all switch closing/opening time appear in the output LIS-file. No such printout if the parameter is set to 0.

Fig. 4.16 - Switch/UM settings.

ITEST: Extra random delay using DEGMIN, DEGMAX and STATFR in STARTUP. Possible values are:

0: Extra random delay for all switches.
1: No random delay.
2: Extra random time delay added to all closing switches.
3: Extra random time delay added to all opening switches.

IDIST: Select probability distribution function of subsequent switching operations. Zero means Gaussian distribution and 1 means uniform distribution.

IMAX: If 1, printout of extrema is written to the ATP output LIS-file for every energization. If 0 (zero), no such printout.

IDICE: Controls use of the random generator. A value of 0 implies computer-dependent random generator and a value of 1 means standard random generator.

KSTOUT: If 0, extra printed (LUNIT6) output for each energization. Output of the time-step loop and variable extrema (if *Extremal values* is selected on the *Output* tab) will be printed. If -1, no such output.

NSEED: Repeatable Monte-Carlo simulations. Possible values are:
0: Every simulation on the same data case will be different.
1: Same result each time the data case is run on the same computer.

Universal machines

Here the user specifies the global data for the Universal electrical machine models in ATP. The selections here apply to all universal machines in the circuit.

Initialization: *Manual*: Terminal quantities of all machines must be specified. *Automatic*: Initial conditions will be calculated by ATP. See section 9D1.5 for more details in the ATP Rule Book.

Units: Input variables are specified in *SI* units or *Per unit* (p.u.)quantities.

Interface:
Compensation: The machine does appear to be a nonlinear element to the external network. Certain rules regarding connecting machines together must be followed. Inclusion of stub lines is often required.

Prediction: The machine does not appear to be a nonlinear element to the external network. This option is not available for single phase machines.

Format settings

The *Format* settings page contains four buttons for setting of ATP input file data format, a button for controlling the auto path generation and several other buttons for miscellaneous request cards. The *Additional* button supports the user to insert any request card or text strings in the ATP-file on precise location.

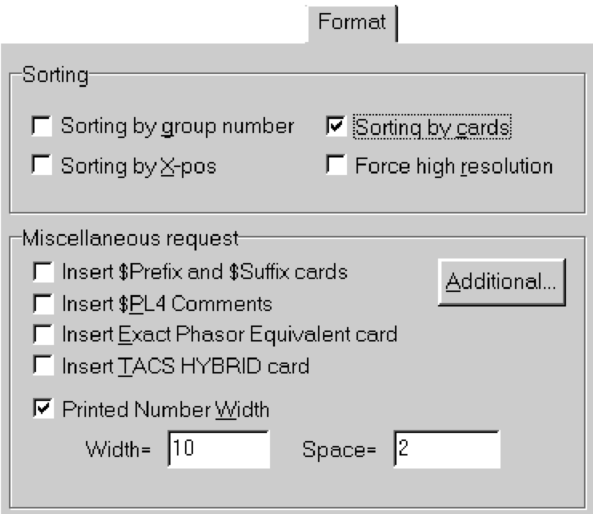


Fig. 4.17 - ATP-file format settings.

Sorting

Sorting by cards: The sequence of ATP input data follows the default sequence of / data sorting cards (i.e. BRANCH cards are written first, followed by SWITCH cards and the SOURCE cards).
Sorting by group number: The group number that can be specified in the component dialog box for each object determines the sequence of cards. The lowest group number comes first.
Sorting by X-pos: The leftmost object in the circuit window is written first.

Any combination of the three different sorting mechanisms can be specified.

Force high resolution: Use $Vintage, 1 (if possible), for high precision data input.

Miscellaneous request

Insert $Prefix and $Suffix cards: If this option is checked, ATP run time calls for library or data file via *$Include* commands will suppose that these external files are located in the /USP folder of ATPDraw and have the extension *.lib*. Having this control enabled, only the pure file name need to be specified in the *$Include* field of a component dialog box, because the default extension *.lib* and the complete path will be appended by ATPDraw using the *$Prefix*, *$Suffix* features of ATP-EMTP.

Each library file specification is verified to meet the above requirements. If the path of a library file specifies a different folder or the extension is not *.LIB,* an error dialog appears during the ATP-file generation process. The user has 3 options:

a) correct the erroneous specification by stripping off path and extension,
b) continue the operation using an unresolvable ATP *$Include* reference,
c) cancel the entire input file generating process.

If you prefer storing library files outside of the /USP folder or file extensions other than *.lib* are often used, it is wise **not** to select this option and specify a complete path in the $Include field.

Insert $PL4 Comments: If checked, ATPDraw writes the circuit comments in a $BEGIN PL4 COMMENTS...$END PL4 COMMENTS block. This may result in an error for some (older) ATP versions.

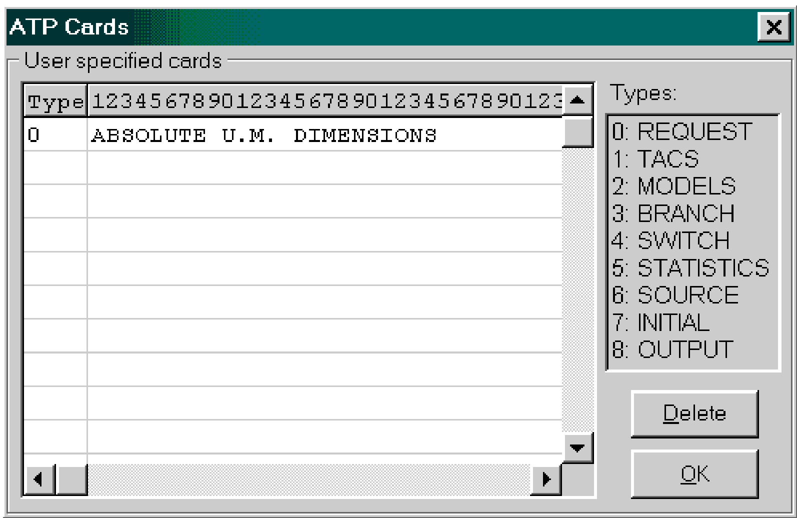
Insert Exact Phasor Equivalent card: If checked, ATPDraw writes an EXACT PHASOR EQUIVALENT request in the ATP-file. This is recommended for Frequency Scan simulations including constant and distributed parameter overhead lines.

Insert TACS HYBRID card: Checking this button forces TACS HYBRID .. BLANK TACS to be written to the ATP-file. Useful when TACS objects are only present inside a *User Specified Object*.

Printed Number width: Enables the PRINTED NUMBER WIDTH request card, which controls the printout of the LUNIT6 device (output LIS-file). *Width:* is the total column width of printed output including blanks separating the columns. *Space:* is the number of blanks between columns of printed output.

Additional

The *Additional* button enables the user to insert text strings on precise locations in the ATP-file. Text strings to be written can be specified in an editor-like dialog box as shown in Fig. 4.18.

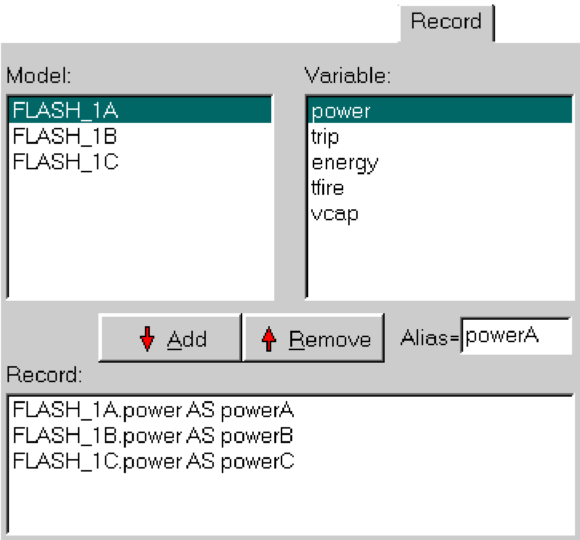


First, the user must specify a type (0-8; see the list at right), then the text string to be written in the ATP-file. The header of columns 1-80 helps to locate the string properly, according to the ATP specification. The text string will be written after a / data sorting card **at the end** of other cards belonging to that group (/REQUEST, /TACS, etc.).

Fig. 4.18 - Specification of additional miscellaneous request cards.

Record

ATPDraw supports the RECORD feature of ATP MODELS via the *Record* dialog box which is shown in Fig. 4.19. All MODELS objects in the active circuit window are listed with their USE AS name in the list box under *Model*. When selecting a model in this field, variables (declared under VAR section of the .mod file) are listed under *Variable*. Each variable has a default alias name that appears in the *Alias* field, but it can be modified according to the user need.



To record a variable, one has to click on the *Add* button. The alias name can be changed later by selecting an item in the *Record* list box and type in a new name.

This record list is stored in the circuit file, but it does not follow the circuit when the circuit is copied to the clipboard or the export group option is used. The record requests can be removed by selecting the item and click on the *Remove*.

Fig. 4.19 - Record of MODELS variables.

Variables

The *Variables* dialog box is new in ATPDraw 3.x to support the `$PARAMETER` feature of ATP-EMTP. The user is allowed to specify a 6-character text string instead of a numerical value in the component dialog boxes as shown in Fig. 4.20.

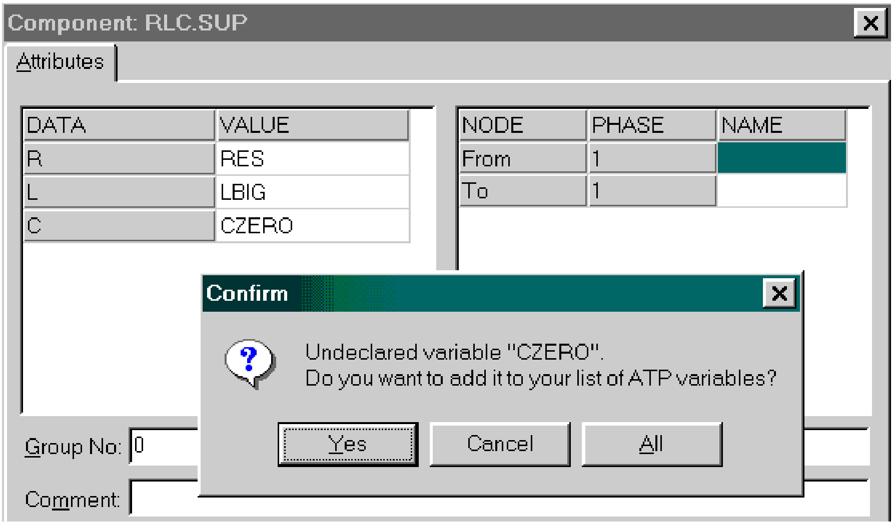
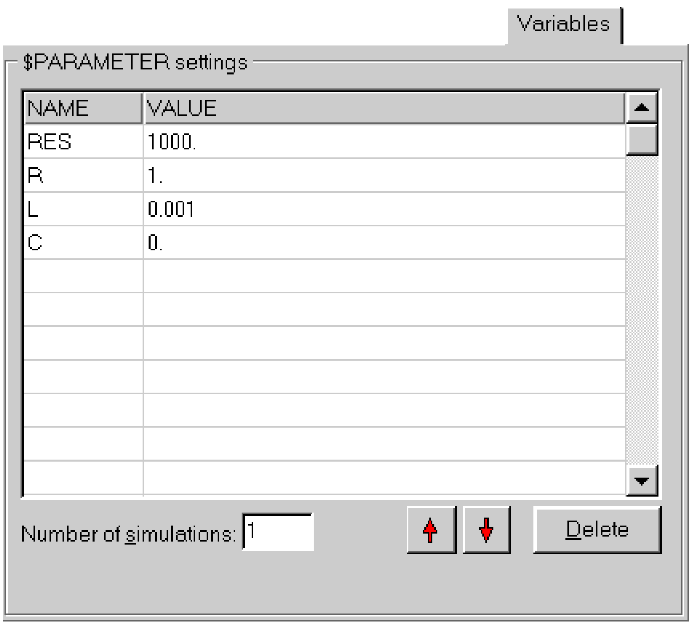


Fig. 4.20 - Using text string instead of variables in the RLC component dialog box.

A numerical value then can be assigned later to these text strings under *Variables*. The text strings (variables) specified by the user appear to the left and the user now has to assign data values for variables. This is done in free format in the column to the right as shown by Fig. 4.22. Users do not have to think about the number of characters in the final ATP-file since ATPDraw automatically adds underscore characters to obtain the maximum resolution. A variable `R` used both for high and low precision resistances will thus be declared twice with 5 and 15 underscore characters added. This process is hidden, but the result is seen in the final ATP-file under the `$Parameter` declaration.

IMPORTANT! Always use a period '.' after a number in the value field.



Changing the names in the left side column will affect the text strings (variables) specified in the component dialog boxes. An example is shown in Fig. 4.22. If the variable `R` is changed to `RES`, the variable `R` used in some component dialog box will no longer be assigned. Therefore, the user is requested about what action to take.

a) reset the variable `R` to zero,
b) set default value (from the `.sup` file),
c) select a parameter and decide which one of the variables should replace `R`.

Fig. 4.21 - Setting values to text strings.

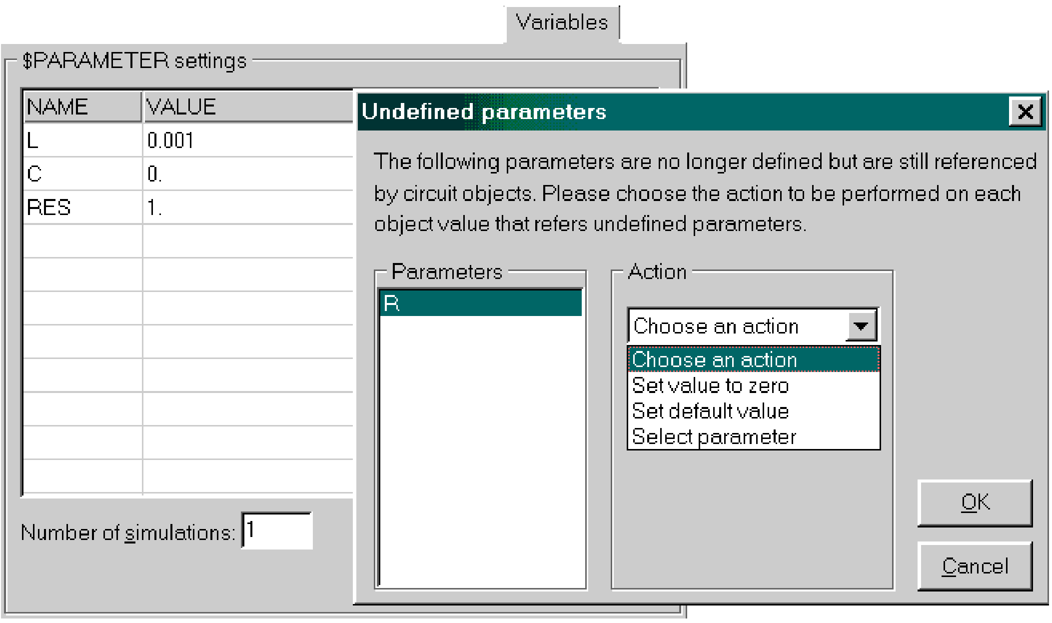


Fig. 4.22 - Actions to take when non-defined parameters are found.

4.2.4.2 Run ATP

Executing the *run ATP* command at the top of the ATP menu or pressing the *F2* function key first will create an ATP input file with the same name as the circuit file (with extension `.atp`) in the `/ATP` system folder. Then an ATP run will be performed via the *default ATP command*, which is specified in the *ATP* field of the *Preferences* page under *Tools* | *Options*. The current ATP-file is sent as parameter to the ATP-EMTP.

It is important to note that users do not need to select *Make Names* and *Make File As* before running the simulation. These commands are internally executed before the ATP run. If the user needs to do manual changes of the ATP-file and run the modified case, a new command must be created (with the current ATP-file as parameter) using the *ATP* | *Edit Commands...* feature. After executing ATP, ATPDraw examines the LIS-file and displays any error or warning messages if exist.

4.2.4.3 Edit ATP-file

This selection calls a text editor, which enables the user to contemplate or edit the ATP-file. When the *Edit File* option is selected (or the *F4* function key is pressed) a file having the same name as the active circuit file with extension `.atp` is searched for, and will be opened in the built in Text Editor as shown in Fig. 4.23.

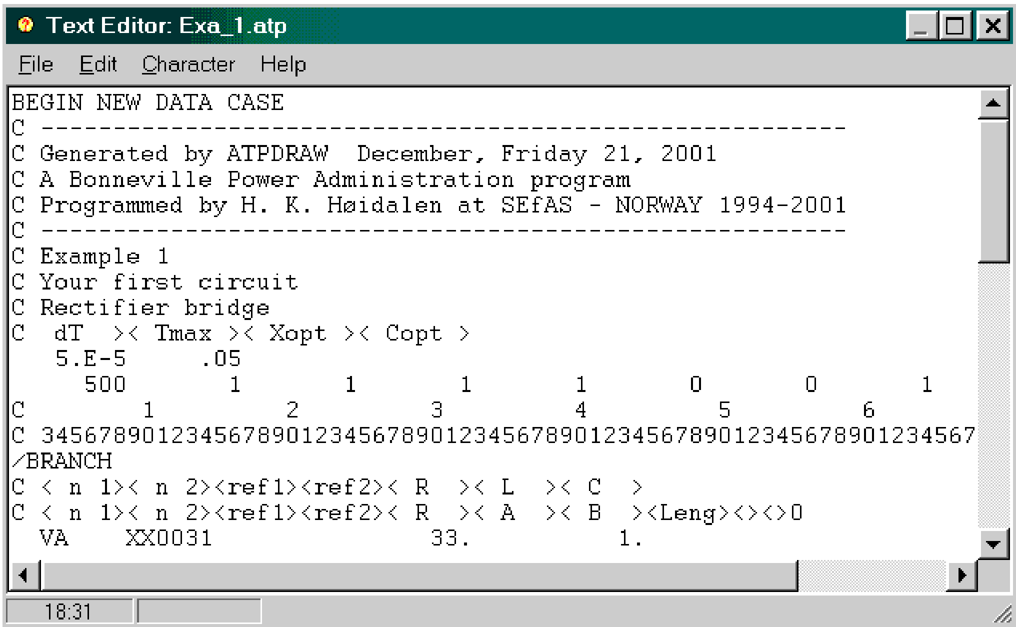


Fig. 4.23 - The main window of the built in text editor.

The status bar at the bottom of the window displays the current line and column position of the text cursor, and the buffer modified status. Basic text editing facilities (Open/Save, Print, Copy/Paste, Find & Replace) are supported. The default text font can be changed by selecting the *Font* option in the *Character menu*. A detailed description of all the available options can be found in the menu options help topic. The text buffer of this editor is limited to maximum 32kB in size, however the user can specify his own favorite text editor (wordpad.exe, write.exe, notepad.exe) on the *Preferences* page of the *Tools* | *Options* dialog box.

Text Editor option in the *Tools* menu provides an alternative way of invoking this editor. In that case the text buffer will initially be empty.

4.2.4.4 Edit LIS-file

This selection calls the built-in text editor, which enables the user to contemplate the LUNIT6 output of ATP (often called as LIS-file). This file has extension `.lis` and can be found in the `/ATP` system folder following a successful simulation. In certain cases when the simulation is halted by an operating system interrupt or a fatal error in the ATP input file (e.g. division by zero, or I/O-xx bad character in input field) the LIS-file does not exist and can not be displayed either.

4.2.4.5 Make File As

Generates an ATP input file for the active circuit window. User will be asked to confirm the name of the file. Default filename is the same as that of the circuit file, but with extension `.atp`. ATPDraw first calls the *Make Names* procedure then generates an ATP input file on the form specified in the *Settings* menu.

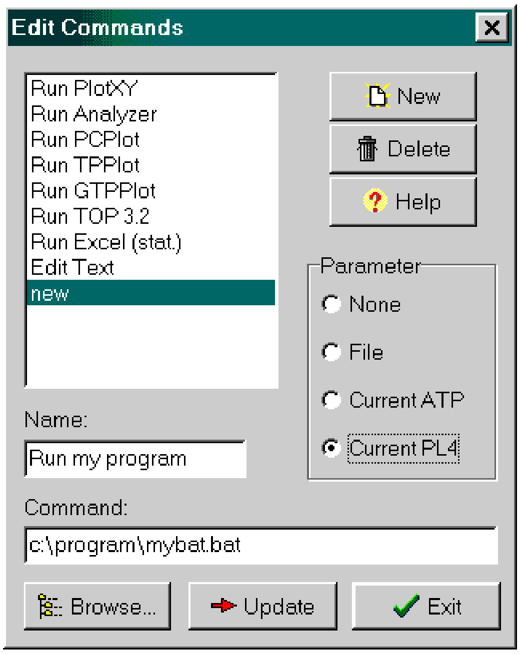
4.2.4.6 Make Names

When this field is selected, ATPDraw examines the current circuit and gives unique names to all nodes left blank in the active circuit window. Connected or overlapped nodes are given the same name. Whenever a "same name on different nodes" or "duplicate names" for nodes connected by short circuit were to be found, ATPDraw produces a warning and the user is asked to confirm this operation. While ATPDraw establishes the node names a **`Generating node names`** message is displayed in the middle of the current circuit window. Following *Make Names*, the node name and phase sequence attributes in the *Component* dialog box and in the *Node data* window will be updated.

IMPORTANT! All nodes will automatically receive names from ATPDraw, so the user should normally only give names to nodes of special interest, e.g. a node which appears on the `/OUTPUT` list is preferred to have a user specified name.

4.2.4.7 Edit Commands...

This feature enables to specify executable files (*.exe or *.bat) to run from the ATP menu. New commands will appear as menu items below the *Edit Commands...* After clicking on the *New* button of the dialog box as shown in Fig. 4.24, the user is requested to specify:



- the *Name* of the command displayed under the ATP menu
- name and path of the executable file(*.exe or *.bat),
- *Parameter* is the file to send as parameter when calling the executable file.
  None: No file sent as parameter
  File: A file open dialog box is displayed where the user can select a file
  Current ATP: send the current ATP-file
  Current PL4: send the current PL4-file

Parameter options can be selected by radio buttons. If the *File* is selected, ATPDraw performs an open dialog box, where the user can select a file name, to be sent as parameter when executing the batch file.

Fig. 4.24 - Specifying your own executable commands.

When you completed editing the batch job settings, click on the *Update* button and the new commands will be inserted into the *ATP* menu.

As any other program options, the previous settings can be saved to the `ATPDraw.ini` file by using the *Tools | Save Options* command or by selecting the "Save options on exit" program options on the *General* page of the *Tools | Options* menu.

This feature can be used for many different purposes in ATP simulation: e.g. running different ATP versions (Salford, Watcom, GNU-MingW32) within ATPDraw; running external post-processors like TPPLOT, PCPlot or PlotXY; or launching any other data assembler.

4.2.5 Objects

This menu contains options for creating and customizing component support files. Support files contain information on data and node values, icon and help text. Circuit objects in ATPDraw can be either:

1. Standard components,
2. Line/Cable (LCC) components,
3. BCTRAN transformer objects,
4. MODELS components,
5. GROUP objects,
6. User specified (USP) components.

Each object has a unique support file, which includes all information about the input data and nodes of the object, the default values of the input variables, the graphical representation of the object and the associated help file. Each circuit object has different name internally in ATPDraw. Support files of standard components are zipped together in a single file called `ATPDraw.scl` of the main ATPDraw directory. Support files of MODELS, GROUP and USP objects have extension `.sup` and are stored in system folders `/MOD`, `/GRP` and `/USP`, respectively. Support files of the BCTRAN and LCC objects are stored in the Standard Component's Library `ATPDraw.scl`, while other files created by ATPDraw or ATP when using these objects are stored in the `/BCT` and `/LCC` folders. The full path of the support files are included in the data structure of the project files, so the `.sup` files can be stored anywhere, but the usage of the directory structure given below is recommended to ensure compatibility with other users:

Object type	Support file `(.sup)`	LIB file `(.lib)`	Other `(.mod,.alc,.pch)`
Standard components	`ATPDraw.scl`	-	-
Line/Cable components	`ATPDraw.scl`	`\LCC`	`\LCC`
BCTRAN trafo objects	`ATPDraw.scl`	-	`\BCT`
MODELS components	`\MOD`	-	`\MOD`
GROUP objects	`\GRP`	-	-
User specified components	`\USP`	`\USP`	-

The objects support files can be edited in the *Objects* menu. The user can create new MODELS and User Specified components as described in the Advanced Manual.

4.2.5.1 Edit Standard

The standard component support files stored in the `ATPDraw.scl` file can be customized here. Selecting the *Edit Standard* field will first perform a select file dialog box of Fig. 4.25, where the support file to be edited can be selected, then a dialog box shown in Fig. 4.26 appears.

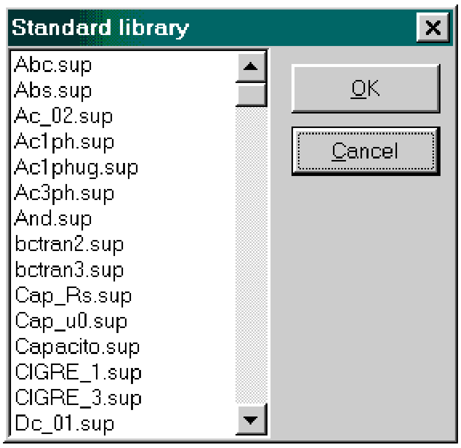


Fig. 4.25 - Specify the support file of the standard component to be edited.

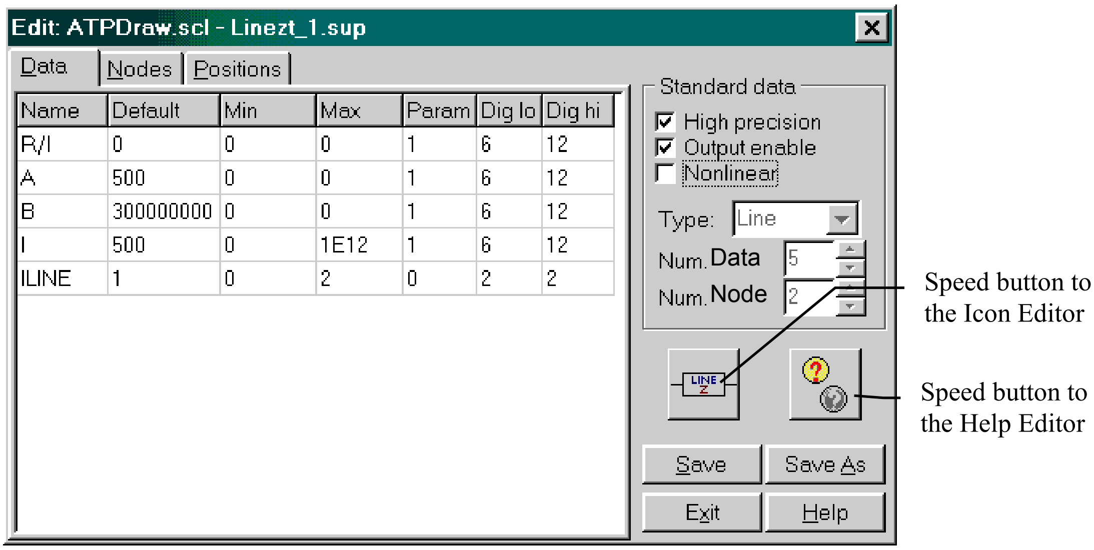


Fig. 4.26 - Control page of object data.

On the *Data* page of the *Edit Object* dialog box, control variables of the support file (one row for each object data) can be specified.

Name	The name of the parameter. Used to identify the parameter in the *Component* dialog box. This name often reflects the name used in the ATP Rule Book.
Default	Initial value of the parameter.
Min/Max	Minimum/Maximum value allowed.
Param	If set equal to 1, a variable text string can be assigned to the data value. These values are assigned under ATP|Settings/Variables.
Digits	Maximum number of digits allowed in the ATP-file. When high precision is checked, $Vintage, 1 is enabled and *Digits* is split in two values for high and low precision. The *Digits* parameter is in version 3.0 and below used only for user specified components in $INCLUDE statement.

An error message will appear in the *Component* dialog box if a parameter value is out of range. To cancel range checking, set Min=Max (e.g. set both equal to zero).

On the *Node* page of the *Edit object* dialog box, the node attributes of the support file (one row for each component node) can be specified.

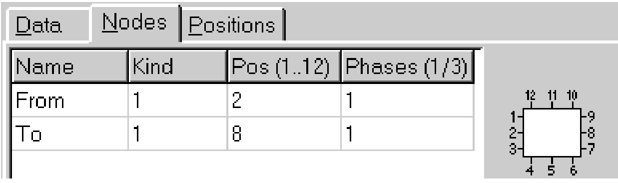


Fig. 4.27 - Node control page.

Name: The name of the node. Used to identify the node in the *Open Node* and *Component* dialog boxes.

Kind: 3-phase circuit number of the object. The number is used to handle transposition of 3-phase nodes correctly for objects having more than 3 phases. Kind=1 for all nodes of single phase objects. 3-phase nodes with the same Kind get the same phase sequence.
1: 1st to 3rd phase
2: 4th to 6th phase
3: 7th to 9th phase
4: 10th to 12th phase
The *Kind* parameter has a different meaning for MODELS or TACS component nodes. It is used to specify the type of input/output.
<u>MODELS node values:</u>
0: Output node.
1: Current input node.
2: Voltage input node.
3: Switch status input node.
4: Machine variable input node.
5: TACS variable (tacs)
6: Imaginary part of steady-state node voltage (imssv)
7: Imaginary part of steady-state switch current (imssi)
8: Output from other model. Note that the model, which produces this output, must be *USE*d before the current model. This can be done by specifying a lower group number for the model and then select the Sorting by group number option under *ATP|Settings/Misc*.
<u>TACS node values:</u>
0: Output node.
1: Positive sum input node.
2: Negative sum input node.
3: Disconnected input node.

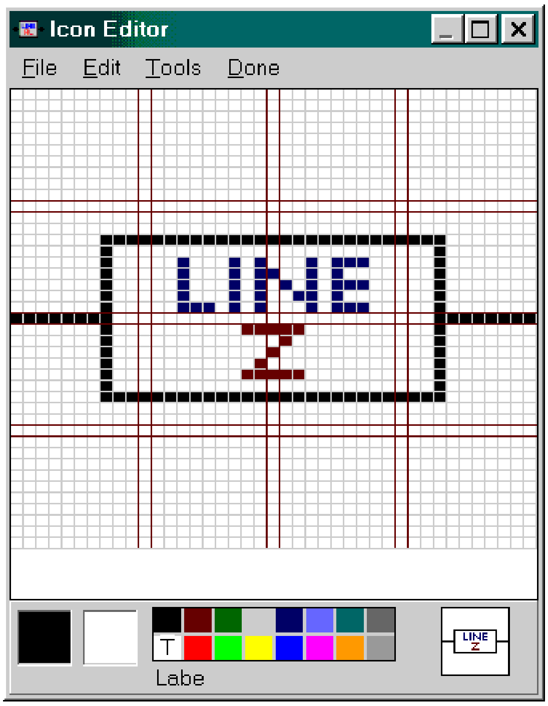
Pos: Specifies the node position on the icon border. The icon template on the right hand side of Fig. 4.27 shows the possible border positions. The position should correspond with icon drawing (positions indicated with dark red colored lines in the *Icon Editor*).

Phases: Number of phases (1 or 3) for the component node. If *Phase* is set to 3 the length of the node name is limited to 5. The last character of nodes (in the proper phase sequence according to *Kind*) will be appended by ATPDraw.

The *Position* page of the *Edit object* dialog box is not yet used in ATPDraw. This option is reserved for future use to specify the card format of the object in the `.atp` file.

Each circuit object has an icon, which represents the object on the screen. A speed button on the right hand side of the *Edit Object* dialog box invokes the built in pixel editor where icons can be edited. Each icon has equal width and height of 41x41 pixels on the screen.

Clicking with the left mouse button will draw the current color selected from a 16 colors palette at the bottom. Clicking the right button will draw with the background color. Dark red colored lines indicate the possible node positions on the icon border. Menu field items of the *Icon Editor* are described in the section 4.2.6.1 of this manual.



Each standard component has a pre-defined help file, which can be edited by a built in *Help Editor* accessible via the speed button on the right hand side of the *Edit Object* dialog. Using the help editor, users can write optional help file for the objects or add their notes to the existing help text.

Available functions and menu field items of the *Help Editor* are described in the 4.2.6.2 section of this manual.

When the user completed all modifications on the component data and on the icon and help files, the new support file can be saved to disk using *Save* (existing support file will be overwritten) or *Save As* (new file will be created in the `/USP` folder) buttons.

Fig. 4.28 - Icon Editor.

4.2.5.2 New User Specified sup-file

User specified objects are either customized standard objects or objects created for the use of $INCLUDE and Data Base Modularization feature of ATP-EMTP. The *Objects | User Specified | New sup-file* menu enables the user to create a new support file for a user specified object or customize data and node values, the icon and the help text of an existing one.

Support files of USP objects are normally located in the `/USP` folder. The *Edit Object* dialog box opens with empty *Data* and *Nodes* tabs in this menu. Number of nodes and data must be in line with the `ARG` and `NUM` declarations in the header section of the Data Base Module (DBM) file. The number of data can be in the range of 0 to 36, and the number of nodes in the range of 0 to 12. Control parameters for the object data can be entered on the *Nodes* and *Data* pages of Fig. 4.29.

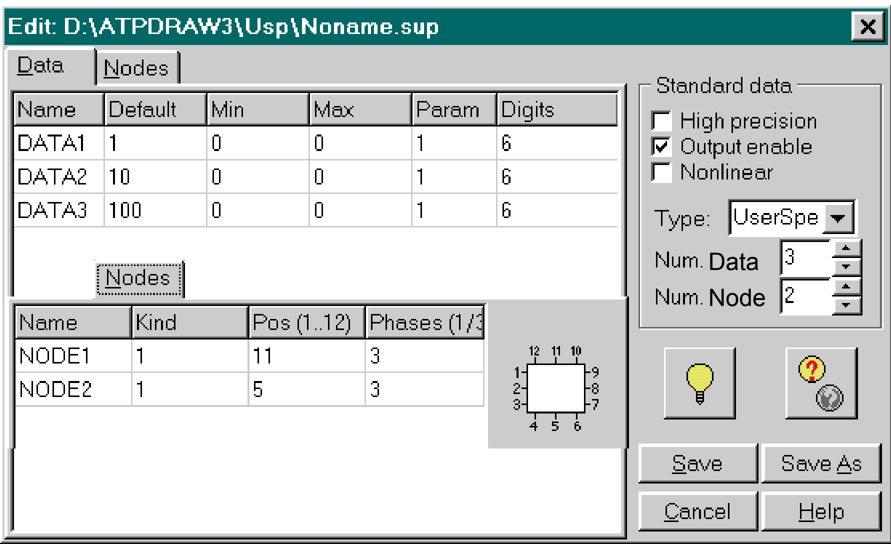


Fig. 4.29 - Control page of a new user specified object.

Each user specified components must have an icon, which represents the object on the screen and may have an optional help text assigned to it, which describes the meaning of parameters. These properties can be edited using the built in *Help* and *Icon Editor* exactly the same way as described in session 4.2.5.1.

Finally, *Save* or *Save As* buttons can be used to save the new support file to disk. User specified support files are normally located in the `\USP` folder.

4.2.5.3 Edit User Specified sup-file

An existing user specified object can be edited in the same way as any standard components as described in session 4.2.5.1.

4.2.5.4 New Model sup-file

Usage of MODELS [4] in ATPDraw is described in the Advanced Manual. To use this feature, the user first must write a model file using the built in *Model Editor* as shown in section 4.2.5.5. This file must have a legal MODELS structure (e.g. starting with MODEL name and ending with ENDMODEL), have an extension `.mod` and stored in the `/MOD` system folder. ATPDraw is capable of reading such a `.mod` file, examining its input/output and data variables and suggesting a support file on the correct format (see in section 4.9.9 and 5.5.2.2). If the user wants a different icon or other node positions on the icon border, he is free to modify the default sup-file, or create a new one by selecting the *Objects | Model | New sup-file* menu. This menu item will perform the *Edit Object* dialog as shown in Fig. 4.30.

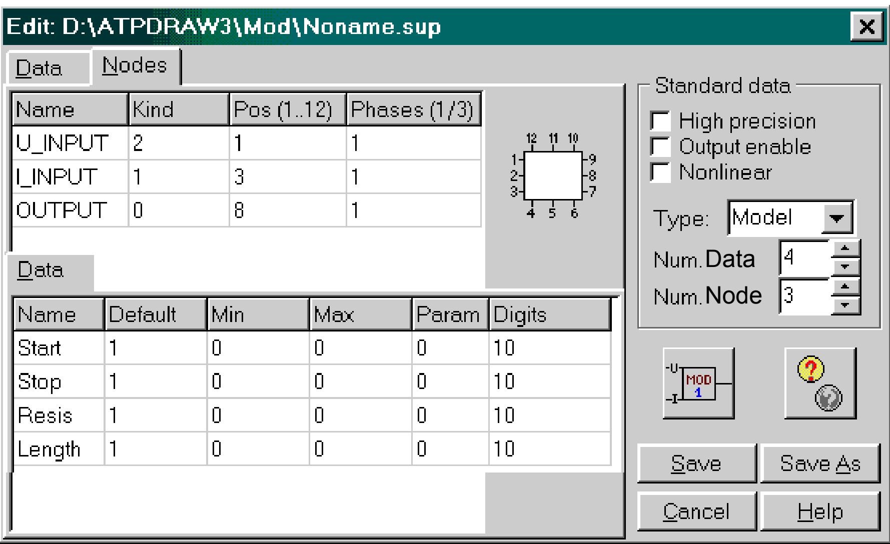


Fig. 4.30 - Control page for a New Model sup-file.

Name: Identifies the node in the *Node* and *Component* dialog boxes.
Kind: Specifies the input/output type of the node.
Pos: Specifies the node position on the icon border.
Phases: Number of phases must be set to 1 for all Models node. i.e. only single-phase nodes are supported.

```
Supported Kind values for MODELS objects are shown next:
        0:     Output node.
        1:     Current input node.
        2:     Voltage input node.
        3:     Switch status input node.
        4:     Machine variable input node.
        5:     TACS variable (tacs)
        6:     Imaginary part of steady-state node voltage (imssv)
        7:     Imaginary part of steady-state switch current (imssi)
        8:     Output from other model. Note that the model which produces this
               output must be USEd before the current model. This can be done by
               specifying a lower Group number for the model and then select the
               Sorting by group number option under ATP|Settings/Misc.
```

The number of *Nodes* is the sum of inputs and outputs to the Model. The number of *Data* must be equal to the number of DATA declarations of the actual Model. The *Kind* parameter can be changed later in the Model node input window (right click on the node dot). All model nodes are assumed a single-phase one. The maximum number of nodes is 12 and the maximum number of data that can be passed into a Model is 36.

The *Save* or *Save As* buttons can be used to save the new support file to disk. Default location of Model support files is the `\MOD` folder.

4.2.5.5 New Model mod-file

In addition to a support file and icon definition, each Model component needs a text file which contains the actual Model description. This file may be created outside ATPDraw or using the built in *Model Editor*. Selecting the *Objects* | *Model* | *New mod-file* menu, the well-known internal text editor of ATPDraw pops-up.

ATPDraw supports only a simplified usage of MODELS. It is the task of the user to write the model-file and ATPDraw takes care of the INPUT/OUTPUT section of MODELS along with the USE of each model. The following restrictions apply:

Only INPUT, OUTPUT and DATA supported in the USE statement.
Not possible to call other models under USE
Exchanging data between models is rather limited (see remarks at *Kind*=8)
Not possible to specify HISTORY of DELAY CELLS under USE

4.2.5.6 Edit Model sup-file

A model object can be edited like any other circuit object. If the user clicks on the *Objects* | *Model* | *Edit sup-file,* the well-known *Edit Object* dialog box appears with the model object controls. Here the user is allowed to customize data and node values, icon and help text of the object.

4.2.5.7 Edit Model mod-file

Selecting the *Objects* | *Model* | *Edit mod-file* menu, the well-known internal text editor of ATPDraw pops-up. Each model object has a `.mod` file which contains the description of the model. This file can be edited inside ATPDraw using the built in *Model Editor*.

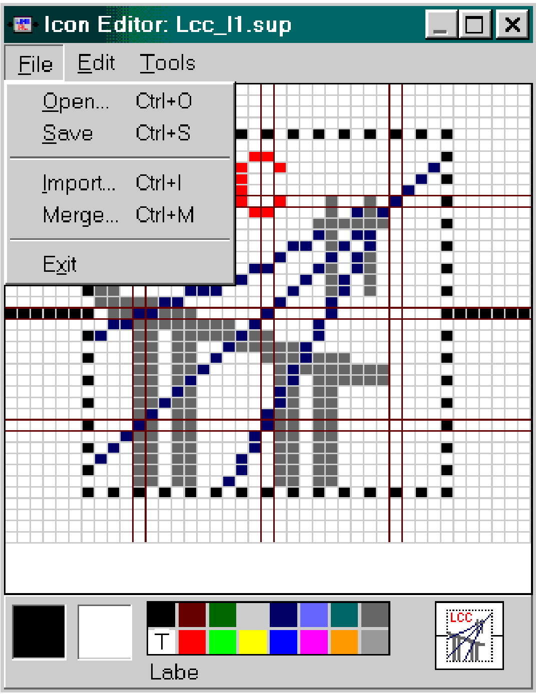
4.2.6 Tools

Items under the *Tools* menu enable you to edit component icons or help text, view or edit text files, customize several program options and save them to the `ATPDraw.ini` file. Fig. 4.31 shows the available commands of the *Tools* menu.

Fig. 4.31 - Tools menu.

4.2.6.1 Icon Editor

Brings up an icon editor shown in Fig. 4.32 where the user can edit the icon of the component. It can be invoked either from the *Edit Component* dialog box or by selecting the *Icon Editor* option in the *Tools* menu



Depending on how the editor was invoked, the file menu provides different options. When called from the *Objects* menu (*Edit Standard, User Specified* or *Edit Model sup-file)*, the user is allowed to import icons from other support files or cancel the edit operation and close the editor window. In this case, the *Done* option in the main menu is seen to accept and store the modified icon in the `.sup` file as shown on Fig. 4.28.

When the icon editor is called from the *Tools* menu, additional options like the *Open* and *Save* appears in the *File* menu.

Fig. 4.32 - Icon Editor menus.

At the bottom of the editor window there is a color palette with two boxes indicating the current foreground and background color selections, and the real-size image of the icon at right. In the color palette, the color marked with a capital letter *T* is the transparent color.

To select a color from the palette, click either the left or the right mouse button in one of the color boxes. The selected color will be assigned to the mouse button you clicked until you use the same mouse button to select another color. The leftmost box displays the color currently assigned to the left mouse button. The one to the right displays the color assigned to the right mouse button.

The foreground color is normally used to draw with, and the background color to erase any mistakes made during the drawing. It is therefore convenient to assign the transparent color (indicated by *T*) to the right mouse button, and desired drawing color to the left button. Mistakes can then easily be corrected by alternating left/right mouse button clicks.

The vertical and horizontal lines of dark red color indicate the icon node positions. These are in the same position as indicated on the *Nodes* pages of the *Edit Component* dialog boxes.

The icon editor has a *File* menu, an *Edit* menu and a *Tools* menu. In addition, a *Done* option appears to the right of the *Tools* menu if the editor has been called from the *Edit Component* dialog box. Selecting *Done,* changes made to the icon will be accepted. Available menu options are described below:

File options

Open	Loads the icon of a support file into the icon buffer.
Save	Stores the contents of the icon buffer to disk.
Import	Reads the icon of a support file and inserts it into the icon buffer.
Exit/Cancel	Closes the icon editor window. If the option Exit is selected and the icon buffer have been modified, you are given a chance to save the icon before closing. If the Done option is visible in the main menu, the name of this menu item is Cancel, and the icon editor window is closed without any warning with respect to loss of modified data.

Edit options

Undo	Cancels the last edit operation.
Redo	Cancels the undo command.
Cut	Copies a bitmap version of the icon to the Clipboard and clears the icon buffer. This bitmap can be pasted into other applications (e.g. pbrush.exe).
Copy	Places a bitmap version of the icon in the Clipboard.
Paste	Inserts the bitmap in the Clipboard into the icon buffer. If colors are different from those used in the original bitmap, it is because the icon editor calculates which color in its own color palette provides the nearest match to any bitmap color.
Delete	Clears the icon buffer.

Tools options

Pen	Selects the pen drawing tool, enabling you to draw single icon pixels, or lines or shapes by pressing and holding down the left or right mouse button while you move the mouse.
Fill	Selects the flood fill tool. Fills any shape with the current color.
Line	Selects the line drawing tool, enabling you to draw a rubber band line by pressing and holding down the left or the right mouse button while you move the mouse.
Circle	Selects the circle drawing tool, enabling you to draw a dynamically sized circle by pressing and holding down the left or the right mouse button while you move the mouse.
Rectangle	Selects the box drawing tool, enabling you to draw a rubber band box by pressing and holding down the left or the right mouse button while you move the mouse.

4.2.6.2 Help Editor/Viewer

Displays the *Help Editor* where the current help text assigned to components can be modified. The *Help Editor* and the *Viewer* has actually the same window as the built-in *Text Editor*, but with different menu options and capabilities. To edit help file of standard objects, the user must select the *Help Editor* speed button in any *Edit Component* dialog boxes. In this cases a *Done* option appears in the main menu and the *File* menu provides printing options and a *Cancel* choice. By selecting *Done* you accept any changes made to the help text. To edit help file of a *User Specified* or *Model* object, the user has two choices: to select the *Help Editor* in the *Tools* menu or to click on the *Help Editor* speed button in any *User Specified* or *Model* dialog boxes.

When the editor is called from the *Tools* menu, the *File* menu contains an *Open* and a *Save* option, as well. In that case the text buffer is initially empty, so the user must select the *File* | *Open* first to load the help text of a support file. The default font can be changed by selecting the *Font* option in the *Character* menu. This menu will bring up the Windows standard font dialog box where you can specify a new font name and character style, size or color. Note that ATPDraw does not remember the current font setting when you terminate the program, so if you don't want to use the default font, you have to specify a new one each time you start ATPDraw. The *Word Wrap* option toggles wrapping of text at the right margin so that it fits in the window.

When the built in editor is used as a viewer of component help text, editing operations are not allowed and the *File* menu provides printing options only. Additionally, the *Find & Replace* option is missing in the *Edit* menu.

The status bar at the bottom of the window displays the current line and character position of the text buffer caret, and the buffer modified status. This status bar is not visible when viewing component help. A more detailed description of menu options is given in the next sub-section.

4.2.6.3 Text Editor

To invoke the editor you may select the *Text Editor* option in the *Tools* menu or the *Edit ATP-file* or *Edit LIS-file* in the *ATP* menu. In the latter case, the file having the same name as the active circuit file with extension `.atp` or `.lis` are automatically loaded. When the program is called from the *Tools* menu, the text buffer will initially be empty.

The status bar at the bottom of the window displays the current line and character position of the text buffer caret, and the buffer modified status. The text buffer of the built in text editor is limited to 32kB therefore not be suitable for editing large files. However, any other text processor (e.g. notepad.exe or wordpad.exe) can be used, if *Text editor*: setting of the *Preferences* page in the *Tools* | *Options* menu overrides the default one.

A detailed description of the menu options are given below:

File options

New	Opens an empty text buffer. *(Built-in text editor only!)*
Open	Loads the help text of a support file or the contents of a text file into the text buffer.
Save	Stores the contents of the text buffer to disk.
Save As	Stores the contents of the text buffer to a specified disk file. *(Built-in text editor only!)*
Print	Sends the contents of the text buffer to the default printer.
Print Setup	Enables you to define default printer characteristics.
Exit/Cancel	Closes the editor or viewer window. If the option displays Exit and the text buffer has been modified, you are given a chance to save the text before closing. If a Done option is available from the main menu, this option displays Cancel, and the window will close without any warning with respect to loss of modified data.

Edit options

Undo	Cancels the last edit operation.
Cut	Copies selected text to the Clipboard and deletes the text from the buffer.
Copy	Puts a copy of the selected text in the Clipboard.

Paste	Inserts the text in the Clipboard into the text buffer at the current caret position.
Delete	Deletes any selected text from the text buffer.
Select All	Selects all the text in the buffer.
Find	Searches the text buffer for the first occurrence of a specified text string and jumps to and selects any matching text found. This option displays the Windows standard Find dialog box.
Find Next	Searches for the next occurrence of the text string previously specified in the Find dialog.
Find&Replace	Searches the text buffer for one or all occurrences of a specified text string and replaces any instance found with a specified replacement string. This option displays theWindows standard Replace dialog box.
Character options	
Word Wrap	Toggles wrapping of text at the right margin so that it fits in the window.
Font	From the Windows standard Font dialog box you can change the font and text attributes of the text buffer.

4.2.6.4 Options

In the *Tools* | *Options* menu several user customizable program options for a particular ATPDraw session can be set and save to the `ATPDraw.ini` file read by all succeeding sessions. During the program startup, each option is given a default value. Then, the program searches for an `ATPDraw.ini` file in the current directory, the directory of the `ATPDraw.exe` program, the Windows installation directory and each of the directories specified in the `PATH` environment variable. When an initialization file is found, the search process stops and the file is loaded. Any option values in this file override the default settings.

The *ATPDraw Options* dialog enables you to specify the contents of the `ATPDraw.ini` file without having to load and edit the file in a text editor. As shown on Fig. 4.33 this dialog box has four sub-pages: *General, Preferences, Directories* and *View/ATP*.

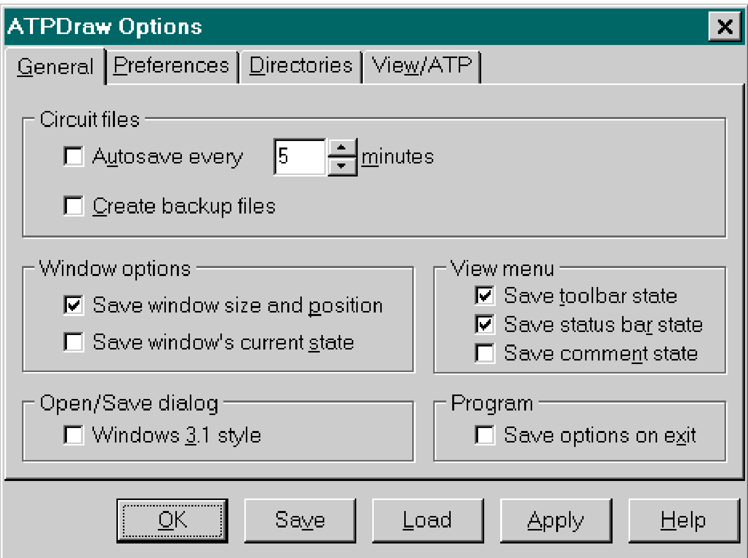


Fig. 4.33 - Customizing program options.

General

The *General* tab specifies the project file and ATPDraw main window options. The following list describes the available options:

Option	Description
Autosave every ? minutes	Saves all modified circuits to a separate disk file every specified interval of minutes. The file name is the same as the project file but with extension '.$ad'. Modified state of the circuit window does not change as a consequence of autosave operation.
Create backup files	Changes the extension of the original project file to '.~ad' each time the circuit is saved. This option does not apply to autosave operations.
Save window size and position	Records the current size and position of the main window. When ATPDraw is started next, it will be displayed with the same size and in the same position as the previous instance.
Save window's current state	Records the current main window state (maximized or normalized). The next time ATPDraw is started, it will be displayed in the same state.
Save toolbar state	Records the current view state (visible or hidden) of the main window toolbar, so it can be redisplayed in the same state next time when ATPDraw is started.
Save status bar state	Records the current view state (visible or hidden) of the main window's status bar, so it can be redisplayed in the same state next time when ATPDraw is started.
Save comment State	Records the current view state (visible or hidden) of the circuit window comment line, so it can be redisplayed in the same state next time when ATPDraw is started.
Windows 3.1 Style	Causes the Open/Save dialogs to be drawn in the Windows 3.1 style.
Save options on exit	Causes program options to be automatically saved to the initialization file when the program is terminated.

Note that the 'save state' options will have no effect unless program options are saved to the initialization file (`ATPDraw.ini`) by the *Save* command at the bottom of the *ATPDraw Options* dialog, or by selecting the 'Save options on exit' check box, or by the *Tools* | *Save Options* menu.

At the bottom of the *ATPDraw Options* dialog box the five buttons provide the following functionality:

Option	Description
OK	Stores current settings into program option variables, updates the screen and closes the dialog box. Changes made will only affect the current session.
Save	Saves the current settings to the ATPDraw.ini file.
Load	Loads settings from the ATPDraw.ini file.
Apply	Same as OK, but does not close the dialog box.
Help	Displays the help topic related to the options on the current page.

Note that, if no initialization file exists, ATPDraw will create a new file in its installation directory when the user selects the *Save* button or the *Save Options* in the *Tools* menu.

Preferences

On the *Preferences* page the user can set the size of undo/redo buffers, specify the default text editor and command files to execute ATP-EMTP (TPBIG*.EXE) and *Armafit* programs.

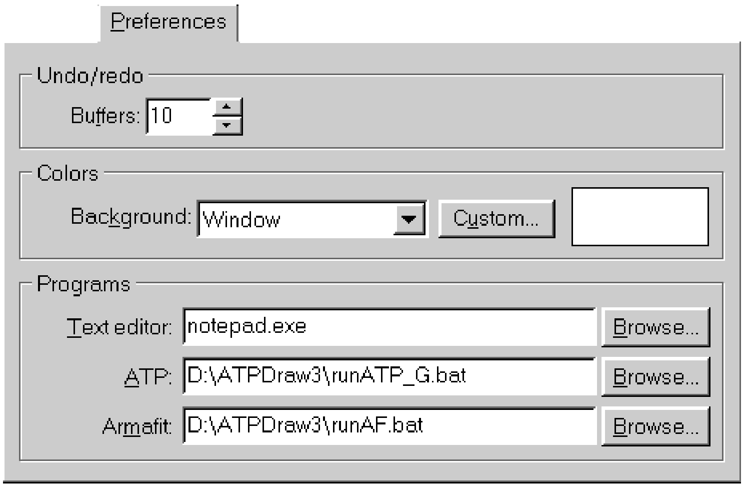


Fig. 4.34 - Customizable program options on the Preferences page.

Option	Description
Undo/redo buffers:	Specifies the number of undo and redo buffers to allocate for each circuit window. Changing this option does not affect the currently open circuit windows; only new windows will make use the specified value. Almost all object manipulation functions (object create, delete, move, rotate, etc) can be undone (or redone). These functions also update the circuit's modified state to indicate that the circuit needs saving. During an undo operation, the modified state is reset its previous value, so if you undo the very first edit operation, the '*Modified*' text in the status bar will disappear. Any operation undone can be redone. Since only a limited number of buffers are allocated, you are never guaranteed to undo all modifications. For example, if the number of undo/redo buffers is set to 10 (default) and eleven successive modifications to the circuit are made, the first modification can no longer be undone, and the modified state will not change until you save the circuit.
Background color:	Selects the background color of circuit windows. The color list provides available system colors, but you may customize your own from the Windows standard Color dialog displayed by the Custom button. The current color selection is shown in the box to the right of the Custom button.
Text editor program:	Holds the name and path of the text editor program to use for editing ATP-files (e.g. notepad.exe or wordpad.exe). If no program is specified (the field is empty), the built-in text editor will be used. Note that the program specified here must accept a filename on the command-line; otherwise the ATP-file will not be automatically loaded by the editor.
ATP:	Holds the ATP program command, which is executed by the *run ATP* command (or *F2* key) at the top of the *ATP menu*. A batch file is suggested as default (runATP_S.bat for the Salford, runATP_W.bat for the Watcom and runATP_G.bat for the MingW32/GNU versions). Watcom/GNU versions can also be executed directly as %WATDIR% TPBIGW.EXE DISK $$ * -r or %GNUDIR%TPBIGG.EXE DISK $$ s -r where $$ replaces the %1 sign normally used in a batch file.

ARMAFIT	Holds the name of the Armafit program used for NODA line/cable models. A batch file runAF.bat is suggested.

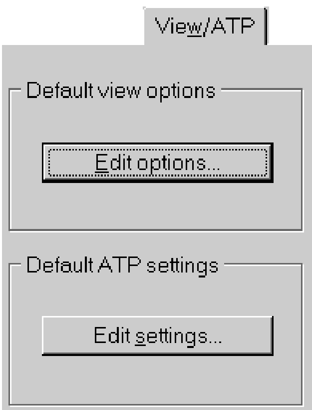
Directories

The following table describes the available options on the *Directories* page:

Option	Description
Project folder	The directory where ATPDraw stores the project files (.adp).
ATP folder	Specifies the directory in which .atp files are created.
Model folder	Directory containing support (.sup) and model (.mod) files for MODELS components.
Group folder	The container of the group object support files (.sup). The group support files are automatically deleted when the ATPDraw is closed.
User spec. folder	Directory containing support (.sup), library (.lib) and punch (.pch) files for user specified components.
Line/Cable folder	Default folder for the line and cable models. This folder will contain .alc files (ATPDraw line/cable data), intermediate .atp and .pch files, and .lib files (include). If the .alc files are stored in that directory, the resultant .lib files used in $Include statements in the final ATP input file are also stored in this directory. The $Prefix/$Suffix option should in this case be turned off. The Noda format in ATP does not allow to specify the full path for $include files. Therefore, Noda lines (.alc files) must be stored in the same directory as the final ATP-file.
Bctran folder	The default folder for BCTRAN multi-phase, multi-winding linear transformer models. This folder will contain .bct files (ATPDraw Bctran data), intermediate .atp, .pch and .lis files.

View/ATP

Two groups of options can be specified in the *View/ATP* page. These are the *Default view options* and the *Default ATP settings*.



The *Edit options* button opens the *View Options* dialog, which enables you to specify view options to apply as default to all new circuit windows. Available options are described in section 4.2.3.9.

Note that all circuit windows maintain their own set of view options, and only the new circuit windows you open will use the options specified here. To change the view options of an existing circuit window, select the *Options* item in the *View* menu (section 4.2.3.9).

Fig. 4.35 - Setting default view and ATP options.

The *Edit settings* button calls the *ATP settings* dialog described in section 4.2.4.1 of this manual. ATP settings specified here will be applied as default to all new project files.

Note that all circuits have their own settings; stored together with the objects in the project files. The settings specified here will only be used by the new circuits you create. To customize ATP settings of an existing project select the *Settings...* item in the *ATP* menu or press *F3* function key.

4.2.6.5 Save Options

Saves program options into the `ATPDraw.ini`. This file is normally located in the program installation directory and can be used to store default options and settings. Description of variables in the `ATPDraw.ini` file is given in Appendix.

4.2.7 Window

The *Window menu* contains options for activating or rearranging circuit windows and showing or hiding the *Map window*.

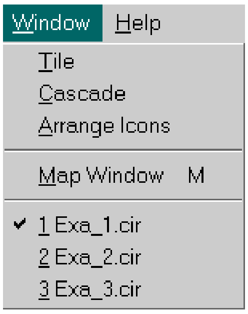


Fig. 4.36 - Supported options on the Window menu.

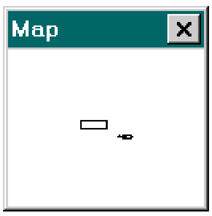
Tile
The *Tile* command arranges the circuit windows horizontally in equal size on the screen. To activate a circuit, click the title bar of the window. The active circuit window is marked by a ✓symbol in front of the circuit file name.

Cascade
The *Cascade* command rearranges the circuit windows so that they overlap such a way that the title bar remains visible. To activate a circuit click the title bar of the window.

Arrange Icons
The *Arrange Icons* command arranges the icons of minimized circuit windows so that they are evenly spaced and don't overlap.

4.2.7.1 Map Window

The *Map Window* command (Shortcut: *M*) displays or hides the map window. The map window is a stay-on-top style window, meaning that it will always be displayed on top of all other windows. You can show or hide the map by pressing the *M* character of the keyboard to enable it when you need it, or hide it when it conceals vital circuit window information.



The map window displays the entire contents of the active circuit. The circuit window itself is represented by a map rectangle and the circuit objects are drawn as black dots.

Fig. 4.37 - Map window.

When you press and hold down the left mouse button in the map rectangle, you can move it around in the map window. When the mouse button is released, the circuit window displays the part of the circuit defined by the new rectangle size and position, and the circuit window scrollbars are repositioned to reflect the updated circuit view.

If any circuit objects are currently selected when you reposition the map rectangle, selected objects will also be moved, and their relative position retained in the new window. This functionality can be used to quickly move a collection of objects a relatively large distance.

4.2.8 Help

The *Help menu* contains options for displaying the on-line help of ATPDraw, and the copyright and version information.

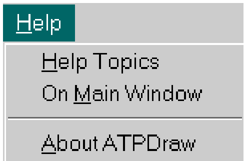


Fig. 4.38 - Help menu.

ATPDraw's on-line help is a standard Windows dialog, which provides help on all Main menu options and also gives a short introduction how to build up a circuit.

4.2.8.1 Help Topics

The *Help Topics* command invokes the MS-Windows standard help dialog box. Several links and a relatively large index register support the users in searching. Selecting the *Contents* tab you get a lists of available help functions as shown on Fig. 4.39.

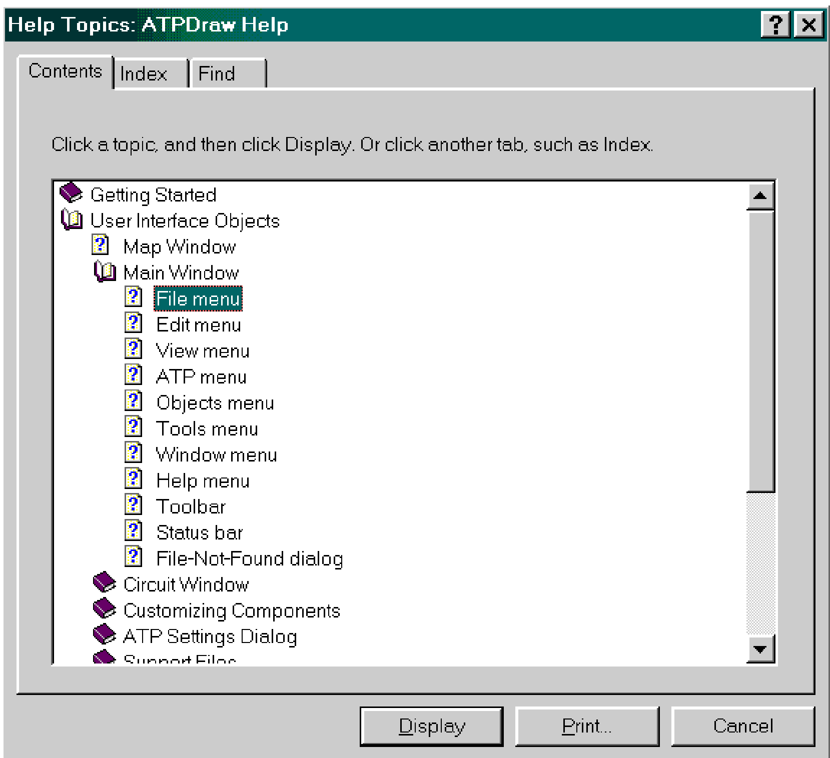
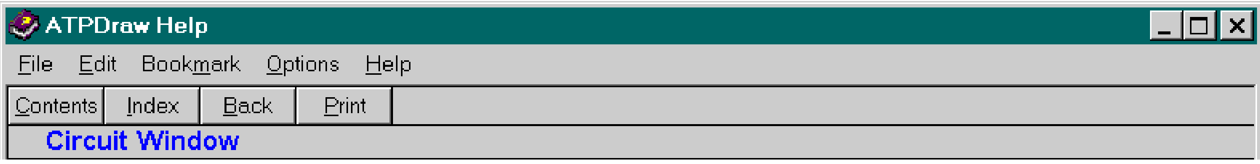


Fig. 4.39 - On-line help of ATPDraw.

This page allows you to move through the list and select an entry on which you need help. To display an entry select one from the list by a simple mouse click and press *Display*, or double click on the entry with the mouse.

Index and *Find* tabs can be used to get help by the name of a topic. E.g. if you ask for help on topics “Circuit Window” type this phrase into the input field of the *Index* page and press the *Display* button. Then a description of the Circuit Window topics will be displayed as shown below including several links marked as green underlined text:



The circuit window is the container of circuit objects. From the file menu you can load circuit objects from disk or simply create an empty window to start building a new circuit. Circuit objects include components (standard, user specified, MODELS and TACS), connections and relations.

The resolution of a circuit is 5000x5000 pixels (screen points); much higher than your screen normally would support. Consequently, the circuit window displays only a small portion of the circuit. To move around in the circuit you use the window scrollbars, or you drag the view rectangle of the map window to another position. You may also want to use the zoom options in the view menu in order to zoom in or out on objects.

From the components menu you select components to insert into the circuit. This menu pops up when you click the right mouse button in an empty space of the circuit window. To start drawing a new connection, click the left mouse button on a component node or the end-point of an existing connection. An inverted connection line will then follow the mouse cursor until you finish the drawing operation by clicking the left mouse button again, or canceling it by clicking the right mouse button. Relations are drawn in much the same way, except that you have to select the TACS | Draw relation option in the components menu to start the relation drawing operation. You can then draw multiple relations until you click the right mouse button. Relations are used to visualize information flow into Fortran statements and are drawn as blue connections, but have no influence on component connectivity.

To select and move an object, simply press and hold down the left mouse button on the object while you move the mouse. Release the button and click in an empty space to unselect and confirm its new position. The object is then moved to the nearest grid point (known as gridsnapping). If two or more components overlap as a consequence of a move operation, you are given a warning and can choose to cancel the un-selection.

If you select a single component and press the Crtl+F1 key combination, component specific help is displayed.

Selecting a group of objects for moving can be done in three ways: If you hold down the Shift key while you left-click an object, you add it to the selected objects group. Pressing and holding down the left mouse button in an empty space enables you to drag a rectangular outline around the objects you want selected. And finally, if you double-click the left mouse button in an empty space, you can define a polygon shaped region by repeatedly clicking the left mouse button in the circuit window. To close the region, click the right mouse button on the last polygon point you want to set. Objects that are defined to fall within the indicated region or rectangle are added to the selected objects group. For components this means that the center point of a component icon must lie within the defined region or rectangle. For connections and relations the region or rectangle must surround both end-points. To move the selected group of objects, press and hold down the left mouse button inside the group while you move the mouse. You unselect and confirm the new position by clicking in an empty space. Any overlapping components will produce a warning.

To move objects outside the visible part of the circuit, use the window scrollbars or the view rectangle in the map window. Any selected object or group will follow the window to its new position.

You rotate objects by clicking the right mouse button inside the selected object or group. Other object manipulation functions, such as undo/redo and clipboard options, are found in the edit menu. However, the most frequently used object manipulation functions can be accessed by holding down the Shift key while clicking with the right mouse button on an object or a selected group of objects. This will display and activate the circuit window shortcut menu.

Components and component nodes can be opened for editing. If you right-click or double-click an unselected component or node, either the Component, Open Probe or Open Node dialog box will show, allowing you to change component or node attributes and characteristics. If you double-click in a selected group of objects, the Open Group dialog box will show, allowing you to change attributes common to all components in that group, such as group number and hide and lock state. Default component attributes are stored in support files. Access to create and customize support files is provided by the objects menu.

Components are connected if their nodes overlap or if a connection is drawn between the nodes. To draw a connection between nodes, click on a node with the left mouse button. A line is drawn between that node and the mouse cursor. Click the left mouse button again to place the connection (clicking the right button cancels the operation). The gridsnap facility helps overlapping the nodes. Nodes connected together are given the same name by the Make Names and Make File options in the ATP menu. Nodes can be attached along a connection as well as at connection end-points. A connection should not unintentionally cross other nodes (what you see is what you get). A node naming warning appears during the ATP file creation if a connection exists between nodes of different names or if the same name has been given to unconnected nodes. Connections are selectable as any other object. To resize a connection, click on its end-point with the left mouse button, hold down and drag. If several connections share the same node, the desired connection to resize must be selected. Selected connection nodes appear as squares at both ends of the selection rectangle.

4.2.8.2 On Main Window

The menu item *On Main Window* displays help about the ATPDraw main window.

4.2.8.3 About ATPDraw

Selecting this menu item shows the ATPDraw copyright information and the program version actually used.

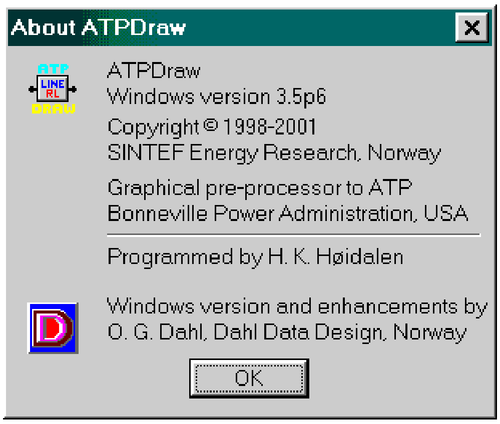


Fig. 4.40 - About window of ATPDraw.

4.3 Shortcut menu

The *Shortcut menu* provides access to the most frequently used object manipulation functions. To show and activate the shortcut menu, hold down the *Shift* key while you click the right mouse button on an object or a selected group of objects in the circuit window. Most of the items on this menu are identical with that of the *Edit menu* (section 4.2.2). The *Open* menu item at the top of the menu is an addition to these normal edit functions. If this command is performed on a single object, the *Component* dialog box appears. If you select this command for a group of selected objects, the *Open Group* dialog box appears.

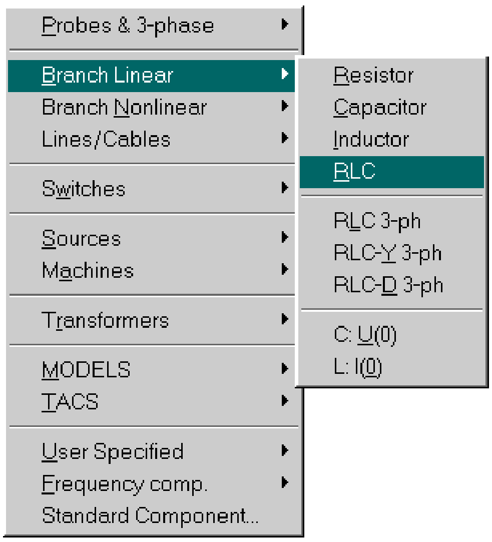
Open: Enables the component customization by bringing up the Component dialog box of the object.
Cut, Copy: Provides access to the standard clipboard functions
Delete, Duplicate
Flip, Rotate: Rotates and flips the objects' icon
Select/Unselect: Select/unselect the object(s)
Compress: replace a group of selected objects with a single icon

Extract: The group is extracted on the current circuit layer.
Edit Group: The group is extracted in a separate window. Here it can be edited with some limitation.
Edit Circuit: Displays the circuit to which the current group belongs.
Actually, the grouping structure can be taken as a multi-layer circuit, where the *Edit Group* brings the user one step down in details, while *Edit Circuit* brings one step back.

Fig. 4.41 - Available options in the Shortcut menu.

4.4 Component selection menu

The *Component selection menu* provides options for inserting new components into the circuit window. This menu is normally hidden. To open it you must click on the right mouse button in an empty area of the circuit window. The component selection menu collects all the available circuit objects of ATPDraw in a structured way as shown in Fig. 4.42. After selecting a component in one of the floating menus, the selected object is drawn in the circuit window.



The upper section of the menu provide access to the probe, splitter and transposition and reference objects, the next four to many standard ATP components: linear and nonlinear elements, lines and cables, switches, sources, electrical machines and transformers. The next section is dedicated for MODELS and TACS components. User specified objects and Frequency dependent components for Harmonic Frequency Scan (HFS) studies are accessible in the next group. The lowermost section lists all the supported components in a scrollable menu where any object can be selected upon its support file name.

Fig. 4.42 - Component selection menu.

4.5 Component dialog box

After selecting a component in the *Component selection menu* the new circuit object appears in the middle of the circuit window enclosed by a rectangle. Click on it with the left mouse button to move, or the right button to rotate, finally click in the open space to unselect and place the object. The *Component* dialog box appears when you click the right mouse button on a circuit object (or double click with the left mouse). Assuming you have clicked on the icon of an RLC element, a dialog box shown in Fig. 4.43 appears. These dialog boxes have the same layout for all circuit objects except probes, which can be edited from the *Probe* dialog box.

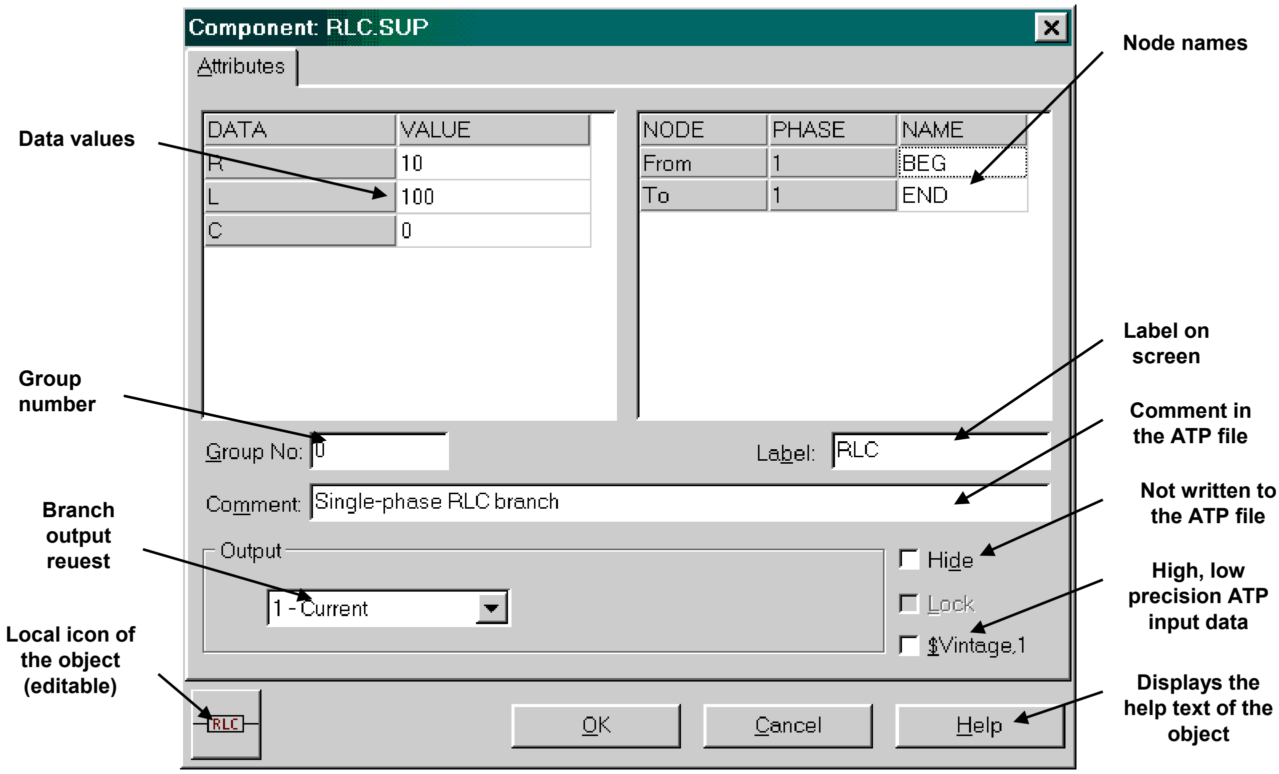


Fig. 4.43 - The Component dialog box.

Component data can be entered in the *Value* field of the *Attributes* page. The *Node*, *Phase* and *Name* fields are initially empty and you are not allowed to enter values here. After having node attributes specified by either the user in the *Node* dialog box, or by the *Make name* or *Make File As...* procedure of ATPDraw the component dialog will exhibit the real values of these fields, too.

Numerical values in the data input fields can be specified as real or integer, with an optional exponential integer, identified by 'E' or 'e'. Many data parameters have a legal range specified. To check this legal range, place the input caret in a data field and press the Ctrl+F1 keys. If you specify an illegal value, an error message is issued when you move to another data field, or select the *OK* button. The user is also free to assign a 6 (or less) character text string as input data for most of the standard components. Numerical values can later be assigned to these variables under *ATP|Setting/Variables* using the `$PARAMETER` feature of ATP-EMTP (see in 4.2.4.1).

Just below the data input column, there is a *Group No* input field, where an optional group number can be specified to the object. It can be used later as optional sorting criteria (the lowest group number will be written first in the ATP-file) on the *ATP | Settings / Format* page.

The content of the *Label* input text field is written on the screen and into the project file, too. The visibility of the component label is controlled by the *Labels* option in the *View | Options* dialog box. The label is movable on the screen. The component dialog box has a *Comment* input text field. If you specify a text in this field, it will be written to the ATP-file as a comment (i.e. as a comment line before the data of the object).

Many standard component such as branches, non-linear, switches and transformers contains an *Output* section for setting the branch output request in a combo box. Possible values are Current, Voltage, Current&Voltage, Power&Energy or none.

Like the *Group No*, *Label* and *Comment* fields, the *Hide* and *Lock* buttons are common to all components. Hidden components are not included in the ATP-file and are displayed as light gray icons in the circuit window. The Lock option is not implemented in this version. Locked components are meant to have fixed positions and not subject for customization.

All components where the high precision format is available has a *$Vintage, 1* check button in the component dialog box. It is thus possible to control the precision format for each individual component. Selecting *Force high resolution* under the *ATP | Settings/Format* page will overrule the individual setting and force *$Vintage, 1* for all components if possible.

The non-linear components (non-linear branches, saturable transformers, and TACS Device 56) have a *Characteristic* page too, as shown in Fig. 4.44.

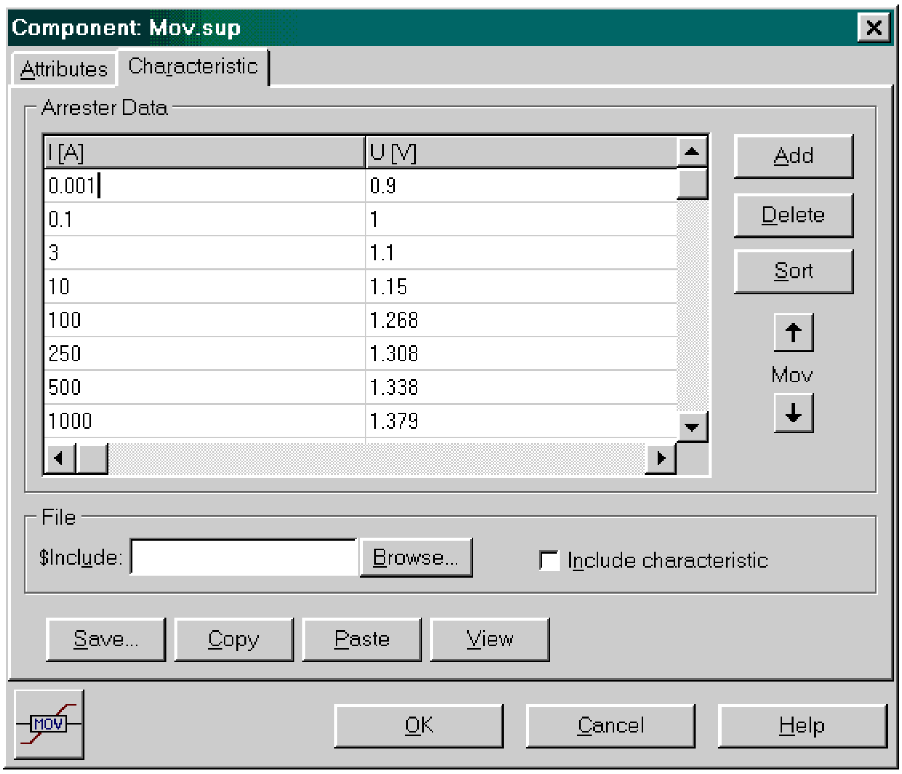


Fig. 4.44 - The Characteristic page of non-linear components.

On the *Characteristic* tab of the dialog box, you define the input characteristic for non-linear components. Data pairs can be specified in a standard string grid. To add new points after the cursor position, click on *Add*. Delete the marked point by clicking on *Delete*. You can manipulate the order of points by the *Sort* button (the characteristic for non-linear components is automatically sorted after increasing x-values, starting with the lowest number) or the ↑ and ↓ arrows. The user can edit the data points directly any time.

It is possible the export the characteristic to an external file or to the Windows clipboard as text. The whole characteristic is copied (no marking is supported or required). You can also paste a characteristic from the clipboard. It is thus possible to bring an old `.atp` file up in a text editor, mark the characteristic (the flag 9999 is optional) and copy it to the clipboard, then paste it into the characteristic page. The number of points will automatically be adjusted (the pasted characteristic could be truncated to ensure that the number of data is less or equal to 36). Therefore, you do not have to click on *Add* or *Delete* buttons before pasting. ATPDraw uses fixed format 16 character columns to separate the numbers. Note! Pasting in from a text file with 'C' in the first column is not possible; Delete leading 'C' characters first.

The *File* section at the bottom of the page contains an *$Include* field where you can specify the name of a standard text file containing nonlinear characteristic. If the *'Include characteristic'* button is checked, this file will be referenced in the `$INCLUDE` statement in the ATP-file rather than including each of the value pairs from the points table. This file must have extension `.lib` and be stored in the `\USP` system folder if the *Insert $Prefix and $Suffix cards* check box is selected in the *ATP* | *Settings / Format* menu. If the file is located outside `\USP`, unselect the *$Prefix and $Suffix* settings and use the *Browse* button to specify the complete path to the file.

The nonlinear characteristic specified by the user can be displayed by clicking on the *View* button. In the *View Nonlinearity* window (Fig. 4.45) the min and max axis values are user selectable as well as the use of logarithmic scale (if min>0). It is also possible to copy the graphic to the Windows clipboard in a metafile format. The Add (0,0) check box will add the origo point graphically only. Selecting *Done* will close the nonlinearity display.

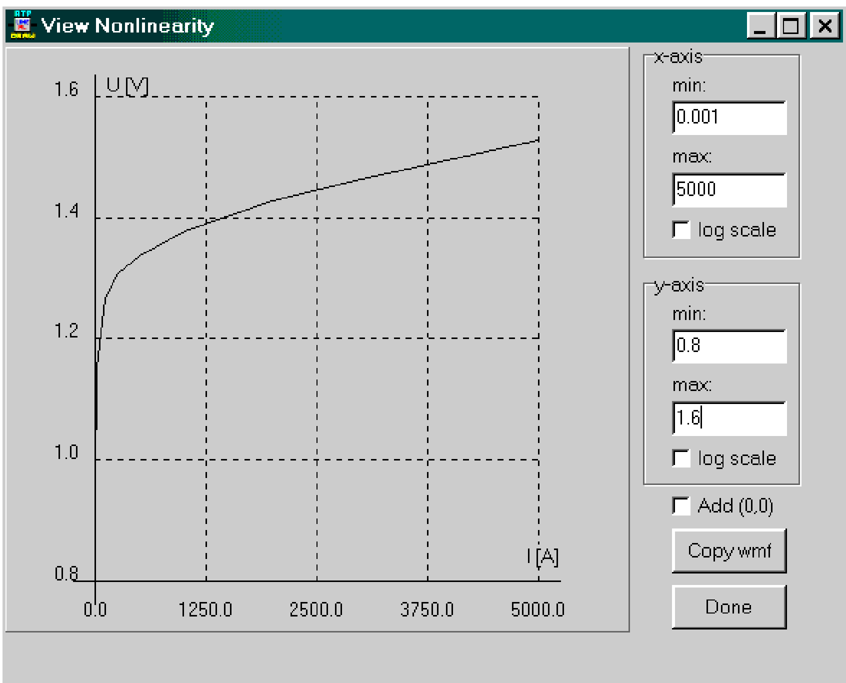


Fig. 4.45 - The View nonlinearity window.

The *OK* button will close the dialog box and the object data and all properties are updated in the data structure. Then the red drawing color of the object icon will be turned off, indicating that the object now has user specified data. When you click on the *Cancel* button, the window will be closed without updating. The *Help* button calls the *Help Viewer* to show the help text of the object. Further help about the *Component* dialog is also available through the Windows standard on-line help system of ATPDraw if you press the *F1* key.

The following components deviate somewhat from the above description and will be referenced in the Advaced part of this Manual:

General 3-phase transformer (GENTRAFO)
Universal machine (UM_1, UM_3, UM_4, UM_6, UM_8)
Statistical switch (SW_STAT)
Systematic switch (SW_SYST)
Harmonic source (HFS_SOUR)
BCTRAN transformer (BCTRAN3)
Line/Cable LCC objects (LCC_x)

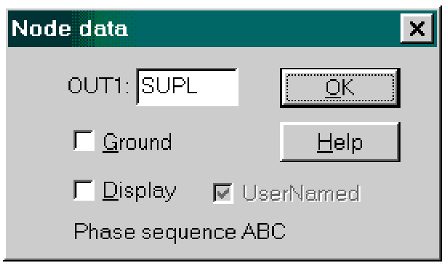
Depending on the type of component opened, the group box in lower-left corner of the *Attributes* page may display additional options:

a) you can specify here the name of the `.mod` file name of the MODELS object and the *Use As* string. If the file name does not include a path, the file is expected to exist in the `/MOD` folder.
b) for the Fortran TACS components ATPDraw provides an extra `OUT` field here to specify the Fortran expression.
c) for user specified components you specify the name of the library file in the *$Include* field. If *Send parameters* option is selected, the *Internal phase seq.* controls how the node names are passed. i.e. unselect this option if your library file expects 5-character 3-phase node names. If the library file name does not include a path, the file is expected to exist in the `/USP` folder.

4.6 Node dialog box

In the *Node data* dialog box you specify data for a single component node. Input text in this dialog boxes should contain only ASCII characters, but characters like * - + / $ etc. should not be used. Avoid using space in the node name and lower case letters, as well. The user does ***not*** need to give names to all nodes, in general. The name of the nodes without special interest are recommended to be left unspecified and allow ATPDraw to give a unique name to these nodes. The node dots given a name by the program are drawn in red, while those whose names were specified by the user are drawn with black color.

There are four different kinds of nodes, each treated slightly different in this dialog box:



1) Standard and user specified nodes
2) MODELS object nodes
3) TACS object nodes
4) TACS controlled machine nodes

Fig. 4.46 - Node dialog box for standard components.

Parameters common to all nodes are:

Name A six or five (3-phase components) characters long node name. The parameter caption is read from the support file. If you try to type in a name on the reserved ATPDraw format (XX1234 for single phase or X1234 for three-phase nodes) you will be warned. Ignoring this warning can result in unintentional naming conflicts.

Display If checked, the node name is written on screen, regardless of the current setting of the Node names option in the *View | Options* dialog box.

UserNamed This checkbox shows whether this node name is specified by the user or ATPDraw. If the user wants to change a user specified node name he must do this where the *UserNamed* box is checked. If not, duplicate node name warnings will appear during the compilation. Node with UserNamed set are also drawn with a black node dot.

The following list explains the type specific node parameters:

<u>Standard and USP components:</u>

Ground If checked, the node is grounded.

<u>MODELS node:</u>

Type
0=Output.
1=Input current (i)
2=Input voltage (v)
3=Input switch status (switch)
4=Input machine variable (mach)
5=TACS variable (tacs)
6=Imaginary part of steady-state node voltage (imssv)
7=Imaginary part of steady-state switch current (imssi)
8=Output from other model. Note that the model that produces this output must be *USE*d before the current model. This is done by specifying a lower group number for the model and then select the *Sorting by group number* option under *ATP | Settings / Format*.

<u>TACS node:</u>

Type
0=Output.
1=Input signal positive sum up.
2=Input signal negative sum up.
3=Input signal disconnected. (necessary only if the node name is user specified)

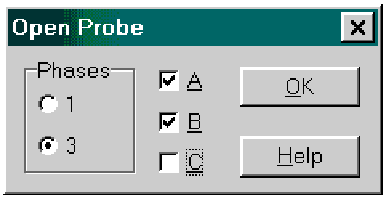
<u>TACS controlled machine node:</u>

Type
0=No control.
1=D-axis armature current. Out.
2=Q-axis armature current. Out.
3=Zero-sequence armature current. Out.
4=Field winding current. Out.
5=D-axis damper current. Out.
6=Current in eddy-current winding. Out.
7=Q-axis damper current. Out.
8=Voltage applied to d-axis. Out.
9=Voltage applied to q-axis. Out.
10=Zero-sequence voltage. Out.
11=Voltage applied to field winding. Out.
12=Total mmf in the machines air-gap. Out.
13=Angle between q- and d-axis component of mmf. Out.
14=Electromagnetic torque of the machine. Out.
15=Not used.
16=d-axis flux linkage. Out.
17=q-axis flux linkage. Out.
18=Angle mass. Out.
19=Angular velocity mass. Out.

```
20=Shaft torque mass. Out.
21=Field voltage. In.
22=Mechanical power. In.
```

4.7 Open Probe dialog box

Probes are components for output of node- or branch voltages, branch current or TACS values, and are handled differently than other components you open. In the *Open Probe dialog* you can specify the number of phases of a probe and which phases to produce output in the PL4-file. There are four different probes exist in ATPDraw:

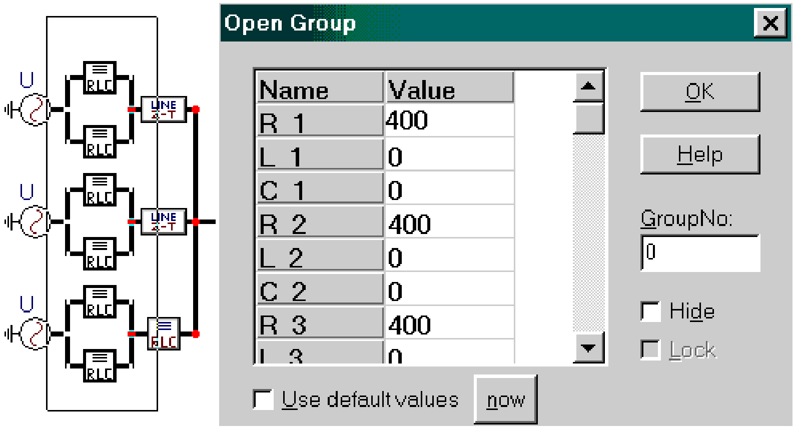


Probe_v:	Node voltages output request.
Probe_b:	Branch voltage output request.
Probe_i:	Branch current output request.
Probe_t:	TACS variable output request. Type 33.

Fig. 4.47 - Node dialog box for standard components.

4.8 Open Group dialog box

If you double-click in a selected group of objects, the *Open Group dialog* box will appear, allowing you to change attributes common to all components in that group, such as data values, group number and hide and lock state. The common data parameters are listed in a dialog as of Fig. 4.48 where you can change the data for all the involved components, simultaneously. The data labels from the support files are used to classify the data.



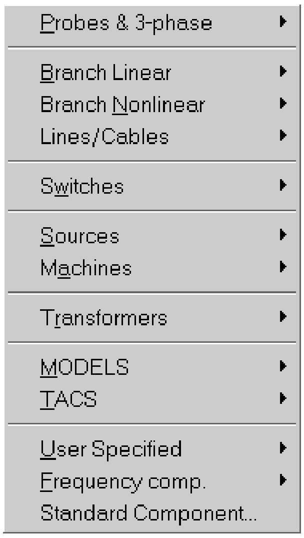
An alternative way to change the data parameter for several component simultaneously is to use `$PARAMETER` feature (see Fig. 4.20 in section 4.2.4.1).

Fig. 4.48 - Open Group dialog box for simultaneous data setting

Every component has a group number. By specifying a value in the *Group No* field, all components in the selected group of objects are assigned the same number. The group number serves as an optional sorting criterion for the ATP-file (components with the lowest group number are written to the `.atp` file first).

The *Hide* state of multiple components can also be specified. Hidden components are not included in the ATP-file and are displayed as gray icons. The *Lock* option is not yet implemented. You can also choose to reload the default values from the support files by clicking on the *now* button. Selecting the *Use default values* check box will cause default values to be loaded automatically next time the dialog box is opened.

4.9 Circuit objects in ATPDraw



The *Component selection menu* provides options for creating and inserting new components into the circuit window. This menu is normally hidden. To show and activate the menu, click the right mouse button in an empty circuit window space. Following a selection in one of the floating sub-menus, the selected object will be drawn in the center of the active circuit window enclosed by a rectangle. You can move (left mouse click and drag), rotate (right mouse button) or place the object (click on open space).

The *Component selection menu* has several sub-menus; each of them include circuit object of similar characteristics as briefly described below:

Fig. 4.49 - Component selection menu.

Probes & 3-phase

- Probes for node voltage-, branch voltage, current-, and TACS monitoring
- Various 3-phase transposition objects
- Splitter (coupling between 3-phase and single phase circuits)
- ABC/DEF Reference objects for specifying the master node for phase sequence

Branches

- Branch linear: 1-phase and 3-phase non-coupled components
- Branch nonlinear: 1-phase nonlinear R and L components. Single and 3-phase MOV. Type-93, 96 and 98 nonlinear inductors including initial conditions for the fluxlinked reactors
- TACS controlled and time dependent resistor

Lines/Cables

- Lumped, PI-equivalents (type 1, 2...) and RL coupled components (type 51, 52...)
- Distributed lines of constant, frequency independent parameters. Transposed (Clarke) up to 9-phases, untransposed 2 or 3-phase (KCLee) line models.
- LCC, the user can select 1-9 phase models of lines/cables. In the input menu of these components, the user can specify a LINE CONSTANT or CABLE PARAMETER data case. The resulting include file contains the electrical model and the LIB-file is generated automatically if the ATP setup is correct. Bergeron (KCLee/Clarke), nominal PI, JMarti, Semlyen and Noda models are supported.
- Read PCH-file. This is a module in ATPDraw to read the punch-files from Line Constants, Cable Constants or Cable Parameters and to create an ATPDraw object automatically (sup-file and lib-file). ATPDraw recognizes: PI-equivalents, KCLee, Clarke, Semlyen, and JMarti line formats.

Switches

- Time and voltage controlled. 3-phase time controlled switch
- Diode, thyristor, triac
- Simple TACS controlled switch
- Measuring switch
- Statistic and systematic switches

Sources

- AC and DC sources, 3-phase AC source. Ungrounded AC and DC sources.
- Ramp sources
- Surge sources
- TACS controlled sources

Machines

- Type 59 synchronous machine
- Universal machines (type 1, 3, 4, 6, and 8)

Transformers

- Single phase and 3-phase ideal transformer
- Single phase saturable transformer
- 3-phase, two- or three-winding saturable transformer
- 3-phase, two winding saturable transformer, 3-leg core type of high homopolar reluctance
- BCTRAN. Automatic generation of .pch file. 1-3 phases, 2-3 windings. Auto-transformers, Y-, and D- connections with all possible phase shifts. External nonlinear magnetizing inductance(s) supported.

MODELS

- Under MODELS the user can select a model component either by specifying a sup-file or a mod-file. If a .mod-file is selected the corresponding sup-file required by ATPDraw is created automatically (if the model is recognized successfully). A mod-file is a text file in the MODELS language. The mod-file must have a name equal to the name of the model. The following restrictions apply when ATPDraw reads a mod-file:
  - Not allowed with indexed input, output or data variables.
  - Names of all input, output and data variables must be less than 6 characters.

  Only input, output, data and variables declared in front of TIMESTEP, INTERPOLATION, DELAY, HISTORY, INIT and EXEC are recognized by ATPDraw when reading the mod-file.
- Type 94: When selecting this menu item the user is requested to specify a mod-file describing the Type-94 models component. ATPDraw then examines this file and creates the required sup-file for the component. The same rules as specified under MODELS apply. ATPDraw does not check if the format of the mod-file corresponds to Type-94 declarations.

TACS

- Coupling to Circuit. Input to TACS from the circuit must be connected to this object.
- 4 types of TACS sources: DC, AC, Pulse, Ramp.
- Transfer functions: General Laplace transfer function. If the Limits are not specified or connected, no limits apply. Simple Integral, Derivative, first order Low and High Pass transfer functions.
- TACS devices. All devices except Type-55.
- Initial condition for TACS objects (Type-77)
- Fortran statements: General Fortran statement (single line expression). Simplified Math statements or Logical operators.
- Draw relations. Relations are drawn in blue and are used just to visualize connections between Fortran statements and other objects. Relations will not affect the ATP input file.

User specified

- Library: $Include is used to include the lib-file into the ATP input file. The user must keep track of internal node names in the include file.
- Files: $Include is used to include the user specified lib-file into the ATP input file and pass node names and data variables as parameters. Each user specified objects must have a sup-file containing an icon, specification of data and node parameters, and a lib-file describing the component in the Data Base Module format of ATP.
- Single and 3-phase reference: These objects are not represented in the ATP input data file and serve only as visualization of connectivity.

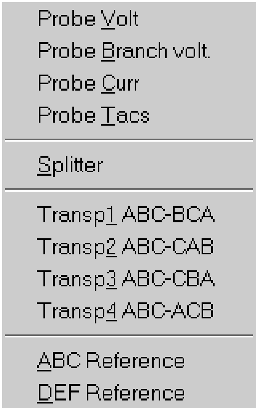
Frequency components

- Harmonic source for Harmonic Frequency Scan studies
- Single and 3-phase frequency dependent loads in CIGRÉ format
- Single phase RLC element with frequency dependent parameters

Standard Component..

- Complete list of standard components in alphabetical order sorted by support file names.

4.9.1 Probes & 3-phase



The menu *Probes & 3-phase* appears when the mouse moves over this item in the *Component selection menu* or when the user hits the *P* character.

Probes are components for monitoring the node or branch voltage, branch current or TACS values. In the *Open Probe dialog* you can specify the number of phases to connect to and select phases to be monitored.

Fig. 4.50 - Drawing objects on the Probe & 3-phase menu.

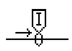
Probe Volt

Selecting this field draws the voltage probe to specify a node voltage-to-ground output request in the ATP-file.

Probe Branch volt.

Selecting this field draws the branch voltage probe to specify a branch voltage output requests in the ATP-file. ATPDraw inserts a 1E+9 ohm resistance.

Probe Curr

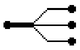


Selecting this field inserts a current probe (measuring switch) into the circuit to specify current output request in column 80 in the ATP-file. The number of monitored phases are user selectable.

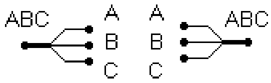
Probe Tacs

Selecting this field draws the Tacs probe to specify signal output and inserts TACS Type-33 object into the ATP-file.

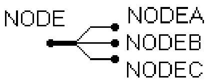
Splitter



The *Splitter* object is a transformation between a 3-phase node and three 1-phase nodes. The object has 0 data and 4 nodes. The object can be moved, rotated, selected, deleted, copied and exported as any other standard components.

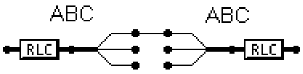


When a splitter is rotated the phase sequence of the single-phase side changes as shown left.

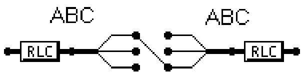


If a name is given to the 3-phase node, the letters *A B C* are added automatically on the single-phase side of splitters.

Note! Do not give names to nodes at the single-phase side of splitters and do not connect splitters together on the single-phase side. I.e. next examples are illegal!



disconnection is illegal this way!



transposition is illegal this way!

Transp 1 ABC-BCA ...Transp 4 ABC-ACB

Transposition objects can be used to change the phase sequence of a 3-phase node. The following transpositions are supported:

 Change the phase sequence from *ABC* to *BCA*.

 Change the phase sequence from *ABC* to *CAB*.

 Change the phase sequence from *ABC* to *CBA*.

 Change the phase sequence from *ABC* to *ACB*.

Handling of transpositions for objects with several 3-phase nodes can be accomplished by specifying a circuit number *Kind* under *Objects | Edit Standard / Nodes* (see in 4.2.5.1). 3-phase nodes having the same *Kind* will receive the same phase sequence.

ABC reference

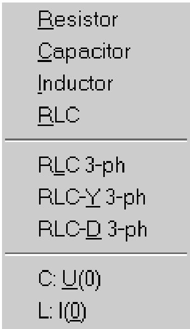
When attached to a 3-phase node in the circuit this node becomes the "master" node with phase sequence *ABC*. The other nodes will adapt this setting.

DEF reference

When attached to a 3-phase node in the circuit this node becomes the "master" node with phase sequence *DEF*. The other nodes will adapt this setting. A combination of *ABC* and *DEF* references is possible for e.g. in 6-phase circuits.

4.9.2 Branch Linear

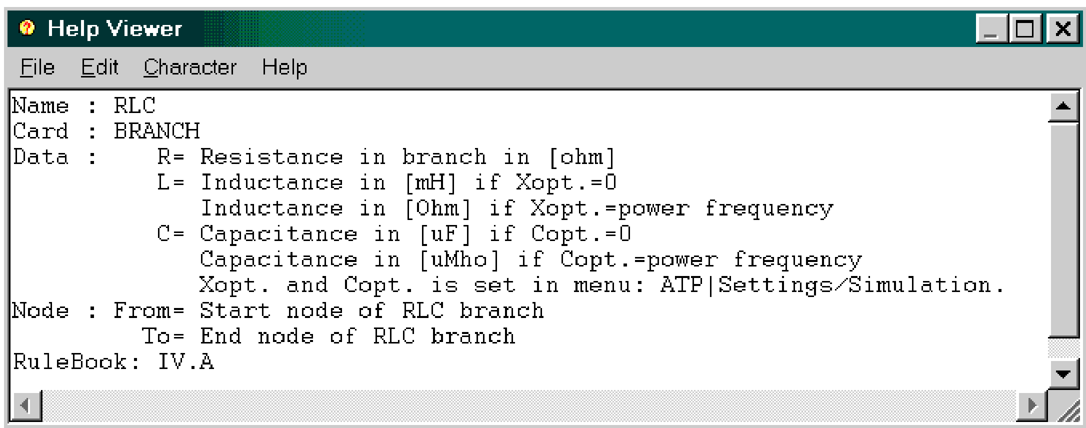
This sub-menu contains linear branch components. The name and the icon of linear branch objects, as well as a brief description of the components are given next in tabulated form. Data parameters and node names to all components can be specified in the *Component* dialog box (see Fig. 4.43), which appears if you click on the icon of the component with the right mouse button in the circuit window.



The *Help* button on the Component dialog boxes calls the *Help Viewer* in which a short description of parameters and a reference to the corresponding ATP Rule Book chapter is given. As an example, Fig. 4.52 shows the help information associated with the ordinary RLC branch.

Fig. 4.51 - Supported linear branch elements.

Selection	Object name	Icon	ATP card	Description
Resistor	RESISTOR		BRANCH type 0	Pure resistance in Ω.
Capacitor	CAP_RS		BRANCH type 0	Capacitor with damping resistor. C in μF if Copt=0.
Inductor	IND_RP		BRANCH type 0	Inductor with damping resistor. Inductance in mH if Xopt=0.
RLC	RLC		BRANCH type 0	R, L and C in series.
RLC 3-ph	RLC3		BRANCH type 0	3-phase R, L and C in series. Independent values in phases.
RLC-Y 3-ph	RLCY3		BRANCH type 0	3-phase R, L and C, Y coupling. Independent values in phases.
RLC-D 3-ph	RLCD3		BRANCH type 0	3-phase R, L and C, D coupling. Independent values in phases.
C : U(0)	CAP_U0		BRANCH + initial condition	Capacitor with initial condition.
L : I(0)	IND_I0		BRANCH + initial condition	Inductor with initial condition.



Help Viewer

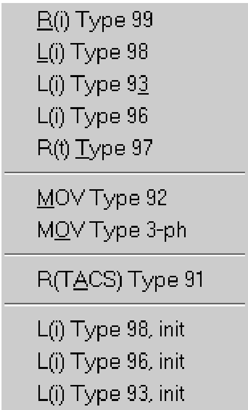
File Edit Character Help

```
Name : RLC
Card : BRANCH
Data :     R= Resistance in branch in [ohm]
           L= Inductance in [mH] if Xopt.=0
              Inductance in [Ohm] if Xopt.=power frequency
           C= Capacitance in [uF] if Copt.=0
              Capacitance in [uMho] if Copt.=power frequency
              Xopt. and Copt. is set in menu: ATP|Settings/Simulation.
Node : From= Start node of RLC branch
         To= End node of RLC branch
RuleBook: IV.A
```

Fig. 4.52 - Help information associated with the series RLC object.

4.9.3 Branch Nonlinear

This menu contains the supported nonlinear resistors and inductors. All the objects except the TACS controlled resistor can also have a nonlinear characteristic. These attributes can be specified by selecting the *Characteristic* tab of the *Component* dialog boxes as shown in Fig. 4.44. The nonlinear characteristic of objects can be entered as piecewise linear interpolation. The number of data points allowed to enter on the current/voltage, current/flux or time/resistance characteristics are specified in the *Help* file of objects.



U/I characteristics of nonlinear resistances are assumed symmetrical, thus (0, 0) point should not be entered. If the saturation curve of a nonlinear inductor is symmetrical start with point (0, 0) and skip the negative points. The hysteresis loop of Type-96 reactors is assumed symmetrical, so only the lower loop of the hysteresis must be entered. The last point should be where the upper and lower curves meet in the first quadrant. If you specify a metal oxide arrester with MOV Type-92 component, ATPDraw accepts the current/voltage characteristic and performs an exponential fitting in the log-log domain to produce the required ATP data format.

Fig. 4.53 - Nonlinear branch elements.

Selection	Object name	Icon	ATP card	Description
R(i) Type 99	NLINRES		BRANCH type 99	Current dependent resistance.
L(i) Type 98	NLININD		BRANCH type 98	Current dependent inductor.
L(i) Type 93	NLIND93		BRANCH type 93	True non-linear current dependent inductor.
L(i) Type 96	NLIND96		BRANCH type 96	Pseudo-nonlinear hysteretic inductor.
R(t) Type 97	NLINR_T		BRANCH type 97	Time dependent resistor.
MOV Type 92	MOV		BRANCH type 92	Current dependent resistance on exponential form.
MOV Type 3-ph	MOV_3		BRANCH type 92	3-phase current dependent resistance.
R(TACS) Type 91	TACSRES		BRANCH type 91	TACS / MODELS controlled time dependent resistor.
L(i) Type 98, init	NLIN98_I		BRANCH type 98	Current-dependent inductor. with initial flux.
L(i) Type 96, init	NLIN96_I		BRANCH type 96	Pseudo-nonlinear hysteretic inductor with initial flux.
L(i) Type 93, init	NLIN93_I		BRANCH type 93	True non-linear inductor with initial flux.

4.9.4 Lines/Cables

The *Lines/Cables* menu has several sub-menus for different types of line models. Available line models are: Lumped parameter models (RLC π, RL coupled), distributed parameter lines with constant (i.e. frequency independent) parameters, lines and cables with constant or frequency dependent parameters (Bergeron, PI, JMarti, Noda or Semlyen), calculated by means of the LINE CONSTANTS, CABLE CONSTANTS or CABLE PARAMETERS supporting routine of ATP-EMTP.

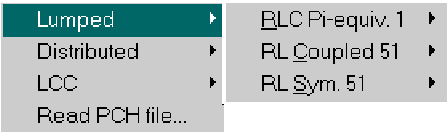


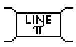
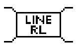
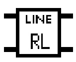
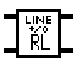
Fig. 4.54 - Line models with lumped parameters.

4.9.4.1 Lumped parameter line models

RLC Pi-equiv. 1: These line models are simple, lumped, non-symmetric π-equivalents of ATP Type 1, 2, 3 etc. branches of ATP.

RL Coupled 51: These line models are simple, lumped, non-symmetric mutually RL coupled components of Type-51, 52, 53 etc. branches of ATP.

RLC Sym. 51: These line models are symmetric with sequence value input. The line models are special applications of the RL coupled line models in ATP. The following selections are available on the three pop-up menus:

Selection	Object name	Icon	ATP card	Description
RLC Pi-equiv. 1 +1 phase	LINEPI_1		BRANCH type 1	Single phase RLC π-equivalent.
RLC Pi-equiv. 1 + 2 phase	LINEPI_2		BRANCH type 1-2	2-phase RLC π-equivalent Non-symmetric.
RLC Pi-equiv. 1 + 3 phase	LINEPI_3		BRANCH type 1-3	3-phase RLC π-equivalent Non-symmetric. 3-phase nodes.
RL Coupled 51 + 2 phase	LINERL_2		BRANCH type 51-52	2-phase RL coupled line model. Non-symmetric.
RL Coupled 51.. + 3 phase	LINERL_3		BRANCH type 51-53	3-phase RL coupled line model. Non-symmetric. 3-phase nodes.
RL Coupled 51.. + 6 phase	LINERL_6		BRANCH type 51-56	2x3 phase RL coupled line model. Non-symmetric. Off-diagonal R is set to zero.
RL Sym. 51 + 3 ph	LINESY_3		BRANCH type 51-53	3-phase RL coupled line model with sequence impedance (0, +) input. Symmetric.
RL Sym. 51 + 6 ph	LINESY_6		BRANCH type 51-56	2x3-phase RL coupled line model with sequence impedance (0, +) input. Symmetric.

4.9.4.2 Distributed parameter line models

Selecting *Distributed* opens a popup menu where two different types of line models can be selected: *Transposed lines* or *Untransposed lines*. Both of that are distributed parameters, frequency independent lines.

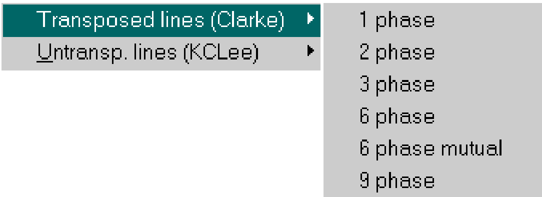


Fig. 4.55 - Distributed transmission line models.

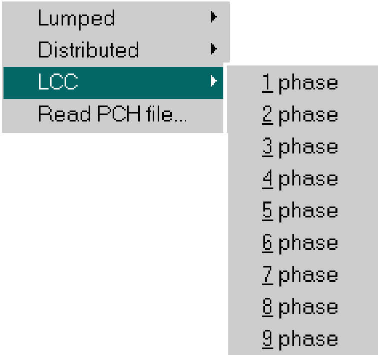
Transposed lines (Clarke): These components can be characterized as symmetrical, distributed parameter and lumped resistance models (called as Clarke-type in the ATP Rule-Book). Six different types are supported:

Selection	Object name	Icon	ATP card	Description
Transposed lines + 1 phase	LINEZT_1		BRANCH type -1	Single phase, distributed parameter line, Clarke model.
Transposed lines + 2 phase	LINEZT_2		BRANCH type -1.. -2	2-phase, distributed parameter, transposed line, Clarke model.
Transposed lines + 3 phase	LINEZT_3		BRANCH type -1.. -3	3-phase, distributed parameter, transposed line, Clarke model.
Transposed lines + 6 phase	LINEZT6N		BRANCH type -1.. -6	6-phase, distributed parameter, transposed line, Clarke model.
Transposed lines + 6 phase mutual	LINEZT_6		BRANCH type -1.. -6	2x3 phase, distributed Clarke line. with mutual coupling between the circuits.
Transposed lines + 9 phase	LINEZT_9		BRANCH type -1.. -9	9-phase, distributed parameter, transposed line, Clarke model.

Untransposed lines (KCLee): Parameters of these nonsymmetrical lines are usually generated outside ATPDraw. These components can be characterized as untransposed, distributed parameter and lumped resistance models with real or complex modal transformation matrix (called as KCLee-type in the ATP Rule-Book). Double-phase and 3-phase types are supported:

Selection	Object name	Icon	ATP card	Description
Untransposed lines (KCLee)+ 2 phase	LINEZU_2		BRANCH	2-phase, distributed parameters, untransposed (KCLee) line model with complex transformation matrix.
Untransposed lines (KCLee)+ 3 phase	LINEZU_3		BRANCH	3-phase, distributed parameters, untransposed (KCLee) line model with complex transformation matrix

4.9.4.3 LCC objects



In this part of the program, you specify the geometrical and material data for an overhead line or a cable and the corresponding electrical data are calculated automatically by the LINE CONSTANTS, CABLE CONSTANTS or CABLE PARAMETERS supporting routine of ATP-EMTP.

Fig. 4.56 - LCC supports line modeling up to 9 phases.

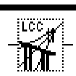
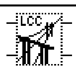
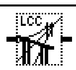
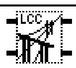
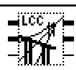
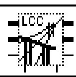
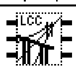
To use the *LCC* module of ATPDraw the user must first select a line/cable component with the desired number of phases (1..9). This will display an object in the circuit window that can be connected to the circuit as any other component. Clicking on this component with the right mouse button will bring up a special input dialog box called *Line/Cable Data* dialog box with two sub-pages: *Model* and *Data*, where the user selects between the supported *System type*:

- Overhead Line: LINE CONSTANTS
- Single Core Cables: CABLE PARAMETERS or CABLE CONSTANTS
- Enclosing Pipe: CABLE PARAMETERS or CABLE CONSTANTS

and *Model type* of the line/cable :

- Bergeron: Constant parameter KCLee or Clark models
- PI: Nominal PI-equivalent (short lines)
- JMarti: Frequency dependent model with constant transformation matrix
- Noda: Frequency dependent model
- Semlyen: Frequency dependent simple fitted model.

The *Line/Cable Data* dialog box completely differs from the *Component* dialog box of other components, therefore it is described in chapter 5.3 of the Advanced Manual.

Selection	Object name	Icon	ATP card	Description
LCC + 1 phase	`Line/Cable`		$Include	Single phase LCC object.
LCC + 2 phase	`Line/Cable`		$Include	2-phase LCC object.
LCC + 3 phase	`Line/Cable`		$Include	3-phase LCC object.
...	...	...	...	...
LCC + 6 phase	`Line/Cable`		$Include	2x3-phase LCC object.
LCC + 7 phase	`Line/Cable`		$Include	7-phase LCC object.
LCC + 8 phase	`Line/Cable`		$Include	8-phase LCC object.
LCC + 9 phase	`Line/Cable`		$Include	3x3-phase LCC object.

4.9.4.4 Read PCH file...

ATPDraw is able to read the `.pch` output files obtained by external run of ATP-EMTP's LINE CONSTANTS or CABLE CONTSTANTS supporting routines. Selecting the *Read PCH file...* menu item, the program performs an *Open Punch File* dialog in which the available `.pch` files are listed. If you select a file and click *Open,* ATPDraw attempts to read the file and if succeed in creates a `.lib` file in the `\USP` folder in the Data Base Module format of ATP. When the `.lib` file is successfully created, the icon of the new LCC component appears in the middle of the circuit window.

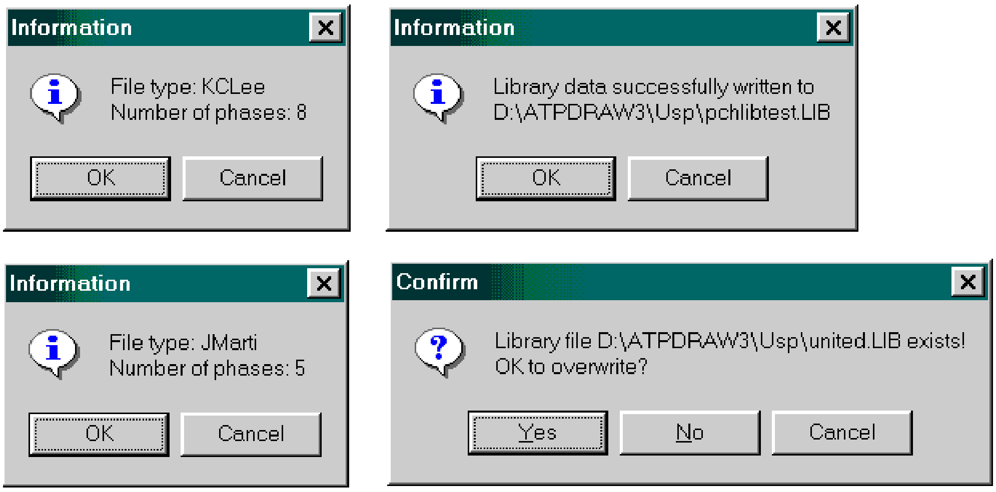
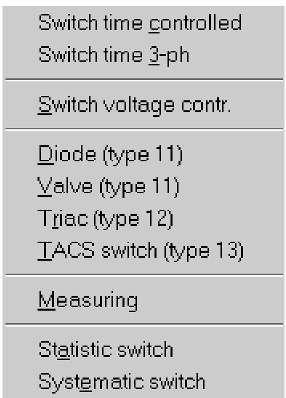


Fig. 4.57 - Results of pch-file reading are communicated in dialog boxes.

4.9.5 Switches



ATPDraw supports most of the switch type elements in ATP, such as ordinary time- or voltage-controlled switches, options for modeling diodes, valves and triacs, as well as measuring and statistical switches.

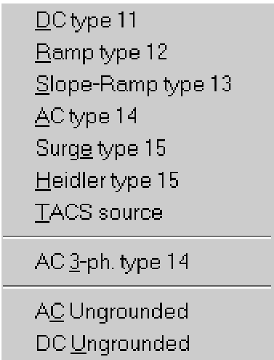
The *Switches* sub-menu contains the following switch objects:

Fig. 4.58 - Supported switch type ATP components.

Selection	Object name	Icon	ATP card	Description
Switch time controlled	SWITCHTC		SWITCH type 0	Single phase time controlled switch.
Switch time 3-ph	SWIT_3XT		SWITCH type 0	Three-phase time controlled switch, Independent operation of phases.
Switch voltage contr.	SWITCHVC		SWITCH type 0	Voltage controlled switch.
Diode (type 11)	DIODE		SWITCH type 11	Diode. Switch type 11. Uncontrolled.

Selection	Object name	Icon	ATP card	Description
Valve (type 11)	VALVE		SWITCH type 11	Valve/Thyristor. Switch type 11. TACS/MODELS- controlled.
Triac (type 12)	TRIAC		SWITCH type 12	Double TACS/MODELS controlled switch.
TACS switch (type 13)	TACSSWIT		SWITCH type 13	Simple TACS/MODELS controlled switch.
Measuring	SWMEAS		SWITCH type 0	Measuring switch. Current measurements.
Statistic switch	SW_STAT		SWITCH	Statistic switch. See *ATP* | *Settings / Switch/UM*.
Systematic switch	SW_SYST		SWITCH	Systematic switch. See *ATP* | *Settings / Switch/UM*.

4.9.6 Sources

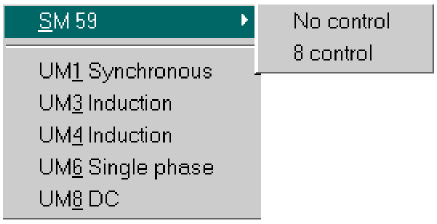


The popup menu under *Sources* contains the following items:

Fig. 4.59 - Electrical sources in ATPDraw.

Selection	Object name	Icon	ATP card	Description
DC type 11	DC1PH		SOURCE type 11	DC step source. Current or voltage.
Ramp type 12	RAMP		SOURCE	Ramp source. Current or voltage.
Slope-Ramp type 13	SLOPE_RA		SOURCE type 13	Two-slope ramp source. Current or voltage.
AC type 14	AC1PH		SOURCE	AC source. Current or voltage.
Surge type 15	SURGE		SOURCE	Double exponential source Type-15.
Heidler type 15	HEIDLER		SOURCE	Heidler type source. Current or Volt.
TACS source	TACSSOUR		SOURCE type 60	TACS/MODELS controlled source. Current or voltage.
AC 3-ph. type 14	AC3PH		SOURCE type 14	AC source. Current or voltage. 3-phase node.
AC Ungrounded	AC1PHUG		SOURCE type 14+18	Ungrounded AC source. Voltage only.
DC Ungrounded	DC1PHUG		SOURCE type 11+18	Ungrounded DC source. Voltage only.

4.9.7 Machines



Two categories of electrical machines are available in ATPDraw: *Synchronous Machines* and *Universal Machines*. ATPDraw does not support machines in parallel or back-to - back.

Fig. 4.60 - Supported electric machine alternatives.

The *Synchronous Machine* models in ATPDraw have the following features/limitations:

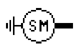
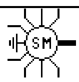
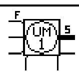
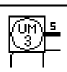
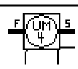
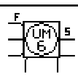
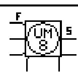
- With and without TACS control.
- Manufacturers data.
- No saturation.
- No eddy-current or damping coils.
- Single mass.

The *Universal Machine* models in ATPDraw have the following features/limitations:

- Manual and automatic initialization.
- SM, IM and DC type supported.
- Raw coil data (internal parameters).
- Saturation is supported in d, q, or both axes.
- One single excitation coil in each axis (d, q).
- Network option for mechanical torque only.
- Single torque source.

The *Component* dialog box of *Universal Machines* is significantly different than that of the other objects. A complete description of parameters in this dialog box is given in chapter 5.7.2 of the Advanced Manual.

The popup menu under *Machines* contains the following items:

Selection	Object name	Icon	ATP card	Description
SM 59 + No control	SM59_NC		MACHINE type 59	Synchronous machine. No TACS control. 3-phase armature.
SM 59 + 8 control	SM59_FC		MACHINE type 59	Synchronous machine. Max. 8 TACS control. 3-phase armature.
UM1 Synchronous	UM_1		UM-MACHINE type 1	Synchrounous. Set initialization under *ATP* | *Settings/Switch/UM*.
UM3 Induction	UM_3		UM-MACHINE type 3	Induction. Set initialization under *ATP* | *Settings/Switch/UM*.
UM4 Induction	UM_4		UM-MACHINE type 4	Induction. Set initialization under *ATP* | *Settings/Switch/UM*.
UM6 Single phase	UM_6		UM-MACHINE type 6	Single phase. Set initialization under *ATP* | *Settings/Switch/UM*.
UM8 DC	UM_8		UM-MACHINE type 8	DC machine. Set initialization under *ATP* | *Settings/Switch/UM*.

4.9.8 Transformers

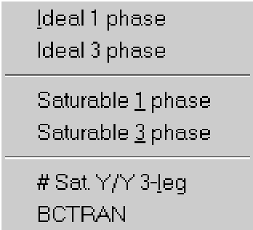


Fig. 4.61 - Transformer models in ATPDraw.

The popup menu under *Transformers* contains the following items:

Selection	Object name	Icon	ATP card	Description
Ideal 1 phase	`TRAFO_I`		SOURCE type 18	Single-phase ideal transformer.
Ideal 3 phase	`TRAFO_I3`		SOURCE type 18	3-phase ideal transformer.
Saturable 1 phase	`TRAFO_S`		BRANCH TRANSFORMER	Single-phase saturable transformer.
Saturable 3 phase	`GENTRAFO`		BRANCH TRANSFORMER	General saturable transformer. 3-phase. 2 or 3 windings.
# Sat. Y/Y 3-leg	`TRAYYH_3`		BRANCH TRANSFORMER THREE PHASE	3-phase saturable transformer. High homopolar reluct. (3-leg). 3-ph node. Preprocessing of manufacturer data.
BCTRAN	`BCTRAN`		BRANCH Type 1...9	Direct support of BCTRAN transformer matrix modeling.

The characteristic of the nonlinear magnetizing branch of the three saturable-type transformers can be given in the *Characteristic* tab of the component dialog box. The saturable transformers have an input window like the one in Fig. 4.44. In this window the magnetizing branch can be entered in I_{RMS}/U_{RMS} or $I_A/FLUX_{Vs}$ coordinates. The *RMS* flag on the *Attributes* page select between the two input formats. If the *Include characteristic* check box is selected on the *Attributes* page, a disk file referenced in the *$Include* field will be used in the ATP input file. If the nonlinear characteristic is given in I_{RMS}/U_{RMS}, ATPDraw will calculate the flux/current values automatically and use them in the final ATP input file.

The BCTRAN transformer component provides direct support of BCTRAN transformer matrix modeling. The user is requested to specify input data (open circuit and short circuit factory test data) in BCTRAN supporting routine format, then ATPDraw performs an ATP run to generate a punch-file that is inserted into the final ATP-file describing the circuit. The user can specify where the factory test was performed and where to connect the excitation branch. The excitation branch can be linear or non-linear. In the latter case, the nonlinear inductors must be connected to the winding closest to the iron core as external elements.

The *BCTRAN* dialog and the *Component* dialog box of the *Saturable 3-phase* `GENTRAFO` differ in many ways from the input data window of other objects. A more comprehensive description of the input parameters is given in chapters 5.6 and 5.7.1 of the Advanced Manual, respectively.

4.9.9 MODELS

Besides the standard components, the user can create his/her own models using the MODELS simulation language in ATP [4]. ATPDraw supports only a simplified usage of MODELS. The user writes a model-file and ATPDraw takes care of the INPUT/OUTPUT section of MODELS along with the USE of each model. The following restriction applies:

- Only `INPUT`, `OUTPUT` and `DATA` supported in the `USE` statement.
- Not possible to call other models under `USE`
- Exchanging data between models is restricted (see remarks in 4.2.5.4)
- Not possible to specify `HISTORY` of `DELAY CELLS` under `USE`

Using this feature requires knowledge about the syntax and general structure of MODELS language. There are two options for creating a model object in ATPDraw:

- to create a support file manually under *Object* | *Model* | *New sup-file*.
- to select a `.mod` file directly under *MODELS* in the selection menu.

The Advanced part of this Manual gives detailed information about both procedures and a general overview about the use of MODELS in ATPDraw. In this chapter only the automatic support file generation is introduced. The process normally consists of two steps:

1. To create a model file (`.mod`) containing the actual model description.
2. To load this file via the *Files (sup/mod)...* or *Type 94* sub-menus under *MODELS*

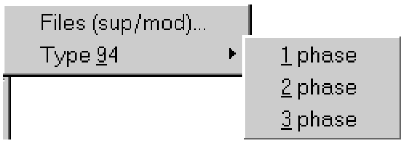


Fig. 4.62 - Options under the MODELS sub-menu.

Files (sup/mod)...

Selecting *MODELS* | *Files (sup/mod)...* in the component selection menu performs an *Open Model* dialog box where the user can choose a model file name or a support file name. These files are normally stored under the `/MOD` folder. If a `.mod` file was selected ATPDraw interprets the file as shown in Fig. 4.63 and creates a support file (`.sup`) for it. If the support file already exists, i.e. it has been created manually by the user under the *Object* | *Model* | *New sup-file* or by ATPDraw earlier, the new model object appears immediately in the circuit window, i.e. the *Information* dialog does not show up.

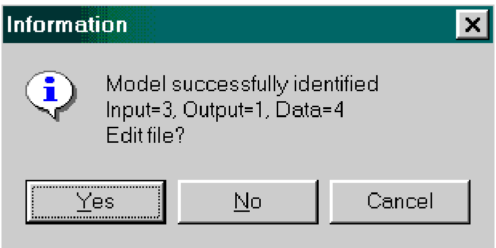


Fig. 4.63 - Interpretation of the model.

The *Component* dialog box of model objects has a new input section *Models* below the *DATA* and *NODES* attributes as shown in Fig. 4.64. This new section has two input fields: *Model file* for locating the model description file and a *Use As* field for specification of the *model_name* in the `USE model AS model_name` statement of MODELS.

The input/output to MODELS, the use of the model and interfacing it with the rest of the circuit are handled by ATPDraw, automatically. Model descriptions are written directly in the ATP file instead of using $Include. Blank lines are removed when inserting the model file in the ATP-file. The general structure of the MODELS section in the `.atp` input file is shown below:

```
MODELS
/MODELS
INPUT
  IX0001 {v(U1_ZNO)}
  IX0002 {v(U2_ZNO)}
  IX0003 {v(I_ZNO )}
OUTPUT
  TRIP_A
MODEL FLASH_1
------------------------------------------
Description of the model. A .mod file e.g.
D:\ATPDraw3\Mod\Flash_1.mod is pasted here
------------------------------------------
ENDMODEL
USE FLASH_1 AS FLASH_1
INPUT
  V1:= IX0001
  V2:= IX0002
  iczn:= IX0003
DATA
  Pset:=      1.
  Eset:=     10.
  fdel:=      5.
  fdur:=     20.
OUTPUT
  TRIP_A:=trip
ENDUSE
ENDMODELS
```

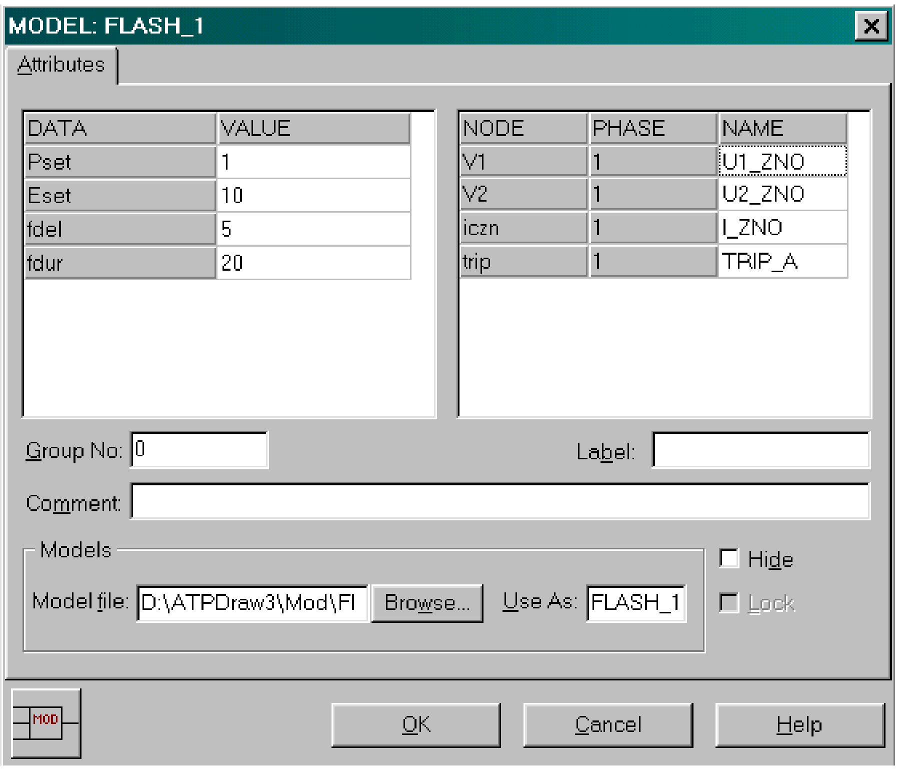


Fig. 4.64 - The component dialog box of model object FLASH_1.

Type 94 + 1 phase ... 3 phase

Selecting *MODELS* | *Type 941+ 1...3 phase* in the component selection menu performs an *Open Model* dialog box where the user can choose a Type-94 compatible `.mod` file. These files are normally stored under the `/MOD` folder. Depending on the selected *phase*, ATPDraw associates the corresponding Type-94 `.sup` file with the model then interprets the number of input data as shown in Fig. 4.65.

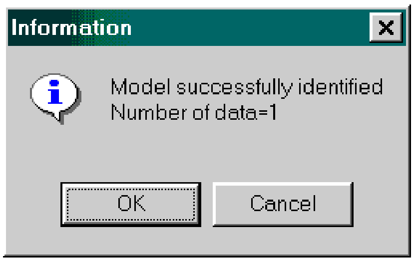


Fig. 4.65 - Interpretation of the Type-94 model file.

The *Component* dialog box of Type-94 model objects has a new input section *Type 94* besides the *DATA* and *NODES* attributes (see on Fig. 4.66). This new section has three input fields: *THEV, ITER* and *NORT* to specify the solution method for ATP when interfacing the Type-94 object with the rest of the electrical network, and a *Use As* field for specification of the name of the model referenced in the Type-94 declaration of the ATP input file.

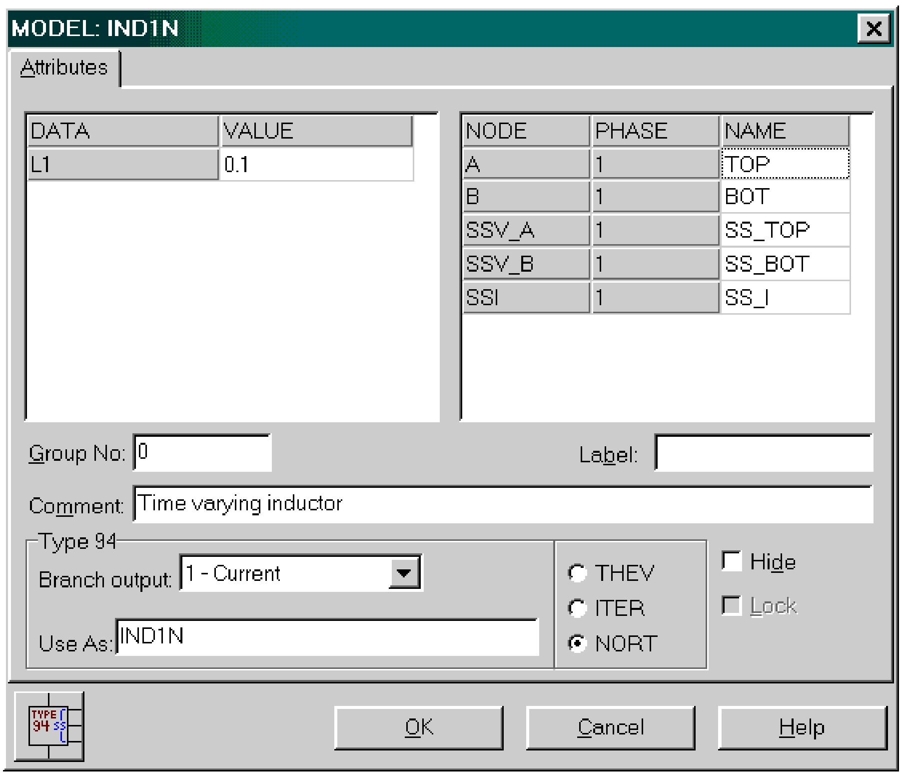


Fig. 4.66 - Component dialog box of Type-94 model objects.

Signal input and data values for a Type-94 object are loaded by ATP and the output of the object are also used automatically by ATP. Interfacing it with other components of the circuit is handled by ATPDraw. A Type-94 compatible `.mod` files must have a fixed structure and the use of such an object also requires special declarations in the ATP input file as shown next:
Structure of a Type-94 compatible `.mod` file:

```
MODEL ind1n
 comment ---------------------------------------------------------------
 |                                                         _______       |
 | Internal circuit:  1-ground : L1            |       |       |
 |                                           1 o       |       |
 |                                             |_______|       |
 |                                               _|_           |
 |                                                -  ground    |
 |                                                             |
 | Built for use as a 1-phase non-transmission type-94 Norton component  |
  ------------------------------------------------------------- endcomment
 comment ------------------------------------------------------------------
 | First, declarations required for any type-94 iterated model            |
 |   (the values of these data and input are loaded automatically by ATP)  |
 |   (the values of these outputs are used automatically by ATP)          |
 |   (DO NOT MODIFY THE SEQUENCING OF THE DATA, INPUT, AND VAR IN THIS GROUP)  |
 |   (the names may be modified, except 'n')                              |
 |   (when built for n=1, the array notation is not required)             |
  ------------------------------------------------------------------- endcomment
 DATA  n                     -- number of phases
       ng {dflt: n*(n+1)/2} -- number of conductances
 INPUT v          -- voltage(t) at terminal 1
       v0         -- voltage(t=0) at terminal 1
       i0         -- current(t=0) into terminal 1
 VAR   i          -- current(t) into terminal 1
       is         -- Norton source(t+timestep) at terminal 1
       g          -- conductance(t+timestep) at terminal 1
       flag       -- set to 1 whenever a conductance value is modified
 OUTPUT i, is, g, flag
 comment ------------------------------------------------------------------
 | Next, declarations of user-defined data for this particular model      |
 | (values which must be defined when using this model as a type-94 component) |
  ------------------------------------------------------------------- endcomment
 DATA  L1      -- [H] reference value of inductance L
 comment ------------------------------------------------------------------
 | Next, declarations private to the operation of this model              |
  ------------------------------------------------------------------- endcomment
 VAR   st     -- used for converting Laplace s to time domain
       L      -- [H] variable value of inductance L
 INIT
  st    := 2/timestep  -- trapezoidal rule conversion from Laplace
  L := L1              -- initialize variable inductance value
  g:= 1/(st*L)         -- conductance converted from Laplace 1/sL
 ENDINIT
 EXEC   -- L is constant in this example
  IF t=0 THEN
   flag := 1           -- conductance values have been changed
   i    := i0          -- t=0 current through L
   is   := -i0 -g*v0   -- history term for next step
  ELSE
   flag := 0           -- reset flag
   i    := g*v -is     -- applying trapezoidal rule, calculate from v(t)
   is   := -i -g*v     -- history term from trapezoidal rule, for next step
  ENDIF
 ENDEXEC
ENDMODEL
```

The use of a Type-94 Norton model in the ATPDraw generated input file is shown next. The Type-94 declaration refers to the *Use As* name of the MODEL in which the operation of the object is described:

```
C Time varying inductor
94TOP   BOT    IND1N NORT                                                  1
 >DATA  L1           0.1
 >SSV   SS_TOP
 >SSV   SS_BOT
 >SSI   SS_I
 >END
```

4.9.10 TACS

The TACS menu gives access to most type of TACS components of ATP. Exceptions are Device 55 and multi-line Fortran statements ($ continuation), which are not supported. The TACS sub-menu on the component selection menu contains the following items:

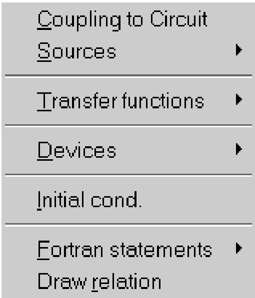


Fig. 4.67 - Supported TACS objects.

4.9.10.1 Coupling to circuit

The *Coupling to circuit* object provides an interface for TACS HYBRID simulations. This object must be connected with an electrical node to pass node voltages, or the branch currents / switch status to TACS. The type of the variable sent to TACS is controlled by the *Type* settings in the EMTP_OUT component dialog box. Users are warned that only single-phase electrical variables can be interfaced with TACS input nodes, this way. In case of 3-phase modeling, a splitter object is also required, and the coupling to circuit object must be connected at the single-phase side of the splitter as shown in Fig. 4.68.

Selection	Object name	Icon	ATP card	Description
Coupling to Circuit	EMTP_OUT		TACS type 90-93	Value from the electrical circuit into TACS. 90 - Node voltage 91 - Switch current 92 - internal variable special EMTP comp. 93 - Switch status.

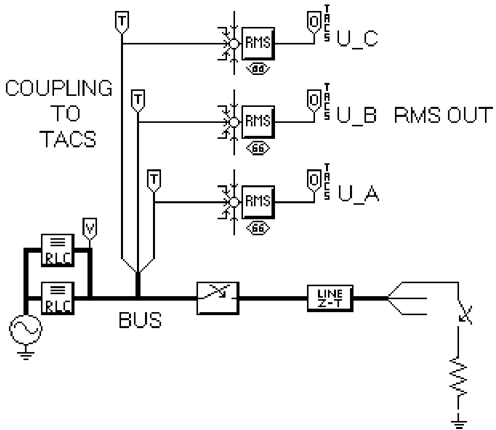
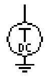
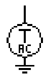
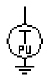
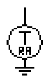


Fig. 4.68 - Coupling a 3-phase electrical node to TACS.

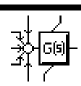
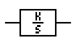
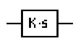
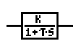
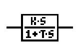
4.9.10.2 TACS sources

The *Sources* of TACS menu contains the following items:

Selection	Object name	Icon	ATP card	Description
DC - 11	DC_01		TACS type 11	TACS step signal source.
AC - 14	AC_02		TACS type 14	TACS AC cosine signal source.
Pulse - 23	PULSE_03		TACS type 23	TACS pulse train signal.
Ramp - 24	RAMP_04		TACS type 24	TACS saw-tooth train signal.

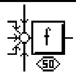
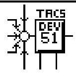
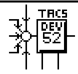
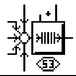
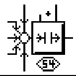
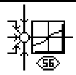
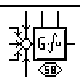
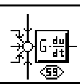
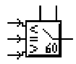
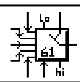
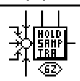
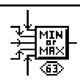
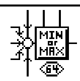
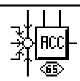
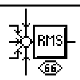
4.9.10.3 TACS transfer functions

All the older TACS transfer functions of previous ATPDraw versions are supported in version 3, but some of them has been removed from the component selection menu and replaced by a more general component: the *General transfer function*. This object defines a transfer function in the *s* domain and it can be specified with or without limits. Four more simple transfer functions are also supported: *Integral*, *Derivative*, first order *High* and *Low pass* filters.

Selection	Object name	Icon	ATP card	Description
General	TRANSF		TACS	General transfer function in *s* domain. Order 0-7. Named dynamic limits.
Integral	INTEGRAL		TACS	Integral of the input multiplied by K.
Derivative	DERIV		TACS	Simple derivative transfer function.
Low pass	LO_PASS		TACS	First order low pass filter.
High pass	HI_PASS		TACS	First order high pass filter.

4.9.10.4 TACS devices

The following TACS *Devices* are supported in ATPDraw:

Selection	Object name	Icon	ATP card	Description
Freq sensor - 50	DEVICE50		TACS type 88,98 or 99	Frequency sensor.
Relay switch - 51	DEVICE51		TACS type 88,98 or 99	Relay-operated switch.
Level switch - 52	DEVICE52		TACS type 88,98 or 99	Level-triggered switch.
Trans delay - 53	DEVICE53		TACS type 88,98 or 99	Transport delay.
Pulse delay - 54	DEVICE54		TACS type 88,98 or 99	Pulse delay.
User def nonlin - 56	DEVICE56		TACS type 88,98 or 99	Point-by-point non-linearity.
Multi switch - 57	DEVICE57		TACS	Multiple open/close switch.
Cont integ - 58	DEVICE58		TACS type 88,98 or 99	Controlled integrator.
Simple deriv - 59	DEVICE59		TACS type 88,98 or 99	Simple derivative.
Input IF - 60	DEVICE60		TACS type 88,98 or 99	Input-IF component.
Signal select - 61	DEVICE61		TACS type 88,98 or 99	Signal selector.
Sample_track - 62	DEVICE62		TACS type 88,98 or 99	Sample and track.
Inst min/max - 63	DEVICE63		TACS type 88,98 or 99	Instantaneous minimum/maximum.
Min/max track - 64	DEVICE64		TACS type 88,98 or 99	Minimum/maximum tracking.
Acc count - 65	DEVICE65		TACS type 88,98 or 99	Accumulator and counter.
Rms meter - 66	DEVICE66		TACS type 88,98 or 99	RMS value of the sum of input signals.

4.9.10.5 Initial conditions

The initial condition of a TACS variable can be specified by selecting TACS object (type 77) under the *TACS* | *Initial cond.* menu. The name of this component is `INIT_T` and its icon is .

4.9.10.6 Fortran statements

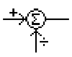
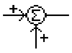
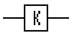
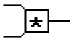
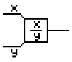
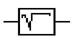
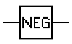
The component dialog box of the *Fortran statements* | *General* object provides a *Type* field where the user is allowed to specify the type of the object (input, output, inside) and an *OUT* field for the single line Fortran-like expression. These statements are written into the `/TACS` subsection of the ATP input file starting at column 12.

The *Fortran statements* | *Math* and *Logic* sub-menus include additional simple objects for the basic mathematical and logical operations.

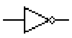
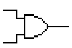
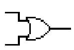
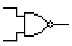
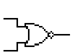
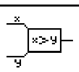
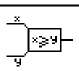
General

Selection	Object name	Icon	ATP card	Description
General	FORTRAN1		TACS type 88,98 or 99	User specified FORTRAN expression.

Fortran statements / Math

Selection	Object name	Icon	ATP card	Description
x - y	DIFF2		TACS type 98	Subtraction of two input signals.
x + y	SUM2		TACS type 98	Addition of two input signals.
*x * K*	MULTK		TACS type 98	Multiplication by a factor of *K*.
*x * y*	MULT2		TACS type 98	Multiplication of x by y.
x / y	DIV2		TACS type 98	Ratio between two input signals.
sqrt (x)	SQRT		TACS type 98	Square root of the input signal.
| *x* |	ABS		TACS type 98	Absolute value of the input signal.
- *x*	NEG		TACS type 98	Change the sign of the input signal.

Fortran statements / Logic

Selection	Object name	Icon	ATP card	Description
NOT	NOT		TACS type 98	Logical operator. OUT = NOT IN.
AND	AND		TACS type 98	Logical operator. OUT = IN_1 AND IN_2.
OR	OR		TACS type 98	Logical operator. OUT = IN_1 OR IN_2.
NAND	NAND		TACS type 98	Logical operator. OUT = IN_1 NAND IN_2.
NOR	NOR		TACS type 98	Logical operator. OUT = IN_1 NOR IN_2.
>	GT		TACS type 98	Logical operator. Output = 1 if x > y, 0 otherwise.
>=	GE		TACS type 98	Logical operator. Output = 1 if x >= y, 0 otherwise.

4.9.10.7 Draw relation

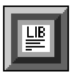
When you select *TACS* | *Draw relation,* the mouse cursor will change to a pointing hand and the program is waiting for a left mouse click on a circuit node to set the starting point of a new relation. You can then draw multiple relations until you click the right mouse button or press the *Esc* key. Relations are used to visualize information flow into Fortran statements. These objects are drawn as blue connections, but have no influence on the component connectivity. You can work with relations exactly the same way as with connections: relations can be selected, rotated, deleted, or moved to another position.

4.9.11 User Specified

Selecting the *Library* item will draw the predefined user specified object `LIB`. This object has no input data and cannot be connected with other objects because it has no input or output nodes.

Fig. 4.69 - Supported user specified objects.

Library



Using this object will result in a $Include statement in the ATP-file. The *User specified* field at the bottom of the component dialog box specifies the name of the file (and path if *$Prefix* misc. request is unselected under *ATP* | *Settings / Format*) that is included by ATP at run time. The user must keep track of internal node names in this file, and if an internal node is connected with other nodes of the circuit, the two node must be given the same name.

Ref. 1-ph

Selecting *Ref. 1-ph* will draw the object `LIBREF_1`. This object has zero parameters and two nodes. Reference objects are not represented in the ATP input data file, but serve only as visualization of connectivity.

Ref. 3-ph

Selecting *Ref. 3-ph* will draw the object `LIBREF_3`. This object has zero parameters and two nodes. Reference objects are not represented in the ATP input data file, but serve only as visualization of connectivity.

Files...

Besides the standard components, the user is allowed to create *User Specified* components. The usage of this feature requires knowledge about ATP's DATA BASE MODULARIZATION technique. The procedure that is described in the Advanced part of this Manual consists of two steps:

1. Creating a new support file (`.sup`) using the *Objects* | *User Specified / New sup-file* menu.
2. Creating a Data Base Module file (`.LIB`), which describes the object.

Selecting *Files...* in the component selection menu executes the *Open Component* dialog and the existing support files in the `\USP` directory are listed. If you select a `.sup` file from the list and click on the *Open* button, the icon of the object will appear in the middle of the active circuit window. Henceforth the user specified objects operate similarly than standard objects.

4.9.12 Frequency comp.

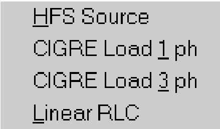
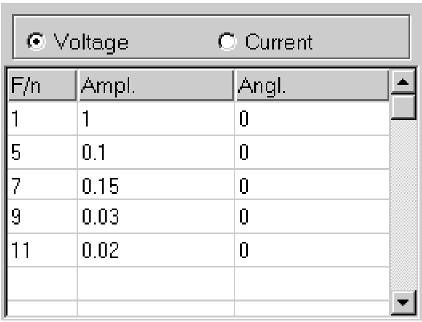


Fig. 4.70 - Supported HFS components.

The Harmonic Frequency Scan (HFS) is one of the options under *ATP | Settings / Simulation*. The *Frequency comp.* menu supports four HFS objects for such a simulation:

Selection	Object name	Icon	ATP card	Description
HFS Source	HFS_SOUR		SOURCE type 14	Harmonic frequency source
Cigre load 1 ph	CIGRE_1		BRANCH type 0	Single-phase CIGRE load
Cigre load 3 ph	CIGRE_3		BRANCH type 0	3-phase CIGRE load
Linear RLC	RLC_F		BRANCH type 0	Linear RLC for HFS studies



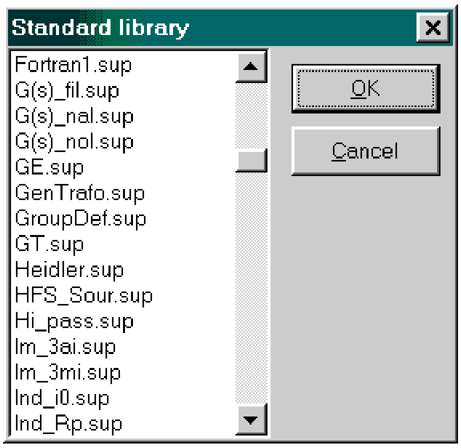
Selecting HFS under *ATP | Settings / Simulation* will run the ATP data case so many times as specified in the *Harmonic source* component dialog box. The frequency of the harmonic source will for each ATP run be incremented. In the example shown at left, 5 harmonic components are specified in the *F/n* column, and the ATP data case will run 5 times.

Fig. 4.71 - Specification of harmonic source frequencies.

In the first run the source frequency will be 1x50 Hz, the second run 5x50 Hz etc. up to the fifth run f = 11x50 Hz = 550 Hz. The *Freq.* value specified by the user under *ATP | Settings / Simulation* is used here as base frequency. The source frequency can also be specified directly in Hz and in such case the first *F/n* must be greater or equal to the Power Frequency. Specifying the frequencies F/n like 50, 250, 350, 450, and 550 would be equivalent to what is shown in Fig. 4.71.

4.9.13 Standard Component...

In ATPDraw 3.x the standard component support files are stored in a single file called `ATPDraw.scl`. The *Standard library* dialog is the container of supported circuit objects in alphabetical order. Any component can be selected from this list, then the object's icon appears in the circuit window exactly the same way as after other selections in the component selection sub-menus.



Support files of the present and even all retired objects (which once were supported in earlier program versions, but have been removed from the component selection menu) are included in the standard library. An old circuit file may of course contain such an older object, which are also supported internally in ATPDraw and the program will produce correct output.

Fig. 4.72 - Standard component library.

SINTEF

5. Advanced Manual . . .

. . . .

ATPDraw™
for Windows
3.5

This chapter gives an overview of several new features in ATPDraw: Grouping and $Parameter, usage of the integrated LINE/CABLE CONSTANTS, BCTRAN and the UNIVERSAL MACHINE support, and describes how to use MODELS in ATPDraw and how to create new user specified object by means of ATP's $Include and DATA BASE MODULARIZATION features. You will not be shown how to create the example circuits, but these project files (`Exa_*.adp`) are part of the ATPDraw distribution. To load these example circuits into ATPDraw, use the *File* | *Open* command (or *Ctrl* + *O*) and select the file name in the *Open Project* dialog box.

5.1 Grouping: a new ATPDraw feature for multilevel modeling

The new grouping facility in ATPDraw allows multilevel modeling by replacing a group of objects with a single icon in unlimited numbers of layers. The grouping structure can be taken as a multi-layer circuit, where the *Edit* | *Edit Group* brings the user one step down in details, while the *Edit* | *Edit Circuit* menu brings one step back. This feature increases the readability of the circuit a great deal and the feature is especially useful for TACS blocks. The grouping feature is demonstrated by re-designing the circuit `Exa_4.adp` in the ATPDraw distribution. This circuit is an induction machine supplied by a pulse width modulated (PWM) voltage source. The induction machine is represented by a Universal Machine type 3 with a typical mechanical load.

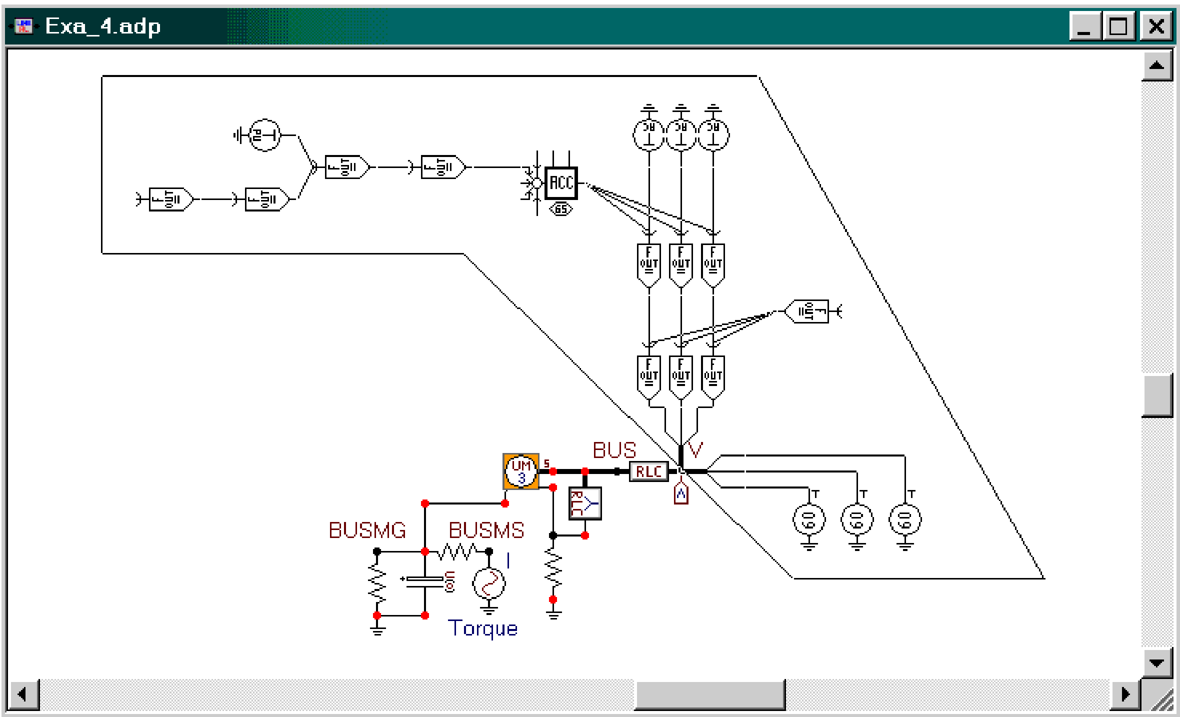


Fig. 5.1 - An induction machine supplied by a pulse width modulated voltage source.

The process of creating a group is as follows:

- Select a group of components (inside the polygon in Fig. 5.1).
- Select *Edit*| *Compress* in the main menu (or Shift+right mouse click + *Compress*).

After selecting a group the *Edit* |*Compress* command will replace it with a single icon. As shown in Fig. 5.2 in the *Compress* dialog box the user can specify the data parameters and the nodes for the new group-object. The selected data and nodes appear as input to the group and their original values are automatically transferred.

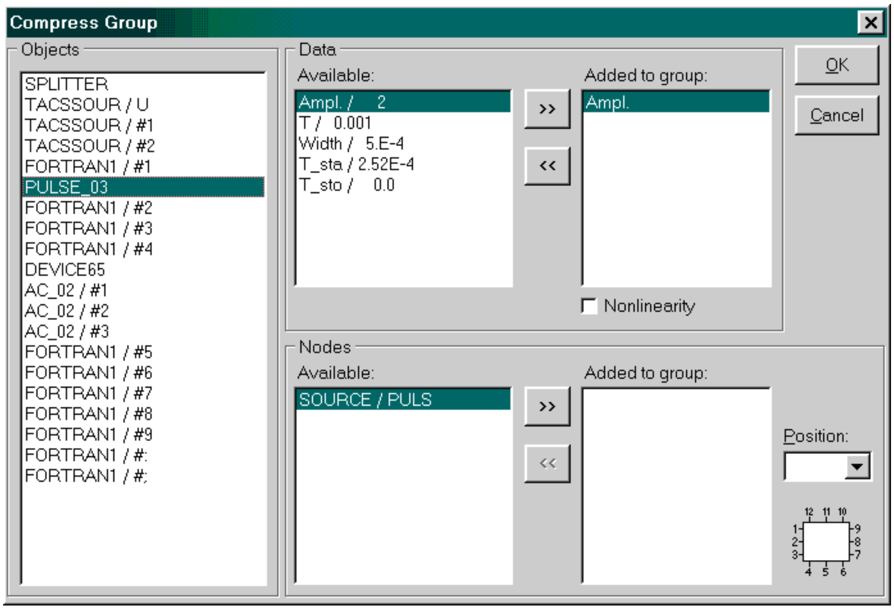


Fig. 5.2 - The new *Compress* dialog window.

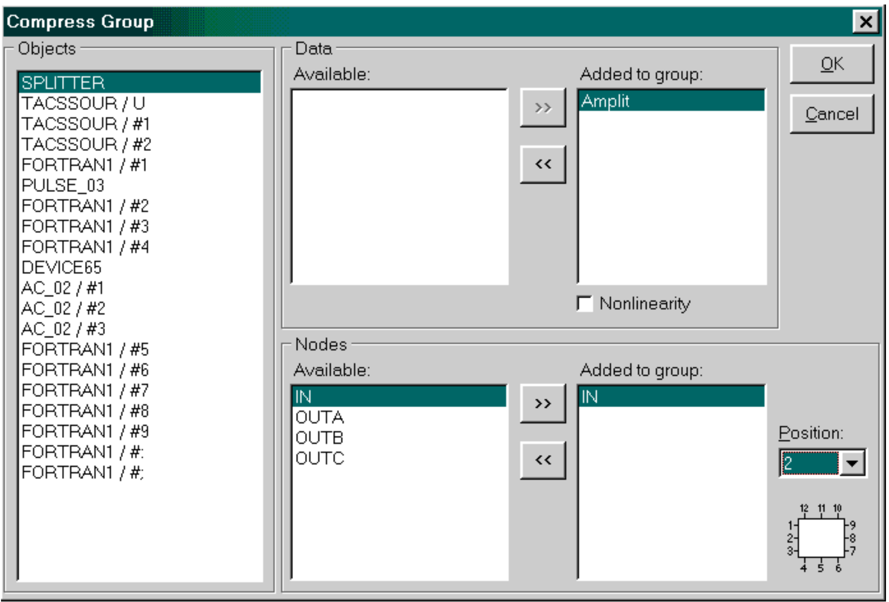
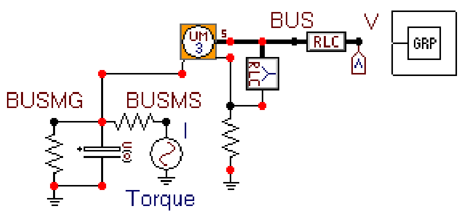


Fig. 5.3 - Name and position of the external nodes of the group.

To the left under *Objects* all components in the group are listed with their name (support file) followed by ' / ' and the label specified by the user in the component dialog box, or by an (alpha)numeric identifier if the label was not specified by the user. When the user clicks on one of the component's name under *Objects*, its data and nodes appears under *Available:* starting with data/node name and followed by their names and values. The user can select a data parameter here and click on the **>>** arrow symbol. The selected data parameter will appear in the field to the right *Added to group:*. The name of the selected data parameter is editable by double-clicking on it. The default data values of the new group object will be equal to the original components' data values. Under *Nodes* the available nodes for the selected component are similarly listed. The user can select the desired nodes and specify the position of the group-object node on the icon border 1-12 as shown in Fig. 5.3. All data and nodes listed in the *Added to group:* will be an external attribute of the new group-object.
Selected data and nodes can also be removed from the *Added to group:* by clicking on the **<<**

arrow button. As all other components, the group object is limited to 36 data and 12 nodes. It is not possible to edit the node and data attributes after the group has been created. If you need to change the group attributes, the only way is to extract the group in the main menu *Edit* | *Extract* (or Shift+right click to call the shortcut menu and select *Extract*) and reselect a new group.



After selecting all the required data and nodes click on *OK*, then a new support file will automatically be created as a temporary file in the `/GRP` folder. The old components disappear and the new group object is drawn in the circuit window as shown in Fig. 5.4. The user is then allowed to connect this group object to the rest of the circuit.

Fig. 5.4 - On return from the *Compress* the circuit is redrawn.

Group objects operate like any other objects. You can drag and place the new group in the desired location. The component dialog of the group can be opened by a right or double mouse click and it appears as shown in Fig. 5.5. The data and node values are specified as in Fig. 5.2 and Fig. 5.3.

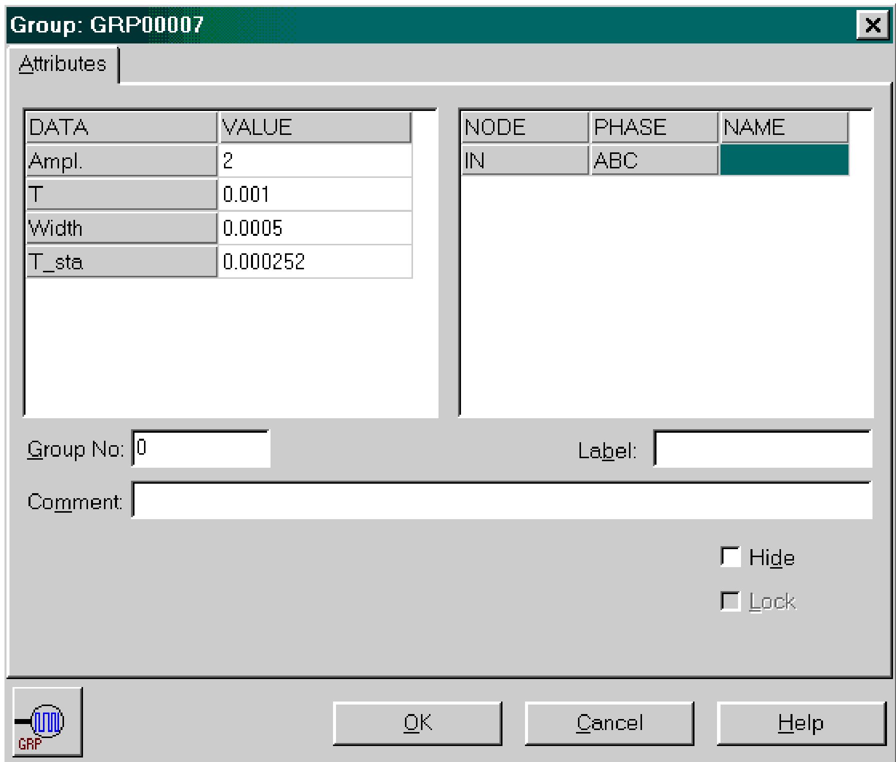


Fig. 5.5 - Opening the new group dialog box.

When changing the data parameter in this window the value will be transferred to the member components. A change in the node name will be transferred in the same way. In this specific case the Fortran TACS objects are connected to the single-phase side of a splitter. The name of the 3-phase node `V` will be transferred as the names `VC`, `VB` and `VA` (from left to right) to the Fortran objects' output node. The user must follow this phase sequence in the PWM group object, too.

The *Compress* process for the mechanical load of the induction machine and the component dialog of the new group can be seen in Fig. 5.6 and Fig. 5.7, respectively.

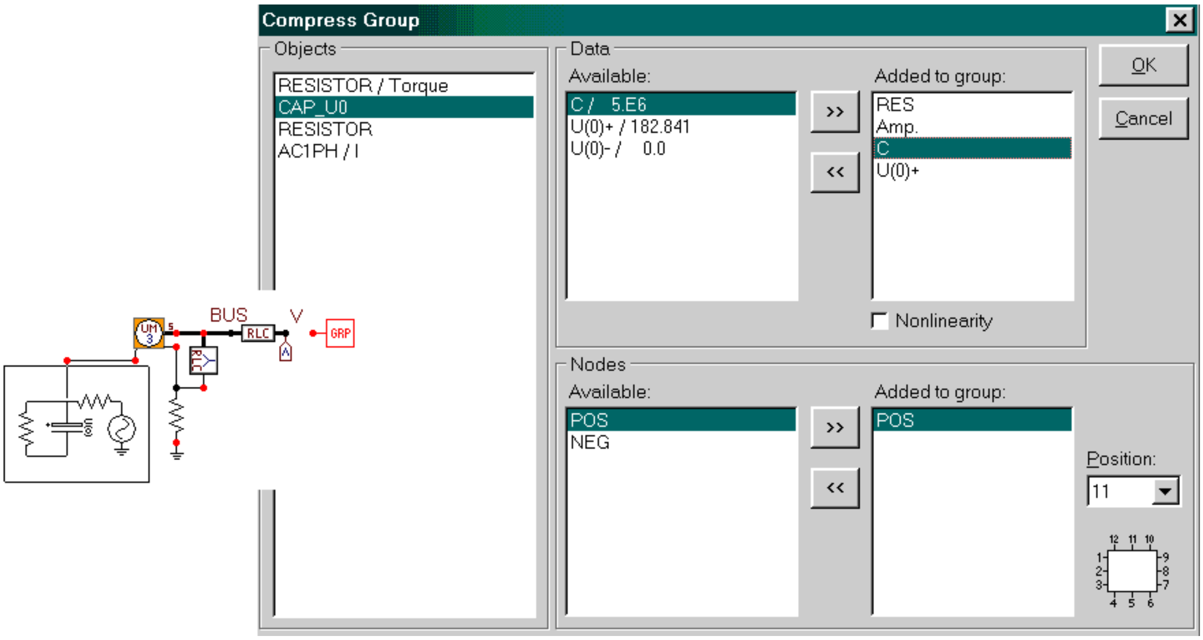


Fig. 5.6 - Selection of data values and external nodes for the mechanical load group.

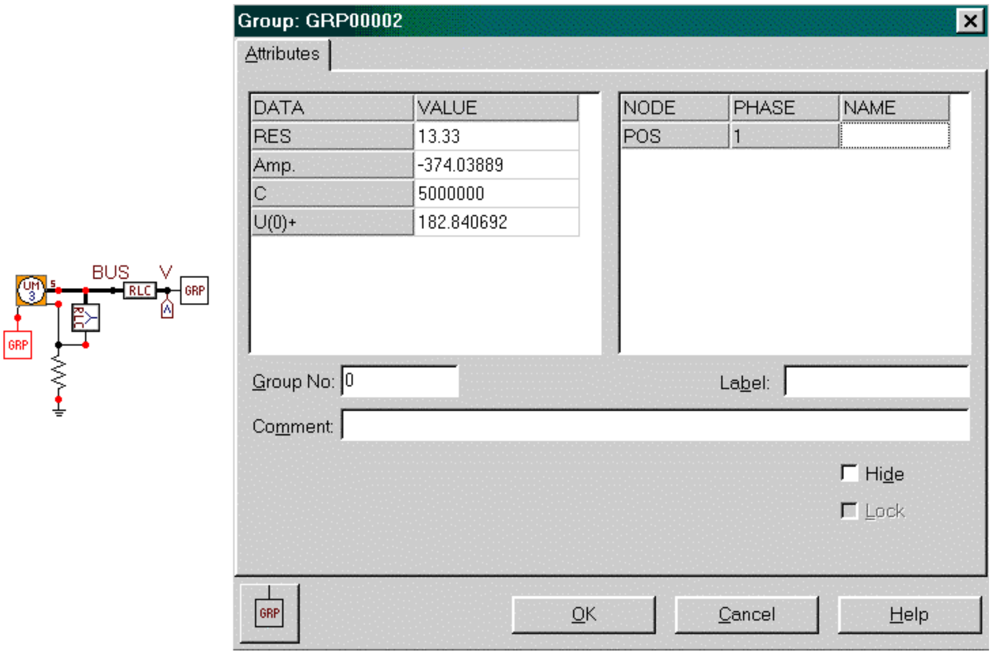
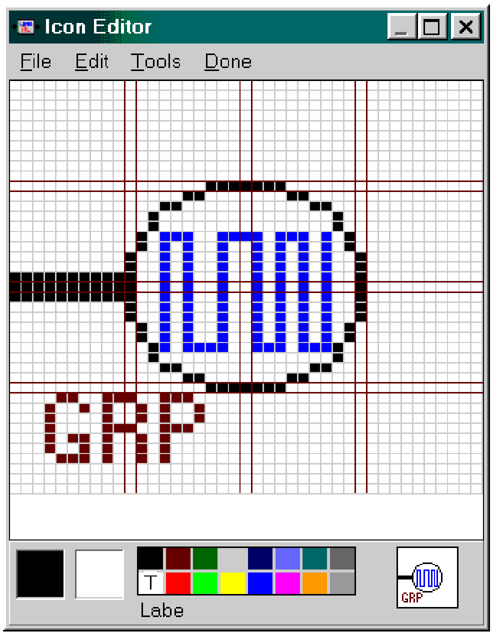


Fig. 5.7- Component dialog box of the mechanical load group-object.

To view/edit a group the user must first select it and then click *Edit* | *Edit Group* in the main menu (or *Ctrl+G*). The group is then extracted on the current circuit window. Actually, the grouping structure can be taken as a multi-layer circuit, where the *Edit Group* brings the user one step down in details, while the *Edit Circuit* brings him one step back. The group is editable in normal way, but the user can't delete components with reference nodes or data in the mother group). I.e. components having been referenced in one of the *Added to group:* lists cannot be deleted. If the user attempts to do so, a *"Marked objects are referenced by compressed group..."* warning message reminds him that the operation is not allowed. Selecting the main menu *Edit* | *Edit Circuit* (or short key *Ctrl+H*) will close the group edit window. It is possible with several levels of groups in the circuit. The maximum number of group levels is 1000.

Each group has a name GRPxxxxx as shown in the component dialog box header in Fig. 5.5 and Fig. 5.7. The number following GRP is incremented for every new group. The files GRPxxxxx.sup are found in the `/GRP` sub-directory and are editable as any other component. The group support files are stored in the project file and are taken as temporary files in ATPDraw. It is safe thus to delete the group sup-files when the project is stored on disk, but not necessary to perform this operation manually, because the program will delete all of them when you finish the ATPDraw session.



To customize the icon, click the *Icon Editor* speed button in the lower left corner of the *Component dialog box* as shown in Fig. 5.5. The icon editor will appear where the user is free to modify the icon. Fig. 5.8 shows the `Exa_4.adp` circuit after grouping the PWM-source and the mechanical load and modifying their icons. Such process is convenient for documentation purposes, because increases the readability of the circuit.

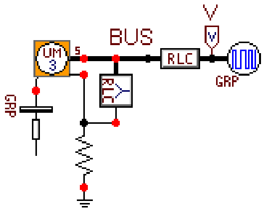


Fig. 5.8 - The icon of the PWM source and the load group has been customized.

Fig. 5.9 - Customizing the icon of the PWM source.

5.1.1 Grouping nonlinear objects

Non-linearity now can be external data in a group object. Up to three objects can share the same external nonlinearity. As an example, this section shows how to create a 3-phase, Type-96 hysteretic inductor. You can draw a circuit as shown to the left of Fig. 5.10. To create a group mark the 3 single-phase inductor and the splitter then select *Edit* | *Compress*. The data `CURR`, `FLUX` and `RESID` are set as external parameters for all the three inductors. The non-linearity button under *Added to group* is checked and the *Add nonlinear* button is checked, too for all three inductors.

When you press *OK* the group object is created. The group dialog box shown in Fig. 5.11 contains only one entry for `CURR`, `FLUX` and `RESID`, which are used for all phases, although 3 copies of them are present in the data structure. This results in 13 free data cells available for the nonlinear characteristic (36-3*3)/2 = 13.5). Only one characteristic is entered in the group's dialog box and it is later copied back to all the three inductors. If that 13 data points were insufficient to describe the characteristic as you wish, select the *Include characteristic* option and specify the characteristic in a disk file. The name of that file must be entered in the *$Include* field.

The new 3-phase Type-96 group object can be stored as a project file in a special library location and later copied into any circuit using the *File* | *Import* command.

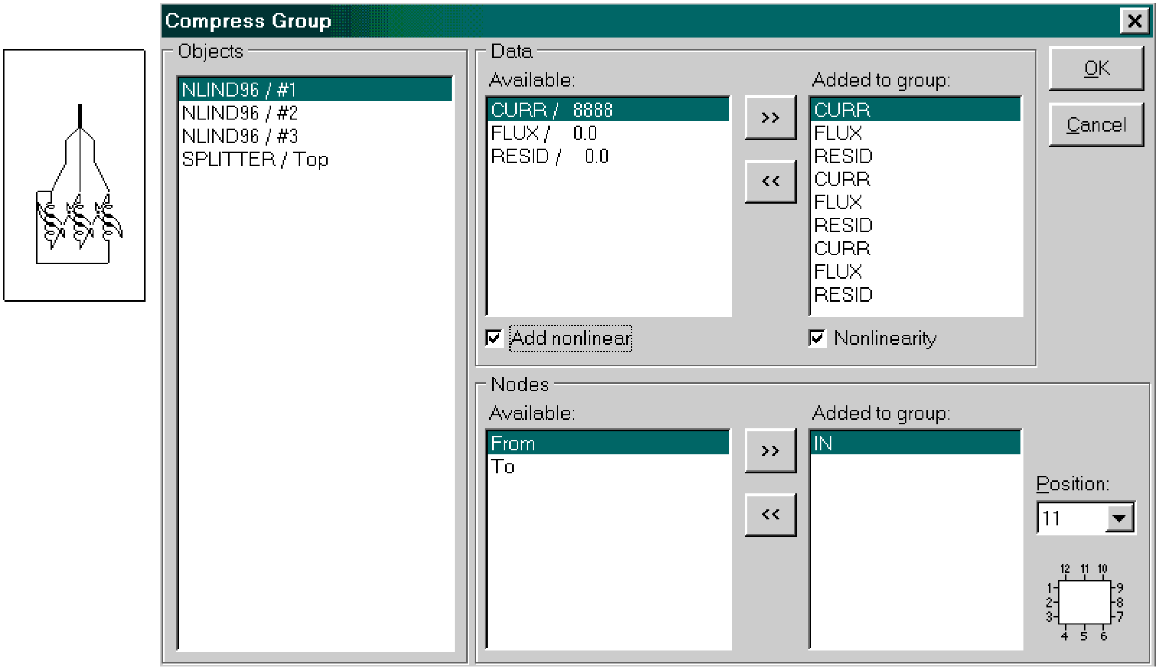


Fig. 5.10 - Creating a 3-phase hysteretic inductor.

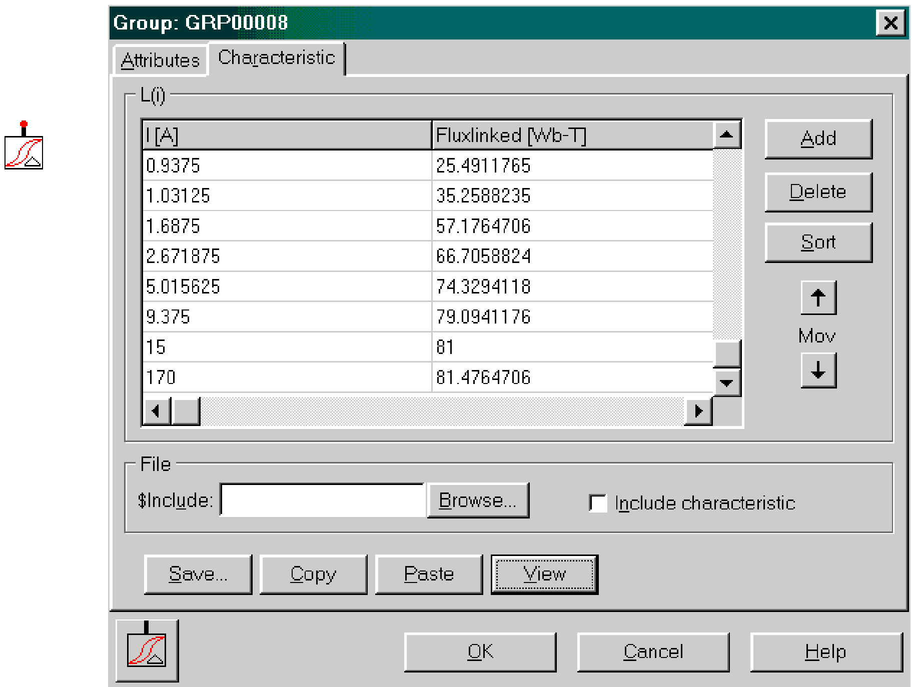


Fig. 5.11 - Nonlinear characteristic of the 3-phase Type-96 group
(notice that only one characteristic is specified, that is used for all phases).

You can customize the group icon as shown in Fig. 5.11. The hysteresis loop originates from the original inductor icon. This is done by executing the next sequence of operations: select the group object and select *Edit* | *Edit Group*, then open the component dialog of one of the inductors and click on the icon editor speed button in it's lower left corner. Select *Edit* | *Copy* in the icon editor, exit the icon editor and the component dialog. Go back to the group object by pressing *Edit* | *Edit Circuit*. In the same way, open the group objects dialog (Fig. 5.11) and click again on the lower left *Icon editor speed button* and select *Edit* | *Paste* in the editor.

5.2 Support of new ATP features $Parameters and Pocket Calculator

$Parameters is a new feature in ATP that allows the user to assign text variables to data and declare these variables for the whole data case later. This feature is particularly useful when a data value is used several times in the circuit. Earlier the user had to open all dialog boxes of the involved components in such cases. This was time consuming and might lead to errors if the user forgets to change values of some components. Since version 3.1 of ATPDraw, the user is also free to assign a 6 (or less) characters text string or variable to most data for standard components instead of a data value in the component dialog box. This is permitted as long as the parameter option is set in the support file. When specifying the data variable the user does not have to think about the number of allowed characters in the ATP-file. ATPDraw will add underscore characters to fill the maximum number of characters. Values can later be assigned to these variables and this is written to the ATP input file within a `$PARAMETER - BLANK` block.

A typical example is shown in Fig. 5.12. This is a single phase rectifying bridge distributed with ATPDraw as `Exa_1.adp`. It consists of 4 diodes with snubber circuits. The RC values of these snubbers are identical for all diodes in the practice and can be specified with text variables `RES` and `CAP` respectively. When the user specifies a variable name for the first time, a message box appears and the user is requested to confirm the operation before the new entry were added to the global list of variables. If you try to enter special characters in this field, an error message prevents this.

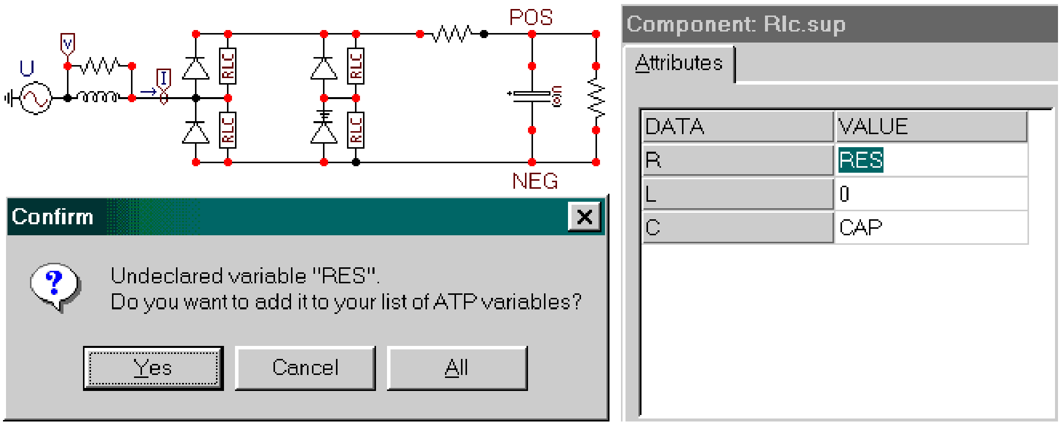
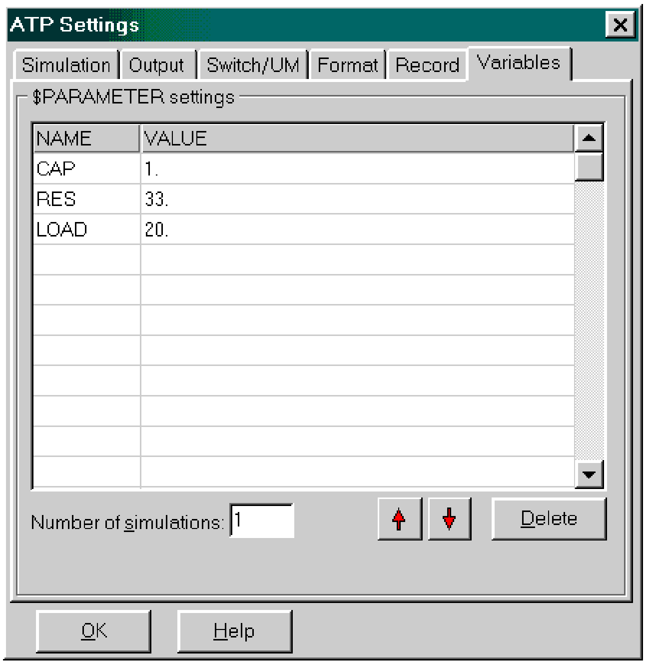


Fig. 5.12 - Specifying text variables `RES` and `CAP` in the component dialog box for an RLC object

Numerical values can be assigned to variables on the *Variables* page under main menu *ATP | Setting* as shown in Fig. 5.13. Variable names are declared in the left column and you can specify data values or a text string in free format in the right column on this page. Sorting the declarations is possible with the arrow buttons. Deleting declarations is also supported.

If variables `RES` and `CAP` are declared twice with different precision settings, i.e. *$Vintage,1* is checked in a component dialog box and unchecked in another, it will be declared twice with 3 and 13 underscore characters added in the $Parameter declaration. This process is hidden, however, but the result is seen in the final ATP-file.

If you change the names in the left side column this will affect the text strings (variables) specified in the components and you will be requested about what action to take (see Fig. 5.14). Available actions are: reset the variable to zero or the default value (from the support file), or select parameter and then decide which variables should replace the no longer defined one.



The *Number of simulations* field is for POCKET CALCULATOR feature of ATP. When this is higher than unity the variable `KNT` can be used in the right column for the current simulation number. This allows multiple ATP simulations where specific data variables can be a function of the simulation number. The specified variables `RES`, `CAP` and `LOAD` are written in the ATP-file followed by underscore characters to enable maximum precision. The `$PARAMETER` cards are written at the bottom of the ATP-file with after a `/REQUEST` card, as shown below.

Fig. 5.13 - Assigning values to the variables.

```
/REQUEST
$PARAMETER
RES_______________ = 33.
RES___= 33.
LOAD__= 20.
CAP_______________ = 1.
CAP___= 1.
BLANK $PARAMETER
```

IMPORTANT! Always use a period '.' after a number in the value field.

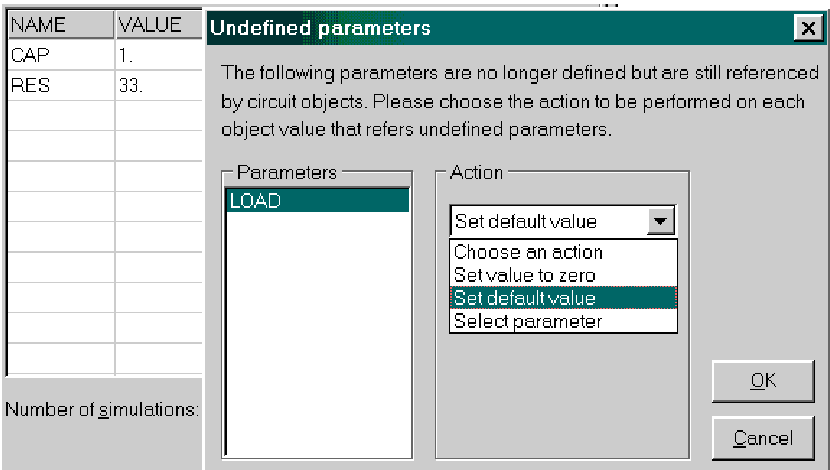


Fig. 5.14 - Action to take when a parameter no longer defined.

5.3 Using the integrated LCC object for line/cable modeling

The integrated LCC objects in ATPDraw are based on the LINE CONSTANTS, CABLE CONSTANTS or CABLE PARAMETERS supporting routines of ATP-EMTP. The user must first describe the geometry of the system and the material constants. ATPDraw then performs an ATP run to process this data case and converts the output punch-file containing the electrical model of the line or cable into standard lib-file format. This lib-file will then be included in the final ATP-file via a `$Include` call. The idea in ATPDraw is to hide as much as possible of the intermediate ATP execution and files and let the user work directly with geometrical and material data in the circuit. Only cases producing an electrical model of the line or cable are supported in ATPDraw.

To use the built-in line/cable module of ATPDraw the user must first select a line/cable component with the desired number of phases (1..9) under *Lines/Cables | LCC* item in the selection menu, as shown in Fig. 5.15. This will display a component in the circuit window that is connected to the circuit as any other component. Clicking on the LCC component with the right mouse button will bring up a special input dialog box called the *Line/cable dialog*. This window contains two sheets; one for the various model specifications and one for the data (geometry and materials) as shown in Fig. 5.16.

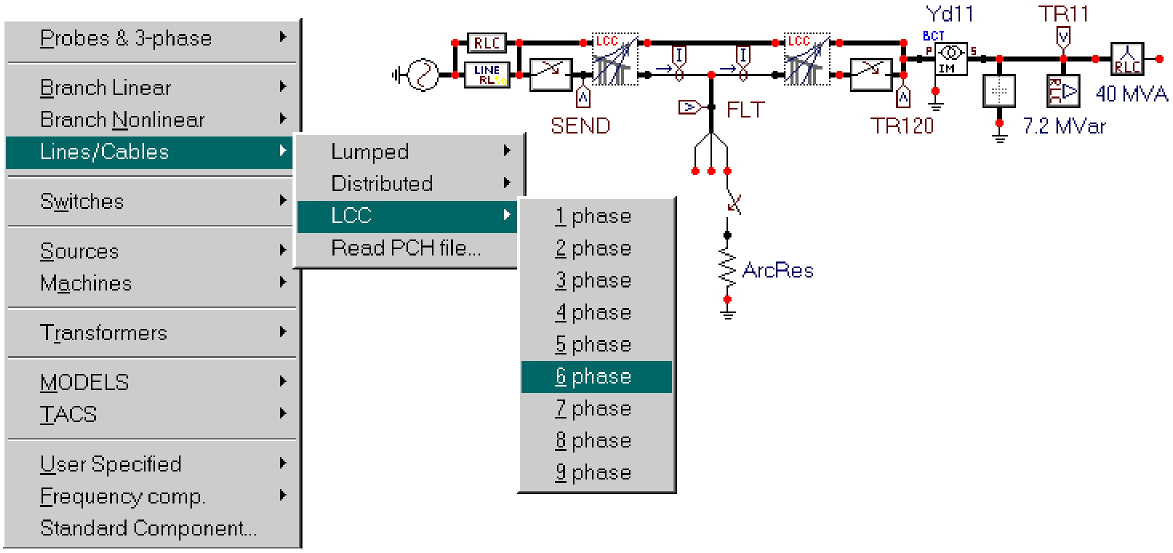


Fig. 5.15 - Selecting a line or cable and connecting the LCC object to the rest of the circuit.

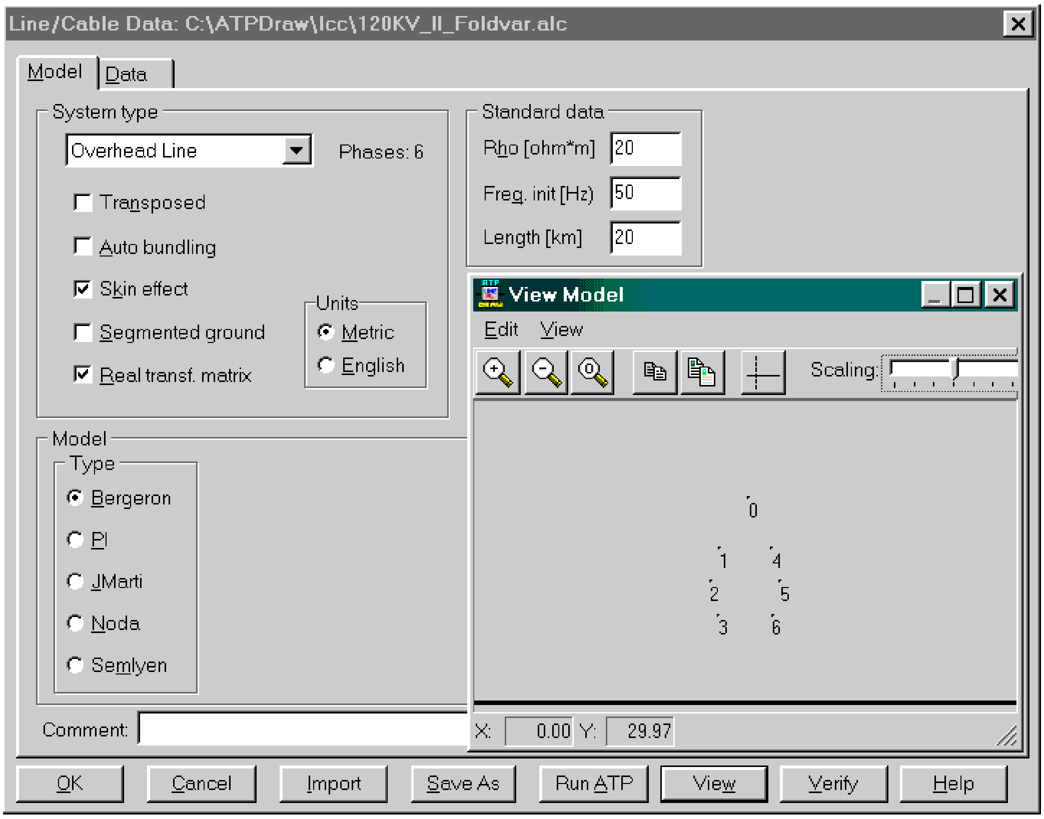


Fig. 5.16 - Line/Cable dialog box: Model specification.

When the required data are specified the user can close the dialog by clicking on *OK*. This will store the specified data to disk in a user selectable `.alc` file (**a**tpdraw **l**ine/**c**able) preferably in the `/LCC` directory. The user is also asked if ATP should be executed to produce the required punch-files. If the user answers *No* on the this question, ATP is not executed, and the user is prompted again later when creating the final ATP-file under *ATP | run ATP* or (*ATP | Make File As...*).

It is very important to ensure a correct ATP installation and setup of the *run ATP* (*F2*) command in ATPDraw. This is done under *Tools | Options / Preferences*. It is recommended to use batch files. Three such files are distributed with ATPDraw (`runATP_S.bat` for the Salford version (DBOS required), `runATP_W.bat` and `runATP_G.bat` for the recommended Watcom or GNU versions of ATP). If the setup of the ATP command is incorrect, the line and cable models will not be produced.

The punch-file output is transferred to a DATA BASE MODULE file by ATPDraw after the successful line parameter calculation, so that the node names are handled correctly. The lib-file required to build the final ATP-file is given the same name as the `.alc` file (with extension `.lib`) and stored in the same folder. The user can also run ATP at any time by clicking on the *Run ATP* button, store the `.alc` data on a different file (*Save As*) or import old `.alc` file data (*Import*). Clicking on the *View* button, displays the cross section of the line/cable as shown in Fig. 5.16. For overhead lines the phase number is displayed (with zero as ground). For cables, the grounded conductors are drawn with a gray color, while the ungrounded conductors are black. The phase number is according to the rule of sequence: first comes the cable with the highest number of conductors and the lowest cable number. The thick horizontal line is the ground surface. Zooming and copying to the Windows clipboard is supported in metafile formats. The *Verify* button of the LCC dialog box helps the user to get an overview of the performance of the model in the frequency domain. This feature is described separately in sub-section 5.4.

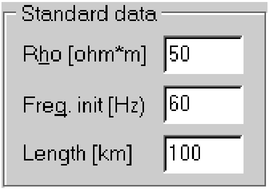
When creating a Noda line/cable model the Armafit program is executed automatically to create the required lib-file. The *Armafit command* is specified under *Tools | Options / Preferences*. The batch file `runAF.bat` is distributed with ATPDraw. The Noda line/cable data format does not allow the full path of the lib-file to be written in the ATP-file. As a result, the `.alc` file should be stored in the same directory as the `.atp` file or the `.lib` file should be moved to this directory manually. For all other line/cable models the *$Prefix/$Suffix* option must be turned off or the lib-files moved to the `/USP` folder.

ATPDraw supports all the various electrical models: Bergeron (KCLee and Clarke), PI-equivalents, JMarti, Noda, and Semlyen. It is straightforward to switch between different models. Under *System type* the user can select between *Overhead Line* and *Single Core Cable* or *Enclosing Pipe*.

In the Line/Cable dialog the user can select between:

System type:	***Model / Type***:
Overhead Line: LINE CONSTANTS *Single Core Cables*: CABLE PARAMETERS or CABLE CONSTANTS *Enclosing Pipe*: CABLE PARAMETERS or CABLE CONSTANTS	*Bergeron*: Constant parameter KCLee or Clark *PI*: Nominal PI-equivalent (short lines) *JMarti*: Frequency dependent model with constant transformation matrix *Noda*: Frequency dependent model (not supported in CABLE CONSTANTS) *Semlyen*: Frequency dependent simple fitted model (not supported in CABLE PARAMETERS)

The *Line/Cable Data* dialog of Fig. 5.16 really consists of three pages: *Model* page, *Line Data* page or *Cable Data* page (although only two of them are visible at a time). The parameter names used in the LCC dialog boxes are identical with that of in Chapter XXI - LINE CONSTANTS and Chapter XXIII - CABLE CONSTANTS parts of the ATP Rule Book [3]. The *Standard data* of the Model page is common for all line and cable types and has the following parameters:



Rho: The ground resistivity in ohmm of the homogeneous earth (Carson's theory).

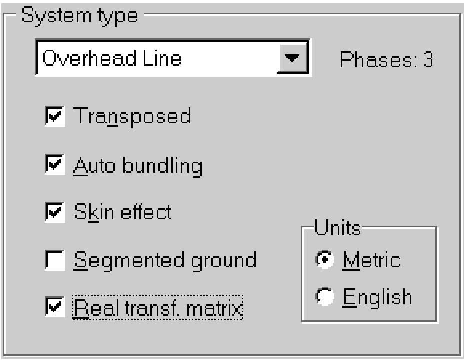
Freq. init: Frequency at which the line parameters will be calculated (Bergeron and PI) or the lower frequency point (JMarti, Noda and Semlyen) of parameter fitting.

Length: Length of overhead line in [m]/[km] or [miles].

Fig. 5.17 - Standard data for all line/cable models.

5.3.1 Model and Data page settings for Overhead Lines

For overhead transmission lines the *System type* settings are as follows. High accuracy (`FCAR=blank`) is used in all cases:



Transposed: The overhead line is assumed to be transposed if the button is checked. Disabled for PI model type.

Auto bundling: When checked this enables the automatic bundling feature of LINE CONSTANTS.

Skin effect: If the button is checked skin effect is assumed (IX=4), if unchecked no skin effect correction. REACT option is set IX=0.

Metric/English: Switching between the Metric and English unit systems.

Fig. 5.18 - System type options for overhead lines.

Segmented ground: Segmented ground wires. If button is unchecked then the ground wires are assumed to be continuously grounded.

Real trans. matrix: If checked the transformation matrix is assumed to be real. The eigenvectors of the transformation matrix are rotated closer to the real axis so that their imaginary part is assumed to become negligible. Recommended for transient simulations. Otherwise a full complex transformation matrix will be used. Recommended for steady state calculations.

5.3.1.1 Model Type settings

Bergeron: No additional settings are required.

PI: For nominal PI-equivalent (short) lines the following optional settings exist under *Data*:

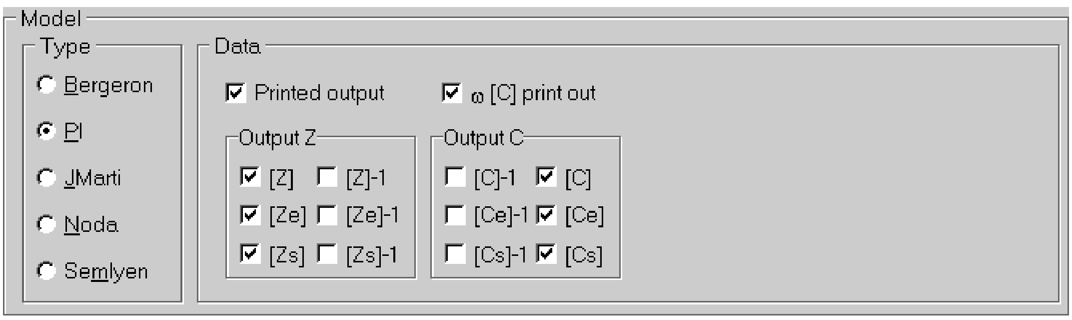


Fig. 5.19 - Optional settings for PI line models.

Printed output: If selected the shunt capacitance, series impedance/admittance matrix of the unreduced system, and/or of the equivalent phase conductor system (after elimination of ground wires and the bundling of conductors), and/or of the symmetrical components will be calculated.

ω[C] print out: Selection between the capacitance matrix and the susceptance matrix (ωC).

JMarti: The JMarti line model is fitted in a frequency range beginning from the standard data parameter *Freq. init* up to an upper frequency limit specified by the mandatory parameters number of *Decades* and the number of sample points per decade (*Points/Dec*). The model also requires a frequency (*Freq. matrix*) where the transformation matrix is calculated and a steady state frequency (*Freq. SS*) for calculation of the steady state condition. Freq. matrix parameter should be selected according to the dominant frequency component of the transient study. The JMarti model needs in some cases modification of the default fitting data under the optional *Model fitting data* field, that can be made visible by unselecting the *Use default fitting* check box. For further details please read in the ATP Rule Book [3].

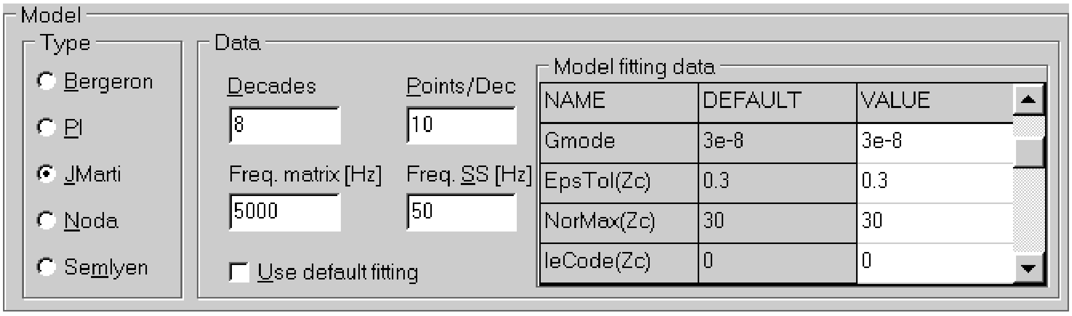


Fig. 5.20 - Parameter settings for the JMarti line model.

Noda: The Noda line model is fitted in a frequency range beginning from the standard data parameter *Freq. init* up to an upper frequency limit specified by the number of *Decades* with the resolution of *Points/Dec*. The model needs a frequency (*Freq. veloc.*), where the wave velocities of the natural modes of propagation are calculated. A value higher than the highest frequency of the frequency scan is usually appropriate. The Noda model needs in some cases modification of the default fitting data under the optional *Model fitting data* field, that can be made visible by unselecting the *Use default fitting* check box. For further details please read in the ATP Rule Book [3].

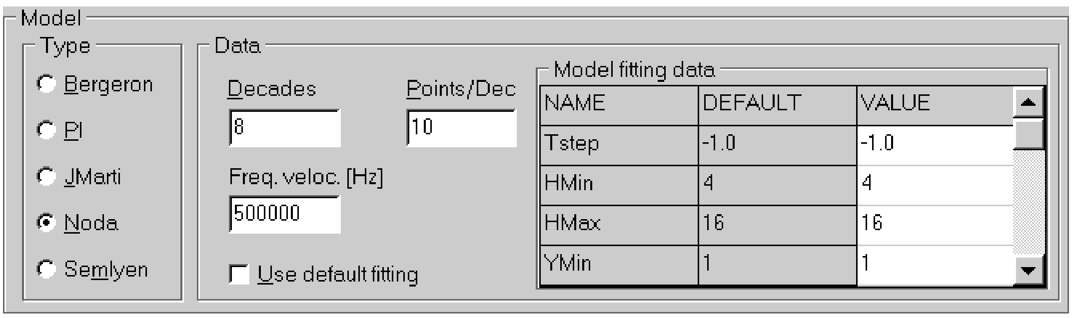


Fig. 5.21 - Parameter settings for the Noda line model.

Semlyen: The Semlyen line model is frequency dependent simple fitted model. Fitting range begins at the standard data parameter *Freq. init* and runs up to an upper frequency limit specified by the parameter number of *Decades*. The model also requires a frequency (*Freq. matrix*) where the transformation matrix is calculated and a steady state frequency (*Freq. SS*) for calculation of the steady state condition. Freq. matrix parameter should be selected according to the dominant frequency component of the transient study. The Semlyen model needs in some cases modification of the default fitting data under the optional *Model fitting data* field, that can be made visible by unselecting the *Use default fitting* check box. For more details please read in the ATP Rule Book.

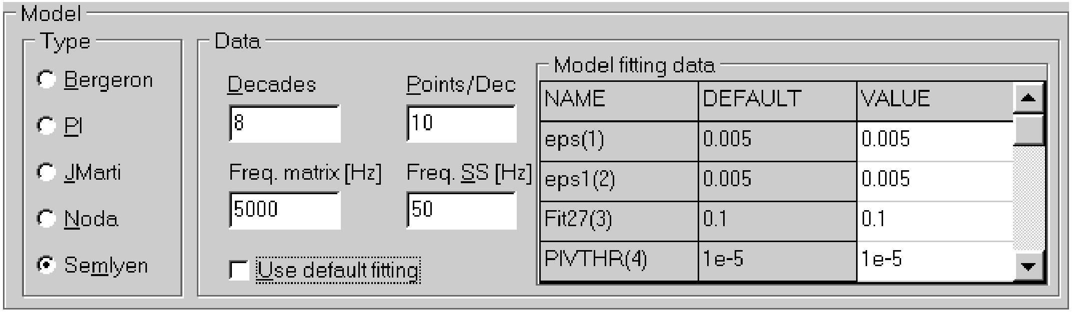


Fig. 5.22 - Parameter settings for the Semlyen line model.

5.3.1.2 Line Data page settings

The data page contains input fields where the user can specify the geometrical or material data. For overhead lines, the user can specify the phase number, conductor diameters, bundling, conductor positions, as shown in Fig. 5.23. The number of conductors is user selectable. ATPDraw set the grounding automatically or gives warnings if the grounding conditions do not match the fixed number of phases. You can *Delete last row* of the table using the gray buttons below or add a new one by clicking on the *Add row* command. Rows inside the table can also be deleted, but it must first be dragged down as last row. To drag a row click on its # identifier in the first column, hold the button down and drag the selected row to a new location or use the ↑ and ↓ arrows at right.

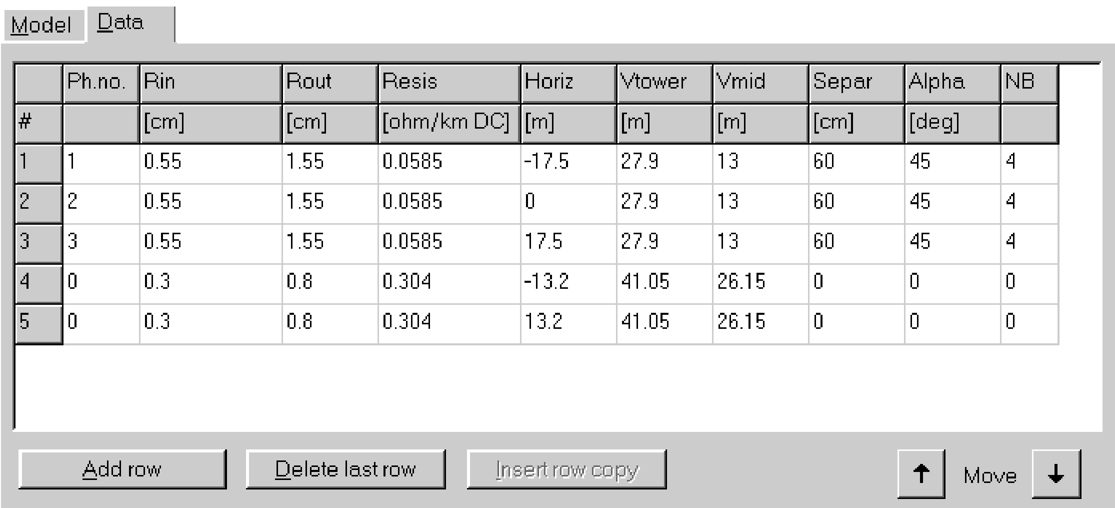


Fig. 5.23 - Line Data dialog box of a 3-phase line. 4 conductors/phase + 2 ground wires.

Ph.no.: phase number. 0=ground wire (eliminated by matrix reduction).

Rin: Inner radius of the conductor. Only available if *Skin effect* check box is selected on the *Model* page (see in Fig. 5.18). If unselected, the *Rin* column is removed and a *React* column appears, where the user specifies the AC reactance of the line in ohm/unit length.

Rout: Outer radius (cm or inch) of the conductor.

RESIS: Conductor resistance (ohm/unit length) at DC (with *Skin effect* checked) or AC resistance at Freq. init (if no *Skin effect* selected).

Horiz: Horizontal distance (m or foot) from the centre of bundle to a user selectable reference line.

Vtower: vertical bundle height at tower (m or foot).

Vmid: vertical bundle height at mid-span (m or foot). The average conductor height calculated from the eq. h= 2/3*Vmid + 1/3*Vtower is used in the calculations.

If *System type / Auto bundling* is checked on the *Model* page (Fig. 5.18):

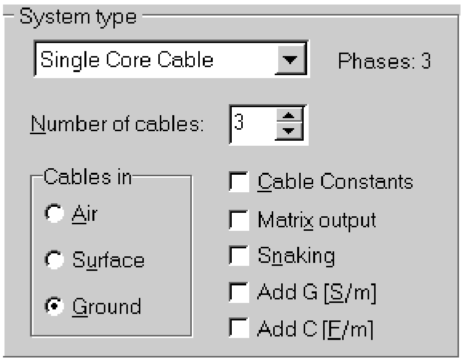
Separ: Distance between conductors in a bundle (cm or inch)

Alpha: Angular position of one of the conductors in a bundle, measured counter-clockwise from the horizontal line.

NB: Number of conductors in a bundle.

5.3.2 Model and Data page settings for Single Core Cable systems

Support of CABLE CONSTANTS and CABLE PARAMETERS has been added to the LCC module of ATPDraw recently and the user can select between the two supporting programs by a single button switch. This enables a more flexible grounding scheme, support of Semlyen cable model instead of Noda and the cascade PI section. On the other hand in CABLE CONSTANTS enabled state ATPDraw does not support additional shunt capacitance and conductance input and Noda model selection. The CABLE CONSTANTS and CABLE PARAMETERS support in ATPDraw does not extend to the special overhead line part and the multi-layer ground model. For Class-A type cable systems which consists of single-core (SC) coaxial cables without enclosing conducting pipe the *System type* settings are as follows:



Cables in: Select if the cables are in the air, on the earth surface or in ground.

Number of cables: Specify the number of cables in the system.

Cable constants: Selects between Cable Constants and Cable Parameters option. If checked, the additional conductance and capacitance option will be switched off and the *Ground* options on the *Cable Data* page will be activated. The Semlyen model is supported only with Cable Constants and the Noda model only with Cable Parameters.

Fig. 5.24 - System type options for SC cables.

Matrix output: Check this button to enable printout of impedance and admittance matrix data (R, ωL and ωC).

Snaking: If checked the cables are assumed to be transposed.

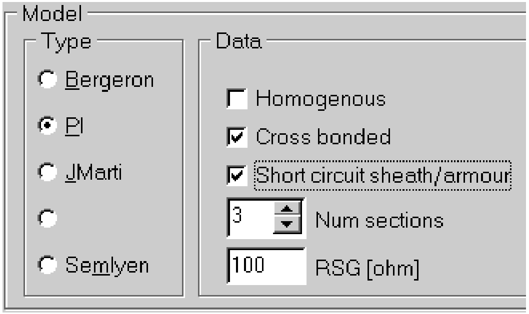
Add G: Check this button to allow conductance between conductors. Not supported for Cable Constants.

Add C: Check this button to allow additional capacitance between conductors. Not supported for Cable Constants.

5.3.2.1 Model Type settings for SC cables

Bergeron, ***JMarti***, ***Noda*** and ***Semlyen:*** The *Model/Type* and *Data* settings for these SC cable models are identical with that of the overhead transmission lines as described in section 5.3.1.1. Users are warned however, that the frequency dependent models may produce unrealistic results, due to neglecting the frequency dependency of the transformation matrix, which is acceptable in overhead line modeling but not for cables.

Cascade ***PI*** model:



If the *Cable Constants* option is selected under the *System type* field, the PI model supports additional input parameters to produce cascade PI-equivalents. The cascade PI model is described in the ATP Rule Book [3]. The *Homogenous* type can be used with all grounding schemes.

Fig. 5.25 - SC cable data for cascade PI output.

5.3.2.2 Cable Data page settings for SC cables

The data page contains input fields where the user can specify the geometrical or material data for cables. The user can turn on sheath/armour by a single button and allowed to copy information between the cables. The cable number is selected in the top combo box with a maximum number specified in *Number of cables* in the Model page.

For CABLE PARAMETERS (*Cable Constants* unselected) the *Ground* options are inactive and number of grounded conductors is calculated internally in ATPDraw based on the total number of conductors in the system and the number of initially selected phases. For CABLE CONSTANTS (*Cable Constants* check box is On) the user must specify which conductor is grounded by checking the appropriate *Ground* buttons. A warning will appear if a mismatch between the number of phases and the number of ungrounded conductors is found. Grounded conductors are drawn by gray color under *View*.

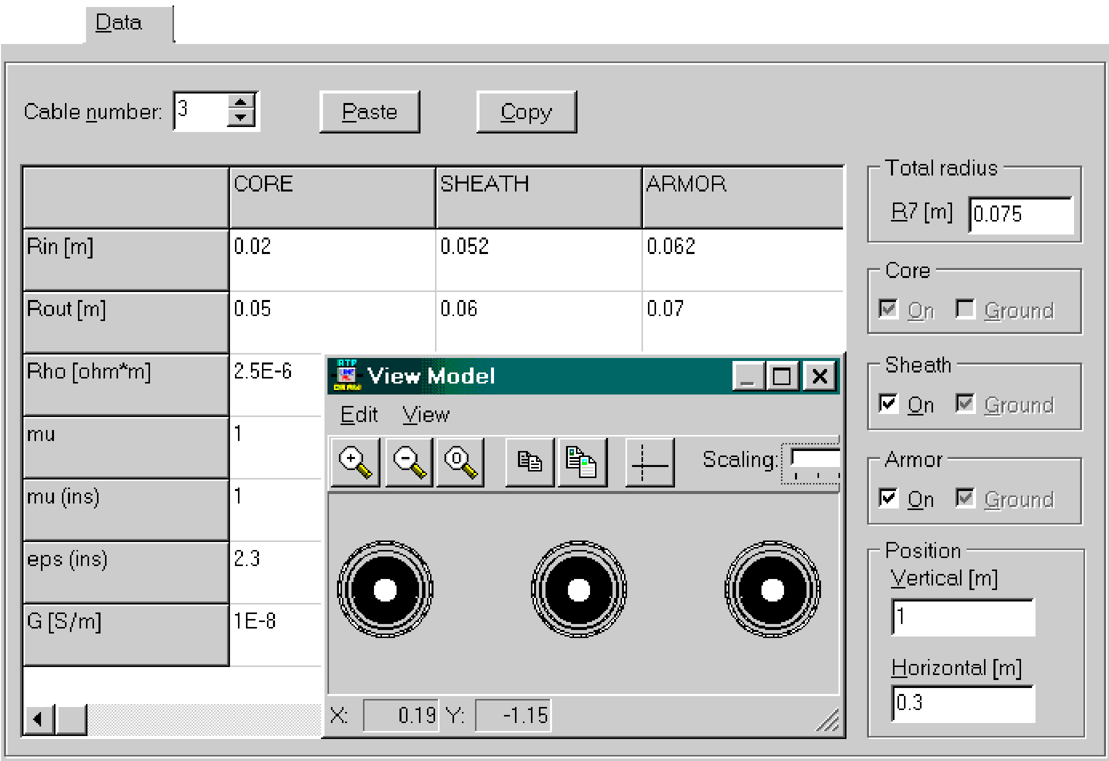


Fig. 5.26 - Cable Data dialog box for a 3-phase SC type cable system.

For each of the conductors Core, Sheath and Armor the user can specify the following data:

Rin: Inner radius of conductor [m].
Rout: Outer radius of conductor [m].
Rho: Resistivity of the conductor material.
mu: Relative permeability of the conductor material.
mu(ins): Relative permeability of the insulating material outside the conductor.
eps(ins): Relative permittivity of the insulating material outside the conductor.
Total radius: Total radius of the cable (outer insulator) [m].
Sheath/Armour On: Turn on optional Sheath and Armour conductors.
Position: Vertical and horizontal positions relative to ground surface and to a user selectable reference line for single core cables.

5.3.3 Model and Data page settings for Enclosing Pipe type cables

This selection specifies a cable system consisting of single-core (SC) coaxial cables, enclosed by a conducting pipe (referred as Class-B type in the ATP Rule Book [3]). The cable system might be located underground or in the air. The *System type* settings are identical with that of the Class-A type cables (see in sub-section 5.3.2). When the button *Cable Constants* is checked the shunt conductance and capacitance options are disabled and a new check box *Ground* controls the grounding condition of the pipe. Transposition of the cables within the pipe is available via the *Snaking* button. Cascade PI options can be specified similarly to SC cables (see Fig. 5.25). For cables with enclosing pipe, the following *Pipe data* are required:

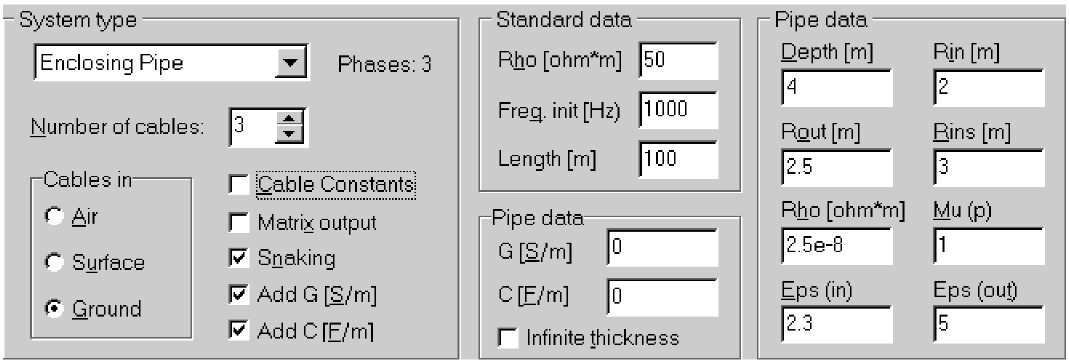


Fig. 5.27 - System type and Pipe data settings for an Enclosing Pipe cable.

Depth: Positive distance in meter between pipe center and ground surface.
Rin: Inner radius of the pipe in meter.
Rout: Outer radius of the pipe in meter.
Rins: Outer radius of outer insulation (total radius) in meter.
Rho: Resistivity of the pipe conductor.
Mu: Relative permeability of the pipe conductor.
Eps(in): Rel. permittivity of the inner insulation (between cables and pipe).
Eps(out): Rel. permittivity of the outer insulation (around pipe).
G and C: Additional shunt conductance and shunt capacitance between the pipe and the cables.
Infinite thickness: Infinit thick pipe. ISYST=0 and (uniform grounding).

The cable *Data* page input fields for Enclosing Pipe type cable systems are identical with that of the SC cables (see sub-section 5.3.2.2). The only difference is the meaning of *Position:*

Position: Relative position to pipe center in polar coordinates (distance and angle).

5.4 Verification of the Line/Cable model performance

The *Verify* button of the LCC dialog box helps the user to get an overview of the performance of the model in the frequency domain. This feature of ATPDraw enables the user to compare the line/cable model with an exact PI-equivalent as a function of frequency, or verify the power frequency benchmark data for zero/positive short circuit impedances, reactive open circuit line charging, and mutual zero sequence coupling. The *Verify* module supports two types of frequency tests:

1) LINE MODEL FREQUENCY SCAN (LMFS) as documented in the ATP benchmark files `DC51/52.dat`. The LMFS feature of ATP compares the punched electrical model with the exact frequency dependent PI-equivalent as a function of a specified frequency range.
2) POWER FREQUENCY CALCULATION (PFC) of zero and positive short circuit impedances and open circuit reactive line charging, and mutual zero sequence impedance for multi circuit lines.

In the *Verify* dialog box as shown in Fig. 5.28 the user can choose between a LINE MODEL FREQUENCY SCAN (LMFS) or a POWER FREQUENCY CALCULATION (PFC) case. Under *Circuit specification,* each phase conductor is listed for which the user should assign a circuit number. The phase order for overhead lines is from the lowest phase number and up to the one assigned under *Data* in the Line/Cable dialog box. For cables, the cable with the highest number of conductors and the lowest cable number comes first (rule of sequence, ATP Rule Book - Chapter XXIII). A circuit number zero means that the conductor is grounded during the frequency test. For the LMFS test the user must specify the frequency range (*Min freq* and *Max Freq*) along with the number of points per decade for the logarithmic space frequencies. For the PFC test, the input parameters are the power frequency and the voltage level (used to calculate the reactive line charging). Note! The LMFS feature of ATP does not work for Noda models.

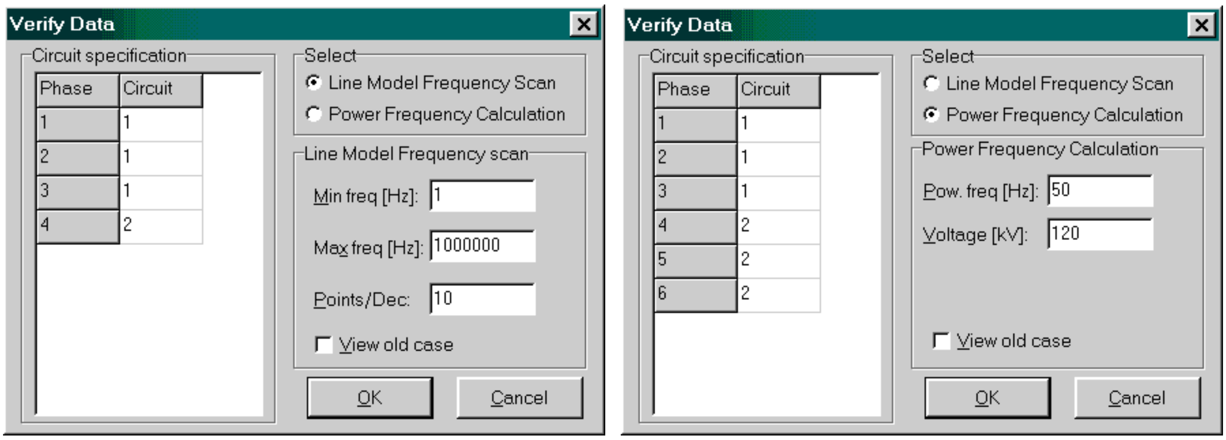


Fig. 5.28 - Frequency range specification for the LMFS run (left) and selecting the line voltage and system frequency for the PFC run (right).

a) Select LMFS: Clicking on *OK* will result in the generation of a LMFS data case called `xVerify.dat` and execution of ATP based on the settings of the default ATP command (*Tools|Options/Preferences*). The sources are specified in include files called `xVerifyZ.dat`, `xVerifyP.dat`, and `xVerifyM.dat` for the zero, positive and mutual sequence respectively. The individual circuits are tested simultaneously. The receiving ends are all grounded (over 0.1 mΩ) and all sending ends (if *Circuit number > 0*) attached to AC current sources of 1 Amps. The phase angle of the applied current source for the i^{th} conductor is $-360\cdot(i-1)/n$ where n is the total number of conductors belonging to that circuit. Phase angle for the zero sequence tests are zero.

The mutual coupling works only for 6-phase lines. For circuit one all phases are supplied with zero phase angle sources, while the phase conductors of the other circuit at the sending end are open. The *View old case* button will skip creation of the LMFS data case and trace the program directly to the procedure that reads the `xVerify.lis` file, which contains the input impedances of the electrical model compared to the exact PI-equivalent as function of frequency under various conditions. ATPDraw can read this file and interpretation of the results is displayed in the *LMFS results* window as shown in Fig. 5.30 for the 4-phase JMarti line-model specified in Fig. 5.29.

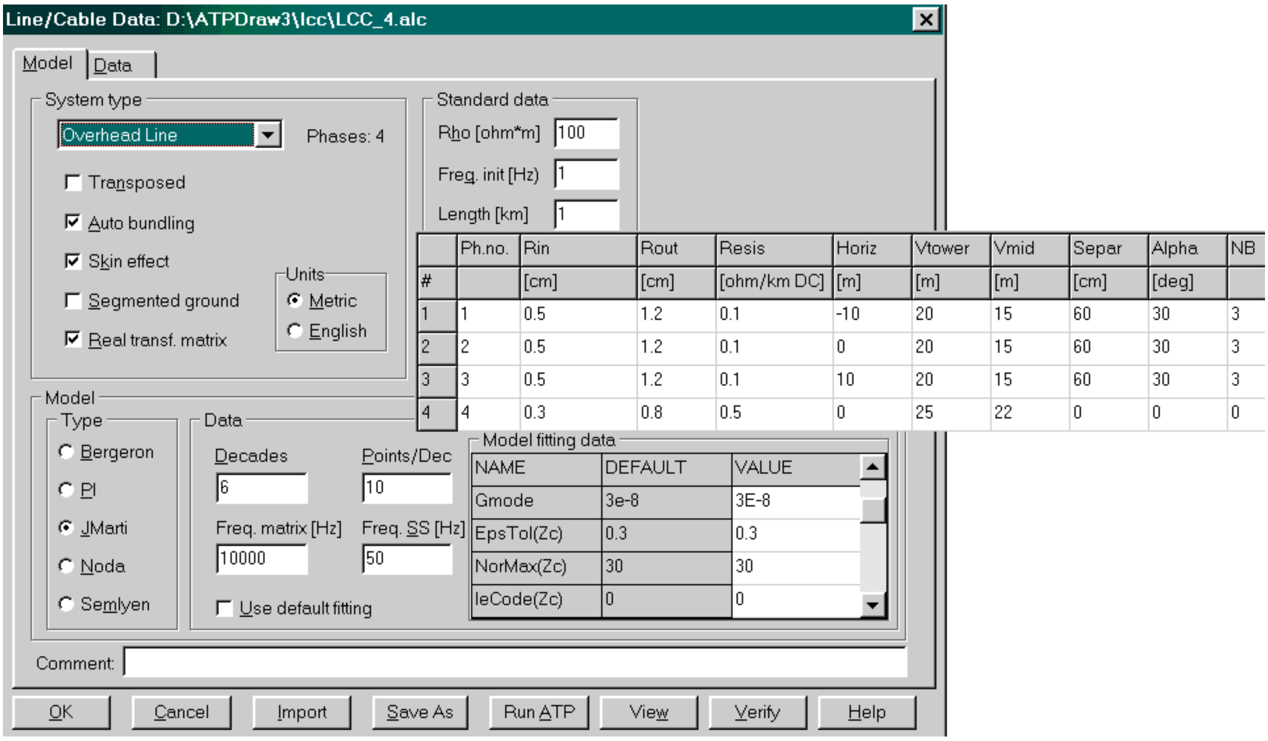


Fig. 5.29 - Specification of a 4-phase JMarti line model.

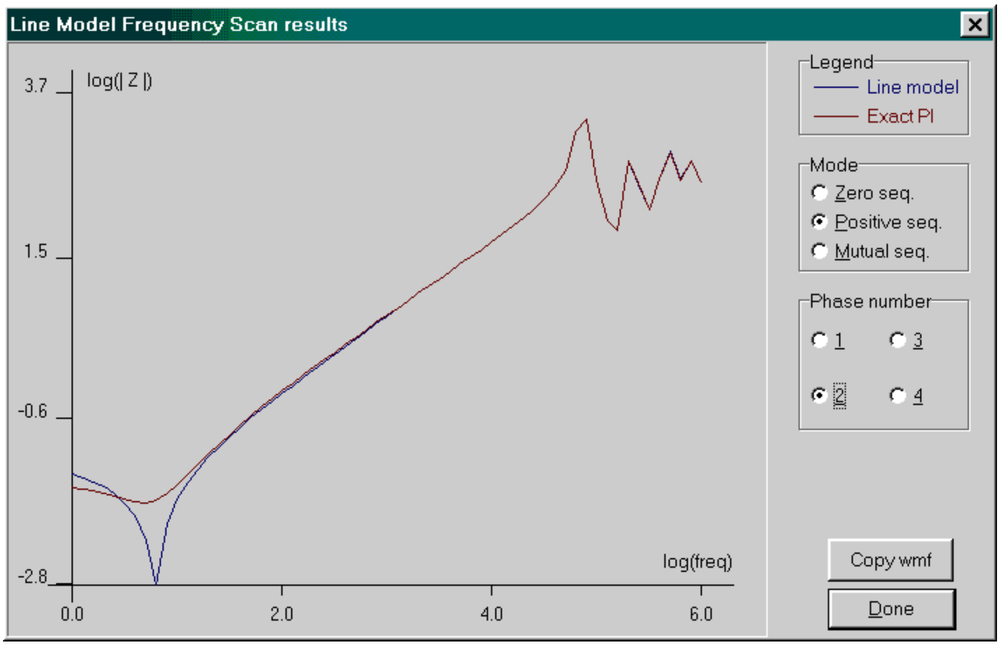


Fig. 5.30 - Verifying a JMarti line model 1 Hz to 1MHz. Model is OK for $f > 25$ Hz.

In Fig. 5.30, the user can select the *Mode* and the *Phase number* of which the absolute value of the input impedance is displayed to the left in a log-log plot. It is also possible to copy the curves to the windows clipboard in metafile format (*Copy wmf*). The absolute value of the input impedance of the model and the exact pi-equivalent can be compared for the following cases:

Zero-sequence: AC currents of 1 A with zero phase angle is applied to all phases simultaneously while the other end of the line/cable is grounded. The zero-sequence impedance is thus equal to the voltage on the sending end of each phase.

Positive sequence: AC currents of 1 A with a phase angle of *-360*(i-1)/n* is applied to all phases, where *i* is the current phase number in the specific circuit and *n* is the total number of phases in the circuit. (A 6-phase line/circuit will result in phase angles 0, -120, -240, 0, -120, -240 while a 4 phase circuit will result in 0, -90, -180, -270). The user specifies a circuit number for each phase under *Circuit specification* of *Verify Data* dialog. The receiving end is grounded.

Mutual sequence: AC currents of 1 A with zero phase angle is applied to all phases of the first circuit, while the other circuit is open. The receiving ends of all phases are grounded. Apparently this works only for 6-phase lines.

b) Select PFC: For the PFC test the user must specify the power frequency and the base voltage level for scaling of the reactive charging. Clicking on *OK* will result in the generation of a PFC data case called `xVerifyF.dat` and execution of ATP based on the settings of the ATP-Command (*Tools* | *Options / Preferences*). In this case, each circuit is tested individually (all other phases are left open while a specific circuit is tested). The library file describing the electrical model of the line/cable is included in a new ATP case an supplied by unity voltage or current sources in order to calculate the steady state short circuit impedances and open circuit reactive line charging. The file `xVerifyF.lis` is read by ATPDraw and the short circuit impedances together with the open circuit line charging is calculated in the zero-sequence and positive-sequence mode. The results of the calculations are displayed in Fig. 5.31.

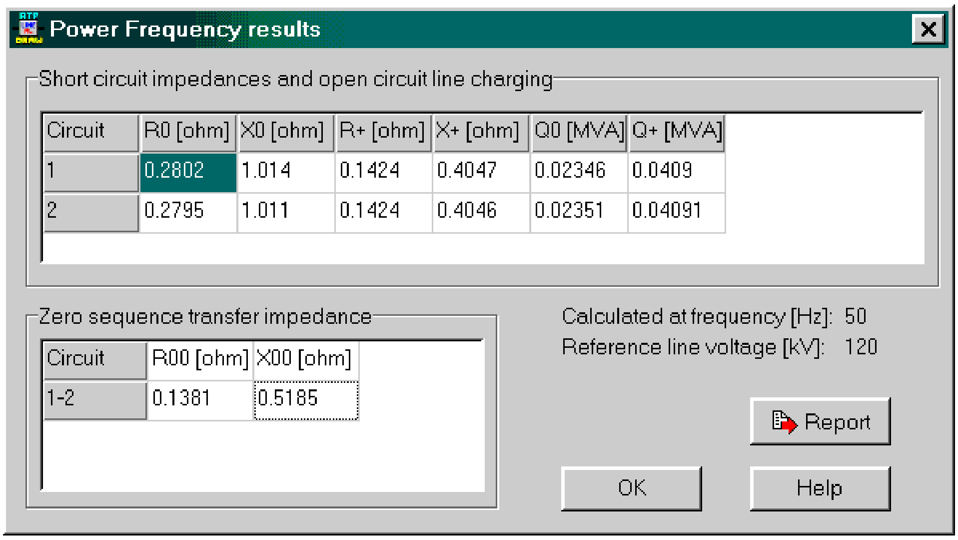


Fig. 5.31 - Results of the PFC run.

If the user clicks on *Report* the content in the string grids of Fig. 5.31 will be dumped to a user selectable text file. Further details about the operation of the *Verify* feature and PFC option can be found in the Appendix part of the Manual.

5.5 Using MODELS simulation language

MODELS is a general-purpose description language supported by a set of simulation tools for the representation and study of time-variant systems. This chapter of the Manual is to a large extent an extract of the *MODELS IN ATP -Language Manual*, February 1996 [4] reference. Please consult this manual for more detailed information on the MODELS language.

MODELS language focuses on the description of the structure of a model and on the function of its elements. There is a clear distinction in MODELS between the description of a model, and the use of a model. Individual models can be developed separately, grouped in one or more libraries of models, and used in other models as independent building blocks in the construction of a system. The description of a model is intended to be self-documenting. A system can be described in MODELS as an arrangement of inter-related sub models, independent from one another in their internal description and in their simulation (e.g. individual models can have different simulation time step). Description of each model uses a free-format, keyword-driven syntax of local context, and does not require fixed formatting in its representation.

The main description features of the MODELS language are the following:

- The syntax of MODELS allows the representation of a system according to the system's functional structure, supporting the explicit description of composition, sequence, concurrence, selection, repetition, and replication;
- The description of a model can also be used as the model's documentation;
- The interface of a model with the outside world is clearly specified;
- The components of a model can be given meaningful names representative of their function
- A system can be partitioned into individual sub models, each with a local name space;
- The models and functions used for describing the operation of a system can be constructed in programming languages other than the MODELS language.

The main simulation features supported by the MODELS language are the following:

- Distinction between the description of a model and its use, allowing multiple independent replications of a model with individual simulation management (time step, dimensions, initial conditions, etc.);
- Hierarchical combination of three initialization methods (default, use-dependent, and built-in), each contributing to the description of the pre-simulation history of a model by a direct representation of the pre-simulation value of its inputs and variables as functions of time;
- Dynamically-controlled modification of the values of the inputs and variables of a model during the course of a simulation;
- Dynamically-controlled modification of the structure of a model (both topological composition and algorithmic flow) during the course of a simulation.

The new MODELS object created in this chapter is part of the ATPDraw's example file `Exa_8.adp`. This example is taken from ATP's benchmark file `DC68.DAT`, subcase 7. The data case is a series compensated 500 kV interconnection. ZnO arresters protect the series capacitors and the energy dissipation of arresters is continuously monitored by a MODELS object. If certain power or energy limits are exceeded the series capacitor is bypassed through TACS/MODELS controlled switches to avoid arrester overheating. The complete model is described in the Application part of this Manual.

ATPDraw supports only a simplified usage of MODELS. In general, ATPDraw takes care of the interface between MODELS and the electrical circuit (`INPUT` and `OUTPUT` of the MODELS section) and the execution of each model (`USE`). Creating a new Model in ATPDraw normally consists of two steps:

3. To create a model file (`.mod`) containing the actual model description.
4. To load this file into the circuit via the *MODELS | Files (sup/mod)...* sub-menu of the component selection menu.

5.5.1 Creating the model file

The actual model file describing the operation of the model must be written outside of ATPDraw using an ASCII text editor or the built in *Model Editor* under *Objects | Model | New mod-file*. Below the model in `DC68.DAT` has been modified a bit since it is not allowed to use expressions in the USE of a model. Instead of calculating the voltage across the arrester in the `USE` statement, the two node voltages at each side are sent as input parameters and the difference is calculated inside the model `vcap:=V1-V2`.

```
MODEL FLASH_1
comment ****************************************************************
*                                                                      *
*    Function: set or cancel the gap firing control signal             *
*    Inputs  : voltage and current across ZnO resistor                 *
*    Output  : the firing signal to the electrical ZnO component       *
*                                                                      *
************************************************************ endcomment
  INPUT  V1     --  Voltage on positive side of ZNO        [V]
         V2     --  Voltage of negative side of ZNO        [V]
         iczn   --  ZNO current                            [Amps]
  DATA   Pset   --  power setting                          [Megajoules/msec]
         Eset   --  energy setting                         [Megajoules]
         fdel   --  firing delay                           [msec]
         fdur   --  firing duration                        [msec]
   VAR power    --  power into ZnO resistor                [Watts]
       trip     --  gap firing control signal              [0 or 1]
       energy   --  energy into ZnO resistor               [Joules]
       tfire    --  time at which the gap was last fired   [sec]
       vcap     --  voltage difference across series caps  [Volts]
  OUTPUT  trip
  HISTORY  INTEGRAL(power) {DFLT:0}
  INIT trip:=0
       tfire:=0
  ENDINIT
  EXEC
    ------------------------------------------------------------------
    vcap:=V1-V2
    power:=vcap*iczn
    energy:=INTEGRAL(power)
    ------------------------------------------------------------------
    IF  trip>0              -- is already firing
    AND t-tfire>fdur*1.e-3  -- has exceeded firing duration
    THEN
      trip:=0     -- cancel the firing signal
      tfire:=0    -- null the firing time
    ENDIF
    ------------------------------------------------------------------
    IF trip=0                         -- is not signaling to fire
    AND tfire=0                       -- firing condition not yet detected
    AND (     power  >= Pset * 1.e9   -- power setting exceeded
          OR energy >= Eset * 1.e6 )  -- energy setting exceeded
    THEN
      tfire:=t    -- set the firing detection time
    ENDIF
    ------------------------------------------------------------------
    IF trip=0                -- is not signaling to fire
```

```
      AND tfire>0              -- firing condition has been detected
      AND t-tfire>=fdel*1.e-3   -- firing delay exceeded
      THEN
        trip:=1     -- set the firing signal
      ENDIF
    ENDEXEC
  ENDMODEL
```

The model file must be given a name with extension .mod and be stored in the \MOD folder. In this example the name FLASH_1.MOD has been chosen. The *name of the disk file* must be equal to the *name of the model* given in the first line of the model description.

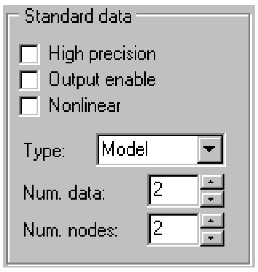
5.5.2 Creating a new MODELS object

An ATPDraw object consists of a file on disk, which is called support file, with extension .sup. Each model must have a support file in order to be used in ATPDraw. A support file is standard for all components in ATPDraw and contains the icon, information on node types and position and on data parameters. There are two options for creating a model object in ATPDraw:

1. Manual operation: creating a support file manually under *Object | Model | New sup-file*.
2. Automatic operation: selecting a .mod file directly under *MODELS* of the component selection menu and let ATPDraw to create the support file.

5.5.2.1 Manual operation

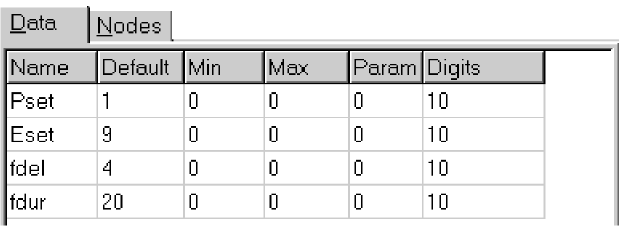
To create a new a support file manually, enter the *Objects* menu and select the *New sup-file* under *Model*. This menu item will perform the *Edit Object* dialog. In the *Standard data* field, you specify the size of the model: number of nodes and number of data as shown in Fig. 5.32.



The *FLASH_1.MOD* file has four nodes (3 input + 1 output) and four data, (*Pset, Eset, fdel, fdur*), so you must enter 4 in both *Num*. fields. Note! All MODELS nodes must be single phase one.

Fig. 5.32 - Specify the size of the model.

After you have specified the node and data values go to the tabbed notebook style part of the dialog box. Select the *Data* page where you specify the values shown in Fig. 5.33. The *Name* of the data must be the same as those used in the DATA declaration part of the .mod file. The *Default* value appears initially in the models dialog. The default values are taken from the Use Model statements in DC68.DAT (you can of course change these values individually for each use of the model). *Min* and *Max* restrict the legal input range. No restriction is applied here to data values, so *Min=Max*.

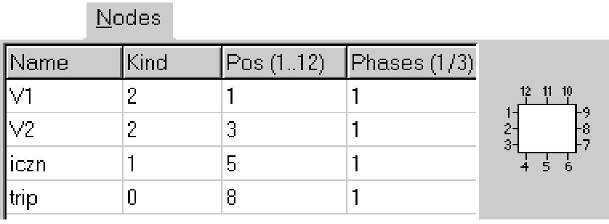


Name	Default	Min	Max	Param	Digits
Pset	1	0	0	0	10
Eset	9	0	0	0	10
fdel	4	0	0	0	10
fdur	20	0	0	0	10

Fig. 5.33 - Specify Data parameters.

Param is set to 0, which means that no variable text string can be assigned to the data value. *Digit* is the maximum number of digits allowed in the ATP input file. When *high precision* is checked, *$Vintage, 1* is enabled and Digits is split in two values for high and low precision.

After you have specified the data values click on the *Nodes* tab to enter to the node window as shown in Fig. 5.34. The *Name* identifies the node in the Node and Component dialog boxes. The name you enter here must be the same as those used in the INPUT and OUTPUT declaration sections of the .mod file. The *Position* field is the node position on the icon border as shown at the right. The *Kind* value is the input/output type of the node. Number of *Phases (1/3)* must be set to 1 for all Models node because only single-phase nodes are supported.

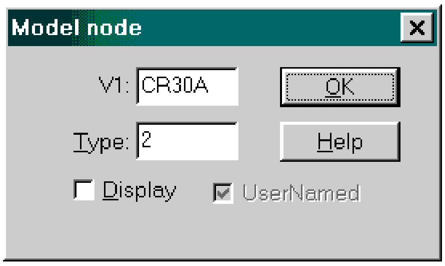


Name	Kind	Pos (1..12)	Phases (1/3)
V1	2	1	1
V2	2	3	1
iczn	1	5	1
trip	0	8	1

Fig. 5.34 - Specifying Node attributes.

Supported *Kind* values for MODELS objects are:

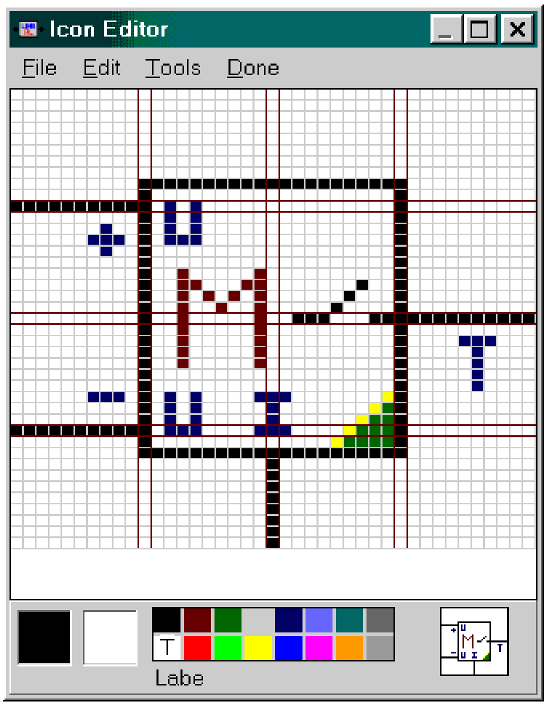
```
0:   Output node.              3:   Switch status input node.
1:   Current input node.       4:   Machine variable input node.
2:   Voltage input node.       5:   TACS variable (tacs)
6:   Imaginary part of steady-state node voltage (imssv)
7:   Imaginary part of steady-state switch current (imssi)
8:   Output from other model.
```

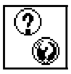


The *Kind* parameter of model object nodes can be changed later in the Node dialog box (input field *Type*), as shown in Fig. 5.35. This window appears when the user clicks on a Model node with the right mouse button.

Fig. 5.35 - Model node dialog box.

Note! If a model output is used as input for another model, the model, which produces the output must be USEd before the use of the model that is supplied with this output. This can be done by specifying a lower group number for the model with output signals and selecting the *Sorting by group number* option under *ATP* | *Settings / Format*.



Model objects also have an icon, which represents the object on the screen and an optional help, which describes the meaning of parameters. If no user supplied help text was given, the *Help Viewer* displays the model definition file (.mod) automatically. If you need a help text, this feature can be overridden by opening the *Help Editor* with the  button at the right hand side of the dialog box.

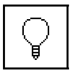
The *Icon Editor* appears similarly, by clicking on the  button. Here you can be creative and draw a suitable icon for the new model object as shown in Fig. 5.36. When you finished select the *Done* menu item.

Fig. 5.36 - The icon of the new model objects.

The *Save* or *Save As* buttons can be used to save the new support file to disk. Default location of Model support files is the \MOD folder. The .sup file does not need to have the same name as the model file, but it is recommended.

The new model object has now been created is ready for use. You can reload and modify the support file of the model objects whenever you like. Selecting the *Edit sup-file* item of the *Objects | Model* menu pops-up the well known *Edit Object* dialog box with the model object controls and the user is allowed to customize data and node values, icon and help text of the object.

Selecting *MODELS | Files (sup/mod)...* in the component selection menu performs an *Open Model* dialog box where you can choose a model support file. If you select the file FLASH_1.SUP the icon of the new model appears immediately in the circuit window and it can be connected with other object in normal way.

The *MODEL:FLASH_1* dialog box of the new model object has an additional input section *Models* besides the standard *DATA* and *NODES* attributes as shown in Fig. 5.37. This new section has two input fields: *Model file* for locating the model description file and a *Use As* field for specification of the *model_name* in the USE model AS model_name statement of MODELS.

The input and output interface for MODELS objects, the use of the model and interfacing it with the rest of the circuit are handled automatically by ATPDraw. The model description is written directly in the ATP input file. Blank lines are removed when inserting the .mod file. The general structure of the MODELS section in an .atp input file is shown below:

```
MODELS
/MODELS
INPUT
  IX0001 {v(CR30A )}
  IX0002 {v(CR20A )}
  IX0003 {i(CRZ2A )}
OUTPUT
  GAPA
-----------------------------------------
MODEL FLASH_1
...
Description of the model.
Complete copy of the
FLASH_1.MOD is pasted here.
...
ENDMODEL
-----------------------------------------
USE FLASH_1 AS FLASH_1
INPUT
  V1:= IX0001
  V2:= IX0002
  iczn:= IX0003
DATA
  Pset:=       1.
  Eset:=      .9.
  fdel:=       4.
  fdur:=      20.
OUTPUT
  GAPA:=trip
ENDUSE
ENDMODELS
```

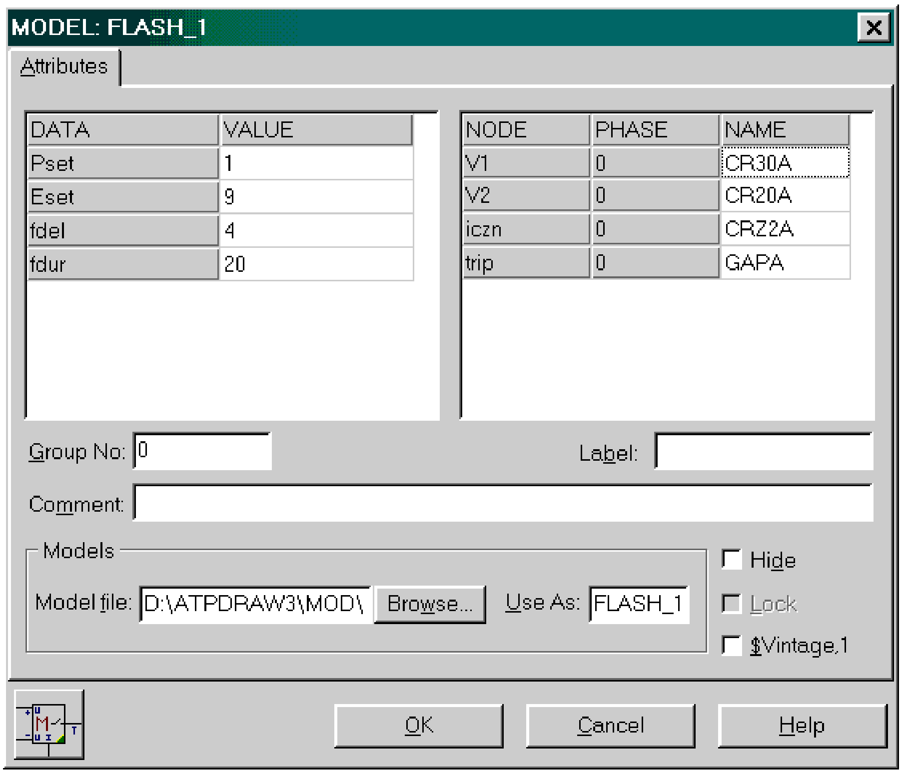
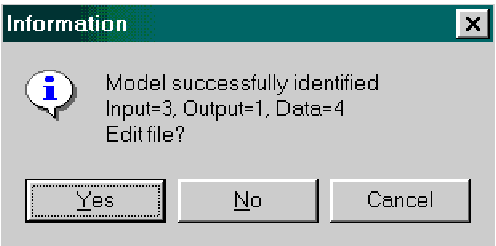


Fig. 5.37 - Component dialog box of the FLASH_1 model object.

5.5.2.2 Automatic operation

Version 3 of ATPDraw for Windows is capable of reading the `.mod` file directly, examine its `INPUT`, `OUTPUT` and `DATA` variables, and suggest a support file on the correct format. Either a comma or CR+LF is accepted as separator characters between variables by ATPDraw. Comment flags 'C' in column 1 and '--' are also handled. Maximum 12 input+output variables are allowed along with 36 data variables. Only single variables are allowed (not indexed).



The automatic sup-file creation procedure can be activated by selecting the *MODELS | Files (sup/mod)...* item of the component selection menu. When reading the mod-file ATPDraw performs a message box shown in Fig. 5.38.

Fig. 5.38 - Interpretation of the model file.

You are free to modify the default support file or accept the default icon and node positions, or *Cancel* the interpretation procedure. If you click on *Yes*, the *Edit Object* dialog box will appear where you can edit the icon, change node positions, set new default values and *Kind* (= I/O Type for current/voltage etc. input). If you select *No*, the default ATPDraw object is drawn in the circuit window immediately (Fig. 5.39). Input nodes are positioned on the left side of the icon and the outputs on the right. In both cases, a support file is automatically created. This file can later be modified under *Objects | Model | Edit sup-file.*

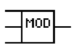
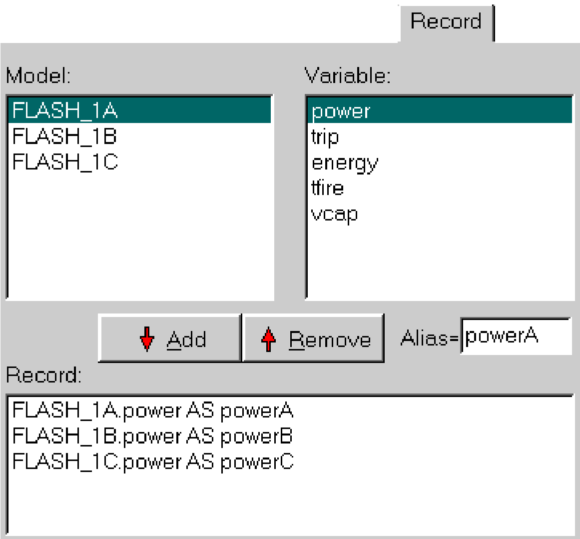


Fig. 5.39 - Default model object (`FLASH_1.SUP`)

5.5.3 Recording internal MODELS variables

ATPDraw supports the `RECORD` feature of MODELS to record any internal variable of a model object in the `.pl4` output. This option is found under the *ATP* | *Settings / Record* menu. All MODELS objects in the active circuit window are listed with their `USE AS` name in the list box under *Model* as shown in Fig. 4.19. When you select a model in this field, variables (declared under `VAR` section of the `.mod` file) are listed under *Variable*. Each variable has a default alias name that appears in the *Alias* field, but it can be modified according to the user need.



To record a variable, click on the *Add* button. The alias name can be changed later by selecting an item in the *Record* list box and type in a new name. The record request can be removed by selecting the item and click on *Remove*.

This record list is stored in the circuit file, but it does not follow the circuit when the circuit is copied to the clipboard or the export group option is used.

Fig. 5.40 - Record of model variables.

5.6 BCTRAN support in ATPDraw

ATPDraw provides a user-friendly interface for the BCTRAN transformer matrix modeling, to represent single and three-phase, two and three winding transformers. After the user has entered the open circuit and short circuit factory test data, the ATPDraw calls ATP and executes a BCTRAN supporting routine run. Finally, ATPDraw includes the punch-file into the ATP-file. The windings can be Y, D or Auto coupled with support of all possible phase shifts. The nonlinear magnetization branch can optionally be added externally.

Fig. 5.41 shows the *BCTRAN* dialog box, which appears when the user selects *BCTRAN* under *Transformers* of the component selection menu. Under *Structure*, the user specifies the number of phases, the number of windings, the type of core (not supported yet, except for single phase cores, triplex and three-phase shell type), and the test frequency. The dialog box format adapts the number of windings and phases. The user can also request the inverse L matrix as output by checking *AR output*. An *Auto-add nonlinearities* button appears when an external magnetizing branch is requested.

Under *Ratings* the line-voltage, rated power, and type of coupling must be specified. Supported winding *Connections* are: A (auto-transformer), Y (wye) and D (delta). The *Phase shift* menu adapts these settings with all types of phase shifts supported. If the connection is A or Y, the rated voltage is automatically divided by $\sqrt{3}$ to get the winding voltage `VRAT`.

Under *Factory test*, the user can choose either the *Open circuit* test or the *Short circuit* test.

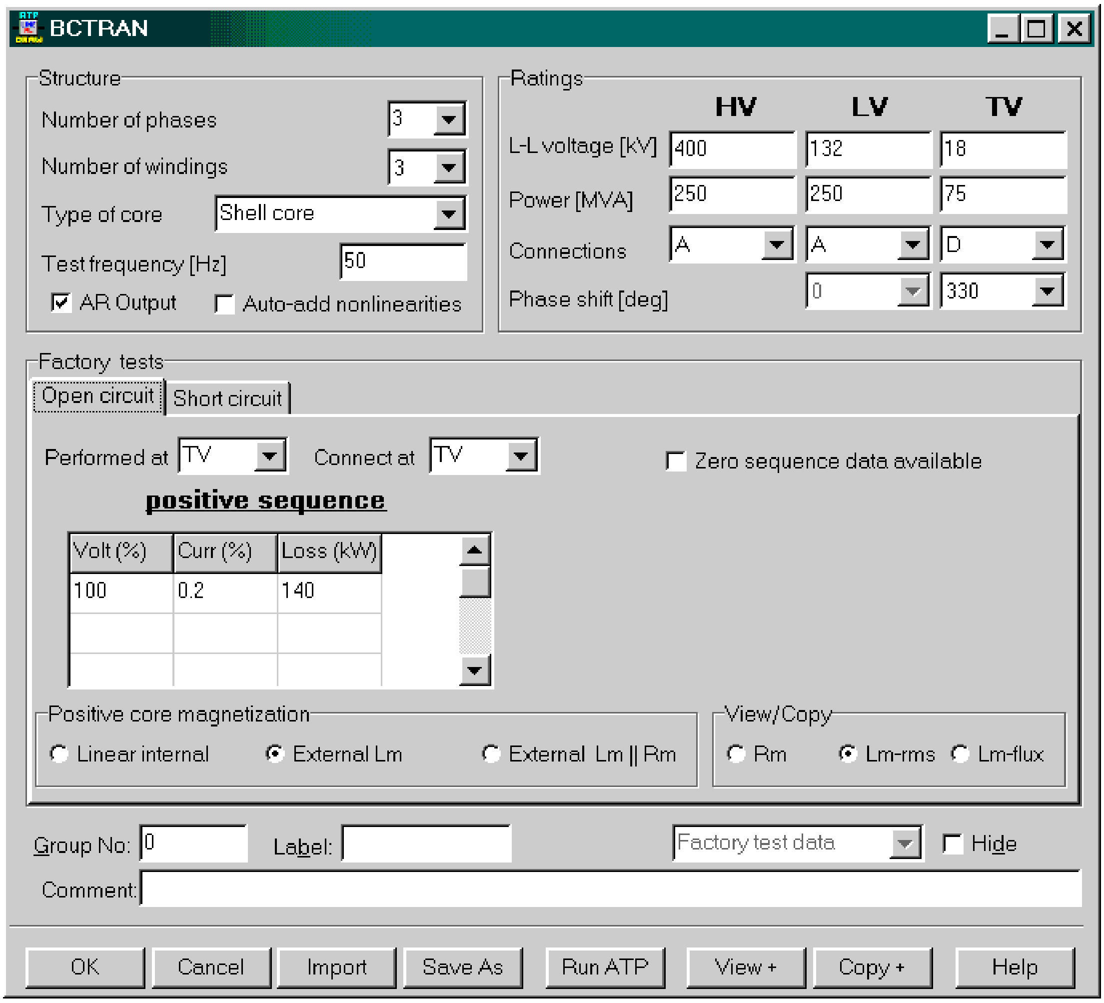


Fig. 5.41 - The BCTRAN dialog box.

Under the *Open circuit* tab the user can specify where the factory test has been performed and where to connect the excitation branch. In case of a three winding transformer one can choose between the HV, LV, and the TV winding. Normally the lowest voltage is preferred, but stability problems for delta-connected nonlinear inductances could require the lowest Y-connected winding to be used. Up to 6 points on the magnetizing curve can be specified. The excitation voltage and current must be specified in % and the losses in kW. With reference to the ATP Rule Book, the values at 100 % voltage is used directly as IEXPOS=Curr [%] and LEXPOS=Loss [kW]. One exception is if *External Lm* is chosen under *Positive core magnetization*. In this case only the resistive current is specified resulting in IEXPOS=*Loss*/(10 · SPOS), where SPOS is the *Power [MVA]* value specified under *Ratings* of the winding where the test has been performed. If zero-sequence open circuit test data are also available, the user can similarly specify them to the right. The values for other voltages than 100 % can be used to define a nonlinear magnetizing inductance/resistance. This is set under *Positive core magnetization:*

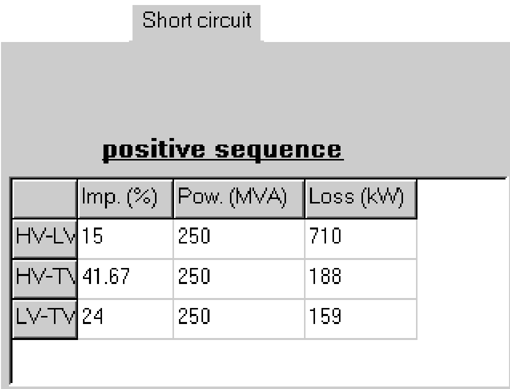
a) Specifying *Linear internal* will result in a linear core representation based on the 100 % voltage values.
b) Specifying *External Lm//Rm* the magnetizing branch will be omitted in the BCTRAN calculation and the program assumes that the user will add these components as external objects to the model.

c) Specifying *External Lm* will result in calculation of a nonlinear magnetizing inductance first as an I_{rms}-U_{rms} characteristic, then automatically transformed to a current-fluxlinked characteristic (by means of an internal SATURA-like routine). The current in the magnetizing inductance is calculated as

$$I_{rms}[A] = \sqrt{(10 \cdot Curr[\%] \cdot SPOS[MVA]/3)^2 - (Loss[kW]/3)^2} \; / V_{ref}[kV]$$

where V_{ref} is actual rated voltage specified under *Ratings*, divided by $\sqrt{3}$ for Y- and Auto-connected transformers.

The user can choose to *Auto-add nonlinearities* under *Structure* and in this case the magnetizing inductance is automatically added to the final ATP-file as a Type-98 inductance. ATPDraw connects the inductances in Y or D dependent on the selected connection for actual winding for a 3-phase transformer. In this case, the user has no control on the initial state of the inductor(s). If more control is needed (for instance to calculate the fluxlinked or set initial conditions) *Auto-add nonlinearities* should not be checked. The user is free to create separate nonlinear inductances, however. The *Copy+* button at the bottom of the dialog box allows the user to copy the calculated nonlinear characteristic to an external nonlinearity. What to copy is selected under *View/Copy*. To copy the fluxlinked-current characteristic used in Type-93 and Type-98 inductances *Lm-flux* should be selected.



	Imp. (%)	Pow. (MVA)	Loss (kW)
HV-LV	15	250	710
HV-TV	41.67	250	188
LV-TV	24	250	159

The *Short circuit* data can be specified as shown in Fig. 5.42. With reference to the ATP Rule Book; *Imp [%]* is equal to `ZPOS`, *Pow. [MVA]* is equal to `SPOS`, and *Loss [kW]* is equal to `P`. These three values are specified for all the windings. If zero-sequence short circuit factory test data are also available, the user can similarly specify them to the right of the positive sequence values after selecting the *Zero sequence data available* check box.

Fig. 5.42 - Short circuit factory test data.

If Auto-transformer is selected for the primary and secondary winding (HV-LV) the impedances must be re-calculated according to Eq. 6.45, 6.46, 6.50 of the EMTP Theory Book [5]. This task is performed by ATPDraw and the values Z^*_{H-L}, Z^*_{L-T}, and Z^*_{H-T} are written to the BCTRAN-file automatically.

$$z^*_{H-L} = z_{H-L}\left(\frac{V_H}{V_H - V_L}\right)^2, \quad z^*_{L-T} = z_{L-T}\ , \quad z^*_{H-T} = z_{H-L}\frac{V_H \cdot V_L}{(V_H - V_L)^2} + z_{H-T}\frac{V_H}{V_H - V_L} - z_{L-T}\frac{V_L}{V_H - V_L}$$

where Z_{L-H}, Z_{L-T}, and Z_{H-T} are the short-circuit impedances *Imp. [%]* referenced to a common *Pow.[MVA]* base.

When the user clicks on *OK* the data structure is stored in a binary disk file with extension *.bct* and stored in the `/BCT` folder. This BCT-file is stored in the ATPDraw project file just like LCC-files for lines/cables. Then the user is offered to generate a BCTRAN-file and run ATP. This is really optional, since often a new BCTRAN-file will be required anyway during the final ATP-file generation. Trying to run ATP is a good practice however, since this will quickly warn the user about possible problems. The button *Run ATP* requests an ATP execution without leaving the dialog box. If the BCTRAN-file is correct, a punch-file will be created. This file is directly included in the final ATP-file and there is no conversion to a library file as for lines/cables. This means in practice that a new BCTRAN-file will be created and ATP executed automatically (when creating the final ATP-file) each times the transformer's node names change.

There is also an *Import* button available to import existing BCT-files. The user can also store the BCT-file with a different name (*Save As*), which is useful when copying BCTRAN-objects. The *View+* and *Copy+* buttons are for the nonlinear characteristic. *Copy+* transfers the selected characteristic to the Windows clipboard in text format with 16 characters fixed columns (the first column is the current). *View+* displays the nonlinear characteristic in a standard *View Nonlin* window. The *Help* button at the lower right corner of the dialog box displays the help file associated with the BCTRAN object. This help text briefly describes the meaning of input data values.

```
1. Excitation test data
Specified under Factory test/Open circuit.
The data required by BCTRAN are:
FREQ   = Test frequency under Structure
IEXPOS = Curr for the 100% voltage value in Open circuit, Positive sequence.
       = Loss for the 100% voltage value divided by 10*SPOS when External Lm
         requested.
SPOS   = Power under Ratings for winding specified under Performed at.
LEXPOS = Loss for the 100% voltage value in Open circuit, Positive sequence.
IEXZERO= Curr for the 100% voltage value in Open circuit, Zero sequence.
SZERO  = Power under Ratings for winding specified under Performed at.
LEXZERO= Loss for the 100% voltage value in Open circuit, Zero sequence.
```

The above input values can be derived from the factory test data as shown next:

```
IEXPOS= Iex*V*100/SPOS for single phase,
IEXPOS= Iex*√3*V*100/SPOS for 3-phase
  where Iex [kA] = excitation current,
        V [kV]   = excitation voltage.
        SPOS[MVA]= power base
IEXZERO= 0 for single phase
IEXZERO= 1/3*Iexh*√3*V*100/SZERO for 3-phase
  where Iexh [kA]= zero-sequence excitation current,
        SPOS[MVA]= power base (normally equal to SPOS)
    Y-connected windings  (typical values):
     3-leg core type: IEXZERO= IEXPOS
     5-leg core type: IEXZERO= 4*IEXPOS
```

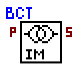
```
2. Winding cards
Specified under Ratings. The data required by BCTRAN are:
VRAT   = L-L voltage [kV] for D-connection or single phase transformers
         L-L voltage [kV] divided by √3 for A (Auto) and Y connections.
         3-phase only.
BUS1-   = The present node names of the transformer component in ATPDraw
BUS6      taking the connection and Phase shift [deg] into account.
         Renaming the nodes will require a new BCTRAN execution performed
         automatically upon ATP|Run ATP or Make File.
```

```
3. Short circuit test data
Specified under Factory test / Short circuit. The data required by BCTRAN are:
Pij    = Loss (kW) under Short circuit, Positive sequence
ZPOSij = Imp (%) under Short circuit, Positive sequence
SPOS   = Pow (MVA) under Short circuit, Positive sequence
ZZEROij= Imp (%) under Short circuit, Zero sequence
SZERO  = Pow (MVA) under Short circuit, Zero sequence
```

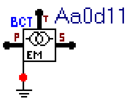
The short circuit input data can be derived from the factory test reports, as shown next:

```
ZPOSij=  Usi/Isi*SPOS/Vri^2*100 for single phase,
ZPOSij=  Ush/√3*Ish)*SPOS/(Vri²)*100 for 3-phase
     where
         Usi [kV] = short-circuit voltage at winding i
         Isi [kA] = nominal current at winding i
         SPOS[MVA]= power base
         Vri [kV] = rated line voltage at winding i
ZZEROij= 0 for single phase
ZZEROij= Ush/Ish*SZERO/(Vri²)*300 for 3-phase
     where
         SZERO[MVA]= power base
         Zero-sequence tests must be performed with open Delta-windings.
```

The BCTRAN component is found under *Transformers* | *BCTRAN* in the component selection menu and it can be edited and connected to the main circuit as any other component.



The default icon is shown left. The nodes appear automatically dependent on the selected connection and number of windings.



The data specified in Fig. 5.41 will result in an icon at left with 3 three-phase terminals and one single-phase neutral point common to the primary and secondary autotransformer windings. The label shows the transformer connection.

5.7 Non-standard component dialog boxes

The component dialog box in which the user is allowed to change the object's attributes shows a considerable similarity nearly for all components: on the *Attributes* page the components data can be specified, on the optional *Characteristic* page you specify the input characteristic of non-linear components, while the node names in the top-right table is just for your information.

The following components deviate somewhat from the above description:

- General 3-phase transformer (GENTRAFO)
- Universal machine (UM_1, UM_3, UM_4, UM_6, UM_8)
- Statistical / Systematic switch (SW_STAT, SW_SYST)
- Harmonic source (HFS_SOUR)

5.7.1 Saturable 3-phase transformer

The component dialog box of this transformer model is shown in Fig. 5.43. This dialog box also has an *Attributes* and a *Characteristic* page, but the former is largely differs from the standard layout. The function of the *Group No*, *Label*, *Comment* and *Output* fields are the same as on any other component dialog boxes, the meaning of the other fields are given next. The pair I_o, F_o defines the magnetizing branch inductance at steady state. R_m is the resistance of the magnetizing branch representing the hysteresis and eddy current losses of the iron core. I_o, F_o, R_m may be left blank if the magnetizing branch is neglected in the simulation. Checking the *3-leg core* turns the transformer into a TRANSFORMER THREE PHASE type with high homopolar reluctance that can be specified in the appearing R_0-field. With the button *3-leg core* unchecked, the model is a saturable transformer with low homopolar reluctance (e.g. a 3-phase transformer with at least one delta winding).

Checking the *RMS* button enables specification of the saturation characteristic in rms values for current and voltage on the *Characteristic* page. A conversion to flux-current values is performed internally in ATPDraw. If the button is unchecked, normal flux-current values should be entered. The tertiary winding can be turned on or off by checking the *3-wind.* button. The nominal voltage of the transformer windings (!) must be given in [kV]. The short circuit inductances may be specified in [mH] if *Xopt.* parameter is 0 (default) on the *ATP | Settings / Simulation* page. Otherwise, the impedance is given in [Ω] at frequency *Xopt.*

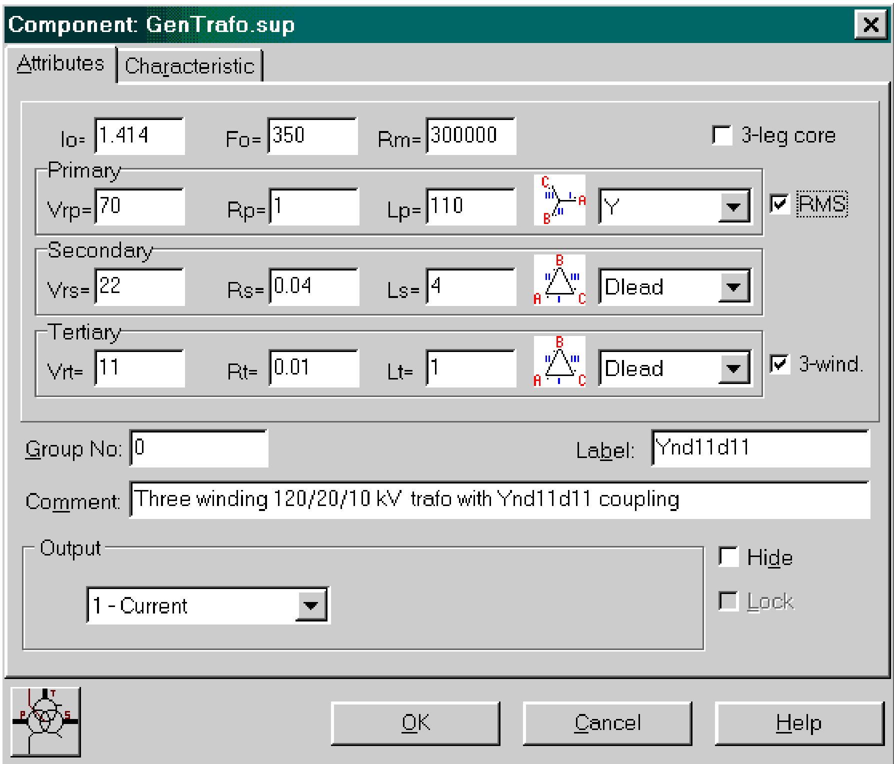
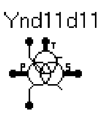


Fig. 5.43 - General saturable transformer dialog.

Four types of winding couplings are supported by this general transformer object: *Wye*, *Delta lead*, *Delta lag*, and *Y180*. Icons to the right of the short circuit impedances visualize the selected coupling. The letters *A*, *B*, and *C* are the phase terminals, while the Roman numbers *I*, *II*, and *III* denotes the leg number. The table below shows how to represent typical 2-winding transformers:

Yd1: Y (primary) + Dlag (secondary)	Dy1: Dlead (primary) + Y (secondary)
Yd5: Y180 (primary) + Dlead (secondary)	Dy5: Dlag (primary) + Y180 (secondary)
Yd7: Y180 (primary) + Dlag (secondary)	Dy7: Dlead (primary) + Y180 (secondary)
Yd11: Y (primary) + Dlead (secondary)	Dy11: Dlag (primary) + Y (secondary)
Yy0: Y (primary) + Y (secondary)	Yy6: Y (primary) + Y180 (secondary)

* Y/zig-zag, D/zig-zag trafos are out of scope at present.



The data specified in Fig. 5.43 will result in an icon at left with 3 three-phase terminals and 2 single phase nodes: one for the primary neutral and one for connection of the magnetization circuit with external saturation (optional).

The *Saturable 3-phase* object is found under *Transformers* in the component selection menu and it can be edited and connected to the main circuit as any other components.

The *Help* button at the lower right corner of the dialog box displays the help file associated with the GENTRAFO object. This help text briefly describes the meaning of input data values:

```
Data:      Io= Current [A] through magnetizing branch (MB) at steady state.
           Fo= Flux [Wb-turn] in MB at steady state.
               The pair Io, Fo defines the inductance in MB at steady state.
           Rm= Resistance in magnetizing branch in [ohm]. 5-leg core or 3-leg

               shell.
           R0= Reluctance of zero-sequence air-return path for flux. 3-leg
               core-type
           Rp= Resistance in primary winding in [ohm].
           Lp= Inductance in primary winding in [mH] if Xopt.=0
               Inductance in primary winding in [ohm] if Xopt.=power freq.
          Vrp= Rated voltage in [kV] primary winding.
           Rs= Resistance in secondary winding in [ohm].
           Ls= Inductance in secondary winding in [mH] if Xopt.=0
               Inductance in secondary winding in [ohm] if Xopt.=power freq.
          Vrs= Rated voltage in [kV] secodary winding.
           Rt= Resistance in tertiary winding in [ohm].
           Lt= Inductance in tertiary winding in [mH] if Xopt.=0
               Inductance in tertiary winding in [ohm] if Xopt.=power freq.
          Vrt= Rated voltage in [kV] tertiary winding.
          RMS= unchecked: Current/Flux characteristic must be entered.
               checked:   Irms/Urms characteristic must be entered.
                          ATPDRAW performs a SATURATION calculation.
 3-leg core = checked: 3-leg core type transformer assumed.
                        TRANSFORMER THREE PHASE
               unchecked: 5-leg or 3-leg shell type assumed. TRANSFORMER.
     3-wind.= turn on tertiary winding.

Points: It's possible to enter 9 points on the current/flux characteristic.
        The required menu is performed immedeately after the input menu.
        The points should be entered as increasingly larger values.
        The point (0,0) is not permitted (added internally in ATP).
```

5.7.2 Universal machines

Handling of electrical machines in version 3 of ATPDraw has been updated substantially to provide a user-friendly interface for most of the electrical machine modeling options in ATP. Supported Universal Machine (UM) types are:

- Synchronous machine (UM type 1)
- Induction machines (UM type 3 & 4)
- DC machine (UM type 8)
- Single-phase machine (UM type 6)

The component dialog box of the Universal Machine object is substantially differs to the standard dialog box layout, as shown in Fig. 5.44. In the UM component dialog box the user enters the machine data in five pages: *General, Magnet, Stator, Rotor, Init.* Several UM models are allowed with global specification of initialization method and interface. These *Global* options can be specified under *ATP | Setting / Switch/UM.*

On the *General* page data like stator coupling and the number of *d* and *q* axis coils are specified. On the *Magnet.* page the flux/inductance data with saturation are specified, while on the *Stator* and *Rotor* pages the coil data are given. *Init* page is for the initial condition settings.

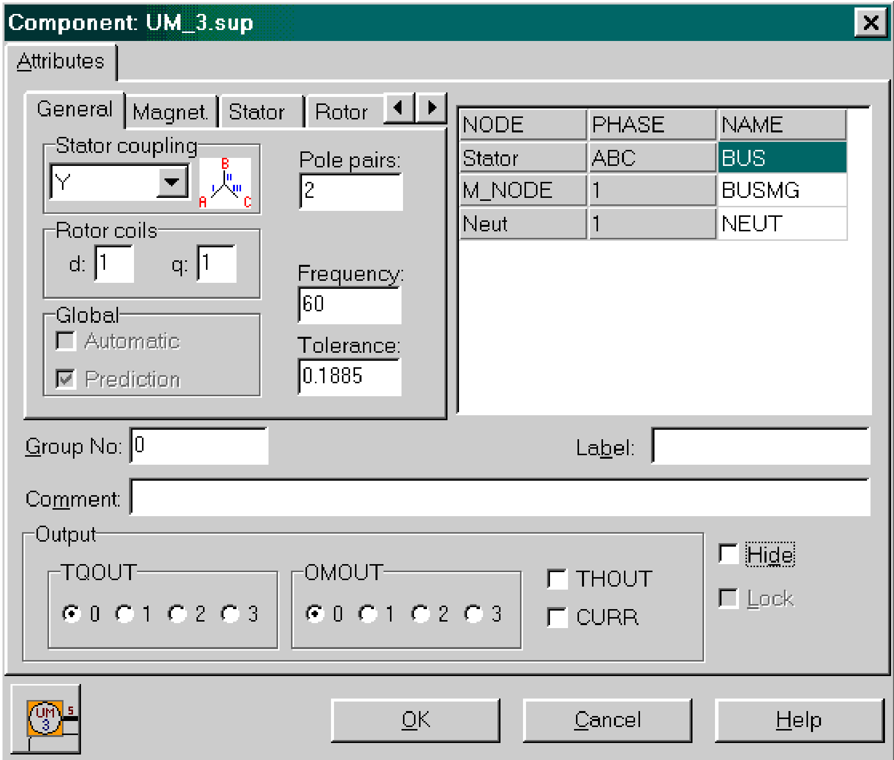


Fig. 5.44 - Universal machine input dialog.

The dialog boxes for all the universal machines are similar. The type 4 induction machine does not have the *Rotor coils* group, since this is locked to 3. None of the type 3 and 4 induction machine have the field node of course.

The single-phase machine (type 6) and the DC machine (type 8) do not have the *Stator coupling* group. For the type 6 machine the number of *d*-axis is locked to 1. Even if the number of rotor coils or excitation coils can be set to maximum 3, only the first *d*-axis coils will have external terminals for a type 1, 6, and 8 machine. The other coils will be short circuited. Rotor coils are short circuited in case of type 3 machine, while the type 4 machine has an external terminal for all its 3 coils.

Fig. 5.45 shows the various pages for universal machine data input. The buttons under the *Saturation* on the *Magnet.* page turns on/off the various saturation parameters for the d- and q-axis. This is equivalent to the parameter JSATD and JSATQ in the ATP data format. Selecting *symm* is equal to having JSATD=5 and JSATQ=0 (total saturation option for uniform air gap).

On the *Stator* page, you specify the Park transformed quantities for resistance and inductance for the armature winding. The number of coils on the *Rotor* page and on the *Init* page for manual initialization adapts the specification of the number of rotor coils. First the d-axis coils are listed then comes the q-axis coils.

The function of the *Group No*, *Label*, *Comment* fields are the same as on any other component dialog boxes. The *Help* button at the lower right corner of the dialog box displays the help file associated with the UM objects.

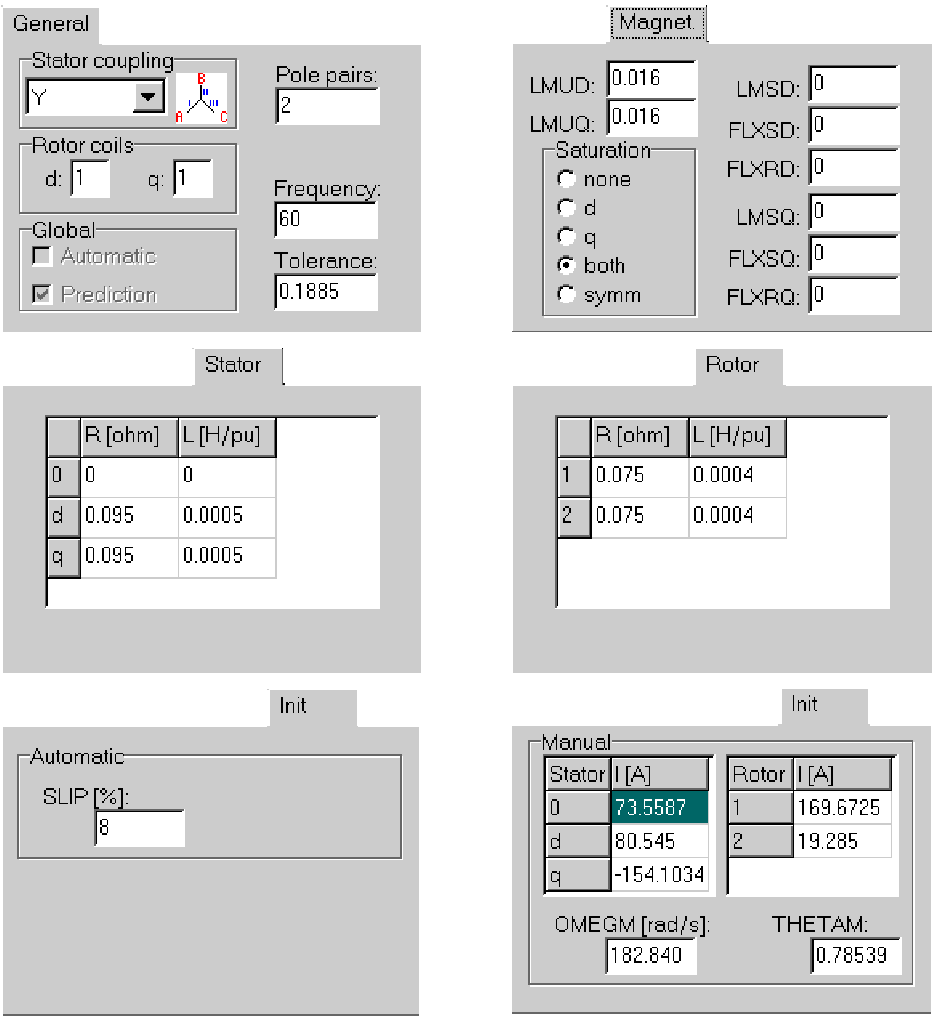


Fig. 5.45 - Data pages of the universal machines dialog box.

The *Help* text briefly describes the meaning of input data values and node names as the example shows next for UM type 1 (Synchronous machine):

```
  Data:
    General page:
        Pole pairs - Number of pole pairs
        Tolerance  - Rotor-speed iteration-convergence margin.
        Frequency  - Override steady state frequency.
        Stator coupling
          Select between Y, Dlead (AC, BA, CB) and Dlag (AB, BC, CA)
          Selecting Y turns neutral node Neut on.
        Rotor coils
          Specify the number of d- and q- axis rotor coils. Maximum total
          number is 3. Only terminals for 1st d-axis coil. The other coils
          are assumed short circuited.
        Global
          Visualization of mode of initialization and interface.
          Set under the main menu ATP|Settings/Switch/UM for each circuit.
    Stator page:
        Specify resistance and inductance in Park transformed
        quantities (d- q- and 0- system). All inductances in H or pu.
    Rotor page:
        The total number of coils are listed and given data on the Rotor
        page. First the d-axis coils then the q-axis coils are listed.
        Specify resistance and inductance for each coil. All the coils
        except the first is short circuited. All inductances in H or pu.
```

```
    Magnet. page:
        LMUD  - d-axis magnetization inductance.
        LMUQ  - q-axis magnetization inductance.
        Turn on/off the saturation.
        Symm. is equal saturation in both axis, specified only in d.
        LMSD  - d-axis saturated inductance.
        FLXSD - d-axis flux-linkage at the saturation knee point.
        FLXRD - d-axis residual flux-linkage (at zero current).
        LMSQ  - d-axis saturated inductance.
        FLXSQ - q-axis flux-linkage at the saturation knee point.
        FLXRQ - q-axis residual flux-linkage (at zero current).
        NB! All inductances in H or pu.
    Initial page:
        Initial conditions dependent on manual or automatic
        initialization is chosen under ATP|Settings/Switch/UM
        Automatic:
          AMPLUM - initial stator coil (phase) voltage [V].
          ANGLUM - angle of phase A stator voltage [deg].
        Manual:
          Specify stator current in the d- q- and 0-system
          Specify rotor current inn all coils
          OMEGM  - initial mechanical speed [mech rad/sec or unit]
          THETAM - initial pos of the rotor [elec rad]
    Output:
        TQOUT=1: air gap torque
             =2: 1 + d-axis common flux
             =3: 2 + d-axis magnetization current
        OMOUT=1: rotor shaft speed in [rad/sec]
             =2: 1 + q-axis common flux
             =3: 2 + q-axis magnetization current
        THOUT=checked: rotor position in [mech rad]
        CURR =checked: all physical coil currents
  Node:
    Stator - 3-phase armature output terminal.
    M_NODE - air-gap tourque node.
    FieldA - Pos. terminal of exitation rotor coil.
             (the other coils are grounded)
    FieldB - Neg. terminal of exitation rotor coil.
    BUSM   - torque-source node for automatic initialization.
    BUSF   - field-source node for automatic initialization.
    Neut   - Neutral point of Y-coupled stator coils.
```

The final section of the *Help* file describes the equivalent electrical network of the mechanical network for torque representation:

```
Shaft mass (moment of inertia) ↔ Capacitance          (1kg/m2 ↔ 1 Farad)
Shaft section (spring constant) ↔ Inverse inductance. (1 Nm/rad ↔ 1/Henry)
Shaft friction (viscous damping) ↔ Conductance.       (1 Nm/rad/s ↔ 1/ohm)
Angular speed ↔ Voltage        (1 rad/s ↔ 1 Volt)
Torque ↔ Current               (1 Nm ↔ 1 Amp)
Angle ↔ Charge                 (1 rad ↔ 1 Coulomb)
```

```
                                            L1
   __       __                    _____ ___oooo________________
  |  |     |  |                   |     |         |     |     |
  |J1| K1  |J2|  --> T           _|_    <        _|_    <     |
==|  |=====|  |====      ====>   ___C1   >R1     ___C2   >R2  O I
  |  |     |  |                   |     <         |     <     |
  |__|     |__|                   |     |         |     |     |
   D1       D2
          C1=J1, C2=J2, R1=1/D1, R2=1/D2, L1=1/K1, I=T
```

5.7.3 Statistic/systematic switch

Handling of statistic/systematic switches in version 3 of ATPDraw has been made more general by introducing the independent/master/slave concept. The component dialog boxes of the statistical switches slightly differs however from the standard switch dialog box layout as shown in Fig. 5.46.

The user can select the *Switch type* in a combo box out of the supported options: *Independent*, *Master* or *Slave*. This will also enable the possible input fields and change the number of nodes (note that slave switch has 4 nodes). The *Distribution* for the statistical switch takes into account the specification of the IDIST parameter on the miscellaneous switch card (*ATP* | *Settings / Switch/UM*). Selecting IDIST=1 will disable the *Distribution* group and force *Uniform* distribution. The *Open/Close* radio buttons select if the switch closes or opens with *Ie* as current margin for opening switches. The number of ATP simulations is set by the miscellaneous switch parameter *Num*. on the *ATP* | *Settings / Switch/UM* page. This value influences the 1st misc. data parameter NENERG of ATP. ATPDraw sets the correct sign of NENERG: i.e. > 0 for statistic or < 0 for systematic switch studies. The function of the *Group No*, *Label*, *Comment* and *Output* fields are the same as for any other standard components.

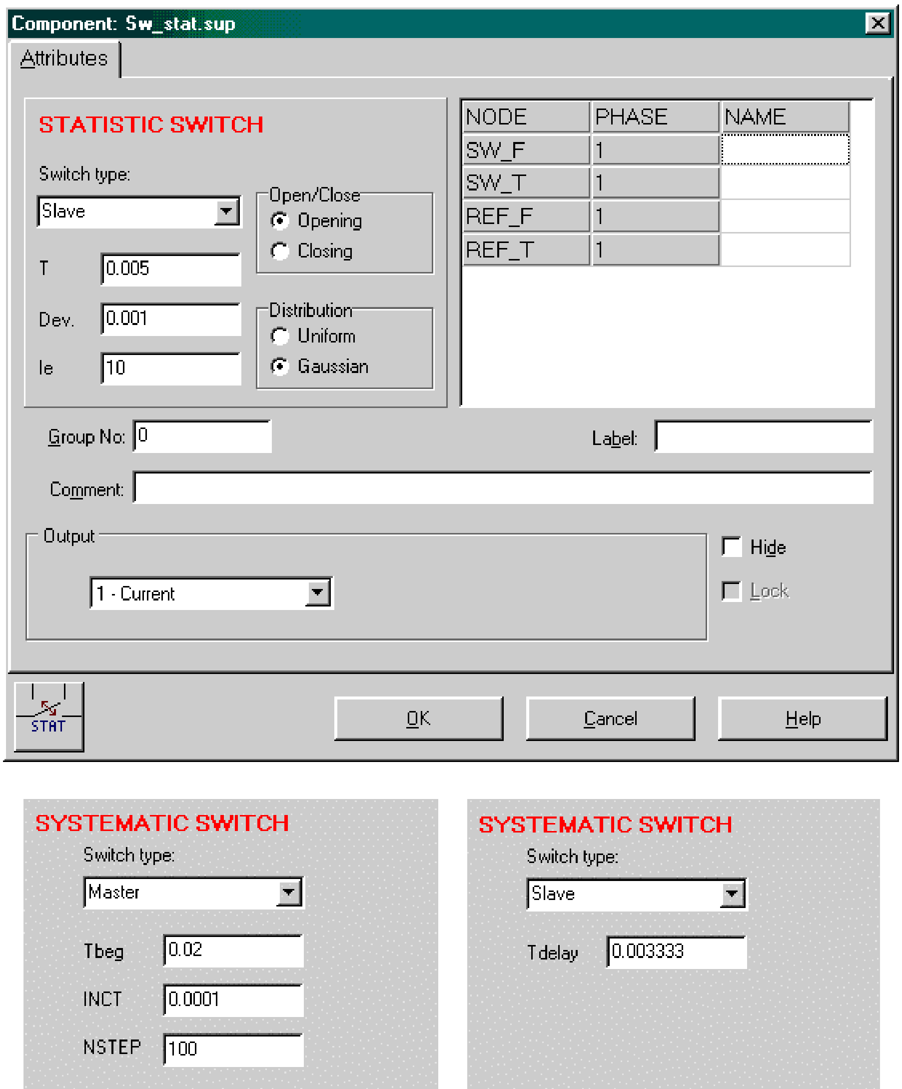


Fig. 5.46 - Dialog box of the statistic switch (top) and data windows of the systematic switch.

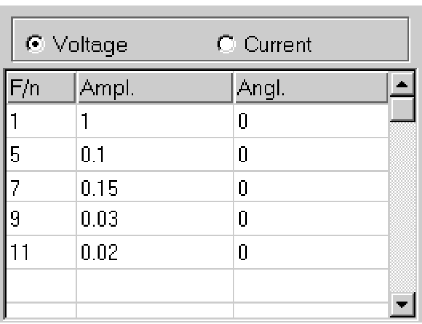
The *Help* button at the lower right corner of the dialog box displays the help file associated with the object. This text briefly describes the meaning of input data values and node names as shown below:

```
SW_STAT -  Statistic switch.
   Distribution: Select uniform or gaussian distribution.
             If IDIST=1 under ATP|Settings/Switch/UM only uniform is possible.
   Open/Close: Select if the switch closes or opens.
         Current margin available for opening switch.
       T   = Average switch opening or closing time in [sec.]
             For Slave switches this is the average delay.
       Dev.= Standard deviation in [sec.].
             For Slave switches this is the deviation of the delay.
       Ie  = Switch opens at a time T>Tmean and the current through
             the switch is less than Ie.
   Switch type:
         INDEPENDENT: Two nodes
         MASTER     : Two nodes. 'TARGET' punched. Only one is allowed.
         SLAVE      : Four nodes. Specify node names of MASTER switch.
         The icon and nodes of the objects adapt the switch type setting.
   Node:  SW_F= Start node of switch.
          SW_T= End node of switch.
         REF_F= Start node of the MASTER switch
         REF_T= End node of the MASTER switch

SW_SYST -  Systematic switch.
       Tbeg  = When ITEST=1 (ATP|Settings/Switch/UM)
       Tmid  = When ITEST=0 (ATP|Settings/Switch/UM)
       Tdelay= For SLAVE switches. If ITEST=0 : T=Tmid.
        INCT = Size of time increment in [sec.].
       NSTEP = Number of time increments.
   Switch type:
         INDEPENDENT: Two nodes
         MASTER     : Two nodes. 'TARGET' punched.
         SLAVE      : Four nodes. Specify node names of MASTER switch.
         The icon and nodes of the objects adapt the switch type setting.
Node :    SW_F = Start node of switch.
          SW_T = End node of switch.
         REF_F = Start node of the MASTER switch
         REF_T = End node of the MASTER switch
```

5.7.4 Harmonic source

The component dialog box of the *Harmonic source* that is used in HFS studies deviates somewhat from the standard source dialog box layout as shown in Fig. 4.71.



Selecting HFS under *ATP* | *Settings / Simulation the* ATP will run the case so many times as specified in the *Harmonic source* component dialog box. The frequency of the harmonic source will for each ATP run be incremented. The user selects the source type by the *Voltage* or *Current* radio button. In the example shown here, the data case will run 5 times because the *F/n* column has 5 harmonics entered.

Fig. 5.47 - Harmonic source dialog box.

The base frequency here is the *Freq.* value specified under *ATP* | *Settings / Simulation*. The amplitude and angle of the *F/n' th* harmonic source is given in columns *Ampl.* and *Angl.*

5.8 Creating new circuit objects in ATPDraw

The user specified objects (USP) are either customized standard objects or objects created for the use of $INCLUDE and DATA BASE MODULARIZATION feature of ATP-EMTP. The *Objects | User Specified | New sup-file* menu enables the user to create a new support file for such a user specified object or customize data/node properties and the icon or the help text of an existing one. The number of nodes and data specified in the *Edit Object* dialog box for USP objects must be in line with the `ARG` and `NUM` declarations in the header section of the Data Base Module (DBM) file. The number of data must be in the range of 0 to 36, and the number of nodes in the range of 0 to 12. The USP support files are normally located in the `/USP` folder.

Two new circuit objects will be created in this section: a 6-pulse controlled thyristor-rectifier bridge that is used as building block for simulating a 12-pulse HVDC station (`Exa_6.adp`) in section 6.4 of the Application Manual, and a generator step-up transformer model with winding capacitances and hysteretic core magnetism included. The latter object is used in a transformer inrush current study (`Exa_11.adp`) in section 6.8.2 of the Application Manual.

5.8.1 Creating a 6-phase rectifier bridge

The Data Base Module (DBM) file shown next describes a 6-pulse thyristor rectifier bridge (based on exercise 54 in [2]). The process of creating a DBM-file is certainly the most difficult part of adding new circuit objects to ATPDraw. The input file to the DBM supporting routine of ATP begins with a header declaration followed by the circuit description. The ATP Rule Book [3] chapter XIX-F explains in detail how to create such a file. The output punch-file of the DBM supporting routine can actually be considered as an external library file which is included to the ATP simulation at run time via a `$INCLUDE` call.

```
BEGIN NEW DATA CASE --NOSORT--
DATA BASE MODULE
$ERASE
ARG,U____,POS___,NEG___,REFPOS,REFNEG,ANGLE_,Rsnub_,Csnub_
NUM,ANGLE_,Rsnub_,Csnub_
DUM,PULS1_,PULS2_,PULS3_,PULS4_,PULS5_,PULS6_,MID1__,MID2__,MID3__
DUM,GATE1_,GATE2_,GATE3_,GATE4_,GATE5_,GATE6_,VAC___,RAMP1_,COMP1_
DUM,DCMP1_,DLY60D
/TACS
11DLY60D  .002777778
90REFPOS
90REFNEG
98VAC___  =REFPOS-REFNEG
98RAMP1_58+UNITY                                  120.00   0.0   1.0VAC___
98COMP1_  =(RAMP1_-ANGLE_/180) .AND. UNITY
98DCMP1_54+COMP1_                                       5.0E-3
98PULS1_  = .NOT. DCMP1_ .AND. COMP1_
98PULS2_54+PULS1_                                                   DLY60D
98PULS3_54+PULS2_                                                   DLY60D
98PULS4_54+PULS3_                                                   DLY60D
98PULS5_54+PULS4_                                                   DLY60D
98PULS6_54+PULS5_                                                   DLY60D
98GATE1_  = PULS1_ .OR. PULS2_
98GATE2_  = PULS2_ .OR. PULS3_
98GATE3_  = PULS3_ .OR. PULS4_
98GATE4_  = PULS4_ .OR. PULS5_
98GATE5_  = PULS5_ .OR. PULS6_
98GATE6_  = PULS6_ .OR. PULS1_
/BRANCH
$VINTAGE,0
  POS___U____A            Rsnub_      Csnub_
  POS___U____BPOS___U____A
```

```
  POS___U____CPOS___U____A
  U____ANEG___POS___U____A
  U____BNEG___POS___U____A
  U____CNEG___POS___U____A
/SWITCH
11U____APOS___                                                      GATE1_
11U____BPOS___                                                      GATE3_
11U____CPOS___                                                      GATE5_
11NEG___U____A                                                      GATE4_
11NEG___U____B                                                      GATE6_
11NEG___U____C                                                      GATE2_
BEGIN NEW DATA CASE
C                     <= "C" in the 1st column is mandatory here!
$PUNCH
BEGIN NEW DATA CASE
BLANK
```

The header section of the DBM-file starts with an `ARG` declaration after the special ATP request card `DATA BASE MODULE.` It's function is to specify the external variables (numerical + node names) and the sequence of arguments for the `$INCLUDE` procedure. The `NUM` card tells what arguments are numerical. `DUM` card lists the dummy or local variables, which are typically internal node names. ATP gives dummy nodes a unique name and thus let you use the same DBM-file several times in a data case avoiding node name conflicts. The rest of the DBM-file describes the rectifier bridge in a normal ATP data structure, except that sorting cards `/TACS`, `/BRANCH`, `/SWITCH` etc., are used in a special way. Sorting cards are required, but no `BLANK TACS`, `BLANK BRANCH`, etc. indicators are needed.

The 3-phase thyristor bridge has a 3-phase AC input node and two single phase DC output nodes. The firing angle is taken as input data and the snubber parameters are also practical to consider as numerical input to the model. The model created here accepts external reference signals for the zero crossing detector (alternatively the DBM module file could have detected its own AC input), thus the new USP object will have 5 nodes and 3 data:

U_____ :	The AC 3-phase node
POS___:	The positive DC node
NEG___:	The negative DC node
REFPOS:	Positive reference node.
REFNEG:	Negative reference node.
ANGLE_:	The firing angle of the thyristors.
Rsnub_:	The resistance in the snubber circuits.
Csnub_:	The capacitance in the snubber circuits.

Note the importance of the number of characters used for each parameter. The `U_____` parameter has only 5 characters, because it is a 3-phase node and the extensions *A*, *B* and *C* are added inside the DBM-file. Underscore characters '_ ' has been used to force the variables to occupy the 6 characters space for node names and 6 columns (`$VINTAGE, 0`) for the snubber data. Running the DBM-file through ATP will produce a `.pch` punch file shown below:

```
KARD  3  4  5  6  6  6  7  7  8  8  8  9  9 10 10 10 11 11 11 12 12 12 13 13 13
     14 14 14 15 15 15 16 16 16 17 17 17 18 18 18 19 19 19 20 20 20 21 21 21 24
     24 24 24 25 25 25 25 26 26 26 26 27 27 27 27 28 28 28 28 29 29 29 29 31 31
     31 32 32 32 33 33 33 34 34 34 35 35 35 36 36 36
KARG-20  4  5  4  5-16-16-17  6-17-18-18-19 -1-18-19 -1 -2-20 -2 -3-20 -3 -4-20
     -4 -5-20 -5 -6-20 -1 -2-10 -2 -3-11 -3 -4-12 -4 -5-13 -5 -6-14 -1 -6-15  1
      2  7  8  1  1  2  2  1  1  2  2  1  1  2  3  1  1  2  3  1  1  2  3  1  2
    -10  1  2-12  1  2-14  1  3-13  1  3-15  1  3-11
KBEG  3  3  3 12 19  3 69  3 20 13  3 12  3  3 32 19 12  3 69 12  3 69 12  3 69
```

```
     12  3 69 12  3 69 13 25  3 13 25  3 13 25  3 13 25  3 13 25  3 25 13  3  9
      3 27 39  9 21  3 15  9 21  3 15  3 21 15  9  3 21 15  9  3 21 15  9  3  9
     65  3  9 65  3  9 65  9  3 65  9  3 65  9  3 65
KEND  8  8  8 17 24  8 74  8 25 18  8 17  8  8 37 24 17  8 74 17  8 74 17  8 74
     17  8 74 17  8 74 18 30  8 18 30  8 18 30  8 18 30  8 18 30  8 30 18  8 13
      8 32 44 13 25  8 20 13 25  8 20  7 25 20 14  7 25 20 14  7 25 20 14  7 14
     70  7 14 70  7 14 70 13  8 70 13  8 70 13  8 70
KTEX  1  1  1  1  1  1  1  1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
      1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
      1  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
      1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
$ERASE
/TACS
11DLY60D  .002777778
90REFPOS
90REFNEG
98VAC___  =REFPOS-REFNEG
98RAMP1_58+UNITY                                  120.00   0.0   1.0VAC___
98COMP1_  =(RAMP1_-ANGLE_/180) .AND. UNITY
98DCMP1_54+COMP1_                                       5.0E-3
98PULS1_  = .NOT. DCMP1_ .AND. COMP1_
98PULS2_54+PULS1_                                                   DLY60D
98PULS3_54+PULS2_                                                   DLY60D
98PULS4_54+PULS3_                                                   DLY60D
98PULS5_54+PULS4_                                                   DLY60D
98PULS6_54+PULS5_                                                   DLY60D
98GATE1_  = PULS1_ .OR. PULS2_
98GATE2_  = PULS2_ .OR. PULS3_
98GATE3_  = PULS3_ .OR. PULS4_
98GATE4_  = PULS4_ .OR. PULS5_
98GATE5_  = PULS5_ .OR. PULS6_
98GATE6_  = PULS6_ .OR. PULS1_
/BRANCH
$VINTAGE,0
  POS___U____A              Rsnub_      Csnub_
  POS___U____BPOS___U____A
  POS___U____CPOS___U____A
  U____ANEG___POS___U____A
  U____BNEG___POS___U____A
  U____CNEG___POS___U____A
/SWITCH
11U____APOS___                                                  GATE1_
11U____BPOS___                                                  GATE3_
11U____CPOS___                                                  GATE5_
11NEG___U____A                                                  GATE4_
11NEG___U____B                                                  GATE6_
11NEG___U____C                                                  GATE2_
$EOF  User-supplied header cards follow.         31-May-02  15.46.06
ARG,U_____,POS___,NEG___,REFPOS,REFNEG,ANGLE_,Rsnub_,Csnub_
NUM,ANGLE_,Rsnub_,Csnub_
DUM,PULS1_,PULS2_,PULS3_,PULS4_,PULS5_,PULS6_,MID1__,MID2__,MID3__
DUM,GATE1_,GATE2_,GATE3_,GATE4_,GATE5_,GATE6_,VAC___,RAMP1_,COMP1_
DUM,DCMP1_,DLY60D
```

This file is very similar to the DBM input file, but with a different header and with the original DBM-file header given at the bottom instead. This file is ready to `$INCLUDE` into an ATP input file by ATPDraw. The file must be given a name and extension `.LIB` and stored in the default `\USP` directory. The name `HVDC_6.LIB` is used here as an example.

When the punch-file from the DBM-file has been created, the next step is to create a support file for the new `HVDC_6` object in the the *Objects | User Specified* menu. The process of creating a new object consists of two steps: create parameter support and create the icon.

First select the *New sup-file* in the popup menu. A notebook-style dialog box shown in Fig. 5.48 appears where you specify the number of data and nodes. The number of arguments on the `NUM` card(s) of the DBM-file tells you the *Number of data*, which is 3 in this example. The number of arguments on the `ARG` card(s) minus number of arguments on the `NUM` card(s) specifies the total *Number of nodes*, which is 5 in this example.

On the *Data* tab, you specify the names of the data parameters, number of digits (it must be less or equal the space used in the DBM-file, which is 6 in this case) a default value, and the *Min/Max* values. The name of data need not be equal to the names used in the DBM punch-file, but the sequence of data must be the same as on the `ARG` and `NUM` card(s). After specifying data properties, click on the *Node* tab and set the node control parameters as shown in Fig. 5.48. The *Name* of nodes, the number of *Phases (1/3)* and the node position on the icon border (1-12) are to be given here. Codes for the available node positions are shown in the icon at right. *Kind* is not used here. It must be left unity (default) for all nodes. The name of the nodes need not be identical with the names used in the DBM-file, but the node sequence must be the same as on the `ARG` card.

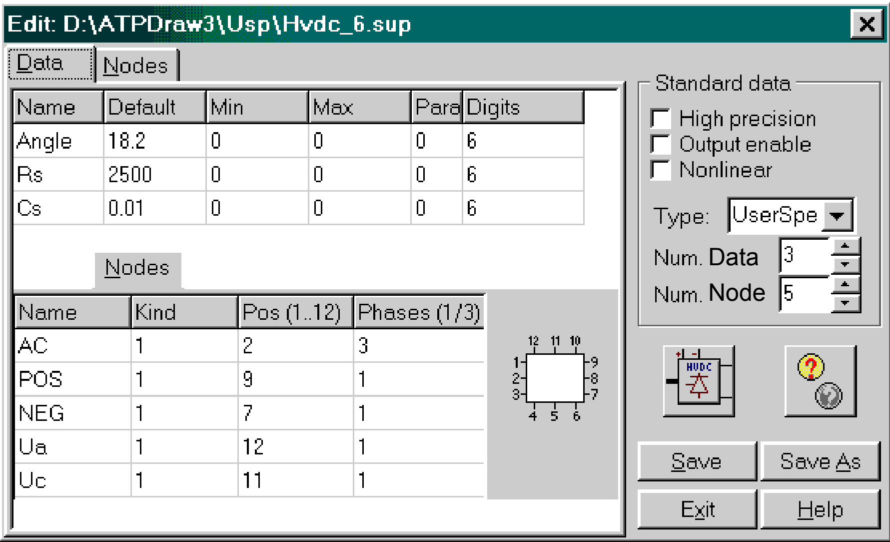


Fig. 5.48 - Properties of the new HVDC_6 object.

ATPDraw writes all three names of a 3-phase node in the `$INCLUDE` statement. In this example only the core name of the 3-phase node is expected on the argument list, because the phase identifiers `A-B-C` are added internally in the DBM-file. This option requires the *Internal phase seq.* checked box be selected in the component dialog box of the HVDC_6 object, as shown in Fig. 5.51. If it is selected, ATPDraw writes only the 5-character long core names in the `$INCLUDE` statement and let the extensions *A*, *B* and *C* be added inside the DBM library file.

Note that ATPDraw does not perform any diagnosis of the include file before sending the node names. Moreover, the *Internal phase seq.* option may result in conflict with transposition objects. As a result, this option should in general not be used in transposed circuits. To avoid the conflict use three input names for 3-phase nodes in DATA BASE MODULE files.

Each user specified objects might have a unique icon, which represents the object on the screen and an optional on-line help, which describes the meaning of parameters. These properties can be edited using the built in *Help* and *Icon Editors*. Fig. 5.49 shows an example file that is associated with the user specified 6-phase rectifier bridge.

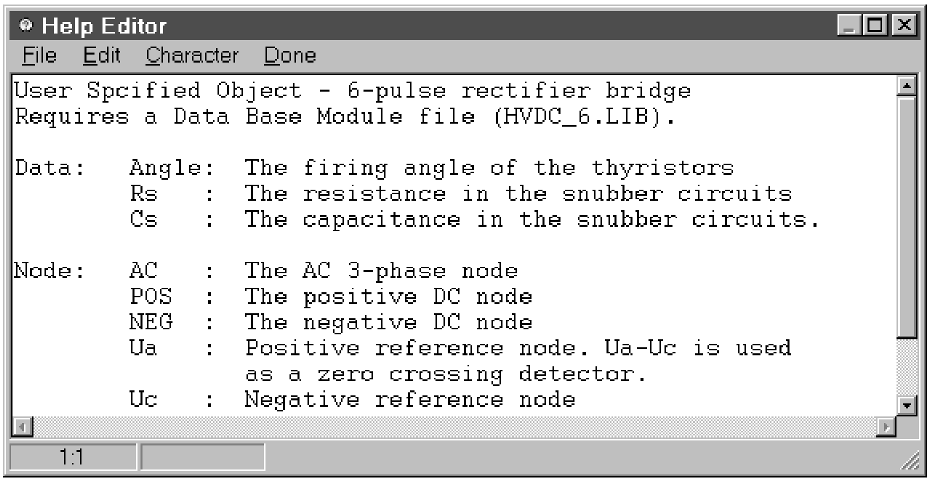


Fig. 5.49 - Help file of the HVDC_6 object.

Fig. 5.50 shows the icon editor window. The red lines in the background indicate the possible node positions on the icon border. Connecting lines to the external nodes of the object should be drawn from the symbol in the middle and out to the node positions specified in Fig. 5.48. The completed icon of the 6-pulse rectifier bridge is shown in Fig. 5.50.

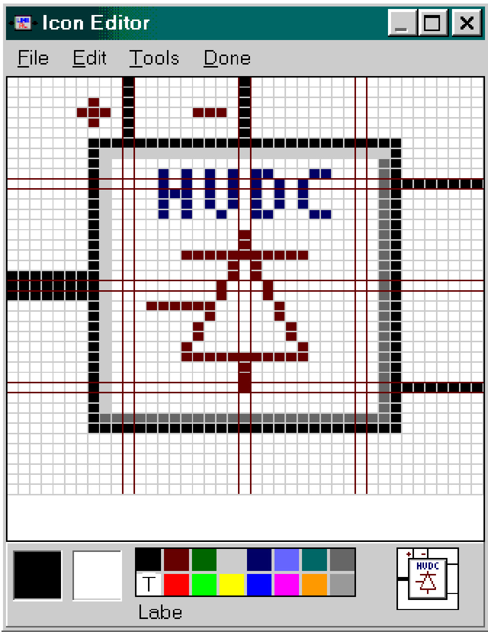


Fig. 5.50 - The icon associated with the new HVDC_6 object.

Finally, the just created support file must be saved to disk using the *Save* or *Save As* buttons. User specified sup-files are normally located in the `\USP` folder and their default extension is `.sup`. You can reload the support file of any user specified objects whenever you like, using the *User Specified | Edit sup-file* option of the *Objects* menu.

The *User Specified | Files* in the component selection menu provides access to the user specified objects. The component dialog box of the `HVDC_6` object is very similar to that of the standard objects, as shown in Fig. 5.51. The name of the DBM-file which is referenced in the final ATP input file must be specified in the *$Include* field under *User specified.* The *Send parameters* check box is normally selected, if the USP object has at least one input node or data.

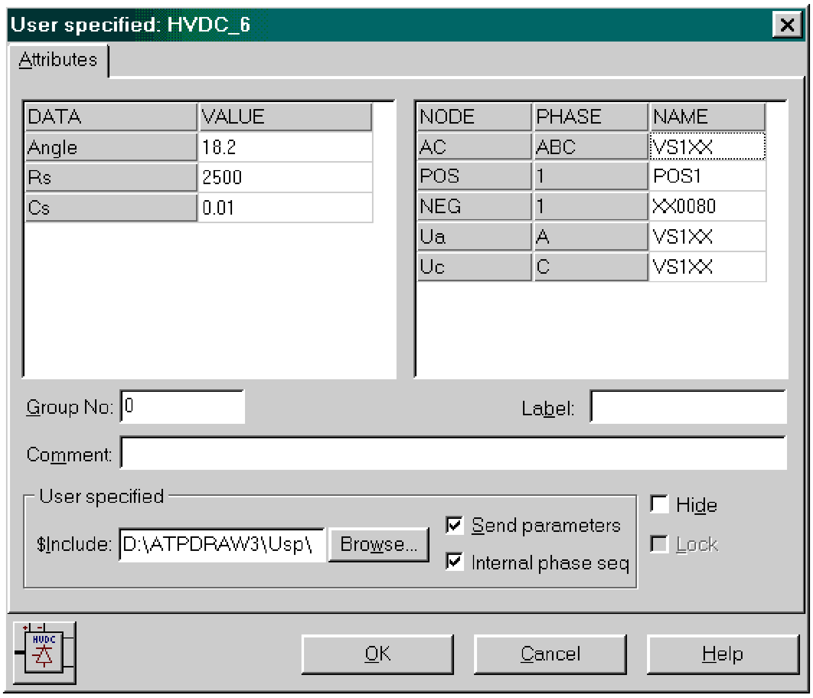


Fig. 5.51 - Component dialog box of the new user specified HVDC_6 object.

5.8.2 Creating a user specified, nonlinear transformer model

Supporting routine BCTRAN can be used to derive a linear representation of a single or 3-phase multi-winding transformer, using excitation and short circuit test data. If the frequency range of interest does not exceed some kHz, the inter-winding capacitances and earth capacitance of the HV and LV windings can be simulated by adding lumped capacitances connected to the terminals of the transformer. Although BCTRAN produces only a linear representation of the transformer, connecting nonlinear inductances to the winding closest to the iron core as external elements, provides an easy way to take the saturation and/or hysteresis into account. It is noted that the BCTRAN object is now supported by ATPDraw in a user friendly way (see in section 5.6), but the procedure described here gives more flexibility in handling of the iron core nonlinearities and allows incorporation of winding capacitances in the USP object, if needed. Further advantage of the USP based modeling is that users do not need to run the BCTRAN supporting routine as many times as such kind of transformers present in the circuit before the execution of the time domain simulation. Creating such a user specified component however requires some experience in two ATP supporting routines: DATA BASE MODULE and BCTRAN.

The BCTRAN model requires easily available input data only, like the name-plate data of a generator step-up transformer shown below:

Voltage rating V_{high}/V_{low}	132/15 kV
Winding connection:	Ynd11
Power rating:	155 MVA
Excitation losses:	74 kW
Excitation current:	0.3% / 2.67 A
Short circuit losses:	461 kW
Short circuit reactance:	14 %

The zero sequence excitation current and losses are approximately equal to the positive sequence measurements because the presence of delta connected secondary winding. Taking that the nonlinear magnetizing inductance is going to be added to the model as an external element, only the resistive component of the excitation current (0.05%) must entered in the BCTRAN input file shown next:

```
BEGIN NEW DATA CASE
ACCESS MODULE BCTRAN
$ERASE
 2       50.     0.05       155.       74.     0.05       155.       74. 0 2 2
  1     76.21           HVBUSASTRPNTHVBUSBSTRPNTHVBUSCSTRPNT
  2      15.0           LVBUSALVBUSCLVBUSBLVBUSALVBUSCLVBUSB
 1 2      461.      14.0      155.      14.0      155. 0 1
BLANK
$PUNCH
BLANK
BEGIN NEW DATA CASE
BLANK
BLANK
```

Running this file through ATP will produce an output punch-file that can be used as input for the Data Base Module (DBM) run. The process of creating a DBM-file is certainly the most difficult part of adding new circuit objects to ATPDraw. The input file to the DBM supporting routine of ATP begins with a header declaration followed by the circuit description. The ATP Rule Book [3] chapter XIX-F explains in detail how to create such a file. The output of the DBM supporting routine is a *.lib* file, that can actually be considered as an external procedure which is included to the ATP simulation at run time via a `$INCLUDE` call.

5.8.2.1 Creating a Data Base Module file for the BCTRAN object

The DBM-file begins with a header declaration followed by the ATP request card `DATA BASE MODULE` and ends with a `$PUNCH` request. The `ARG` declaration together with the `NUM` card (if needed) specifies the external variables (numerical + node names) and the sequence of arguments for the `$INCLUDE` procedure. The rest of the file describes the BCTRAN model. Note that data sorting card `/BRANCH` is part of the file, but no `BLANK BRANCH` indicator is required.

The `ARG` declaration of the DBM-file includes 7 node names in this example:

`HVBUSA, HVBUSB, HVBUSC`: The 3-phase node of the high voltage terminal
`LVBUSA, LVBUSB, LVBUSC`: The 3-phase node of the low voltage terminal
`STRPNT`: The 1-phase node of the HV neutral

The rest of the DBM-file is the transformer model description as produced by the BCTRAN supporting routine of ATP. The structure of the DBM input file is shown below:

```
BEGIN NEW DATA CASE --NOSORT--
DATA BASE MODULE
$ERASE
ARG,HVBUSA,HVBUSB,HVBUSC,LVBUSA,LVBUSB,LVBUSC,STRPNT
<<<< The .PCH file generated by the >>>>
<<<< BCTRAN supporting routine must >>>>
<<<< be inserted here               >>>>
BEGIN NEW DATA CASE
C          !!! This comment line here is mandatory !!!
$PUNCH, MYTRAFO.LIB
BEGIN NEW DATA CASE
BLANK
BLANK
```

Running the DBM-file through ATP will produce a file *mytrafo.lib* that must be stored in the \USP folder of ATPDraw.

```
KARD  3  3  4  4  6  6 10 10 11 11 13 13 16 16 20 20 25 25
KARG  4  6  4  5  5  6  1  7  4  6  2  7  4  5  3  7  5  6
KBEG  3  9  9  3  9  3  3  9  3  9  3  9  9  3  3  9  9  3
KEND  8 14 14  8 14  8  8 14  8 14  8 14 14  8  8 14 14  8
KTEX  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
$ERASE
C  <++++++>  Cards punched by support routine on  28-Jan-02  14.10.13  <++++++>
C ACCESS MODULE BCTRAN
C $ERASE
C  2       50.     0.05       155.       74.     0.05       155.       74. 0 2 2
C   1     76.21           HVBUSASTRPNTHVBUSBSTRPNTHVBUSCSTRPNT
C   2      15.0           LVBUSALVBUSCLVBUSBLVBUSALVBUSCLVBUSB
C  1 2      461.      14.0       155.       14.0       155. 0 1
C BLANK
$VINTAGE, 1,
 1LVBUSALVBUSC             9121.6157726436
 2LVBUSBLVBUSA             0.0
                           9121.6157726436
 3LVBUSCLVBUSB             0.0
                           0.0
                           9121.6157726436
  USE AR
 1HVBUSASTRPNT             19.966704093183 .16716783247242
 2LVBUSALVBUSC             -101.4441679294             0.0
                           515.41471986794 .00647606659729
 3HVBUSBSTRPNT                         0.0             0.0
                                       0.0             0.0
                           19.966704093183 .16716783247242
 4LVBUSBLVBUSA                         0.0             0.0
                                       0.0             0.0
                           -101.4441679294             0.0
                           515.41471986794 .00647606659729
 5HVBUSCSTRPNT                         0.0             0.0
                                       0.0             0.0
                                       0.0             0.0
                                       0.0             0.0
                           19.966704093183 .16716783247242
 6LVBUSCLVBUSB                         0.0             0.0
                                       0.0             0.0
                                       0.0             0.0
                                       0.0             0.0
                           -101.4441679294             0.0
                           515.41471986794 .00647606659729
$VINTAGE, 0,
$UNITS, -1.,-1.
  USE RL
C ----------------- << case separator >>> -----------
$EOF   User-supplied header cards follow.         28-Jan-02  14.28.28
ARG,HVBUSA,HVBUSB,HVBUSC,LVBUSA,LVBUSB,LVBUSC,STRPNT
```

5.8.2.2 Creating new support file and icon

Next step is to create a new user specified object via the *Object | User Specified | New sup file* menu of ATPDraw. The process of creating a new object consists of two steps: creating parameter support and creating an icon. Since no NUM card exists in the DBM header the number of data is 0, the number of nodes is 3 in this example as shown in Fig. 5.52.

On the *Nodes* tab, a *Name* can be assigned to each nodes. The number of phases and the node position on the icon border must also be specified here. The name of the nodes may differ from the name used in the *.lib* file, but the node sequence must be the same as specified on the ARG list. Each user specified component might have an icon and an optional on-line help, which describes the meaning of input parameters. The appearance of this icon is up to the users' creativity, but it is recommended to indicate three phase nodes with thick lines and to locate them according to the *Pos (1..12)* setting on the *Nodes* tab. Finally, the support file of the object must be saved to disk using the *Save* button (the default location is the `/USP` folder), to make the new USP object accessible via the *User Specified | Files* option of the component selection menu.

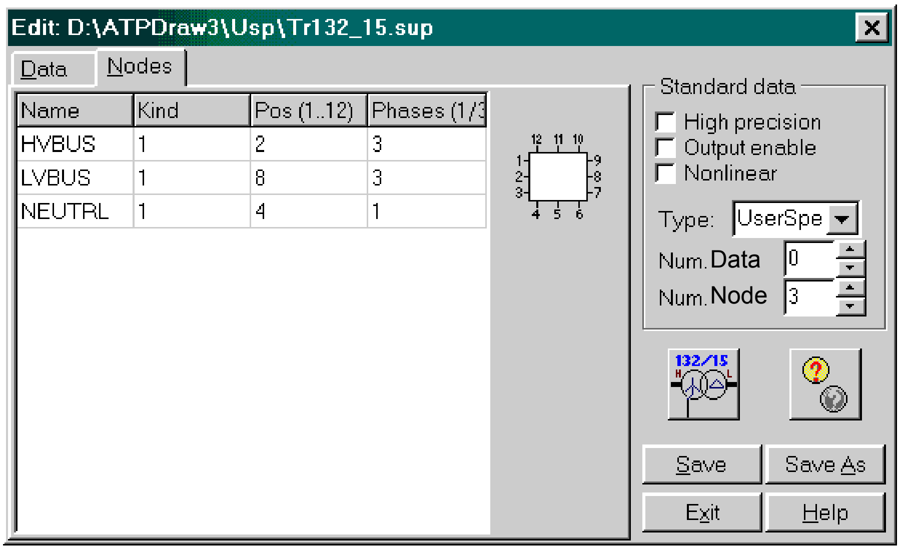


Fig. 5.52 – Creating support file for the new BCTRAN object.

The user specified components can be used in combination with the new grouping feature of ATPDraw as shown in Fig. 5.53. In this example, the linear part of the transformer model has been completed with winding capacitances as external components and three nonlinear Type-96 hysteretic inductors in delta connection at the 15 kV terminals, which represent the nonlinear magnetic core.

The *Compress* feature of ATPDraw supports single icon replacement of these 7 objects. The inter-winding and winding-to-earth capacitances are input parameters to the group object. As shown below, the group object's icon can be customized, as well. An artistic icon may improve the readability of the circuit and help in understanding of the circuit file for others.

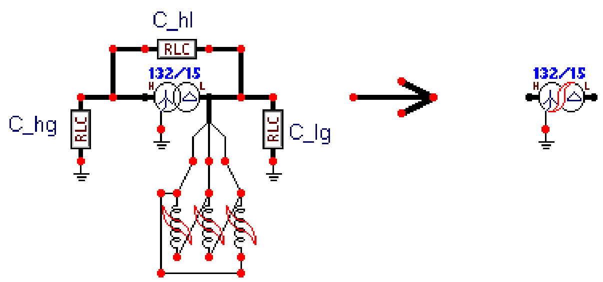


Fig. 5.53- Compressing the transformer model into a single object.

6. Application Manual . . .

. . . .

ATPDraw™
for Windows
3.5

This chapter begins with some simple examples. You will not be shown how to create these circuits, but the circuits files `Exa_*.adp` are part of the ATPDraw distribution. To load these example circuits into the circuit window of ATPDraw, use the *File | Open* command (or *Ctrl + O*) and select the file name in the *Open Project* dialog. The resulting ATP-files will be given at the end of each description. Simulation results and/or comparison with measurements are also presented in some cases. These figures have been obtained by processing the `.pl4` output file or field test records with prost-processors PlotXY or ATP_Analyzer.

6.1 Switching in 500 kV system *(Exa_3.adp)*

This example shows how to perform a switching analysis in a simplified network. The line model used in this example is a Π-equivalent line calculated for 5000 Hz, by LINE CONSTANTS outside of ATPDraw. How to create a more accurate JMarti line model and how to use LCC objects, the interactive graphical preprocessor of ATPDraw for LINE/CABLE CONSTANTS/PARAMETERS support, are shown in section 5.3 of the Advanced Manual and section 6.5 of this manual.

The example circuit is shown on Fig. 6.1/a, the equivalent ATPDraw circuit in Fig. 6.1/b.

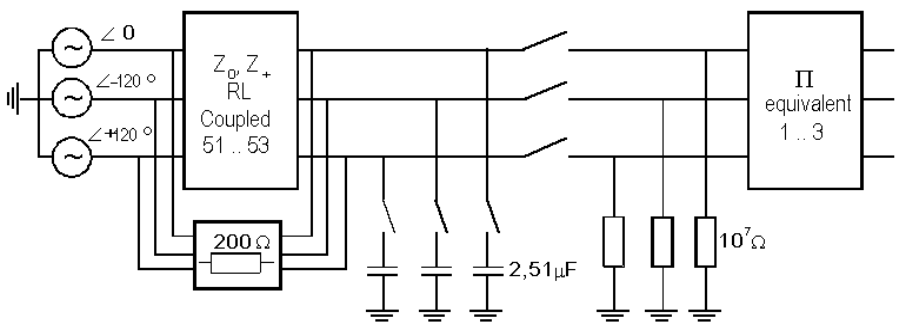


Fig. 6.1/a - Switching example circuit.

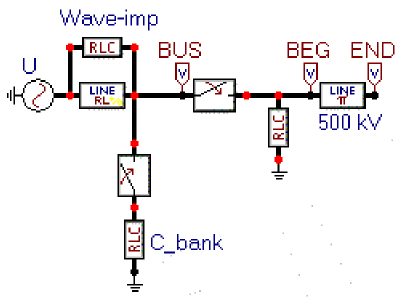


Fig. 6.1/b - Example circuit 3, Line switching `(Exa_3.adp)`.

Source:

The source is a 500 kV 3-phase AC source. The amplitude of the voltage source (U/I = 0) is

$\text{Amp} = 500 \cdot \frac{\sqrt{2}}{\sqrt{3}}$ kV . The input menu for the source is shown in Fig. 6.2:

Line switch:

The Π-equivalent line is connected to the source with a three-phase switch, having independent closing and opening times in all phases. The switches are initially open and close at:

Phase *A*: 33.33 ms, Phase *B*: 36.10 ms, Phase *C*: 38.80 ms

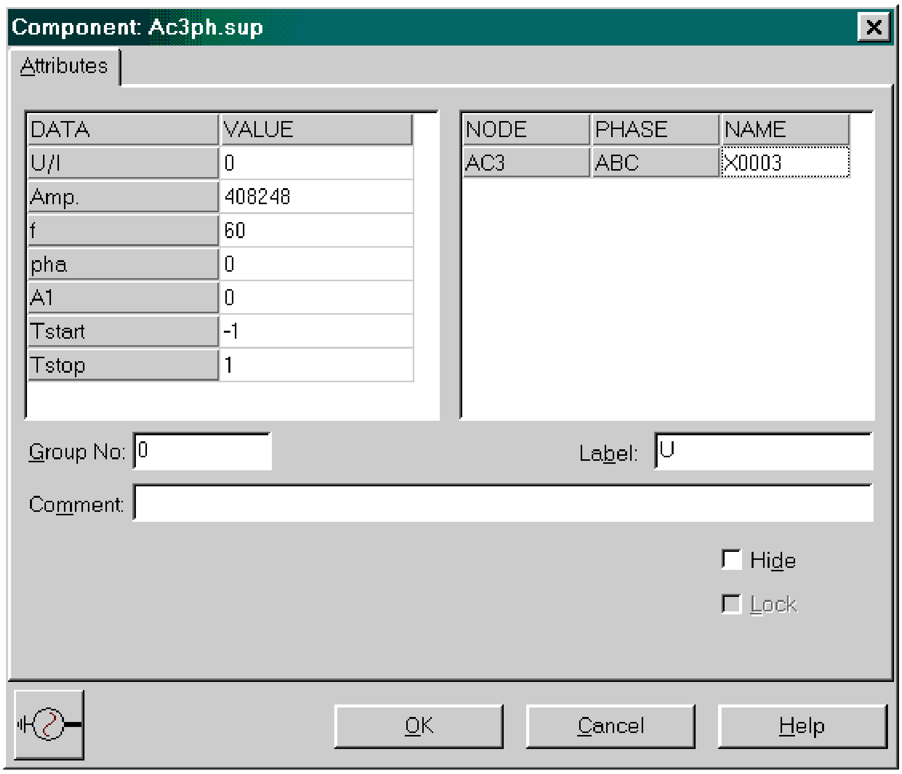


Fig. 6.2 - Three phase source input window.

Capacitor bank:

The capacitor bank is 2.51 μF in all phases. The switches connecting the bank to the network is initially open and close at

Phase *A*: 133.33 ms,   Phase *B*: 136.10 ms,   Phase *C*: 138.80 ms

Pi-equivalent line:

The Π-equivalent line R-L-C data were calculated by the LINE CONSTANTS supporting routine of ATP outside ATPDraw. The line is a 500 kV overhead line with 2x3 phase conductors and 2 ground wires. Line parameters were taken from ATP benchmark file `DCN3.DAT`. The calculation resulted in the following line matrices for that 138 miles line (only lower-triangle part is given). Read the inductances in [mH], the resistances in [Ω] and the capacitances in [μF]:

R

474.76		
353.81	315.13	
442.02	353.81	474.76

L

288.8		
71.35	266.15	
73.45	71.35	288.8

C

2.3224		
-0.3637	2.3684	
-0.2725	-0.3637	2.3224

If you click the right mouse button on the line Π icon of Fig. 6.1/b, the object dialog box shown in Fig. 6.3 appears, where you can enter the above R-L-C matrix data. The imaginary part of the line impedance may be specified in [Ω], as well. In that case the *Xopt.* parameter under *ATP* | *Settings / Simulation* must be set equal to the frequency, at which the impedances were calculated (e.g. Xopt. should be set 5 kHz in this example).

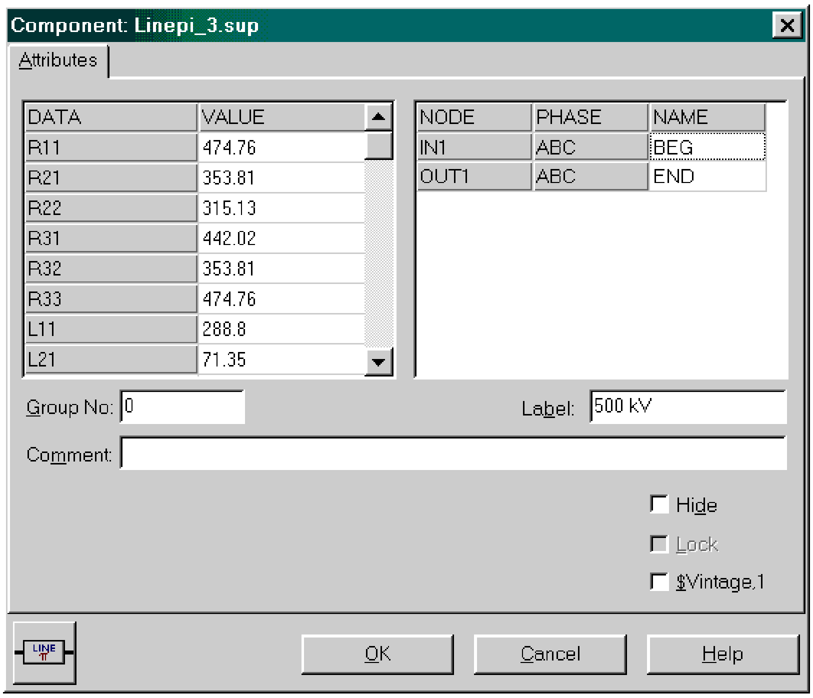
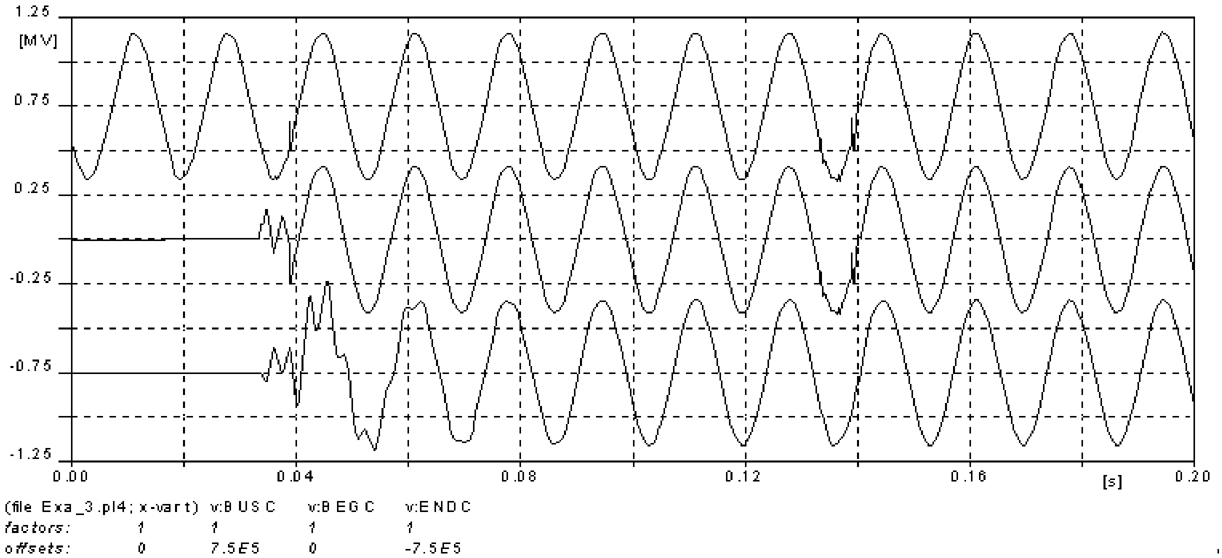


Fig. 6.3 - Three phase pi-equivalent input window

The ATP input file created by ATPDraw and the simulated voltages at the 500 kV bus, at the sending and at the receiving end of the line are shown below:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  June, Sunday 30, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-5      .2
     500       5       1       1       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
1 BEGA  ENDA              474.76 288.82.3224
2 BEGB  ENDB              353.81 71.35-0.364315.13266.152.3684
3 BEGC  ENDC              442.02 73.45-0.273353.81 71.35-0.364474.76 288.82.3224
51X0003ABUSA                 .55        8.98
52X0003BBUSB                .711      11.857
53X0003CBUSC
  X0003ABUSA                200.                                               0
  X0003BBUSB                200.                                               0
  X0003CBUSC                200.                                               0
        X0010A                          2.51                                   0
        X0010B                          2.51                                   0
        X0010C                          2.51                                   0
        BEGA                1.E7                                               0
        BEGB                1.E7                                               0
        BEGC                1.E7                                               0
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  BUSA  BEGA      .03333       10.                                             0
  BUSB  BEGB       .0361       10.                                             0
  BUSC  BEGC       .0388       10.                                             0
  X0010ABUSA      .13333       10.                                             0
  X0010BBUSB       .1361       10.                                             0
  X0010CBUSC       .1388       10.                                             0
/SOURCE
```

```
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14X0003A 0   408248.       60.                                  -1.         1.
14X0003B 0   408248.       60.      -120.                       -1.         1.
14X0003C 0   408248.       60.       120.                       -1.         1.
/INITIAL
/OUTPUT
  BEGA  BEGB  BEGC  ENDA  ENDB  ENDC  BUSA  BUSB  BUSC
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```



6.2 TACS controlled induction machine *(Exa_4.adp)*

This example shows the usage of the Universal Machine type 3, manual initialization along with usage of TACS. The use of info arrows, whose purpose is to visualize information flow between the TACS FORTRAN objects are also shown here. The info arrows can be selected under *TACS | Draw relation* in the component selection menu and they are handled graphically as normal connections. They do not affect the ATP-file, however. The example is taken from exercise 46 in [2]. The ATPDraw constructed circuit is shown in Fig. 6.4/b:

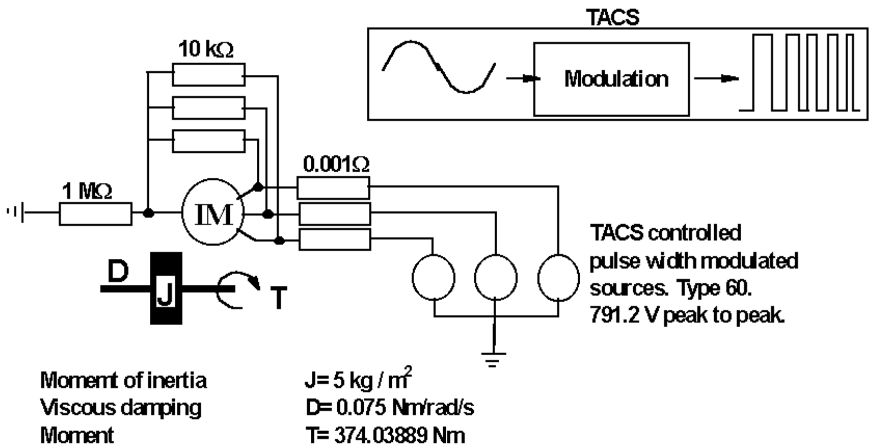


Fig. 6.4/a - Induction machine + TACS

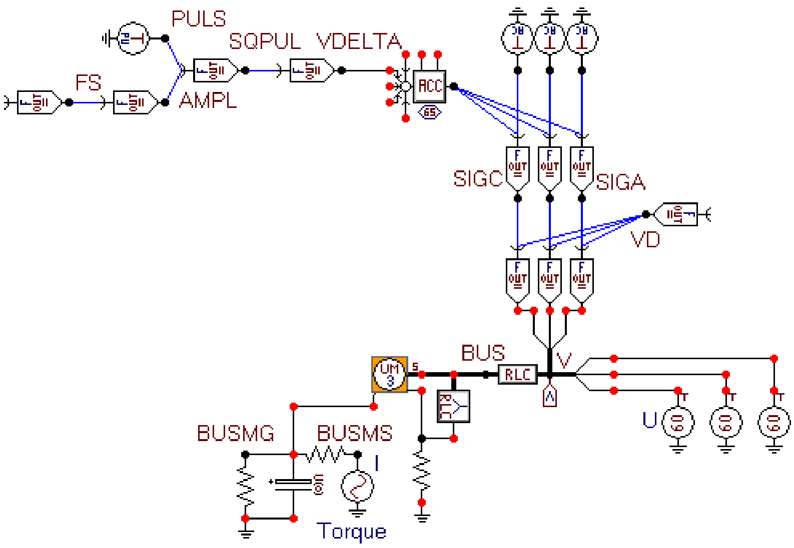


Fig. 6.4/b - ATPDraw scheme of the induction machine example (`Exa_4.adp`)

The TACS part of the circuit controls three sources producing a pulse width modulated armature voltage. The TACS objects FORTRAN1 is referenced in the Reference part of this Manual.

The input window of the TACS object at the end of the TACS chain is shown in Fig. 6.5. This TACS object creates the armature voltage in phase *A* of the 3-phase node *V*.

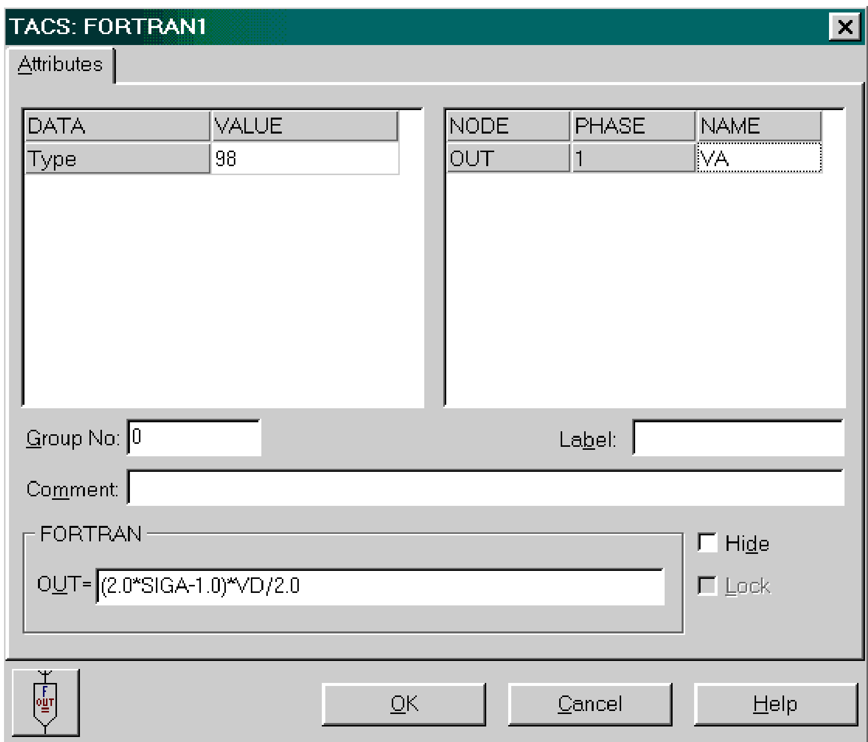


Fig. 6.5 - TACS Fortran input window

In the TACS statement the user must type in the expression(s). Only single phase TACS Fortran objects are supported. The two (blue) info arrows into this TACS object serve as visualization of the *SIGA* (from node *SIGA*) and *VD* signals.

The induction machine was given the data shown in Fig. 6.6:

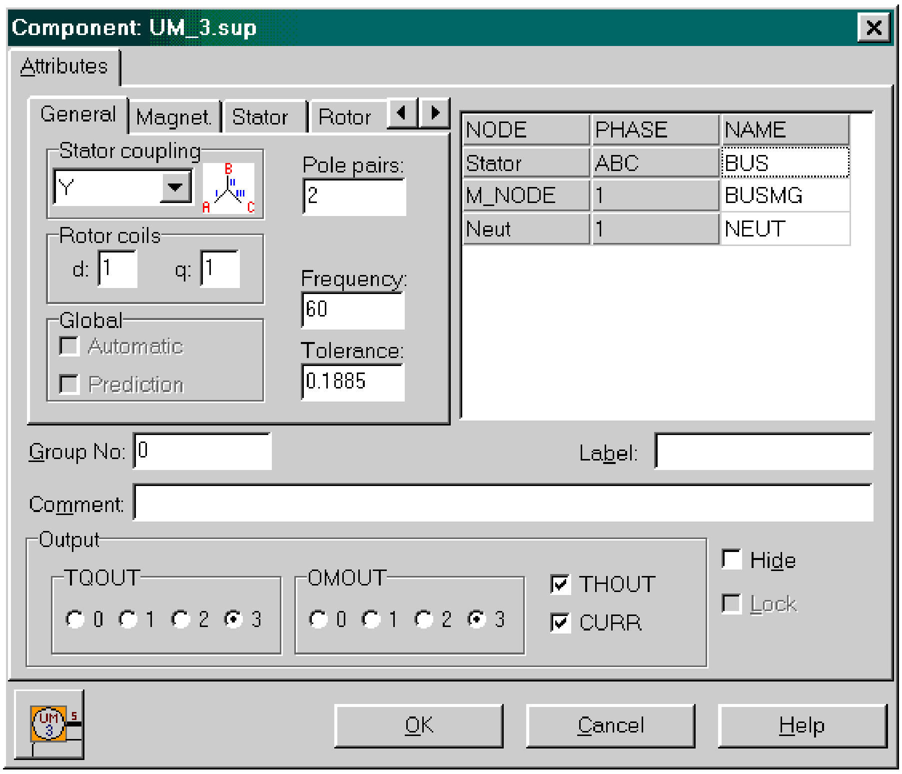


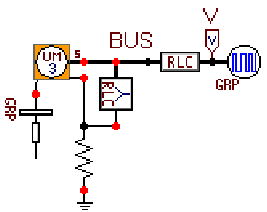
Fig. 6.6 - Induction machine input window

The numerical values in Fig. 6.6 must be specified by the user as in the case for all object input windows. The identity text in front of each attribute strictly follows the input variable in the ATP Rule Book [3]. The ATP-file created by ATPDraw is shown below:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Tuesday 30, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
C Induction motor supplied by a
C pulse width modulated source.
C Test example 1.
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-5      .1
     500        3       0       0       1       0       0       1       0
TACS HYBRID
/TACS
98FS        =1000
23PULS            2.       .001      .0005                       .000252
98AMPL      =4.0*FS
98SQPUL     =AMPL*(UNITY-PULS)
98VDELTA    =SQPUL*DELTAT
98VTRI   65         +VDELTA
14VCONTC         .95        60.       -90.
14VCONTB         .95        60.      -210.
14VCONTA         .95        60.        30.
98VB        =(2.0*SIGB-1.0)*VD/2.0
98VC        =(2.0*SIGC-1.0)*VD/2.0
98SIGC      =VCONTC .GT. VTRI
98VA        =(2.0*SIGA-1.0)*VD/2.0
98SIGB      =VCONTB .GT. VTRI
98SIGA      =VCONTA .GT. VTRI
98VD        =791.2
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
        NEUT                1.E6                                              2
        BUSMG              13.33                                              1
  BUSMG                                 5.E6                                  3
  BUSMG BUSMS              1.E-6                                              1
  BUSA  VA                  .001                                              1
  BUSB  VB                  .001                                              1
  BUSC  VC                  .001                                              1
  BUSA  NEUT                1.E4                                              0
  BUSB  NEUT                1.E4                                              0
  BUSC  NEUT                1.E4                                              0
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14BUSMS -1-374.03889     1.E-5
60VC      0
60VB      0
60VA      0
C Next comes Universal Machines
19 UM
00            0
BLANK general UM specification
 3 1 1331BUSMG        2                             .1885          60.
C Magnetization inductances
    182.840692         .0160
    .785398163         .0160
C Stator coils
                            BUSA  NEUT        1       73.5587
          .095         .0005BUSB  NEUT        1        80.545
          .095         .0005BUSC  NEUT        1     -154.1034
C Rotor coils
          .075         .0004                  1      169.6725
          .075         .0004                  1        19.285
BLANK UM
/INITIAL
 2BUSMG      182.840692
 3BUSMG                           182.840692
/OUTPUT
  VA    VB    VC
BLANK TACS
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

The new *Grouping* feature of ATPDraw can be used in a creative way in this example, too. The pulse width modulated source and the mechanical load might be compressed into a single icon. The compressed version of this example circuit is also part of the ATPDraw distribution with the name of `Exa_4g.adp`.



As shown left an artistic icon may improve the readability of the circuit and help in understanding the circuit for non-author users.

Fig. 6.7 - PWM source and mechanical load compressed into a single icon.

6.3 Usage of the Library and Reference objects *(Exa_5.adp)*

This example shows how can ATPDraw be used efficiently by a professional ATP user. The ATP expert simply creates the required ATP-file for a power system (which is often already available) outside of ATPDraw, and then builds a simple ATPDraw case, where the power system is included with $Include and a limited number of additional components e.g. switches are added. Node names internally in the include file must be the same as those used in the ATPDraw data case (*F1 - F4*, left adjusted in this example!).

The data case can then be sent to a person whose knowledge about ATP is rather limited. This person can run the case, manipulate the switches and establish the ground fault current along a transmission line, e.g. for relay setting purposes.

Any user specified objects might be used as a simple $Include library (*Send parameters* button off), but a predefined object is available under the *User Specified* field in the selection menu. Two other special objects are also available: the `Ref.1-ph` and `Ref.3-ph`. These objects are not written in the ATP-file at all, and their purpose is only to visualize a connectivity in the $Include file.

An example, where these objects (Library (`LIB1.A-B`) and 3-phase reference (`LIBREF_3`)) have been used, is shown in Fig. 6.8.

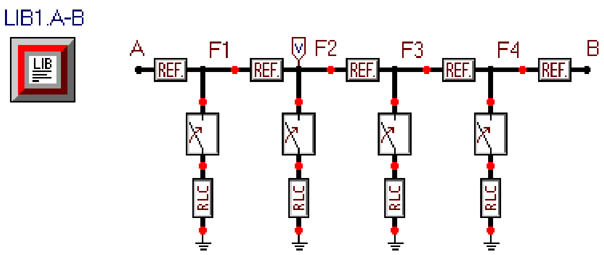


Fig. 6.8 - Usage of a library reference objects (`Exa_5.adp`)

If you click with the right mouse button on the Library object, (identified by the LIB1.A-B label) you get the input window to set the element attributes, as shown in Fig. 6.9.

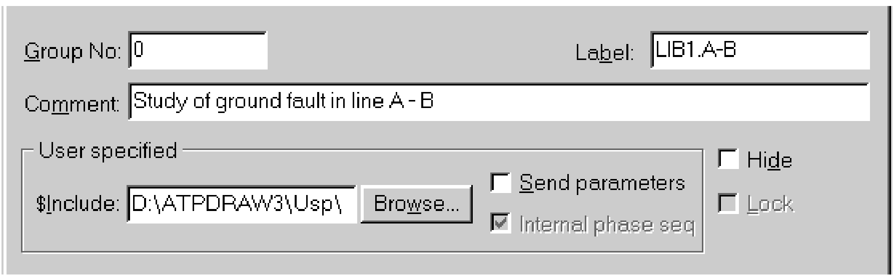


Fig. 6.9 - Library input window.

The most important menu field in this window is the *$Include* which gives the name and path of the file to be included. The *Browse* button allows you to select a file in the *Select Libray* dialog box. The *Send parameters* check box must be off. *Label* and *Comment* are optional fields.

If you click with the right mouse button on one of the 3-phase reference objects, a similar dialog appears with the difference that it has node name fields, too.

The reference objects are not represented in the `.atp` data file. Their role is just the visualization of the connectivity. The ATPDraw generated ATP-file is shown below:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  June, Sunday 30, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-6    .001
     500       1       1       1       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
        X0011A               10.                                               0
        X0011B               10.                                               0
        X0011C               10.                                               0
        X0013A               10.                                               0
        X0013B               10.                                               0
        X0013C               10.                                               0
        X0015A               10.                                               0
        X0015B               10.                                               0
        X0015C               10.                                               0
        X0017A               10.                                               0
        X0017B               10.                                               0
        X0017C               10.                                               0
C Study of ground fault in line A - B
$INCLUDE, D:\ATPDRAW3\USP\LIB.LIB
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  X0011AF1A         .033        2.                                             0
  X0011BF1B           2.        2.                                             0
  X0011CF1C           2.        2.                                             0
  X0013AF2A           2.        2.                                             0
  X0013BF2B           2.        2.                                             0
  X0013CF2C           2.        2.                                             0
  X0015AF3A           2.        2.                                             0
  X0015BF3B           2.        2.                                             0
  X0015CF3C           2.        2.                                             0
  X0017AF4A           2.        2.                                             0
  X0017BF4B           2.        2.                                             0
  X0017CF4C           2.        2.                                             0
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
/INITIAL
/OUTPUT
  F2A   F2B   F2C
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

When creating the ATP-file above the *Insert $Prefix and $Suffix cards* was unselected on the *ATP | Settings | Format* page, so the full path of the library file was written to the ATP-file in this example. Otherwise, if this option would have been selected, the library files were supposed to be located in the `\USP` folder and have the extension `.LIB`. Each library file specification is verified to meet these requirements. If the path of a library file specifies a different folder or the extension is not `.LIB`, an error dialog is displayed during the ATP-file generation process, enabling the user to correct the erroneous specification by stripping off the path and extension, continue the operation using an un-resolvable ATP include reference, or cancel the entire ATP-file generating process. Fig. 6.10 shows this error dialog, as an example.

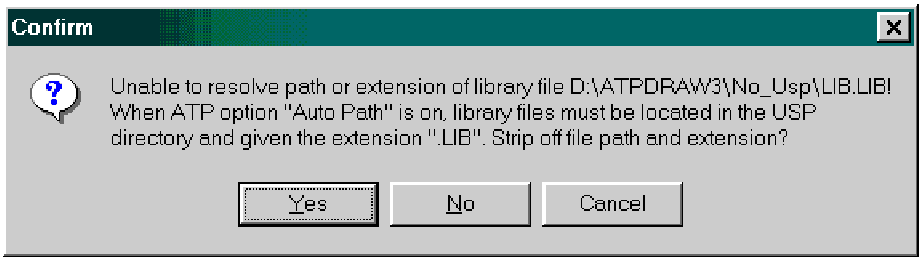


Fig. 6.10 - Error dialog when un-resolvable LIB-file specification was detected.

6.4 Modeling an HVDC station *(Exa_6.adp* and *Exa_6g.adp)*

In section 5.8.1 of the Advanced Manual, it is shown how to create a 6-pulse controlled thyristor-rectifier bridge and make it available in ATPDraw as a user specified single object. In this part of the manual, a practical example with the use of this user specified library object to build up a 12-pulse HVDC station will be shown. The example is based on exercise 54 in [2]. Two different project file format exists in the ATPDraw distribution:

a) `Exa_6.adp`: The thyristor bridge and its control is specified as user specified object by means of the DATA BASE MODULARIZATION feature of ATP and interfaced with the rest of the circuit via a `$Include` call.
b) `Exa_6g.adp`: The thyristor firing is simulated directly in ATPDraw by means of the TACS support and connected with the thyristors via coupling to TACS objects. Finally, it is shown how to compress the control circuit and valves into a single icon to get a similar circuit layout than in case a).

The first alternative (`Exa_6.adp`) is shown in Fig. 6.11, the second one (`Exa_6g.adp`) in Fig. 6.13. The HVDC station in both circuits is supplied by a 3-phase AC source in front of two transformers. Fig. 6.12 shows the data input dialog box of the new object. As it can be seen, the source frequency is not one of the input parameters of the USP object. The frequency is "hard-wired" in the DATA BASE MODULE code, which means that this circuit works only for 60 Hz systems. For systems of 50 Hz the library file `HVDC_65.LIB` should be used. This object is also given in the lower left corner of the circuit as hidden object.

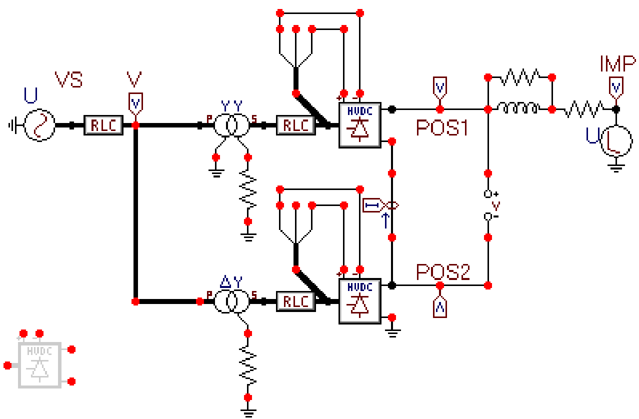


Fig. 6.11 - 12-pulse HVDC station (`Exa_6.adp`)

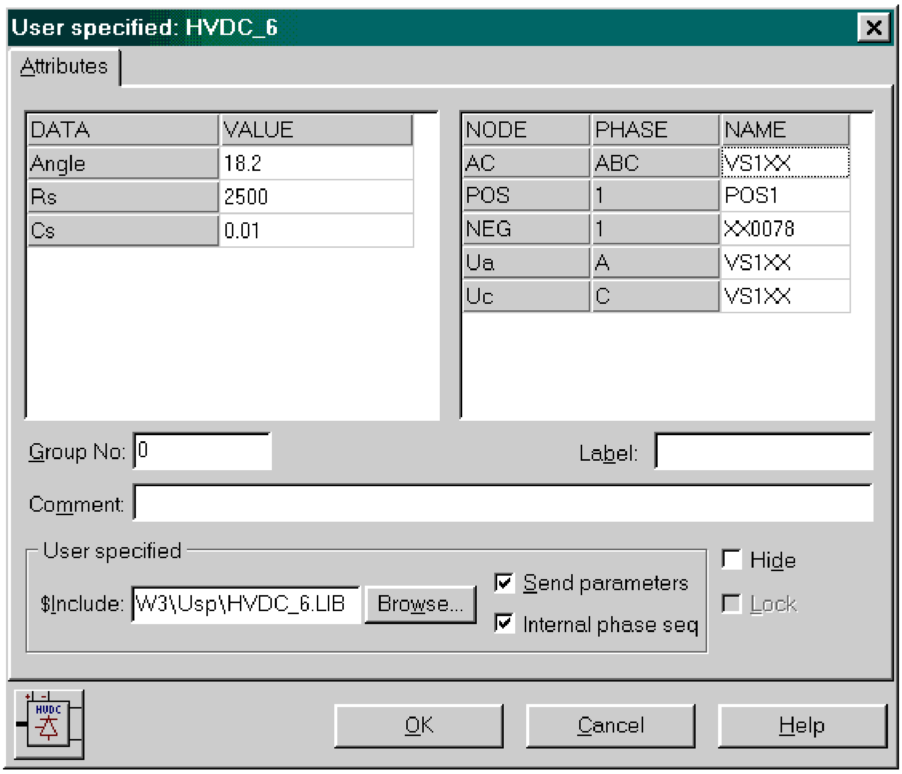


Fig. 6.12 - Input dialog box of the users specified 6-pulse rectifier bridge.

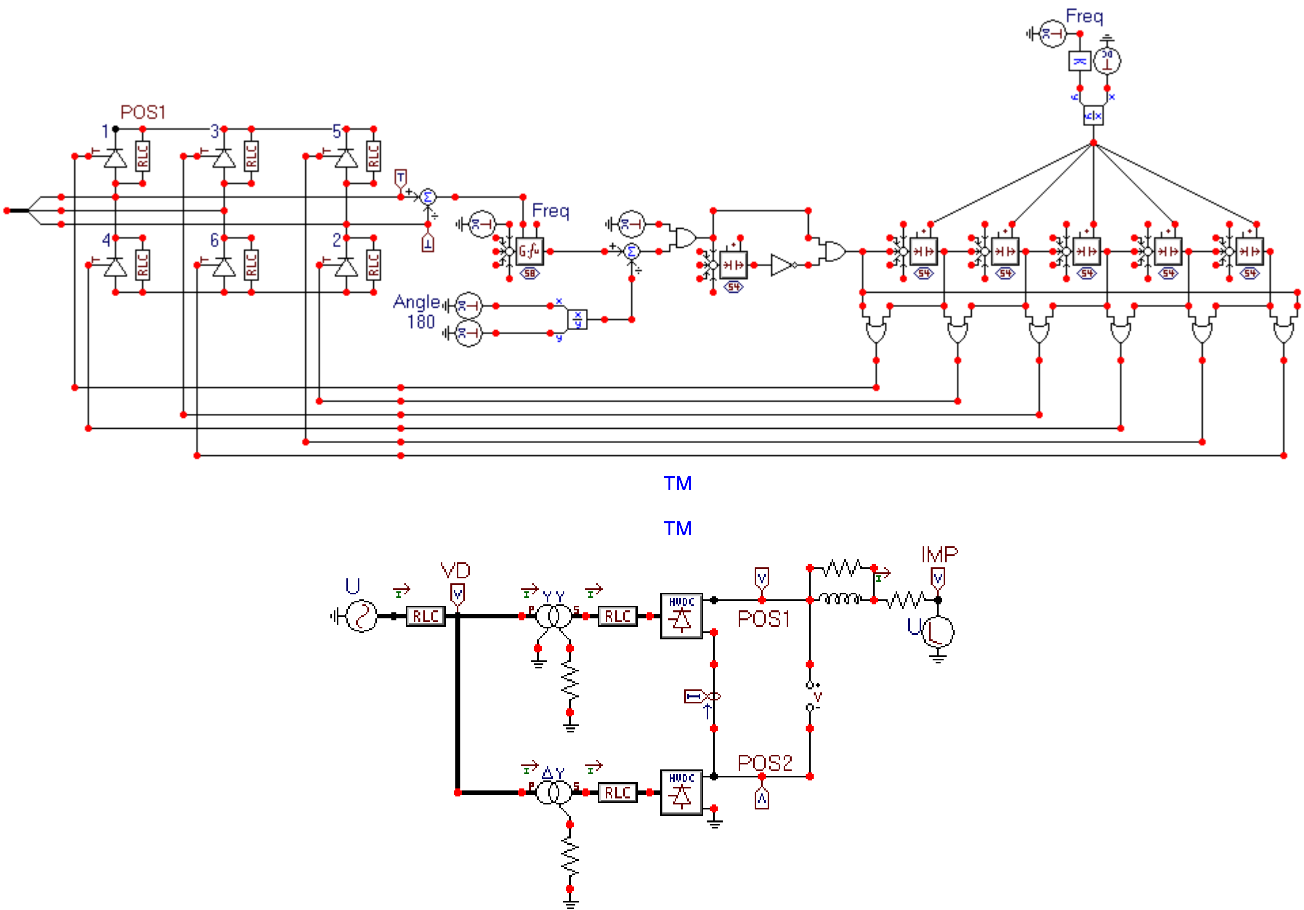


Fig. 6.13 - 12-pulse HVDC station (`Exa_6g.adp`).
Compressing the valves and their control into a single HVDC group object.

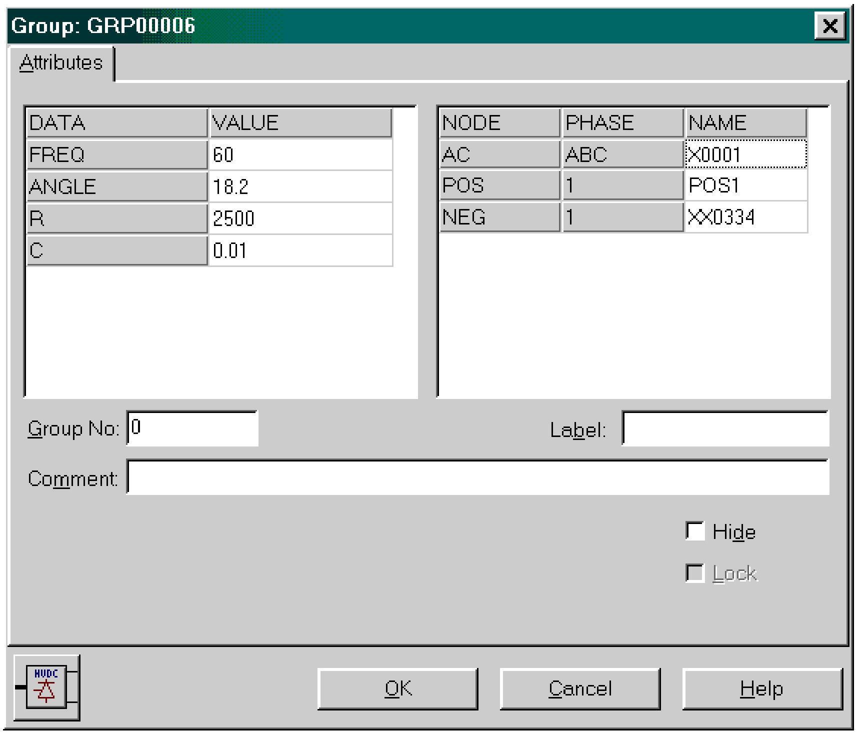


Fig. 6.14 - Input dialog box of the HVDC group object.

The ATP-file created by ATPDraw is shown next:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Monday 1, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT   >< Tmax >< Xopt >< Copt >
   2.E-5    .035
     500       1       1       1       1       0       0       1       0
TACS HYBRID
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
  TRANSFORMER                 TX0001 1.E12                             1
          9999
 1VA                       .1 98.57    1.
 2VS1MA XX0017            .022 21.59  .468
  TRANSFORMER TX0001                 TX0002
 1VB
 2VS1MB XX0017
  TRANSFORMER TX0001                 TX0003
 1VC
 2VS1MC XX0017
  VS1MA VS1XXA           .0001                                               1
  VS1MB VS1XXB           .0001                                               1
  VS1MC VS1XXC           .0001                                               1
        XX0017            1.E7                                               0
  VSA   VA               .0001                                               1
  VSB   VB               .0001                                               1
  VSC   VC               .0001                                               1
  TRANSFORMER                 TX0004 1.E12                             1
          9999
 1VA    VC                 .3 295.7    1.
 2VS2MA XX0032            .022 21.59 .2702
  TRANSFORMER TX0004                 TX0005
```

```
 1VB    VA
 2VS2MB XX0032
  TRANSFORMER TX0004                  TX0006
 1VC    VB
 2VS2MC XX0032
  VS2MA VS2XXA           .0001                                              1
  VS2MB VS2XXB           .0001                                              1
  VS2MC VS2XXC           .0001                                              1
        XX0032            1.E7                                              0
  POS1  XX0038                  100.                                        0
  POS1  XX0038            8.E3                                              0
  XX0038IMP              15.35                                              1
  POS1  POS2             1.E+9                                              2
$INCLUDE, D:\ATPDRAW3\USP\HVDC_6.LIB, VS1XX, POS1##, XX0078, VS1XXA, VS1XXC $$
  ,  18.2, 2500.,  0.01
$INCLUDE, D:\ATPDRAW3\USP\HVDC_6.LIB, VS2XX, POS2##, ######, VS2XXA, VS2XXC $$
  ,  18.2, 2500.,  0.01
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  POS2  XX0078                                        MEASURING             1
/SOURCE
C < n 1><>>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14VSA    0   187794.      60.      -60.                          -1.        1.
14VSB    0   187794.      60.     -180.                          -1.        1.
14VSC    0   187794.      60.       60.                          -1.        1.
12IMP    0   229660.               .01
/INITIAL
/OUTPUT
  VA    VB    VC    IMP   POS1  POS2
BLANK TACS
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

Note the `TACS HYBRID` request marked as bold in the ATP-file. This is the result of the `Insert TACS HYBRID card` check box selected under the *Miscellaneous* field of the *ATP | Settings / Format* page. Leaving this request un-selected, ATPDraw would not know that TACS is present in the USP library object `HVDC_6`. I.e. ATPDraw does not analyze the LIB-files before connecting them via $Include to the circuit. It is similarly important to note that a full 5-character node name (e.g. `VS1XX`) has to be used for the 3-phase node of the `HVDC_6` object. It is because the structure of the LIB-file requires *A*, *B* and *C* phase identifier letters are to be added as the 6th character. Furthermore, the *Internal phase seq* indicator must be selected (see Fig. 6.12) to pass the first parameter as a 3-phase node rather than 3 single-phase nodes, which is the default procedure in ATPDraw.

Only case 1) below includes the `HVDC_6.LIB` into the ATP-file properly, where the node has a 5-character name and the *Internal phase seq* is active. The other two are equally bad and produce a run time error or give incorrect results at the simulation because only 3-character name has been specified in case 2) or the *Internal phase seq* was unselected in case 3):

```
1) $INCLUDE, HVDC_6, VS1XX, POS1##, XX0011, VS1XXA, VS1XXC . . .
2) $INCLUDE, HVDC_6, VS1##, POS1##, XX0011, VS1A##, VS1C## . . .
3) $INCLUDE, HVDC_6, VS1XXA, VS1XXB, VS1XXC, POS1##, XX0011, VS1XXA, VS1XXC
```

Using ATPDraw's TACS support for modeling the firing control and then compressing the circuit into a single HVDC group, as shown in Fig. 6.13 seems a better alternative today, because the restrictions and possible conflicts concerning the use of USP LIB-files do not exist.

In Fig. 6.15 four curves from the simulation is shown which are equal to the results in [2].

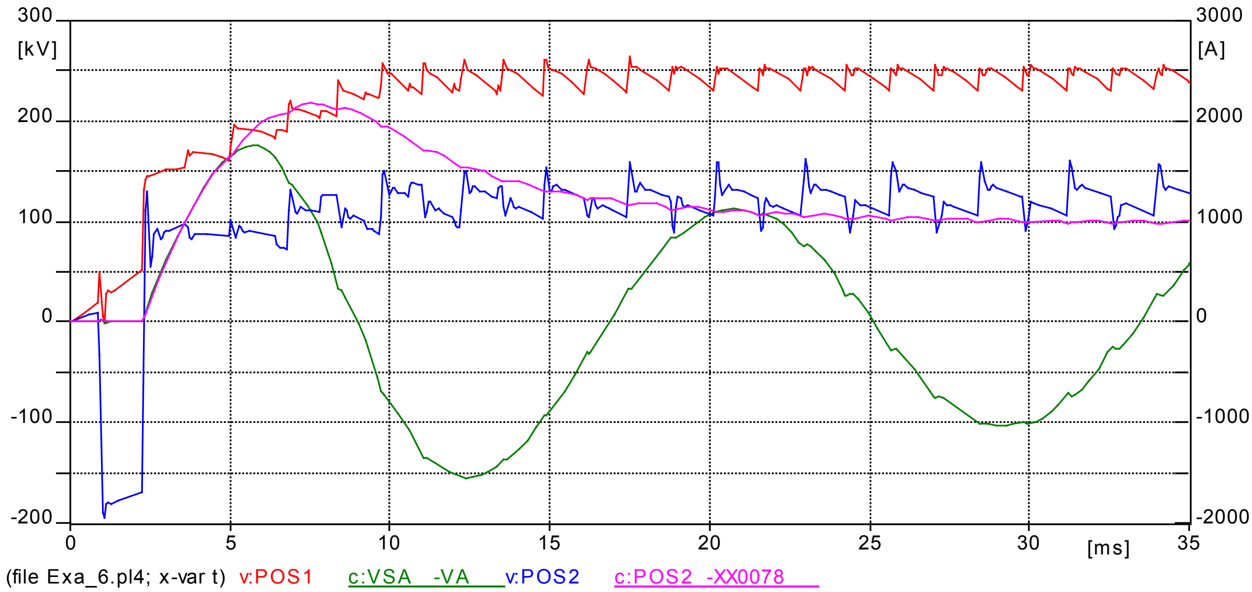


Fig. 6.15 - ATP simulation results.

6.5 Switching studies using JMarti LCC objects

The LCC modeling features of ATPDraw are described in detail in section 5.3 of the Advanced Manual. Line modeling by LCC objects means that user specifies the geometrical arrangement and material constants, then ATPDraw executes ATP's Line/Cable Constants routine and converts the output punch-file to DBM library format. The resulting LIB-file will then be included in the final ATP-file via a `$Include` call. The JMarti option is one out of the five alternatives supported by ATPDraw's LCC object. Here two switching transient simulation examples are presented.

6.5.1 JMarti model of a 500 kV line (*Exa_7.adp*)

The 3-phase switching example (`Exa_3.adp`) created in section 6.1 of this manual is repeated here. This time the 500 kV overhead line is represented by a frequency dependent 3-phase JMarti model. The example circuit is shown in Fig. 6.16.

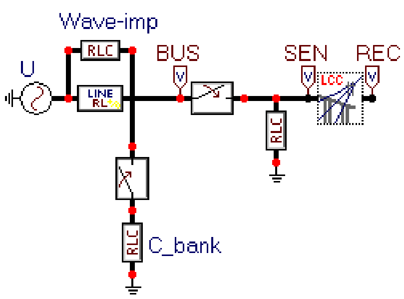


Fig. 6.16 - JMarti line in switching study (`Exa_7.adp`)

The overhead in this example is a 138 miles long single circuit 500 kV line (from John Day to Lower Monumental in Oregon, USA) with two sky wires taken from ATP benchmark `DCN3.DAT`. The line configuration is given in Fig. 6.17.

The line parameters are given in English units, the phase wires are represented by a tubular conductor with parameters:

DC resistance = 0.05215 Ω/mile
Outside diameter of the conductor = 1.602 inch
Inner radius of the tube = 0.2178 inch
ATPDraw calculates the thickness/diameter value internally (T/D = 0.364).

The sky wires are solid, so inner radius is zero and T/D = 0.5. Other parameters are:

DC resistance = 2.61 Ω/mile
Outside diameter of the conductor = 0.386 inch.

The resistivity of the soil equals to 100 Ωm. The conductor separation in the bundle is 18 inch.

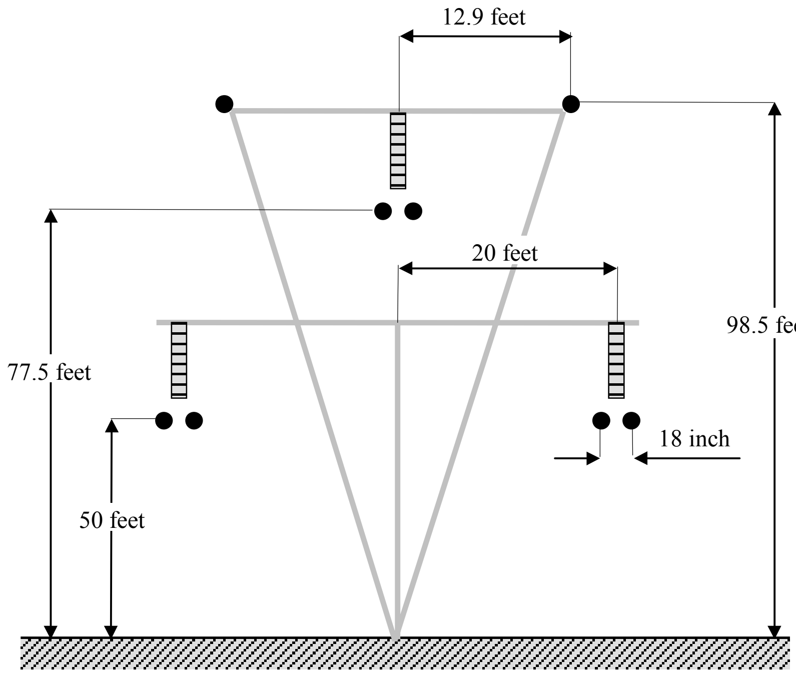
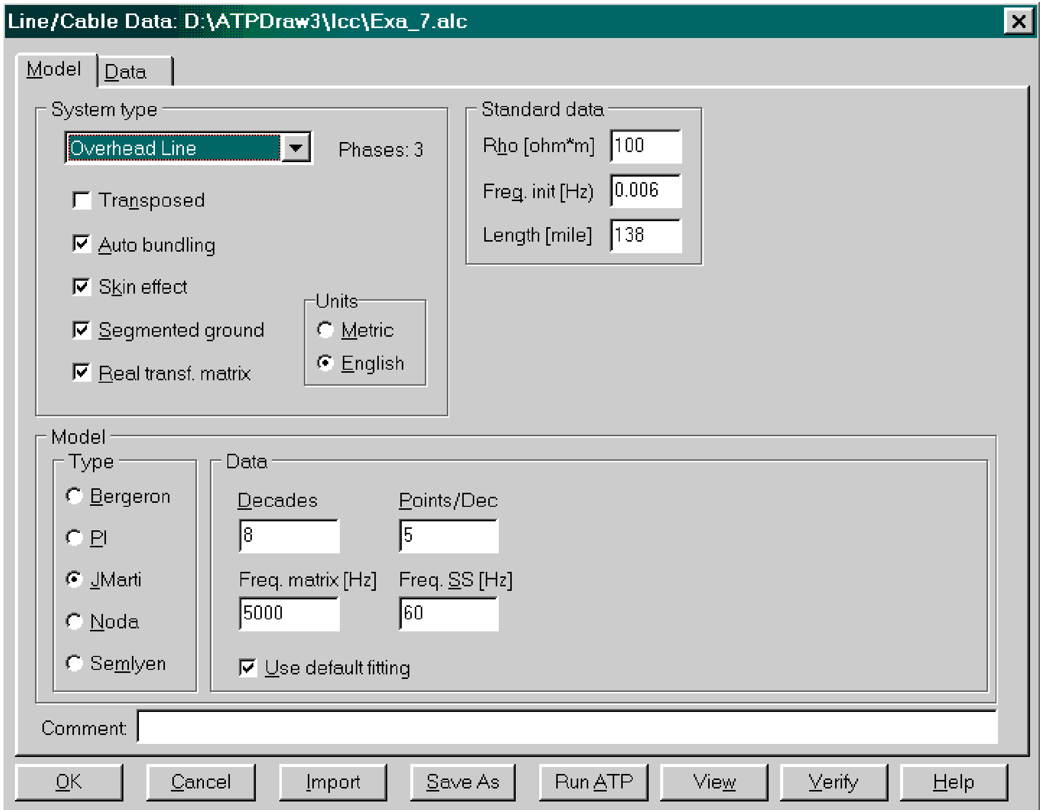


Fig. 6.17 - Line configuration

The geometrical and material data of the system can be specified in a notebook style *Line/Cable Data* window. This window appears when you click on the LCC object with the right mouse button. The case `Exa_7.adp` requires a 3-phase un-transposed, frequency dependent line model with segmented sky wires.

The JMarti line model is fitted in a frequency domain beginning from the standard data parameter *Freq. init* up to an upper frequency limit specified by the mandatory parameter *Decades*. The number of sample points per decade (*Points/Dec*) is given on the *Model* page of the Line/Cable Data dialog box in Fig. 6.18. The model also requires a frequency (*Freq. matrix*) where the transformation matrix is calculated and a steady state frequency (*Freq. SS*). The JMarti model needs in some cases modification of the default fitting data under the optional *Model fitting data* fields, which is not seen here because the default fitting option was selected by checking the *Use default fitting* check box.

For further details please read in Chapter XVII of the ATP Rule Book [3].



Data

	Ph.no.	Rin	Rout	Resis	Horiz	Vtower	Vmid	Separ	Alpha	NB
#		[inch]	[inch]	[ohm/mile DC]	[feet]	[feet]	[feet]	[inch]	[deg]	
1	1	0.2178	0.801	0.05215	-20	50	50	18	0	2
2	2	0.2178	0.801	0.05215	0	77.5	77.5	18	0	2
3	3	0.2178	0.801	0.05215	20	50	50	18	0	2
4	0	0	0.193	2.61	-12.9	98.5	98.5	0	0	0
5	0	0	0.193	2.61	12.9	98.5	98.5	0	0	0

Fig. 6.18 - Model and Data tab of the LCC object with parameters of the 500 kV line.

Click on *OK* or *Run ATP* will produce an ATP-file in the `/LCC` folder for the Line Constants run:

```
BEGIN NEW DATA CASE
JMARTI SETUP
$ERASE
BRANCH  IN___AOUT__AIN___BOUT__BIN___COUT__C
LINE CONSTANTS
ENGLISH
  10.364 0.05215 4            1.602    -20.      50.      50.      18.    0.0          2
  20.364 0.05215 4            1.602     0.0    77.5     77.5      18.    0.0          2
  30.364 0.05215 4            1.602     20.      50.      50.      18.    0.0          2
  0   0.5     2.61 4          0.386   -12.9    98.5     98.5      0.0    0.0          0
  0   0.5     2.61 4          0.386    12.9    98.5     98.5      0.0    0.0          0
BLANK CARD ENDING CONDUCTOR CARDS
    100.       5.E3                                    138.      1           1
    100.        60.                                    138.      1           1
    100.      0.006                                    138.      1   8   5   1
BLANK CARD ENDING FREQUENCY CARDS
BLANK CARD ENDING LINE CONSTANT
DEFAULT
$PUNCH
BLANK CARD ENDING JMARTI SETUP
BEGIN NEW DATA CASE
BLANK CARD
```

Running this file through ATP will produce a punch-file (`Exa_7.pch`) which is not really suitable for human reading, but ATPDraw is able to interpret this file and create a corresponding Data Base Module file from it (`Exa_7.lib`), as shown next.

```
KARD  2  2 31 31 58 58
KARG  1  4  2  5  3  6
KBEG  3  9  3  9  3  9
KEND  8 14  8 14  8 14
KTEX  1  1  1  1  1  1
/BRANCH
-1IN___AOUT__A              2.  0.00              -2 3
      15        4.8596368020077023000E+02
   1.2882691239202958OE+02   4.2131403903614775OE+01   3.2777895265625199OE+02
   3.4212055615475543OE+02   2.3170639589650824OE+02   2.0878626093469154OE+02
   8.6292588620988010OE+02   3.9636478168013964OE+03   1.3775052557123036OE+04
   5.0492226446635977OE+04   1.5426086395089221OE+05   4.8229330415545043OE+05
   1.4212825547149207OE+06   4.3788689414350782OE+06   2.3554548881207641OE+07
   2.4982625099348874OE-01   5.3561571379863138OE-01   4.0728122968037472OE-01
   8.0365103643704283OE-01   1.8127856457046017OE+00   3.5038562176561632OE+00
   1.1405754063642254OE+01   9.8196357914838956OE+01   3.6024293154400004OE+02
   1.3833757172242829OE+03   4.4920760387136297OE+03   1.4830366446875329OE+04
   4.6550147169583703OE+04   1.5344888975550994OE+05   8.7131839689626149OE+05
      23        8.7989087598041335000E-04
   1.4478419597275533OE-03   1.7261890082003288OE-02   6.0805295175568624OE-02
   8.1851456306844969OE-02   9.8036913313247473OE-02   1.1967068082593109OE-01
   1.3123191344559845OE-01   2.0102849766340994OE-01   4.2616544030575892OE-01
   6.7200887547965882OE-01   1.6369054042178495OE+00   1.3506394586639852OE+01
   4.9411343454449444OE+01   4.7211861583780899OE+02   1.0999415740795148OE+02
   3.9808461764936459OE+02   1.9725218531411820OE+03  -3.2527799656935659OE+05
   3.3669985699992796OE+05   1.5221603700733047OE+07  -1.4845957757132890OE+07
   6.8223116334356992OE+06  -7.2123985208627293OE+06
   1.1951946782903891OE+00   1.4187317136820422OE+01   4.9119755582070837OE+01
   6.5845308012319038OE+01   7.9935137618943784OE+01   9.7181431343961975OE+01
   1.1322420249927887OE+02   1.4501637801736263OE+02   1.7797817649474706OE+02
   2.4496316881193894OE+02   3.1208546710802301OE+02   5.9701794168341212OE+02
   1.0540237713020520OE+03   2.7211171115428065OE+03   3.7236943517170371OE+03
   3.3278030654258523OE+03   7.0419868669577636OE+03   1.4552030337712056OE+04
   1.4398597661733342OE+04   2.9660783465008917OE+04   2.9690444248473927OE+04
   2.7565182883627487OE+04   2.7592748066511100OE+04
-2IN___BOUT__B              2.  0.00              -2 3
      23        2.8643042611746085000E+02
   3.0899826091448898OE+02   1.4228805032631267OE+00   6.9059637950597065OE+01
   1.7913661809211629OE+02   3.2452503624158288OE+02   2.9613385079024351OE+02
   3.4182495189017374OE+02   4.1185398380291889OE+02   1.4238258903259339OE+02
   1.2888734004108196OE+02   1.3612550194747570OE+02   6.3577694539832464OE+01
   6.4150506015788523OE+01   4.9782209770622039OE+01   6.7875660136049930OE+01
   8.8346750136556395OE+01   1.4448278806167154OE+03   1.3047437993333117OE+03
   1.3302142241957931OE+03   6.9177103282959288OE+03   4.2478932393957512OE+04
   2.5674163304244276OE+05   9.1180559427318594OE+05
   2.2159964576520674OE-01   2.5587238057882045OE-01   3.2160611358495289OE-01
   6.0079938876346861OE-01   1.1981987382401258OE+00   2.0350189240685594OE+00
   3.4972420970045048OE+00   6.1148787312920350OE+00   9.3669996241826006OE+00
   1.3655669710402602OE+01   2.1246261641853813OE+01   3.4557385495654060OE+01
   6.2006366865975785OE+01   9.0620051952405021OE+01   1.2352235203782720OE+02
   1.5527603591314087OE+02   2.4624359709503678OE+03   2.1970377729105426OE+03
   2.0827577944345044OE+03   1.1441991775110384OE+04   7.0308006033777754OE+04
   4.2547997410248499OE+05   1.5161300323233320OE+06
      12        7.5238143341787320000E-04
   1.2618775233089337OE-02   9.5741730686957123OE+00   8.0655385590171775OE+00
   3.2291678080455092OE+01   1.8228477356618643OE+02   3.2871727349107023OE+03
   1.0546973602151655OE+04   6.2551967432822945OE+04   1.9503758548118628OE+05
   6.9379332782826736OE+05   5.1776813629786789OE+08  -5.1873358555373180OE+08
   2.7994872763437173OE+00   2.1452272486031302OE+03   1.7377141432643684OE+03
   6.7614740519189208OE+03   9.9209934120600155OE+03   6.4147774110826569OE+04
   1.0281753802362646OE+05   2.1364169705337539OE+05   3.6121265920938581OE+05
   1.3033494096500115OE+06   8.0306452606104722OE+05   8.0386759058710933OE+05
-3IN___COUT__C              2.  0.00              -2 3
      16        2.7264948016650357000E+02
   2.6781700549104619OE+02   7.7120492099546894OE+01   1.0539279677448022OE+01
   2.0658228235613805OE+02   2.9247683804661460OE+02   2.5310997492539735OE+02
   3.8872650432586215OE+02   4.8695361163148596OE+02   4.8816640049728178OE+01
```

```
    1.66169955375051840E+02   1.29239096414975530E+02   1.21965197092375600E+02
    5.96392678473994590E+01   2.70618243235187040E+03   6.07857003394846650E+03
    2.37257840791754220E+05
    2.10625410983532740E-01   2.55828054354881100E-01   2.99125254942318730E-01
    6.08748241252635050E-01   1.15829187052544840E+00   1.95077180759634070E+00
    3.59037789940054400E+00   6.45398556690523950E+00   9.01612071865698720E+00
    1.49697084932160820E+01   2.16271239076000870E+01   3.55733792508496410E+01
    6.13057163861530870E+01   2.30400350959367960E+03   5.24672997983652660E+03
    2.04681439259560140E+05
       22        7.58258454233264650000E-04
    2.20731340332155700E-03   8.80044308390158510E-03   1.67547403381400310E-01
    3.38210124111855760E+00   4.84129319769218820E+00   5.09033384432569220E+00
    5.10224088106603180E+00   6.73212550175557480E+00   6.62153387662519320E+00
    9.41335010019846590E+00   4.71274435719055300E+01   3.14107222102525500E+02
    1.15548032274707750E+03   5.54708706438469110E+03   5.75228052962539590E+04
    5.44880732380170960E+05  -3.90321051349684770E+05   1.37373189871937010E+05
    1.65367860810841260E+05   1.18542118698838820E+06   9.64975663973962550E+08
   -9.66683013861545090E+08
    1.05671037383603570E+00   4.22725304647530290E+00   8.05485324487847550E+01
    1.58978414422771360E+03   2.24770604565210580E+03   2.38901260864032970E+03
    2.52338910685068050E+03   3.16422187793352670E+03   3.41412051476461280E+03
    4.72306721626081530E+03   1.07327518402612950E+04   3.92086218410649090E+04
    3.11707656822105780E+04   7.75866280161038010E+04   2.71296348716925770E+05
    4.94008668415039310E+05   5.07878665220420980E+05   7.56313129864476970E+05
    1.19007030519145770E+06   4.62903423922179640E+06   2.78672081158244750E+06
    2.78950753239403340E+06
   0.57153211  0.70710678 -0.41762016
   0.00000000  0.00000000  0.00000000
   0.58881414  0.00000000  0.80696147
   0.00000000  0.00000000  0.00000000
   0.57153211 -0.70710678 -0.41762016
   0.00000000  0.00000000  0.00000000
$EOF
ARG, IN___A, IN___B, IN___C, OUT__A, OUT__B, OUT__C
```

The ATPDraw generated ATP-file for this 500 kV example circuit is show next. The new JMarti LCC object is interfaced with the rest of the circuit by a `$Include` call:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Monday 1, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-5      .2
     500       3       0       0       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
51X0001ABUSA                .55        8.98
52X0001BBUSB               .711      11.857
53X0001CBUSC
  X0001ABUSA               200.                                                0
  X0001BBUSB               200.                                                0
  X0001CBUSC               200.                                                0
        X0008A                          2.51                                   0
        X0008B                          2.51                                   0
        X0008C                          2.51                                   0
        SENA               1.E7                                                0
        SENB               1.E7                                                0
        SENC               1.E7                                                0
$INCLUDE, D:\ATPDRAW3\LCC\EXA_7.LIB, SENA##, SENB##, SENC##, RECA##, RECB## $$
  , RECC##
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  BUSA  SENA      .03333      10.                                              0
  BUSB  SENB       .0361      10.                                              0
```

```
    BUSC  SENC        .0388        10.                                               0
    X0008ABUSA       .13333        10.                                               0
    X0008BBUSB        .1361        10.                                               0
    X0008CBUSC        .1388        10.                                               0
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14X0001A 0   408248.       60.                                  -1.         1.
14X0001B 0   408248.       60.      -120.                       -1.         1.
14X0001C 0   408248.       60.       120.                       -1.         1.
/INITIAL
/OUTPUT
  SENA  SENB  SENC  RECA  RECB  RECC  BUSA  BUSB  BUSC
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

The name of the LIB-file is specified by complete path in this example. If the *Auto path* (*Insert $Prefix and $Suffix*) option would be selected on the *ATP* | *Settings / Format* page, ATPDraw would specify the default prefix (`D:\ATPDraw3\USP\`) and suffix (`.LIB`) in the ATP-file. In that case the LCC files had to be stored in the `/USP` folder (i.e. not in the default `/LCC` folder), otherwise an un-resolvable path conflict would appear, as shown in Fig. 6.19. It is rather preferred *not* to use *Auto path* option if LCC objects are used together with other user specified library objects. If the *Auto path* option is disabled, ATPDraw always enters the full path in the `$Include` call making possible to keep the library files in different directories.

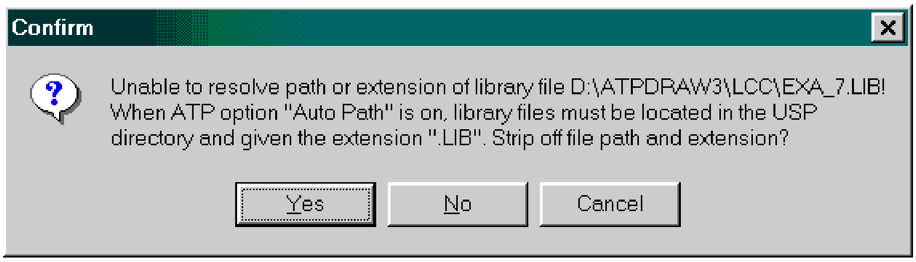


Fig. 6.19- Un-resolvable library file conflict.

Running the ATP data case will produce simulation results as shown in Fig. 6.20.

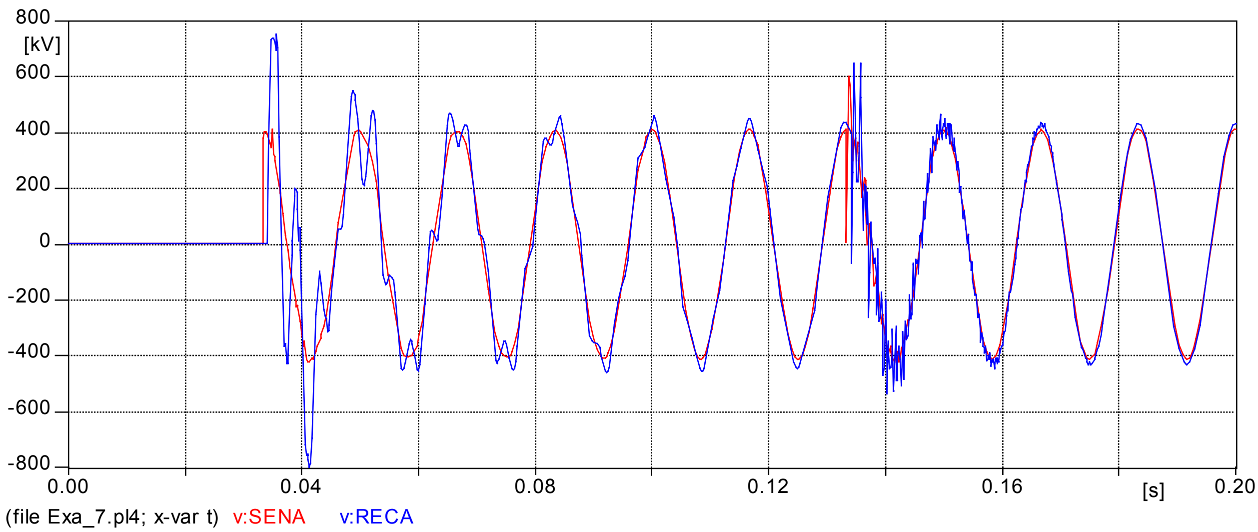


Fig. 6.20 - Calculated voltages at sending and receiving end of the 500 kV line.

6.5.2 JMarti model of a 750 kV line

The JMarti line models introduced in this section will be used in the subsequent single-line-to-ground fault study on a 750 kV shunt compensated transmission line with total length of 487 km. Transpositions separate this line into four sections. Each section of the line is represented by 3-phase un-transposed LCC object with JMarti option enabled. The ATPDraw project of the SLG study includes four such objects with name `LIN750_x.ALC`, where x runs from 1 to 4. The line configuration is shown in Fig. 6.21.

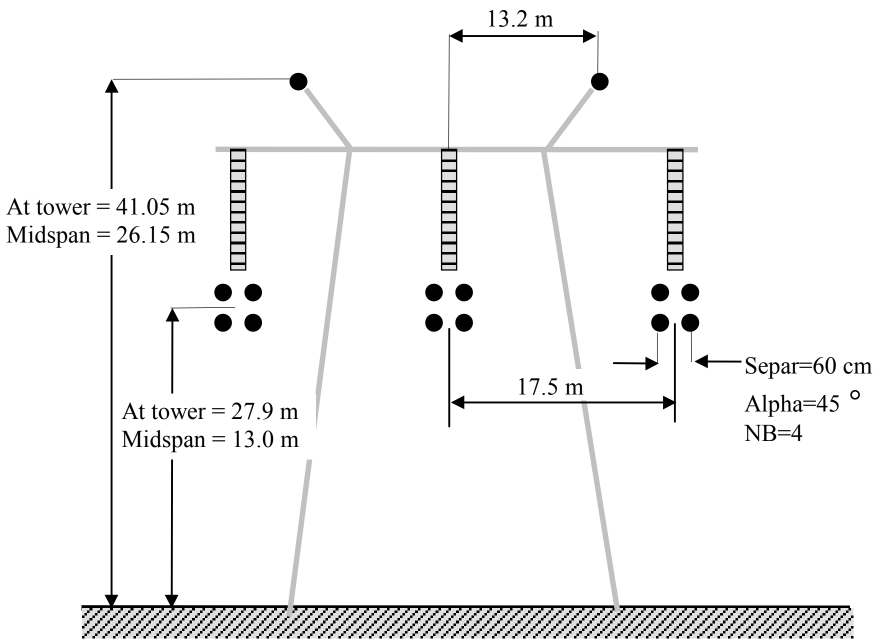


Fig. 6.21 - Tower configuration of the 750 kV line.

The line parameters are given in Metric units. The *Auto bundling* option is enabled to simplify the data entry for this 4 conductor/phase in rectangular arrangement system. Tubular assumption has been applied as in the previous example with the following parameters:

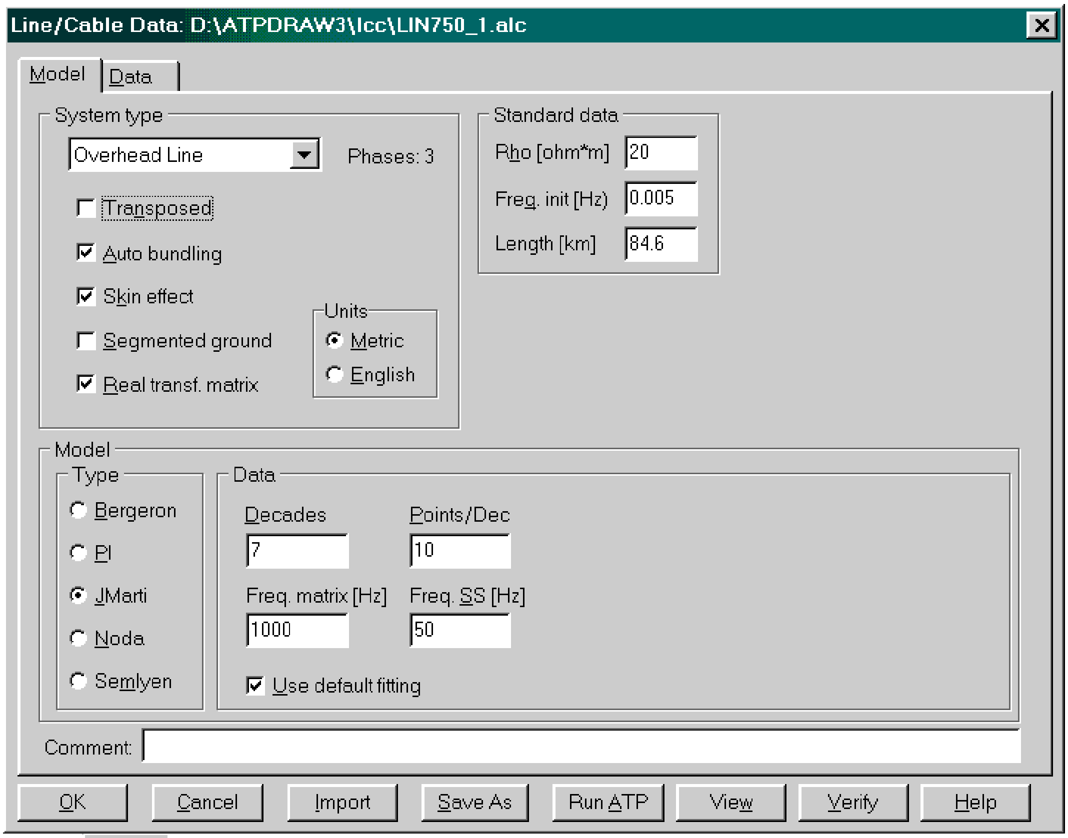
DC resistance = 0.0585 Ω/km
Outside diameter of the conductors = 3.105 cm.
Inner radius of the tube = 0.55 cm
ATPDraw calculates the thickness/diameter value internally (T/D = 0.32).

Sky wires are made from steel reinforced conductors, thus tubular assumption applies here, too:

DC resistance = 0.304 Ω/km
Outside diameter of the sky wire = 1.6 cm
Inner radius of the tube = 0.3 cm
ATPDraw calculates the thickness/diameter value internally (T/D = 0.187).

The resistivity of the soil equals to 20 Ωm. The conductor separation in the bundle is 60 cm.

Entering the geometrical, material data and model options of the line, then executing *Run ATP* will produce a LIB-file in the `/LCC` folder. Since the length of each section is different, four LCC objects with different name are needed. The *Save As* button of the LCC dialog box can be used to save the `.ALC` file with the new length, thus the line parameters need not be entered from scratch.



#	Ph.no.	Rin [cm]	Rout [cm]	Resis [ohm/km DC]	Horiz [m]	Vtower [m]	Vmid [m]	Separ [cm]	Alpha [deg]	NB
1	1	0.55	1.55	0.0585	-17.5	27.9	13	60	45	4
2	2	0.55	1.55	0.0585	0	27.9	13	60	45	4
3	3	0.55	1.55	0.0585	17.5	27.9	13	60	45	4
4	0	0.3	0.8	0.304	-13.2	41.05	26.15	0	0	0
5	0	0.3	0.8	0.304	13.2	41.05	26.15	0	0	0

Fig. 6.22- LCC Model and Data tab of the 1st section of the 750 kV line.

```
BEGIN NEW DATA CASE
JMARTI SETUP
$ERASE
BRANCH  IN___AOUT__AIN___BOUT__BIN___COUT__C
LINE CONSTANTS
METRIC
  10.323  0.0585 4             3.1  -17.5    27.9     13.     60.   45.       4
  20.323  0.0585 4             3.1    0.0    27.9     13.     60.   45.       4
  30.323  0.0585 4             3.1   17.5    27.9     13.     60.   45.       4
  00.313   0.304 4             1.6  -13.2   41.05   26.15     0.0   0.0       0
  00.313   0.304 4             1.6   13.2   41.05   26.15     0.0   0.0       0
BLANK CARD ENDING CONDUCTOR CARDS
     20.      1.E3                        84.6                 1
     20.       50.                        84.6                 1
     20.     0.005                        84.6         7 10    1
BLANK CARD ENDING FREQUENCY CARDS
BLANK CARD ENDING LINE CONSTANT
DEFAULT
$PUNCH
BLANK CARD ENDING JMARTI SETUP
BEGIN NEW DATA CASE
BLANK CARD
```

6.5.3 Line to ground fault and fault tripping transients *(Exa_7a.adp)*

Single-phase to ground fault transients on a 750 kV interconnection are investigated in this study. The one-line diagram of the simulated network is shown in Fig. 6.23. At the sending end of the line shunt reactors are connected with neutral reactors to reduce the secondary arc current during the dead time of the single phase reclosing. The staged fault has been initiated at the receiving end of the line.

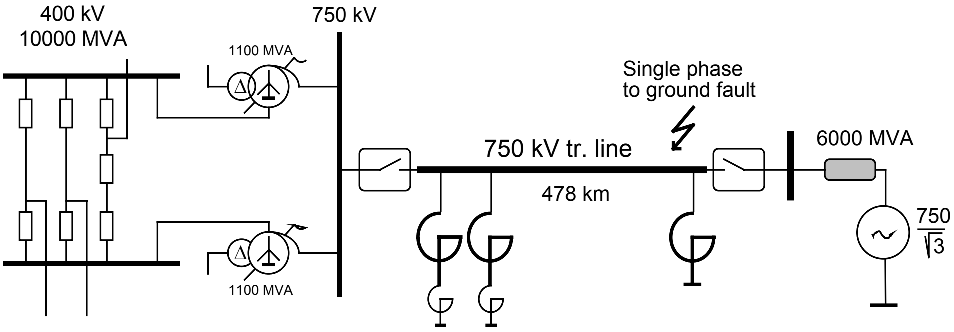


Fig. 6.23 - One line diagram of the faulted line.

The layout of the completed ATPDraw circuit is shown in Fig. 6.24. Along the route three transposition exist, so each LCC object represents a line section between two transpositions with length 84.6 km, 162.7 km, 155.9 km, 75.7 km, respectively.

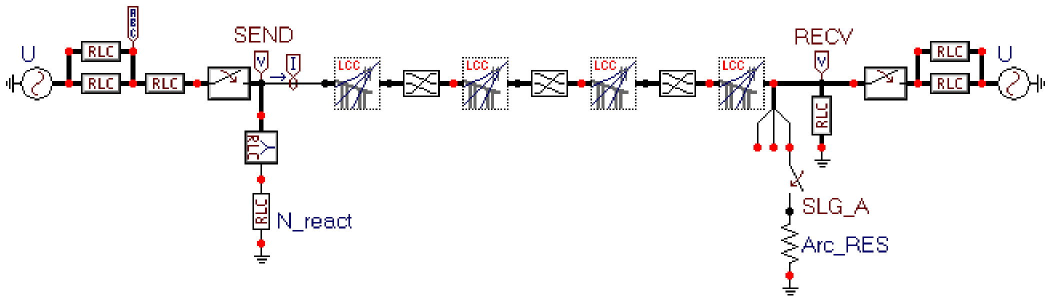


Fig. 6.24 - Line-to-ground fault study (`Exa_7a.adp`)

The supply network model is rather simple: a Thevenin equivalent 50 Hz source and a parallel resistor representing the surge impedance of the lines erected from the 400 kV bus. An uncoupled series reactance simulates the short circuit inductance of the 400/750 kV transformer bank. The single-phase shunt reactors are represented by linear RLC components. Nononlinearities need not been considered here, because the predicted amplitude of the reactor voltage is far below the saturation level of the air gapped core. The impedance of the fault arc is considered as 2 ohm constant resistance.

The ATPDraw generated ATP-file for this 750 kV example circuit is shown next:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Monday 1, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT   >< Tmax >< Xopt >< Copt >
   2.E-5      .5
     500        3        0        0        1        0        0        1        0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
  SLG_A                       2.                                               0
        XX0008                1.  300.                                         0
  X0012CX0014C                5.  180.                                         0
  X0012AX0014A                5.  180.                                         0
  X0012BX0014B                5.  180.                                         0
  X0012CX0014C              150.                                               0
  X0012AX0014A              150.                                               0
  X0012BX0014B              150.                                               0
  X0022CX0021C                5.  300.                                         0
  X0022AX0021A                5.  300.                                         0
  X0022BX0021B                5.  300.                                         0
  X0022CX0021C              150.                                               0
  X0022AX0021A              150.                                               0
  X0022BX0021B              150.                                               0
        RECVC                20.  6.E3                                         0
        RECVA                20.  6.E3                                         0
        RECVB                20.  6.E3                                         0
  X0014CX0017C                2.  200.                                         0
  X0014AX0017A                2.  200.                                         0
  X0014BX0017B                2.  200.                                         0
  SENDC XX0008               10.  3.E3                                         0
  SENDA XX0008               10.  3.E3                                         0
  SENDB XX0008               10.  3.E3                                         0
$INCLUDE, D:\ATPDRAW3\LCC\LIN750_2.LIB, TRAN1B, TRAN1C, TRAN1A, TRAN2B $$
  , TRAN2C, TRAN2A
$INCLUDE, D:\ATPDRAW3\LCC\LIN750_1.LIB, LN1C##, LN1A##, LN1B##, TRAN1C $$
  , TRAN1A, TRAN1B
$INCLUDE, D:\ATPDRAW3\LCC\LIN750_3.LIB, TRAN2A, TRAN2B, TRAN2C, TRAN3A $$
  , TRAN3B, TRAN3C
$INCLUDE, D:\ATPDRAW3\LCC\LIN750_4.LIB, TRAN3C, TRAN3A, TRAN3B, RECVC# $$
  , RECVA#, RECVB#
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  RECVC SLG_A      .0285      .225       10.                                   0
  X0017CSENDC        -1.      .075                                             0
  X0017ASENDA        -1.        1.                                             0
  X0017BSENDB        -1.        1.                                             0
  SENDC LN1C                                         MEASURING                 1
  SENDA LN1A                                         MEASURING                 1
  SENDB LN1B                                         MEASURING                 1
  RECVC X0022C       -1.      .075                                             0
  RECVA X0022A       -1.        1.                                             0
  RECVB X0022B       -1.        1.                                             0
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14X0012C 0   612300.       50.                                    -1.        1.
14X0012A 0   612300.       50.     -120.                          -1.        1.
14X0012B 0   612300.       50.      120.                          -1.        1.
14X0021C 0   612300.       50.       10.                          -1.        1.
14X0021A 0   612300.       50.     -110.                          -1.        1.
14X0021B 0   612300.       50.      130.                          -1.        1.
/INITIAL
/OUTPUT
  SENDC SENDA SENDB RECVC RECVA RECVB
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

Fig. 6.25 shows the results of the simulation. The upper curve is the phase-to-ground voltage at the receiving end of the line. Following the secondary arc extinction an oscillating trapped charge appears on the faulty phase, which is the characteristics of the shunt compensated lines. The blue (lower) curve shows the line current at the faulty phase during the fault and henceforth.

Fig. 6.26 shows the recorded phase voltages and line currents obtained by a high-speed transient recorder at a staged fault tests of the same 750 kV line.

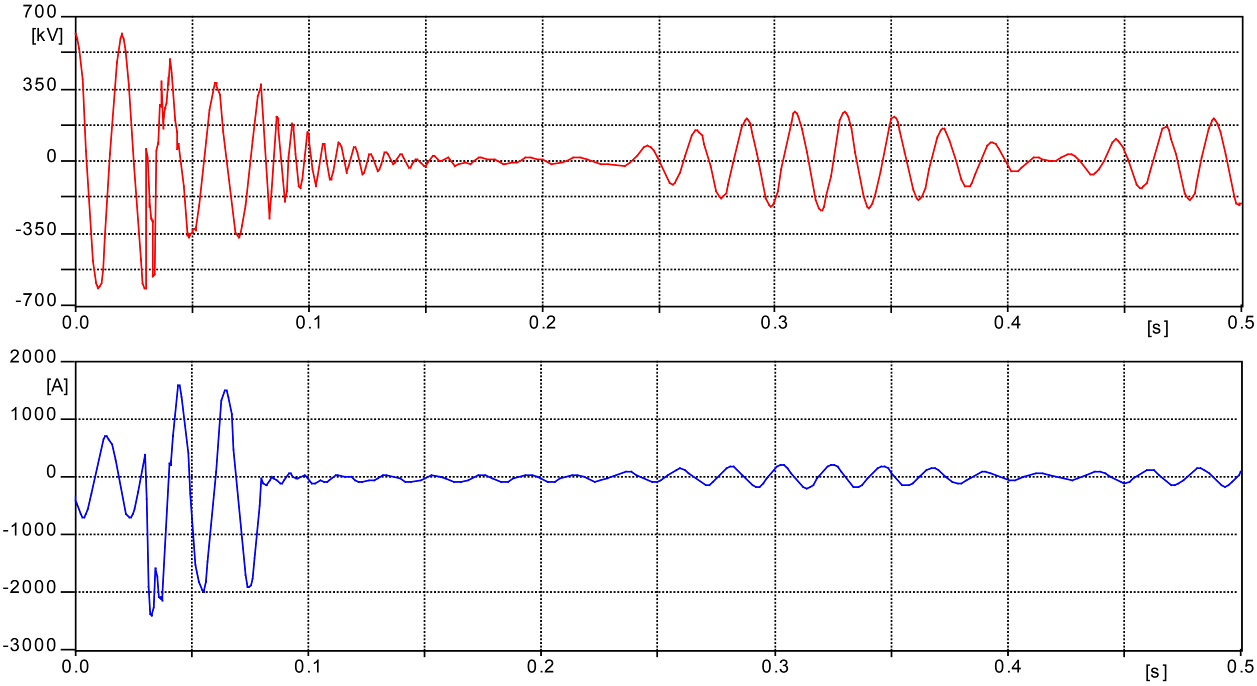


Fig. 6.25 - SLG fault and fault clearing transients (simulation).
upper curve: phase to ground voltage, lower curve: line current

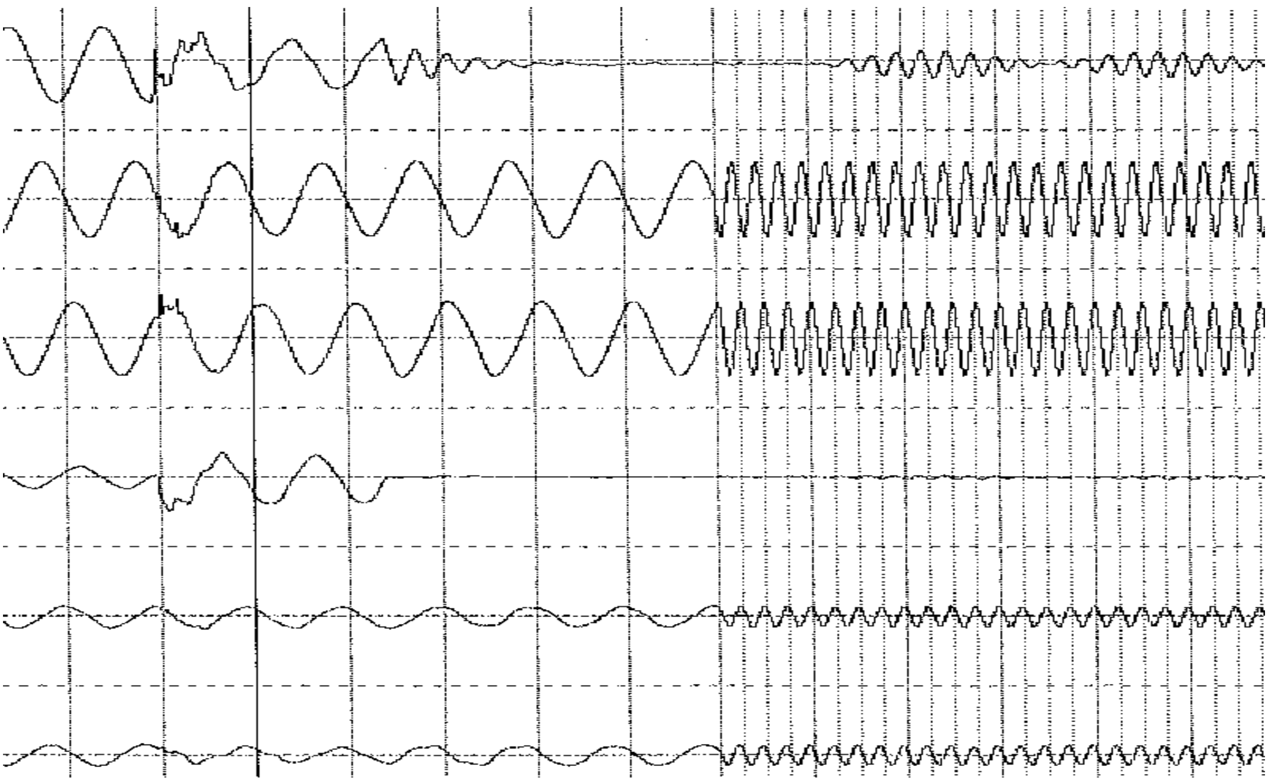


Fig. 6.26 - SLG fault and fault clearing transients. Phase currents and voltages recorded at a staged fault test by a variable sampling frequency disturbance recorder.

6.6 Using MODELS controlled switches (DC68.DAT) *(Exa_8.adp)*

This example is taken from the sub-case 9 of ATP benchmark file `DC68.DAT`, in which a series compensated 500 kV interconnection is outlined. As a rule, metal-oxide arresters protect the series capacitors against overheating in such circuits. In practice the arresters' energy dissipation is continuously monitored and if certain power or energy limits are exceeded the series capacitor is bypassed by a switch. In the simulation, the monitoring function is provided by a MODELS object and the bypass function is by a TACS/MODELS controlled switch. Chapter 5.5.2 of the Advanced Manual introduce you how to create the MODELS object applied in this example. The actual circuit is shown in Fig. 6.27.

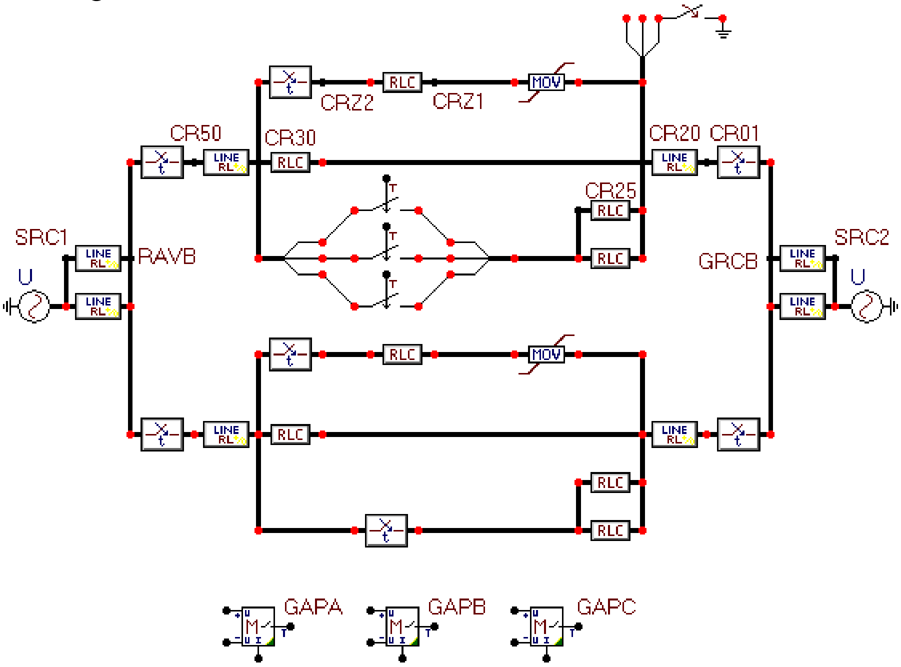


Fig. 6.27 - Usage of MODELS objects in ATPDraw (`Exa_8.adp`)

As Fig. 6.27 shows, no connections are drawn between the three MODELS object and the nodes in the electrical network. This connection is made by the ATP logic by specifying identical name for the MODELS output node and for the control node of the TACS switch. This will result in a warning such as shown in Fig. 6.28 when creating the ATP-file, which can be ignored in this case by clicking on *OK*. If you chose the next button *Abort* then re-draw the circuit the nodes with identical name will be printed in cyan color.

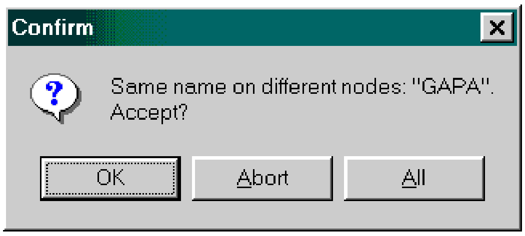


Fig. 6.28 - Warning message that can be ignored here.

If you wish, you can use the *Draw relation* feature (*TACS | Draw relation* of the component selection menu) to visualize the connection between the MODELS objects and the electrical network.

If you click the right mouse button on the object controlling GAPA, an input dialog box appears where you can specify the attributes. The *Model file:* field is the name and path of the file, in which depicts the operation of the object according to the syntax of the MODELS simulation language [4]. The *Use As* field may contain any name, but the name must be different for the three MODELS object in Fig. 6.27.

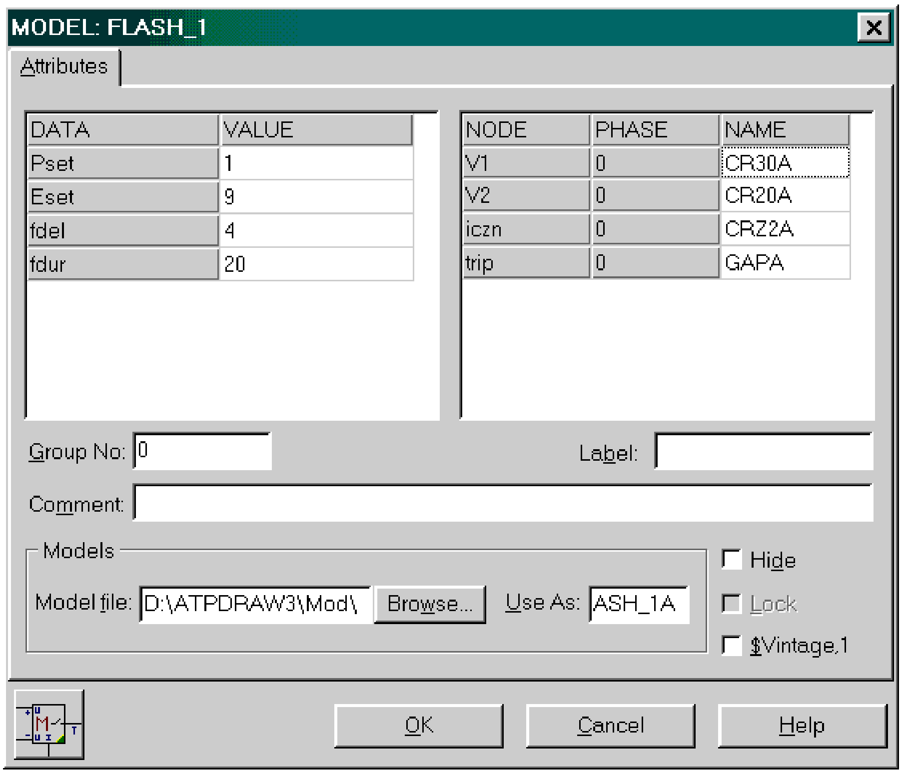


Fig. 6.29 - Attributes of the MODELS object FLASH_1.

If you right click on one of the arresters an input window appears where you can specify the parameters. In section 4.5 of the Reference Manual, the use of the *Attributes* and *Characteristic* page of ZnO arresters are explained in detail.

The characteristics of ZnO arresters cannot be specified directly by exponential functions in ATPDraw, since ATPDraw uses the current/voltage characteristic and performs an exponential fitting internally. An external nonlinear characteristic can be taken into account using $Include, however. This requires a text file named e.g. `ZNO_1.LIB` and stored in the `\USP` directory on the form:

```
                1.0                      40.                       .80
              9999.
```

The name of this LIB-file can be specified in *$Include* field of the *Characteristic* page and the *Include characteristic* button should be selected. The user must be careful with *$Prefix* and *$Suffix* use when the include files are located in different sub-directories. E.g. in this example the nonlinear characteristic of the ZnO blocks are located in the `\USP` folder and the MODELS object description file is in the `\MOD` folder. To avoid conflicts users are advised to specify the full path of all include files, or copy all include files into the same subdirectory `\LIB`.

The new *Grouping* feature of ATPDraw (see in section 5.1 of this manual) provides an alternative way for the TACS switch and MODELS object representation. This circuit is shown in Fig. 6.30, and the name of the project file is `Exa_8g.adp`.

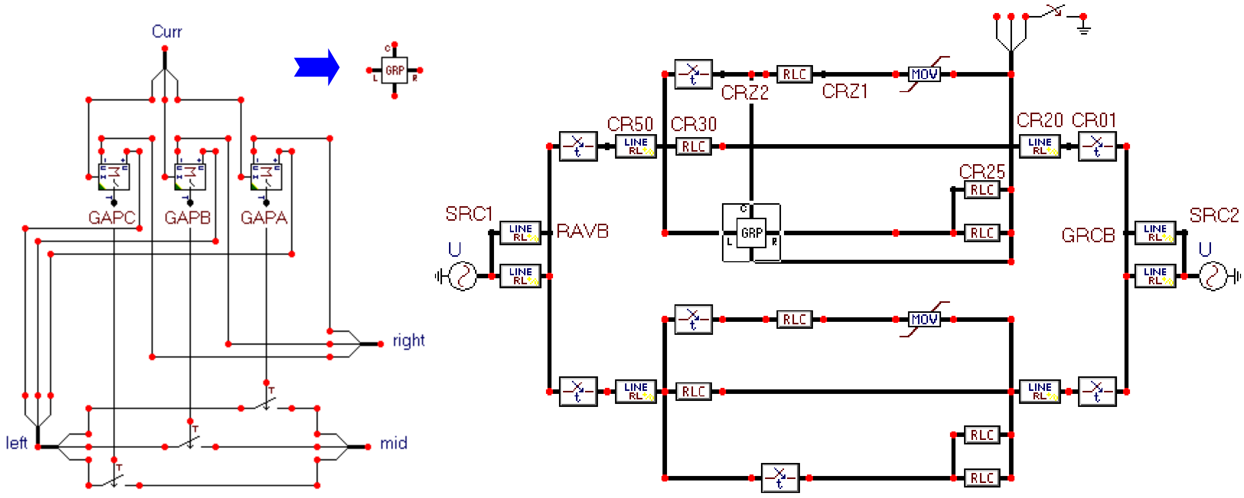


Fig. 6.30 - TACS switch and MODELS object compressed into a group.

The ATP-file of the example circuit is shown on the next 3 pages. The *Record* requests, which result in MODELS variable output in the PL4-file, are marked bold in the data file. These requests can be specified under *ATP* | *Settings / Record* as shown in Fig. 6.31. PRINTER PLOT requests cards, which are also part of the original `DC68.DAT` are not much used today, so were omitted. Similarly the original data case includes a clock-wise rotating 3-phase voltage source (negative sequence), which is not reproduced here. It could be reproduced easily by replacing the 3-phase sources with three single-phase sources of independent delay angle, or make these changes manually by means of the *ATP* | *Edit ATP-file* feature and run the data case by the *Run ATP (file)* command.

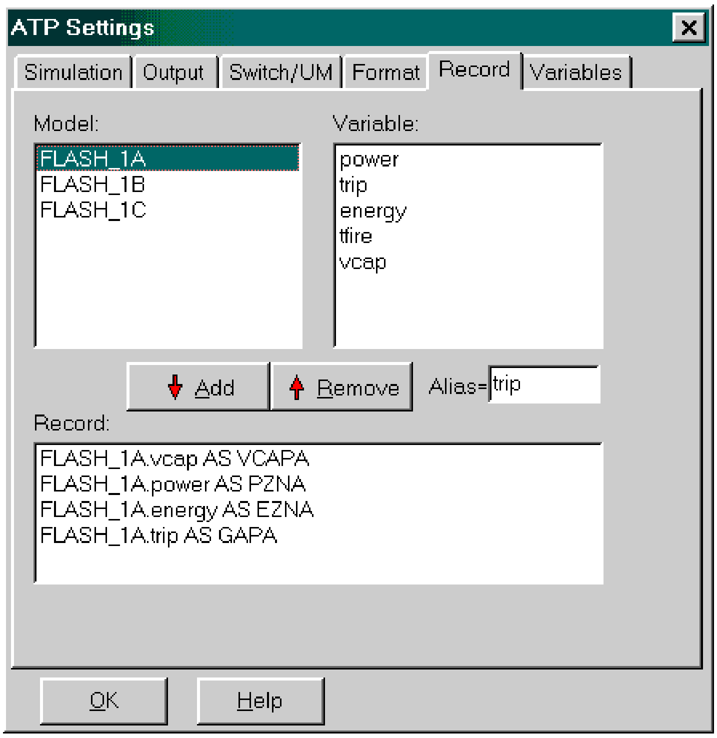


Fig. 6.31 - Selecting MODELS variables for output.

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Wednesday 3, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
C  dT  >< Tmax >< Xopt >< Copt >
   5.E-5     .05     60.
       1       9       0       0       1      -1       0       1       0
       5       5      20      20     100     100       0       0
MODELS
/MODELS
INPUT
  IX0001 {v(CR30A )}
  IX0002 {v(CR20A )}
  IX0003 {i(CRZ2A )}
  IX0004 {v(CR30B )}
  IX0005 {v(CR20B )}
  IX0006 {i(CRZ2B )}
  IX0007 {v(CR30C )}
  IX0008 {v(CR20C )}
  IX0009 {i(CRZ2C )}
OUTPUT
  GAPA
  GAPB
  GAPC
MODEL FLASH_1
comment ****************************************************************
*                                                                      *
*    Function: set or cancel the gap firing control signal             *
*    Inputs  : voltage and current across ZnO resistor                 *
*    Output  : the firing signal to the electrical ZnO component       *
*                                                                      *
************************************************************* endcomment
  INPUT  V1     --  Voltage on positive side of ZNO        [V]
         V2     --  Voltage of negative side of ZNO        [V]
         iczn   --  ZNO current                            [Amps]
  DATA   Pset   --  power setting                          [Megajoules/msec]
         Eset   --  energy setting                         [Megajoules]
         fdel   --  firing delay                           [msec]
         fdur   --  firing duration                        [msec]
  VAR  power    --  power into ZnO resistor                [Watts]
       trip     --  gap firing control signal              [0 or 1]
       energy   --  energy into ZnO resistor               [Joules]
       tfire    --  time at which the gap was last fired   [sec]
       vcap     --  voltage difference across series caps  [Volts]
  OUTPUT  trip
  HISTORY  INTEGRAL(power) {DFLT:0}
  INIT trip:=0
       tfire:=0
  ENDINIT
  EXEC
    -----------------------------------------------------------------
    vcap:=V1-V2
    power:=vcap*iczn
    energy:=INTEGRAL(power)
    -----------------------------------------------------------------
    IF  trip>0              -- is already firing
    AND t-tfire>fdur*1.e-3  -- has exceeded firing duration
    THEN
      trip:=0     -- cancel the firing signal
      tfire:=0    -- null the firing time
    ENDIF
    -----------------------------------------------------------------
    IF trip=0                         -- is not signaling to fire
    AND tfire=0                       -- firing condition not yet detected
    AND (    power  >= Pset * 1.e9    -- power setting exceeded
          OR energy >= Eset * 1.e6 ) -- energy setting exceeded
    THEN
      tfire:=t    -- set the firing detection time
    ENDIF
    IF trip=0                 -- is not signaling to fire
```

```
    AND tfire>0               -- firing condition has been detected
    AND t-tfire>=fdel*1.e-3   -- firing delay exceeded
    THEN
      trip:=1     -- set the firing signal
    ENDIF
  ENDEXEC
ENDMODEL
USE FLASH_1 AS FLASH_1A
INPUT
  V1:= IX0001
  V2:= IX0002
  iczn:= IX0003
DATA
  Pset:=       1.
  Eset:=       9.
  fdel:=       4.
  fdur:=      20.
OUTPUT
  GAPA  :=trip
ENDUSE
USE FLASH_1 AS FLASH_1B
INPUT
  V1:= IX0004
  V2:= IX0005
  iczn:= IX0006
DATA
  Pset:=       1.
  Eset:=       9.
  fdel:=       4.
  fdur:=      20.
OUTPUT
  GAPB  :=trip
ENDUSE
USE FLASH_1 AS FLASH_1C
INPUT
  V1:= IX0007
  V2:= IX0008
  iczn:= IX0009
DATA
  Pset:=       1.
  Eset:=       9.
  fdel:=       4.
  fdur:=      20.
OUTPUT
  GAPC  :=trip
ENDUSE
RECORD
  FLASH_1A.vcap AS VCAPA
  FLASH_1A.power AS PZNA
  FLASH_1A.energy AS EZNA
  FLASH_1A.trip AS GAPA
ENDMODELS
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
51SRC1A RAVBA                 .1        20.
52SRC1B RAVBB                 .1        18.5
53SRC1C RAVBC
51SRC1A RAVBA               300.
52SRC1B RAVBB               150.
53SRC1C RAVBC
51CR50A CR30A                 .5         14.
52CR50B CR30B                 8.         52.
53CR50C CR30C
  CR30A CR20A                           93.4                                   0
  CR30B CR20B                           93.4                                   0
  CR30C CR20C                           93.4                                   0
  CRZ2A CRZ1A                .01                                               1
  CRZ2B CRZ1B                .01                                               1
  CRZ2C CRZ1C                .01                                               1
```

```
92X0102AX0104A                        5555.                                       0
                  147500.                    -1.                                  1
$INCLUDE, D:\ATPDRAW3\USP\ZNO_1.LIB
92X0102BX0104BX0102AX0104A            5555.                                       0
92X0102CX0104CX0102AX0104A            5555.                                       0
$VINTAGE,1
  CR25A CR20A                          5.              5.              5.         0
  CR25B CR20B                          5.             .23                         0
  CR25C CR20C                          5.             .23                         0
  CR25A CR20A                         200.            200.            200.        0
  CR25B CR20B                         200.                                        0
  CR25C CR20C                         200.                                        0
$VINTAGE,0
51CR20A CR01A                 .4          9.
52CR20B CR01B                 5.          36.
53CR20C CR01C
51X0048AX0050A                .5          14.
52X0048BX0050B                8.          52.
53X0048CX0050C
  X0050AX0104A                           93.4                                     0
  X0050BX0104B                           93.4                                     0
  X0050CX0104C                           93.4                                     0
  X0100AX0102A               .01                                                  0
  X0100BX0102B               .01                                                  0
  X0100CX0102C               .01                                                  0
$VINTAGE,1
  X0132AX0104A                         5.              5.              5.         0
  X0132BX0104B                         5.             .23                         0
  X0132CX0104C                         5.             .23                         0
  X0132AX0104A                        200.            200.            200.        0
  X0132BX0104B                        200.                                        0
  X0132CX0104C                        200.                                        0
$VINTAGE,0
51GRCBA SRC2A                 .1          7.
52GRCBB SRC2B                 .1         10.7
53GRCBC SRC2C
51GRCBA SRC2A               350.
52GRCBB SRC2B               150.
53GRCBC SRC2C
51X0104AX0073A                .4          9.
52X0104BX0073B                5.          36.
53X0104CX0073C
92CRZ1A CR20A                         5555.                                       0
                  147500.                    -1.                                  1
$INCLUDE, D:\ATPDRAW3\USP\ZNO_1.LIB
92CRZ1B CR20B CRZ1A CR20A             5555.                                       0
92CRZ1C CR20C CRZ1A CR20A             5555.                                       0
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  RAVBA CR50A        -1.       10.                                                0
  RAVBB CR50B        -1.       10.                                                0
  RAVBC CR50C        -1.       10.                                                0
  CR30A CRZ2A        -1.        1.                                                0
  CR30B CRZ2B        -1.        1.                                                0
  CR30C CRZ2C        -1.        1.                                                0
13CR30A CR25A                                           CLOSED          GAPA      0
13CR30B CR25B                                                           GAPB      0
13CR30C CR25C                                                           GAPC      0
  CR01A GRCBA     -0.006       10.                                                0
  CR01B GRCBB     -0.006       10.                                                0
  CR01C GRCBC     -0.006       10.                                                0
  RAVBA X0048A       -1.       10.                                                0
  RAVBB X0048B       -1.       10.                                                0
  RAVBC X0048C       -1.       10.                                                0
  X0050AX0100A       -1.        1.                                                0
  X0050BX0100B       -1.        1.                                                0
  X0050CX0100C       -1.        1.                                                0
  X0050AX0132A       -1.       10.                                                0
  X0050BX0132B       -1.       10.                                                0
  X0050CX0132C       -1.       10.                                                0
  X0073AGRCBA        -1.       10.                                                0
  X0073BGRCBB        -1.       10.                                                0
```

```
    X0073CGRCBC        -1.       10.                                           0
    CR20A            .01998      10.                                           0
  /SOURCE
  C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
  14SRC1A  0     4.4E5       60.       -20.                              -1.       10.
  14SRC1B  0     4.4E5       60.      -140.                              -1.       10.
  14SRC1C  0     4.4E5       60.       100.                              -1.       10.
  14SRC2A  0     4.4E5       60.                                         -1.       10.
  14SRC2B  0     4.4E5       60.      -120.                              -1.       10.
  14SRC2C  0     4.4E5       60.       120.                              -1.       10.
  /INITIAL
  /OUTPUT
  BLANK MODELS
  BLANK BRANCH
  BLANK SWITCH
  BLANK SOURCE
  BLANK INITIAL
  BLANK OUTPUT
  BLANK PLOT
  BEGIN NEW DATA CASE
  BLANK
```

6.7 Lightning overvoltage study in a 400 kV substation *(Exa_9.adp)*

This example demonstrates the use of ATPDraw in a lightning protection study. The one-line diagram of the investigated 400 kV substation is drawn in Fig. 6.32. The numbers written on the top of the bus sections specify the length in meters. The simulated incident is a single-phase back-flashover caused by a lightning strike to the tower structure 900 m away from the substation. Severe lightning parameters were chosen with 120 kA amplitude and 4/50 µs front/tail times. In the investigated cases, only Line1 and Line2 are connected with the transformer bus. The transformer is protected by conventional SiC arresters.

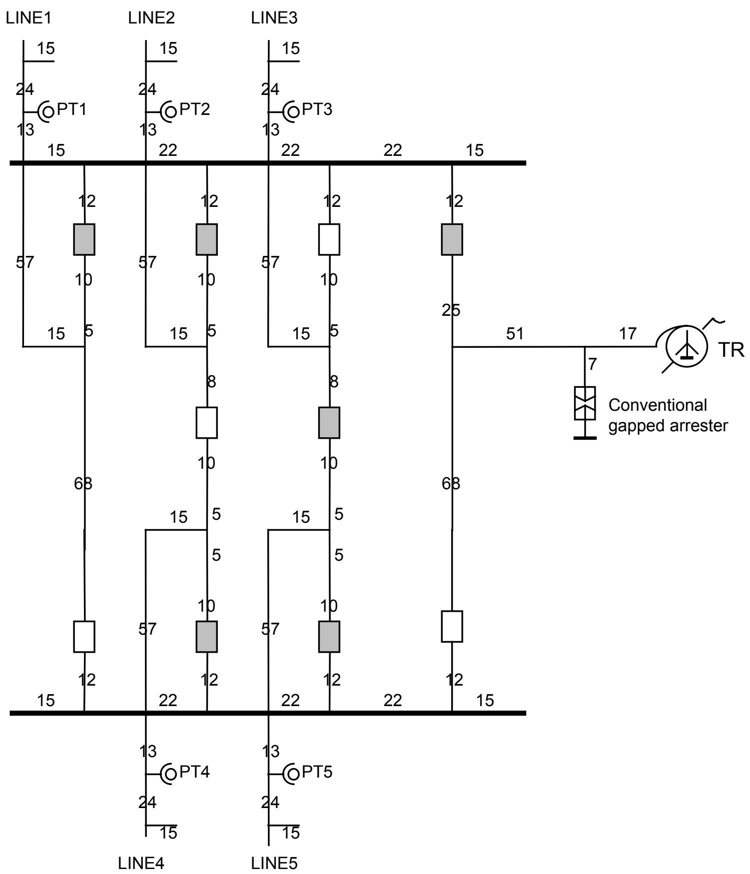


Fig. 6.32 - One-line diagram of the substation

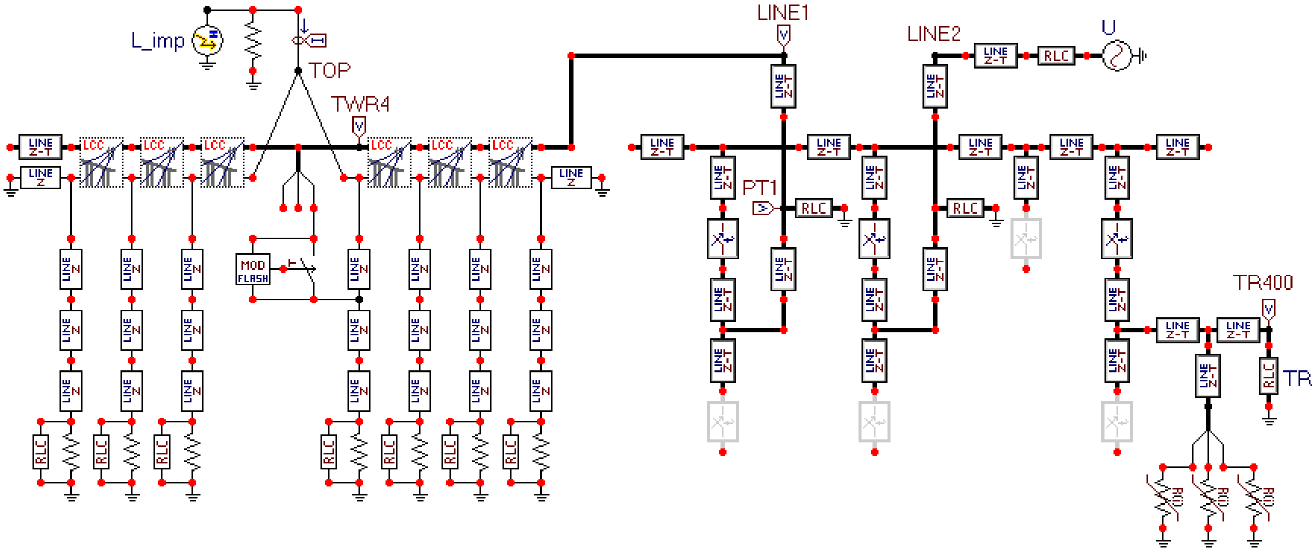


Fig. 6.33 - Example circuit (`Exa_9.adp`)

The ATPDraw circuit of the complete network (substation+incoming line) is shown in Fig. 6.33. The *Copy&Paste* or *Grouping* (*Compress*) feature of ATPDraw could be used effectively when creating such a model because the circuit has many identical blocks. I.e. the user needs to define the object parameters only once and copy them as many times as needed.

Close to the lightning strike, the line spans are represented by 4-phase JMarti LCC objects (phase conductors + sky wire). The surge propagation along the tower structure has been taken into account in this model by representing the vertical pylon sections as single-phase constant parameter transmission lines. The R-L branches below the tower model simulate the tower grounding impedance. The front of wave flashover characteristic of the line insulators plays a significant role in such a back-flashover study. It can be simulated quite easily using a MODELS object - like the `Flash` of this example-, which controls a TACS/MODELS controlled switch. The influence of the power frequency voltage on the back-flashover probability can't be neglected either at this voltage level. In this study case, it was considered by a Thevenin equivalent 3-phase source connected to the remote end of Line2.

The ATP-file created by ATPDraw is shown below. Note! This case exceeds the storage cell limit of ATP if the program runs with DEFAULT=3.0 table size (default `LISTSIZE.DAT` setting). To run the simulation successfully the user must increase this limit from 3.0 to 6.0.

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Thursday 4, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT  >< Tmax >< Xopt >< Copt >
   5.E-9  2.5E-5
     500       3       0       0       1       0       0       1       0
MODELS
/MODELS
INPUT
  IX0001 {v(TWR4A )}
  IX0002 {v(XX0016)}
OUTPUT
  XX0048
```

```
MODEL Flash
comment---------------------------------------
| Front of wave flashover characteristic    |
| of the HV insulator.                      |
| Input: Voltage accross the insulator.     |
| Output: Close command for the TACS switch |
------------------------------------endcomment
INPUT UP, UN
OUTPUT CLOSE
DATA UINF {DFLT:650e3}, UO {DFLT: 1650e3}, TAU {DFLT:8.e-7}, UINIT {DFLT:1E5}
VAR  CLOSE, TT, U, FLASH
INIT
  CLOSE:=0
  TT:=0
  FLASH:=INF
ENDINIT
EXEC
  U:= ABS(UP-UN)
  IF (U>UINIT) THEN
    TT:=TT+timestep
    FLASH:=(UINF + (UO-UINF)*(EXP(-TT/TAU)))
    IF (U>FLASH) THEN CLOSE:=1 ENDIF
  ENDIF
ENDEXEC
ENDMODEL
USE FLASH AS FLASH
INPUT
  UP:= IX0001
  UN:= IX0002
DATA
  UINF:=    1.4E6
  UO:=     3.E6
  TAU:=    8.E-7
  UINIT:=    3.5E5
OUTPUT
  XX0048:=CLOSE
ENDUSE
RECORD
  FLASH.U AS U
  FLASH.CLOSE AS CLOSE
ENDMODELS
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
-1XX0010XX0167              10.  200. 2.5E5  .008 1 0                          0
-1XX0012XX0010              10.  200. 2.5E5  .007 1 0                          0
-1XX0014XX0012              10.  200. 2.5E5  .018 1 0                          0
-1XX0016TOP                 10.  200. 2.5E5  .008 1 0                          0
-1      XX0019              20.  600. 2.9E5    .3 1 0                          0
-1XX0020XX0016              10.  200. 2.5E5  .007 1 0                          0
        XX0014              40.                                                0
        XX0014              13.  .005                                          0
-1XX0026XX0171              10.  200. 2.5E5  .008 1 0                          0
-1XX0028XX0020              10.  200. 2.5E5  .018 1 0                          0
-1X0032AX0033A              20.  650. 2.4E5    3. 1 0                          0
-2X0032BX0033B               2.  400. 2.9E5    3. 1 0                          0
-3X0032CX0033C                                                                 0
        XX0028              40.                                                0
-1XX0036                    20.  600. 2.9E5    .3 1 0                          0
        XX0028              13.  .005                                          0
-1XX0040XX0179              10.  200. 2.5E5  .008 1 0                          0
-1XX0042XX0040              10.  200. 2.5E5  .007 1 0                          0
-1XX0044XX0042              10.  200. 2.5E5  .018 1 0                          0
        XX0044              40.                                                0
        XX0044              13.  .005                                          0
-1XX0054XX0183              10.  200. 2.5E5  .008 1 0                          0
-1XX0056XX0026              10.  200. 2.5E5  .007 1 0                          0
        LIGHT              400.                                                0
-1XX0060XX0054              10.  200. 2.5E5  .007 1 0                          0
-1XX0062XX0056              10.  200. 2.5E5  .018 1 0                          0
```

```
-1XX0064XX0060               10.  200. 2.5E5  .018 1 0                         0
        XX0064               40.                                               0
-1XX0069XX0019               10.  200. 2.5E5  .008 1 0                         0
        XX0064               13.   .005                                        0
-1X0073AX0074A               20.  400. 2.4E5  .008 1 0                         0
-2X0073BX0074B                2.  260. 2.9E5  .008 1 0                         0
-3X0073CX0074C                                                                 0
-1XX0075XX0036               10.  200. 2.5E5  .008 1 0                         0
-1X0078AX0211A               20.  400. 2.4E5  .012 1 0                         0
-2X0078BX0211B                2.  260. 2.9E5  .012 1 0                         0
-3X0078CX0211C                                                                 0
-1X0257AX0081A               50.  650. 2.4E5  .015 1 0                         0
-2X0257BX0081B               10.  360. 2.9E5  .015 1 0                         0
-3X0257CX0081C                                                                 0
-1X0082AX0083A               20.  400. 2.4E5  .068 1 0                         0
-2X0082BX0083B                2.  260. 2.9E5  .068 1 0                         0
-3X0082CX0083C                                                                 0
-1X0271ALINE2A               20.  650. 2.4E5  .024 1 0                         0
-2X0271BLINE2B                2.  360. 2.9E5  .024 1 0                         0
-3X0271CLINE2C                                                                 0
-1X0086AX0269A               20.  400. 2.4E5  .012 1 0                         0
-2X0086BX0269B                2.  260. 2.9E5  .012 1 0                         0
-3X0086CX0269C                                                                 0
-1X0088AX0293A               20.  650. 2.4E5  .015 1 0                         0
-2X0088BX0293B                2.  360. 2.9E5  .015 1 0                         0
-3X0088CX0293C                                                                 0
-1X0074AX0090A               20.  400. 2.4E5  .015 1 0                         0
-2X0074BX0090B                2.  260. 2.9E5  .015 1 0                         0
-3X0074CX0090C                                                                 0
-1X0074AX0271A               20.  400. 2.4E5  .085 1 0                         0
-2X0074BX0271B                2.  260. 2.9E5  .085 1 0                         0
-3X0074CX0271C                                                                 0
  X0271A                               .0005                                   0
  X0271B                               .0005                                   0
  X0271C                               .0005                                   0
-1X0269AX0211A               20.  650. 2.4E5  .022 1 0                         0
-2X0269BX0211B                2.  360. 2.9E5  .022 1 0                         0
-3X0269CX0211C                                                                 0
-1X0211AX0257A               20.  650. 2.4E5  .022 1 0                         0
-2X0211BX0257B                2.  360. 2.9E5  .022 1 0                         0
-3X0211CX0257C                                                                 0
99SICC                     1.1E6    1.                                         1
            100.           6.5E5
            1.E3           7.6E5
            2.E3            8.E5
            4.E3          8.34E5
            5.E3           8.5E5
            1.E4          9.35E5
            2.E4         1.082E6
            3.E4           1.2E6
            9999
-1X0104AX0105A               20.  400. 2.4E5  .068 1 0                         0
-2X0104BX0105B                2.  260. 2.9E5  .068 1 0                         0
-3X0104CX0105C                                                                 0
-1X0106AX0257A               20.  400. 2.4E5  .012 1 0                         0
-2X0106BX0257B                2.  260. 2.9E5  .012 1 0                         0
-3X0106CX0257C                                                                 0
-1X0108ATR400A               20.  650. 2.4E5  .017 1 0                         0
-2X0108BTR400B                2.  360. 2.9E5  .017 1 0                         0
-3X0108CTR400C                                                                 0
-1X0105AX0110A               20.  400. 2.4E5  .025 1 0                         0
-2X0105BX0110B                2.  260. 2.9E5  .025 1 0                         0
-3X0105CX0110C                                                                 0
99SICB                     1.1E6    1.                                         1
            100.           6.5E5
            1.E3           7.6E5
            2.E3            8.E5
            4.E3          8.34E5
            5.E3           8.5E5
            1.E4          9.35E5
            2.E4         1.082E6
            3.E4           1.2E6
```

```
                 9999
-1PT1A  LINE1A            20.  650. 2.4E5  .024 1 0                         0
-2PT1B  LINE1B             2.  360. 2.9E5  .024 1 0                         0
-3PT1C  LINE1C                                                              0
-1X0118AX0293A            20.  400. 2.4E5  .012 1 0                         0
-2X0118BX0293B             2.  260. 2.9E5  .012 1 0                         0
-3X0118CX0293C                                                              0
-1X0083AX0120A            20.  400. 2.4E5  .015 1 0                         0
-2X0083BX0120B             2.  260. 2.9E5  .015 1 0                         0
-3X0083CX0120C                                                              0
        TR400A                        .003                                  0
        TR400B                        .003                                  0
        TR400C                        .003                                  0
-1X0105AX0108A            20.  650. 2.4E5  .051 1 0                         0
-2X0105BX0108B             2.  360. 2.9E5  .051 1 0                         0
-3X0105CX0108C                                                              0
-1SICA  X0108A            20.  400. 2.4E5  .007 1 0                         0
-2SICB  X0108B             2.  260. 2.9E5  .007 1 0                         0
-3SICC  X0108C                                                              0
99SICA                  1.1E6    1.                                         1
            100.        6.5E5
            1.E3        7.6E5
            2.E3         8.E5
            4.E3       8.34E5
            5.E3        8.5E5
            1.E4       9.35E5
            2.E4      1.082E6
            3.E4        1.2E6
            9999
  X0132AX0133A             1.   50.                                         0
  X0132BX0133B             1.   50.                                         0
  X0132CX0133C             1.   50.                                         0
-1XX0135XX0075            10.  200. 2.5E5  .007 1 0                         0
-1X0083APT1A              20.  400. 2.4E5  .085 1 0                         0
-2X0083BPT1B               2.  260. 2.9E5  .085 1 0                         0
-3X0083CPT1C                                                                0
  PT1A                               .0005                                  0
  PT1B                               .0005                                  0
  PT1C                               .0005                                  0
-1X0293AX0269A            20.  650. 2.4E5  .022 1 0                         0
-2X0293BX0269B             2.  360. 2.9E5  .022 1 0                         0
-3X0293CX0269C                                                              0
-1XX0143XX0135            10.  200. 2.5E5  .018 1 0                         0
        XX0062            40.                                               0
        XX0062            13.  .005                                         0
-1XX0149XX0069            10.  200. 2.5E5  .007 1 0                         0
-1XX0151XX0149            10.  200. 2.5E5  .018 1 0                         0
        XX0151            40.                                               0
        XX0151            13.  .005                                         0
        XX0143            40.                                               0
        XX0143            13.  .005                                         0
-1LINE2AX0132A            20.  650. 2.4E5    3. 1 0                         0
-2LINE2BX0132B             2.  360. 2.9E5    3. 1 0                         0
-3LINE2CX0132C                                                              0
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, X0033A, X0033B, X0033C, XX0019, X0166A $$
  , X0166B, X0166C, XX0167
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, X0166A, X0166B, X0166C, XX0167, X0170A $$
  , X0170B, X0170C, XX0171
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, X0170A, X0170B, X0170C, XX0171, TWR4A# $$
  , TWR4B#, TWR4C#, TOP###
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, TWR4A#, TWR4B#, TWR4C#, TOP###, X0178A $$
  , X0178B, X0178C, XX0179
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, X0178A, X0178B, X0178C, XX0179, X0182A $$
  , X0182B, X0182C, XX0183
$INCLUDE, D:\ATPDRAW\LCC\EXA_9.LIB, X0182A, X0182B, X0182C, XX0183, LINE1A $$
  , LINE1B, LINE1C, XX0036
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  LIGHT TOP                                                 MEASURING       1
  X0090AX0086A       -1.     1.001                                          0
  X0090BX0086B       -1.     1.001                                          0
  X0090CX0086C       -1.     1.001                                          0
```

```
    X0110AX0106A       -1.     1.001                                                 0
    X0110BX0106B       -1.     1.001                                                 0
    X0110CX0106C       -1.     1.001                                                 0
    X0120AX0118A       -1.     1.001                                                 0
    X0120BX0118B       -1.     1.001                                                 0
    X0120CX0118C       -1.     1.001                                                 0
  13XX0016TWR4A                                                             XX0048   0
  /SOURCE
  C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><    T1   >< TSTART >< TSTOP  >
  15LIGHT -1      1.2E5     4.E-6     5.E-5        5.                             1.
  14X0133A 0     -3.3E5      50.                                       -1.        1.
  14X0133B 0     -3.3E5      50.      -120.                            -1.        1.
  14X0133C 0     -3.3E5      50.       120.                            -1.        1.
  /INITIAL
  /OUTPUT
    LINE1ALINE1BLINE1CTWR4A TWR4B TWR4C TR400ATR400BTR400CPT1A  PT1B  PT1C
  BLANK MODELS
  BLANK BRANCH
  BLANK SWITCH
  BLANK SOURCE
  BLANK INITIAL
  BLANK OUTPUT
  BLANK PLOT
  BEGIN NEW DATA CASE
  BLANK
```

Some results of the simulation are drawn in Fig. 6.34. The blue line is the voltage stress appearing at the transformer terminal, the red line shows the incoming surge measured at the voltage transformer of Line1 (node PT1 of the circuit). The discharge current of the gapped arrester is drawn at the bottom if the figure. As it can be seen, the instantaneous value of the power frequency voltage was set opposite to the polarity of the lightning surge in the simulation.

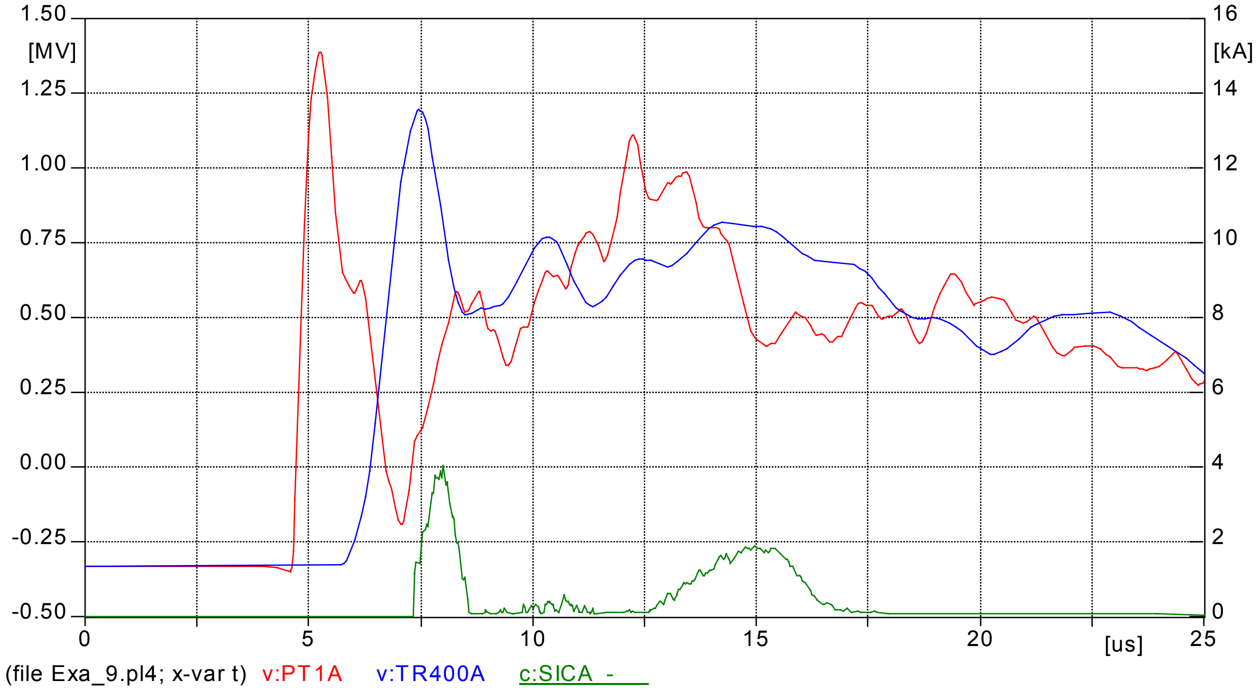


Fig. 6.34 - ATP simulation results. Red: incoming surge at the substation entrance. Blue: voltage stress at the transformer terminal. Green: arrester discharge current.

6.8 Simulating transformer inrush current transients

The magnetic coupling between the windings and the nonlinear characteristic of the magnetizing reactance are the most important factors in transformer energizing transient studies. The BCTRAN supporting routine of ATP can be used to derive the R L or (L^{-1} R) matrix representation of a single or 3-phase multi-winding transformer. ATPDraw now provides a similar interface to the BCTRAN supporting routine like the one provided for the LCC objects. The BCTRAN input data are the excitation and short circuit factory test data, which can easily be obtained from the transformer manufacturers. Additionally, the user can select between several options for modeling the nonlinear magnetizing branch.

The first example circuit of this section demonstrates the use of BCTRAN objects for transformer energization studies. In the second example, readers are familiarized with the application of *user specified objects* and the *Grouping* feature for transformer modeling.

6.8.1 Energization of a 400/132/18 kV auto-transformer *(Exa_10.adp)*

The study case is the energization of a 3-phase, three-winding Yyd coupled transformer. The wye connected 132 kV windings and the delta coupled 18 kV windings are unloaded in this study. The schematic diagram of the simulated case is shown in Fig. 6.35, the corresponding ATPDraw circuit is depicted in Fig. 6.36.

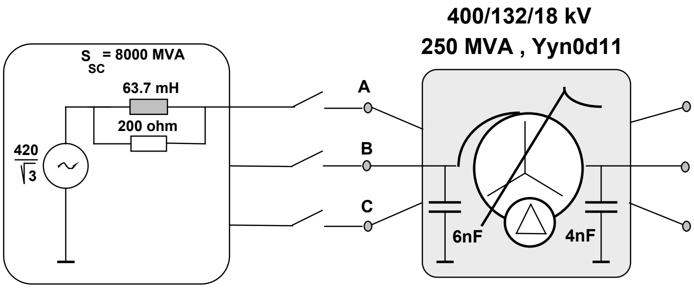


Fig. 6.35 - One-line scheme of the transformer and the 400 kV source.

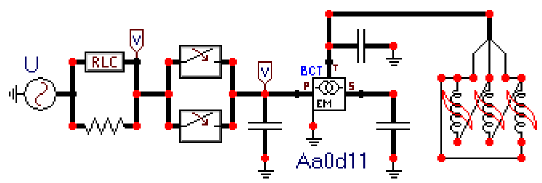


Fig. 6.36 - ATPDraw circuit (`Exa_10.adp`).

The nameplate data of the transformer are as follows:

Voltage rating $V_{high}/V_{low}/V_{tertiary}$:		400/132/18 kV, Yyn0d11	
Power rating:		250 MVA (75 MVA tertiary)	
Positive seq. excitation loss/current:		140 kW / 0.2 %	
Positive seq. reactance:	High to Low:	15 % (S_{base}=250MVA)	15 % (S_{base}=250MVA)
	High to Tertiary:	12.5 % (S_{base}=75MVA)	41.6667 % (S_{base}=250MVA)
	Low to Tertiary:	7.2 % (S_{base}=75MVA)	24 % (S_{base}=250MVA)
Short circuit loss:	High to Low:	710 kW	
	High to Tertiary:	188 kW	
	Low to Tertiary:	159 kW	

In the *BCTRAN* dialog box, you specify first the number of phases and the number of windings per phase under *Structure* (see Fig. 6.37). Under *Ratings,* the nominal line-to-line voltage, power ratings, the type of coupling of windings and the phase shift must be entered. For auto-transformers, the nominal voltage of the *windings* (which is the required input for BCTRAN) is calculated automatically by ATPDraw and the short-circuit impedances are also re-defined according to the Eq. 6.45, 6.46, 6.50 of the EMTP Theory Book [5]. The zero sequence excitation and short circuit parameters are approximately equal to the positive sequence values for an auto-transformer having tertiary delta winding, so the *Zero sequence data available* check boxes are unselected in this example. The *External Lm* option is chosen under *Positive core magnetization* because external Type-96 hysteretic inductors are used to represent the magnetizing inductance. Accordingly, only the resistive component of the magnetizing current will be entered as `IEXPOS` in the BCTRAN input file.

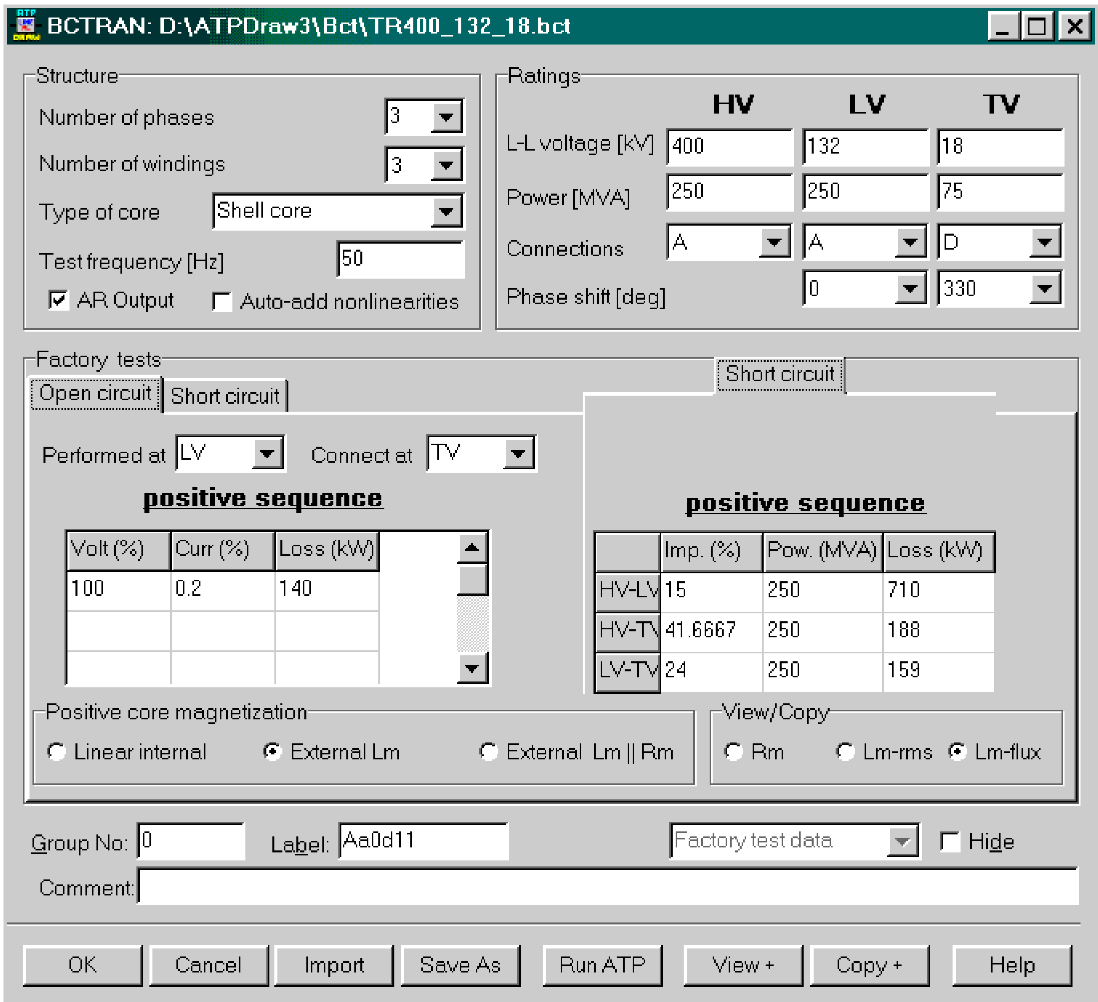


Fig. 6.37 - BCTRAN dialog box of the 400/132/18 kV transformer.

Following data specification the program offers to generate a BCTRAN input file and run ATP. It can either be performed by a *Run ATP* requests, (without leaving the dialog box), or selecting *OK.* If the BCTRAN-file is correct, a punch-file will be created. This file is directly included in the final ATP-file and there is no conversion to a library file as for lines/cables. The BCTRAN input file generated by ATPDraw is shown next. This file is given extension `.atp` and stored in the `/BCT` folder.

```
BEGIN NEW DATA CASE
ACCESS MODULE BCTRAN
$ERASE
C Excitation test data card
C <  FREQ  >< IEXPOS ><  SPOS  >< LEXPOS ><IEXZERO >< SZERO  ><LEXZERO ><><><><>
 3       50. .05600056      250.      140.                                0 2 3 0
C Winding data cards
C ><  VRAT  ><    R    >><  PHASE1  ><  PHASE2  ><  PHASE3  >
 1 154.729872          H_BUSAL_BUSAH_BUSBL_BUSBH_BUSCL_BUSC
 2 76.2102355          L_BUSA      L_BUSB      L_BUSC
 3       18.           T_BUSAT_BUSCT_BUSBT_BUSAT_BUSCT_BUSB
C Short-circuit test data cards
C <><  PIJ   >< ZPOSIJ ><  SPOS  ><ZZEROIJ >< SZERO  ><><>
 1 2       710.33.4150145      250.33.4150145      250. 0 1
 1 3       188.61.3951637      250.61.3951637      250. 0 1
 2 3       159.       24.      250.       24.      250. 0 1
BLANK card ending short-circuit test data
$PUNCH
BLANK card ending BCTRAN data
BEGIN NEW DATA CASE
BLANK CARD
```

The nonlinear magnetizing branch of the 400/132/18 kV auto-transformer is represented by delta coupled Type-96 hysteretic inductors in this study. The flux-current characteristic of these inductors can be obtained by means of the HYSDAT supporting routine of ATP. Fig. 6.38 shows the hysteresis loop of the Itype-1 material of ATP and of the magnetic core of the transformer.

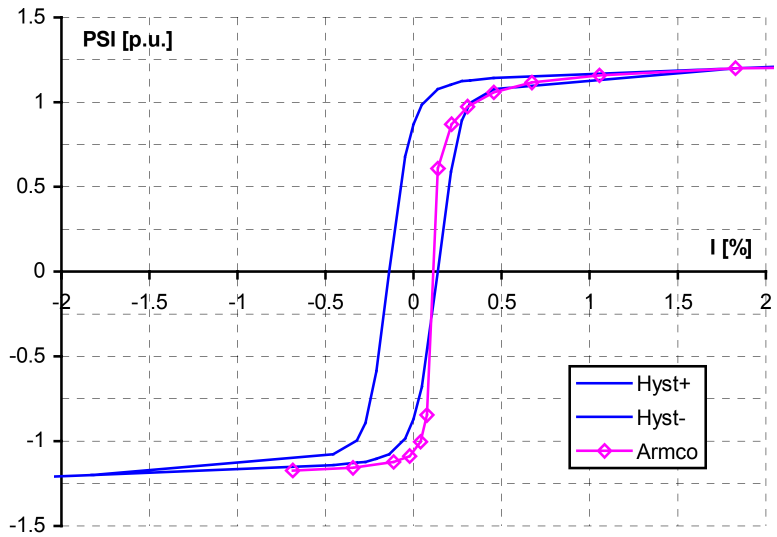


Fig. 6.38 - The shape of the hysteresis loop of the transformer magnetic core compared with the material type 1 of ATP's HYSDAT supporting routine.

The output file generated by the HYSDAT supporting routine is listed below. In this example the file is given a name HYSTR400.LIB and stored in the /USP folder.

```
C  <++++++>  Cards punched by support routine on  21-Jul-02  14.08.23  <++++++>
C HYSTERESIS
C $ERASE
C C  ITYPE   LEVEL   { Request Armco M4 oriented silicon steel -- only 1 availab
C        1       4   { That was ITYPE=1.  As for LEVEL=2, moderate accuracy outp
C     98.2    97.2   { Current and flux coordinates of positive saturat
 -3.68250000E+01 -9.49129412E+01
 -2.45500000E+01 -9.43411765E+01
 -1.10475000E+01 -9.23400000E+01
 -4.91000000E+00 -9.03388235E+01
 -1.84125000E+00 -8.86235294E+01
  6.13750000E-01 -8.51929412E+01
  2.14812500E+00 -8.11905882E+01
  3.55975000E+00 -7.43294118E+01
```

```
4.29625000E+00 -6.28941176E+01
4.91000000E+00 -4.57411765E+01
6.13750000E+00  3.05894118E+01
6.75125000E+00  4.23105882E+01
8.59250000E+00  5.71764706E+01
1.10475000E+01  6.86117647E+01
1.33797500E+01  7.43294118E+01
1.74918750E+01  8.00470588E+01
2.39362500E+01  8.51929412E+01
3.28356250E+01  8.91952941E+01
4.29625000E+01  9.20541176E+01
6.13750000E+01  9.49129412E+01
9.82000000E+01  9.72000000E+01
1.35025000E+02  9.77717647E+01
         9999.
```

Such a nonlinear characteristic can be connected to the Type-96 inductor in two ways: include as an external file, or enter flux-current data pairs directly in the *Characteristic* page as shown in Fig. 6.39. The *Copy* and *Paste* buttons of the dialog box provide a powerful way to import the whole characteristic from an external text file via the Windows clipboard or export it to another Type96 objects. It is thus possible to bring a HYSDAT punch-file up in a text editor, mark the characteristic, copy it to the clipboard and paste it into the *Characteristic* page. The number of data however must be less or equal to 36. In practice it means that you cannot select HYSDAT input parameter *Level* = 4. No such limit exists for the included nonlinear characteristics.

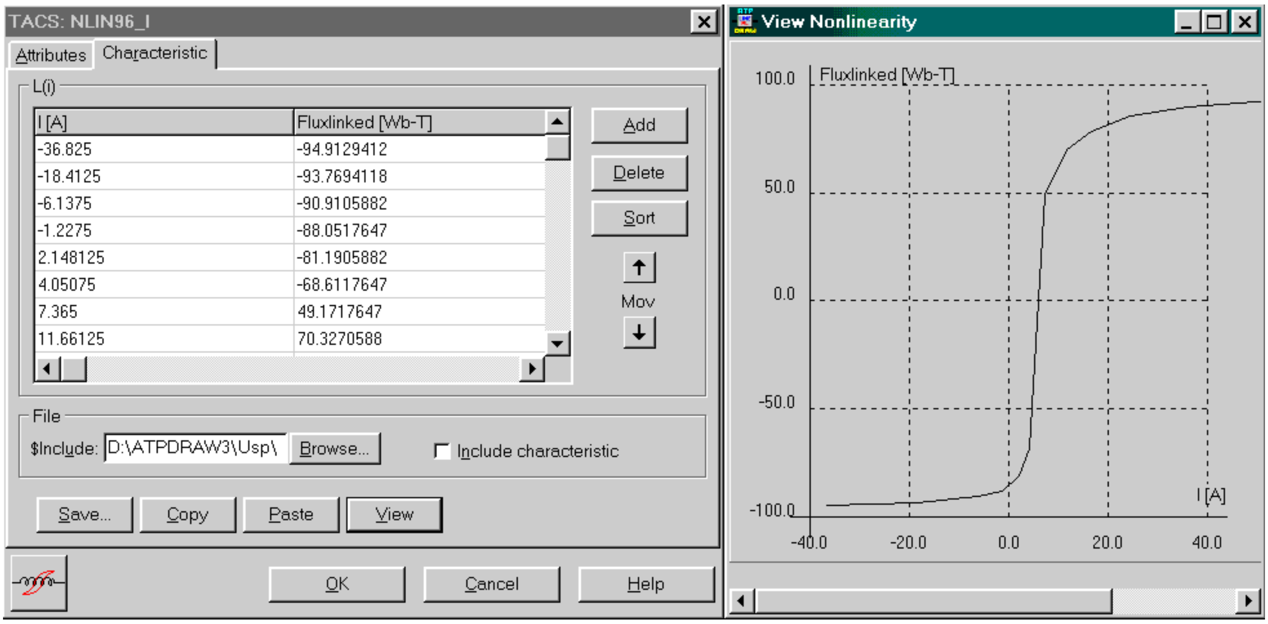


Fig. 6.39 - Importing the nonlinear characteristic from a HYSDAT punch-file.

The complete ATP input file generated by ATPDraw for this study case is listed next:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Sunday 21, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT  >< Tmax >< Xopt >< Copt >
   5.E-6     .15
     500       5       0       0       1       0       0       1       0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
```

```
        L_BUSA                         .004                                    0
        L_BUSB                         .004                                    0
        L_BUSC                         .004                                    0
  SOURCASUPLA               2.  63.7                                           0
  SOURCBSUPLB               2.  63.7                                           0
  SOURCCSUPLC               2.  63.7                                           0
  SOURCASUPLA             200.                                                 0
  SOURCBSUPLB             200.                                                 0
  SOURCCSUPLC             200.                                                 0
  T_BUSA                        .01                                            0
  T_BUSB                        .01                                            0
  T_BUSC                        .01                                            0
96T_BUSBT_BUSC           8888.   0.0                                           1
         -36.825     -94.9129412
        -18.4125     -93.7694118
         -6.1375     -90.9105882
         -1.2275     -88.0517647
        2.148125     -81.1905882
         4.05075     -68.6117647
           7.365      49.1717647
        11.66125      70.3270588
        16.57125      78.9035294
           24.55      85.7647059
        36.21125      90.3388235
          56.465      93.7694118
            98.2            97.2
         135.025      97.7717647
            9999
96T_BUSAT_BUSB           8888.   0.0                                           1
         -36.825     -94.9129412
        -18.4125     -93.7694118
         -6.1375     -90.9105882
         -1.2275     -88.0517647
        2.148125     -81.1905882
         4.05075     -68.6117647
           7.365      49.1717647
        11.66125      70.3270588
        16.57125      78.9035294
           24.55      85.7647059
        36.21125      90.3388235
          56.465      93.7694118
            98.2            97.2
         135.025      97.7717647
            9999
96T_BUSCT_BUSA           8888.   0.0                                           1
         -36.825     -94.9129412
        -18.4125     -93.7694118
         -6.1375     -90.9105882
         -1.2275     -88.0517647
        2.148125     -81.1905882
         4.05075     -68.6117647
           7.365      49.1717647
        11.66125      70.3270588
        16.57125      78.9035294
           24.55      85.7647059
        36.21125      90.3388235
          56.465      93.7694118
            98.2            97.2
         135.025      97.7717647
            9999
        H_BUSA                         .006                                    0
        H_BUSB                         .006                                    0
        H_BUSC                         .006                                    0
$VINTAGE, 1,
 1T_BUSAT_BUSC           6942.8436268432
 2T_BUSBT_BUSA           0.0
                         6942.8436268432
 3T_BUSCT_BUSB           0.0
                         0.0
                         6942.8436268432
  USE AR
 1H_BUSAL_BUSA           3.2888630659697 .42462348721612
```

```
  2L_BUSA                    -7.231251366149             0.0
                             34.681001957452 .09492595191772
  3T_BUSAT_BUSC              2.3450004639366             0.0
                             -84.67537379274             0.0
                             338.34949508527             0.0
  4H_BUSBL_BUSB              .1936225317E-15             0.0
                             -.677127449E-15             0.0
                             .1202491824E-14             0.0
                             3.2888630659697 .42462348721612
  5L_BUSB                    -.677127449E-15             0.0
                             .2041578689E-14             0.0
                             -.282318606E-14             0.0
                             -7.231251366149             0.0
                             34.681001957452 .09492595191772
  6T_BUSBT_BUSA              .1202491824E-14             0.0
                             -.282318606E-14             0.0
                             -.6542678427E-4             0.0
                             2.3450004639366             0.0
                             -84.67537379274             0.0
                             338.34949508527             0.0
  7H_BUSCL_BUSC              .1936225317E-15             0.0
                             -.677127449E-15             0.0
                             .1202491824E-14             0.0
                             .1936225317E-15             0.0
                             -.677127449E-15             0.0
                             .1202491824E-14             0.0
                             3.2888630659697 .42462348721612
  8L_BUSC                    -.677127449E-15             0.0
                             .2041578689E-14             0.0
                             -.282318606E-14             0.0
                             -.677127449E-15             0.0
                             .2041578689E-14             0.0
                             -.282318606E-14             0.0
                             -7.231251366149             0.0
                             34.681001957452 .09492595191772
  9T_BUSCT_BUSB              .1202491824E-14             0.0
                             -.282318606E-14             0.0
                             -.6542678427E-4             0.0
                             .1202491824E-14             0.0
                             -.282318606E-14             0.0
                             -.6542678427E-4             0.0
                             2.3450004639366             0.0
                             -84.67537379274             0.0
                             338.34949508527             0.0
$VINTAGE, 0,
$UNITS, -1.,-1.
  USE RL
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type   >
  SUPLA H_BUSA       -1.      .045        1.                                  1
  SUPLB H_BUSB       -1.      .045        1.                                  1
  SUPLC H_BUSC       -1.      .045        1.                                  1
  SUPLA H_BUSA     .0735        1.                                            1
  SUPLB H_BUSB     .0785        1.                                            1
  SUPLC H_BUSC     .0785        1.                                            1
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14SOURCA 0   326600.       50.                                     -1.        1.
14SOURCB 0   326600.       50.      -120.                          -1.        1.
14SOURCC 0   326600.       50.       120.                          -1.        1.
/INITIAL
/OUTPUT
  SUPLA SUPLB SUPLC H_BUSAH_BUSBH_BUSC
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BEGIN NEW DATA CASE
BLANK
```

Some results of the simulation are shown in Fig. 6.40. In the reported case, the steady state magnetizing current of the unloaded transformer is interrupted at 45 ms producing high residual flux in two phases. As a result, a high amplitude inrush current may occur at a subsequent transformer energization.

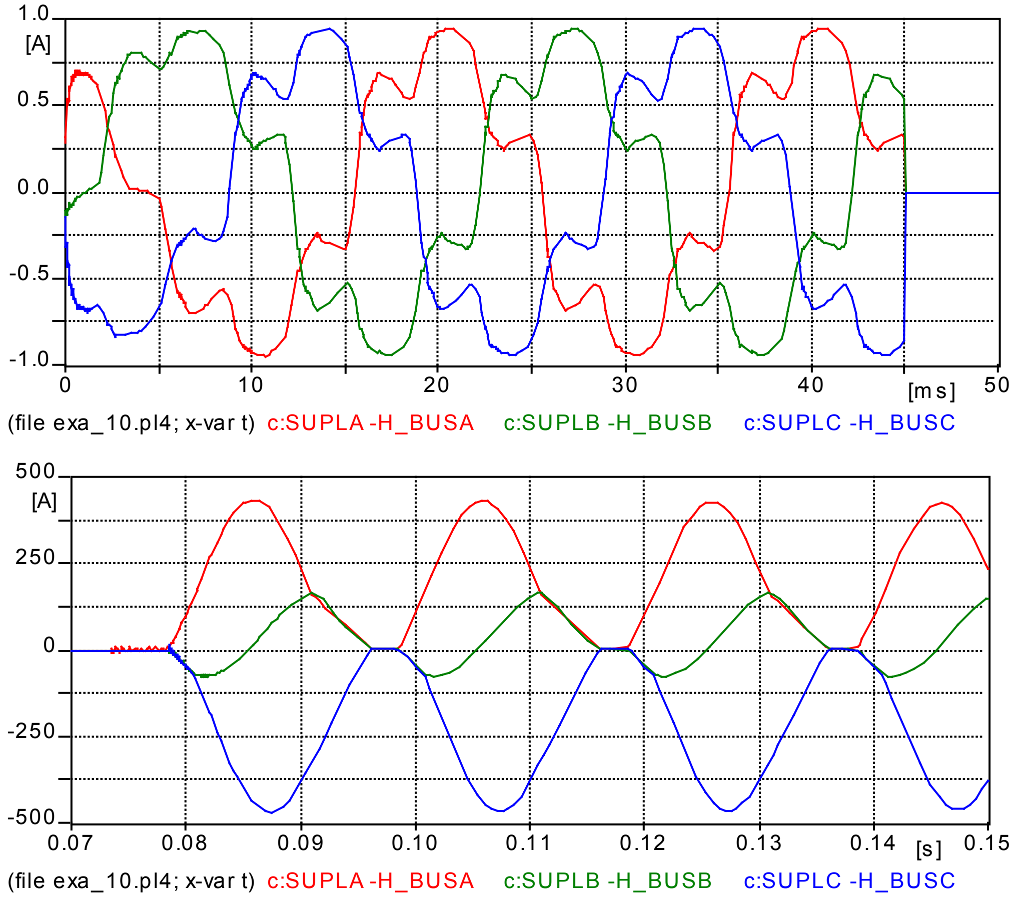


Fig. 6.40 - Steady-state magnetizing current (upper curves) and the inrush current (lower curves) at a subsequent energization.

6.8.2 Energization of a 132/15 kV generator step-up transformer *(Exa_11.adp)*

The use of the icon customization and the advantages of the grouping feature of ATPDraw are demonstrated in this example. The simulated case is again a transformer switching study, in which a 155 MVA 132/15 kV Y/d coupled step-up and a 4 MVA 15/6.9 kV D/d coupled auxiliary transformer are energized together. The fast start gas turbine plant is located near to a 400/220/120 kV substation and the transformers are connected with the substation by a 120 kV single core XLPE cable. During the step-up transformer energization the generator is still disconnected, so need not be considered in this study. The ATPDraw circuit of the simulation is shown in Fig. 6.41.

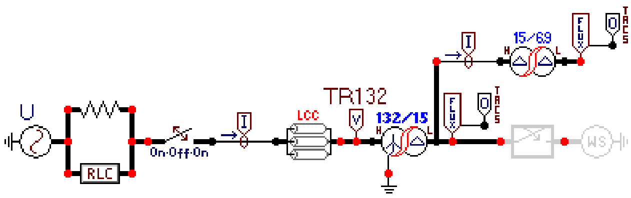


Fig. 6.41 - ATPDraw circuit (`Exa_11g.adp`)

Fig. 6.41 shows several customized ATPDraw objects created by the *Edit* | *Compress* command. If you are not familiar with this grouping feature please read in section 5.1 of this Advanced Manual. This feature provides a powerful tool in advanced modeling. On Fig. 6.41 the nonlinear, hysteretic transformer objects, the parallel connected 3-phase breakers and the TACS objects for flux measurement were compressed into single objects, and the icon of each group has been customized, as well. The icon of some non-group objects were also customized, e.g. the LCC object of the XLPE cable. The uncompressed version of this case is also part of the ATPDraw's example collection and is shown in Fig. 6.42. Therefore, you can see how the grouping feature makes the circuit more readable.

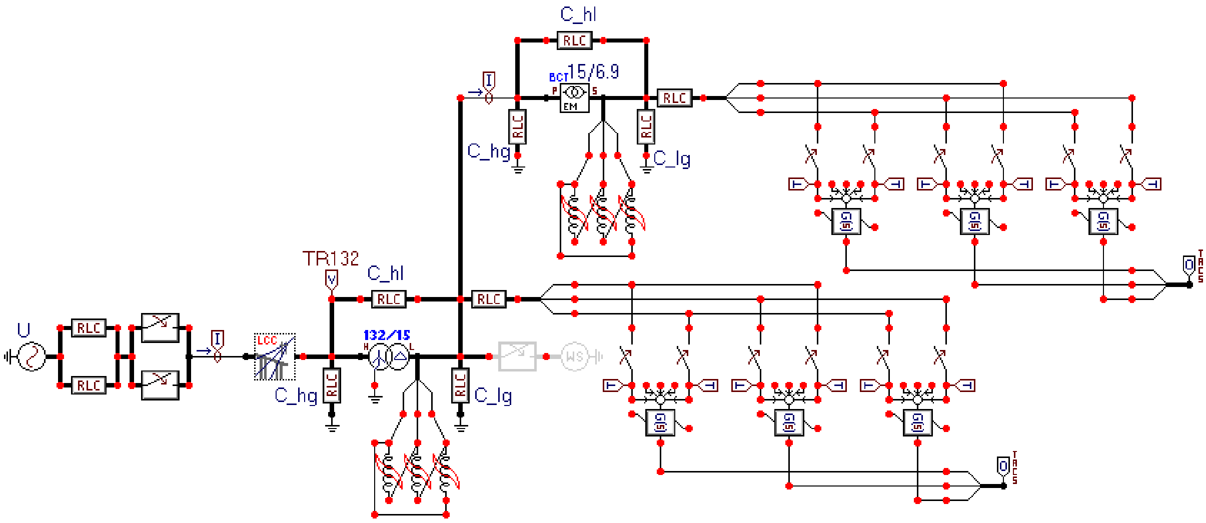


Fig. 6.42 - ATPDraw circuit without using compress (`Exa_11.adp`).

The model of the Ynd11 and the Dd0 transformers consists of a linear part (user specified library object or BCTRAN object) and a nonlinear hysteretic inductor. The capacitances between the transformer windings and ground have been considered, as well. These capacitances do not influence the inrush current significantly, but they need to be taken into account especially at delta coupled transformer terminals to avoid "floating subnetwork found" simulation errors. For more details about the model parameters please read in section 5.8.2 of the Advanced Manual.

The compress option of ATPDraw can be used effectively to create new probe-type objects, as well. The 3-phase *Flux probe* of this example is constructed by integrators (*TACS* | *Transfer functions* | *General*) objects, time controlled switches (to set zero initial conditions) and coupling to TACS objects. The output of the *Flux probe* (the instantaneous flux linkage of the transformer windings) can be used to analyze the operation of the model during steady state no-load conditions, and during the transformer de-energization/re-energization, as shown in Fig. 6.43.

The circuit breaker of the transformer has a common drive with mechanical phase shift of 60 electrical degrees. The making sequence is A-C-B with 3.33 ms delay between the poles and the breaking sequence is B-C-A. Some results of the simulation obtained by the elaborated model are shown next. Fig. 6.44 shows the flux linkage and the phase-to-ground voltages of the step-up transformer during the no-load breaking process. The residual flux is quite low in all phases, thus a subsequent energization will not produce high amplitude inrush current even if the making is done at the voltage zero crossing. When synchronizing the first pole to close with the bus voltage and energize the transformer close to the voltage peak, the inrush current amplitude will not exceed the peak value of the nominal load current of the transformer (see in Fig. 6.45).

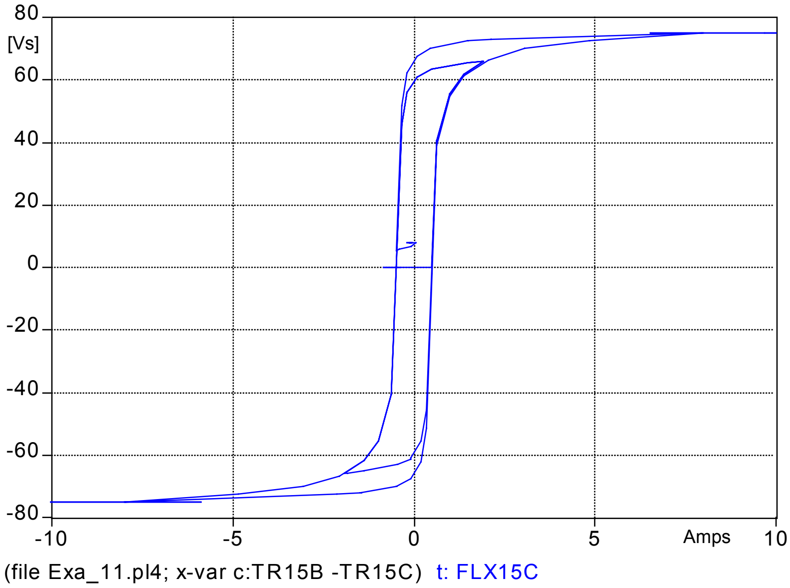


Fig. 6.43 - Roaming of the operating point on the hysteresis loop in steady-state and during the subsequent non-sinusoidal oscillations at transformer de-energization.

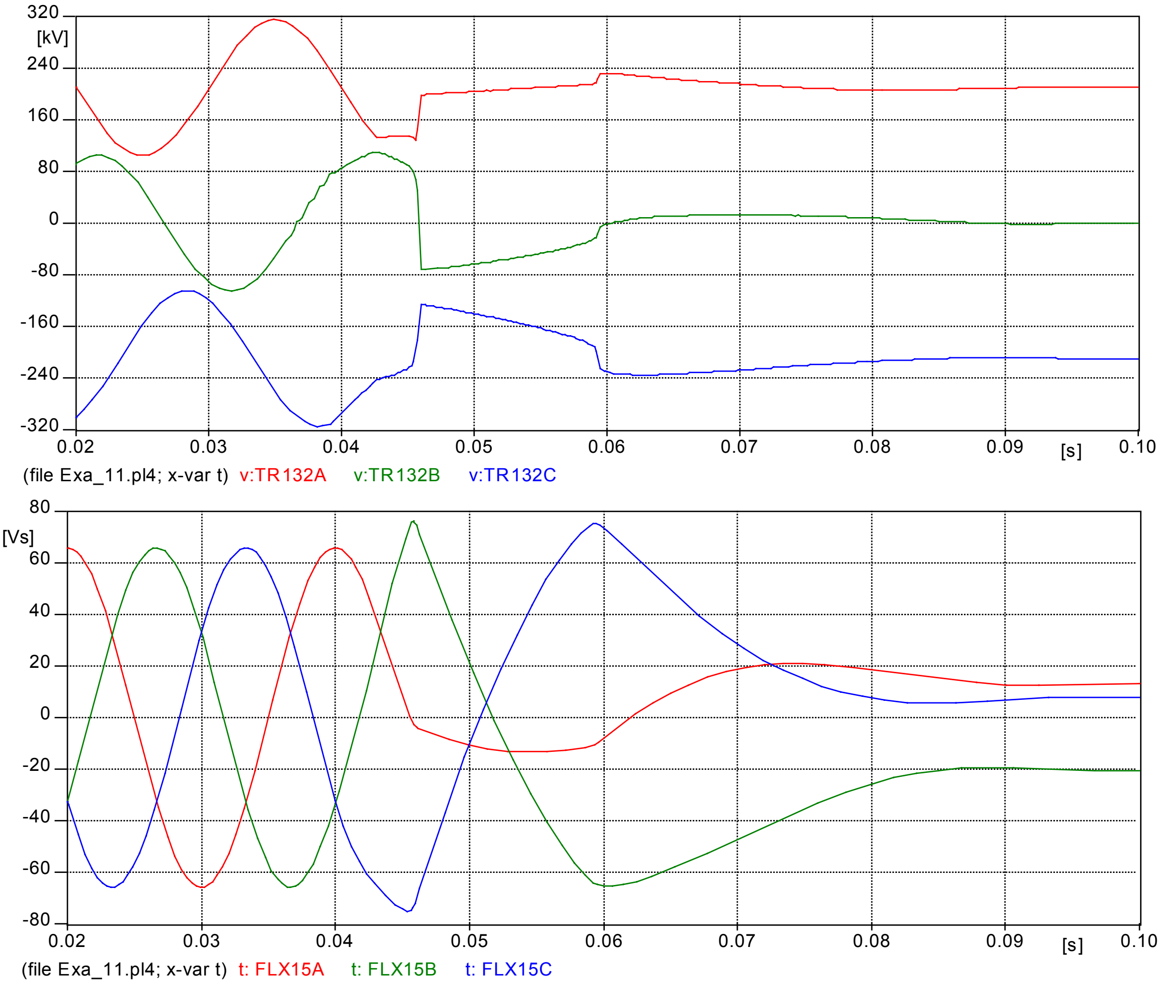


Fig. 6.44 - Non-sinusoidal voltage oscillations appear after de-energizing the step-up transformer (upper curves). The residual flux is less then 30% in each phases (lower curves).

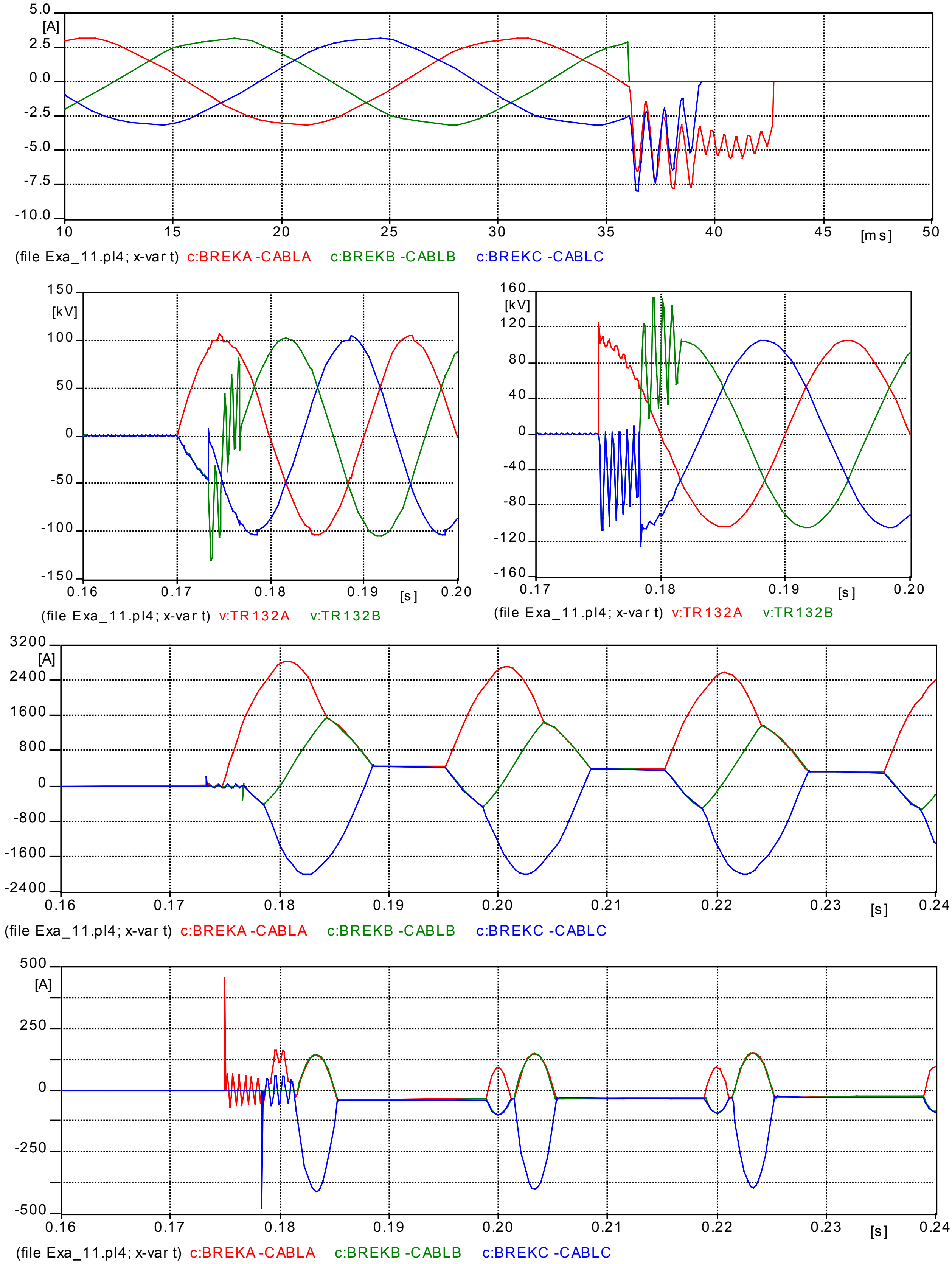


Fig. 6.45 - Interrupting the steady state no-load current of the step-up transformer (upper curves) and the inrush current amplitude (below) when energizing the first pole of the breaker:
a) at the voltage zero crossing, b) close to the voltage peak.

6.9 Switching overvoltage studies with statistical approach (*Exa_12.adp*)

The switching impulse withstand level of EHV line insulators are generally lower than the lightning impulse withstand level. Therefore, some measures are needed to protect the line against switching overvoltages, especially when the insulation level is rather low, like in case of line uprating. One or more of the following measures could be applied to reduce these overvoltages:

- mounting surge arresters at the line terminals and along the line
- application of circuit breaker with closing resistors
- synchronizing the breaker operations at line energization and reclosing
- limiting or eliminating the trapped charge at dead time of the 3-phase reclosing

The influence of the latter two measures to the switching overvoltage distribution is analyzed in this example. The use of the master/slave feature of ATP's statistical switches is also introduced.

The EMTP model shown in Fig. 6.46 has been elaborated for a line upgrading feasibility study to analyze the switching performance of a 400 kV compact line. The clearances, the location of the phase- and ground wires, and the length of the composite insulator strings are assumed known in this example.

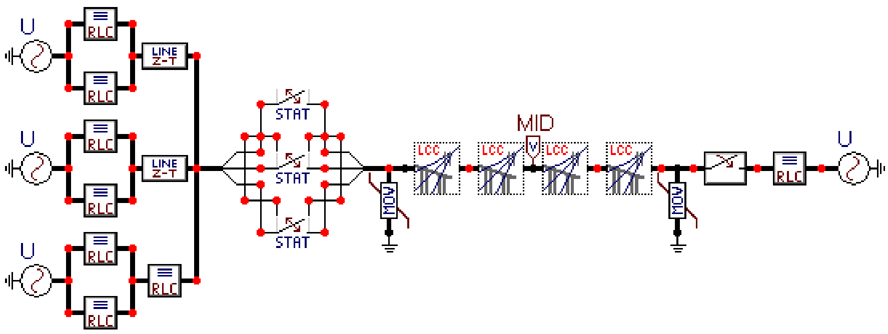


Fig. 6.46 - ATPDraw circuit for the statistical switching study (`Exa_12.adp`).

The investigated line has been divided into four sections, each of them represented by an LCC JMarti object. To set up the initial conditions of the line easily, a 3-phase voltage source is connected to the line at right having voltage amplitude equal to the desired trapped charge. This source is disconnected before the operation of the statistical switches to make the line unloaded. It is worth to mention that some care is needed when constructing the EMTP model for such a statistical simulations, because the unnecessary over-complication of the model may increase the overall simulation time of that many statistical runs significantly.

6.9.1 Setting program options for the statistical simulation

The simulated switching incidence is a 3-phase reclosing in this study. Statistical switches of Gaussian-type represent the reclosing breaker. The master/slave dependency is now supported by ATPDraw, thus phase A is specified as *master* and the remaining two as *slave*. ATP requires the master switch be specified earlier in the ATP-file then a slave. To ensure this ordering, the *Group No:* of a master switch must be set lower than that of a slave, as shown in Fig. 6.47. Additionally, the *Sorting by group number* option must also be activated on the *ATP | Settings/Format* page. The above selections will result in data cards sorted in the ATP-file by class, then sorted by group numbers in each classes.

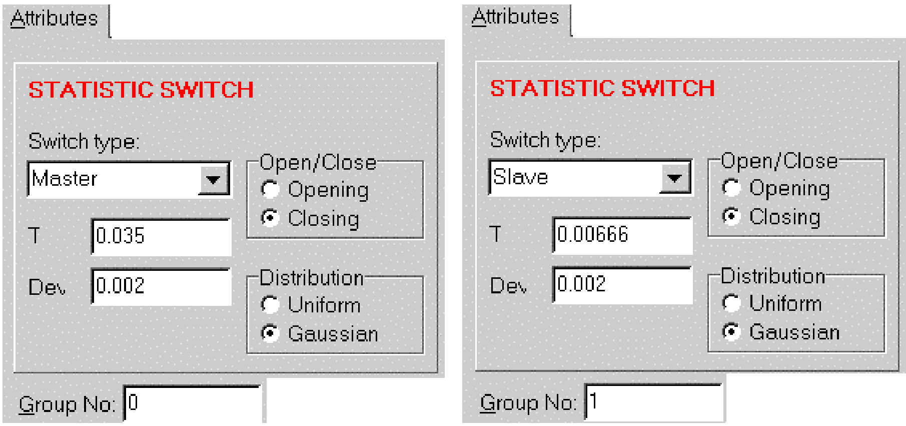
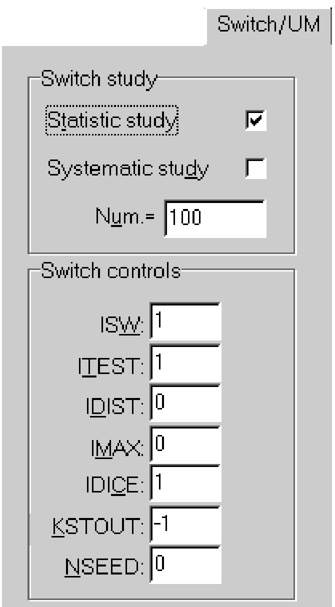


Fig. 6.47 - Input parameters of master and slave statistical switches.

The rest of program options and circuit parameter settings for a statistical study is very similar to that of any other time domain simulations. There is one addition however. You need to specify the *Switch study* and *Switch controls* under *ATP* | *Settings / Switch* before generating the ATP-file.



Unless you need special settings, the *Switch controls* parameters need not be modified. At the time of writing of this manual ATPDraw does not write the `BLANK STATISTICS` request card at the end of the ATP-file. In consequence of the missing request card, the statistical evaluation results do not appear at the end of the LIS-file. To get this valuable part of a statistical simulation, enter this request manually by means of the *Edit ATP-file* feature as shown below, and run the simulation with the corrected file by using the *Run ATP (file)* option. I.e. do not use F2 command key to run the simulation.

```
.....
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BLANK STATISTICS
BEGIN NEW DATA CASE
BLANK
```

Fig. 6.48 - Setting the parameters of the statistical study.

6.9.2 Results of the statistical study

As worst-case assumption the fault, which precedes the 3-phase reclosing in one or more phases has not been considered here. Taking that the inductive voltage transformers play a significant role in eliminating the trapped charge in the healthy phases during the dead time of reclosing, but CVTs or CCVT has no such effect, two different cases have been considered:

a1) the trapped charge is equal to the phase to ground voltage peak
a2) the trapped charge is 30% of the phase to ground voltage peak.

The reclosing operations are synchronized to the bus voltage in this simulation. It means that the master switch is closed when the instantaneous value of the phase-to-ground bus voltage is equal to zero. The average delay for the slave switches in phase B and C is set 120 and 60 electrical degrees, respectively. The standard deviation of the operating time of the synchronous controller and the breaker has been considered as an additional parameter in the study:

b1) accumulated deviation of the breaker and the controller operating time is 1 ms
b2) accumulated deviation of the breaker and the controller operating time is 2 ms.

The ATPDraw generated input file for the case a1) - b1) is shown below:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  July, Thursday 25, 2002
C A Bonneville Power Administration program
C Programmed by H. K. Høidalen at SEfAS - NORWAY 1994-2002
C --------------------------------------------------------
$DUMMY, XYZ000
C  dT  >< Tmax >< Xopt >< Copt >
   1.E-5     .06
     500       1       0       0       1       0       0       1     100
       1       1       0       0       1      -1                               0
C        1         2         3         4         5         6         7         8
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
/BRANCH
C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
C < n 1>< n 2><ref1><ref2>< R  >< A  >< B  ><Leng><><>0
  X0078AX0076A              400.                                               0
  X0078BX0076B              400.                                               0
  X0078CX0076C              400.                                               0
  X0074AX0072A              400.                                               0
  X0074BX0072B              400.                                               0
  X0074CX0072C              400.                                               0
-1X0068AX0001A                1.  700. 2.4E5  170. 1 0                         0
-2X0068BX0001B                .1  400. 2.8E5  170. 1 0                         0
-3X0068CX0001C                                                                 0
-1X0072AX0001A                1.  700. 2.4E5   45. 1 0                         0
-2X0072BX0001B                .1  400. 2.8E5   45. 1 0                         0
-3X0072CX0001C                                                                 0
  X0070AX0068A              400.                                               0
  X0070BX0068B              400.                                               0
  X0070CX0068C              400.                                               0
  X0074AX0072A                2.   50.                                         0
  X0074BX0072B                2.   50.                                         0
  X0074CX0072C                2.   50.                                         0
  X0078AX0076A                5.  125.                                         0
  X0078BX0076B                5.  125.                                         0
  X0078CX0076C                5.  125.                                         0
  X0070AX0068A                2.   65.                                         0
  X0070BX0068B                2.   65.                                         0
  X0070CX0068C                2.   65.                                         0
  X0076AX0001A                5.  225.                                         0
  X0076BX0001B                5.  225.                                         0
  X0076CX0001C                5.  225.                                         0
92      ENDA                       5555.                                       0
                  6.E5                     -1.                                 1
             531.55881849             20.203433531            0.52072093164
             340.85559425             18.920649837            0.70833333333
             217.84036322             16.914334208                      0.8
             266.23677718             19.220050355            0.91666666667
             282.39443055             17.423190053             1.0333333333
              662.1475865             10.787756369             1.1333333333
                  9999
92      ENDB        ENDA           5555.                                       0
92      ENDC        ENDA           5555.                                       0
  X0032AX0033A                2.  100.                                         0
  X0032BX0033B                2.  100.                                         0
  X0032CX0033C                2.  100.                                         0
92      BEGA                       5555.                                       0
                  6.E5                     -1.                                 1
             531.55881849             20.203433531            0.52072093164
             340.85559425             18.920649837            0.70833333333
             217.84036322             16.914334208                      0.8
```

```
              266.23677718            19.220050355           0.91666666667
              282.39443055            17.423190053            1.0333333333
               662.1475865            10.787756369            1.1333333333
                      9999
92      BEGB        BEGA              5555.                              0
92      BEGC        BEGA              5555.                              0
$INCLUDE, D:\ATPDRAW3\LCC\C_400.LIB, BEGA##, BEGB##, BEGC##, X0044A, X0044B $$
  , X0044C
$INCLUDE, D:\ATPDRAW3\LCC\C_400.LIB, X0044A, X0044B, X0044C, MIDA##, MIDB## $$
  , MIDC##
$INCLUDE, D:\ATPDRAW3\LCC\C_400.LIB, MIDA##, MIDB##, MIDC##, X0048A, X0048B $$
  , X0048C
$INCLUDE, D:\ATPDRAW3\LCC\C_400.LIB, X0048A, X0048B, X0048C, ENDA##, ENDB## $$
  , ENDC##
/SWITCH
C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
  ENDA  X0032A       -1.      .001                                       0
  ENDB  X0032B       -1.       .01                                       0
  ENDC  X0032C       -1.       .01                                       0
  X0001ABEGA        .035      .001                  STATISTICSTARGET     0
  X0001BBEGB      .00666      .001                  STATISTICSX0001ABEGA 0
  X0001CBEGC      .00333      .001                  STATISTICSX0001ABEGA 0
/STATISTICS
0             MIDA  MIDB  MIDC
/SOURCE
C < n 1><>< Ampl.  >< Freq.  ><Phase/T0><   A1   ><   T1   >< TSTART >< TSTOP  >
14X0078A 0    3.43E5       50.                                   -1.        1.
14X0078B 0    3.43E5       50.     -120.                         -1.        1.
14X0078C 0    3.43E5       50.      120.                         -1.        1.
14X0074A 0    3.43E5       50.                                   -1.        1.
14X0074B 0    3.43E5       50.     -120.                         -1.        1.
14X0074C 0    3.43E5       50.      120.                         -1.        1.
14X0070A 0    3.43E5       50.                                   -1.        1.
14X0070B 0    3.43E5       50.     -120.                         -1.        1.
14X0070C 0    3.43E5       50.      120.                         -1.        1.
14X0033A 0    3.43E5       50.                                   -1.        1.
14X0033B 0    3.43E5       50.     -120.                         -1.        1.
14X0033C 0    3.43E5       50.      120.                         -1.        1.
/INITIAL
/OUTPUT
  MIDA  MIDB  MIDC
BLANK BRANCH
BLANK SWITCH
BLANK SOURCE
BLANK INITIAL
BLANK OUTPUT
BLANK PLOT
BLANK STATISTICS
BEGIN NEW DATA CASE
BLANK
```

Following the manual addition of the `BLANK STATISTICS` request card, the statistical tabulation of the overvoltage distribution will be part of the LIS-file, as shown next:

```
1 ) ------------------------------------------------------------------------------------------
Statistical output of  node  voltage   0.3430E+06 |0          MIDA  MIDB  MIDC
Statistical distribution of peak voltage at node  "MIDA  ".
The base voltage for per unit printout is  V-base = 3.43000000E+05
  Interval          voltage         voltage  in     Frequency    Cumulative            Per cent
   number         in per unit      physical units   (density)     frequency     .GE. current value
       51          1.2750000       4.37325000E+05           0             0           100.000000
       52          1.3000000       4.45900000E+05           2             2            98.000000
.........
       87          2.1750000       7.46025000E+05           1            99             1.000000
       88          2.2000000       7.54600000E+05           1           100              .000000
Summary of preceding table follows:        Grouped data     Ungrouped data
                              Mean = 1.66850000E+00     1.66882696E+00
                          Variance = 3.85116162E-02     3.81739314E-02
                Standard deviation = 1.96243767E-01     1.95381502E-01
.........
4 ) ------------------------------------------------------------------------------------------
  SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY   SUMMARY
4 ) ------------------------------------------------------------------------------------------
The following is a distribution of peak overvoltages among all output nodes of the last data card that have
the same base voltage.
```

```
This distribution is for the maximum of the peaks at all output nodes with  V-base = 3.43000000E+05
  Interval           voltage          voltage  in     Frequency    Cumulative               Per cent
    number          in per unit      physical units    (density)    frequency     .GE. current value
        51            1.2750000      4.37325000E+05            0             0            100.000000
        52            1.3000000      4.45900000E+05            1             1             99.000000
............
        91            2.2750000      7.80325000E+05            1            99              1.000000
        92            2.3000000      7.88900000E+05            1           100               .000000
Summary of preceding table follows:          Grouped data      Ungrouped data
                                Mean = 1.77125000E+00     1.77305706E+00
                            Variance = 5.25173611E-02     5.27332819E-02
                  Standard deviation = 2.29166667E-01     2.29637283E-01
```

Finally, a brief summary of the simulation results is given next. Considering the metal-oxide arresters with 2 p.u. protection level at both ends of the line, the highest overvoltages appear in the inner points of the line. As an example, Fig. 6.49 shows the probability distribution functions of the switching overvoltages arising in the middle of the line. The four curves correspond to the following cases:

a) Three phase reclosing with 30% trapped charge. Standard deviation of the accumulated operating time of the synchronous controller and the breaker is 1 ms.
b) Three phase reclosing with 100% trapped charge. Standard deviation of the accumulated operating time of the synchronous controller and the breaker is 1 ms.
c) Three phase reclosing with 30% trapped charge. Standard deviation of the accumulated operating time of the synchronous controller and the breaker is 2 ms.
d) Three phase reclosing with 100% trapped charge. Standard deviation of the accumulated operating time of the synchronous controller and the breaker is 2 ms.

As it can be seen, the reclosing overvoltages are quite low even if the trapped charge is close to the voltage peak, if the reclosing operations are synchronized to the bus-side voltage zero by a point on wave controller.

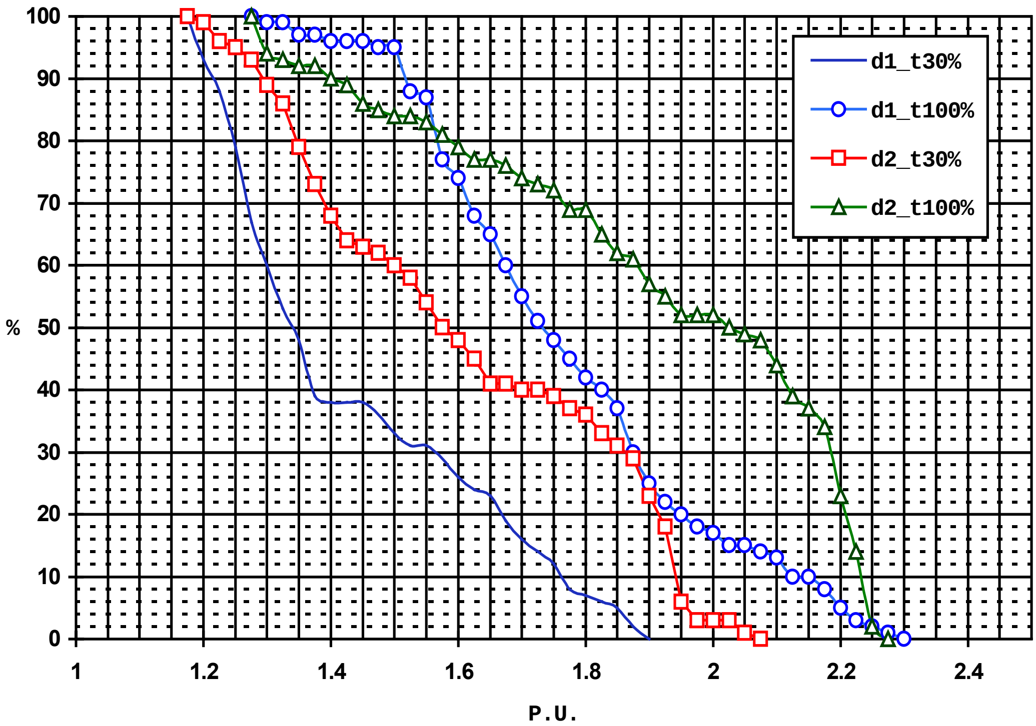


Fig. 6.49- Probability distribution function of the 3-phase reclosing overvoltages.

7. Appendix . . . .

. . . .

ATPDraw™
for Windows
3.5

7.1 ATPDraw.ini file settings

The ATPDraw.ini file is optional and it is specified/edited under *Tools | Options* in ATPDraw. At program startup, all optional program variables are given a default value. Then ATPDraw searches the disk for an initialization file and if found, the new parameters are read from the file into the optional variables, overriding the default values. If no such file is found, the default values apply. ATPDraw will create/update the file on a *Save* or *Load* request under the *Tools | Options* menu. The ATPDraw.ini file has 8 sections:

[ATPDraw]	[Preferences]	[Directories]	[View Options]
[ATP Settings]	[1024x768]	[Reload]	[Objects]

The ATPDraw.ini file is a standard Windows initialization file. Each section contains one or more parameter = value lines. Empty lines or lines beginning with a semicolon (;) are ignored. The following sections list and describe the name and legal value range of available parameters, as well as the default settings (**bold**). One line is required for each parameter and at least one space is needed between the parameter and any in-line comment string.

7.1.1 [ATPDraw]

The [ATPDraw] section contains information on the state and general behavior of ATPDraw. These settings can be modified on the *Tools | Options / General* page as shown in Fig. 7.1.

Parameter	Range	Description
AutoSave	On|**Off**	Enables/disables circuit file auto saving.
AutoSaveInterval	1-60 [**5**]	Specifies the autosave interval in number of minutes.
CreateBackupFile	On|**Off**	Enables/disables the creation of circuit backup files.
SaveWindowSizePos	**On**|Off	Enables/disables the storing of window size and position.
SaveWindowState	On|**Off**	Enables/disables the recording of main window current state
SaveToolbarState	**On**|Off	Enables/disables the recording of toolbar visibility state.
SaveStatusBarState	**On**|Off	Enables/disables the recording of status bar visibility state.
SaveCommentLineSt	On|**Off**	Enables/disables the circuit window comment line visibility.
Win31DialogStyle	On|**Off**	Enables/disables Windows 3.1 style open and save dialogs.
SaveOnExit	**On**|Off	Enables/disables the auto saving of program options on exit.
PolyDots	On|**Off**	Enables/disables the removal of extra points left on screen by the polygon drawing function used to select groups of objects in the circuit window.
PolyBug	On|**Off**	Enables/disables the use of an internal PolyLine function. This is a workaround function that corrects a problem which may appear with some display adapters when a polygon is moved.
DeleteTempFiles	On|**Off**	Controls whether the user specified components are treated as temporary files and deleted from the default directories at program exit or not. <u>Disabled from 3.5p7!</u>

Note that the PolyDots and PolyBug options cannot be set from the *Tools | Options* dialog box. If you experience problems during polygon drawing operations, try to set one or both parameters to On using a text editor.

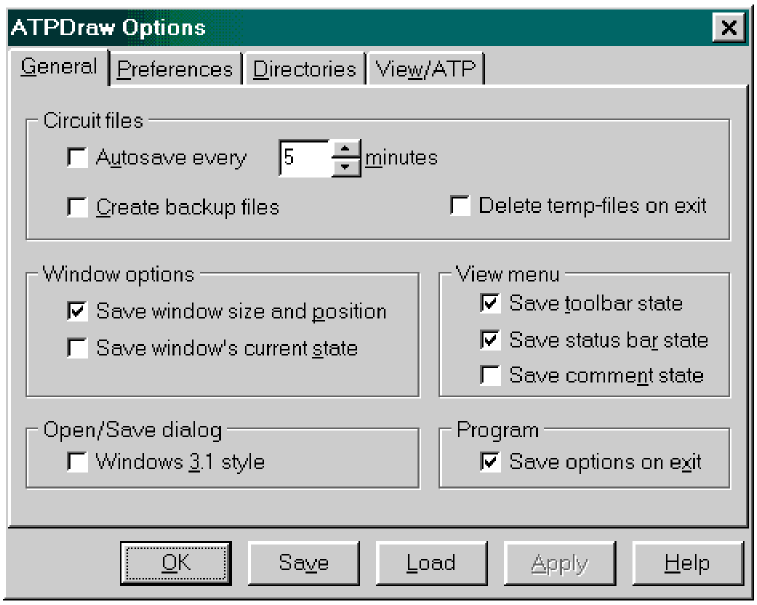


Fig. 7.1 - General program options.

7.1.2 [Preferences]

The [Preferences] section contains information on undo, background color and the standard ATP and ARMAFIT commands. The default settings can be modified on the *Tools* | *Options / Preferences* page (Fig. 7.2).

Parameter	Range	Description
UndoBuffers	1-100 [**10**]	The number of undo/redo buffers to allocate for each circuit window.
BackgroundColor		Circuit windows background color. You can specify one of the system color identifiers or a numeric value to set the red, green and blue color intensity.
TextEditor		Full path of text editor program to use for ATP-file editing. If this parameter is an empty string (default), the built-in text editor is used.
ATPCommand		The default command which executes the ATP simulation.
ArmafitCommand		The default command which executes the ARMAFIT program.

7.1.3 [Directories]

This section contains information on the default directories for projects, ATP-files, and user-generated components. These are the directories suggested by the *Open* and *Save file* dialogs in ATPDraw initially. When a project is opened, ATPDraw will unzip the files on these directories if the stored path does not exist. The user is free to store his components in any directory, however.

Parameter	Range	Description
Projects		The folder name where you store the project files (.adp).
ATP		Specifies the folder in which the ATP-files created (.atp).
Models		Default folder for the MODELS definition (.mod) and

support files (.sup).

Parameter	Range	Description
Group		Container of the compressed GRP object support files (.sup).
UserSpecified		The default folder of the DBM library (.lib) and support files (.sup) for user specified components.
LineCables		Container of the Line/Cable objects related files (.alc, .atp, .lib, .lis, .pch).
Bctran		The default folder for the BCTRAN transformer object related files (.bct, .atp, .lis, .pch).

The default settings can be modified on the *Tools* | *Options / Directories* page (Fig. 7.3).

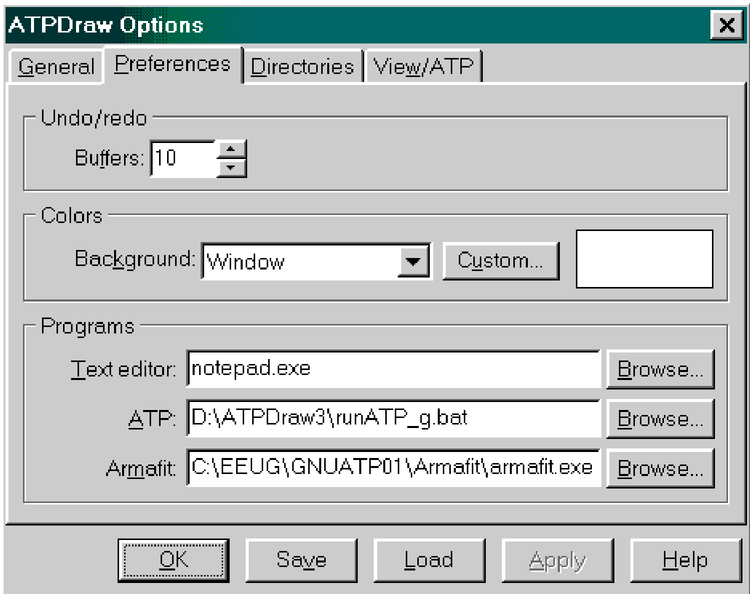


Fig. 7.2 - Specification of commands to execute external programs.

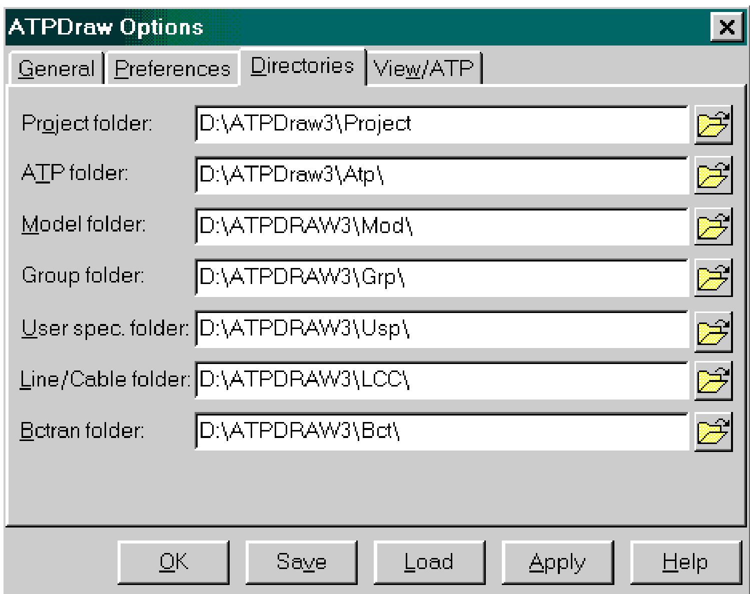


Fig. 7.3- Default location of program folders.

7.1.4 [View Options]

These options control the default appearance of the circuit. The individual settings for the current circuit can be specified under the *View* | *Options* menu. It is possible to select what to draw on the screen by changing the check box status. The default view options for new projects can be specified under the *Tools* | *Options / View/ATP* dialog as shown in Fig. 7.4.

Parameter	Range	Description
NodeNames	On/**Off**	Enables/disables the visibility of node names.
Labels	**On**/Off	Enables/disables the visibility of component labels.
Components	**On**/Off	Enables/disables the visibility of components.
Models	**On**/Off	Enables/disables the visibility of MODELS components.
Tacs	**On**/Off	Enables/disables the visibility of TACS components.
Connections	**On**/Off	Enables/disables the visibility of connection lines.
Relations	**On**/Off	Enables/disables the visibility of relation lines.
NodeDots	**On**/Off	Enables/disables the visibility of node dots (filled circles).
DragIcon	**On**/Off	Enables/disables complete icon drawing during single component or selected group move operations.
NoDataWarning	**On**/Off	Enables/disables the visible warning of components and nodes not opened and given meaningful data.
BranchOutOn=Off	On/**Off**	Enables/disables to show branch output (draw small symbols to show branch output requests.

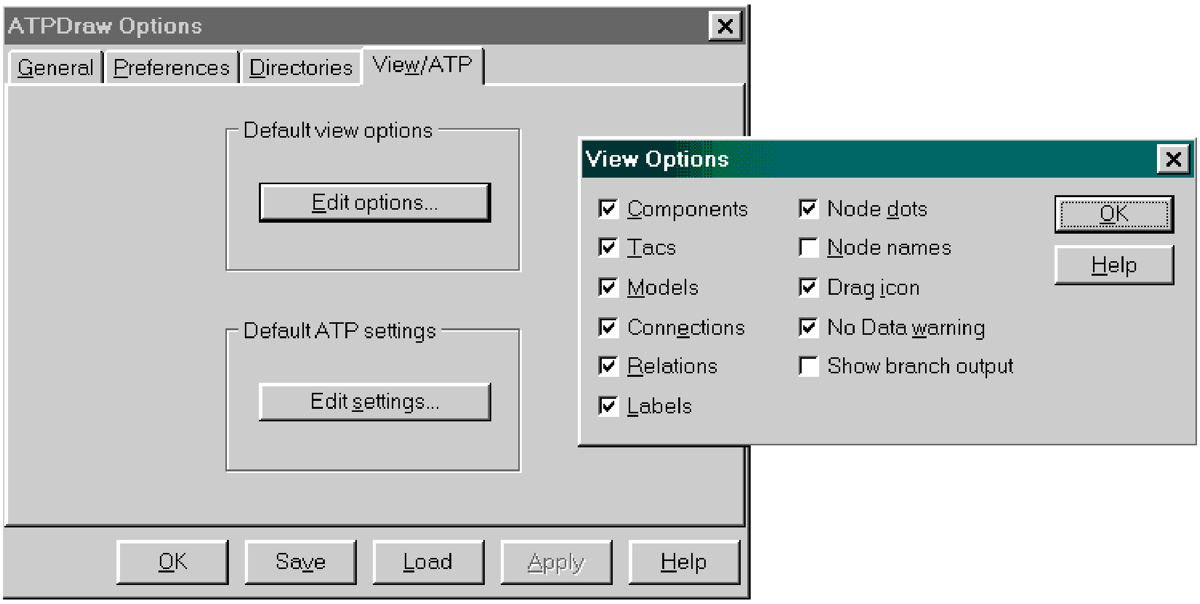


Fig. 7.4 - Setting view options for all new circuits.

7.1.5 [ATP Settings]

Specify the default value of ATP specific variables (initial miscellaneous data cards). The setting for each individual circuit is found under the *ATP* | *Settings* menu. Of special importance is the default time step, simulation length and power frequency. On the *Output* page printout control is set and Auto-detection of simulation errors can also be specified. The default values for new projects can be modified on the *View/ATP* page of the *Tools* | *Options* menu as shown in Fig. 7.5.

Parameter	Type	Description
DeltaT	Real	Time step of simulation in seconds.
Tmax	Real	End time of simulation in seconds.
Xopt	Real	Inductances in [mH] if zero; otherwise, inductances in [Ohm] with Xopt as frequency.
Copt	Real	Capacitances in [uF] if zero; otherwise, capacitances in [uMho] with Copt as frequency.
SysFreq	Real	System frequency in Hz.
IOut	Integer	Frequency of LUNIT6 output within the time-step loop. For example, 3 means that every third time step will be printed.
IPlot	Integer	Frequency of saving solution points to the PL4 output file. For example, a value of 2 means that every second time step will be written to the PL4-file.
IDouble	Integer	If 1, table of connectivity written in the LUNIT6 output file. If 0 (zero), no such table written.
KssOut	Integer	Controls steady state printout to the LUNIT6 output file.
MaxOut	Integer	If 1, extrema printed at the end of the LUNIT6 output file.
IPun	Integer	Flag that request additional card for the IOUT frequency.
MemSave	Integer	Controls the dumping of EMTP tables to disk at the end of simulation if START AGAIN request is specified.
ICat	Integer	Controls saving of raw plot data points that is written to the I/O channel LUNIT4. Possible values are:
Nenerg	Integer	Number of simulations. A value of 0 (zero) means single, deterministic simulation; otherwise, statistic switch study.
ISW	Integer	If 1, printout of all variable switch closing/opening time to LUNIT6.
ITEST	Integer	Extra random delay is added to all switch operations in statistical studies.
IDIST	Integer	Select probability distribution of switch. 0 (zero) means Gaussian and 1 means uniform distribution.
IMAX	Integer	If 1, printout of extrema to LUNIT6 for every energization.
IDICE	Integer	Use of standard random generator. A value of 0 (zero) implies computer-dependent random generator and a value of 1 standard random generator.
KSTOUT	Integer	Extra printed (LUNIT6) output for each energization.
NSEED	Integer	Repeatable MonteCarlo simulations.
HighResolution	On/**Off**	Usage of $Vintage 1 (if possible).
SortByCard	**On**/Off	Data file written with BRANCH cards first, followed by SWITCH cards and the SOURCE cards.
SortByGroup	On/**Off**	The group number given to each object determines the sequence of cards. The lowest group number comes first.
SortByXpos	On/**Off**	The leftmost object is written first.
AutoPath	On/**Off**	Library files are supposed to be located in the USP folder and have the extension. When this option is enabled the $Prefix, $Suffix option is written to the ATP-file.
AutoError	On/**Off**	Auto-detect LIS-file error messages.

AutoErrorCode	=0	Default errors to detect. Binary format: 1-2-4-8-16. 31 means, that all trigger string is active (1+2+4+8+16=31).

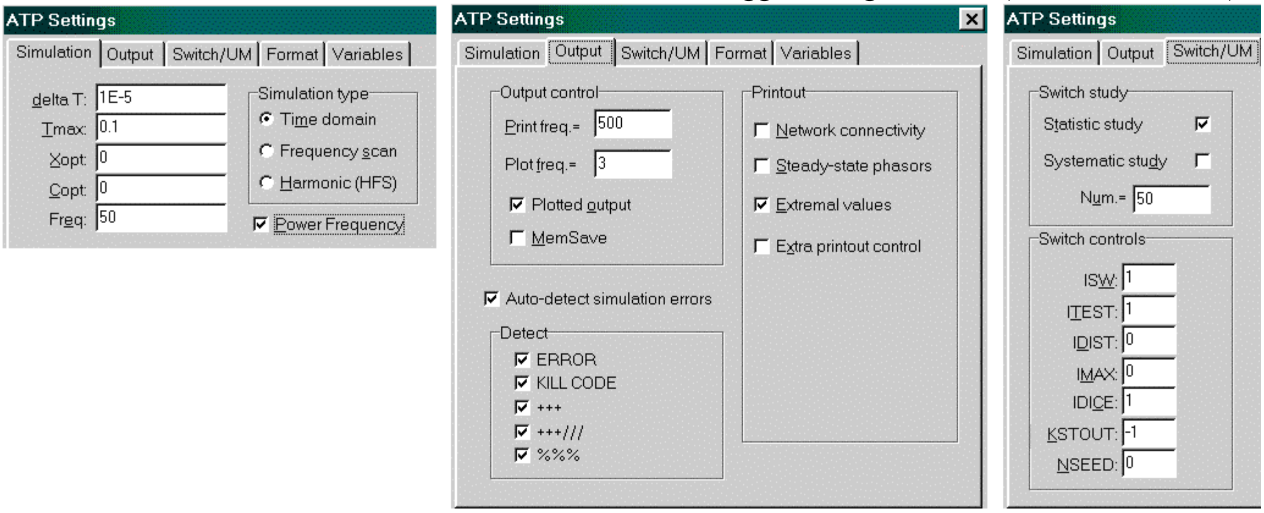


Fig. 7.5 - Setting ATP specific variables.

This section also includes a section for *Commands* (max 10) found under the *ATP* | *Edit Commands*. Each command takes three lines. The first one specifies the command name that appears in the menu, the second is the program/file to launch and the third is a code for what kind of file to send as parameter.

Parameter	Type	Description
BatchJobx.name	Text	Name of the user specified commands. Number *x* specifies the location of the commands in the ATP menu.
BatchJobx.filename	Text	Name of the batch (or .EXE) file executed by ATPDraw when the command is selected in the ATP menu.
BatchJobx.parameter	Integer	Specifies which file is sent as parameter, before executing the batch job: =0: No file name is sent, =1: File dialog opens =2: ATP-file name is sent, =3: PL4-file name is sent.

Default settings for the *Pocket Calculator* features of ATP are also specified in the [ATP settings] section:

Parameter	Type	Description
Var.Simulations	=1	Number of Pocket Calculator simulations.
Var.Count	=0	Number of $Parameter variables.

7.1.6 [1024x768]

This controls the default window position on the screen (Left, Top, Width, Height).

7.1.7 [Reload]

Contains the 5 last loaded project files. These files appear under the *File* | *Reload* menu.

7.1.8 [Objects]

This controls the visibility status of the Toolbar and the Status bar. Default setting makes them visible at program startup, which can be modified in the *View* menu.

7.2 PFC simulations in ATPDraw

The *Verify* feature of ATPDraw enables the user to compare the line/cable model with an exact PI-equivalent as a function of frequency, or verify the power frequency benchmark data for zero/positive short circuit impedances, reactive open circuit line charging, and mutual zero sequence coupling. The *Verify* module supports the POWER FREQUENCY CALCULATION (PFC) of zero and positive short circuit impedances and open circuit reactive line charging, along with mutual zero sequence impedance for multi circuit lines.

The supporting programs LINE CONSTANTS and CABLE CONSTANTS calculate the series impedance and the shunt admittance from geometrical data and material properties. These electrical parameters are part of the printout file (`.lis`). The power frequency calculations give in principle the short circuit impedances and the open circuit reactive power. The line/cable may be a single circuit component with an arbitrary number of phases or a multi-circuit component where all circuits normally are three-phase. The following parameters are calculated for a single circuit in a line/cable with *n* conductors:

a) *Short circuit impedances*

All terminals at one end of the line/cable are connected to ground. A positive sequence symmetrical voltage is applied to the terminals at the other end and the positive sequence impedance is calculated: $Z_+ = E_+ / I_+$

The voltage applied to the terminal *i* is:

$E_i = E_+ \cdot \exp(-j \cdot 2\pi \cdot (i-1)/n)$, where *n* is the number of phases in the circuit.

The positive sequence current is obtained from the terminal currents by the formula:

$$I_+ = \frac{1}{n} \cdot \left[I_1 + I_2 \cdot \exp(j2\pi/n) + \cdots + I_i \cdot \exp(j2\pi(i-1)/n) + \cdots + I_n \cdot \exp(-j2\pi/n) \right]$$

The zero sequence impedance is calculated in a similar way:

$$Z_0 = E_0 / I_0$$

The voltage E_0 here is applied to all terminals and I_0 is the average current supplied by the source.

b) *Open-circuit reactive power*

All terminals at one end of the component are open (except the conductors which are specified to be grounded). A positive sequence symmetrical voltage is applied to the terminals at the other end and the positive sequence current component is calculated by the same formula as for the positive sequence impedance. The positive sequence open-circuit reactive power is then calculated by the formula:

$Q_+ = \mathrm{Im}\left(n \cdot E_+ \cdot I_+^*\right)$, where E_+ is the line to line voltage.

Using the voltage between two adjacent phases for an *n*-phase circuit gives $E_+ = V / \left[2 \cdot \sin(\pi/n)\right]$. The calculation I_+ is based on an ATP calculation with E_+ =1.0. Using this value for I_+ implies that

$$Q_+ = \frac{-V^2 \cdot n}{4 \cdot \sin^2(\pi/n)} \mathrm{Im}(I_+)$$

ATP also automatically calculates the reactive power supplied by the source (Q_1..Q_n). The open-circuit reactive power can thus also be calculated by taking the average of these quantities for all phases and multiply by a factor 2 (since a peak value 1.0 is used in the calculation and the line-to-line voltage is specified as rms):

$$Q_{+} = \frac{-V^2 \cdot 2}{n}(Q_1 + Q_2 + \cdots + Q_n)$$

The zero sequence open-circuit reactive power is calculated as well. The same voltage is then applied to all terminals at one end of the line. The zero sequence current is the average value of the current injected into the terminals. This current I_0 is calculated by ATP with $E_0 = 1.0$. Using this value for I_0 implies that

$$Q_0 = \frac{-V^2 \cdot n}{4 \cdot \sin^2(\pi / n)} \mathrm{Im}(I_0)$$

In this case ATP automatically calculates the reactive power Q , injected into the circuit from the source. Similarly to the positive sequence values, the zero sequence open-circuit reactive power is also equal to

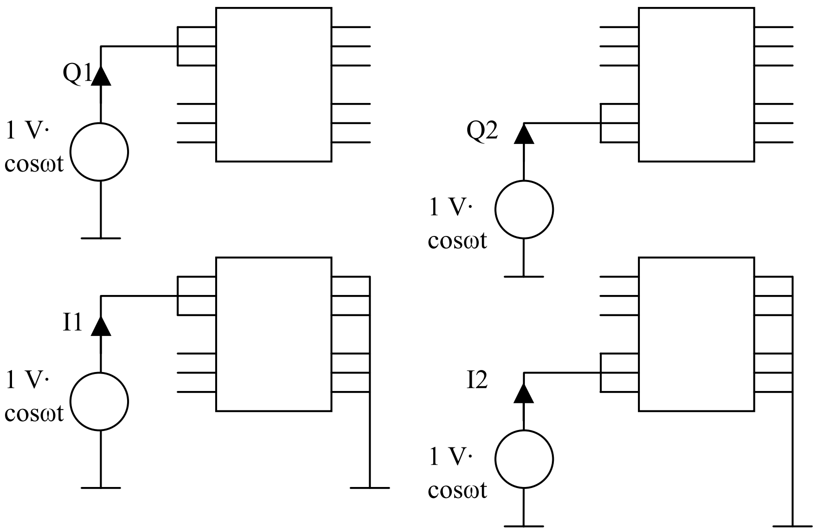
$$Q_0 = \frac{-V^2 \cdot 2}{n}(Q)$$

For a line/cable with several circuits, each circuit is tested separately. For short-circuit calculation the other circuit(s) is/are is also grounded at one end, while for open-circuit calculations all terminals are open. The mutual coupling between the circuits is calculated as well and called *zero sequence transfer impedance*. This is done by connecting all phases of each individual circuit to a common node. A current $3 \cdot I_0$ is then applied to one of these common nodes circuit and the voltage on the other node is measured. All terminals at the other end of the component is grounded. The procedure is repeated for all circuits except the last one. Below is listed the `xVerifyF.dat` file for a 6-phase line.

```
BEGIN NEW DATA CASE
1.667E-9    -1.0
       1       1               1
$PREFIX, D:\ATPDraw3\lcc\
$INCLUDE, LCC_6.lib, INZO1_, INZO1_, INZO1_, INZO1D, INZO1E, INZO1F $$
  , OUTO1A, OUTO1B, OUTO1C, OUTO1D, OUTO1E, OUTO1F
$INCLUDE, LCC_6.lib, INZO2A, INZO2B, INZO2C, INZO2_, INZO2_, INZO2_ $$
  , OUTO2A, OUTO2B, OUTO2C, OUTO2D, OUTO2E, OUTO2F
$INCLUDE, LCC_6.lib, INZS1_, INZS1_, INZS1_, INZS1D, INZS1E, INZS1F $$
  , ######, ######, ######, ######, ######, ######
$INCLUDE, LCC_6.lib, INZS2A, INZS2B, INZS2C, INZS2_, INZS2_, INZS2_ $$
  , ######, ######, ######, ######, ######, ######
$INCLUDE, LCC_6.lib, INPO1A, INPO1B, INPO1C, INPO1D, INPO1E, INPO1F $$
  , OUPO1A, OUPO1B, OUPO1C, OUPO1D, OUPO1E, OUPO1F
$INCLUDE, LCC_6.lib, INPO2A, INPO2B, INPO2C, INPO2D, INPO2E, INPO2F $$
  , OUPO2A, OUPO2B, OUPO2C, OUPO2D, OUPO2E, OUPO2F
$INCLUDE, LCC_6.lib, INPS1A, INPS1B, INPS1C, INPS1D, INPS1E, INPS1F $$
  , ######, ######, ######, ######, ######, ######
$INCLUDE, LCC_6.lib, INPS2A, INPS2B, INPS2C, INPS2D, INPS2E, INPS2F $$
  , ######, ######, ######, ######, ######, ######
$INCLUDE, LCC_6.lib, INMS11, INMS11, INMS11, INMS12, INMS12, INMS12 $$
  , ######, ######, ######, ######, ######, ######
BLANK BRANCH
BLANK SWITCH
14INZO1_+1      1.0       50.       0.0                           -1.0
14INZO2_+1      1.0       50.       0.0                           -1.0
14INPO1A+1      1.0       50.       0.0                           -1.0
14INPO1B+1      1.0       50.     -120.                           -1.0
14INPO1C+1      1.0       50.     -240.                           -1.0
14INPO2D+1      1.0       50.       0.0                           -1.0
14INPO2E+1      1.0       50.     -120.                           -1.0
14INPO2F+1      1.0       50.     -240.                           -1.0
14INZS1_+1      1.0       50.       0.0                           -1.0
14INZS2_+1      1.0       50.       0.0                           -1.0
14INPS1A+1      1.0       50.       0.0                           -1.0
14INPS1B+1      1.0       50.     -120.                           -1.0
14INPS1C+1      1.0       50.     -240.                           -1.0
14INPS2D+1      1.0       50.       0.0                           -1.0
```

```
14INPS2E+1        1.0       50.     -120.                              -1.0
14INPS2F+1        1.0       50.     -240.                              -1.0
14INMS11-1         3.       50.        0.0                              -1.0
BLANK SOURCE
  INMS12
BLANK OUTPUT
BLANK CARD PLOT
BEGIN NEW DATA CASE
BLANK
```

The `xVerifyF.dat` file describes the following 9 cases:



Cir. 1:

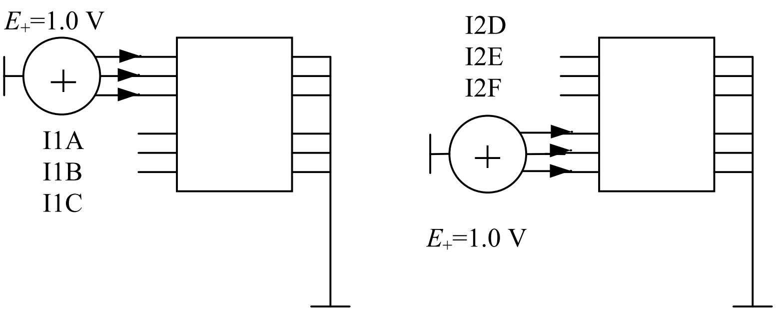
$$Q0 = \frac{2V^2}{-3}\text{Q1}$$

Cir. 2:

$$Q0 = \frac{2V^2}{-3}\text{Q2}$$

Cir. 1: $Z0 = \frac{1}{3 \cdot \text{I1}}$

Cir. 2: $Z0 = \frac{1}{3 \cdot \text{I2}}$

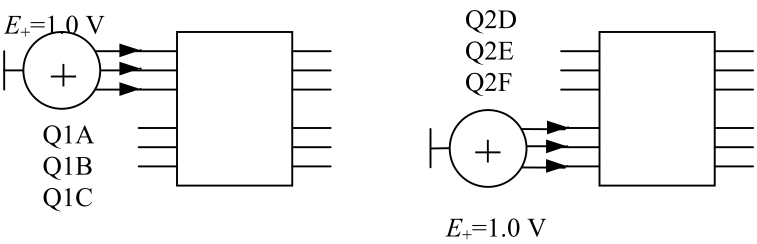


Cir. 1:

$$Z+ = \frac{1.0 \cdot 3}{\text{I1A} + \text{I1B} \cdot e^{+j120} + \text{I1C} \cdot e^{-j120}}$$

Cir. 2:

$$Z+ = \frac{1.0 \cdot 3}{I\text{1D} + I\text{1E} \cdot e^{+j120} + I\text{1F} \cdot e^{-j120}}$$

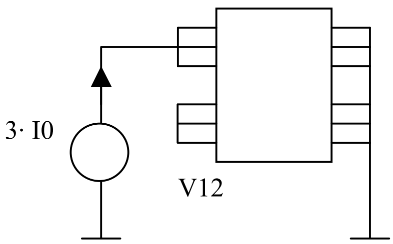


Cir. 1:

$$Q+ = \frac{2V^2}{-3}(\text{Q1A} + \text{Q1B} + \text{Q1C})$$

Cir. 2:

$$Q+ = \frac{2V^2}{-3}(\text{Q2D} + \text{Q2E} + \text{Q2F})$$



Cir. 1- Cir. 2:

$$Z00 = \text{V12/I0}$$

Zero sequence short circuit impedance: (real and imaginary part). $Z_0 = R_0 + jX_0$.

Each phase of a circuit is connected to a 1 V amplitude voltage source with zero phase angle. The other end of the line is grounded. Z_0 is calculated as the inverse of the injected current divided by the number of phases in the circuit. All phase conductors of other phases are open.

Positive sequence short circuit impedance: (real and imaginary part). $Z_+ = R_+ + jX_+$.

The phases of a circuit are connected to a 1 V amplitude voltage source with phase angle *-360*(i-1)/n* where *i* is the phase number (1,2,3..) and *n* is the number of phases of the tested circuit. The other end of the line is grounded. Z_+ is calculated as the inverse of the positive sequence current. All phase conductors of other phases are open.

Zero sequence line charging: Q_0

Each phase of a circuit is connected to a 1 V amplitude voltage source with zero phase angle. The other end of the line is open. Q_0 is the injected reactive power multiplied by the square of the user specified base voltage (multiplied with 2/n). All phase conductors of other phases are open.

Positive sequence line charging: Q_+

The phases of a circuit are connected to a 1 V amplitude voltage source with phase angle *-360*(i-1)/n* where *i* is the phase number and *n* is the number of phases of the tested circuit. The other end of the line is open. Q_+ is calculated as the average injected reactive power multiplied by the square of the user specified base voltage (multiplied with *2/n*). All phase conductors of other phases are open.

Mutual zero sequence impedance: (real and imaginary part). $Z_{00} = R_{00} + jX_{00}$.

Each phase of the i^{th} circuit is connected to a 1 A amplitude current source with zero phase angle. The receiving end of the circuits *i* and *j* is grounded. The j^{th} circuit is short-circuited and open in the sending end. Z_{00} is calculated as the voltage at the sending end of the j^{th} circuit. The process is repeated for all circuits. All phase conductors of phases not belonging to the i^{th} and j^{th} circuit are open.

7.3 References

[1] *ATPDRAW version 3*, User Manual, TR A4389, EFI, Norway, 1996

[2] Ned Mohan, *Computer Exercises for Power Electronic Education*, 1990, Department of Electrical Engineering, University of Minnesota.

[3] *ATP-EMTP Rule Book*, Canadian-American EMTP Users Group, 1997

[4] Lauren Dube, *MODELS in ATP*, Language manual, February 1996

[5] H.W. Dommel, *Electromagnetic Transients Program. Reference Manual (EMTP Theory Book*), Bonneville Power Administration, Portland, 1986.

[6] L. Prikler, Main Characteristics of Plotting Programs for ATP, EEUG News Vol. 6, No. 3-4, August-November 2000, pp. 28-33

7.4 Index