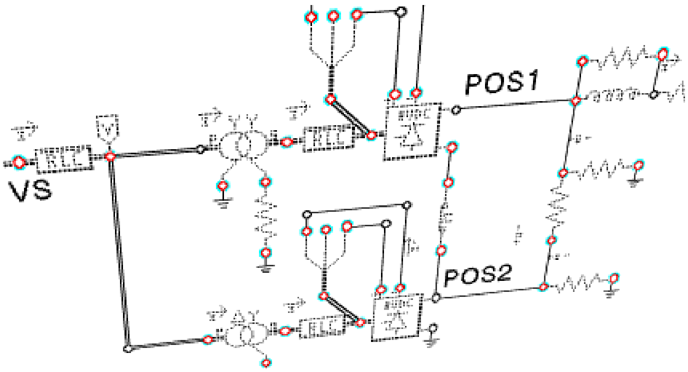
ATPDRAW

version 5.6
for Windows 9x/NT/2000/XP/Vista

Users' Manual



László Prikler,
Hans Kristian Høidalen

ATPDraw™
for Windows
5.6

Preliminary Release No. 1.0
November 2009

PREFACE

This Users' Manual documents all main features of ATPDraw version 5.6. The manual is an extensive update of the previous User Manual prepared by László Prikler at SYSTRAN Engineering Services Ltd. in Budapest for version 3.5 (SINTEF TR F5680) dated 2002. Version 5.6 is substantially updated compared to version 3.5; New design, new and extended components, new handling of Models, Hybrid Transformer, multi-phase nodes, vector graphics, Output Manager, Line Check, Circuit Texts, Optimization etc.. The Reference Manual gives a summary of menu items and menu options. The Advanced Manual covers the features Grouping, Models, electrical machine, line/cable-, and transformer modeling, and optimization. Finally the Application Manual is extended with several examples. New ATPDraw users are advised to start with the Installation and Introductory manuals.

ATPDraw is developed by NTNU and SINTEF Energy Research. Program and development have been financed by Bonneville Power Administration, USA, version 5 in co-operation with EEUG and Schneider Electric, France.

For Norwegian University of Technology Trondheim, Norway, November 26th 2009.

Hans Kr. Høidalen
Professor NTNU-Norway

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 CodeGear Delphi 2007 and runs under Windows 9x/NT/2000/XP/Vista. 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 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. Special components support the user in machine and transformer modeling based on the powerful Universal Machine and BCTRAN components in ATP-EMTP. In addition the advanced Hybrid Transformer model XFMR and Windsyn support is included.

ATPDraw supports hierarchical modeling by replacing selected group of objects with a single icon in an almost unlimited numbers of layers. Components have an individual icon in either bitmap or vector graphic style and an optional graphic background. ATPDraw supports up to 10.000 components each with maximum 64 data and 32 nodes.

TABLE OF CONTENTS

1. Introduction . . .

. . . .

ATPDraw™
for Windows
5.6

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/flip, 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. An advanced Hybrid Transformer model based on Test Report, Design or Typical values with topologically correct core is also supported. ATPDraw also integrated with Windsyn for Universal Machine modeling based on manufacturers data.

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 3.6 and above of ATPDraw for 9x/NTx/2000/XP Windows platforms are written in Borland Delphi 6.0. From version 5.3 CodeGear Delphi 2007 is used. This version uses the html help file system supported in Windows VISTA.

ATPDraw™ is a trademark and copyrighted by © 2005-2009 Norwegian University of Science and Technology, Norway. Program developer is Dr. Hans Kristian Høidalen 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 mainly been financed by Bonneville Power Administration in Portland, Oregon, USA, with Pacific Engineering Corporation as project coordinator. Development in version 5 has in addition been co-funded by the European EMTP User's Group and Schneider Electric.

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.

[1] Source: WWW.EMTP.ORG

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) 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).

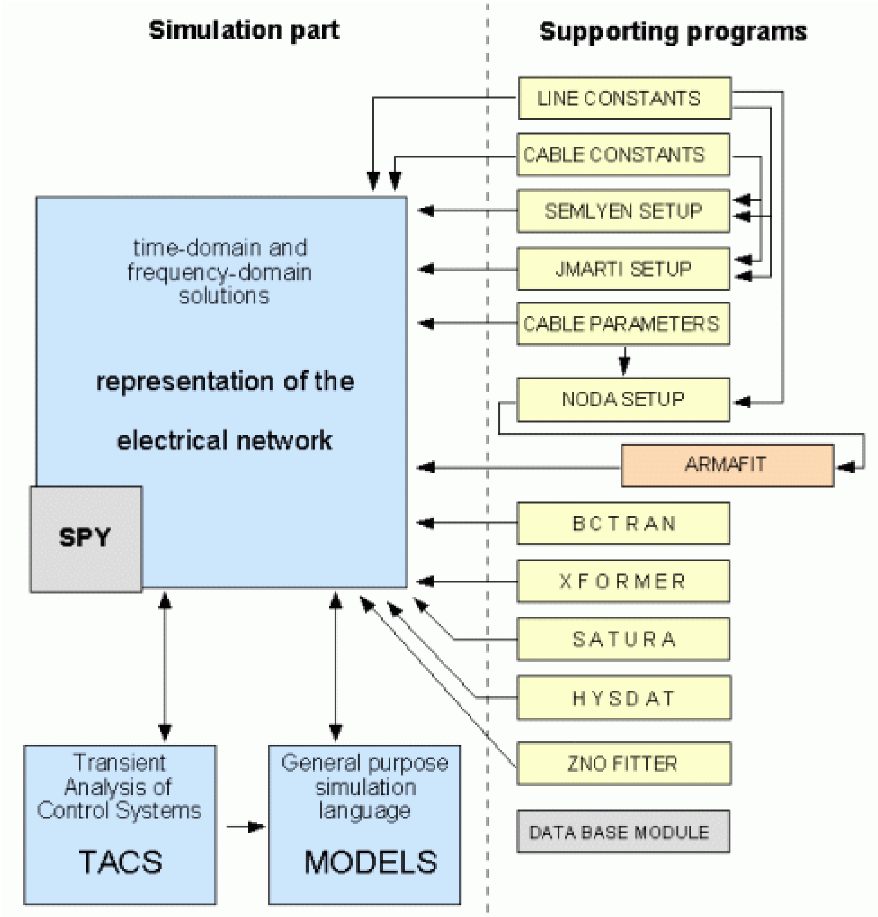


Fig. 1.1 - Supporting routines in ATP.

Source: www.emtp.org

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.

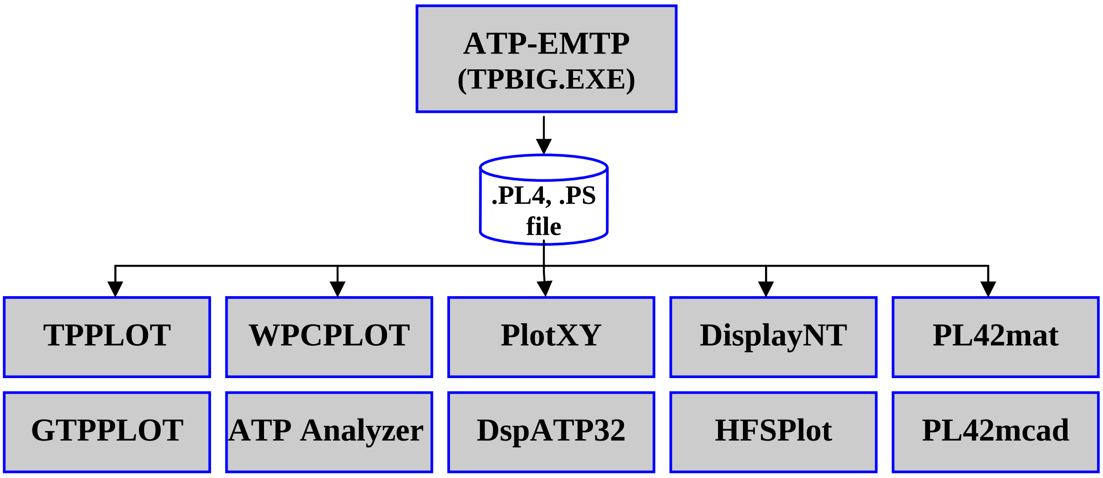


Fig. 1.2 – Plotting programs for ATP.

ATP Analyzer is a Windows based program intended for observing and analyzing analog signals and discrete channel data associated power generation, transmission and distribution systems. The programis capable of reading and displaying analog signals produced by ATP as type PL4 output file data, industry standard COMTRADE file and analog and digital data produced from protective relays and fault recorder equipment, analog signals from table ASCII text data, and audio wave files. A total of 254 signals can be managed.

Signals can be displayed in time domain in multiple overlay charts. One or more signals can be displayed as a function of another on an X versus Y chart. Up to three signals can be displayed simultaneously in the frequency domain as harmonics or as a broad frequency spectrum. Charts may be printed and copied to the Windows clipboard.

The program can process the data for harmonic content and store processed data in a Windows Access Data base.

Developer: Bonneville Power Administration, 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. The program also comes in a multi-window edition **PlotXwin.exe**.

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/Vista™: 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 5.5 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 and connections

REFERENCE MANUAL

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

ADVANCED MANUAL

How to create new circuit objects in ATPDraw
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
How to use the Hybrid Transformer component
Referencing four non-standard Component dialog boxes:
- Saturable 3-phase transformer
- Univeral Machines
- Statistical switches
- Harmonic source
- Windsyn

Vector graphics and picture background
Optimization module

APPLICATION MANUAL

Line/cable constant application examples
Single-phase to ground fault and fault tripping transients
Shunt capacitor bank switching
Lightning studies, arrester modeling
Pulse width modulated induction machine
Synchronous machine controls
HVDC station, rectifier/converter 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 L* : Select *Rotate L* 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>`.

2. Installation Manual . . .

. . . .

ATPDraw™
for Windows
5.6

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:

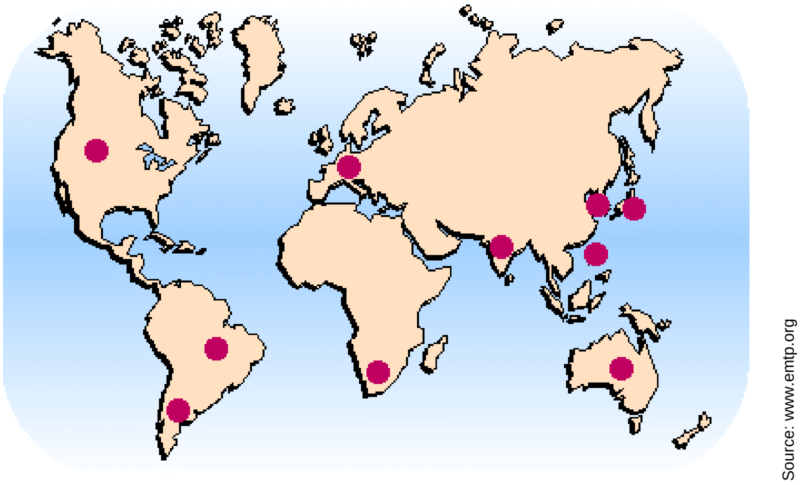


Fig. 2.1 - Location of ATP-EMTP user groups.

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 `atpdraw5x.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 `atpdraw5x.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:\ATP\ATPDraw5`. 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 ATPDraw5.exe first time it will create the necessary system sub-folders /`ATP`, `/BCT`, `/HLP`, `/LCC`, `/MOD`, `/Project` under the main program folder.
5) Download the latest patch file called `patchxv5.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
MOD            <DIR>        10-22-01  9:58p Mod
BCT            <DIR>        03-22-02 12:42p Bct
```

```
ATPDRAW   EXE     2,465,792  04-12-08 10:58a Atpdraw.exe
ATPDRAW   CHM     1,069,056  04-12-08  3:24p ATPDraw.chm
ATPDRAW   SCL       184,320  04-10-08 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 three files are required: `ATPDraw.exe`, `ATPDraw.scl` (standard component library), and `ATPDraw.chm` (help file). Besides, the user can create his own library components (user specified or models) and include files. ATPDraw does not rely on other specific disk files.

Project file: When the user saves a circuit the work is stored in the project file (*.acp = atpdraw circuit project). This file contains the circuit with all data and graphical representation. The project file is compressed by a public domain Pkzip 2.0 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 inherit their properties from 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 (bitmap or vector) and a help text. 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 *Library* menu. The default icon can also be modified by using the built in icon editors. New user specified objects are created by specifying new support files.

ATP file: This file is produced by ATPDraw and used as input to ATP simulation. The `.atp` files with all $Include files are written to the Result Directory with default location is specified as the `\ATP` sub-directory under *Tools|Options/Files&Folders*. The Result Directory can be changed via *ATP|Sup-process|Make ATP file*. The ATP can be edited with any text-processors, including ATPDraw's own *Text Editor* (*Atp|Edit ATP file (F4)*). It is advised, however only for experts to modify this file manually.

Include files: User Specified Objects, Line&Cables, and Windsyn 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 include files are stored in memory and written to the Result Directory (same as ATP file) each time the ATP file is created. Some nonlinear components or saturable transformers might also have an include file for the nonlinear characteristic.

Data files: The user can export data for special components to a libray for later use. A data file is introduced because the involved components have too many data to fit in to the standard component library data structure. The data for a component in the circuit is stored internally in memory. The following file types are used:

- 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. It should preferably be stored in the `\LCC` directory.
- 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. It should preferably be stored in the `\BCT` directory.
- A Hybrid Transformer model is described by a .xfm file. This file contains the winding resistance, leakage inductance, capacitance, and core data. It should preferably be stored in the `\BCT` directory.
- 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.

2.5.1 Organizing the files

When ATPDraw opens a project no file is written to disk. All data are stored in memory. When the project is stored the disk files are not deleted. 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 export/modify yourself outside a project need to be kept. Two house-keeping options are available under *Tools|Options/View/ATP*:

- Delete temp-files after simulation: Deletes all temporary BCTRAN/LCC files (.dat, .lis. pch) and all temporary ATP files *.bin when the simulation is finished. The files required to run ATP outside of ATPDraw (atp- and lib- files) are left on disk. In case of protected elements the lib-files are immediately deleted and the atp-file is modified. During debugging a LCC or BCTRAN model this button should be left unchecked.
- Delete result files on exit: Deletes the all temporary and result files (.atp, .lib, .lis, .pl4, .dat, .pch, .bin, .gnu) from ResultDir (the ATP folder as default) when the circuit is closed. All data is stored in the project files of ATPDraw anyway.

2.5.2 Configuring ATPDraw

The `ATPDraw.ini` file contains customizable program options. One such file for each user of the computer is stored in `%APPDATA%\atpdraw\`. The environmental variable `APPDATA` is system dependent but typical equal to '`c:\Documents and Settings\user\Application Data`'. 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.

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 *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.

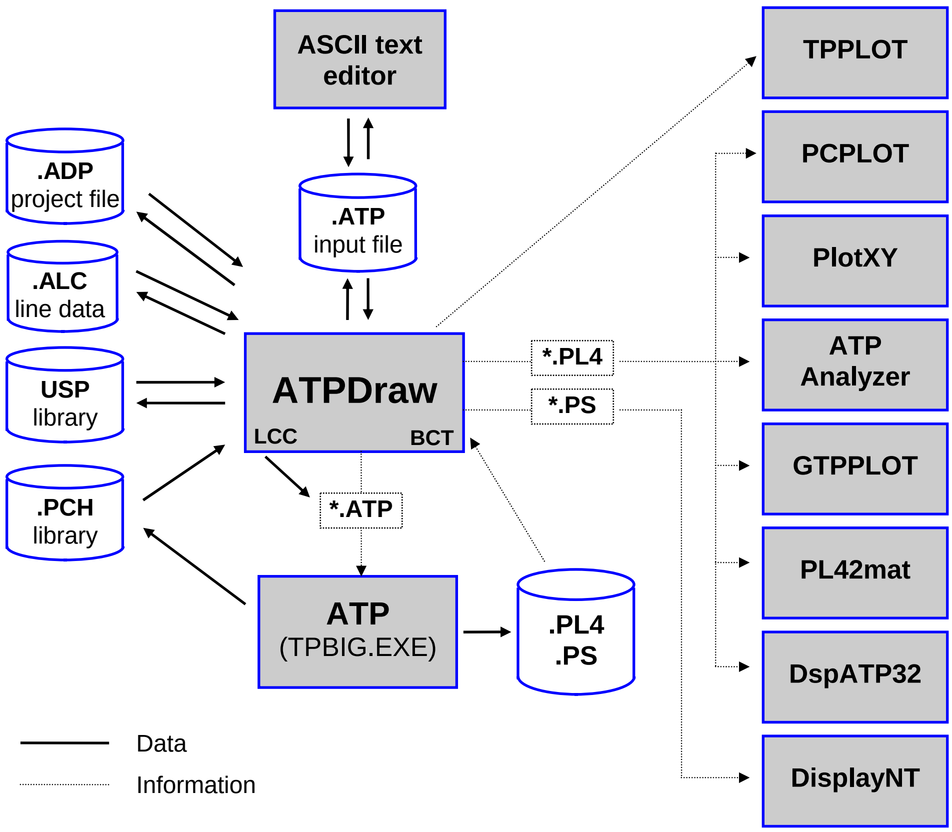
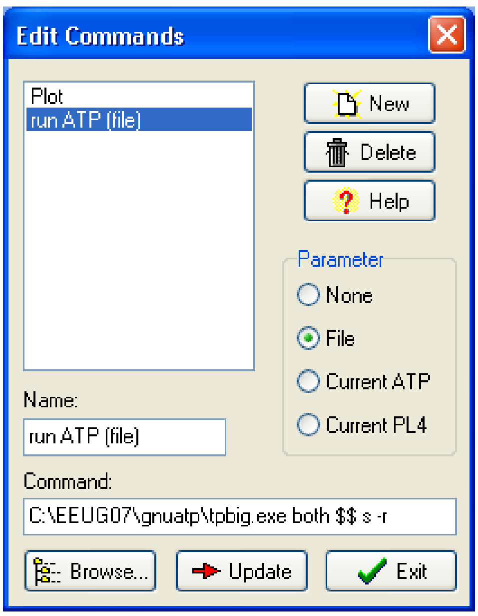


Fig. 2.2 - Interaction between ATPDraw and the other ATP programs.



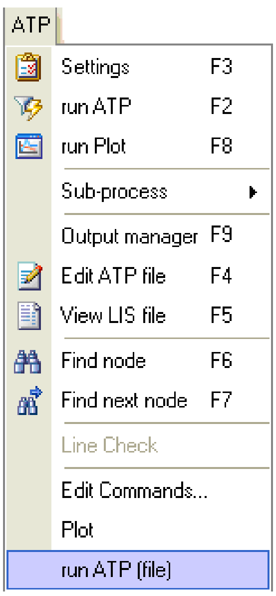


Fig. 2.3 - The Edit Commands dialog box.

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.

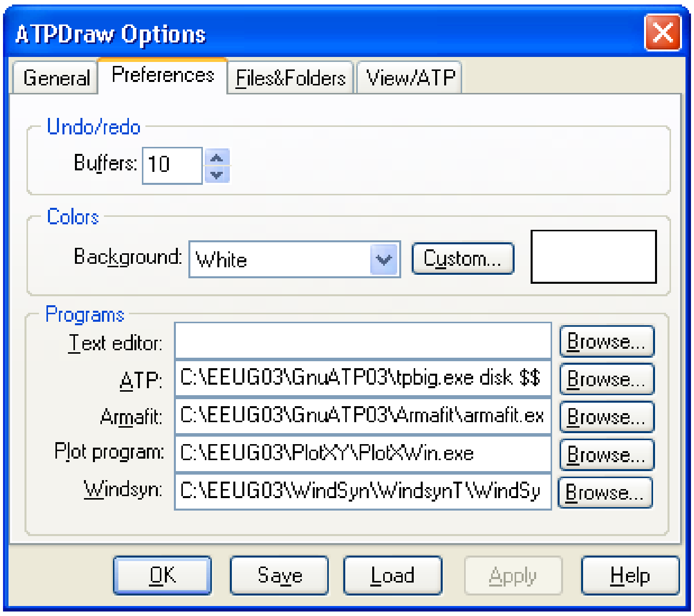


Fig. 2.5 - Default settings to run ATP and Armafit.

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
```

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

```
%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
```

As illustrated in Fig. 2.3 it is in some cases possible to run the **tpbix.exe** program directly from ATPDraw. The batch file flag %1 must then be replace by $$ in the ATP-command.

2.6.2 Calling PlotXY , PCPlot or ATP_Analyzer

A main plotting command can be set as shown in Fig. 2.5. When selecting this command (short cut F8) the plotting program starts with the current ATP-file (with extension .PL4) as parameter. The user can in addition create the *Run ATP Analyzer* and/or *Run WPCPLOT* 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.

2.6.3 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.acp and my2nd.acp will be loaded automatically and displayed in separate circuit windows.

```
C:\ATPDRAW>atpdraw c:\atpdraw\cir\my1st.acp c:\cir\my2nd.acp
```

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**).

2.6.4 Drag and drop project files

ATPDraw accepts project files dragged from the Windows File Manager (from v. 5.6). Dropping the project file (.acp) on the header, main menu of background causes the file to be opened in a new circuit window. Dropping the file in an existing circuit window causes the file to be imported into that circuit.

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
Norwegian University of Science and Technology
Dept. Electric Power Engineering
7491 Trondheim - NORWAY
http://www.ntnu.no
E-mail: Hans.Hoidalen@elkraft.ntnu.no
Phone: + 47 73594225
Fax: + 47 73594279

The ATPDraw Web page is maintained at address:

`http://www.elkraft.ntnu.no/atpdraw`

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 262 circuit object support files. These 262 files support more than 170 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)
Windsyn (separate program, manufacturer data)

Transformers:

Single-phase and 3-phase ideal transformer. Type 18 source
Single-phase saturable transformer
3-phase, 2- or 3 winding saturable transformer (Auto, Delta, Wye, and ZigZag)
BCTRAN. 1-3 phases, 2-3 windings. Auto-transformers, Y-, and D- connections
Hybrid Transformer (XFMR) with topological core; triplex, 3 or 5-legged, shell form. 3-phases. 2-3 windings. Auto, Y- and D-coupled windings.

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
WriteMaxMin cost function

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 (50-66).
Initial condition for TACS objects (type-77)

User specified objects

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

Steady-state 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
Load flow components

3. Introductory Manual . . .

. . . .

ATPDraw™
for Windows
5.6

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*.

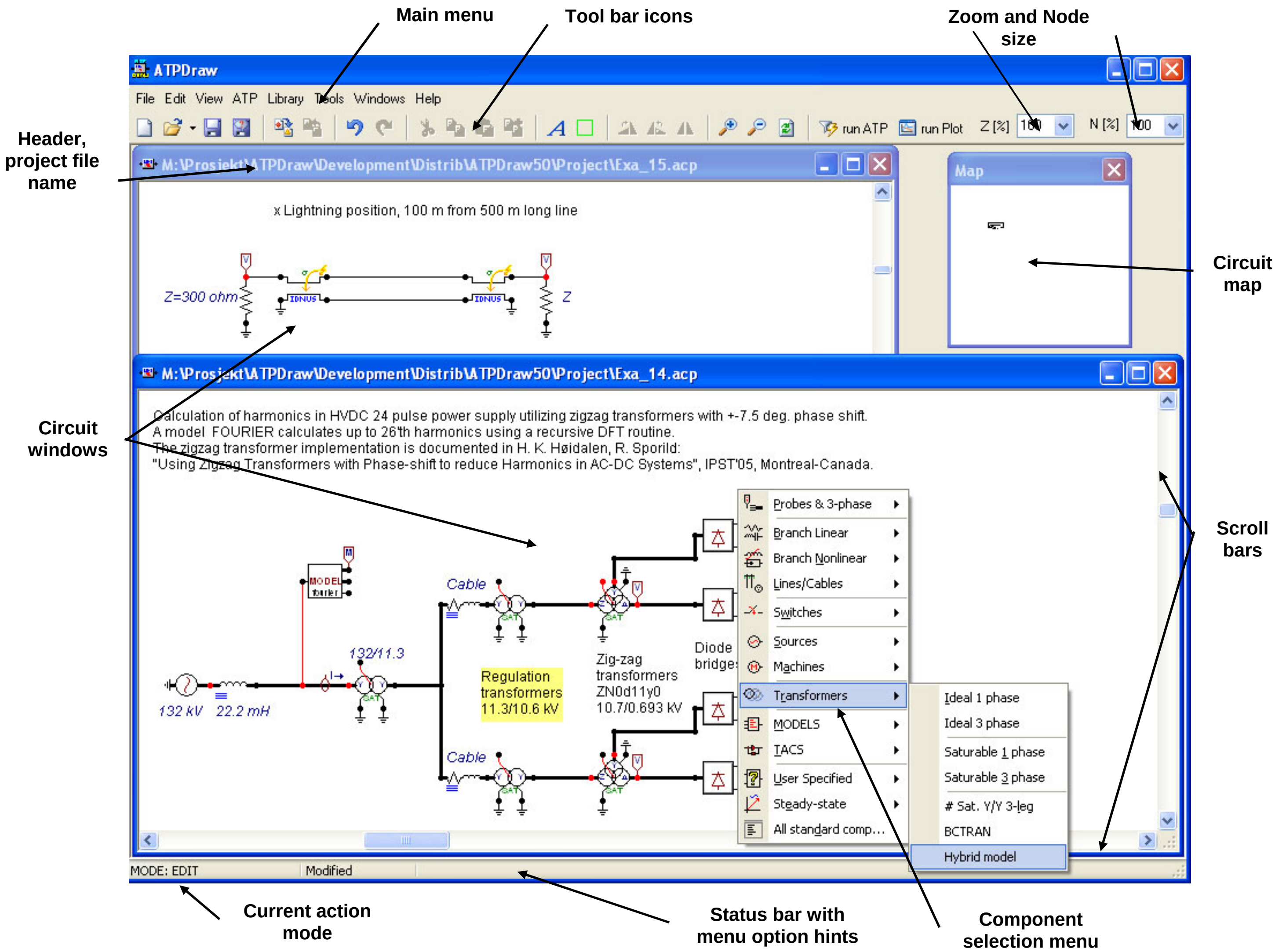


Fig. 3.1 - The Main window and the floating Component selection menu.

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. Find Node and Line Check. Output Manager lists all output requests.
Library:	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.

Zoom and node size:
In these menus you can type in zoom and node size in [%] or select predefined values in the popup box.

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 10000x10000 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 Ctrl+*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*.
EDIT TEXT	Indicates that text editing is preferred. Hold down the Alt key to enter this mode of operation or select *Edit Text* from the *Edit* menu. Click left in an empty space to add a new text. Click the left mouse button on an existing text (circuit text, label, node name) to edit it direcly on screen. Click left, hold down and drag to move it to a new position. If the text is overlapped by a component icon, this mode of operation is required to access the text.
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.

Left simple click:

On object: Selects object or connection.
If the *Shift* key is pressed, the object is added to the current selection group.

On connection: Draw a new connection with the same properties.

On object node: Begins to draw a connection.
Move the mouse to the end node, left click to place, right to cancel.

On text, labels and node names: Edit the text directly on screen. Press *Alt* to favour the text selection compared to components and connections.

In open area of the circuit window: Unselects object.

Right simple click:

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/Connection or Text* 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*.

Left click and hold:

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

On connection: Select connection.

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.

On text, labels and node names: Move the text. Press *Alt* to favour the text selection compared to components and connections.

Left double click:

On object node:
Performs the *Node data* window.

On selected or unselected single object:
Performs the *Component/Connection or Text* 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	Ctrl+Z	*Edit | Undo*
REDO	Ctrl+Y	*Edit | Redo*
Cut/Copy	Ctrl+X/Ctrl+C	*Edit | Cut/Copy*
Delete	DEL	*Edit | Delete*
Paste	Ctrl+V	*Edit | Paste*
Duplicate	Ctrl+D	*Edit | Duplicate*
Select/All	Ctrl+A	*Edit | Select All*
Select/Inside	Ctrl+I	*Edit | Select| Inside* (or left double click in open space)
Select/Properties	Ctrl+P	*Edit | Select| by Properties*
New/Select text	Ctrl+T	*Edit | Edit text*
Rotate clockwise	Ctrl+R	*Edit | Rotate R* (or right click)
Rotate left	Ctrl+L	*Edit | Rotate L*
Flip	Ctrl+F	*Edit | Flip*
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	NUM + / -	*View | Zoom In / Out*
Refresh	Ctrl+Q	*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 lime-colored 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. Components with centre point within the indicated region or rectangle are added to the selected objects group. Connections require both end points within the region to be selected. Select *Edit|Rubber Bands* to stretch connections with one end inside and one end outside. 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 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 shown in Fig. 3.2 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. Some of the advanced components like LCC, BCTRAN, Hybrid Transformer have special dialog boxes for input.

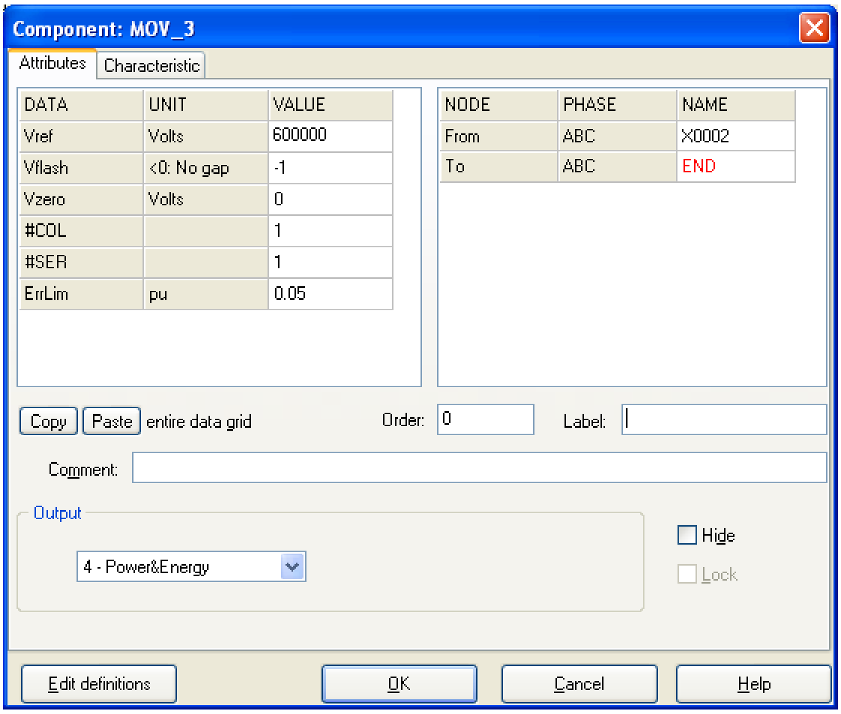


Fig. 3.2 – Component dialog box, attributes page.

The Component dialog box shown in Fig. 3.2 consists of a Data part and a Node part. In the Data part the user can specify values using '.' as the decimal symbol and use 'e' as exponent. A variable name (6-char text string) can also be specified and given a global value under *ATP|Settings/Variables*. Specifying a variable is only possible if the *Param* property in the definitions is set to unity (and the data is not used in internal calculations; RLC-lines with lengths<>1, phase angle of 3-phase AC source etc.). The *Copy/Paste* buttons allows copying the entire data set via the Windows clipboard. Node names (6 or 5 characters) can be specified in the right grid. Node names drawn in a red color are already given a name by the user while black names are inherited. If the user wants to change a node name the red names/nodes should be preferred, otherwise name conflict warnings will appear. Node data are also given in the Node dialog box by clicking on the nodes. Multi-phase nodes can only take a 5 character name, and the phase sequence extension A..Z is added automatically.

Order is optionally used for sorting (*ATP|Settings/Format*; sorting by order (low-high)), *Label* is a 12 character text string on screen, and *Comment* is a line of text written to the ATP file in front of the component's cards. The *Output* panel varies somewhat between components, but is usually used for branch output requests. In the lower left corner there is *Edit definitions* button. This gives access to all the local properties inherited from the support file, including the icon, local help, names of nodes and data, node positions, default values, param flags, limits, and units.

Clicking on *Help* will display the helt text for the component; first comes the global help obtained from the support files (ATPDraw.scl for standard components), second comes local help specific to this component, and finally comes global help from the /HLP directory.

Default component attributes are stored in support files. Access to create and customize support files is provided by the *Library* menu.

Components are connected if their nodes overlap or attached to the same connection. To draw a connection, 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 *run ATP* option 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.

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. 3.3) 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?

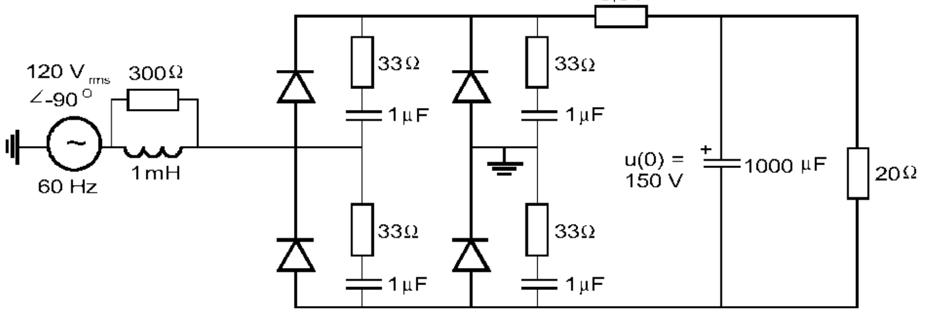


Fig. 3.3 – Single-phase rectifier bridge.

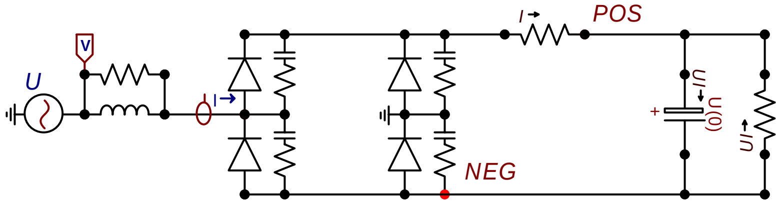


Fig. 3.4 – Your first circuit (`Exa_1.adp`).

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. 3.3 are based on settings of Xopt and Copt equal to zero as will be explained later. The circuit in Fig. 3.4 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 *Component selection menu*, which appears with a right mouse click on open area of the circuit window. Fig. 3.5 shows how to select a general AC (type 14) source under *Sources | AC source (1&3)*.

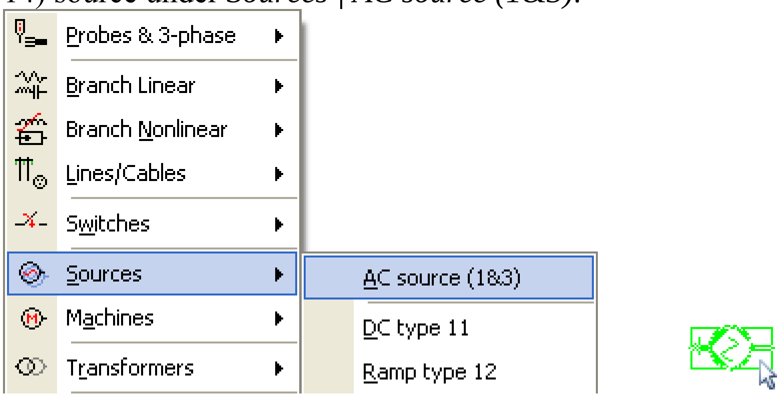


Fig. 3.5 - Selecting an AC source.

After you have clicked in the *AC source (1&3)* field, the selected source appears in the circuit window in lime color, 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 in 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.

To give data to the AC source component, click on with the **right mouse button** (or left double click). You can give data to objects at any time during the building process. If you right click on the AC source icon, a window as shown in Fig. 3.6 appears. Click the radio button *Amplitude-RMS L-G* to specify the rms value 120 volts directly. ATPDraw will then multiply with $\sqrt{2}$ internally (the *RMS L-L* option will also divide by $\sqrt{3}$). To use a Variable (see p. 73) for the AmplitideA value the *Peak L-G* (standard, no scaling) option is required. A negative value for StartA parameter means that the source is active during steady-state initialization.

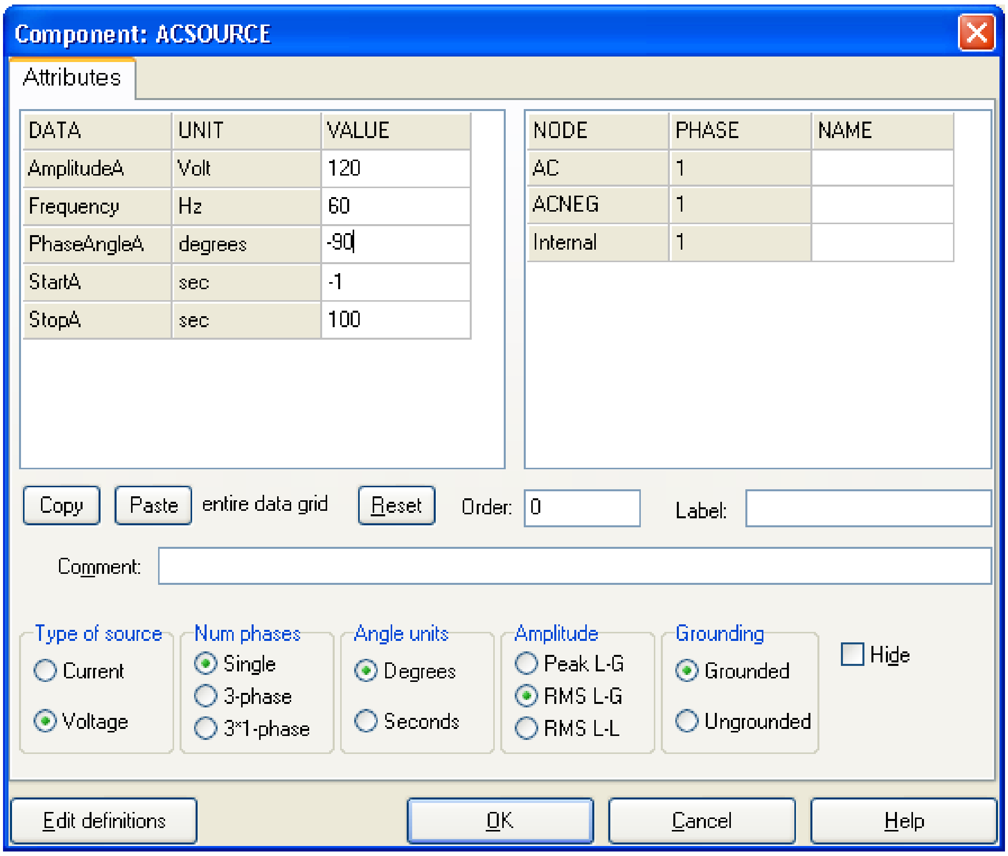


Fig. 3.6 - Component dialog box of the single-phase sinusoidal source.

Data values shown in Fig. 3.6 refer to the circuit parameters of Fig. 3.3. 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 5 input data and one node; AC (ACNEG and Interanl nodes disappear for grounded voltage sources). Click on the *HELP* button to learn about the meaning of parameters.

The node names can also 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. Next select the source inductance as shown in Fig. 3.7:

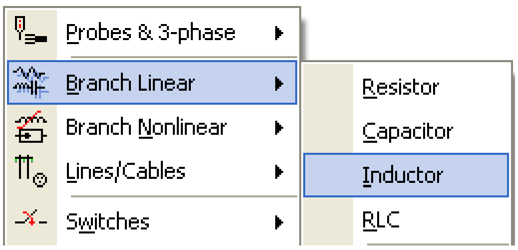


Fig. 3.7 - Selecting an inductor.

After you have clicked in the *Inductor* field, the selected inductor appears in the circuit window enclosed by a rectangle (an optional, parallel damping resistance is included). Click on it with the left mouse button, hold down and drag it to a position shown in Fig. 3.8:

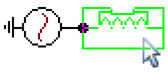
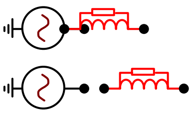


Fig. 3.8

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 gray style.)

The inductor in Fig. 3.8 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 two situations where the inductor has been misplaced and are disconnected. To correct the lower example, 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 below.



Click on the object with the left mouse button, hold down and drag it to the proper position, then click on white space. The grid snap feature will help you.

Fig. 3.9 – Not connected!

When you have placed the inductor, you can add the damping resistance (possibly directly included). 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. 3.10. Click in open space to place/unselect it.

This resistor is supposed to be parallel with the inductor and connections will be drawn later. The resistor in Fig. 3.10 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 (also warnings will be issued by the circuit compiler).

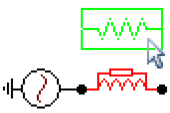


Fig. 3.10

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. 3.11.

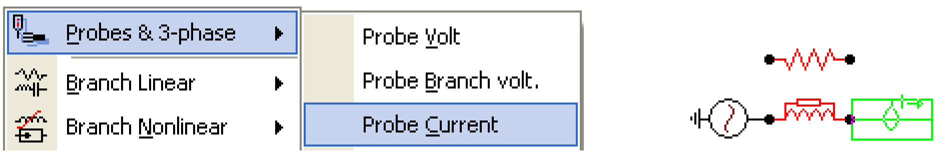
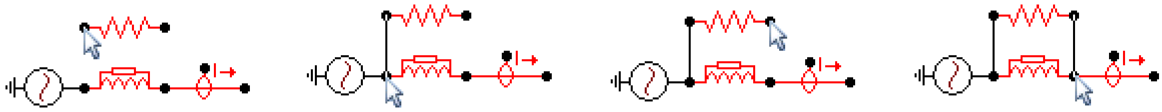


Fig. 3.11 - Selecting a current measuring probe.

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.

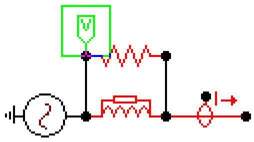
At this stage of the building process, it is time to draw some connections in the circuit diagram. To draw a connection you just click the left mouse button on a node, release the button and move the mouse. The cursor style now changes to a pointing hand and a line is drawn between the starting position 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 Connection dialog (color, phase number) automatically appears for connections drawn between multi- and single phase nodes, but not in this case.



Click left   Click left   Click left   Click left
Release + move   Release + move

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. 3.11). The probe is drawn in the circuit window in marked and moveable mode. Use the left mouse button to drag and place the probe 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 **Feil! Fant ikke referansekilden.**. 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.

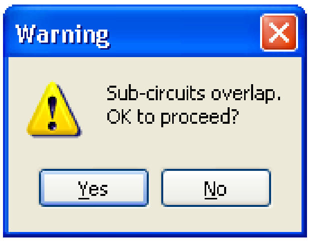


Fig. 3.12 - Prevent object overlap.

If you click on *No*, the object is 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. If objects with the same icon completely overlap the visual unambishiousity is violated (what you see is not what you get). A warning is thus issued during the compilation (MakeFile/run ATP).

Now, give data to the components placed so far. Click with the **right mouse button** on the resistor and inductor icon, respectively. The inductor has a built in damping resistor option, but turn this off by choosing Kp=0.

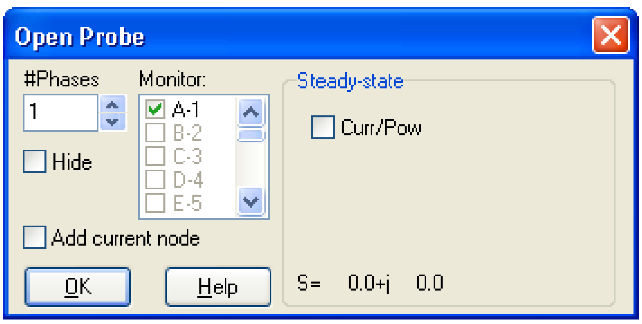


Fig. 3.13 - Open probe dialog box.

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. 3.13

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. 3.14. After you have clicked on *Diode (type 11)* the diode appears in the circuit window enclosed by a rectangle.

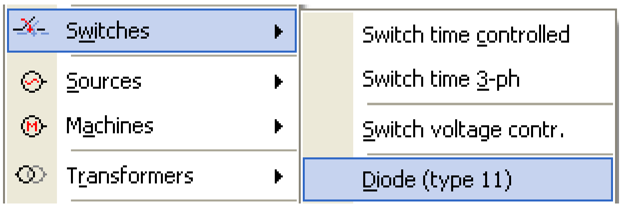


Fig. 3.14 - Selecting a diode.

The diode has to be rotated so click the right mouse button or select *Edit* in the main menu and click on *Rotate L*. 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. 3.15.

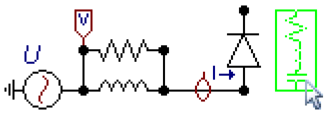


Fig. 3.15

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. 3.7).
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. Click the *HELP* button to see this help text. 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. 3.15 has been given a resistance of 33 Ω and a capacitance of 1 µF. The icon then changes to a resistor in series with a capacitor.

You have now given data to the diode and the RLC branch and instead of repeating the drawing and data entering 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| Inside* 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. 3.16.

Fig. 3.16 - 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. 3.16 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.

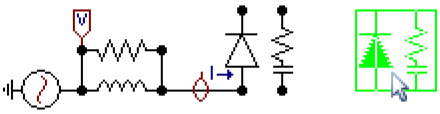


Fig. 3.17 - Move a group.

The enclosing 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. 3.17.

If you misplaced the group you can reselect it or use the *Undo* facility found in the *Edit* main menu field.

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. 3.18.

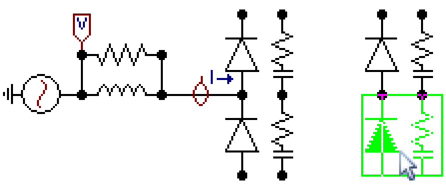


Fig. 3.18

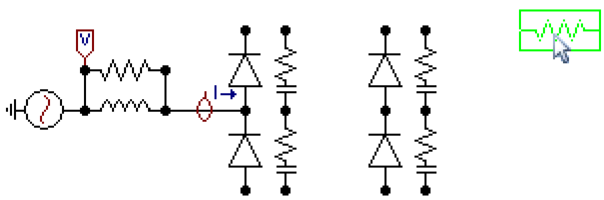


Fig. 3.19

As part of the connection between the rectifier bridge and the load a small resistor is included in Fig. 3.3. 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. 3.19. 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 the branch current in the subsequent ATP run. Having closed the component dialog box a small $\vec{I}$ symbol appears on the top-left side of the resistor indicating the current output request.

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. 3.20. 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. 3.20 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.

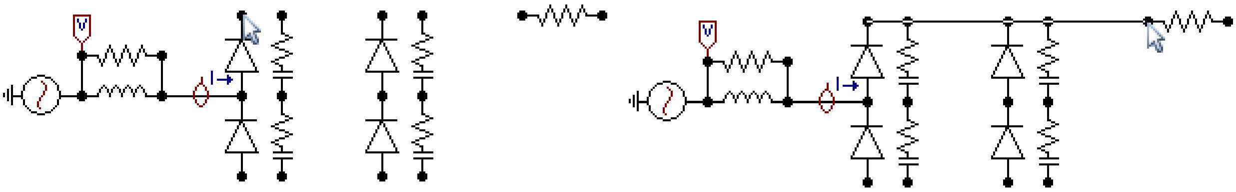


Fig. 3.20 - Click left button. Release + move, then click left button to place the connection.

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 and hold 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. 3.21:

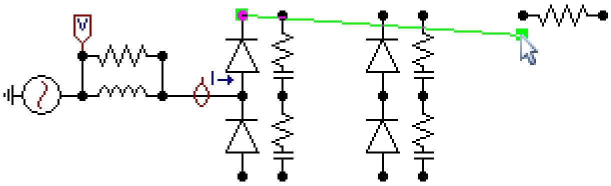


Fig. 3.21 - Edit connection. Click any point of the line then click node squares and stretch.

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. 3.22:

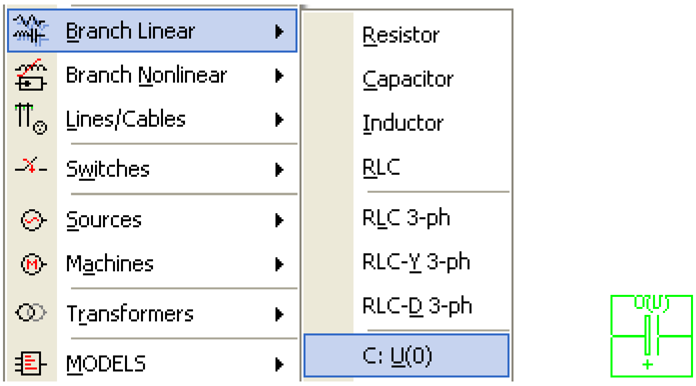


Fig. 3.22 - Select capacitor with initial condition.

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 (or press *Ctrl+R*) to orient the capacitor as shown in Fig. 3.23. Finally, click on open space to place the capacitor.

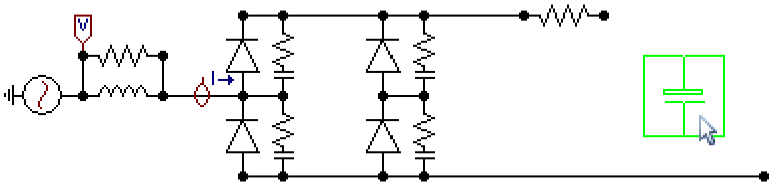


Fig. 3.23 - Placing a capacitor with initial conditions.

Next select the load resistor in the component selection menu *Branch linear* + *Resistor*. The resistor is drawn in moveable mode in the circuit window. 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. 3.24.

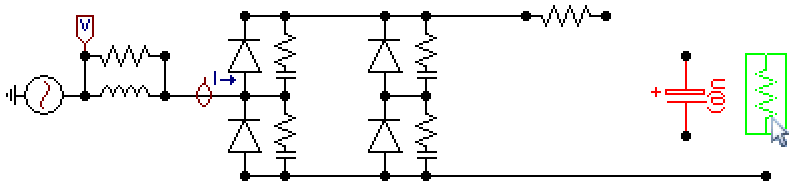


Fig. 3.24 - Place load resistor.

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. 3.25.

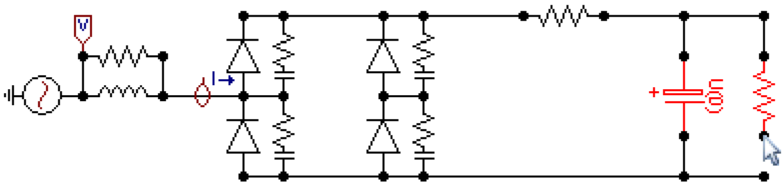


Fig. 3.25 - Your first circuit is almost ready!

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. 3.26.

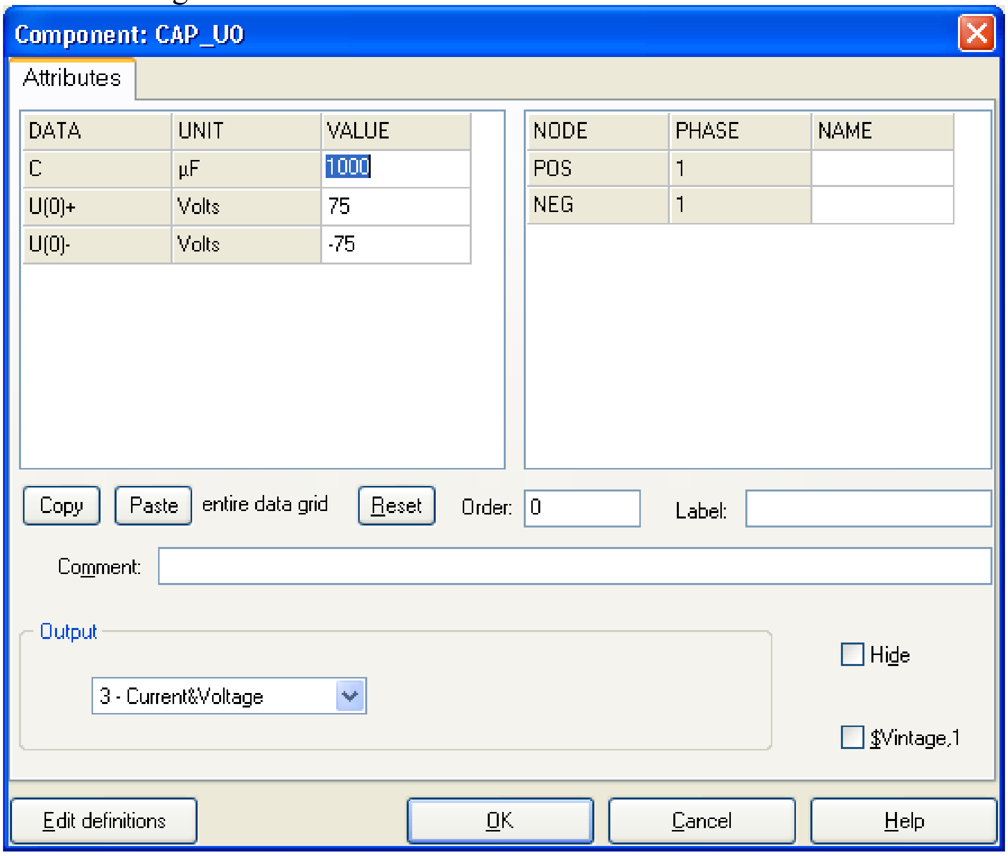


Fig. 3.26 - Capacitor data with initial condition.

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 UI→ symbol appears on the top-left side of the capacitor, indicating the branch voltage and the current output requests (see Fig. 3.27).

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 UI→ 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 name 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. 3.27-Fig. 3.30 show how to give data to four different nodes.

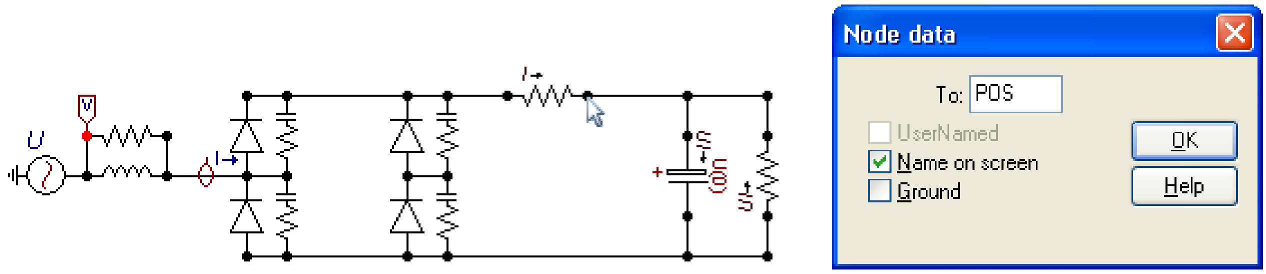


Fig. 3.27 - Click on a node with the right mouse button and specify a name in the dialog box.

When you exit the window in Fig. 3.27 by clicking *OK*, the circuit is updated as shown in Fig. 3.28 and the node dot turns red. 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.

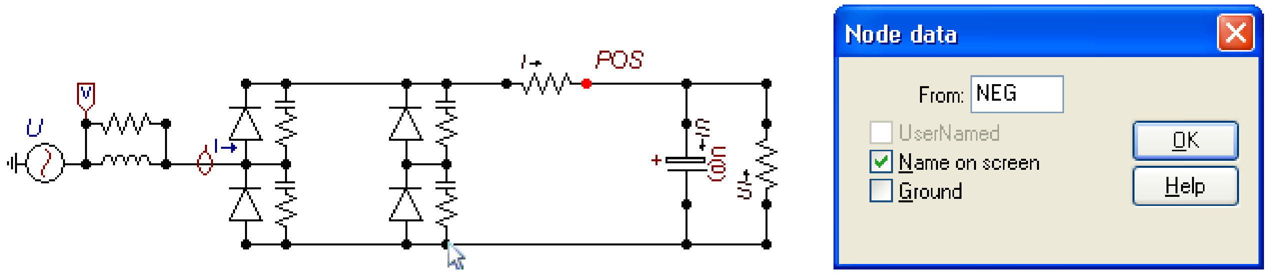


Fig. 3.28 - Click on a node with the right mouse button and specify a name in the node data window. The name ‘NEG’ will be assigned to all nodes visually connected.

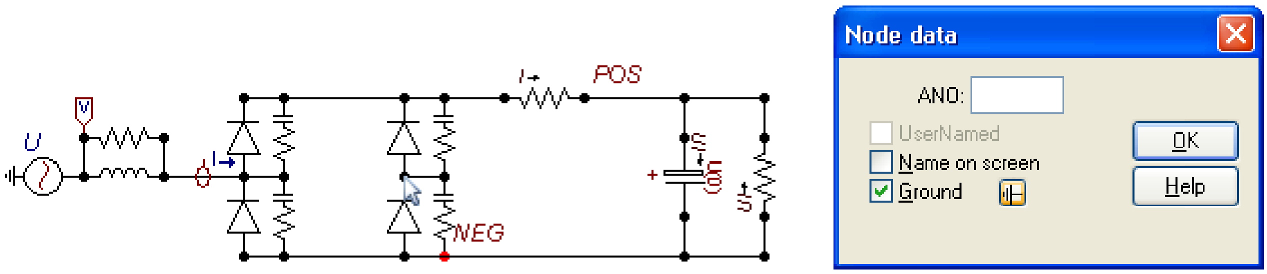


Fig. 3.29 - Click on a node with the right mouse button and check the *Ground* box indicating that the node is connected with the ground reference plane of the circuit. The button right to the *Ground* check box can be clicked to choose the ground symbol orientation.

The ground symbol is drawn at the selected node when you exit the window as Fig. 3.30 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 with a name specified by the user are drawn in a red color and the disabled check box *User Named* in their node dialog box is checked. Fig. 3.30 shows the final step in the drawing process.

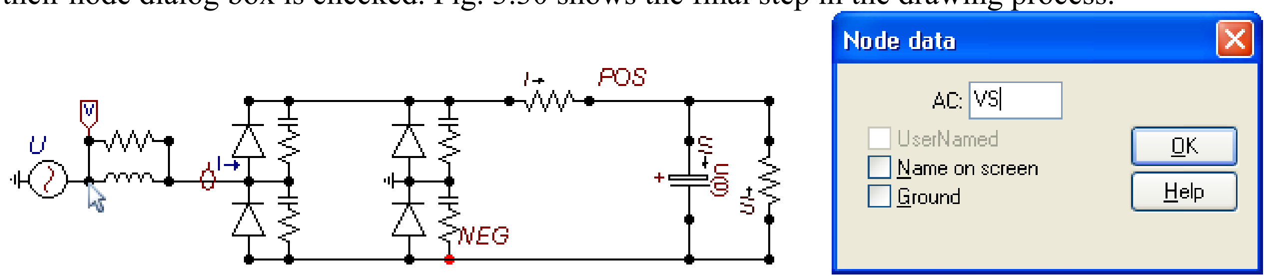


Fig. 3.30 - Click on the voltage source with the right mouse button and specify the node name.

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 *.acp* in both cases and it is automatically added to the file name you enter.

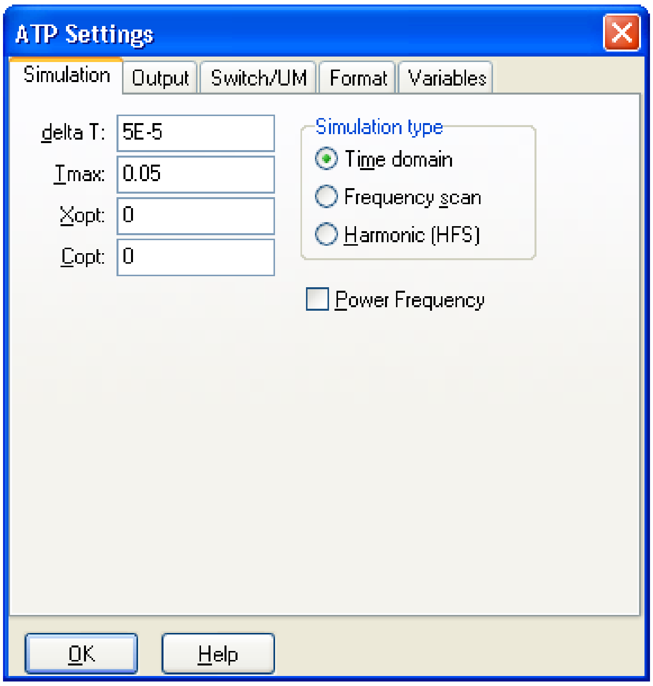
When the circuit is saved, the name of the disk file appears in the header field of the circuit window. Then if you hit *Ctrl+S* or press the *Save circuit* icon in the Toolbar, the circuit file is updated immediately on the disk and the *Modified* flag in the status bar disappears. The *File + Save 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. There are no project file name restrictions.

3.5.3 Creating the ATP input file

The ATP-file describes the circuit according to the ATP RuleBook. You can create this file by selecting *Sub-process|Make ATP File* command in the *ATP* main menu. The ATP-file is regenerated whenever you execute the *run ATP* command (or press *F2*). In the latter case the process is hidden for the user. By default the ATP file inherits its name from the project file.

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. 3.31 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* dialog.



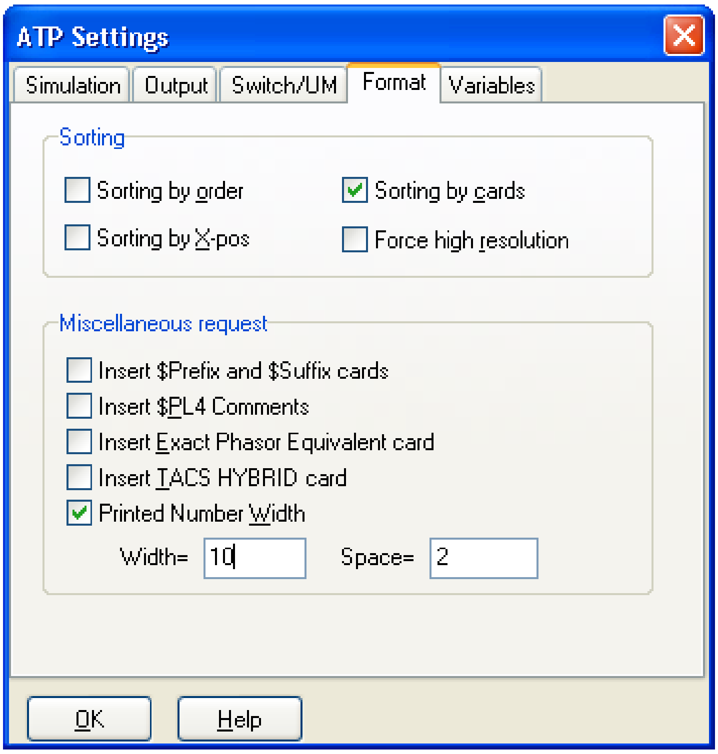
Select:

- Time step *delta T* in sec.
- End time of simulation *Tmax* in seconds.
- Xopt=0: Inductances in mH.
- Copt=0: Capacitances in µF.

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. 3.31 - 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. 3.32 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. This is not a required choice.

All other check boxes are unselected

Fig. 3.32 - The ATP-file format menu.

To create an ATP-file without starting the simulation you must select the *Sub-process|Make ATP File* in the *ATP* menu. This selection will start the compilation, 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 with extension `.acp` file is suggested default. As the ATP file is sent to the ATP solver, the file name should not contain space characters. 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 be as follows:

```
BEGIN NEW DATA CASE
C --------------------------------------------------------
C Generated by ATPDRAW  November, Thursday 5, 2009
C A Bonneville Power Administration program
C by H. K. Høidalen at SEfAS/NTNU - NORWAY 1994-2009
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. ATP|run Plot (F8) starts the default plotting program and sends the pl4 file as parameter. The default commands that is executed when the user selects *run ATP* or *run Plot* under the *ATP* menu can be specified under the *Tools | Options /Preferences* tab as it has been described in section 2.6 of the Installation Manual.

3.6 Multi-phase phase circuits

From ATPDraw version 5 a node can have up to 26 phases (A..Z node name extension). This applies also to MODELS nodes. A more generalized *Connection* is introduced with a special handling between single phase and n-phase nodes. Transpositions will only take place through 3-phase connections. In this case the phase sequence will be further inherited throughout the circuit. Special ABC or DEF reference components found under from *Probes&3-phase* in the Selection menu can be placed on the reference node. The actual phase sequence of the node is written at the top right of the Node dialog box or in the PHASE field in the Component dialog box as shown in Fig. 3.2 (after *ATP|run ATP* or *ATP|Sub-process|Make node names*). A special component SPLITTER is available for connections between 3-phase and single phase nodes. 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, 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! The best solution is to specify a node name on the 3-phase side only.

Color, label, and phase properties are given to the *Connection* as well as the possibility to force node dots on. The connection can also be turned into a *Relation* (no node connection only visualization of flow of information drawn as a dotted line) by the Relation check box. Fig. 3.33 shows the Connection dialog that appears after a right click on the connection and automatically when the user draws a connection between a single phase and a multi-phase node. The *Phase index* field is only enabled for single phase connections. 0-@ is used for connections between two single phase nodes.

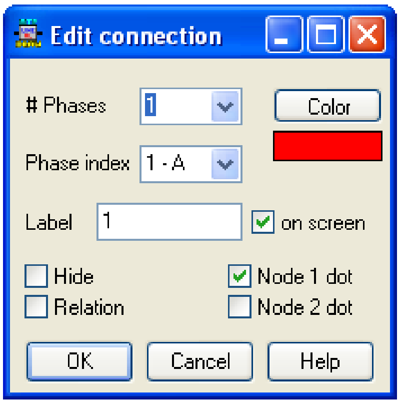


Fig. 3.33 - The Connection dialog box.

Fig. 3.34 illustrates the various options for (3-phase in this case) multiphase circuits in ATPDraw. The flag DEF set at the source node to the left. Consequently, all connections marked with 1 will carry the phase D and so on. The color of the connections is user selectable as shown in Fig. 3.33, but as default the color and phase sequence are inherited when the user clicks on one connection to draw a new one. Connections will inherit the phase number.

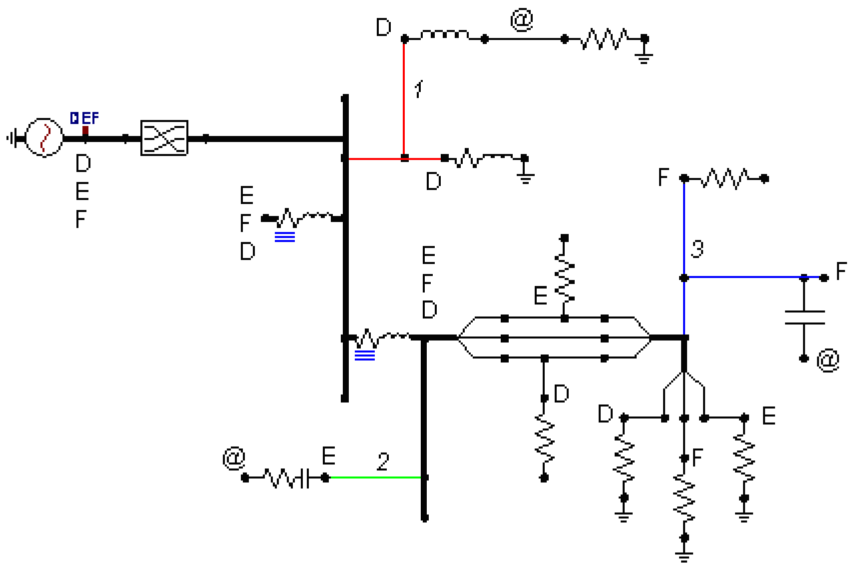


Fig. 3.34 - Illustration of various phase options in ATPDraw.

A typical example of connecting a single phase node to a 3-phase node is the case of a single phase ground fault as shown in Fig. 3.35. Place the switch, then draw the connection between the three phase node and the single phase node. Select 1-A to ground phase 'A' (regardless of transpositions involved).

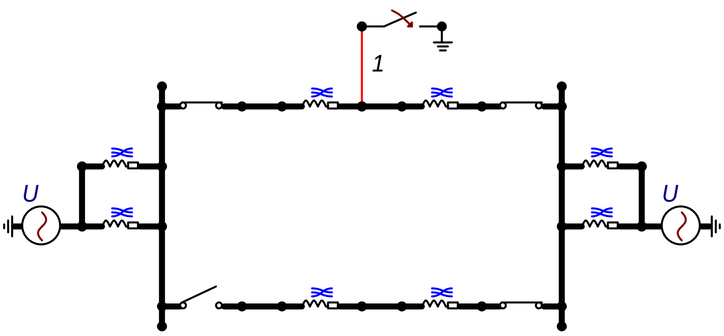


Fig. 3.35 - Single phase ground fault.

Multi-phase nodes are first of all important for *MODELS* and *GROUPS*. An *n*-phase connection could also be useful just to clear up the circuit drawing. As an initial example a 6-phase connection is shown in Fig. 3.36 for communication between a 6-pulse thyristor bridge and its control circuit. This will make the drawing much easier to read.

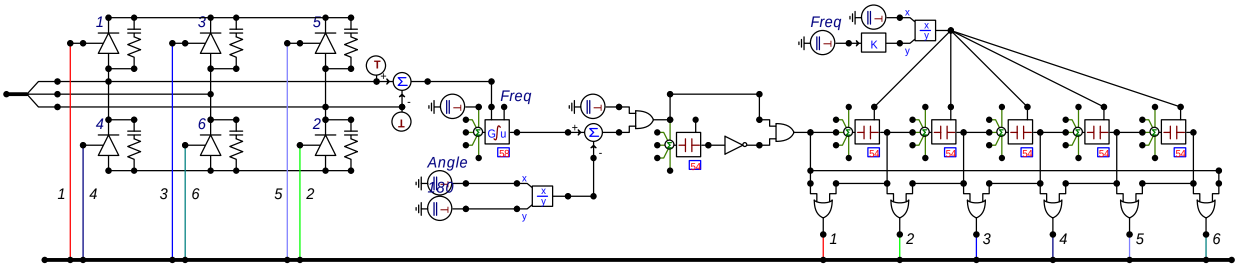


Fig. 3.36 - Communicating a 6-phase signal between a thyristor bridge and its control circuit.

All n-phase nodes have only 5 characters available in the *Node dialog box*. ATPDraw adds the extension *A*, *B* and *C* (etc.) 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. Only 3-phase nodes can be transposed.

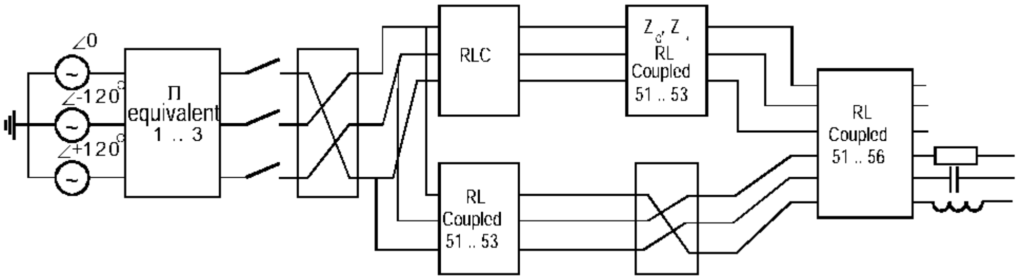


Fig. 3.37/a - Illustrative three-phase circuit.

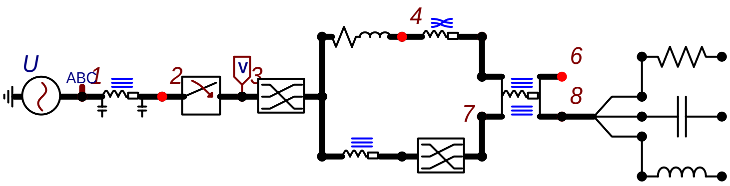


Fig. 3.37/b - Equivalent ATPDraw circuit (`Exa_2.adp`).

The circuit shown in Fig. 3.37 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. Fig. 3.38 shows the Node data dialog for a single phase and a three phase node.

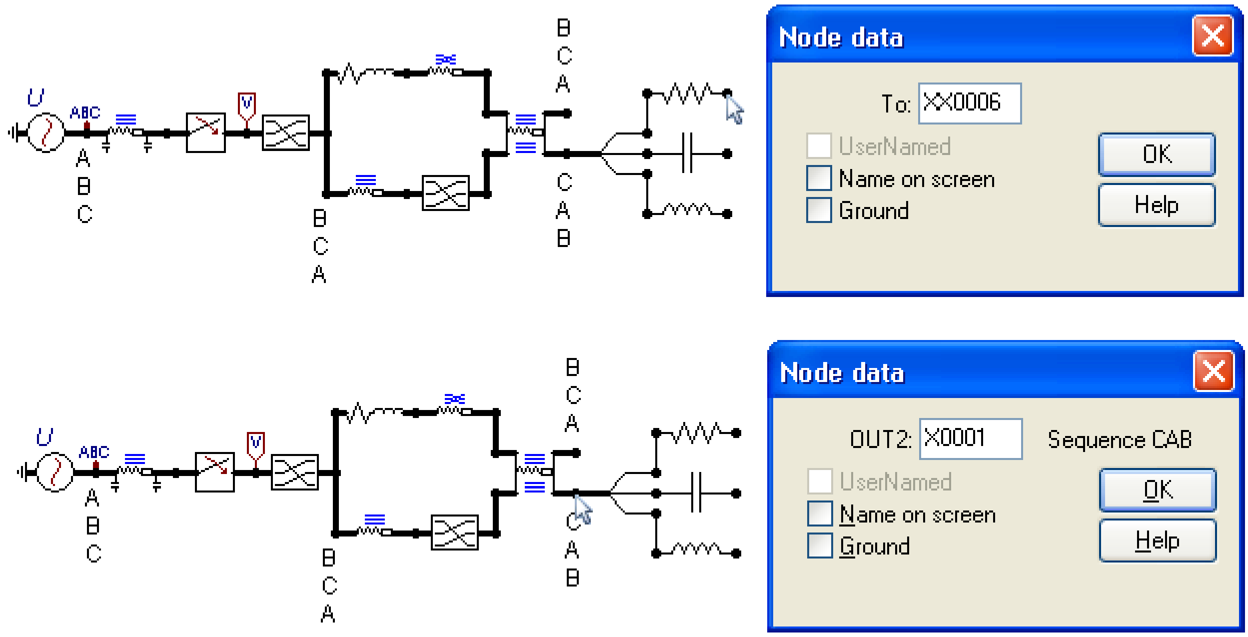


Fig. 3.38 – Default node names and phase sequence. Top: single phase node. Bottom: 3-phase.

4. Reference Manual . . .

. . . .

ATPDraw™
for Windows
5.6

This part of the manual outlines all menu items and program options, and gives an overview of the supported ATP components, TACS, 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

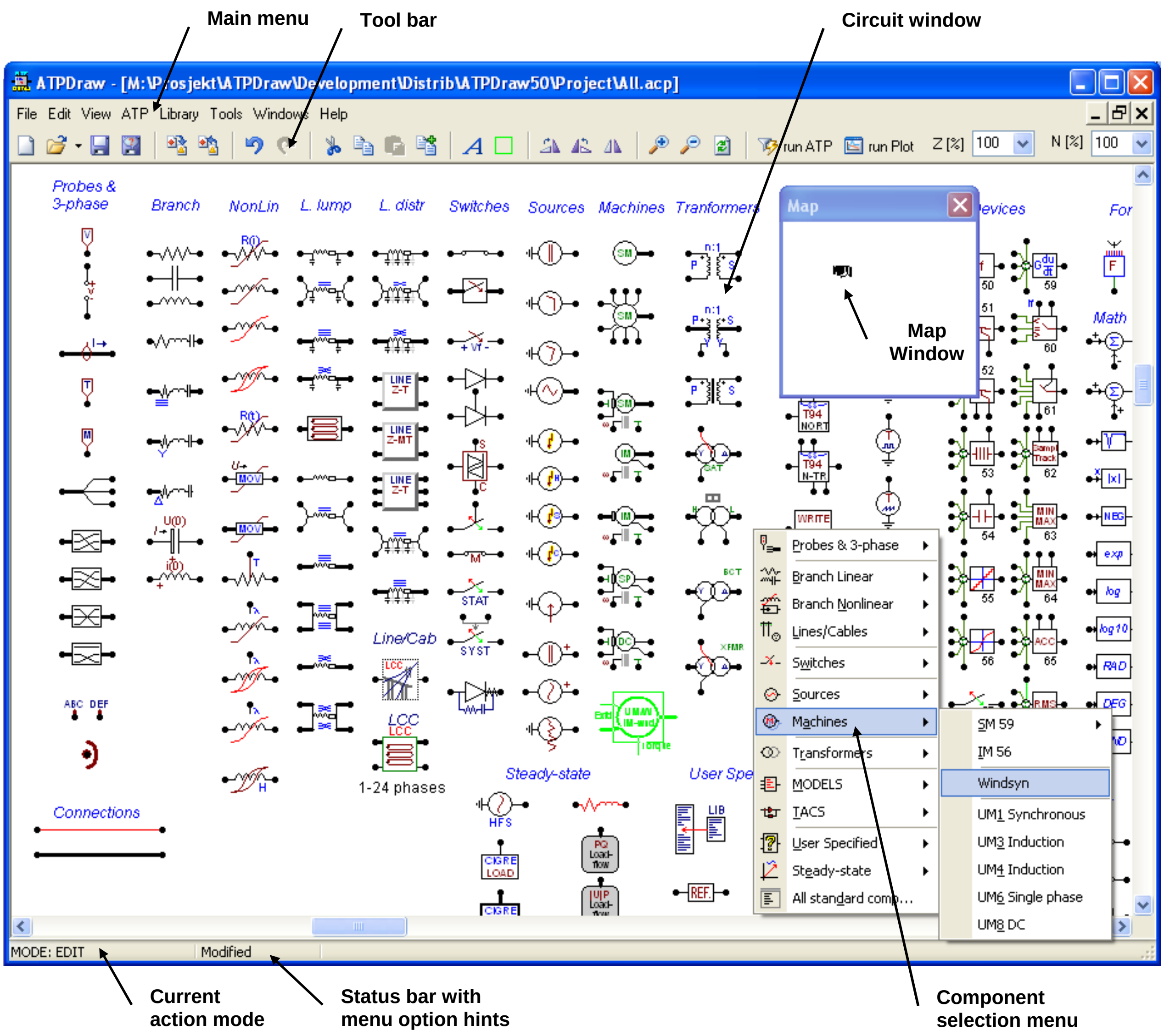
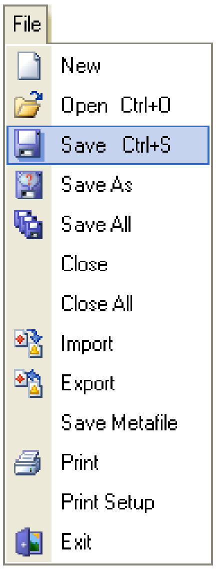


Fig. 4.1 - Components of ATPDraw's main window.

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. In this window the user can select a project file and load it into ATPDraw. Short key: *Ctrl+O*. The default directory is the previously used directory and the first time the dialog is used the *Project Folder* set under *Tools|Options/Files&Folders* (initially read from ithe ATPDraw.ini file) is suggested.

ATPDraw can read both circuit (*.cir*) files created by an earlier version of the program and project files (*.acp* and *.adp*). When opening a project file all data are stored in memory and no files are written to disk. The circuit files and project files are binary data files.

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

4.2.1.3 Save

Activating this menu item will save the project in the active circuit window into a disk file. If the name `Noname.acp` 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*.

4.2.1.4 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*. 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 newest file format. The default extension of the project files on disk is (*.acp*).

4.2.1.5 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.

4.2.1.6 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. 4.3 to confirm before the circuit is closed. If the project has been modified, the user is given a chance to save it first.

4.2.1.7 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.

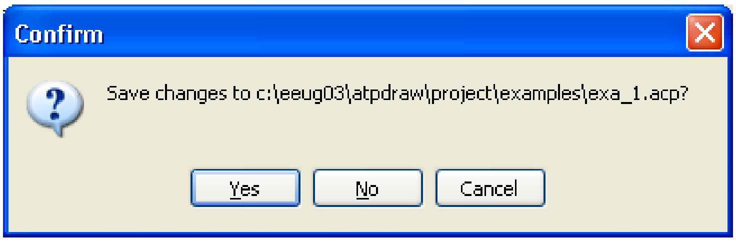


Fig. 4.3 - Confirmation prevents the loss of unsaved project data.

4.2.1.8 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.

4.2.1.9 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 are written to the disk file.

4.2.1.10 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 for this format.

4.2.1.11 Print

Print the graphics on the currently selected printer.

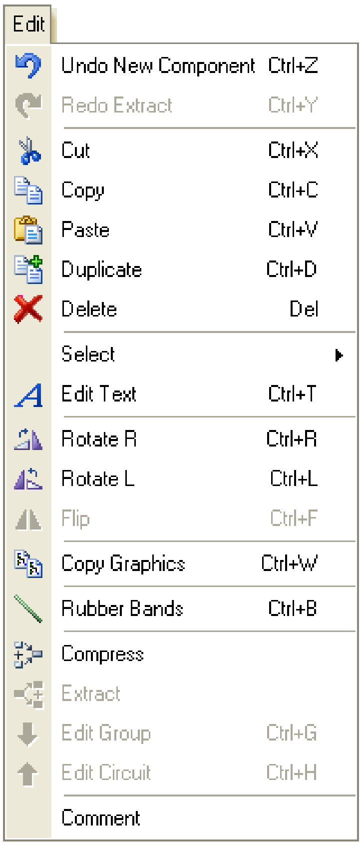
4.2.1.12 Printer Setup

Select and setup the printer.

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. 4.4.

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 enclosed by a lime colored frame indicating that it is selected.

Fig. 4.4 - 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: *Ctrl+Z* and *Ctrl+Y*. 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, edit, 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 (this included also the extensive data in LCC, BCTRAN, XFMR). 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. After *Save*/*Save As* the Undo/Redo buffer is cleared.

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. The node names are deleted when pasting components.

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 Delete

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 five sub-menus:

`None:` To cancels the object selection. Short key: *Ctrl +N*.
`All:` Select all objects in the current circuit window. Short key: *Ctrl +A*.
`Inside:` Enables object selection by a polygon shaped region. Short key: *Ctrl +I* (or double-click with the left button in an empty region of the circuit window).
`by Properties:` Enables selection by objects' support file name or order number (see below). Short key: *Ctrl +P*.
`Overlapped`: Select component that overlap other components. First *ATP|run ATP* must be chosed to identify overlapping component.

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/Delete* 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.

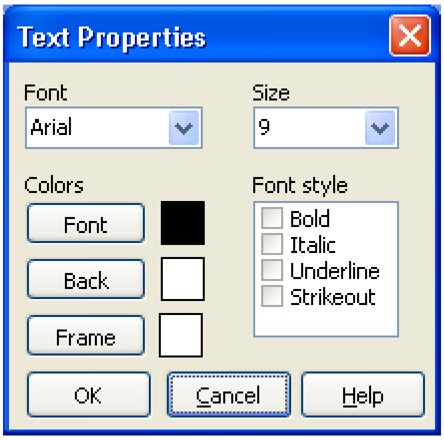
In *Inside* mode, the mouse cursor icon changes its style to a pointing hand and moves to the middle of the circuit window. The current action mode also changes to *MODE:GROUP* in the status bar. To draw a polygon around a group of objects move the cursor to the starting location and click the left mouse 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 *by Properties* selection mode the group of components can be selected by their type and/or *Order* number. The type here is the name of the support file and the Order number is the identifier specified in the component dialog box.

The available component Names and Order numbers are listed in two combo boxes as shown in Fig. 4.5. When you click on *OK* the components with the selected order 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. 4.5 - Selecting objects by name or group no.

4.2.2.9 Edit Text



This menu is used to insert a new circuit text. In addition the selection of texts, component labels or node names is favoured in this mode. An alternative to this last property is to press the *Alt* key. This is beneficial when texts, labels or node names are drawn overlapped by components. If you click on existing texts, labels or node names you can edit the text directly on screen or move them (click and hold). Short key: *Ctrl+T*.

Fig. 4.6 – The circuit text dialog box. It appears after a right click (or left double) on a circuit text.

Selecting the *Edit Text* menu item, the mouse cursor style will change to a pointing hand and forced to stay within the circuit window. The action mode indicator in the status bar will also change to *MODE: EDIT TEXT*. You can leave this mode by pressing the ESC key.

4.2.2.10 Rotate R/L

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

4.2.2.11 Flip

Mirrors the icon left to right. For vector icons the texts are not flipped. This option is useful for instance for transformers since the primary and secondary node will be swapped. Short cut *Ctrl+F*.

4.2.2.12 Copy Graphics

Copy the selected graphical content to the Windows clipboard in MetaFile format.

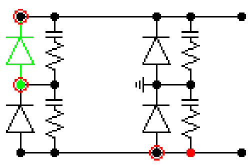
4.2.2.13 Rubber Bands

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.

4.2.2.14 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. 4.7/a) appears where the user designs the new group object. The user can later modify a compressed group by selecting it and click Compress once more.

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. Only the members of the group are shown in the Compress process and moved to the middle of the circuit window.



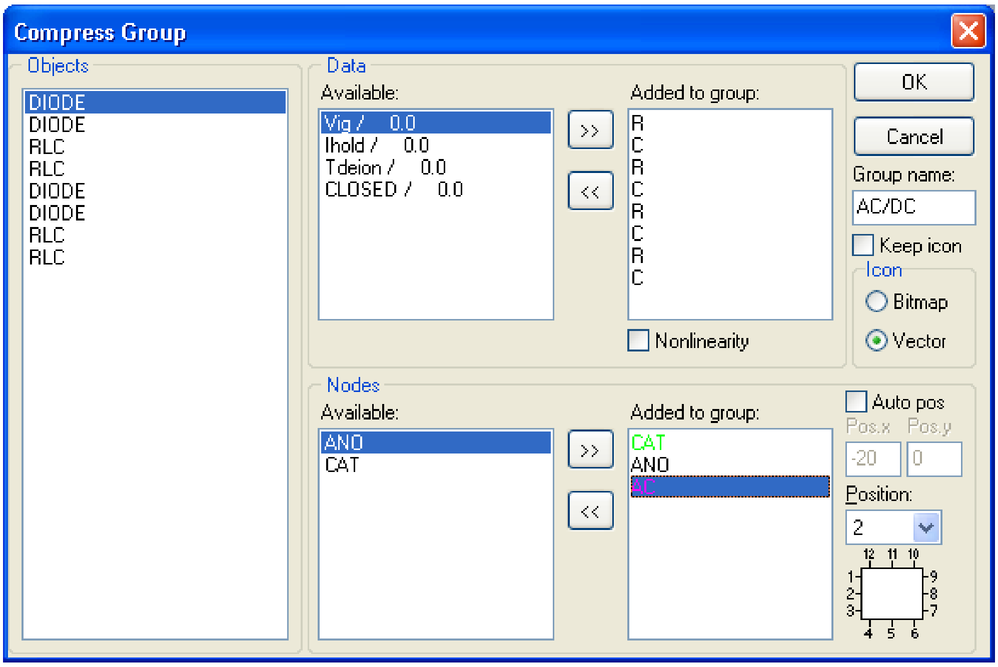


Fig. 4.7/a - The Compress dialog box.

Under *Objects:* all the components in the group are listed with their name followed by their label. When the user clicks on one of the components' name the selected component is drawn in a lime color in the circuit window. Its data and nodes also 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. Data and Nodes in the *Available* list that already are members of the *Added to* lists will be displayed there with a lime colored text. Selected node in the *Available* list will be drawn in a lime color. All data and nodes listed in the *Added to group:* will be an external attribute of the new group object. The selected external nodes are drawn enclosed by a red circle. The position of the external nodes are selected in the *Position* combo box. Positions 1-12 will be on the traditional border as shown in the graphic below, while position 0 will enable the user to specify positions in the Pos.x and Pos.y fields. 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. 4.7/b). Selected data and nodes can also be removed from the *Added to group:* by clicking on the << button. The *Keep icon* check box can be used when Recompressing a group in cases where the user wants to keep its icon.

As all other components, the group object is limited to 64 data and 32 nodes. When you later 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 group members as shown in Fig. 4.7/b. Node that the 8 selected data are represented by two external data in Fig. 4.7/b since the names are duplicated.

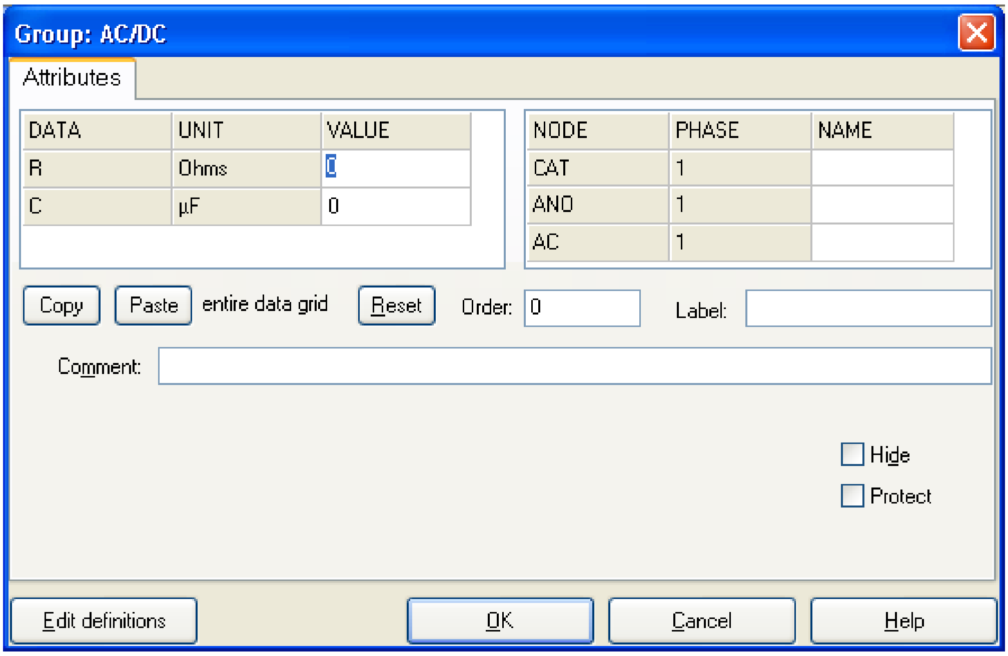


Fig. 4.7/b - Component dialog box for a sub-group object.

4.2.2.15 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.

4.2.2.16 Edit Group

This command shows the group content. Short key: *Ctrl+G*. The group is shown 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.

4.2.2.17 Edit Circuit

Displays the circuit to which 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.

4.2.2.18 Comment...

Opens a comment dialog box, 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.

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. 4.8.

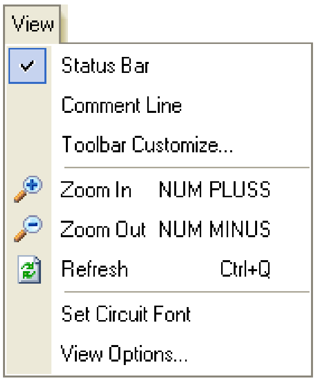


Fig. 4.8 - View menu.

4.2.3.1 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.
EDIT TEXT	Indicates a text edit mode. Add a new circuit text or favour text selection (circuit text, labels and node names). Enter this mode also via the *Alt* 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 to initiate the drawing of a new relation. Relations 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 and Drag-over information.

4.2.3.2 Comment Line

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

4.2.3.3 Toolbar customize

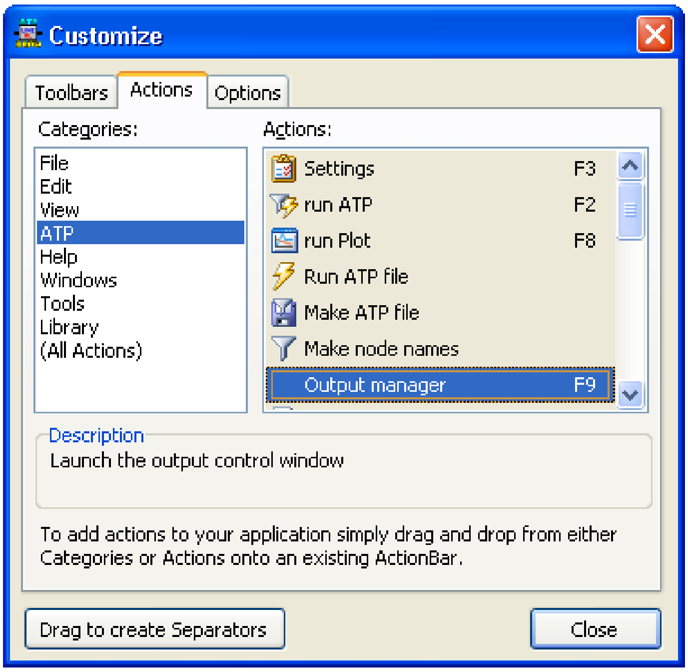


Fig. 4.9 – Customize toolbar dialog.

The toolbar can be customized by the user. The description of the user defined toolbar is stored in the file Toolbar.cfg located together with the ATPDraw.ini file in the %APPDATA%\atpdraw directory. The format and handling of the Toolbar.cfg is managed by Delphi and there might be problems (main menu items missing/wrong) when changing ATPDraw version. Shutting down ATPDraw and deleting the Toolbar.cfg file will fix this (but reset the toolbar to the default content). All main menu items (called actions) can be member of the toolbar. From the Customize dialog shown in Fig. 4.9 the user can drag items/actions on/off the toolbar.

The default toolbar content is:

From the left the tools are:

Item Menu	**Shortcut**	**Description**
New File|New Open	--	Open an empty circuit file.
File|Open	CTRL+O	Loads a circuit file into a new window. Contains also a dropdown with the five recent opened projets.
Save File|Save	CTRL+S	Saves the active circuit window to the current project file.
Save As File|Save As	--	Saves the active circuit window to a new project file.
Import File|Import	--	Inserts a stored circuit into the current circuit.
Export File|Export	--	Export the selected circuit to an external project file.
Undo Edit|Undo	CTRL+Z	Undo the previous operation.
Redo Edit|Redo	CTRL+Y	Redo the previous undo operation.
Cut Edit|Cut	CTRL+X	Copy the current selected circuit to the clipboard and then delete it.
Copy Edit|Copy	CTRL+C	Copy the current selected circuit to the clipboard.
Paste Edit|Paste	CTRL+V	Paste the ATPDraw-content from the clipboard into the circuit.
Edit|Duplicate	CTRL+D	Copy+Paste.
Edit|Edit text	CTRL+T	Go into Edit text mode for adding and selecting text. Required to add new text to the circuit window.
Edit|Select|All	CTRL+A	Select the entire circuit.
Edit|Rotate-R	CTRL+R	Rotate 90 deg. clock-wise.
Edit|Rotate-L	CTRL+L	Rotate 90 deg. counter clock-wise.
Edit|Flip	CTRL+F	Flip left-to-right. The nodes changes postion. Vector text is not flipped.
View|Refresh	CTRL+Q	Redraw circuit.
View|Zoom in	NUM +	Zoom in 20 %.
View|Zoom	NUM –	Zoom out 20 %.
ATP|run ATP	F2	Make node names + write the ATP file+ run ATP by executing the ATP command (Tools|Options/Preferences).
ATP|run Plot	F8	Plot Executed the Plot Command (Tools|Options/Preferences) and send the current PL4 file as parameter.

To the right of the toolbar comes two items for controlling the zoom and the node sizes.

4.2.4 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).*

4.2.4.1 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).*

4.2.4.2 Refresh

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

4.2.4.3 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. This also becomes the default font for circuit text, but this can be adjusted individually.

4.2.4.4 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.

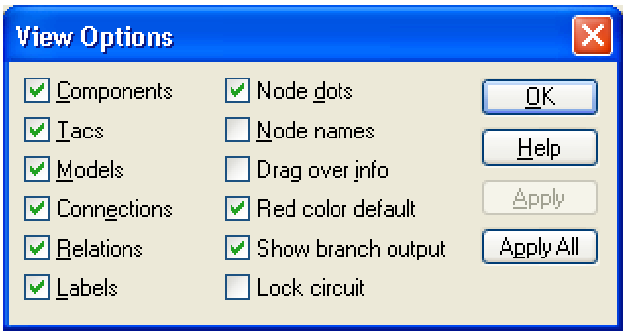


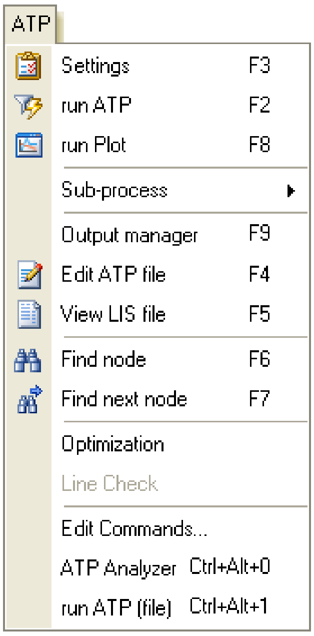
Fig. 4.10 - View Options dialog box.

By default, all objects except node names are visible. The meaning of options assumed checked (☑) 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 over info	List information about the component (name, number of data and nodes) under the mouse cursor. No clicking is required. Can slow down the application in case of large circuits.
Red color default	Components and node dots are drawn with a red color until the component or node is opened for the first time.
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.
Lock circuit	Components can not be selected and moved only opened for input.

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 (this saves the selections to the ATPDraw.ini file).

4.2.5 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. From this menu all output requests can be managed and the ATP and LIS files edited and inspected. The *Find node* and *Find next node* navigation tool is also available here. The Optimization module works with a cost function and perform multiple ATP runs. The *Line Check* feature calculate sequential parameters of transmission lines and sub-circuits. 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. 4.11.

Fig. 4.11 - The ATP menu.

4.2.5.1 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, *Format* for specification of data-card sorting options and miscellaneous request, *Switch/UM* for statistical and Universal Machine studies, and *Variables* for specification of global $Parameter and Pocket Calculator options.

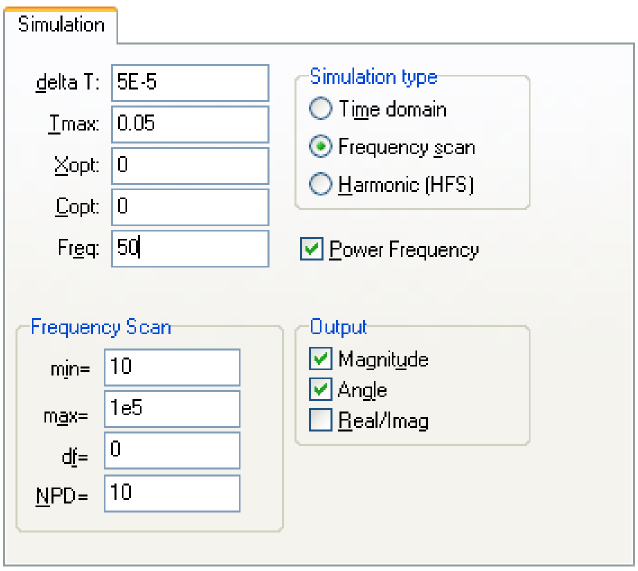


Fig. 4.12 - Simulation settings.

Simulation settings

Simulation type: Select between the simulation methods supported by ATP:

- o Time domain
- o Frequency Scan
- o Harmonic Frequency Scan (HFS)

<u>Time domain</u>

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 [mF] if zero; otherwise, capacitances in [Ohm] with *Copt* as frequency.

Freq: System frequency in Hz.

Power Frequency: when checked the SYSTEM FREQUENCY request card is written in the ATP-file. The ideal transformer component uses this frequency.

<u>Frequency scan</u>

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 4.11.12). The frequency of the harmonic source will for each ATP run be incremented. The power frequency specification is mandatory for HFS simulations.

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.

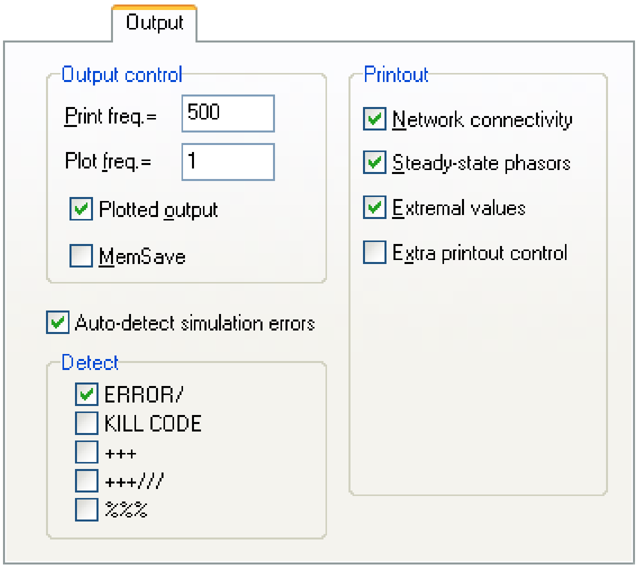


Fig. 4.13 - Output request tab.

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

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 Detect string is found, the corresponding section of the file is displayed in a text editor 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.

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.

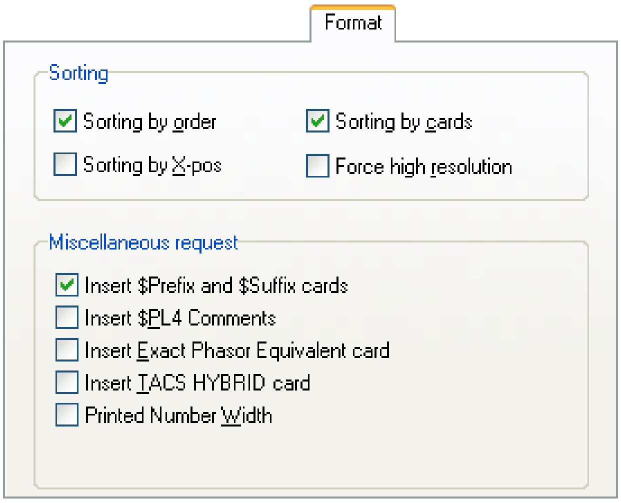


Fig. 4.14 - ATP-file format settings.

<u>Sorting</u>

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 order: The Order number that can be specified in the component dialog box for each object determines the sequence of cards. The lowest Order 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.

<u>Miscellaneous request</u>

Insert $Prefix and $Suffix cards: If this option is checked, ATPDraw will assume that all $Inlude files (User Specified , LCC, external nonlinear characteristics, and Windsyn components) are located in the Result Directory and have the extension '.lib'. Two cards $Prefix and $Suffix will the be inserted into the ATP file and the $Include commands are specified without path and extension. This should be a preferred choice as this path and extension generally are used and that increased readability of the ATP file is obtained this way.

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.

Switch/UM settings

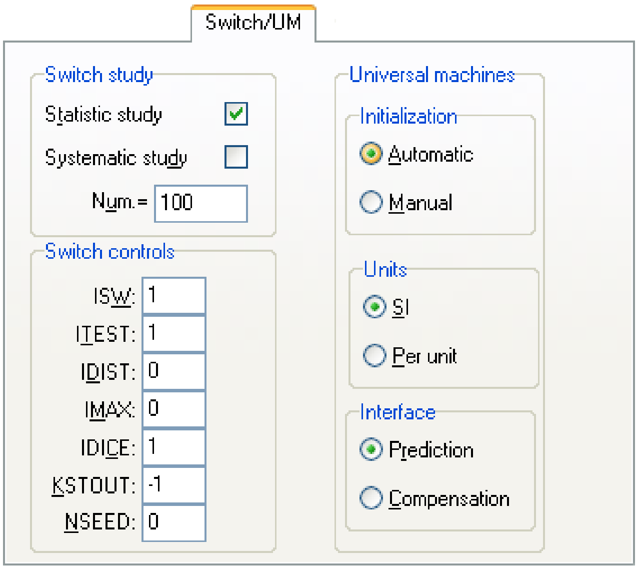


Fig. 4.15 - Switch/UM settings.

Switch study

Statistic study: Study with statistic switches

Systematic study: Study with systematic switches

Num: Number of simulations. This value influences ATP's 1st 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.

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. Preferred method.

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

Load flow

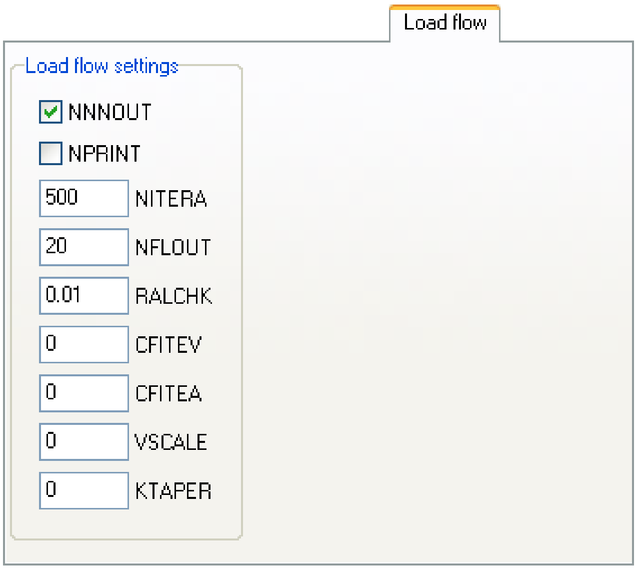


Fig. 4.16 – Global load flow settings.

Sets the global variables of load flow according to RuleBook chapt. X.

NNNOUT Additional interactive output during load flow iteration.
NPRINT Tabular printout for nodes with power constraints.
NITERA Maximum number of iterations. Default 500.
NFLOUT Buffer size convergence monitoring, printout per line. Default 20.
RALCHK Relative convergence tolerance. Default 1/100.
CFITEV Acceleration factor ref. dQ/dU. Default 2/10.
CFITEA Acceleration factor ref. dP/dTh. Default 2.5.
VSCALE Voltage scaling factor. Use 1.4142 to get rms values output. Zero=Unity.
KTAPER =0: Constant acceleration factors. =2 used also in DC25/DC26 examples.

Variables

The *Variables* dialog box 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

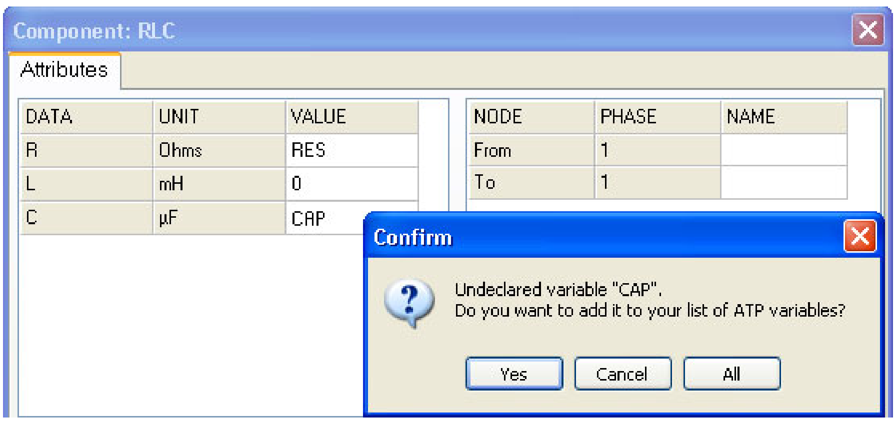


Fig. 4.17. A requirement is that property *Param* of the DATA is set to 1. This can be verified and set under *Edit definitions* in the component dialogs. In addition the data in question must not be involved in subsequent calculations. This is the case for the phase angle of 3-phase AC sources, the damping resistors of

inductors and capacitors, the advanced components LCC, BCTRAN, XFMR, Windsyn etc.

Fig. 4.17 - Using text string instead of variables in the RLC component dialog box.
A numerical value 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 their data values. This is done in free format in the column to the right as shown by Fig. 4.18. Nested syntax (the Name is used in subsequent Value specifications) is allowed from version 5.6 as ATPDraw internally handles the variables as intermediate (a character 'I' is added to the Name and the request '$$' is added to the Value). The user can also add local variables. 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 RES used both for high and low precision resistances will thus be declared twice with 3 and 13 underscore characters added. This process is hidden, but the result is seen in the final ATP-file after the `$Parameter` declaration. Also Models can utilize Variables and the default number of digits is set to 10 in this case. There is a limit in ATP on the number of internal variables.

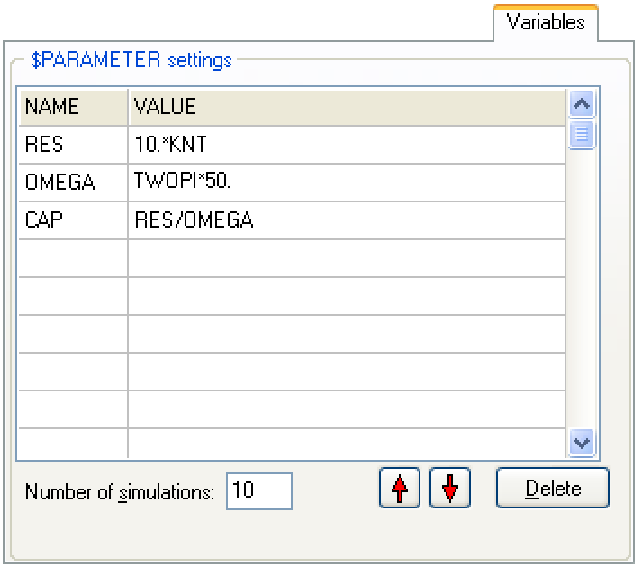


Fig. 4.18 - Setting values to text strings

The variables RES and CAP are circuit variables (6 characters) while OMEGA is a pure local variable. The ATP file becomes:

```
$PARAMETER
RESI =10.*KNT $$
OMEGAI =TWOPI*50. $$
CAPI =RESI/OMEGAI $$
RES___=RESI
CAP___=CAPI
BLANK $PARAMETER
```

KNT is the simulation number (1..10 in this case).

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

ATPDraw support some special syntax for loop control (variables as function of the simulation number KNT). These are:

MyVar=@[a b c … n]
First run (KNT=1): MyVar=*a*
Second run (KNT=2): MyVar=*b*
...
Last item and beyond (KNT >=n): MyVar=*n*
The characters '@[' are used to identify this format. Space or comma can be used to separate the numbers (integer or floating point).

MyVar=@FILE FileName Col

'@FILE' is the keyword, *FileName* is the name of a text file assumed stored in the *ResultDirectory* (same as final ATP file) (enclose the file name within " " if it contains space), and *Col* is an

optional parameter identifying which column in the text file to use. The text file can have integer or floating point values in free format space or comma separated. If Col is not specified the first column of the file is loaded. The length of the file does not need the match the chosen *Number of Simulations*.
First run (KNT=1): MyVar=First value of column Col
Second run (KNT=2): MyVar=Second value of column Col
etc.
Both the '@[' and '@FILE' syntax requires a lot of intermediate variables and ATP puts a limit on this.

MyVar=@LIN Lo Hi

'@LIN' is the keyword. Creates a linear space. MyVar=a*(KNT-1)+b

MyVar=@LOG Lo Hi

'@LOG' is the keyword. Create a logarithmic space. MyVar=10**(a*(KNT-1)+b)

MyVar=@POW Lo Hi P

'@POW' is the keyword. MyVar = a*(KNT-1)**P+b

MyVar=@EXP Lo Hi P

'@EXP' is the keyword. MyVar = a*P**(KNT-1)+b
If P ='e' this is replaced by exp(1)

a and b are calculated based on Lo and Hi: First run (KNT=1) MyVar=Lo, Last run (KNT=Number of Simulations) MyVar=Hi. The last four options could easily be managed directly be the user.

The user should normally not change the name of the variables listed by ATPDraw in the NAME column, but if you do you will be asked to take an Action regarding the old Variable still defined in the circuit, as shown in Fig. 4.19. The action can be to reset the parameter to zero or the default value or to assign a new variable name.

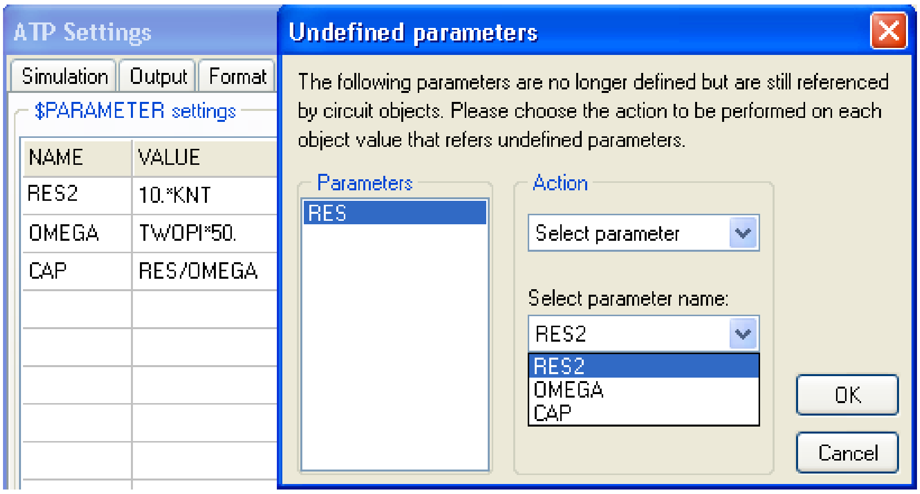


Fig. 4.19 - Actions to take when non-defined parameters are found.

4.2.5.2 Run ATP

Executing the *run ATP* command at the top of the ATP menu will create the ATP input file (the project file name (with extension `.atp`) and the `/ATP` system folder are default, but changeable via *Sub-process|Make ATP file*). Then ATP is executed basded on the default *ATP command* (specified in the *ATP* field of the *Preferences* page under *Tools | Options*). The current ATP-file is sent as parameter to the ATP-EMTP. Note that users do not need to select *Make Names* and *Make ATP File* 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, use *ATP|Sub-process|run ATP file*. After executing ATP, ATPDraw examines the LIS-file and displays any error or warning messages if exist.

4.2.5.3 Run Plot

Execute the *Plot program* (defined under *Tools|Options/Preferences*) with the current ATP file name and the extension .pl4.

4.2.5.4 Sub-process

This sub-menu contains the individual three parts of the run ATP command.

- Run ATP file: Executes ATP and sends the current ATP file as parameters. This choice must be used if the user has manually modified the ATP file under *ATP|Edit ATP file*.
- Make ATP file: Creates the ATP file from the circuit without executing ATP (but calls Make node names first). This choice must be used to change the current ATP file name and the Result Directory.
- Make node names: Gives node names to all nodes in the circuit. Overlapping and/or connected nodes get the same name. Whenever a "same name on different nodes" or "duplicate names on same node" are 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. *Make ATP file* and *run ATP* perform this sub-process initially.

IMPORTANT! All nodes will automatically receive names from ATPDraw, so the user should normally only give names to nodes of special interest, e.g. involved in output requests and displayed in the Output Manager.

4.2.5.5 Output manager

The Output Manager list “all” requested outputs in the data case in the order that they appear in the pl4 file. The sorting option of the components is taken into account. The Output Manager even goes into User Specified, Additional data cards and Windsyn components to find outputs requested there. There is a limit of 32 output requests per component (voltage&current counts as one). The sequence of the output is:

-Branch voltages and power
-Switch voltages and power
-Node voltages
-Switch currents and energy
-Branch currents and energy
-SM
-TACS
-MODELS

-UM

When launching the Output Manager it compiles the circuit to generate the node names and presents a list of the outputs as shown in Fig. 4.20. The Windows Manager is a stay-on-top window that lets the user go back to edit the circuit. Two additional features are available; *Find* and *Edit*. Both are linked to the current selected row in the grid. The *Find* button finds the involved component and displays it in the middle of the screen in a lime color. If necessary it goes down into groups to display internal components. The *Edit* button brings up the involved component's input dialog where the user is allowed to edit the settings. However, the user has to leave the Output Manager and reopen it to actually refresh its content.

When ATPDraw goes into User Specified components it lists the node names found in the expected columns. This could however be an argument in the $Include call, and this in not handled by ATPDraw.

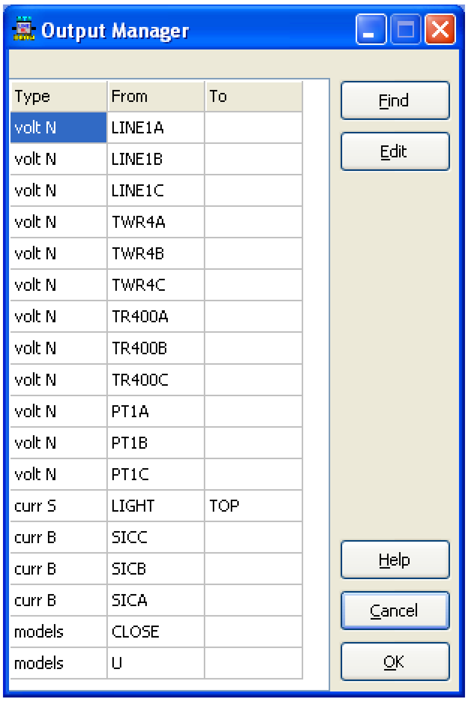


Fig. 4.20- Output Manager from Exa_9.acp

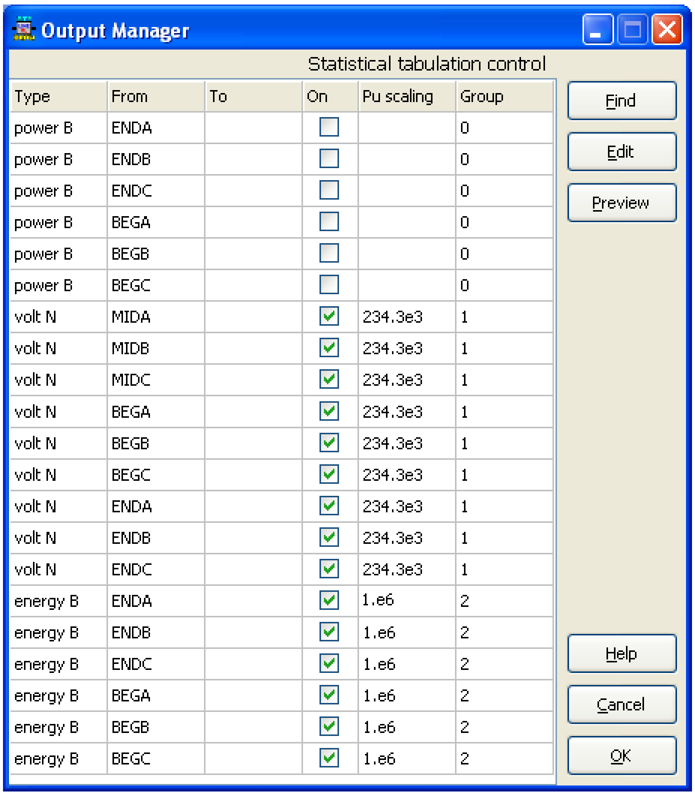


Fig. 4.21 - Output Manager from an extension of Exa_12.acp as shown in Fig. 4.22.

In the case of a statistical study (chosen under *ATP|Settings/Switch*) the Output Manager lists three additional columns as shown in Fig. 4.21. In the fourth columns in Fig. 4.21 the user can turn available output requests on and off for statistical tabulation. Only node voltages are on as default. In the sixth column the user can assign a group number to the statistical output request and in the fifth column assign a scaling factor to this group. There is also a *Preview* button available in this mode that lets the user examine how the final statistical tabulation will look like. This text will appear under /STATISTICS in the final ATP file.

```
/STATISTICS
       234300.MIDA   MIDB   MIDC   BEGA   BEGB   BEGC   ENDA   ENDB   ENDC
-4        1.E6ENDA          ENDB          ENDC          BEGA          BEGB         CONT.
-4        1.E6BEGC
```

There is one challenge related to SATURABLE TRANSFORMERS and the request of magnetizing branch outputs. This would require a very complicated identification of the transformer that is not handled in ATPDraw. The magnetization output is presented in the Output Manager (using an alias node name) but it is not possible to add this to a statistical tabulation.

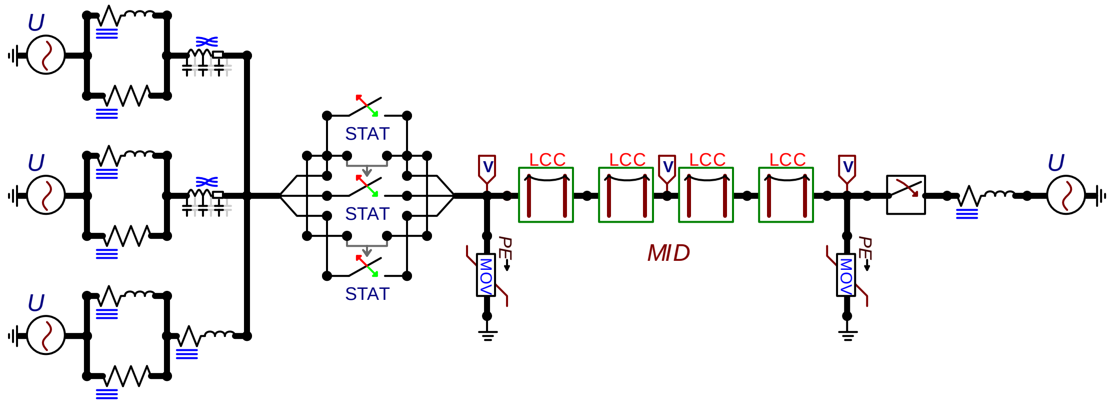


Fig. 4.22 - Exa_12.acp requesting additional output (both side node voltages and arrester powers and energies).

4.2.5.6 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.

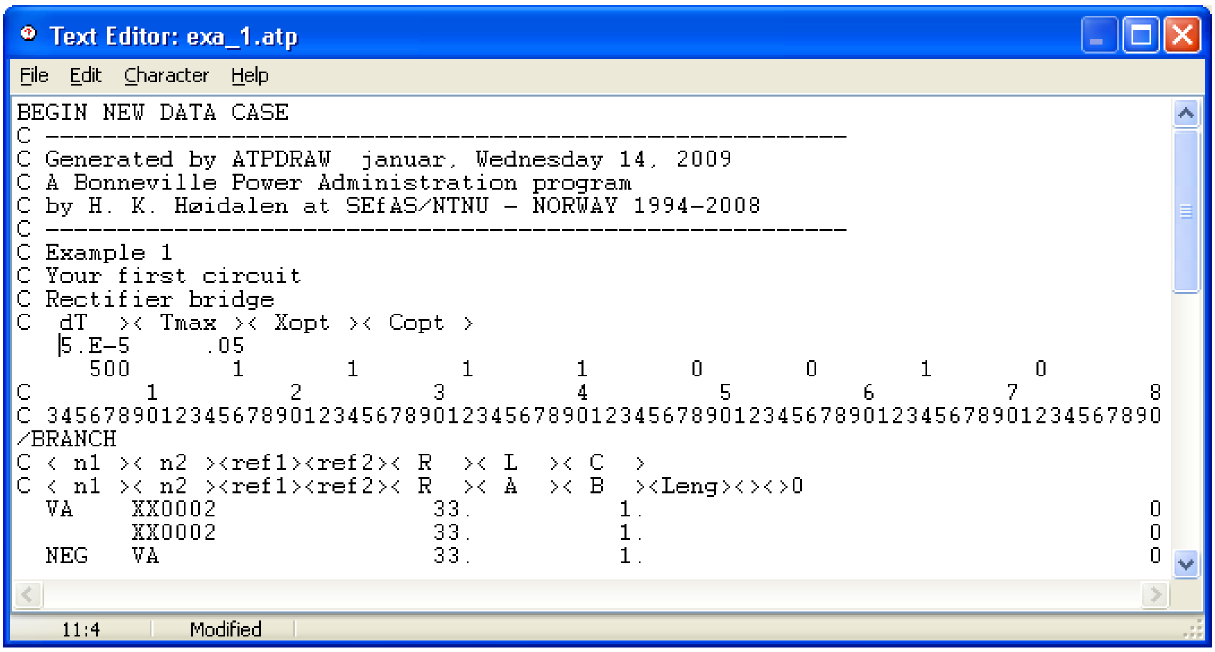


Fig. 4.23 - The main window of the built in text editor.

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 2 GB in size. 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. The right-click context menu offers 50 different request card templates via the *Insert* field..

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.

4.2.5.7 View 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 Result Directory (default 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 (illegal file name, I/O-xx bad character in input field, etc.) the LIS-file does not exist and can not be displayed either.

4.2.5.8 Find node and Find next node

The *Find node* helps the user to find a node with a specific name in the circuit. You type in the node name in the Find node dialog. For multi-phase node you only type in the root name without phase extensions 'A'..'Z'. *Find next node* is used to proceed to the next node with the same name. *Find node* goes into groups as well, and (multiple) *Edit|Edit circuit* (Ctrl+H) may be necessary to nagigate back into the main circuit.

4.2.5.9 Optimization

To use the optimization module there must be variables declared in the circuit and a cost function object must have been added to the circuit (*MODELS|WriteMaxMin*). The optimization module will change chosen circuit variables to optimize the cost function basen on either a Gradient Method, a Genetic Algorithm, or a Simplex Annealing method. This is further documented in the Advanced Manual, chapter 5.11.

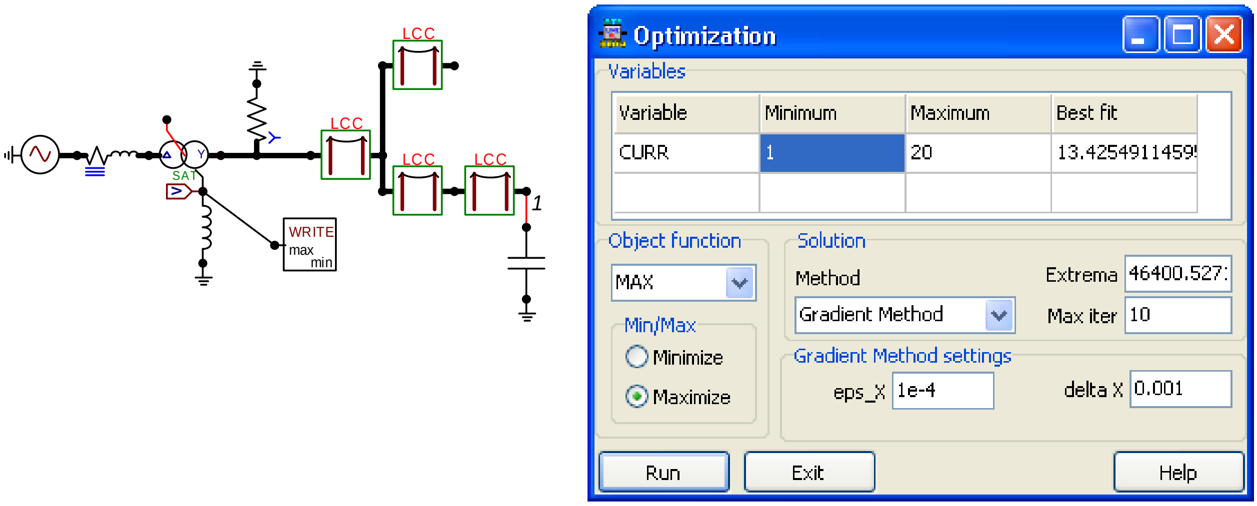


Fig. 4.24 – Finding the neutral grounding coil value giving resonance, Exa_18.acp.

4.2.5.10 Line Check

First, the user selects the line he wants to test and then clicks on *ATP|LineCheck* as shown in Fig. 4.25 . Then the input/output selection dialog box shown in Fig. 4.26 appears.

The LineCheck feature in ATPDraw supports up to 3 circuits. ATPDraw suggests the default quantities. The leftmost nodes in the circuit are suggested as the input nodes, while the rightmost nodes become the output. The circuit number follows the node order of the objects. For all standard ATPDraw components the upper nodes has the lowest circuit number. The user also has to specify the power frequency of the line/cable test. Finally, the user can check the *Exact phasor equivalent* button which will result in a slightly better results for long line sections.

When the user clicks on OK in Fig. 4.26 an ATP-file (`/LCC/LineCheck.dat`) is created and ATP executed. For a 3-phase configuration 4 sequential data cases are created (Z+, Y+, Z0, Y0) while for a 9-phase configuration 24 cases are created (Z11+, Y11+, Z110, Y110, Z12…, Z22…, Z13…, Z23…, Z33…), since symmetry is assumed. Finally the entire LIS-file is scanned. The calculated values are then presented in the result window shown in Fig. 4.25 . The user can switch between polar and complex coordinates and create a text-file of the result. The mutual data are presented on a separate page. The unit of the admittances is given in Farads or Siemens (micro or nano) and the user can scale all values by a factor or by the length.

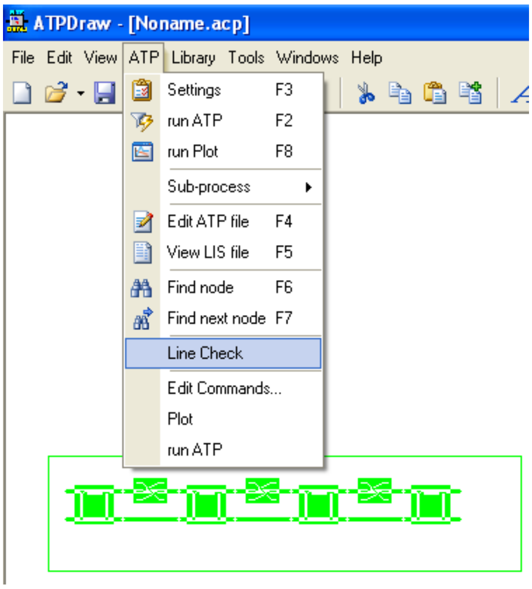


Fig. 4.25 – Selecting a line.

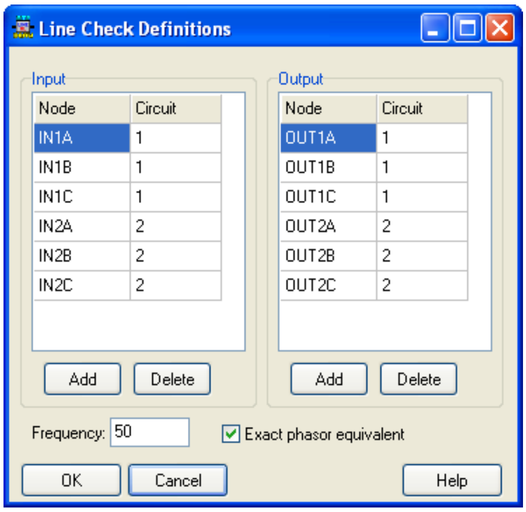


Fig. 4.26 – Selecting the inputs and outputs.

The series impedances are obtained by applying 1 A currents on the terminals while the output ends are grounded (the other circuits are left open and unenegized). For mutual coupling, 1 A is applied at both circuits. On the other hand the shunt admittances are obtained by applying a voltage source of 1 V at one terminal leaving the output end open. For mutual coupling, 1V is applied at one circuit while a voltage of 1E-20 is applied at the other.

Special attention must be paid to long lines and cables. This applies in particular to PI-equivalents. Usage of 'Exact phasor equivalent' is recommended, but is no guarantee of success. No attempt is made in ATPDraw to obtain a better approximation since the line/cable system to be tested in general is unknown. The mutual coupling in the positive sequence system is in symmetrical cases very small and vulnerable to the approximations made. Appendix 7.2 documents the calculation procedure.

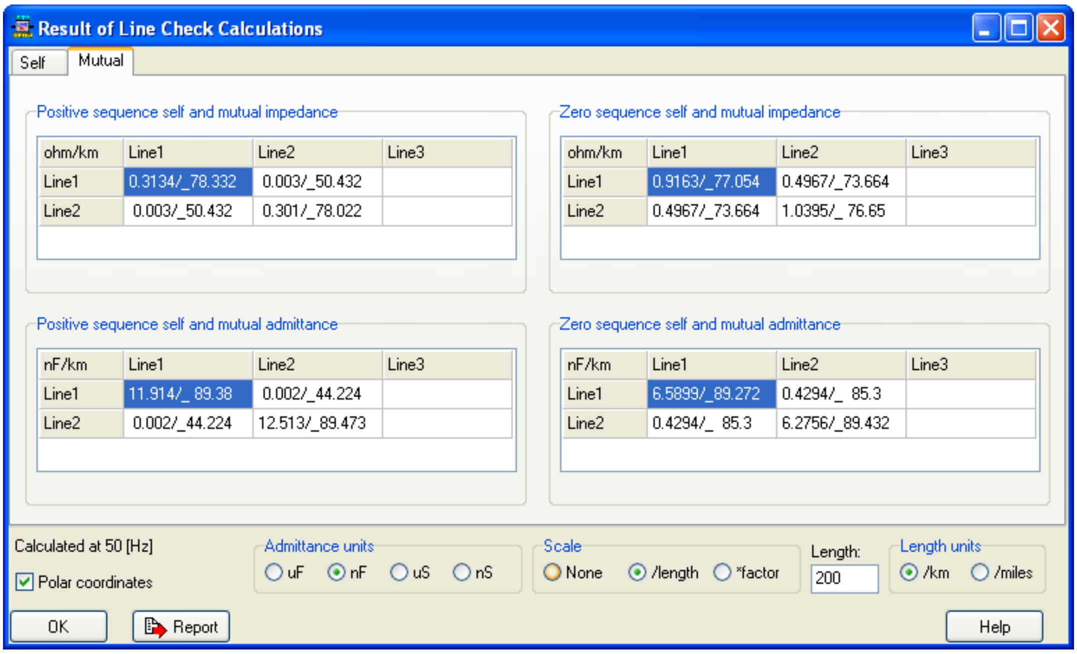
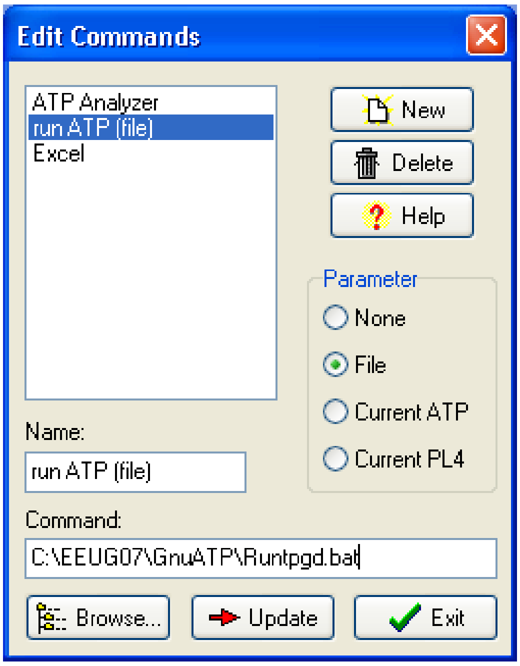


Fig. 4.27 - Presentation of the results.

4.2.5.11 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. 4.28, 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 command.

Fig. 4.28 - 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. 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.

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.

4.2.6 Library

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

1. Standard,
2. User specified, or
3. Model

Each component 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. Standard components has their support files stored in ATPDraw.scl (standard component library). When a component is added to the circuit this component inherit the properties from its support file and the support file is not used anymore. Except for the help text of standard components. In order to define and use User Specified components a support file `.sup` is required. Models can optionally be managed without a support file since a default support "file" can be automatically created based on the Models text header.

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

4.2.6.1 New object

Under this menu the user can create new User Specified and Models Components.

4.2.6.1.1 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 *Library|New Object|User Specified* 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 Definitions* 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 64, and the number of nodes in the range of 0 to 32. Control parameters for the object data can be entered on the *Nodes* and *Data* pages of Fig. 4.29.

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.`

`Units` Maximum 12 character text string with the unit that appear in the Component dialog box. The units COPT and XOPT are defined keywords responding to the users choice of COPT/XOPT under ATP|Settings/Simulation.

`Min/Max` Minimum/Maximum value allowed. Set equal to cancel range checking.

`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.

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).

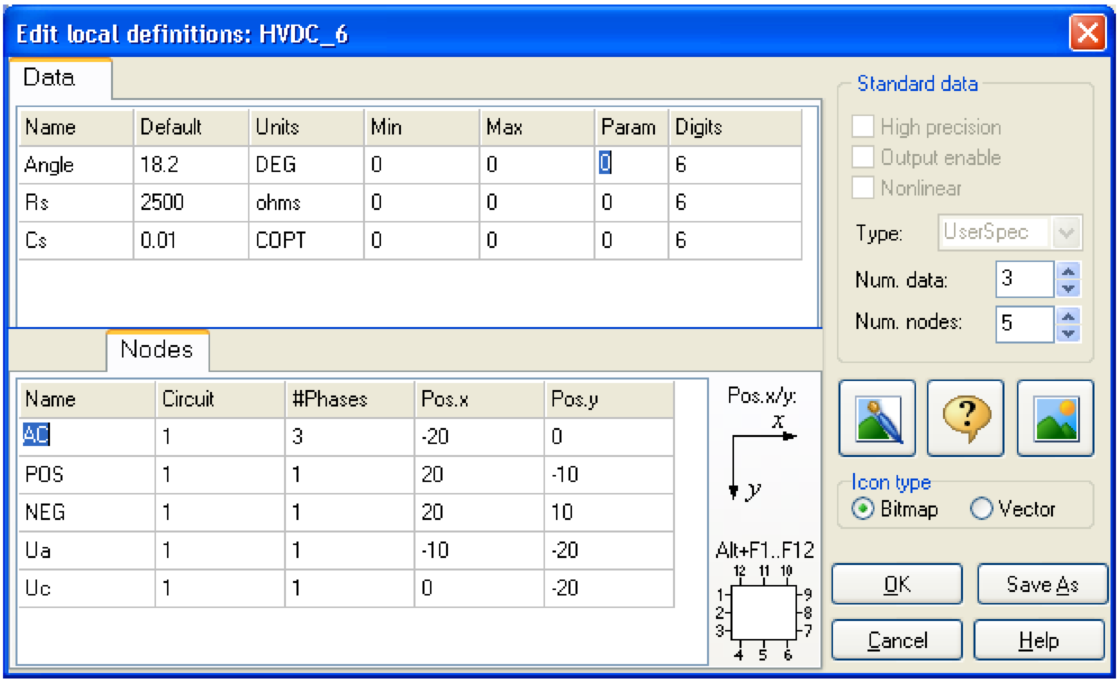


Fig. 4.29 - Control page of a new user specified object.

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

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

Circuit: 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 *Circuit* parameter has a different meaning for MODELS or TACS component nodes. It is used to specify the type of input/output.
MODELS node values:
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)

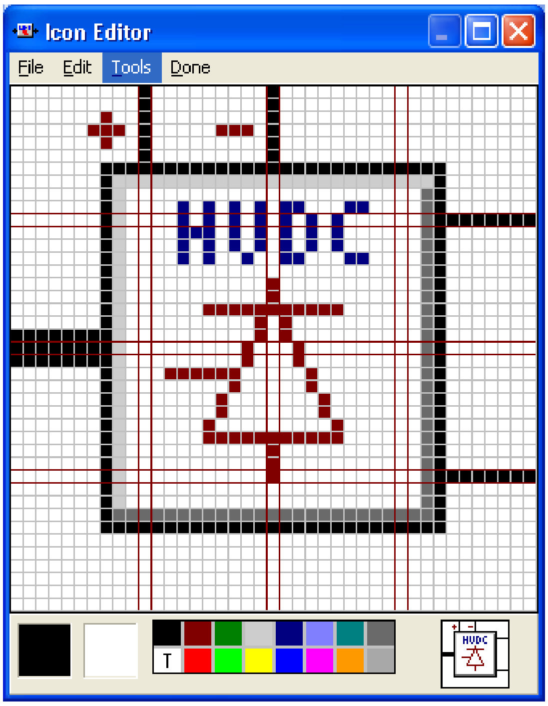
```
8:     Output from other model. Note that the model, which produces
       this output, must be USEd before the current model. This can
       be done by specifying a lower Order number for the model and
       then select the Sorting by Order number option under
       ATP|Settings/Misc.
9:     Global ATP variable input.
```

TACS node values:
0: Output node.
1: Positive sum input node.
2: Negative sum input node.
3: Disconnected input node.

#Phases: Number of phases (1..26) for the component node. If *#Phases* is set to >1 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.

Pos: Specifies the relative node position in steps of 10 pixels (grid). The standard border positions shown in the picture to the left of Fig. 4.29 have short cut keys Alt+F1..Alt+F12. The position (x, y) can in general be in the range -120,-110,..-10,0,10,..,110,120. The x-axis is oriented to the right while the y-axis is oriented downwards. The node positions should correspond with icon drawing.

Each circuit object has an icon, which represents the object on the screen. This icon can be of bitmap type or vector graphic type as selected under *Icon type*. The conversion from Bitmap to Vector style is not possible so you should not unintentionally change the icon style. Vector graphic enables better zooming and graphic export, font handling and editing, but for simplicity reasons the Bitmap option is shown here. The leftmost of the three speed button on the right hand side of the Fig. 4.29 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 4.2.7.1 of this manual. The user can draw individual pixels and in additions line, rectangles, circles, and fills. Text must be manually put together by pixels. The Vecor graphic editor has far better text capabilities.

Fig. 4.30 - Icon Editor.

Each component has a pre-defined help file, which can be edited by a built in *Help Editor* accessible via the speed button on the middle speed button in the *Edit definitions* dialog in Fig. 4.29. 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 4.2.7.2 section of this manual.

With the rightmost speed button in Fig. 4.29 the user can add a background bitmap/metafile image of any size to the icon. This should only be used in special cases since it could heavily occupy memory and increase the project file dramatically. No down-sampling of the imported image is performed.

When the user has completed all modifications of the component data and of the icon and help, 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.

4.2.6.1.2 New Model sup-file

Usage of MODELS [4] in ATPDraw is described in the Advanced Manual. When the user change the Model header (input, output or data section) in a circuit in ATPDraw the component and its icon is automatically updated. So for the usual case of a dynamic Model there is no point in pre-defining support and model files. These files can anyhow be exported from a finished Model. If you want a static Model, however, you can specify a support file under this menu item. To use this feature, you first must write a model file using the built in *Model Editor* as shown in section 4.2.6.1.3. 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 4.11.9 and 5.5.1). 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 Definitions* dialog as shown in Fig. 4.31.

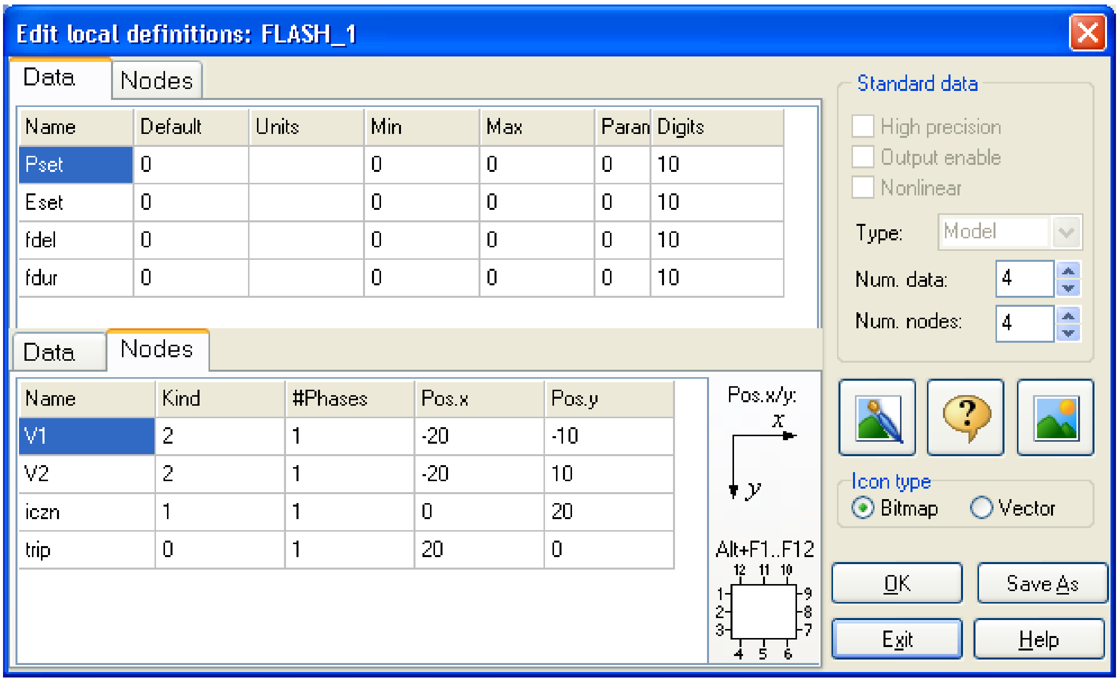


Fig. 4.31 - Control page for a New Model sup-file.

Name: Identifies the node in the *Node* and *Component* dialog boxes. 12 chararacters maximum. Must be equal to the name used in the Model header.

Kind: Specifies the input/output type of the node.

#Phases: Number of phases can be from 1 to 26 and must be defined as V1[1..n].

Pos: Specifies the relative node position in steps of 10 pixels (grid). The standard border positions shown in the picture to the left of Fig. 4.29 have short cut keys Alt+F1..Alt+F12. The position (x, y) can in

```
general be in the range -120,-110,..-10,0,10,..,110,120. The x-axis is
oriented to the right while the y-axis is oriented downwards. on the
icon border. The node positions should correspond with icon drawing.
```

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 Order number for the model and then select the
     Sorting by Order number option under ATP|Settings/Format.
9:   ATP global variable. MNT is for instance the simulation number and
     the Pocket Caculator KNT equivalent.
```

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 32 and the maximum number of data that can be passed into a Model is 64.

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.

4.2.6.1.3 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 *Library* | New object| *Model 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 specify expressions, HISTORY of DELAY CELLS under USE
Not possible to call other models under USE.

4.2.6.2 Edit object

Under this menu item the user can edit existing support files for Standard, User Specified and Models components.

4.2.6.2.1 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. 4.32, where the support file to be edited can be selected, then a dialog box shown in Fig. 4.33 appears.

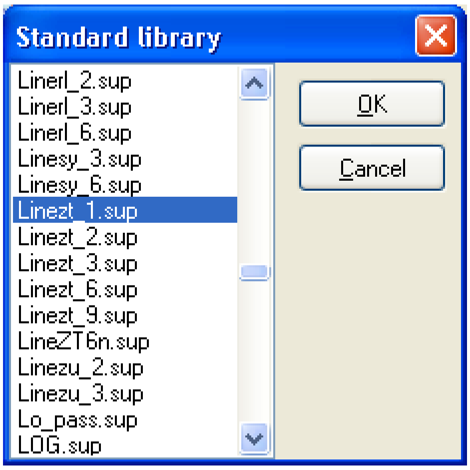


Fig. 4.32 - Specify the support file of the standard component to be edited.

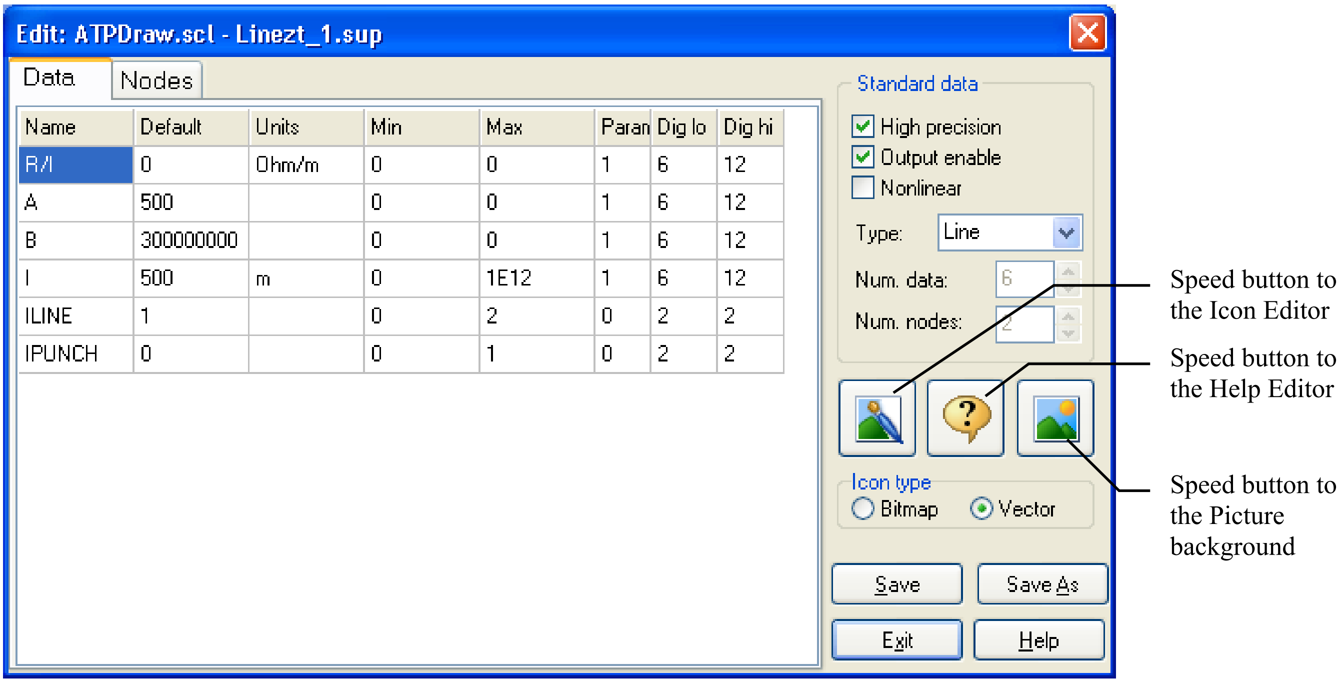


Fig. 4.33 - Control page of object data.

4.2.6.2.2 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 4.2.6.2.1.

4.2.6.2.3 Edit Model sup-file

A model object can be edited like any other circuit object. If the user clicks on the *Library | Edit object | Model sup-file*, the *Edit Edfinitions* 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.

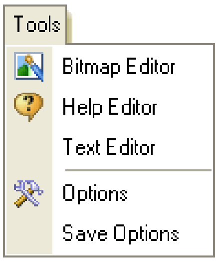
4.2.6.2.4 Edit Model mod-file

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*.

4.2.6.3 Syncronize|Reload Icons

Reads and displays standard 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. User Specified and Models components icons are not updated.

4.2.7 Tools

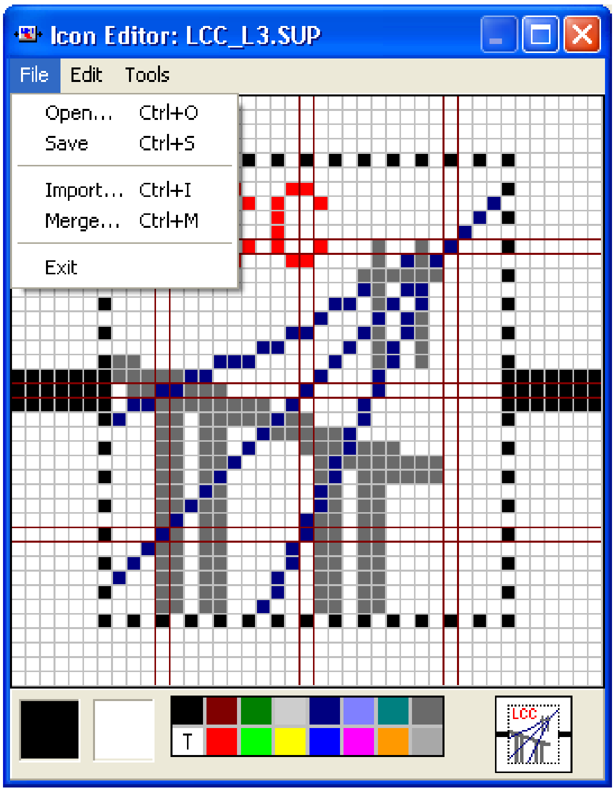


Items under the *Tools* menu enable you to edit component icons or help text, view or edit text files, customize several program options and save them to the `ATPDraw.ini` file.
Fig. 4.34 shows the available commands of the *Tools* menu.

Fig. 4.34 - Tools menu.

4.2.7.1 Icon Editor

Brings up an icon editor shown in Fig. 4.35 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 *Library* 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. 4.30.

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

Fig. 4.35 - 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.
Merge	Request an external support file and adds its icon to the current icon.
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.

4.2.7.2 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.

4.2.7.3 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.

4.2.7.4 Options

In the *Tools* | *Options* menu several user customizable program options for a particular ATPDraw session can be set and saved 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.ini` file is stored under `%APPDATA%/atpdraw` (typically c:\documents and settings\user\program data\atpdraw) and is unique for each user of the computer. The file is ATPDraw version independent.

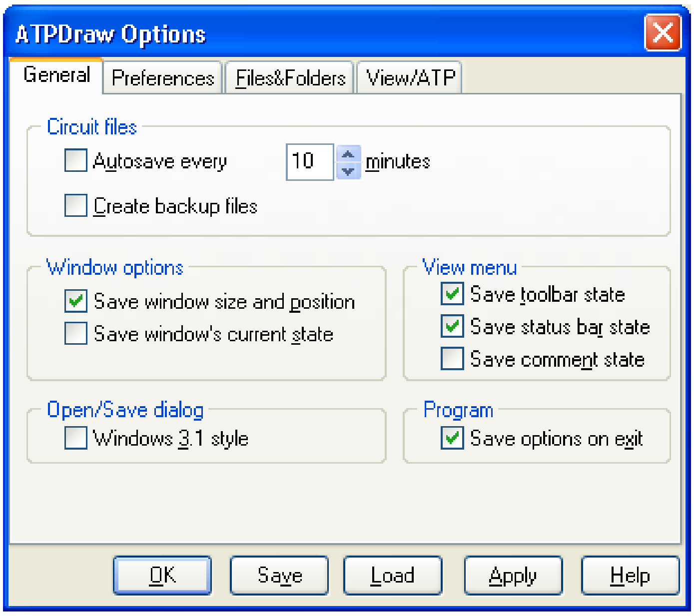


Fig. 4.36 - Customizing program options.

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. 4.36 this dialog box has four sub-pages: *General, Preferences, Directories* and *View/ATP*.

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.

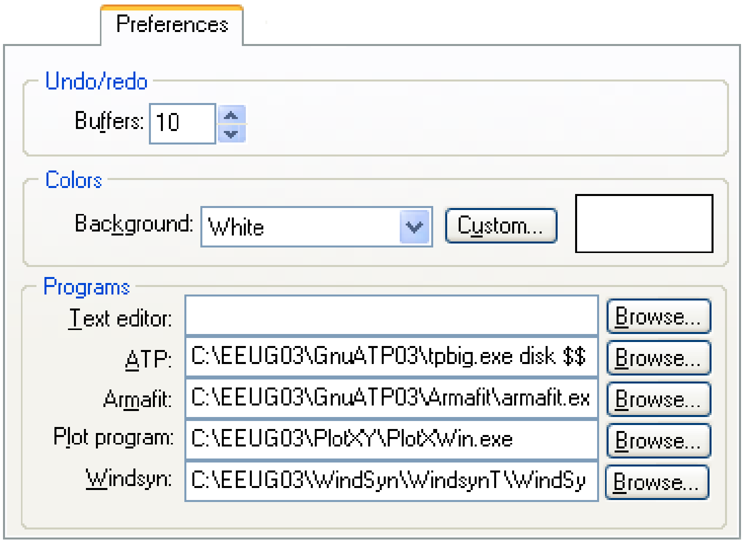
Note that the 'save state' options will have no effect unless program options are saved to the initialization file (`ATPDraw.ini`) by the *Save* command at the bottom of the *ATPDraw Options* dialog, or by selecting the 'Save options on exit' check box, or by the *Tools | Save Options* menu.

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. 4.37 - 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.

Plot:	Holds the preferred plotting command. Executed under ATP|run Plot (F8).
Windsyn:	Holds the compatible Windsyn command. WindsynATPDraw.exe.

Files&Folders

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

Option	Description
Project folder	The directory where ATPDraw stores the project files (.acp).
ATP folder	Specifies the directory in which .atp files are created. This is also the default Result Directory.
Model folder	Directory containing support (.sup) and model (.mod) files for MODELS components.
Help folder	The user can write help text files for instance resistor.txt (same name as the support file and extension txt) and store it in this folder. It will then automatically be added after the standard help text.
User spec. folder	Directory containing support (.sup), library (.lib) 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.
Transformer 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. In addition the Hybrid transformer (XFMR) files could be stored here (.xfm).
Plugins folder	This is a user definedable folder that appears in the bottom of the Selection menu. The user can add project files (acp) and sub-folders to this folder structure.

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 4.2.4.4. 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 4.2.4.4).

The *Edit settings* button calls the *ATP settings* dialog described in section 4.2.5.1 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.

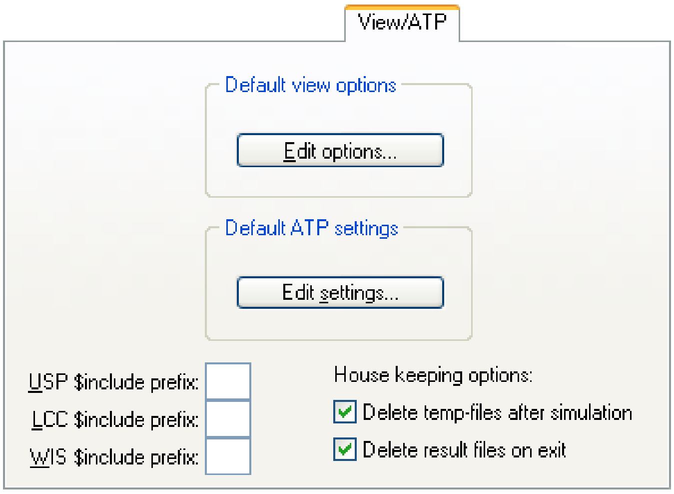


Fig. 4.38 - Setting default view and ATP options.

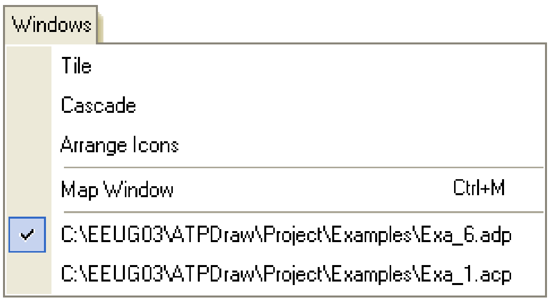
The prefix tags are text strings added in front of the $include file name. This is because User Specified (USP), Line&Cable (LCC), and Windsyn (WIS) components all have their $include file dumped to the Result Directory (same as the ATP-file). In the case duplicate file names in these cathegories, file conflicts will occure. The prefix option can then be used to avoid the conflict. If two UPS component have the same name for instance, the $include file is anyhow forced to be equal.

The House-keeping options delete temporary files after the simulation or exit. In the case of debugging a Line&Cable model the *Delete temp-files after simulation* option should not be checked.

4.2.7.5 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.

4.2.8 Window



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

Fig. 4.39 - 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.

4.2.8.1 Map Window

The *Map Window* command (Shortcut: *Ctrl+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 Ctrl+*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. 4.40 - Map window.

When you press and hold down the left mouse button in the map rectangle, you move the display of the circuit world continuously. If any circuit objects are currently selected when you reposition the map rectangle, they remain in the same position in the circuit window. This functionality can be used to quickly move a collection of objects a relatively large distance.

4.2.9 Help

The *Help menu* contains options for displaying the help of ATPDraw, and the copyright and version information. The help file `ATPDraw.chm` is distributed with ATPDraw and it follows the compressed HTML standard compatible with Windows Vista.

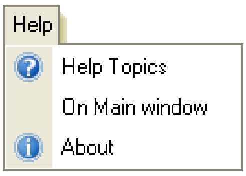


Fig. 4.41 - Help menu.

ATPDraw's HTML help is displayed in a standard Windows dialog, which provides indexed and searchable help on all ATPDraw dialogs and options.

4.2.9.1 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. 4.42.

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. The ATPDraw help file consists of 136 topics.

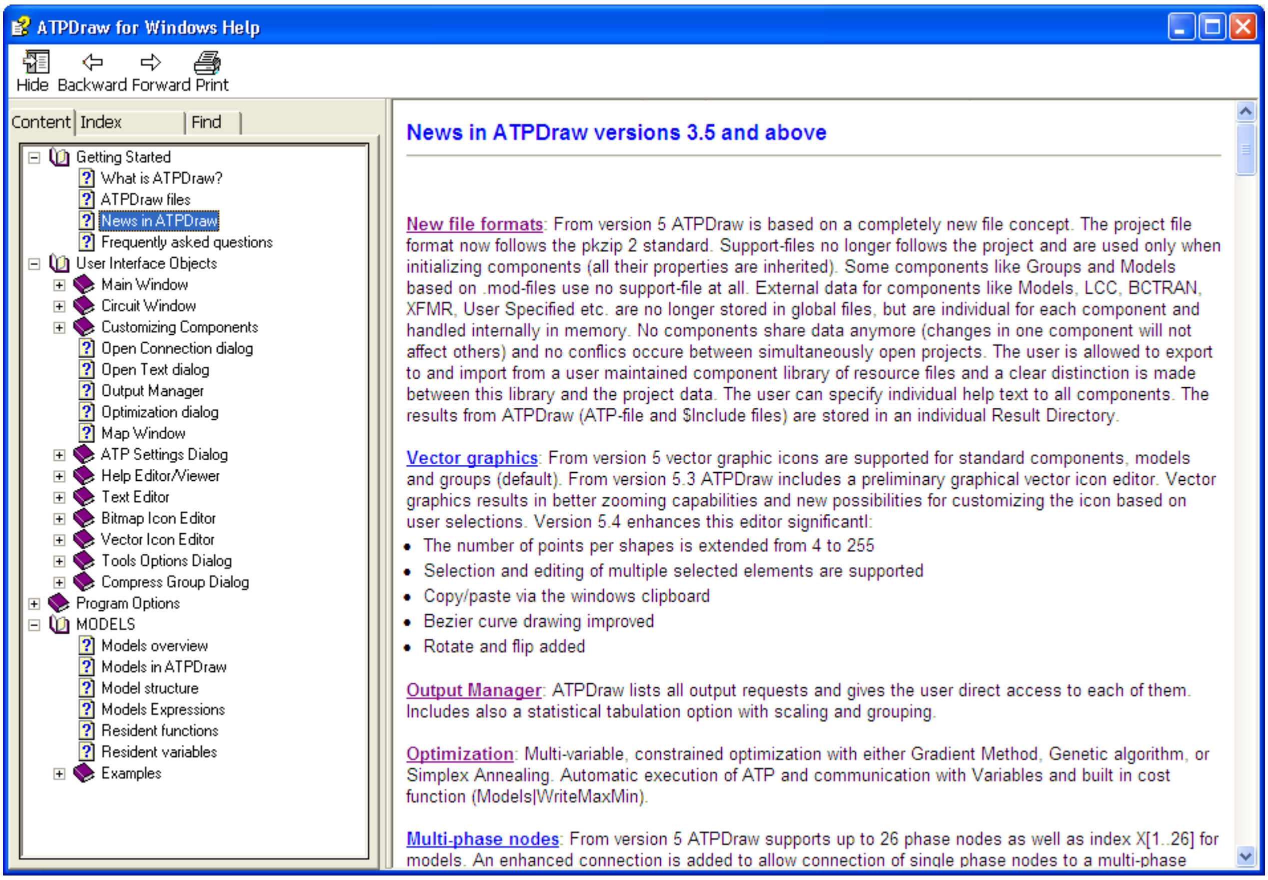


Fig. 4.42 - HTML help of ATPDraw.

4.2.9.2 On Main Window

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

4.2.9.3 About ATPDraw

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

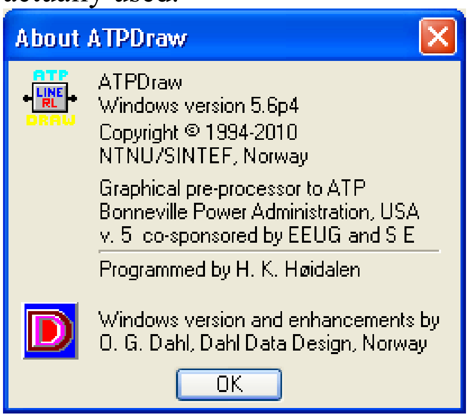
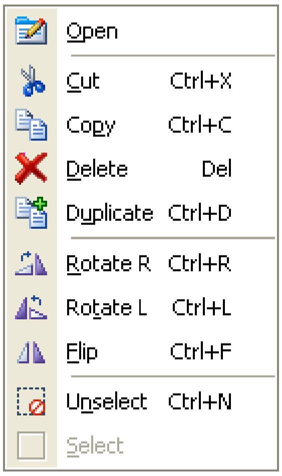


Fig. 4.43 - About window of ATPDraw.

4.3 Shortcut menu

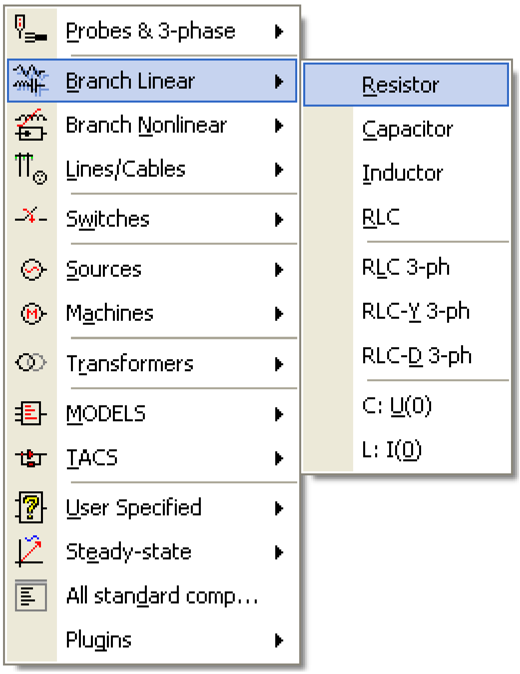
The *Shortcut menu* provides access to the most frequently used object manipulation functions. To show and activate the shortcut menu, hold down the *Shift* key while you click the right mouse button on an object or a selected group of objects in the circuit window. Most of the items on this menu are 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)

Fig. 4.44 - 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. 4.45. After selecting a component in one of the floating menus, the selected object is drawn in the circuit window.

Fig. 4.45 - Component selection menu.

The upper section of the menu provide access to the probe, splitter and transposition and reference objects, the next seven to many standard ATP components: linear and nonlinear elements, lines and cables, switches, sources, electrical machines and transformers. The next section is dedicated for the control systems MODELS and TACS components. User specified objects and Frequency dependent components for Harmonic Frequency Scan (HFS) studies are accessible in the next group followed by a list of all the standard supported components (for instance older component replaced by new versions). The final menu item called Plugins points to a user defined folder structure for import of project files (sub-circuits).

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. 4.46 appears. These dialog boxes have the same layout for all circuit objects except probes, which can be edited from the *Probe* dialog box.

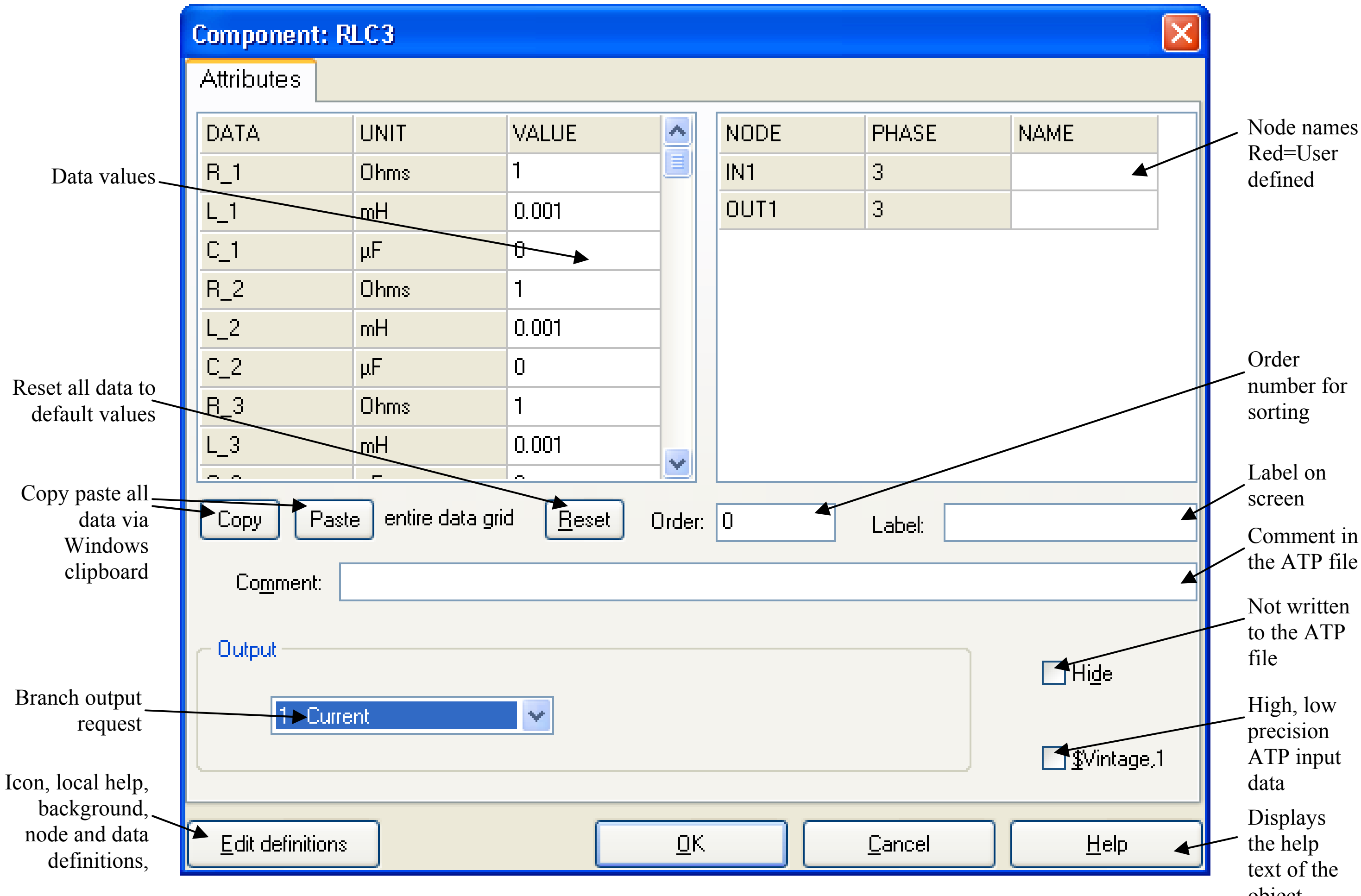


Fig. 4.46 - The Component dialog box.

Component data can be entered in the *Value* field of the *Attributes* page. The *Node*, *Phase* and *Name* fields are initially empty and you can enter node names in the Name field (without phase extensions 'A'..'Z'). You have to run *ATP|Sub-process|Make node names* or (*ATP|run ATP*) to obtain the ATPDraw specified node names.

Numerical values in the data input fields can be specified as real or integer, with an optional exponential integer, identified by 'E' or 'e'. A period '.' is used as decimal point. 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 legal range can be set under *Edit definitions*. Instead of a value you can also assign a 6 (or less) character text string as input data for most of the standard components. This requires the *Param* property of the data to be set to unity (see *Edit definitions*). Numerical values can later be assigned to these variables under

ATP|Settings/Variables using the `$PARAMETER` feature of ATP-EMTP (see in 4.2.5.1).

Just below the node input column, there is a *Order* input field. It can be used later as optional sorting criteria (the lowest order 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. The visibility of the component label is controlled by the *Labels* option in the *View | Options* dialog box. The label is movable and directly editable on the screen. The font of the Label is controlled vie *View|Set circuit font*. 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 *Order*, *Label* and *Comment* fields, the *Hide* button is common to all components. Hidden components are not included in the ATP-file and are displayed as light gray icons in the circuit window. 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 components User specified, Models and Groups has also a *Protect* button for password protection of the their content.

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 HTML help system of ATPDraw if you press the *F1* key.

The non-linear components (non-linear branches, saturable transformers, TSWITCH, and TACS Device 56) have a *Characteristic* page too, as shown in Fig. 4.47.

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 64). 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.

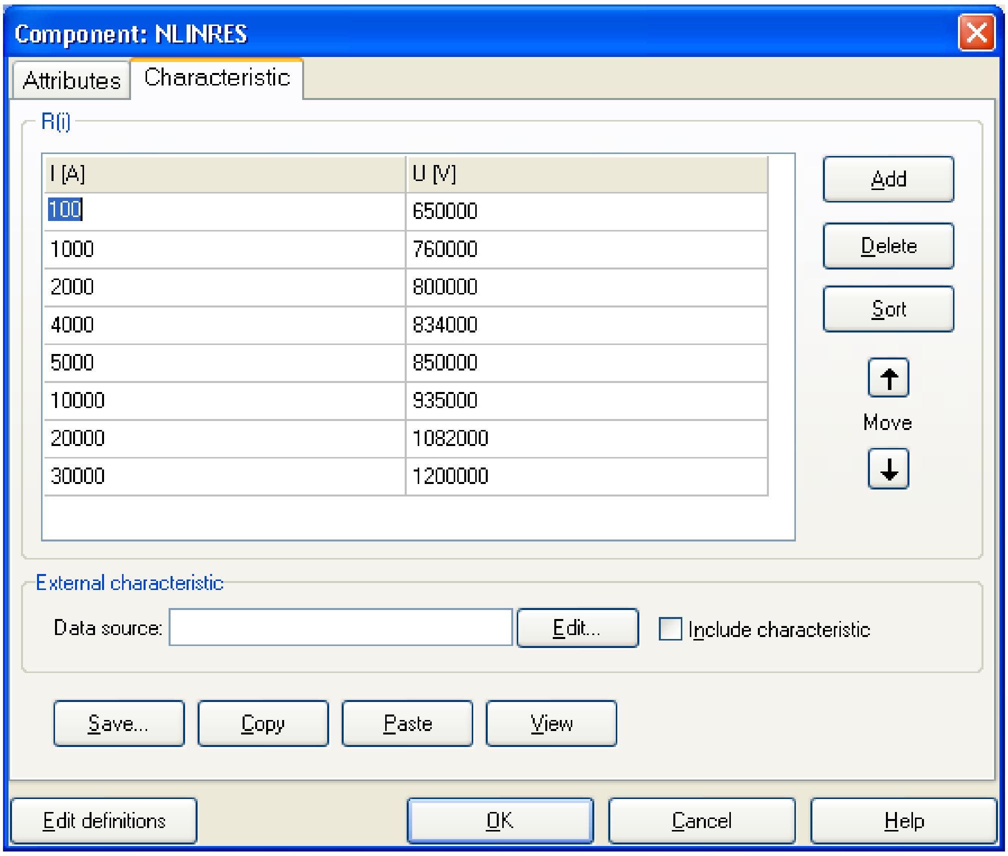


Fig. 4.47 - The Characteristic page of non-linear components.

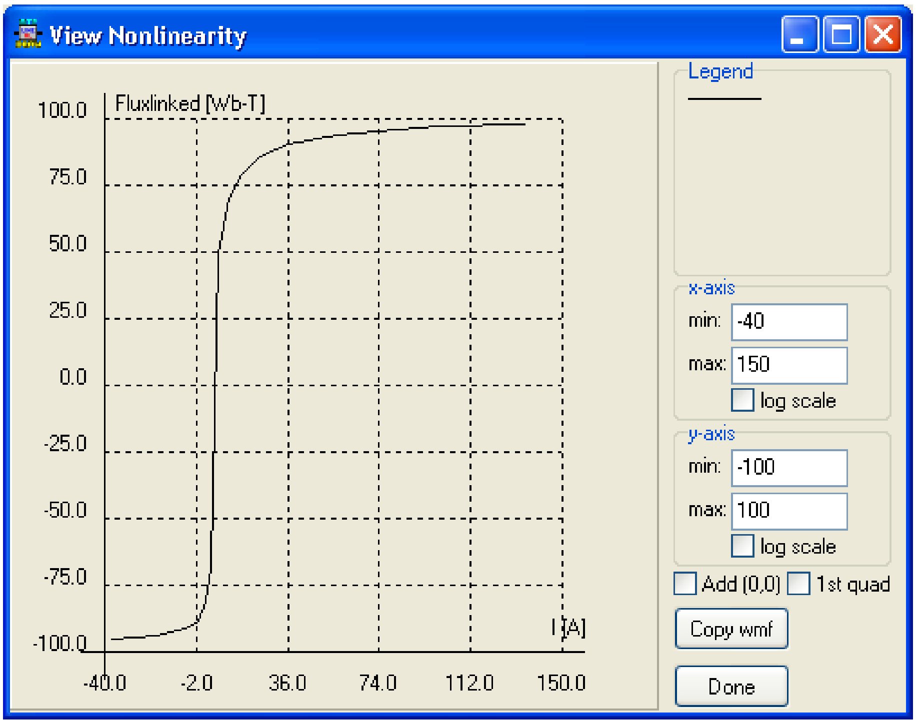


Fig. 4.48 - The View nonlinearity window.

The *External characteristic* section at the bottom of the page contains an *Data source* 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. ATPDraw reads the specified file into memory and inserts it directly in the final ATP file.

The nonlinear characteristic specified by the user can be displayed by clicking on the *View* button. In the *View Nonlinearity* window (Fig. 4.48) the min and max axis values are user selectable as well as the use of logarithmic scale (if min>0). It is possible to left click and drag a rectangle for zooming. Click right to restore. The Add (0,0) check box will add the origo point, and *1st quad* will display only the first quadrant. It is also possible to copy the graphic to the Windows clipboard in a metafile format with *Copy wmf*. Selecting *Done* will close the nonlinearity display.

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

General 3-phase transformer (SATTRAFO)
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)
Windsyn UM component (WISIND, WISSYN)
Hybrid Transformer (XFMR)
Models&Type 94

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

a) For Models you can enter the editor for inspecting or changing the Models text. In addition you can specify a *Use As* string and defined the output of internal variables RECORD.
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 Connection dialog box

The Connection dialog box appears if you draw a Connection between a single phase node and a multi-phase node or double click on a Connection. This dialog allows you to select the number of phases in the Connection and the phase number of a single phase Connection (*Phase index*). A pure single phase connection between two single phase nodes should have the *Phase index* 0-@. You can also select the Color of the Connection and a text *Label* which can be displayed on screen. In addition you can choose to *Hide* the connection and transform it to a *Relation* (not a connection, only a dashed line). In both these cases the connection do not affect the node names. A special option is to force the *Node dots* on regardless of the Node dot size set in the main menu.

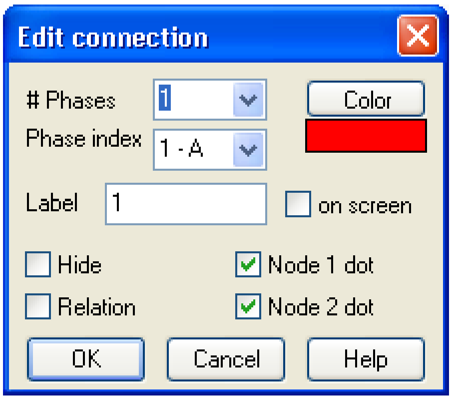
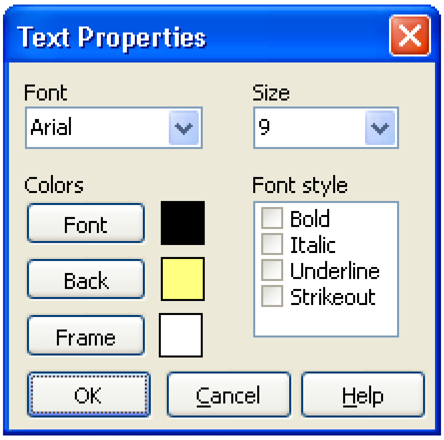


Fig. 4.49 – Connection dialog box

4.7 Text dialog box.



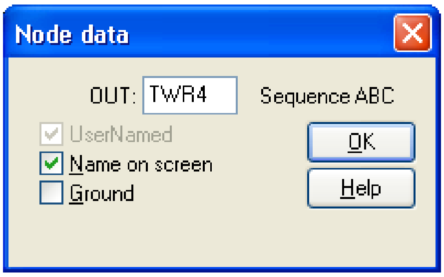
The *Text dialog* box appears if you right click or double click on a *Circuit Text* (not a *Label* or *Node Name*). In this dialog you can specify the *Font*, *Size* and *Colors* of the font used in the *Circuit Text*. You can edit *Circuit Text*, *Label* and *Node Names* directly in the Circuit Window by a left, simple click on them. *Circuit Texts* can hold multiple lines and the entire text uses the same font. You can move the *Circuit Texts*, *Labels*, and *Node Names* by left click and hold. Press the *Alt* key to avoid selecting other circuit objects.

Fig. 4.50 – Circuit Text dialog box

4.8 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 black, while those whose names were specified by the user are drawn with red 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. 4.51 - 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:

Standard and USP components:

Ground If checked, the node is grounded. A ground symbol appears for rotation of the graphical grounding symbol.

MODELS node:

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 Order number for the model and then select the *Sorting by Order number* option under *ATP | Settings / Format*.
9=Global ATP variable.

TACS node:

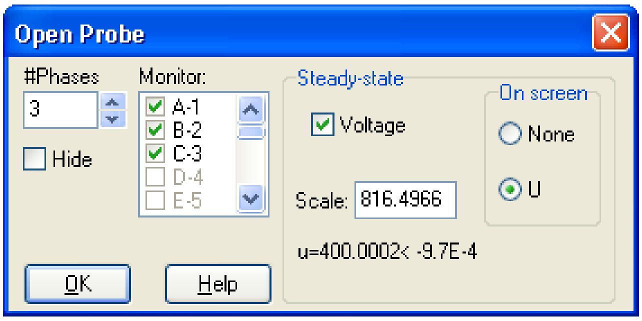
Type
0=Output.
1=Input signal positive sum up.
2=Input signal negative sum up.
3=Input signal disconnected. (necessary only if the node name is user specified)

TACS controlled machine node:

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.

4.9 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 five different probes in ATPDraw:



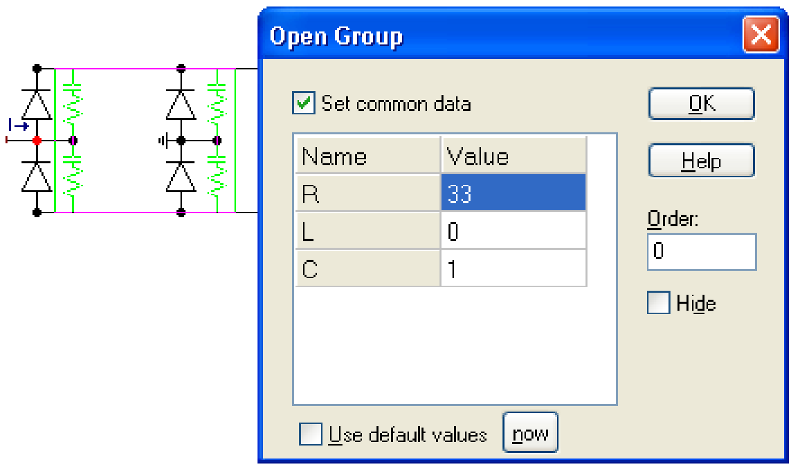
Probe_v: Node voltages output request.
Probe_b: Branch voltage output request.
Probe_i: Branch current output request.
Probe_t: TACS variable output request. Type33.
Probe_m: MODELS output nodes.

Fig. 4.52 - Node dialog box for standard components.

The Steady-state option is only available for Voltage and Current probes. ATPDraw reads the lis file and identifies the steady state ATP output. For multi-phase nodes only phase A is analyzed. The current probe also handles power and energy flow. ATPDraw divied the steady-state value with the *Scale* factor (816.4966-> line voltages in kV RMS value) befor displaying it on screen or immediately below. The symbol '<' is used to indicate the phasor angle. Remember that the number of phases is critical for a current probe and this has to match the circuit.

4.10 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, *Order* number and *Hide* state. The common data parameters are listed in a dialog as of Fig. 4.53 where you can change the data for all the involved components, simultaneously. The data names from the definition properties are used to classify the data.



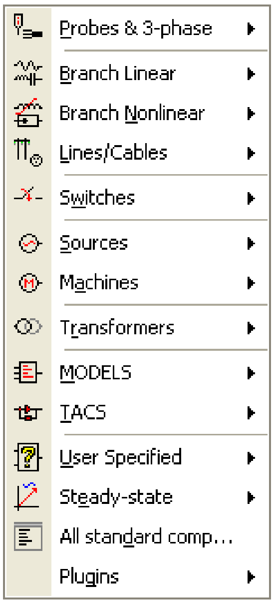
An alternative way to change the data parameter for several component simultaneously is to use $PARAMETER feature (see Fig. 4.17 in section 4.2.5.1).

Fig. 4.53 - Open Group dialog box for simultaneous data setting

Every component has a an order number. By specifying a value in the *Order* field, all components in the selected group of objects are assigned the same number. The order number serves as an optional sorting criterion for the ATP-file (components with the lowest order 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. You can also choose to reset to the default values inherited 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.

4.11 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 where you clicked the mouse button in 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. 4.54 - Component selection menu.

Probes & 3-phase

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

Branches

- Branch linear: 1-phase and 3-phase non-coupled components. RLC.
- 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)
- Windsyn component

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.
- Hybrid Transformer (XFMR). Advanced topologically correct transformer with Test Report, Design data or Typical value input.

MODELS

- Under MODELS the user can either select a default model and write/update the Model text internally, or select an existing external model component 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:
  - Names of all input, output and data variables must be less than 12 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: General, multi-phase type 94 component. Specify the type; THEV, ITER, NORT, NORT-TR and the number of phases. Specify a mod-file describing the Type-94 models component (templates available). The same rules as specified under MODELS apply.

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. First order dynamic icon with limits. Simple Integral, Derivative, first order Low and High Pass transfer functions.
- TACS devices. Type 50-66.
- 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.
- Additional: Free format user specified text for insert in the ATP file. Selection of location.
- Single and 3-phase reference: These objects are not represented in the ATP input data file and serve only as visualization of connectivity.
- Files: Select a support file (sup). Import a lib-file (Data Base Module format) via the Edit menu. $Include is used to include the user specified lib-file into the ATP input file and pass node names and data variables as parameters.

Steady-state components

- RLC Phasor component only present at steady state
- 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
- Load flow components PQ, UP, TQ

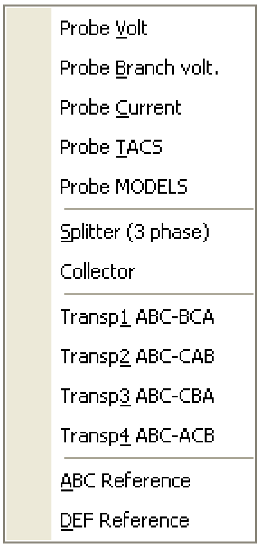
Standard Component..

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

Plugins

- User defined folder structure containing project files (.acp) for import.

4.11.1 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. 4.55 – 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. *Add current node*: Two switches in series. Middle node available.

Probe Tacs

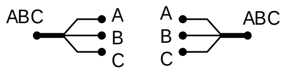
Selecting this field draws the Tacs probe to specify signal output and inserts TACS Type-33 object into the ATP-file.

Probe Model

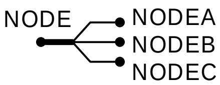
Selecting this field draws the Model probe which can be added to Models output nodes. Inserts RECORDS cards into ATP-file.

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.

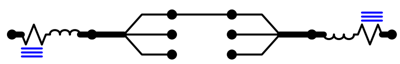


When a splitter is rotated the phase sequence of the single-phase side changes as shown left.

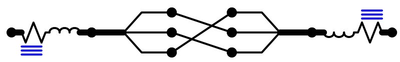


If a name is given to the 3-phase node, the letters *A B C* are added automatically on the single-phase side of splitters.

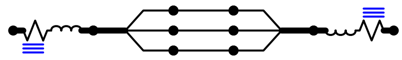
Note! Do not give names to nodes at the single-phase side of splitters and do not connect splitters together on the single-phase side (except all three phases). I.e. next two examples are illegal!



Disconnection is illegal this way!



Transposition is illegal this way!



This is leagal, however.

Collector

The *Collector* object is a component with a single multi-phase node. It is only used in compress added to, sinse only components can have external nodes, not connections.

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 4.2.6.2.1). 3-phase nodes having the same *Kind* will receive the same phase sequence.

ABC reference

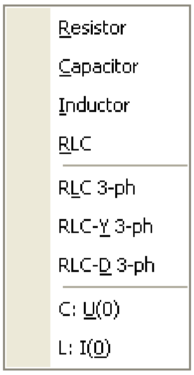
When attached to a 3-phase node in the circuit this node becomes the "master" node with phase sequence *ABC*. The other nodes will adapt this setting.

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.

4.11.2 Branch Linear

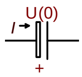
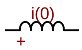
This sub-menu contains linear branch components. The name and the icon of linear branch objects, as well as a brief description of the components are given next in tabulated form. Data parameters and node names to all components can be specified in the *Component* dialog box (see Fig. 4.46), which appears if you click on the icon of the component with the right mouse button in the circuit window.

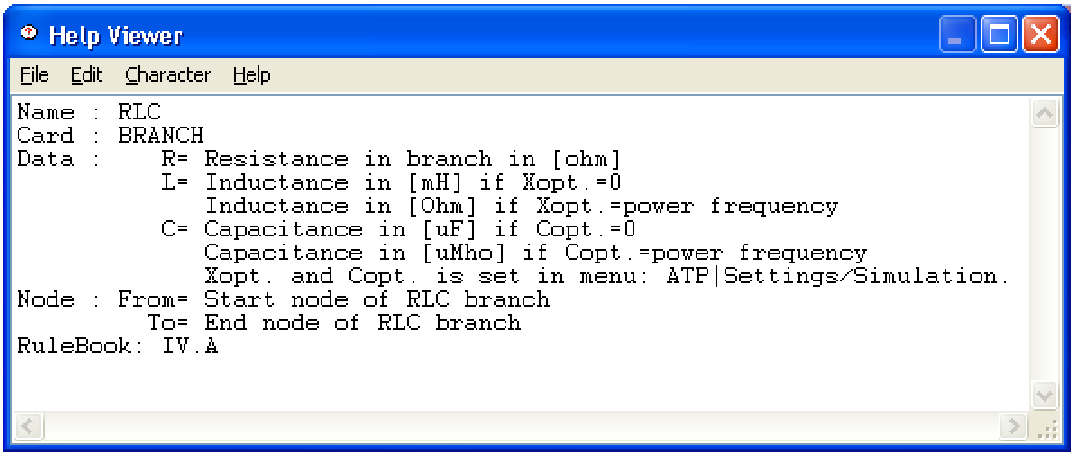


The *Help* button on the Component dialog boxes calls the *Help Viewer* in which a short description of parameters and a reference to the corresponding ATP Rule Book chapter is given. As an example, Fig. 4.57 shows the help information associated with the ordinary RLC branch.

Fig. 4.56 – 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. Dynamic icon.
RLC 3-ph	RLC3		BRANCH type 0	3-phase R, L and C in series. Independent values in phases. Dynamic icon.
RLC-Y 3-ph	RLCY3		BRANCH type 0	3-phase R, L and C, Y coupling. Independent values in phases. Dynamic icon.
RLC-D 3-ph	RLCD3		BRANCH type 0	3-phase R, L and C, D coupling. Independent values in phases. Dynamic icon.

C : U(0)	`CAP_U0`		BRANCH + initial condition	Capacitor with initial condition.
L : I(0)	`IND_I0`		BRANCH + initial condition	Inductor with initial condition.

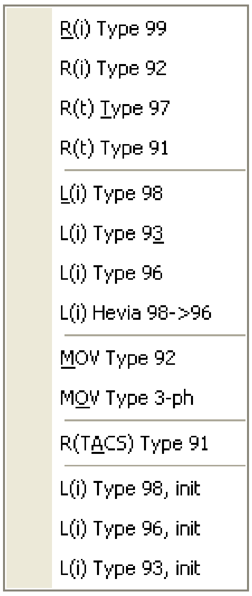


```
Help Viewer
File  Edit  Character  Help
Name  :  RLC
Card  :  BRANCH
Data  :     R= Resistance in branch in [ohm]
            L= Inductance in [mH] if Xopt.=0
               Inductance in [Ohm] if Xopt.=power frequency
            C= Capacitance in [uF] if Copt.=0
               Capacitance in [uMho] 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. 4.57 – Help information associated with the series RLC object.

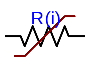
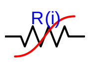
4.11.3 Branch Nonlinear

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. 4.47. 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. 4.58 – Nonlinear branch elements.

Selection	Object name	Icon	ATP card	Description
R(i) Type 99	`NLINRES`		BRANCH type 99	Current dependent resistance.
R(i) Type 92	`NLRES92`		BRANCH type 92	Current dependent resistance.

R(t) Type 97	NLINR_T		BRANCH type 97	Time dependent resistor.
R(t) Type 91	NLRES91		BRANCH type 91	Time dependent resistor.
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.
L(i) Hevia 98->96	HEVIA98		BRANCH type 98	Pseudo-nonlinear hysteretic inductor.
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.

4.11.4 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.

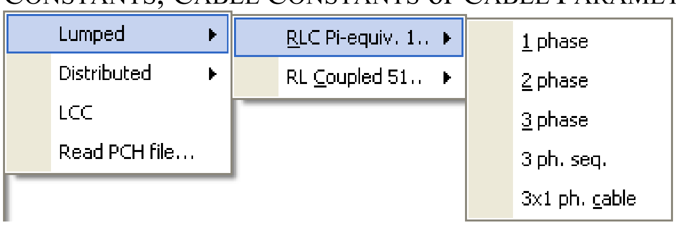


Fig. 4.59 – Line models with lumped parameters.

4.11.4.1 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 ph. Seq.*	LINEPI_3		BRANCH type 1-3	3-phase RLC π-equivalent Non-symmetric. 3-phase nodes.
RLC Pi-equiv. 1.. *+ 3 ph. Seq.*	LINEPI3S		BRANCH type 1-3	3-phase RLC π-equivalent Symmetrical. 3-phase nodes.
RLC Pi-equiv. 1.. *+ 3x1 ph. Cable*	PI_CAB3S		BRANCH type 1-3	3-phase RLC π-equivalent No mutual coupling
RL Coupled 51.. *+ 1 phase*	LINERL_1		BRANCH type 51	Single phase RL coupled line model.
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.. *+ 3 ph. Seq.*	LINESY_3		BRANCH type 51-53	3-phase RL coupled line model with sequence impedance (0, +) input. Symmetric.
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 *+ 6 ph. Seq.*	LINESY_6		BRANCH type 51-56	2x3-phase RL coupled line model with sequence impedance (0, +) input. Symmetric.

4.11.4.2 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 types are distributed parameters, frequency independent lines of class Bergeron. Losses are concentrated at the terminals (R/4) and of the mid-point (R/2). The time step has to be less than half the travel time of the line.

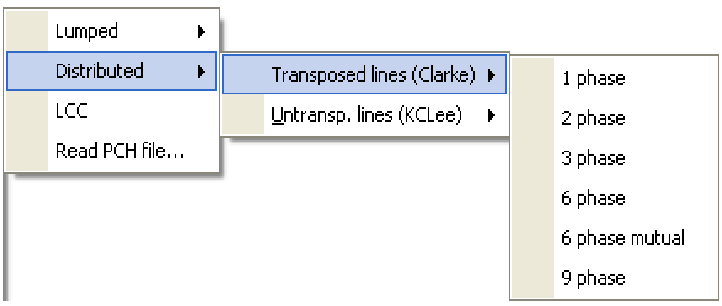
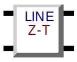
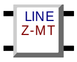
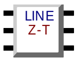
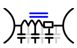
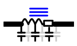


Fig. 4.60 – Distributed transmission line models.

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

4.11.4.3 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. The LCC module supports line/cable modeling up to 21 phases.

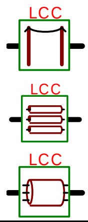
To use the *LCC* module of ATPDraw the user must first select a line/cable component. The number of phases is selected internally in the LCC dialog box. This will display an object (3-phases default) in the circuit window that can be connected to the circuit as any other component. Clicking on this component 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.

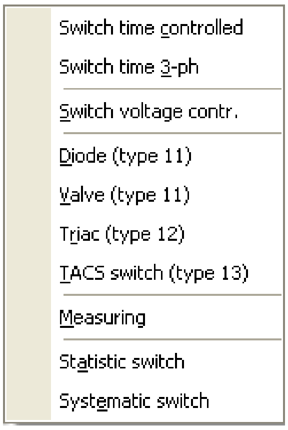
The *Line/Cable Data* dialog box completely differs from the *Component* dialog box of other components, therefore it is described in chapter 5.3 of the Advanced Manual.

Selection	Object name	Icon	ATP card	Description
LCC	`LCC_1..24`		$Include	1..24 phase LCC object. Overhead line Single core cables Enclosing pipe Bergeron/PI/Jmarti/Semlyen/Noda

4.11.4.4 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 and stores it in memory 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.

4.11.5 Switches



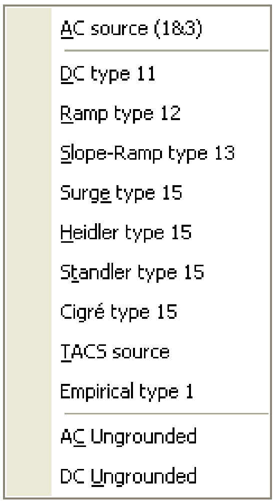
ATPDraw supports most of the switch type elements in ATP, such as ordinary time- or voltage-controlled switches, options for modeling diodes, valves and triacs, as well as measuring and statistical switches.

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

Fig. 4.61 – Supported switch type ATP components.

Selection	Object name	Icon	ATP card	Description
Switch time controlled	TSWITCH		SWITCH type 0	Single or 3-phase time controlled switch. Multiple closing/openings. Dynamic icon; will open, will close...
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.
Valve (type 11)	SW_VALVE		SWITCH type 11	Valve/Thyristor. Switch type 11. TACS/MODELS- controlled. GIFU.
Triac (type 12)	TRIAC		SWITCH type 12	Double TACS/MODELS controlled switch.
TACS switch (type 13)	SW_TACS		SWITCH type 13	Simple TACS/MODELS controlled switch. GIFU.
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*.
Nonlinear diode	DIODEN		SWITCH BRANCH	Ideal or nonlinear resistance with forward resistance and snubbers.

4.11.6 Sources



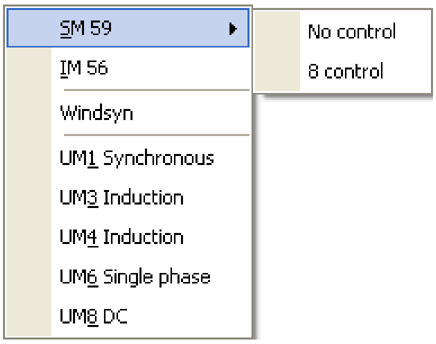
The popup menu under *Sources* contains the following items:

Fig. 4.62 – Electrical sources in ATPDraw.

Selection	Object name	Icon	ATP card	Description
AC source (1&3)	ACSOURCE		SOURCE type 14	AC source. Voltage or current. Single or 3-phase. Ungrounded or grounded. Phase voltage and rms

				scaling.
DC type 11	DC1PH		SOURCE type 11	DC step source. Voltage or current.
Ramp type 12	RAMP		SOURCE type 12	Ramp source. Voltage or current.
Slope-Ramp type 13	SLOPE_RA		SOURCE type 13	Two-slope ramp source. Voltage or current.
Surge type 15	SURGE		SOURCE type 15	Double exponential source Type-15. Voltage or current.
Heidler type 15	HEIDLER		SOURCE type 15	Heidler type source. Voltage or current.
Standler	STANDLER		SOURCE type 15	Standler type source. Voltage or current.
Cigre	CIGRE		SOURCE type 15	Cigre type source. Voltage or current.
TACS source	TACSSOUR		SOURCE type 60	TACS/MODELS controlled source. Voltage or current.
Empirical type1	SOUR_1		SOURCE type 1	Source with user defined time characteristic. Voltage or current.
AC Ungrounded	AC1PHUG		SOURCE type 14+18	Ungrounded AC source. Voltage only.
DC Ungrounded	DC1PHUG		SOURCE type 11+18	Ungrounded DC source. Voltage only.

4.11.7 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. 4.63 – 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.

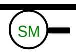
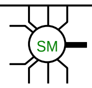
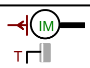
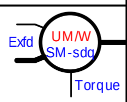
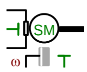
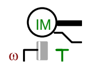
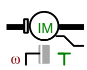
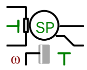
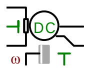
The *Universal Machine* models in ATPDraw have the following features/limitations:

- Manual and automatic initialization.
- SM, IM and DC type supported.

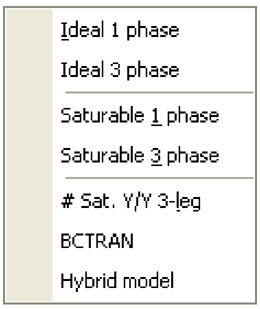
- Raw coil data (internal parameters). Manufacturers data in Windsyn.
- Saturation is supported in d, q, or both axes.
- Maximum five excitation coils, sum d and q axis.
- Network option for mechanical torque only.
- Single torque source.

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 5.2.2 of the Advanced Manual. The Windsyn component depends on a compatible, external program called WindsynATPDraw.exe developed by Gabor Furst. The component takes manufacturers data as input and calles the Windsyn program to fit these to electrical universal machine data. Windsyn supports the following machine types; Synchronous machines with salient or round rotor with damping options. Induction machines with wound, single cage, double cage, or deep-bar rotors. The Windsyn component is documented it chapter 5.2.5 in 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.
IM 56	`IM56A`		MACHINE Type 56	Induction machine with multiple controls. 3-phase armature.
Windsyn	`WISIND/ WISSYN`		UM-MACHINE Type 1, 3, 4	Universal machine with manufacturers data input.
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.*

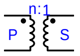
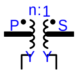
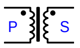
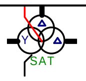
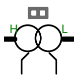
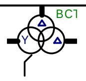
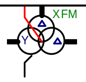
4.11.8 Transformers



ATPDraw supports the transformer components; Ideal transformer, saturable transformer, BCTRAN and the Hybrid Transformer. The BCTRAN model is documented in chapters 5.6 and the Hybrid Model in chapter 5.7 of the Advanced Manual.

Fig. 4.64 – 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	`SATTRAFO`		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.
Hybrid model	`XFMR`		BRANCH	Winding resistance, leakage inductance, topologically correct core, capacitance. Test report, design data or typical.

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. 4.47. 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* `SATTRAFO` 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 5.2.1 of the Advanced Manual, respectively.

The Hybrid Transformer model is based on development made by Dr. Bruce Mork and his group at Michigan Technological University. It offers both advanced and simplified usage. The XFMR dialog box and the implementation is documented in chapter 5.7.2 of the Advanced Manual.

4.11.9 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 with expressions, call of other models or specification of `HISTORY` or `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:

- Create a support file manually under *Object | Model | New sup-file* and a corresponding `.mod` file.
- Create a `.mod` file externally or a Model text internally and relay on ATPDraw for automatic identification and layout/icon.

The Advanced part of this Manual Chapter 5.5 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*

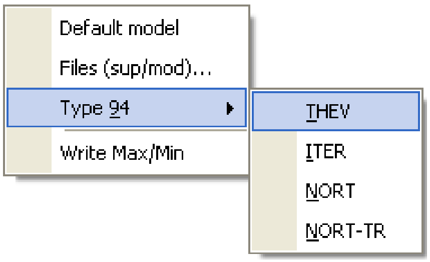


Fig. 4.65 – Options under the MODELS sub-menu.

Default model

This will load a simple, default model and display it in the circuit window. Its input dialog box will look as shown in Fig. 4.66 (no data or nodes). Click on the *Edit* button to modify the Model text directly or to import a model text from file or clipboard. In the standard text editor that pops up you can modify the Model text and import text via *File|Import* or *Edit|Paste*. Click on *Done* in the main menu of the Text Editor when finished. ATPDraw will then try to identify the model and create the component definitions, including icon, see Fig. 4.67. Inputs and outputs are placed to the left and right of the icon, respectively. You can whenever you want go back and modify the Model text, and if you change the number of input and outputs the icon will be recreated.

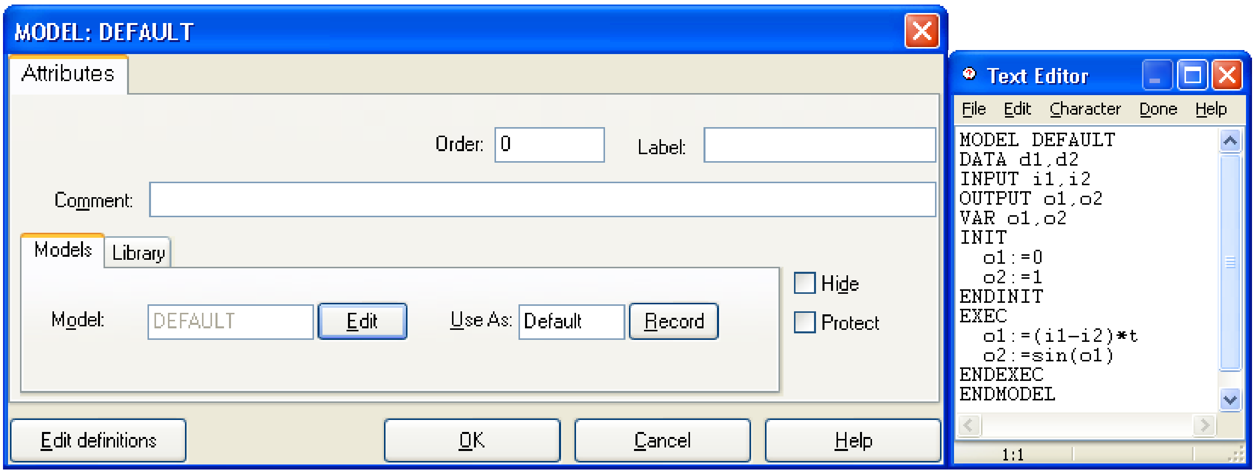


Fig. 4.66 – Model component dialog box. And Text Editor

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. 4.67 and a model component with the corresponding definition and icon appears. If a support file with the same name as the model file exist in the same folder, this file is used instead as basis for the model definitions. In this case the new model object appears immediately in the circuit window, i.e. the *Information* dialog shown in Fig. 4.67 does not show up.

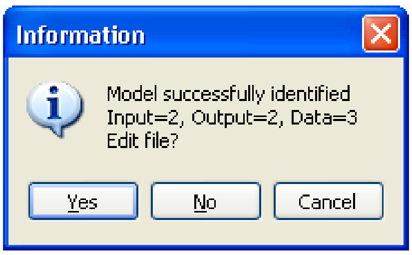


Fig. 4.67 – Interpretation of the model.

The *Component* dialog box of model objects has a new input section *Models* below the *DATA* and *NODES* attributes as shown in Fig. 4.68. This new section has two fields: *Model* which is disabled (but automatically follows what is defined in the Model text found using the *Edit* button) and a *Use As* field for specification of the *model_name* in the `USE model AS model_name` statement of MODELS. The *Record* button is used for output of internal model variables. On the *Library* page the link to the original support file on disk is given and a *Reload* option is made available. Remember that the original support file on disk not necessarily match the present Model text if the user has changed this.

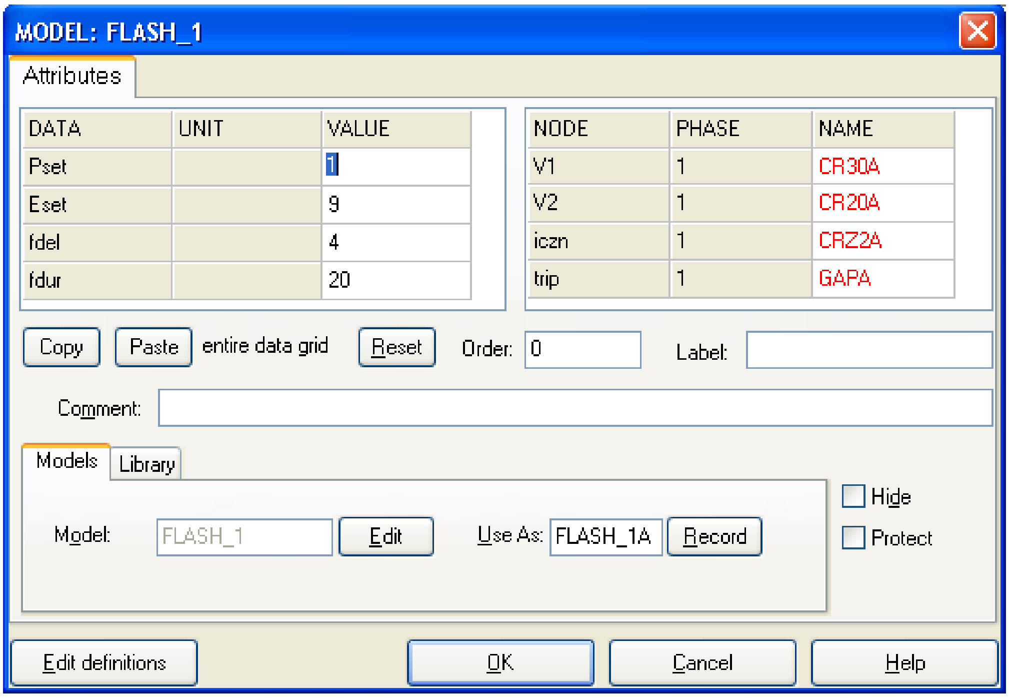


Fig. 4.68 – The component dialog box of model object FLASH_1.

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(CR30A)}
  IX0002 {v(CR20A)}
  IX0003 {v(CRZ2A)}
OUTPUT
  GAPA
MODEL FLASH_1
-----------------------------------------
Description of the model 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
```

Type 94

Selecting *MODELS | Type 94|THEV, ITER, NORT, NORT-TR* will load a corresponding default model component. You can then open the component which will bring up the Type 94 component dialog box as shown in Fig. 4.69. As for simple models you can then click on the *Edit* button to inspect or modify the type 94 models text. When you click on *Done* in the Text Editor ATPDraw tries to identify the model and then displays a message box similar to Fig. 4.67. Be aware of that the name of the models must be six characters or less. The bottom section of the input dialog has to the right four radio buttons: *THEV, ITER* , *NORT* and *NORT-TR* for specification of the solution method for ATP when interfacing the Type-94 object with the rest of the electrical network. The Data, Node fields and the icon will update dependent on the choice of type. You can also specify the number of phases (*#Ph*: 1..26) in the component. Branch output and Record of internal variable are also available.

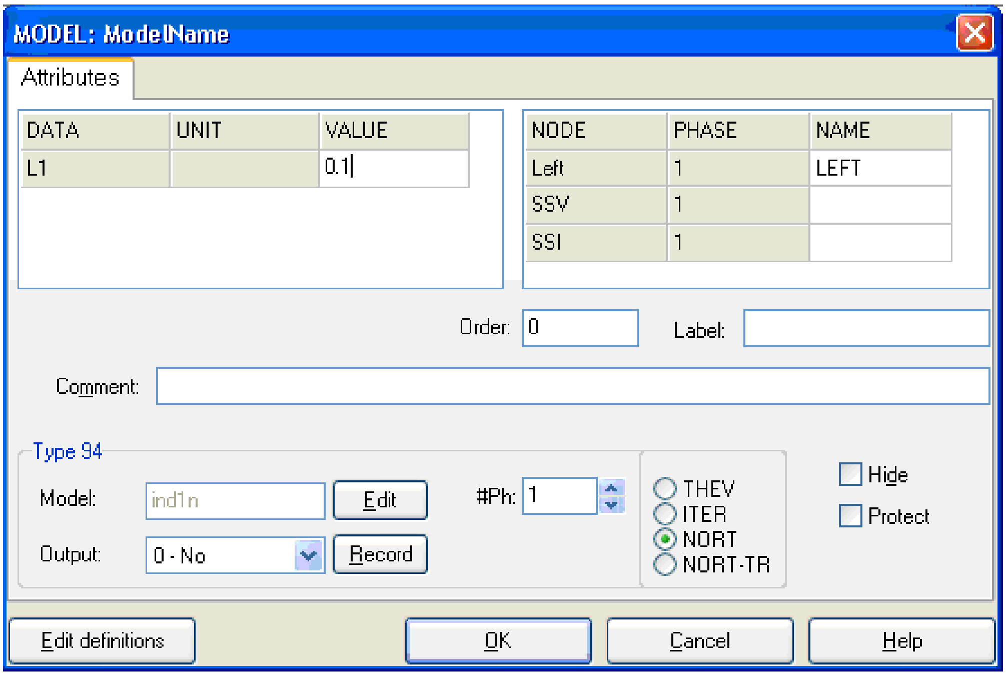


Fig. 4.69 – Component dialog box of Type-94 model objects.

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 - 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  -- –istory term for next step
 ELSE
  flag := 0          -- reset flag
  i    := g*v -is    -- –pplying 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.

```
C Time varying inductor
94LEFT        IND1N NORT                                               1
 >DATA  L1          0.1
 >END
```

Write Max/Min

This is a special cost function or reporting component using Models. The component extracts a value from a simulation by communication with the LIS file. As default the minimum or maximum value of a single input signal is extracted, but the user can add more sophistication to this. Only the signal after a user selectable time *Tlimit* are identified. The component supports multiple run via *ATP|Settings/Variables* and contains a *View* module for displaying the result. A data parameter *AsFuncOf* can be used to pass a loop variable from the global Variables (if a number is specified here, the simulation number is used instead). The component is used extensively in circuit optimization.

4.11.10 TACS

The TACS menu gives access to most type of TACS components of ATP. The TACS sub-menu on the component selection menu contains the following items:

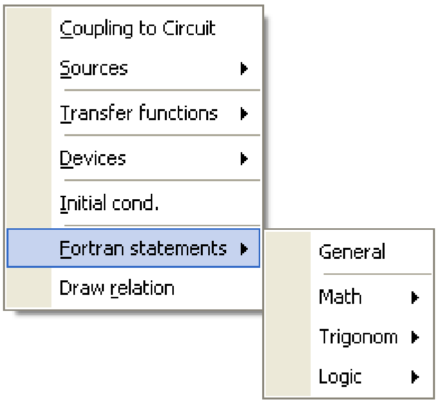


Fig. 4.70 - Supporte– TACS objects.

4.11.10.1 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. 4.71.

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.

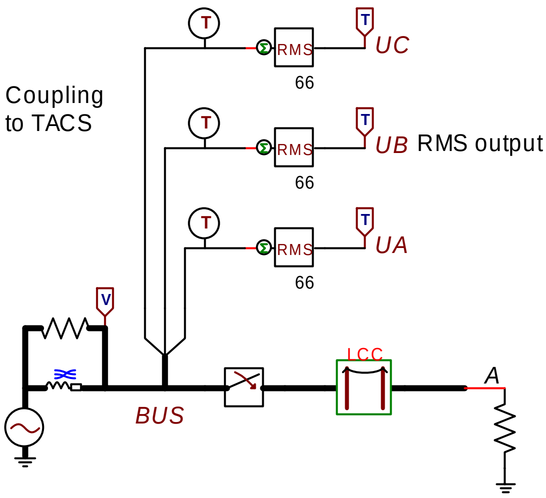


Fig. 4.71 - Coupling a 3-phase electrical node to TACS.

4.11.10.2 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.

4.11.10.3 TACS transfer functions

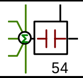
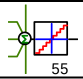
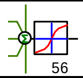
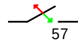
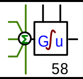
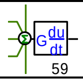
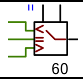
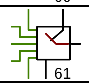
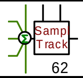
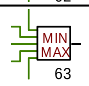
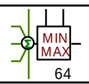
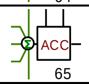
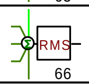
All the older TACS transfer functions of previous ATPDraw versions are supported in version 3, but some of them has been removed from the component selection menu and replaced by a more general component: the *General transfer function*. This object defines a transfer function in the *s* domain and it can be specified with or without limits. The *Order 1* component offers order 0/1 transfer function with a dynamic icon containing values and optional 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.
Order 1	TRANSF1		TACS	Order 0/1 with optional limits. Dynamic icon with transfer function.
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.

4.11.10.4 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.
Digitizer - 55	DEVICE55		TACS type 88,98 or 99	Digitizer.
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.

4.11.10.5 Initial conditions

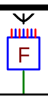
The initial condition of a TACS variable can be specified by selecting TACS object (type 77) under the *TACS* | *Initial cond.* menu. The name of this component is `INIT_T` and its icon is .

4.11.10.6 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

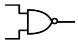
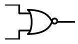
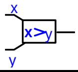
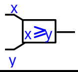
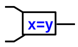
Selection	Object name	Icon	ATP card	Description
x - y	DIFF2	Σ	TACS 98	Subtraction of two input signals.
x + y	SUM2	Σ	TACS 98	Addition of two input signals.
*x * K*	MULTK	K	TACS 98	Multiplication by a factor of *K*.
*x * y*	MULT2	*	TACS 98	Multiplication of x by y.
x / y	DIV2	x/y	TACS 98	Ratio between two input signals.
| x |	ABS	|x|	TACS 98	Absolute value of the input signal.
- x	NEG	NEG	TACS 98	Change sign of the input signal.
sqrt (x)	SQRT	√	TACS 98	Square root of the input signal.
exp(x)	EXP	exp	TACS 98	Exponent of input signal. e^x
log(x)	LOG	log	TACS 98	Natural logarithm og input signal.
log10(x)	LOG10	log10	TACS 98	Logarithm of input signal.
rad(x)	RAD	RAD	TACS 98	Converts the input signal from degrees to radians.
deg(x)	DEG	DEG	TACS 98	Converts the input signal from radians to degrees.
rnd(x)	RND	RND	TACS 98	Random number generator <x.

Fortran statements / Trigonom

Selection	Object name	Icon	ATP card	Description
sin	SIN	sin	TACS 98	Sinus
cos	COS	cos	TACS 98	Cosinus
tan	TAN	tan	TACS 98	Tangens (sin/cos)
cotan	COTAN	cotan	TACS 98	Cotangens (cos/sin)
asin	ASIN	asin	TACS 98	Inverse sinus
acos	ACOS	acos	TACS 98	Inverse cosinus
atan	ATAN	atan	TACS 98	Inverse tangens
sinh	SINH	sinh	TACS 98	Sinus hyperbolic
cosh	COSH	cosh	TACS 98	Cosinus hyperbolic
tanh	TANH	tanh	TACS 98	Tangenss hyperbolic

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.
=?	`EQ`		TACS type 98	Logical operator. Output = 1 if x = y, 0 otherwise.

4.11.10.7 Draw relation

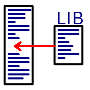
When you select *TACS | Draw relation*, the mouse cursor will change to a pointing hand and the program is waiting for a left mouse click on a circuit node to set the starting point of a new relation. You can then draw multiple relations until you click the right mouse button or press the *Esc* key. Relations are used to visualize information flow into Fortran statements. These objects are drawn as blue, dashed 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.

4.11.11 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.

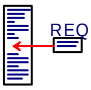
Fig. 4.72 - Supported user specified objects.

Library



Using this object will result in a $Include statement in the ATP-file inserted in the BRANCH part. No parameters are used in this case. The *User specified* section at the bottom contains an *Edit* button that brings up the Text Editor where the user can edit or import an external text. The user can type in the name of the component in the $Include field. The text will be dumped to a file with this name and extension .lib and location in Result Directory (same as ATP file) when the ATP file is created.

Additional



Similar to the *Library* component but in addition it allows the user to choose under which section in the ATP file to insert the text. The input dialog of this component contains a larger memo field where the user can write in free format text with a row and column indication below. The Additional section at the bottom contains an *Edit* button that brings up a more advanced Text Editor that allows the user to import a text from file of clipboard. This Text Editor also has a right-click context menu with an Insert option of 50 predefined request cards. There is no $Include field in this component because the text will be inserted directly into the ATP file. Instead the user can select the section; REQUEST, TACS, MODELS, BRANCH, SWITCH, STATISTICAL, SOURCE, INITIAL, OUTPUT, LOAD FLOW, MACHINE type 59/56, UNIVERSAL MACHINE, FREQUENCY COMP. The *Order* number can be used for fine tuning of the location within each section (together with *ATP|Settings/Format-Sorting by Order*). The three character text in the icon will adapt to the selected section.

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.

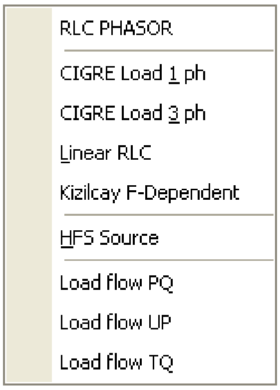
Files...

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 *Library* | *New object*|*User Specified* 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. In the dialog box of this component type there is a *User Specified* section with an *Edit* button which will bring up the Text Editor where a `.lib` file can be imported. A checkbox *Send parameters* is used if the library file is on the Data Base Module format with external parameters. A second checkbox *Internal phase seq.* is used if the phase extension 'A', 'B'… is hard coded inside the Data Base Module and only the five character root node name should be sent. Henceforth the user specified objects operate similarly than standard objects.

4.11.12 Steady-state

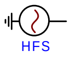
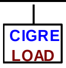
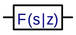
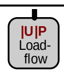
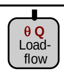


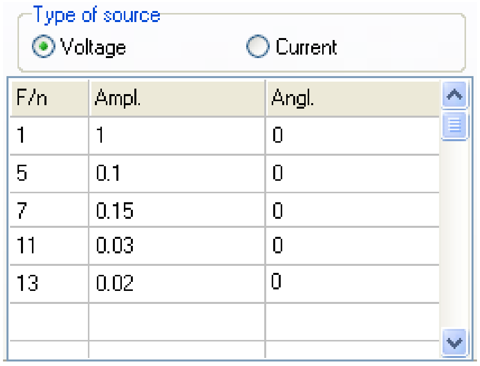
Harmonic frequency scan and load flow

Fig. 4.73 - Supported HFS components.

The Harmonic Frequency Scan (HFS) is one of the options under *ATP* | *Settings / Simulation*. General load flow specification is given under *ATP*|*Settings/Load flow*.

Selection	Object name	Icon	ATP card	Description
RLC Phasor	`RLC_PHASOR`		BRANCH	RLC component only present during steady-state (t<0)

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
KizilcayF-dependent	`KFD`		BRANCH type 0	Frequency dependent branch in s or z domain.
Load flow PQ	`LF_PQ`		SOURCE Load flow	Load flow component with active and reactive power restriction
Load flow UP	`LF_UP`		SOURCE Load flow	Load flow comp. with voltage and active power restriction
Load flow TQ	`LF_TQ`		SOURCE Load flow	Load flow component with angle and reactive power restriction



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. 4.74 - 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. 4.74.

4.11.13 Standard Component...

In ATPDraw 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.

4.11.14 Plugins

The Plugins Item points to a user defineable disk structure with project files (`.acp`) and sub-folders. This thus gives an easy access to a user defineable library of sub-circuits for import. This is similar to *File|Import* but enables the possibility of direct access. The Plugin directory is defined under *Tools|Options/Files&Folders*.

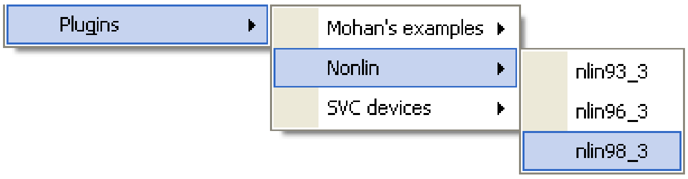


Fig. 4.75 – Example of Plugins menu.

5. Advanced Manual . . .

. . . .

ATPDraw™
for Windows
5.6

This chapter gives an overview of several more advanced features in ATPDraw: Grouping, special components, usage of the integrated LINE/CABLE CONSTANTS, BCTRAN and the UNIVERSAL MACHINE support, including the Hybrid Transformer model and Windsyn. This chapter also 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_*.acp`) 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: an ATPDraw feature for multilevel modeling

The grouping feature in ATPDraw allows multilevel modeling by replacing a group of objects with a single icon in an almost unlimited numbers of layers. The grouping structure can be imagined as a multi-layer circuit, where the *Edit* | *Edit Group* brings you one step down in details, while the *Edit* | *Edit Circuit* menu brings you one step back. This feature increases the readability of the circuit and the feature is especially useful for TACS blocks or frequently reused circuit elements. The grouping feature is demonstrated by re-designing the circuit `Exa_4.acp` 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.

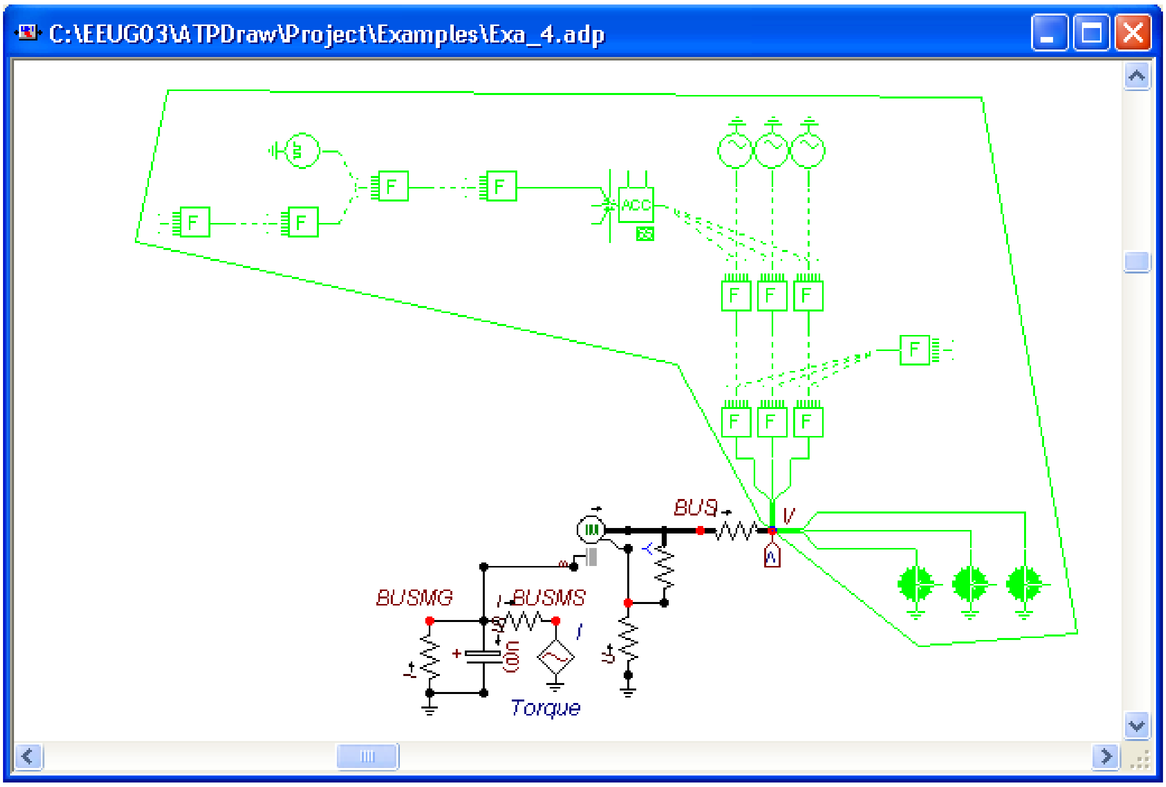


Fig. 5.1 - An induction machine supplied by a pulse width modulated voltage source.

The process of creating a group is as follows:

- Select a group of components (inside the polygon in Fig. 5.1). *Edit|Select|Inside*.
- 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. First the selected sub-circuit is redrawn alone in the middle of the circuit window and the *Compress* dialog appear as shown in Fig. 5.2.

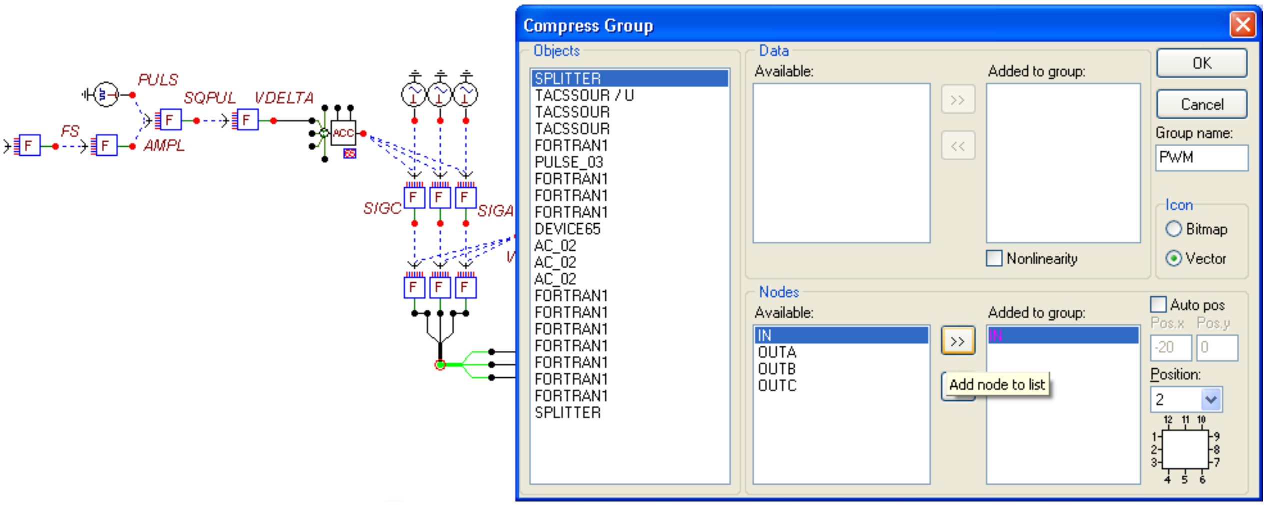


Fig. 5.2 - The *Compress* dialog window.

In the *Compress* dialog box the user can specify the external data and nodes of a group of components. The selected data and nodes appear as input in the group object that replaces selected group and their values are automatically transferred.

Under *Objects* all the components in the group are listed with their name (support file) followed by '/' and their *Label* (which is again specified in the *Component dialog* box). When you click on one of the components it's available data and nodes appears under *Available* listed by the data's/node's name followed by its value. The component is also drawn in a lime color in the circuit window. The already selected external data/node belonging to this component is also drawn with a lime color in the *Added to* groups.

You can then select a parameter and click on the >> button to transfer it to the *Added to* list. Selected nodes in the *Available* node list is also draw in a lime color in the circuit window. If the data/node is already added the corresponding item in the *Added to* lists is highlighted, and you are not allowed to select it twice. Nodes in the *Added to* list are drawn enclosed by a red ring in the circuit window as shown for the 3-phase node of the Splitter chosen to the external in Fig. 5.2. The node position 2 is chosen for this node and this is the middle left standard position.

Vector icon is chosen for this group object. The *Group name* PWM is used in the icon and displayed as an indicator in the *Component dialog*, as shown in Fig. 5.5. The *Auto pos* option is available for vector icons only. Later in this example we will change the icon to bitmap style.

All data and nodes listed in the *Added to* groups will be the external attribute of the new group object. You can also for each selected node specify it's position relative to the object's. The node positions different from the default 1-12 must be specified by selecting *Position* 0 and then give the relative coordinates of the node in the *Pos.x* and *Pos.y* fields. The x-axis is oriented to the right and the y-axis downwards. The *Auto pos* button is only available for Vector graphic icons. Selected data and nodes can also be removed from the *Added to* groups by clicking on the << button. As all other components the group object is limited to 64 data and 32 nodes. When later opening the component dialog box for the group object the selected data and node parameters will appear as input possibilities and the values will automatically be transferred to the sub-group.

It is also possible to change the data/node labels by double-clicking on the texts in *Added to* lists. **Important! Two or more data labels with the same name are treated as a single data in the component dialog box.**

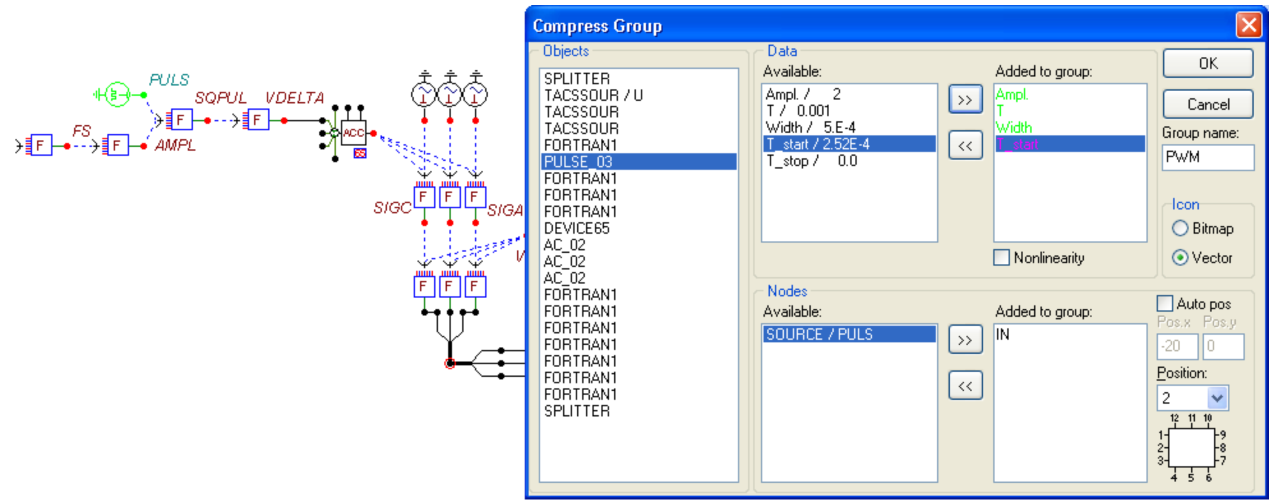
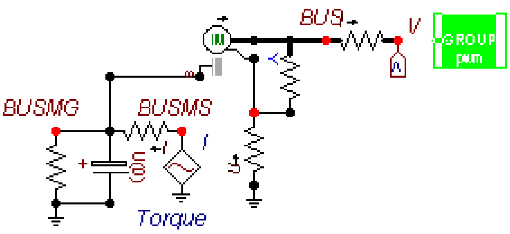


Fig. 5.3 - Name and position of the external nodes of the group.

The Compress process continues in Fig. 5.3 by selection of the external data all belonging to the PULSE_03 object. Click on *OK* when you have finished. If you need to change the group attributes, you can later selecte the group and once again choose *Edit|Compress* to reopen the Compress dialog. In such case a *Keep icon* checkbox enables you to preserve the the groups icon.



After selecting all the required data and nodes click on *OK*, then a object will automatically be created. The group content disappears 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 as specified under Fig. 5.2 and Fig. 5.3.

When changing the data parameter in this window the value will also be transferred to the member components. A change in the node name will be transferred in the same way. In this particular 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 real names VC, VB and VA (from left to right) at the Fortran objects' output node. The user must follow this phase sequence in the PWM group object, too.

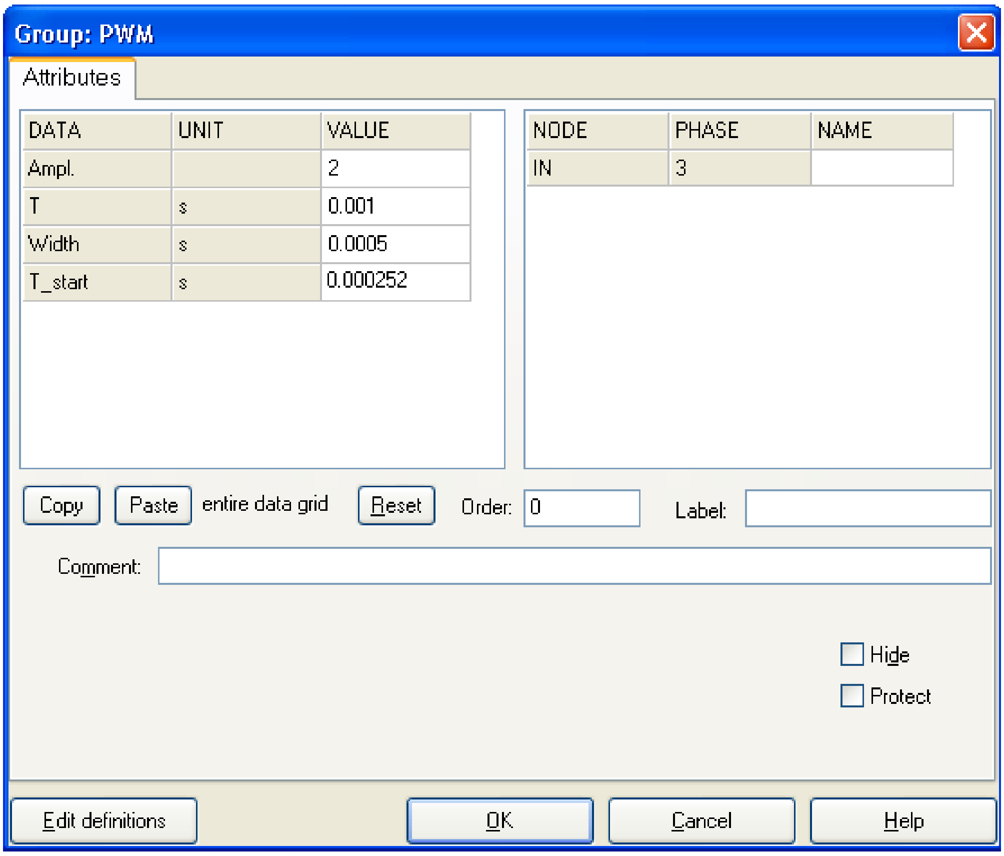


Fig. 5.5 - Opening the new group dialog box.

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.

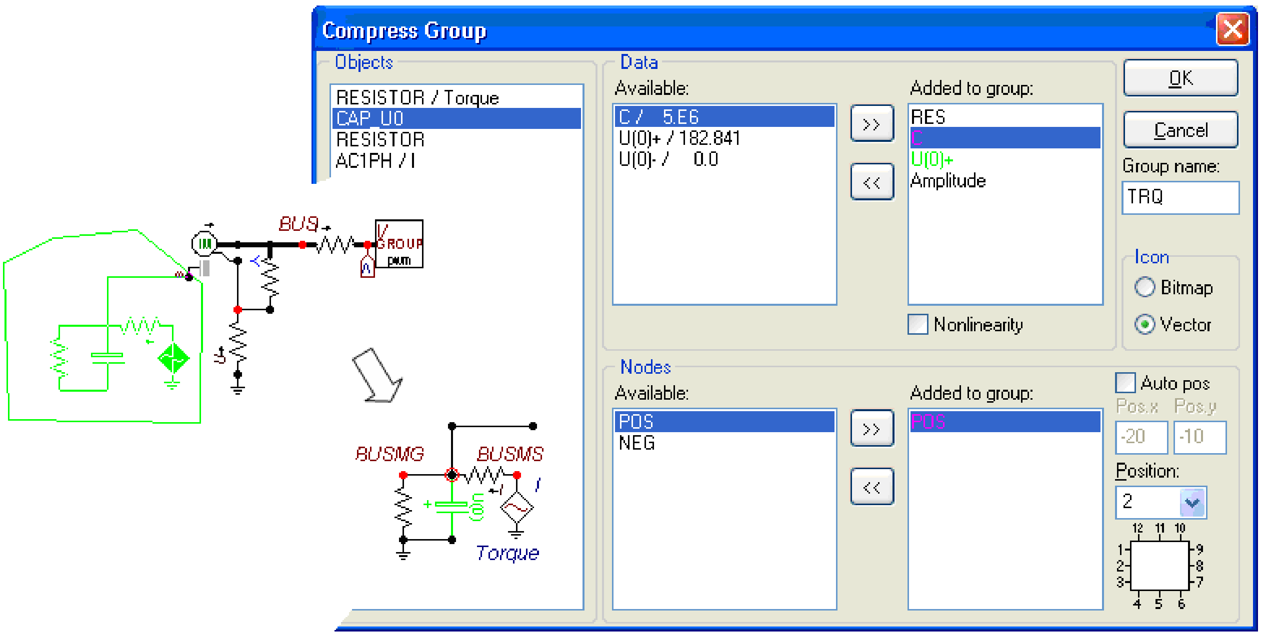


Fig. 5.6 - Selection of data values and external nodes for the mechanical load group.

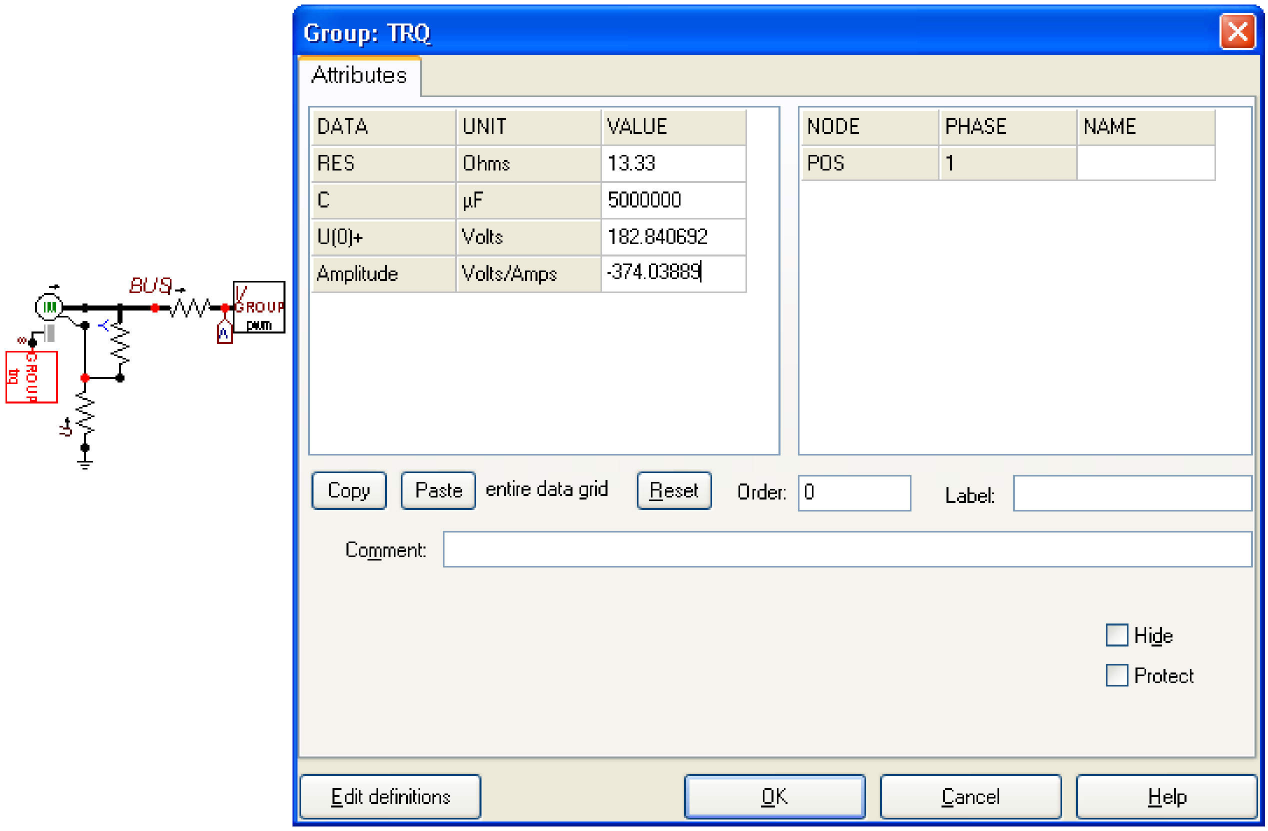


Fig. 5.7- Component dialog box of the mechanical load group-object.

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.

To customize the icon, click the *Edit definitions* 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.acp` 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.

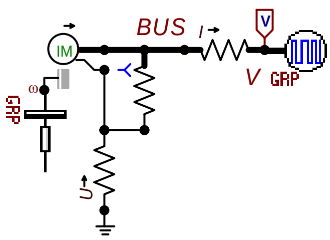


Fig. 5.8 - The icon of the PWM source and the load group has been customized.

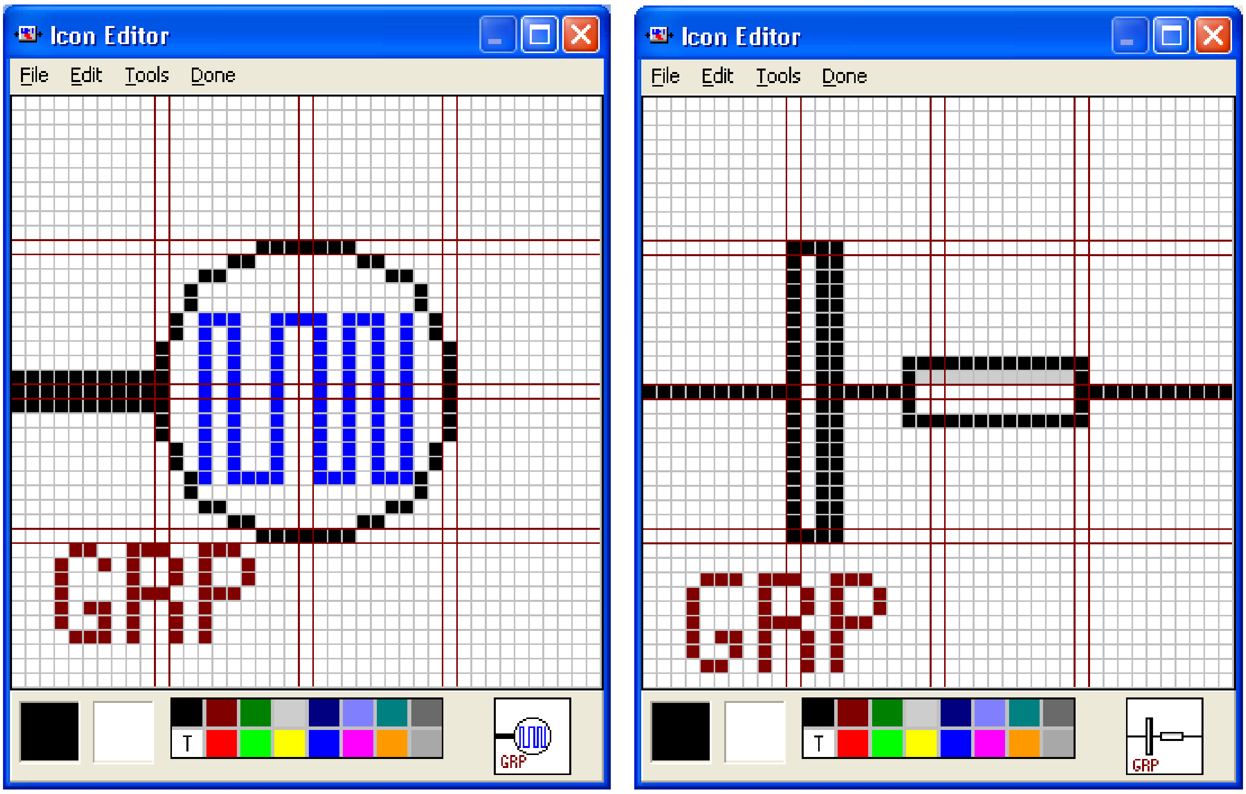


Fig. 5.9 - Customizing the icon of the PWM source and TRQ mechanical torque model. The icon is oriented so that node connections fit with border position 2 (left, middle).

5.1.1 Grouping nonlinear objects

A non-linearity can also 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`, `RESID`, and `FL(0)` which are used for all phases, although 3 copies of them are present in the data structure. This results in 26 free data cells available for the nonlinear characteristic (64-3*4)/2 = 26). Only one characteristic is entered in the group's dialog box and it is later copied back to all the three inductors. If the 26 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, or place in the *Plugins* library.

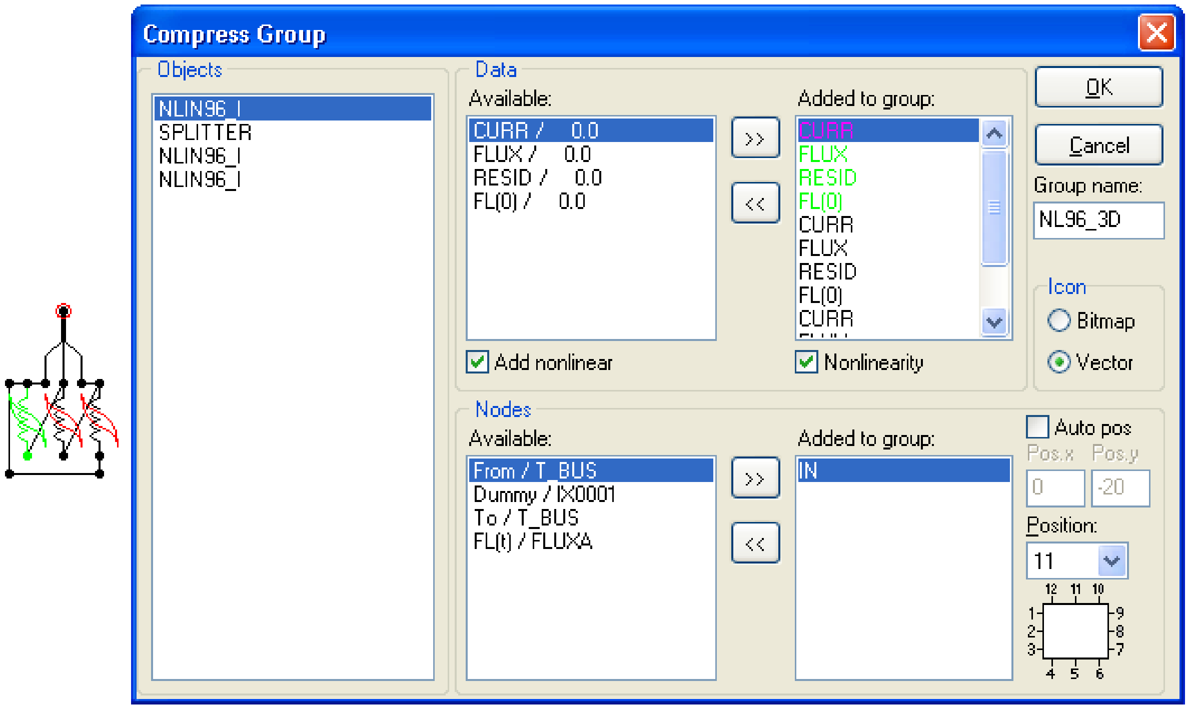


Fig. 5.10 - Creating a 3-phase hysteretic inductor.

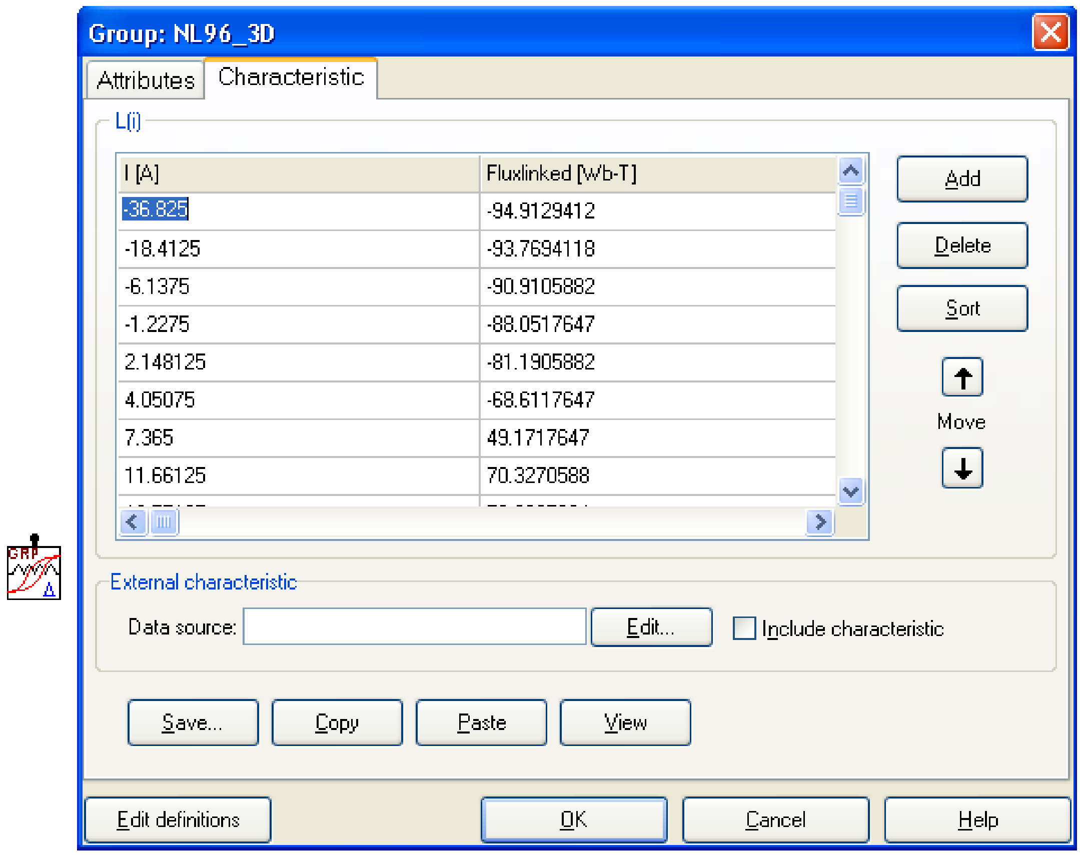


Fig. 5.11 - Nonlinear characteristic of the 3-phase Type-96 group
(notice that only one characteristic is specified, that is used for all phases).

You can customize the group icon as shown in Fig. 5.11 (vector icon illustrated in this case). The hysteresis loop originates from the original inductor icon. This is done by executing the next sequence of operations: click on *Edit definitions* and go into the vector icon editor (leftmost speed button). The default icon is shown as a box with the text 'GROUP' and 'nl96_3d'. Modify the 'GROUP' text to 'GRP' and move it toward the upper left corner of the box. Modify the text 'nl96_3d' to 'D' and choose font 'symbol' (you may also increase the font size and pick a different color) and move it towards the lower right corner of the box. Now choose *File|Append std* and choose the standard icon NLIND96. Adjust the left and right node connections. Click on *Done*.

5.2 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 and nodes can be specified, on the optional *Characteristic* page you specify the input characteristic of non-linear components.

The following components deviate somewhat from the above description:

- Saturable 3-phase transformer (SATTRAFO)
- Universal machine (UM_1, UM_3, UM_4, UM_6, UM_8)
- Statistical / Systematic switch (SW_STAT, SW_SYST)
- Harmonic source (HFS_SOUR)
- Windsyn manufacturers data UM component.

In additions comes Models and User Specified component, explained later.

5.2.1 Saturable 3-phase transformer

The component dialog box of this transformer model is shown in Fig. 5.12. 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 *Order*, *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 is given in volts. 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.*

Windings coupled in wye, delta, auto with all possible phase shifts are supported. In addition zigzag configuration can be selected with arbitrary phase shift from <-60,0>+<0,60>. In this case the winding is split in two parts internally and the leakage inductance recalculated.

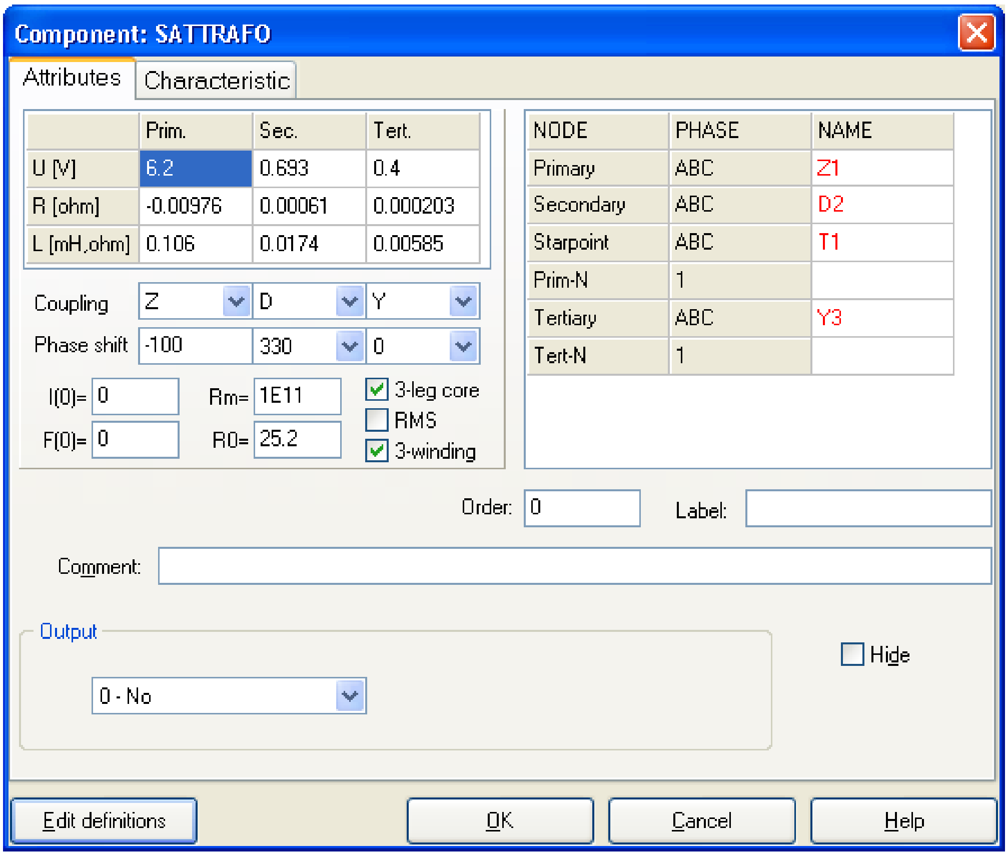


Fig. 5.12 - General saturable transformer dialog.

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 SATTRAFO object. This help text briefly describes the meaning of input data values:

```
Name : SatTrafo - General saturable transformer. 3 phase. 2 or 3 windings.
       Wye, Delta with all phase shifts. Auto, and Zigzag.
Card : BRANCH
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.
               The magnetizing branch is always connected to the PRIMARY winding and Rm
               is referred to this voltage.
           R0= Reluctance of zero-sequence air-return path for flux. 3-leg core-type
          Vrp= Rated voltage in [V] primary winding (only the voltage ratios matter).
           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.
          Vrs= Rated voltage in [V] secodary 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.
          Vrt= Rated voltage in [V] tertiary 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.
          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.
Output specified the magnetization branch output (power&energy not supported).
Node :      P= Primary side. 3-phase node.
```

```
          S= Secondary side. 3-phase node.
         PN= Neutral point primary side.
         SN= Neutral point secondary side.
          T= Tertiary side. 3-phase node.
         TN= Neutral point tertiary side.
        Sat= Internal node, connection of the magnetization circuit with saturation.
The coupling is specified for each winding, with four coupling options: Y, D, A, Z
  All phase shifts are supported.
Special note on Auto-transformers:
  The primary and secondary windings must be of coupling A(uto).
Special note on ZigZag-transformers:
  For this type the user can specify a phase shift in the range <-60,0>&<0,60>.
  Note that the values -60, 0 and +60 degrees are illegal (as one of the winding parts
  degenerates).
  The phase shift is given relative to a Y-coupled winding.
  If the primary winding is Zigzag-coupled, all other windings will be shifted with it.
  If the primary winding is D-coupled, 30 deg. must be added/subtracted to the phase
  shifts.
  For negative phase shifts the phase A winding starts on leg 1 (called z with voltage
  Uz) and continues in the opposite direction on leg 3 (called y with voltage Uy).
  For negative phase shifts the phase A starts on leg 1 and continues in the opposite
  direction on leg 2.
  The normal situation is to specify a phase shift of +/- 30 deg. in which case the two
  parts of the winding have the same voltage level and leakage impedance.
  In general the ratio between the second part of the winding Uy and the first part Uz
  is n=Uy/Uz=sin(a)/sin(60-a) where a is absolute value of the phase shift.
  This gives:
  Uz=U/(cos(a)+n*cos(60-a)) and Uy=Uz*n
  Lz=L/(1+n*n) and Ly=Lz*n*n, Rz=R/(1+n) and Ry=Rz*n
  where Lz and Ly are the leakage inductance of each part of the winding (L is the
  total leakage inductance) and Rz and Ry are the winding resistance of each winding
  part (R is the total).
  The parameters Uz, Uy, Zz, and Zy are automatically calculated by ATPDraw based on
  the equivalent parameters U and Z and the phase shift, a.

Points: It's possible to enter 23 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).
RuleBook: IV.E.1-2 or 3.
```

5.2.2 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. 5.13. 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 | Settings / 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.

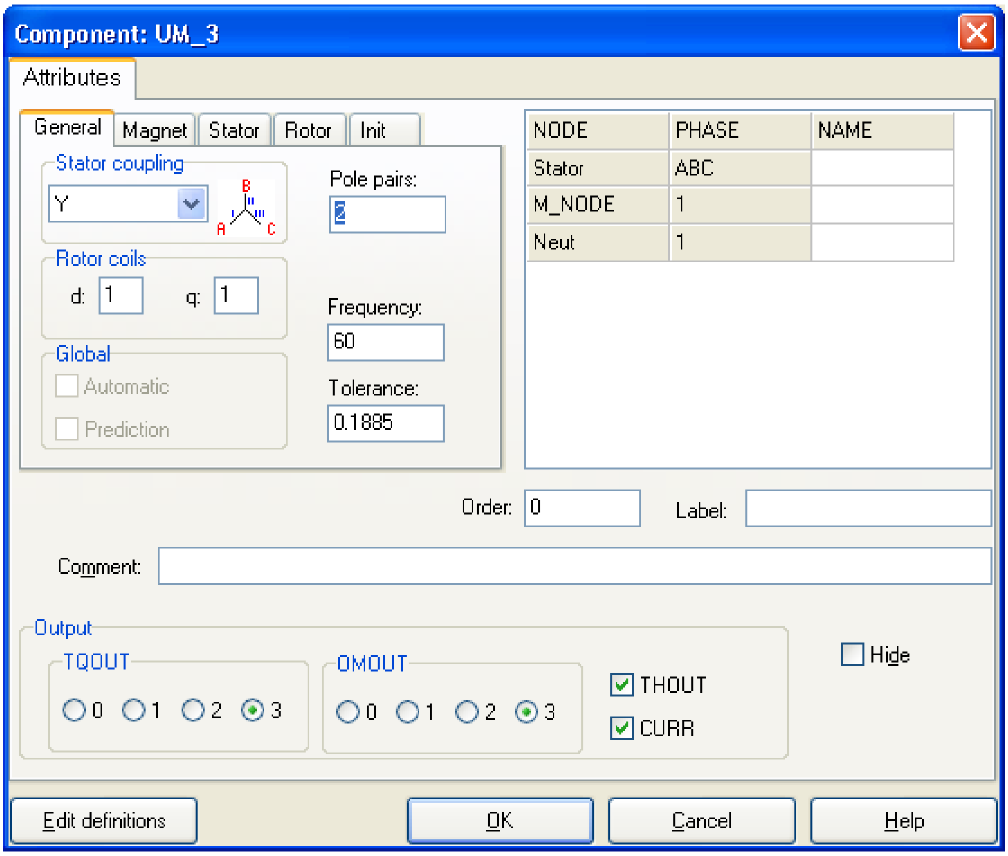


Fig. 5.13 - Universal machine input dialog.

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. 5.14 shows the various pages for universal machine data input (induction machine, type 3). 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 *Order*, *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.

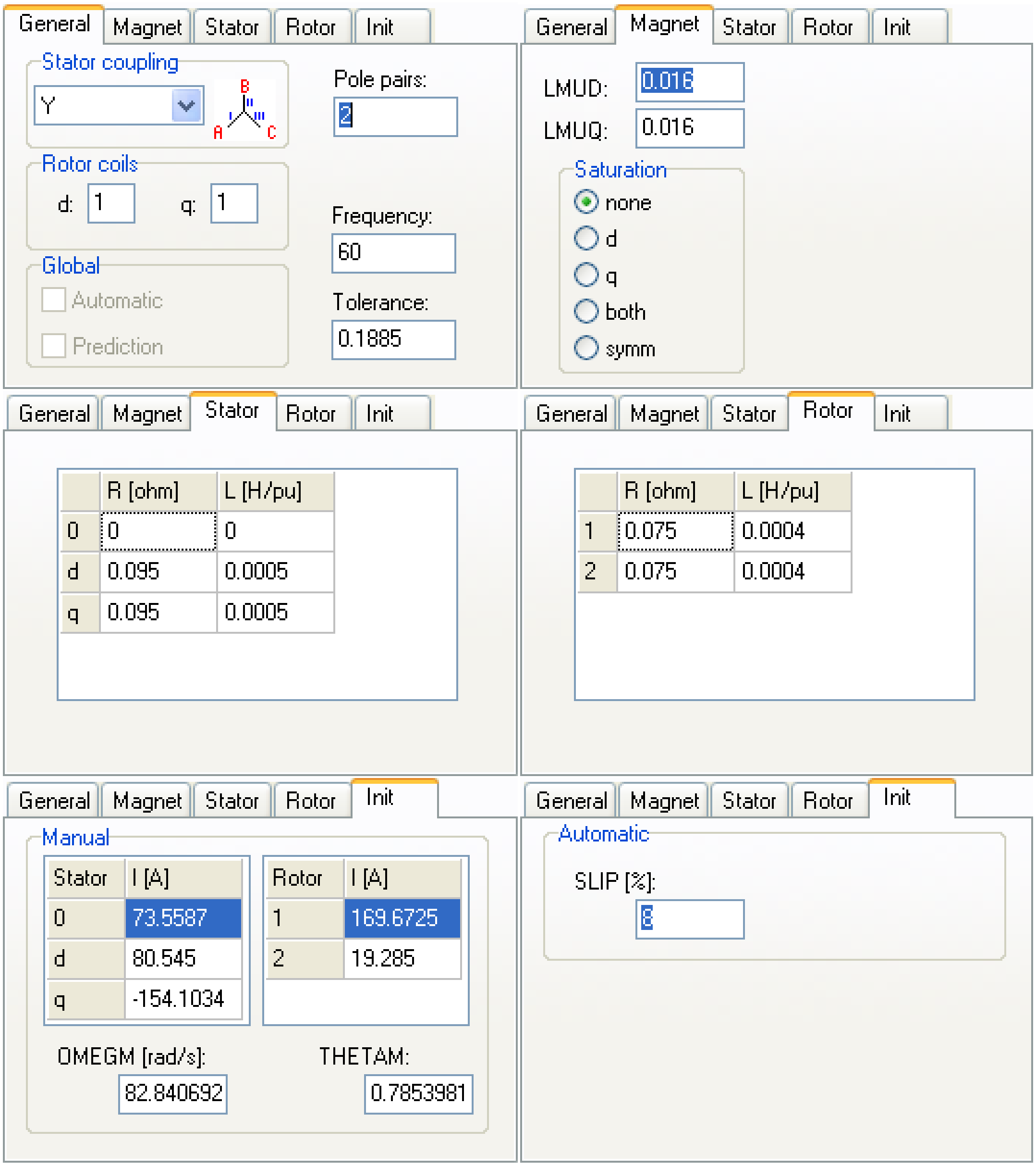


Fig. 5.14 - Data pages of the universal machines dialog box.

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)
```

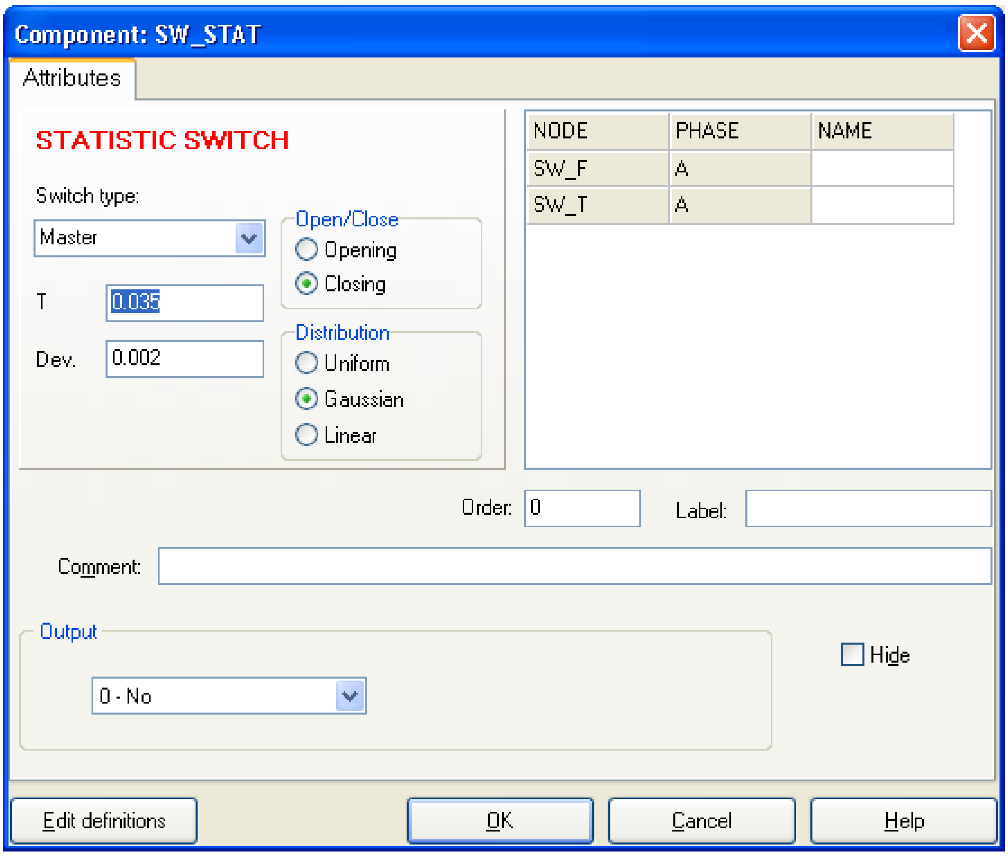
L1

```
          __       __                      _____ ___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
```

5.2.3 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. 5.15.

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 1^{st} 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 *Order*, *Label*, *Comment* and *Output* fields are the same as for any other standard components.



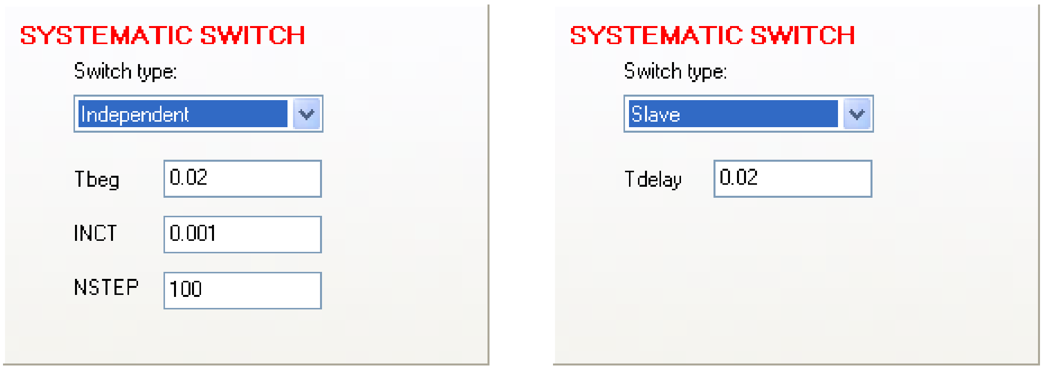


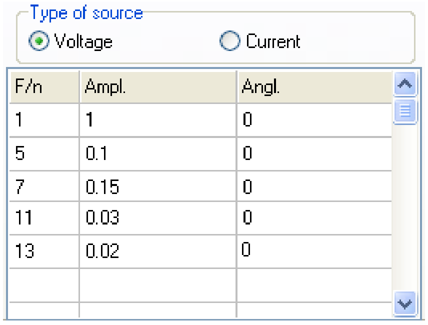
Fig. 5.15 - Dialog box of the statistic switch (top) and data windows of the systematic switch.

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
```

5.2.4 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. 4.74.



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. 5.16 - 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.*

5.2.5 Windsyn component

Windsyn is a program by Gabor Furst in Vancouver-Canada. It takes manufacturers machine data as input, makes a fitting and produced an electrical universal machine model with startup. This facilitates the usage of electrical machines in ATP/ATPDraw considerably. Seven electrical different machine types are supported by Windsyn; Induction machines (wound, single cage, double cage, and deep-bar rotors), Synchronous machines (salient rotor; d-damping, dq-damping, round rotor; dq-damping).

The machine number is used in the control process of the machine, but this number is automatically assigned by ATP as the data file is processed (machine number=sequence in the file). The ATPDraw compatible version of Windsyn is completely transparent related to the machine number. In the newer ATP version all UM machines can have the header (INPU (units), INITUM (initialization method), ICOMP (solution method)) included, but only the settings of the first machine will be used.

The ATPDraw compatible version of Windsyn is modified to accept an input file on its command line and automatically update and jump to the input screen. The input file is a simple, free format, text based file starting with ‘WindSyn Data’ and ending with ‘End of WIS data’. ATPDraw creates an input file from scratch called atpdraw.wis and dumps it to the same directory as the windsyn.exe file. The Windsyn is called as `Path\Windsyn.exe atpdraw.wis, INITUM, ICOMP`. The Windsyn program is documented separately.

Windsyn just dumps a dat-file, and ATPDraw calls ATP to produce the lib-file in data base module format. The call to the lib-file is ‘$Include, Name, BUS, ROTM, TORQUE, EXFD, MachineNumber#’.

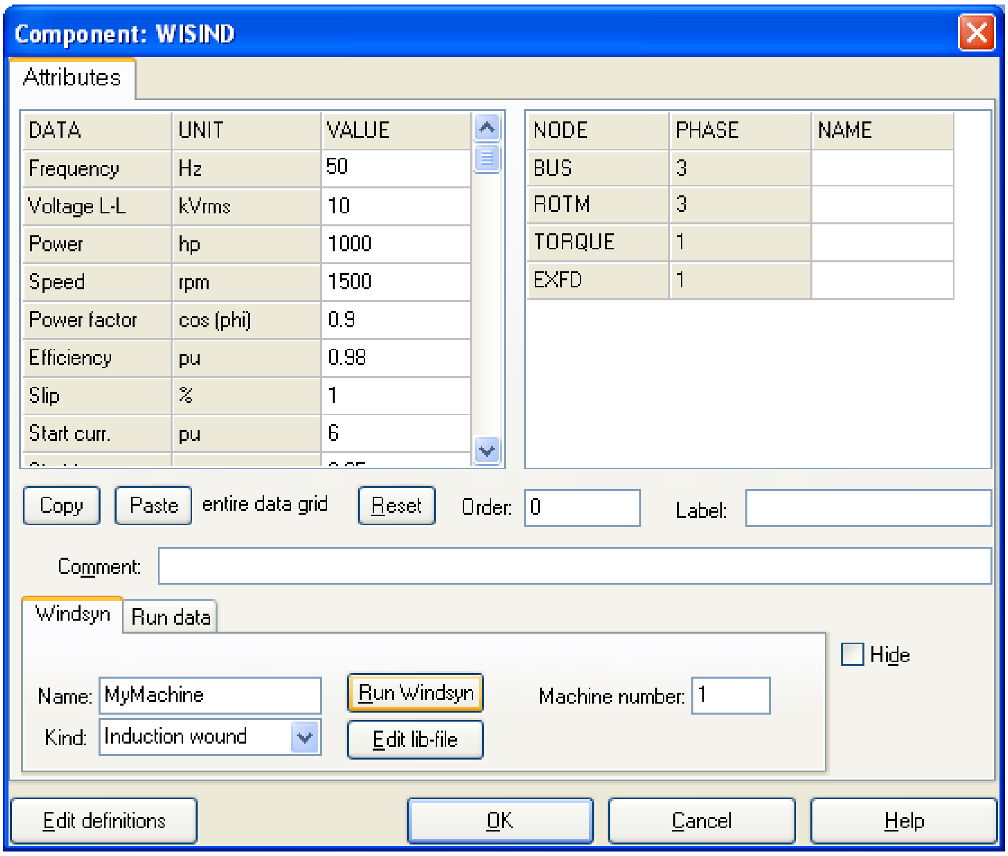


Fig. 5.17 - Windsyn dialog box in ATPDraw.

Fig. 5.17 shows the Windsyn input dialog in ATPDraw. It follows the same design as most components. The input data consists of the standard Data grid to the left and a page control at the bottom. On the Windsyn page the user can select the type of machine and give the machine model a name that is used in the final $Inlude call. The current machine number is presented to the user, but this number could change as the circuit develops. As default an induction machine with wound rotor is assumed. If the user changes the machine type under *Kind* the Data grid is automatically updated. If the *Kind* is changed from induction to synchronous machine or vice versa new default values are read in. In the background there are actually two support files; `WisInd.sup` and `WisSyn.sup`, stored in `ATPDraw.scl` containing icon, help, default values, and units. On the '*Run data*' page the user can set the start-up options for the machine dependent on the machine type and initialization INITUM set under *ATP|Settings/Switch/UM*.

In order to use the Windsyn component in a data case the user has to run the Windsyn program via the *Run Windsyn* button. Then the input file to Windsyn `atpdraw.wis` is created and Windsyn executed. While Windsyn is running the text "Windsyn is running (ESC)" is displayed. ATPDraw waits for Windsyn to terminate befor reading in the result files. The waiting process can be interupted by pressing the escape key. This might be needed if Windsyn terminates incorrectly. In Windsyn the user can change the machine model and in the end create new output files (*Save Data* followed by *Exit*). The output/result files are then automatically loaded by ATPDraw and a $include file is created as seen when clicking the *Edit* lib-file button. It is possible to directly import a lib-file here and thus omitting the Windsyn step. Both the wis-file and the lib-file are read and stored in memory in the data structure of ATPDraw. The final lib-file is dumped to disk in the Result Directory with a name specified in the *Name* field (do not enter path or '.lib' as this is added automatically). The Result Directory is the same location as the ATP-file. As all lib-files (user specified, lcc and windsyn) are dumped to this directory file name conflicts can occur if components of different class have the same name. See the prefix options under *Tools|Options/View/ATP*.

Windsyn adds a TACS control module to the machine. In this model there are a number of predefined names not dummy declared. So watch out for unexpected name sharing. In all cases the machine number is added at the end of the node names as indicated with the 'n#' character. This can be a two digit parameter.

Setup of Windsyn in ATPDraw is done under *Tools|Options/Preferences*, as shown in Fig. 5.18.

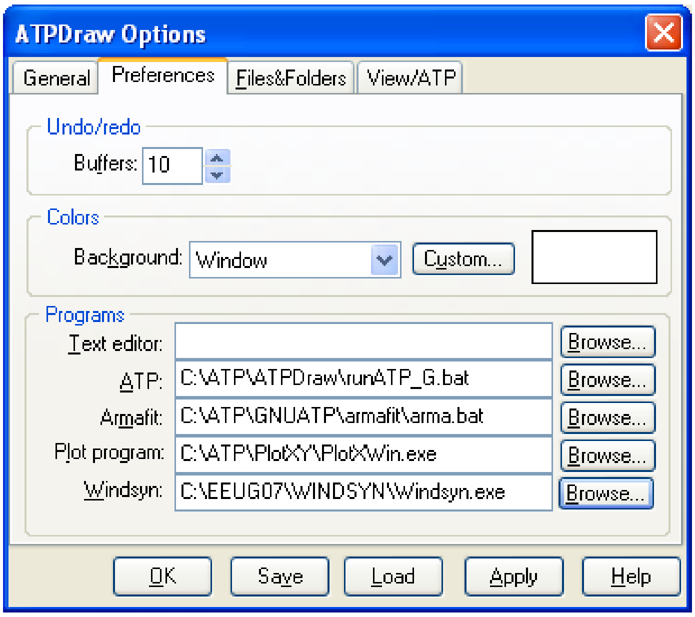


Fig. 5.18 - Windsyn (+ATP and Plot program) setup in ATPDraw.

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 and 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 those 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 under *Lines/Cables* | *LCC* item in the selection menu, as shown in Fig. 5.19. 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. 5.20. The user specifies the number of phases (and the number of cables) under the *System type* group. This choice will directly influence the grounding conditions in cable systems. The icon adapts setting of overhead line/single core cable/enclosing pipe and the number of phases.

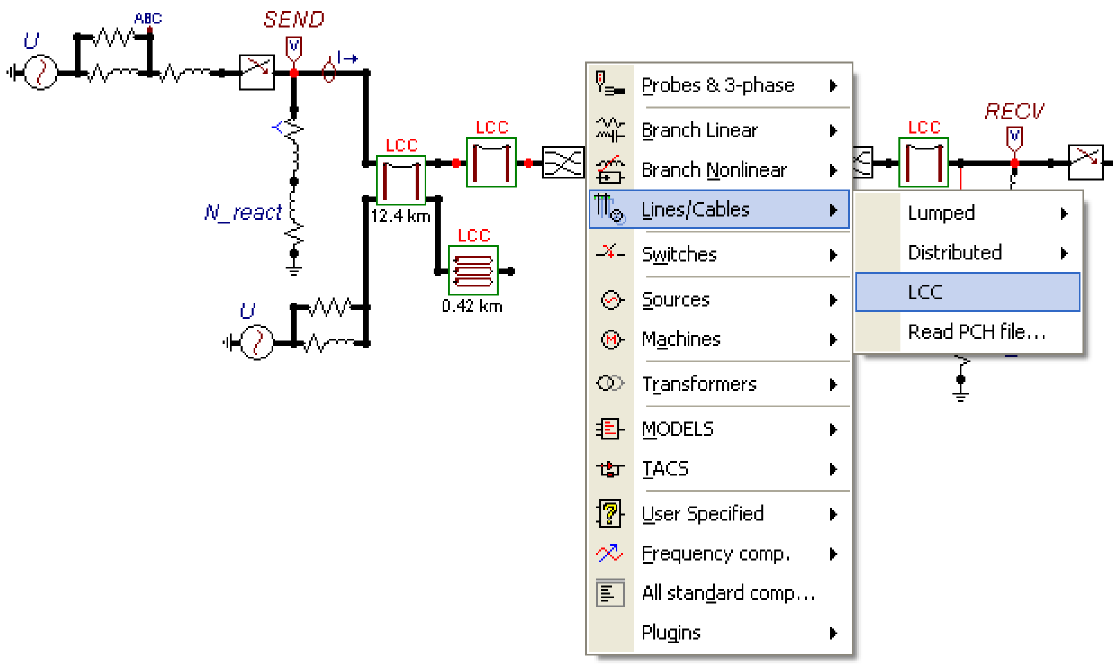


Fig. 5.19 - Selecting a line or cable and connecting the LCC object to the rest of the circuit.

When the required data are specified the user can close the dialog by clicking on *OK*. 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...*). You have to give a name to the component and if you click on the *Run ATP* button you will be asked to confirm the name. You do not have to specify path or extension as all data is stored in the Result Directory (same as the ATP file). If more than one component share the same name they are forced to be equal and the data is copied to the duplicates. When you click on *OK* you are warned about this as shown in Fig. 5.21. If you click on Yes the data of the current component will be copied to the other component with the same name. This cannot be undone directly, but you can undo the edit of the current component. If you then reopen it the old data will be copied to the other duplicates.

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 stored internally in memory and dumped to the Result Directory on demand. If something goes wrong in the generation of a electrical model an error message appear as shown in Fig. 5.22. Typical problems are missing or incorrect data. You can inspect the intermediate files in the Result Directory (c:\atpdraw\atp in this case). File with extensions `.dat` (LINE/CABLE CONSTANTS or CABLE PARAMETER file), and `.pch` (result that is transformed into a `.lib` file) and the same name as the line/cable component should be present.

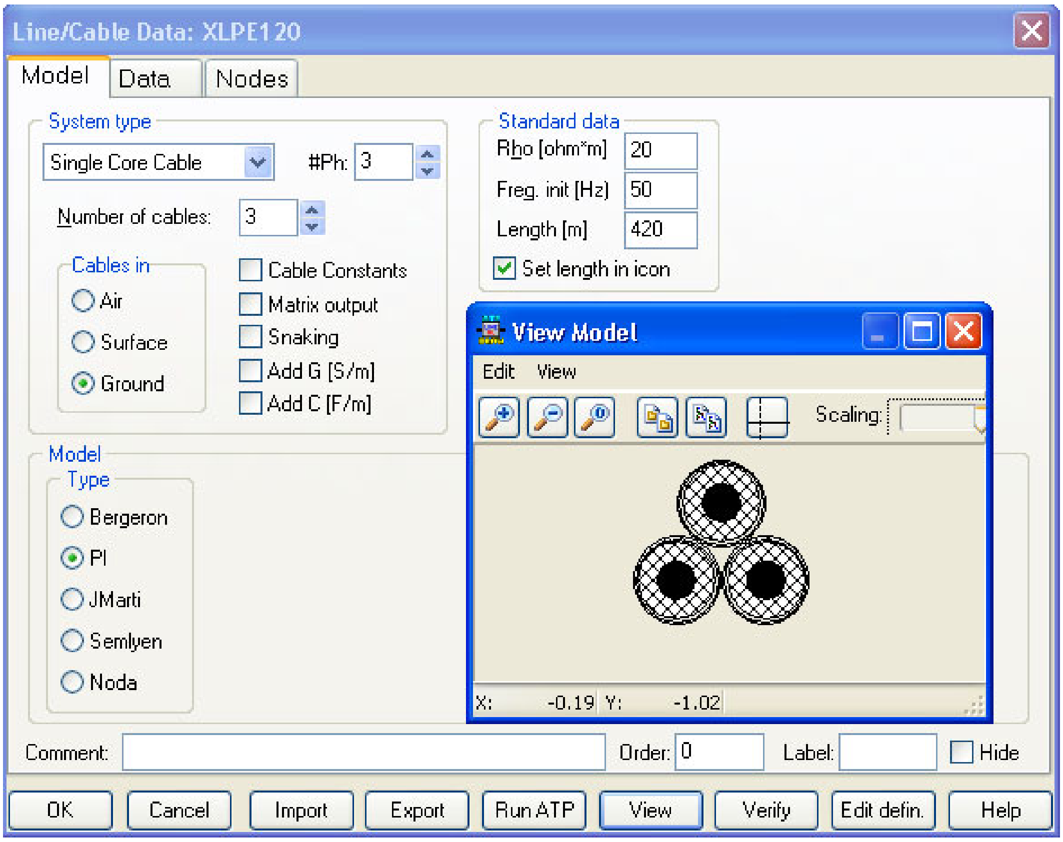


Fig. 5.20 - Line/Cable dialog box: Model specification. *View*- feature.

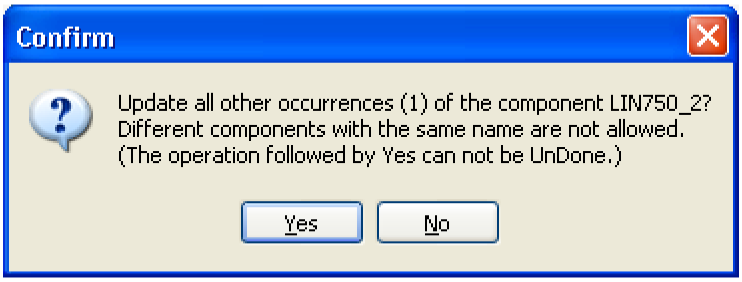


Fig. 5.21 – Duplicate line&cable components.

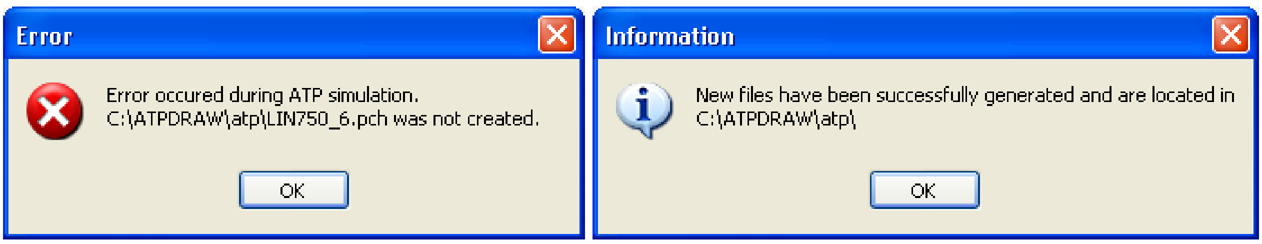


Fig. 5.22 – Model generation messages.

The data is stored internally in memory and the user can choose to export this data to an external library (typically the `/LCC` folder) by clicking the *Export* button. This data file is on a binary format and have extension `.alc`. You can click the *Import* button to load external data from disk. The Line&Cable component can also be copied between project as all other components. Clicking on the *View* button, displays the cross section of the line/cable as shown in Fig. 5.20. The phase numbers (with zero as ground) can be displayed in a red color via *View|Numbering*. 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.

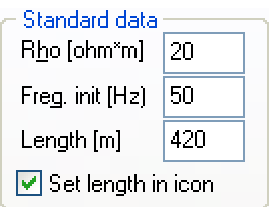
When creating a Noda line/cable model the Armafit program is executed automatically to create the required lib-file. The *Armafit command* is specified under *Tools | Options / Preferences*. The batch file `runAF.bat` is distributed with ATPDraw.

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

The *Line/Cable Data* dialog of Fig. 5.20 really consists of three pages: *Model* page, *Line* or *Cable* page and *Node* page. 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]. Set length as a text in icon option.

Fig. 5.23 - 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. Specify the number of phases in the *#Ph* combo box.

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.

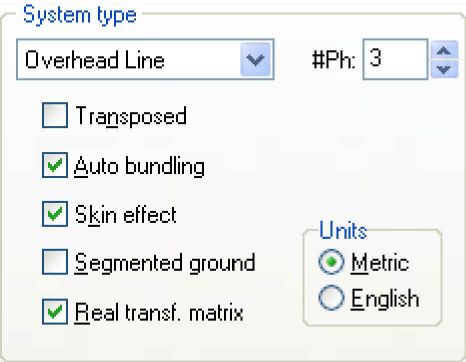


Fig. 5.24 - System type options for overhead lines.

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*:

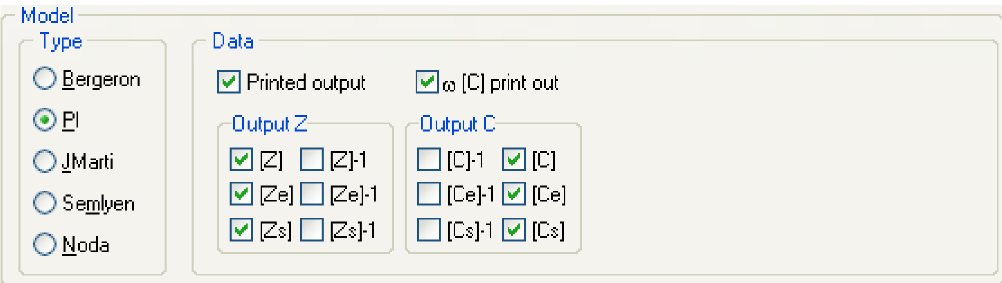


Fig. 5.25 - Optional settings for PI line models.

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].

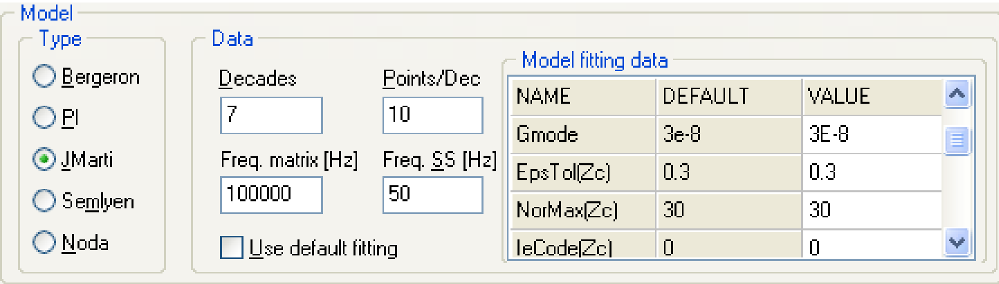


Fig. 5.26 - Parameter settings for the JMarti line model.

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].

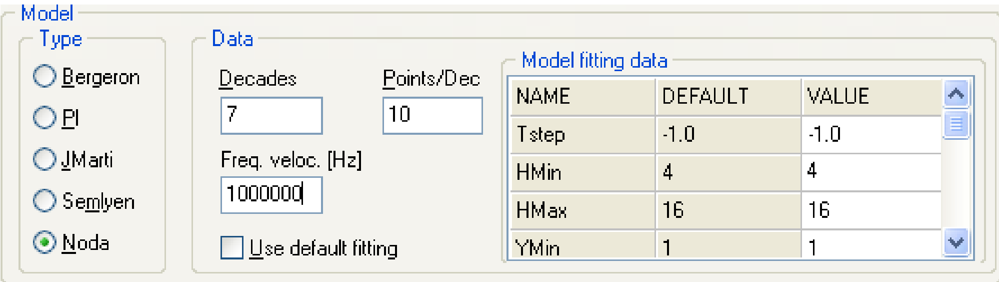


Fig. 5.27 - Parameter settings for the Noda line model.

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.

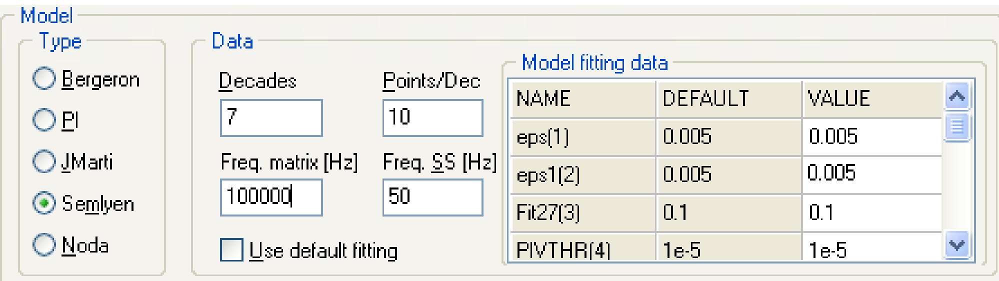
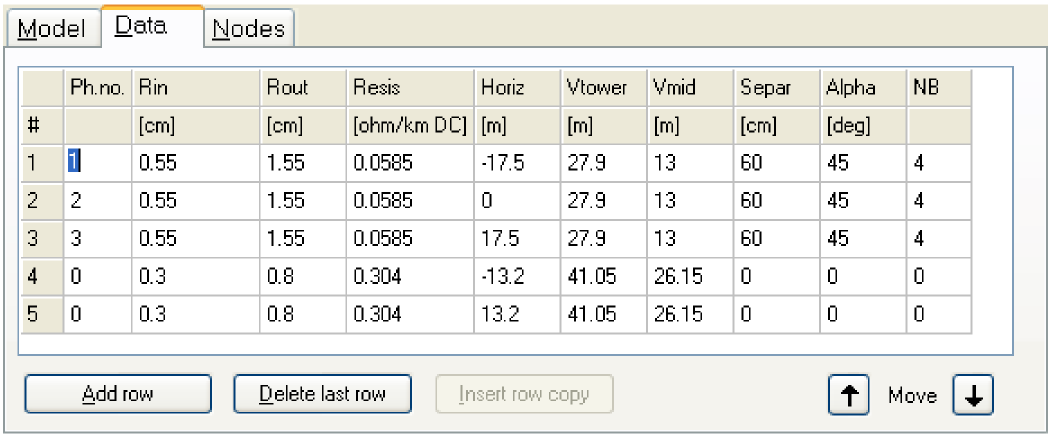


Fig. 5.28 - Parameter settings for the Semlyen line model.

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. 5.29. 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.



#	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. 5.29 - 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. 5.24). 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. 5.24):

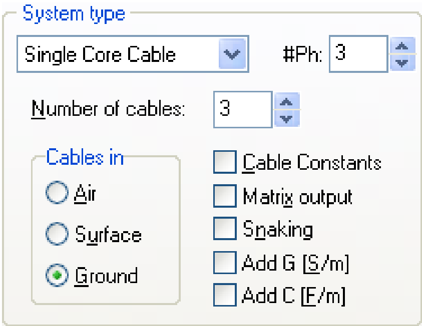
Separ: Distance between conductors in a bundle (cm or inch)

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. Specify the number of phases in the *#Ph* combo box.



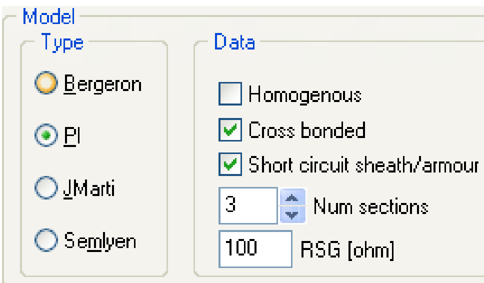
Cables in: Select if the cables are in the air, on the earth surface or in ground.
Number of cables: Specify the number of cables in the system.
Cable constants: Selects between Cable Constants and Cable Parameters option. If checked, the additional conductance and capacitance option will be switched 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. 5.30 - 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. 5.31 - 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*. Selecting *View|Numbering* will show the phase number in red color (0=grounded). The cables will be sorted internally according to the sequence rule of ATP; the cable with most conductors comes first. To avoid confusion and mismatch between expected phase number and conductors the user should try to follow this rule also in the Cable/Data dialog. The *Nodes* page allows the user to rearrange the phase sequence.

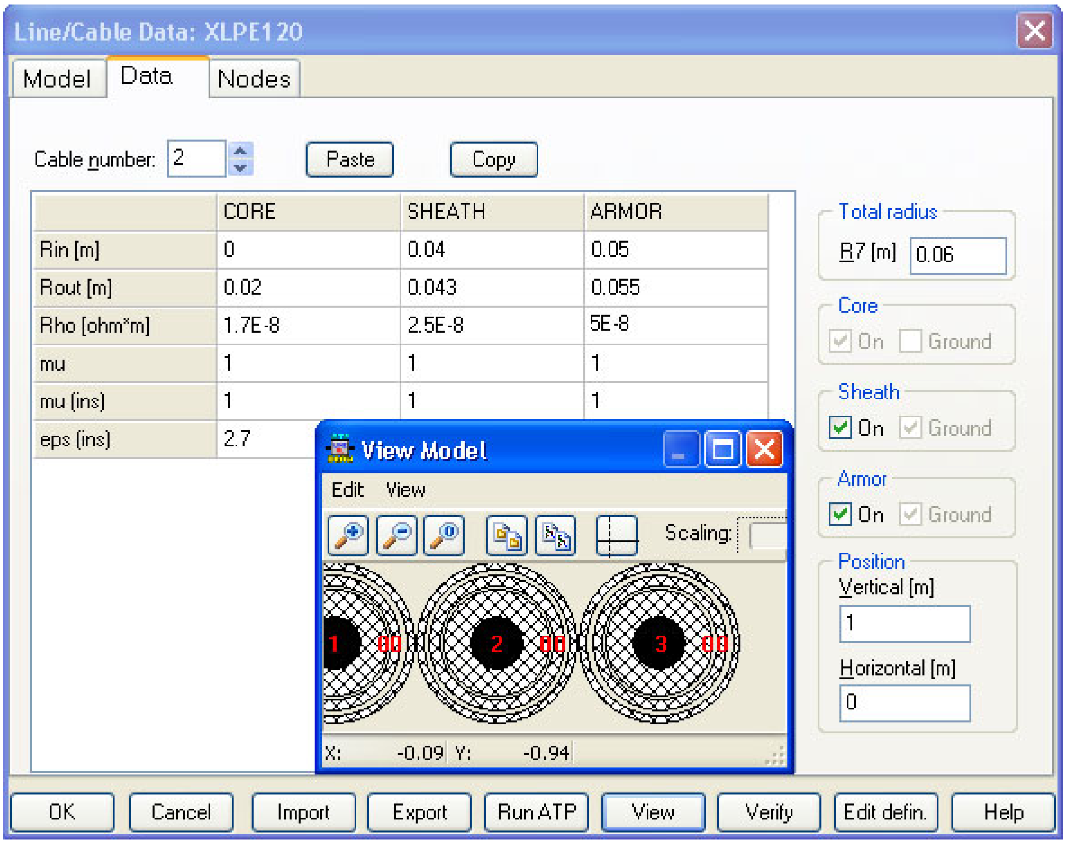


Fig. 5.32 - Cable Data dialog box for a 3-phase SC type cable system.

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. 5.31). For cables with enclosing pipe, the following *Pipe data* are required:

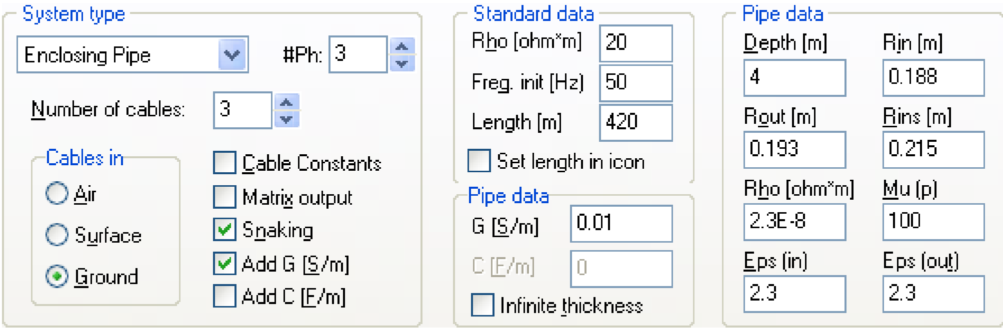


Fig. 5.33 - System type and Pipe data settings for an Enclosing Pipe cable.

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).

5.3.4 Node page settings

The *Node* page was introduced in ATPDraw version 5.3. Normally, the user does not need to specify anything on this page. It gives, however, access to the node names of the LCC component and offers the user to assign conductor numbers to the nodes. Conductor numbering can be desirable for cables since ATP requires a special sequence in this case; first comes the cores, then the sheaths then the armors. The cables with most conductors must be numbered internally in ATP as the first cable. To avoid too much confusion the user should also try to follow this rule. For overhead line the user specifies the conductor number directly in the data grid and there should be no need to alter this.

A cable system consisting of 3 single core cables with sheaths and a fourth ground wire will as default receive an "unexpected" phase sequence. The core of the three cables will be numbered 1-2-3, then the ground wire will be numbered 4, and finally the three sheaths will be numbered 5-6-7. This does not fit well with the 3-phase layout used for this 7-phase system. The core of the cabes will all be a part of IN1/OUT1-ABC, but then the ground wire will become IN2A/OUT2A, the cable sheaths 1 and 2 will be IN2B/OUT2B and IN2C/OUT2C and the third cable sheats will be connected to the single phase nodes IN3/OUT3. To let the ground wire be connected to the single phase node the conductor sequence 1-2-3-5-6-7-4 can be assign in the grid.

The *View* module has a *Number* feature that displays the conductor numbers.

5.4 Verification of the Line/Cable model performance

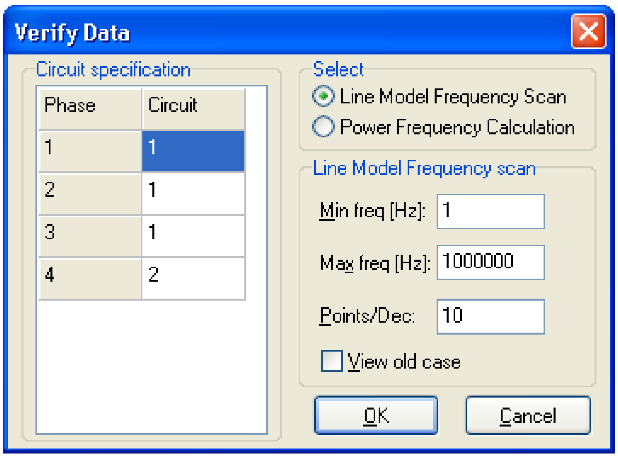
A line or cable model can be verified in two different ways. Internally in the Line/Cable dialog there is a Verify module that supports both a frequency scaning option and a power frequency calculation. Externally under ATP|Line Check there is a module that enables the user to select several sequential line section (including transposition) and perform power frequency calculations of series impedance and shunt admittance. This modele is better for long lines.

5.4.1 Internal Line/Cable Verify

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. 5.34 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.



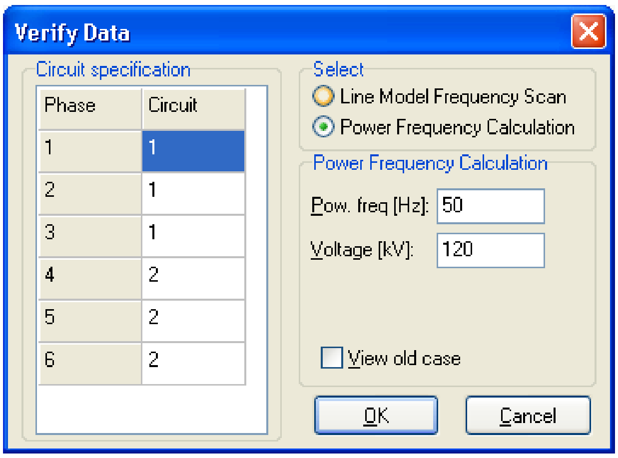


Fig. 5.34 - Frequency range specification for the LMFS run (left) and selecting the line voltage and system frequency for the PFC run (right).

a) <u>Select LMFS:</u> 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. 5.36 for the 4-phase JMarti line-model specified in Fig. 5.35.

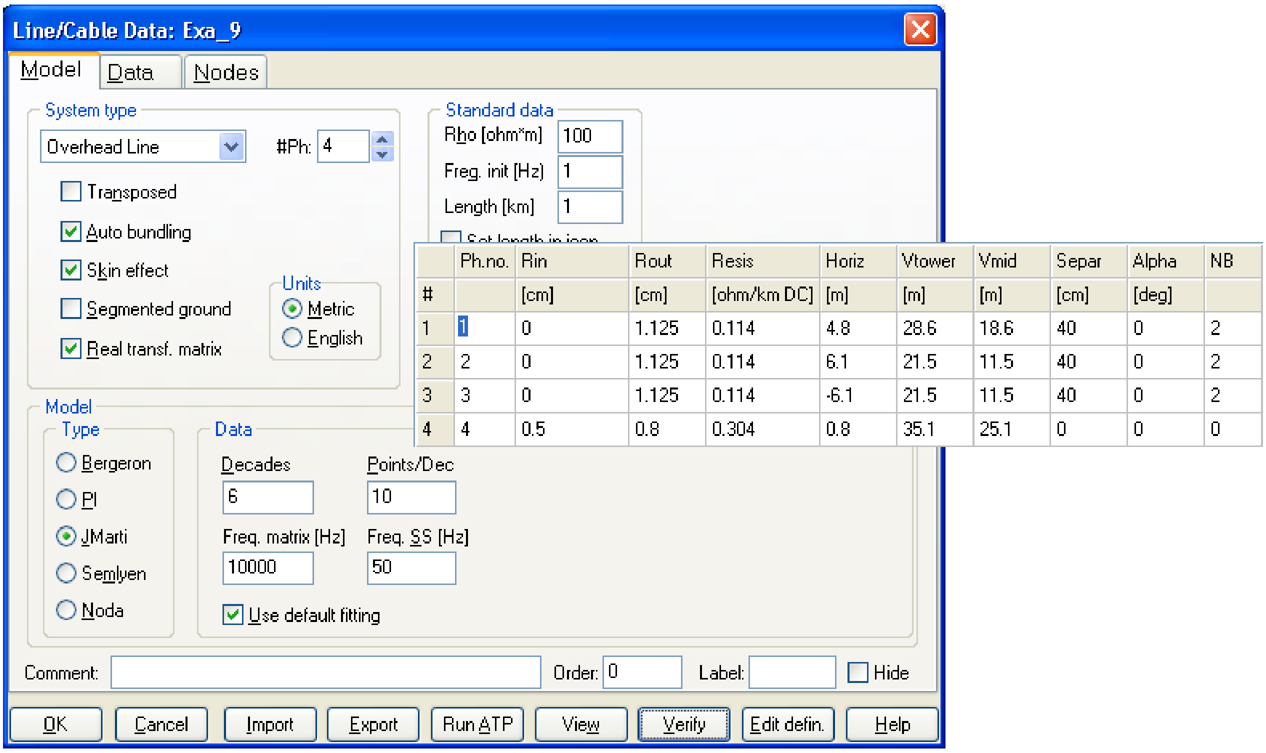


Fig. 5.35 - Specification of a 4-phase JMarti line model.

In Fig. 5.36, 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:

<u>Zero-sequence:</u> 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.

<u>Positive sequence:</u> 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.

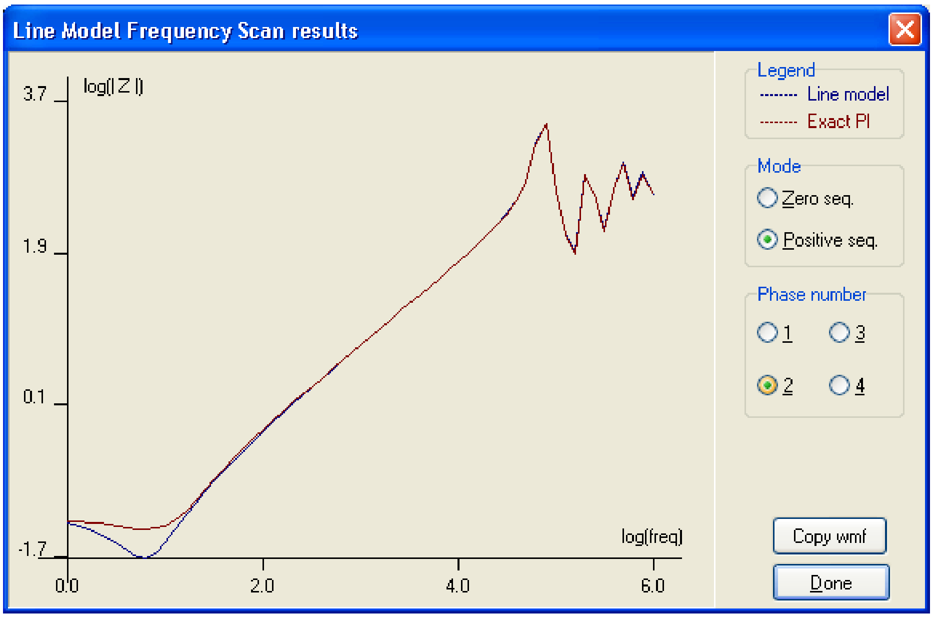


Fig. 5.36 - Verifying a JMarti line model 1 Hz to 1MHz. Model is OK for *f* > 25 Hz.

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. 5.37 .

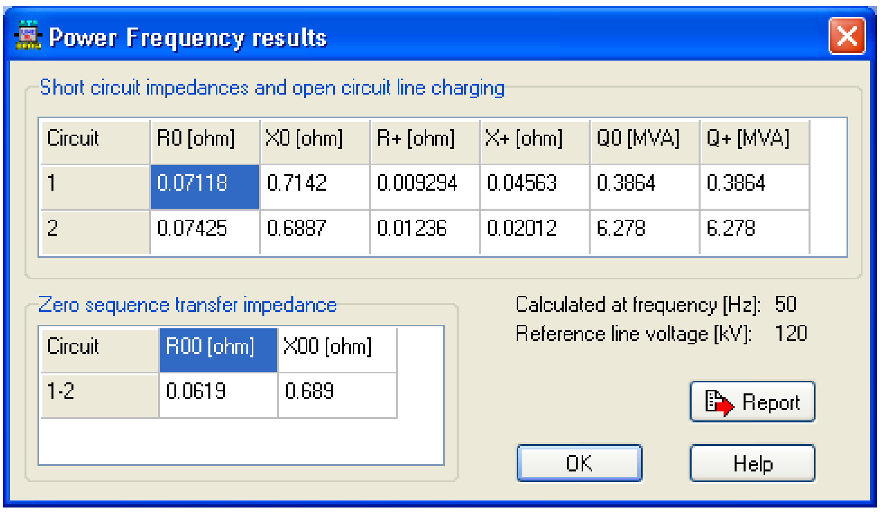


Fig. 5.37 - Results of the PFC run.

If the user clicks on *Report* the content in the string grids of Fig. 5.37 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.

5.4.2 External Line Check

First, the user selects the line he wants to test and then clicks on *ATP|Line Check* as shown in Fig. 5.38. Then the input/output selection dialog box shown in Fig. 5.39 appears.

The LineCheck feature in ATPDraw supports up to 3 circuits. ATPDraw suggests the default quantities. The leftmost nodes in the circuit are suggested as the input nodes, while the rightmost nodes become the output. The circuit number follows the node order of the objects. For all standard ATPDraw components the upper nodes has the lowest circuit number. The user also has to specify the power frequency where the line/cable is tested. Finally, the user can check the *Exact phasor equivalent* button which will result in a slightly better results for long line sections.

When the user clicks on OK in Fig. 5.39 an ATP-file (`/LCC/LineCheck.dat`) is created and ATP executed. For a 3-phase configuration 4 sequential data cases are created (Z+, Y+, Z0, Y0) while for a 9-phase configuration 24 cases are created (Z11+, Y11+, Z110, Y110, Z12…, Z22…, Z13…, Z23…, Z33…), since symmetry is assumed. Finally the entire LIS-file is scanned. The calculated values are then presented in result window as shown in Fig. 5.40. The user can switch between polar and complex coordinates and create a text-file of the result. The mutual data are presented on a separate page. The unit of the admittances is given in Farads or Siemens (micro or nano) and the user can scale all values by a factor or by the length.

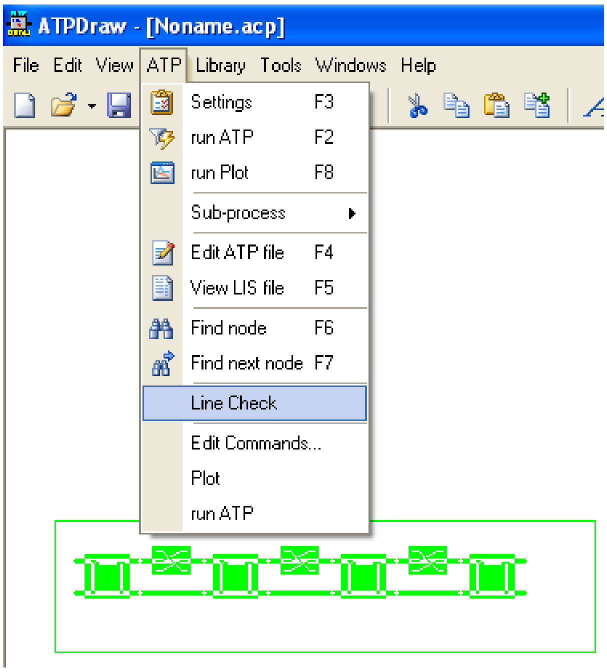


Fig. 5.38 – Select a line/cable sequence

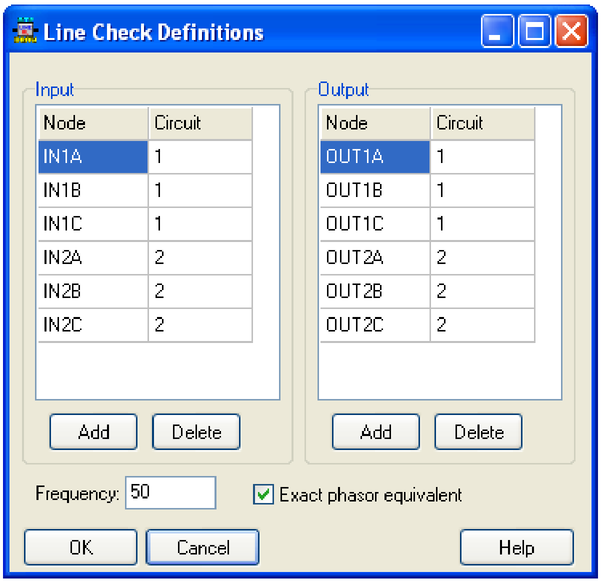


Fig. 5.39 – Specify inputs and outputs

The series impedances are obtained by applying 1 A currents on the terminals and the output ends are grounded (the other circuits are left open and unenegized). For mutual coupling, 1 A is applied at both circuits. On the other hand the shunt admittances are obtained by applying a voltage source of 1 V at one terminal leaving the output end open. For mutual coupling, 1V is applied at one circuit while a voltage of 1E-20 is applied at the other.

Special attention must be paid to long lines and cables. This applies in particular to PI-equivalents. Usage of Exact phasor equivalent is recommended, but is no guarantee of success. No attempt is made in ATPDraw to obtain a better approximation since the line/cable system to be tested in general is unknown. The mutual coupling in the positive sequence system is in symmetrical cases very small and vulnerable to the approximations made.

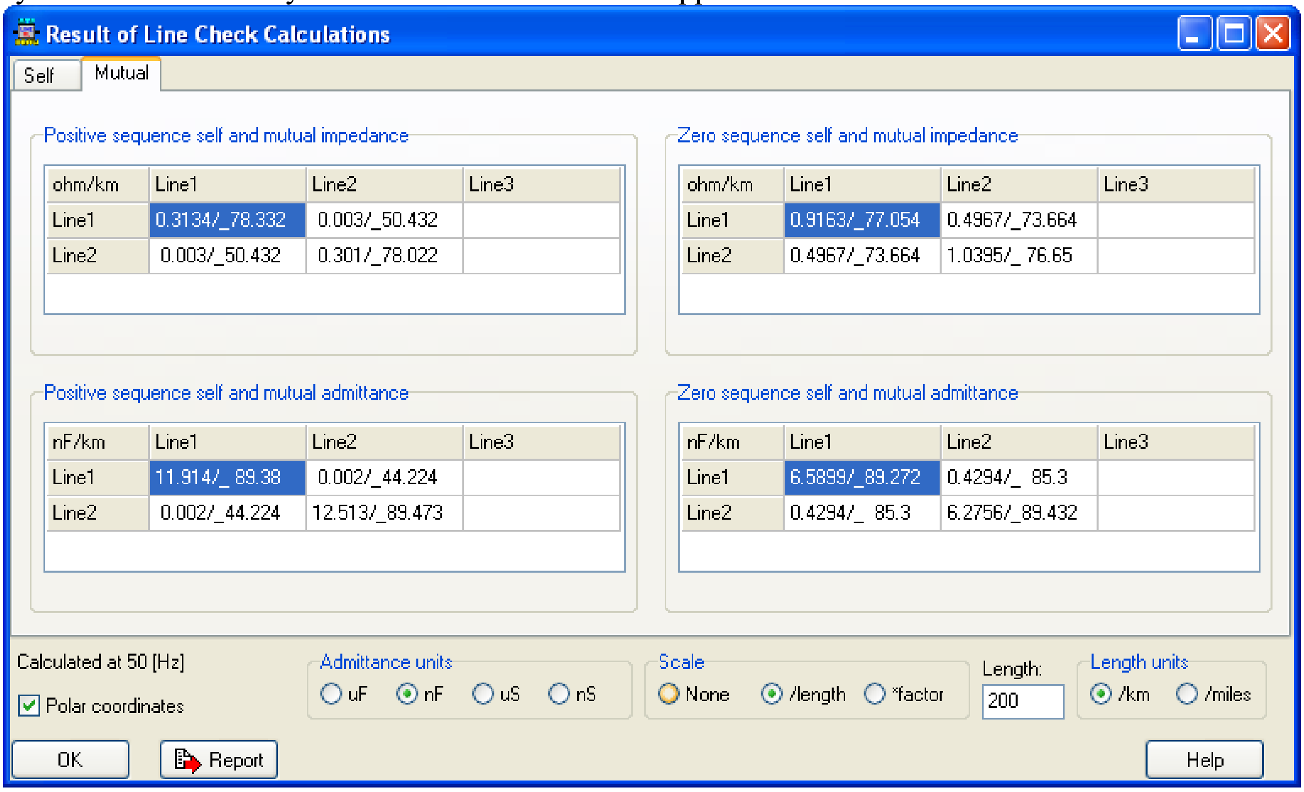


Fig. 5.40 – Presentation of the results.

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.

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`). There can thus not be any expressions in the USE section. Creating a new Model in ATPDraw can follow two approaches:

1. The automatic approach. Select the *Models|Default model* or open an existing `.mod` file and let ATPDraw take care the component definitions with icon and node connections. This is the best approach if the Model is supposed to change during the study.
2. The manual approach. Select *Models|Files mod/sup* and choose a pre-existing support file (accompanioned with a compatible `.mod` file). This is the best choice if the Model will not change much (inputs/outputs fixed) during the study and the icon and node locations is crusial.

The new MODELS object created in this chapter is part of the ATPDraw's example file `Exa_14.adp`. In this example the harmonic content of the line current on the 132 kV supply side of an industrial plan using a 24 pulse AC/DC converter is calculated by MODELS.

5.5.1 The automatic approach

Add a new Model to your circuit by selecting *MODELS|Default model* from the selection menu. A simple Model will appear with an empty dialog box shown as shown in Fig. 5.41. Now, click on the *Edit* button and type in your model description, import a text from file with *File|Import* or paste in a text from the Windows clipboard. Anyway, this is the hard part of the process. In the listing below you will noticed that two indexed outputs are defined absF and angF as [1..26]. This will result in 26-phase nodes (which is the maximum allowed). The low index has to be 1 and the upper must be a number less or equal to 26. Indexed data is also allowed and these are then split in x[1], x[2] etc. The maximum number of data is 64 and the maximum umber of inputs plus outputs is 32.

Click on *Done* when the edit process is completed. ATPDraw will then examine the Model description and identify the Input/Output/Data declarations. If the number of input or outputs have changed the icon is recreated. Inputs are positioned on the left side and Outputs on the right side (from top to bottom). A message box then appears as shown in Fig. 5.42. Typically you should choose not to edit the file, but if you choose *Yes* the *Edit definitions* dialog appears where you can relocate the nodes and change the icon. This might be a tricky process though. Anyway you can whenever click on *Edit definitions* an do this job later on. If yoy click on No, you will return to an updated Component dialog box as shown in Fig. 5.43.

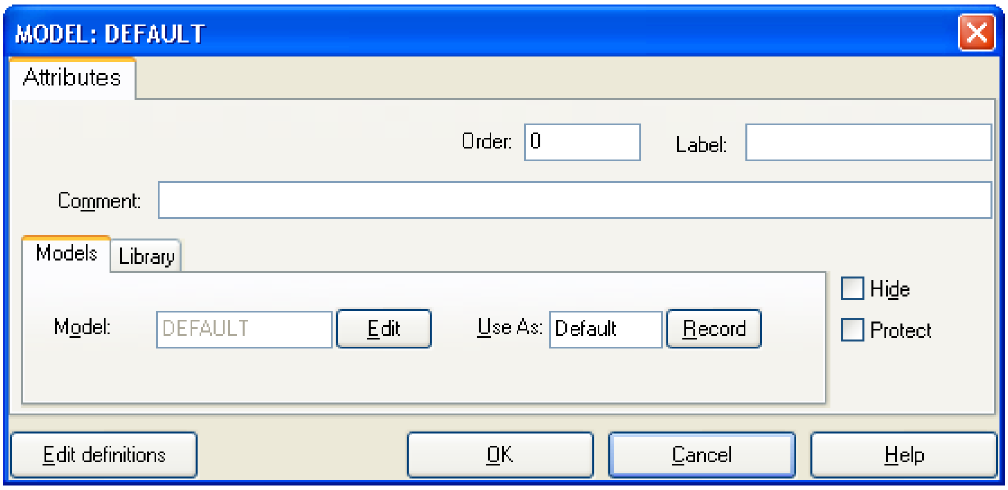


Fig. 5.41 – Component dialog of the Default Model.

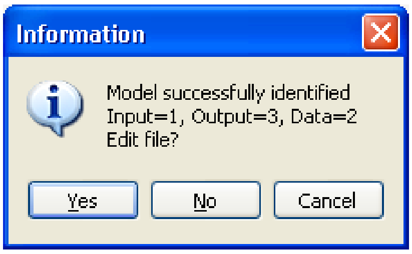


Fig. 5.42 – Identification of the Model text.

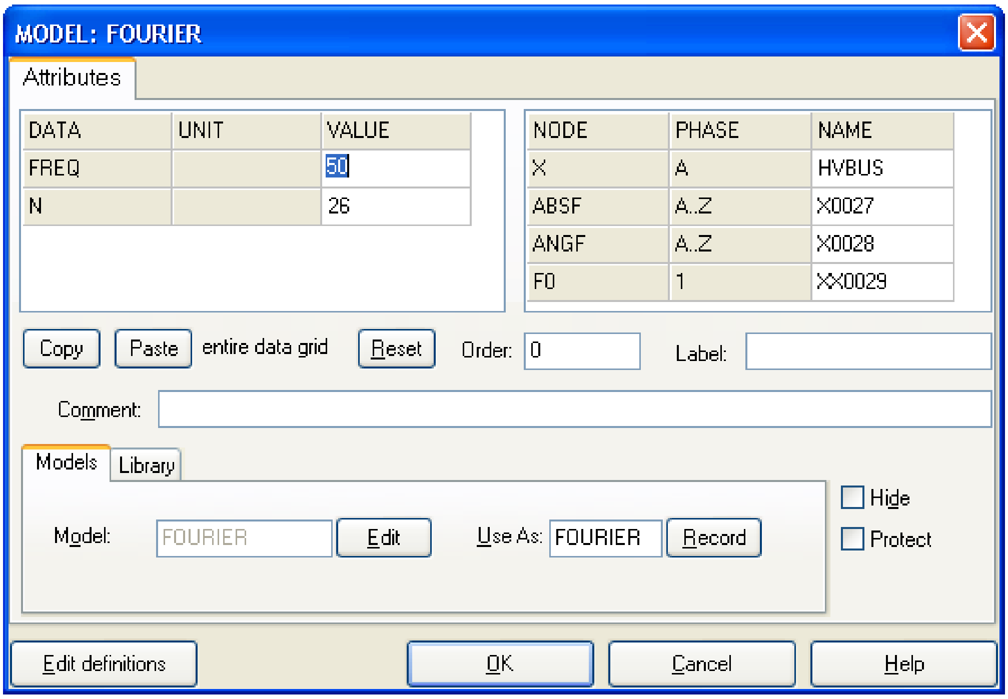


Fig. 5.43 – Component dialog of the FOURIER model.

In the Models section in Fig. 5.43 you must also specify the *Use As* name for `USE model AS model_name` statement of MODELS. *Record* of local variable is also available in this section.

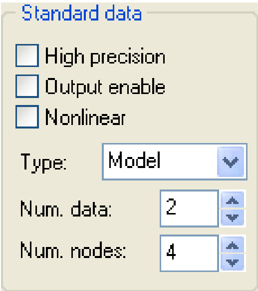
The actual model file describing the calculation of harmonics is shown below:

```
MODEL FOURIER
INPUT X                     --input signal to be transformed
DATA  FREQ {DFLT:50}        --power frequency
      n {DFLT:26}            --number of harmonics to calculate

OUTPUT absF[1..26], angF[1..26],F0 --DFT signals
VAR    absF[1..26], angF[1..26],F0,reF[1..26], imF[1..26],i,NSAMPL,OMEGA
       D,F1,F2,F3,F4

HISTORY
  X {DFLT:0}

DELAY CELLS DFLT: 1/(FREQ*timestep)+1

INIT
  OMEGA:= 2*PI*FREQ
  NSAMPL:=1/(FREQ*timestep)
  F0:=0
  FOR i:=1 to 26 DO
    reF[i]:=0
    imF[i]:=0
    absF[i]:=0
    angF[i]:=0
  ENDFOR
ENDINIT
EXEC
  --window X?
  f1:=delay(X,(NSAMPL+1)*timestep,1)
  f2:=delay(X,NSAMPL*timestep,1)
  f3:=delay(X,timestep,1)
  f4:=X
  F0:=F0+(f4+f3-f2-f1)/(2*NSAMPL)
  FOR i:=1 to n DO
    D:=1/(i*PI)*((f4-f2)*sin(i*OMEGA*T)-(f3-f1)*sin(i*OMEGA*(T-timestep))
       +(f4-f3-f2+f1)/(timestep*i*OMEGA)*
       (cos(i*OMEGA*T)-cos(i*OMEGA*(T-timestep))))

    reF[i]:=reF[i]+D

    D:=1/(i*PI)*(-(f4-f2)*cos(i*OMEGA*T)+(f3-f1)*cos(i*OMEGA*(T-timestep))
       +(f4-f3-f2+f1)/(timestep*i*OMEGA)*
       (sin(i*OMEGA*T)-sin(i*OMEGA*(T-timestep))))

    imF[i]:=imF[i]+D
    absF[i]:=sqrt(reF[i]**2+imF[i]**2)
    IF imF[i]<1E-10
    THEN
      angF[i]:=0
    ELSE
      angF[i]:=atan2(imF[i],reF[i])
    ENDIF
  ENDFOR

ENDEXEC
ENDMODEL
```

5.5.2 The manual approach

You can create an external support file in two ways. Either by click on *Edit definitions* is the *Component dialog* box of your Model and then click on *Save As* (preferable to the `/MOD` directory). This will simply give you a copy of your Model component. The other way is to go via *Library|New object|Model sup-file* and create a support file from scratch. Both these options use the *Edit definitions dialog*. The end result is a support file that you load via *MODELS|Files* *(sup/mod)*.

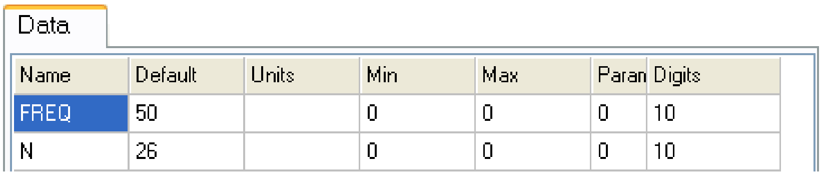
The manual approach requires that you have the mod file finished, or at least you need to know the number and name of all input, outputs and data. Enter the *Library* menu and select the *New objects|Model sup-file*. This menu item will perform the *Edit definitions* dialog. In the *Standard data* field, you specify the size of the model: number of nodes and number of data as shown in Fig. 5.44.



The *FOURIER.MOD* text has four nodes (1 input + 3 outputs) and two data, (FREQ, *n*), so you must enter 4 and 2 in the *Num.* fields.

Fig. 5.44 - 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. 5.45. 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*.



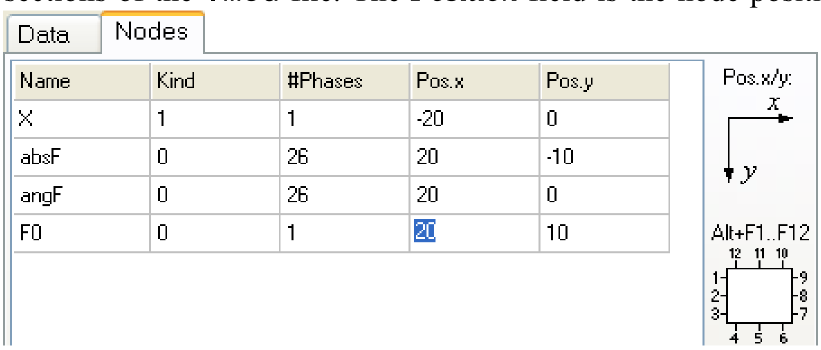
Data

Name	Default	Units	Min	Max	Paran	Digits
FREQ	50		0	0	0	10
N	26		0	0	0	10

Fig. 5.45 - Specify Data parameters.

Param is set to 0, which means that variable text string can not be assigned to the data value. *Digits* 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. 5.46. 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 (Alt+F1..F12 are short keys), but other positions (-120..120) is possible. The *Kind* value specifies the input/output type of the node. Number of *#Phases* must be set to match the array size of the input/outputs.



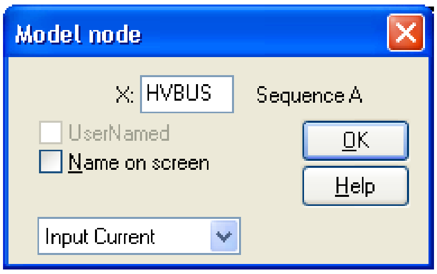
Name	Kind	#Phases	Pos.x	Pos.y
X	1	1	-20	0
absF	0	26	20	-10
angF	0	26	20	0
F0	0	1	20	10

Fig. 5.46 - Specifying Node attributes.

<u>`Supported Kind values for MODELS objects are:`</u>

```
    0:    Output node.              3:   Switch status input node.
```

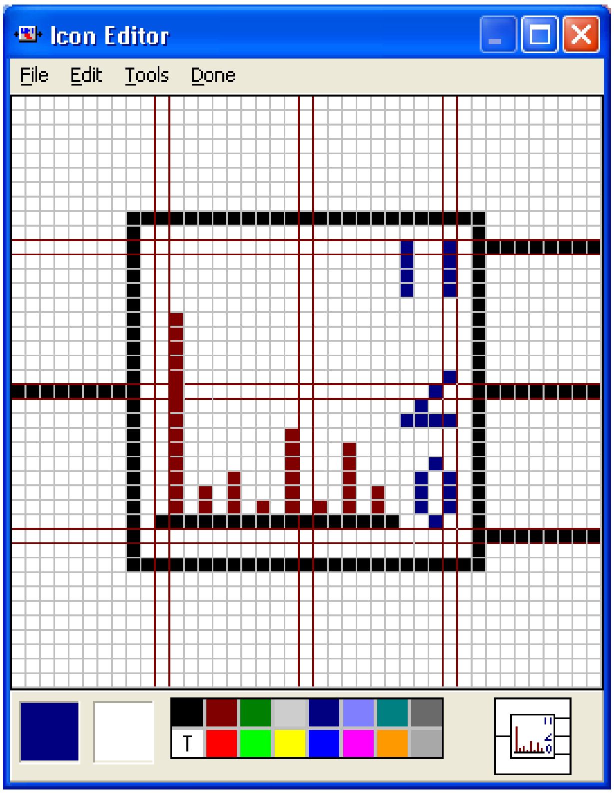
```
1:     Current input node.      4:    Machine variable input node.
2:     Voltage input node.      5:    TACS variable (tacs).
6:     Imaginary part of steady-state node voltage (imssv).
7:     Imaginary part of steady-state switch current (imssi).
8:     Output from other model. 9:    Global ATP variable.
```



The *Kind* parameter of model object nodes can be changed later in the Node dialog box (input field *Type*), as shown in Fig. 5.47. This window appears when the user clicks on a Model node with the right mouse button.

Fig. 5.47 - Model node dialog box.

Note! If a model output is used as input for another model, the model, which produces the output must be `USE`d before the use of the model that is supplied with this output. This can be done by specifying a lower *Order* number for the model with output signals and selecting the *Sorting by Order* 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 really 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. In this case Bitmap icon style is chosen. 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. 5.48 - 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 *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 `FOURIER.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 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
INPUT
M0001A {i(HVBUSA)}
OUTPUT
  X0027A
  X0027B
  ...
  X0027Z
  X0028A
  X0028B
  ...
  X0028Z
  XX0029
------------------------------------------
MODEL FOURIER
...
Description of the model.
Complete copy of the
FOURIER.MOD is pasted here.
...
ENDMODEL
------------------------------------------
USE FOURIER AS FOURIER
INPUT
  X:= M0001A
DATA
  FREQ:=        50.
  N:=        26.
OUTPUT
  X0027A:=ABSF[1]
  X0027B:=ABSF[2]
  …
  X0027Z:=ABSF[26]
  X0028A:=ANGF[1]
  X0028B:=ANGF[2]
  …
  X0028Z:=ANGF[26]
  XX0029:=F0
ENDUSE
```

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. The selection of internal variables is done by clicking the *Record* button in Fig. 5.43. This will bring of the Record dialog shown in Fig. 5.49. The available variables (VAR+OUTPUT) is shown in the list to the left. Select the desired variable and click the >> button. The *Record* field to the right is a free format text field that allows you to easily edit the AS name. In the case of indexed variables you also need to specify the index as well (shown as reF[5]). Remove the variable from the *Record* list by the << button. The Outputs from a Model can alternatively be recoreded with the Model Probe as shown to the right in Fig. 5.49.

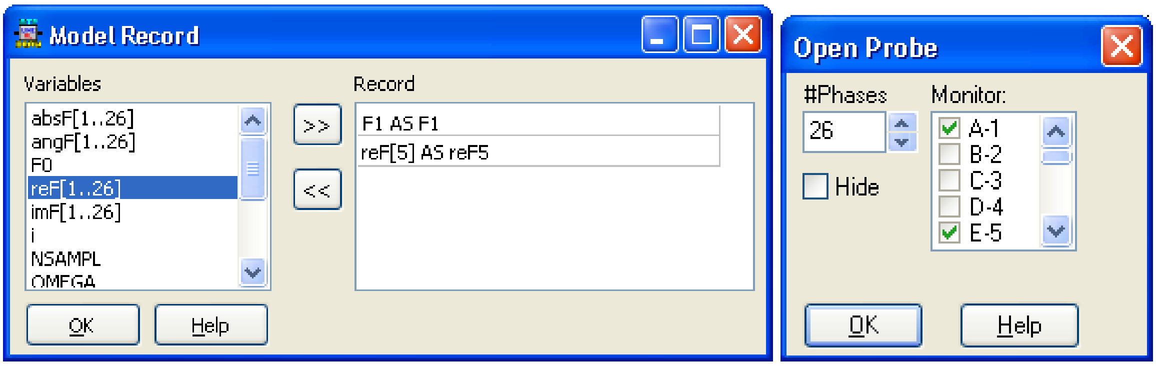


Fig. 5.49 - Record of model variables. Right: Models Probe connected to Output node.

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.

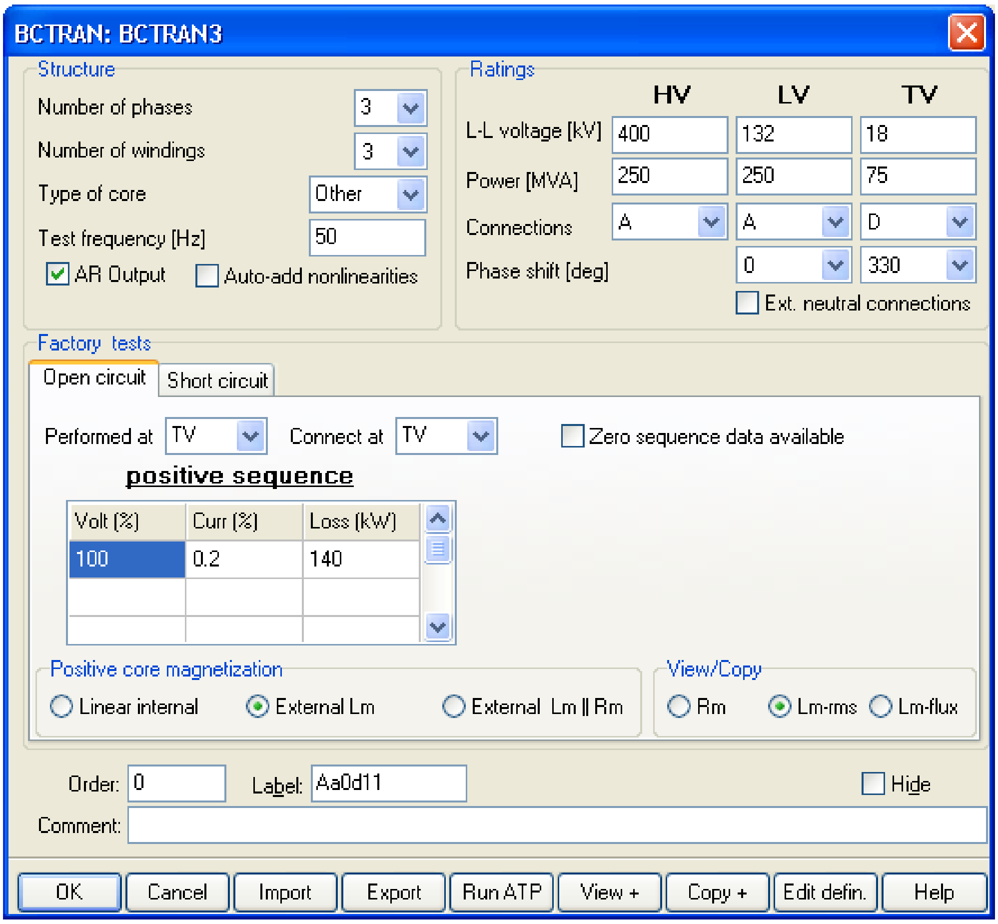


Fig. 5.50 - The BCTRAN dialog box.

Fig. 5.50 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 tests*, the user can choose either the *Open circuit* test or the *Short circuit* test.

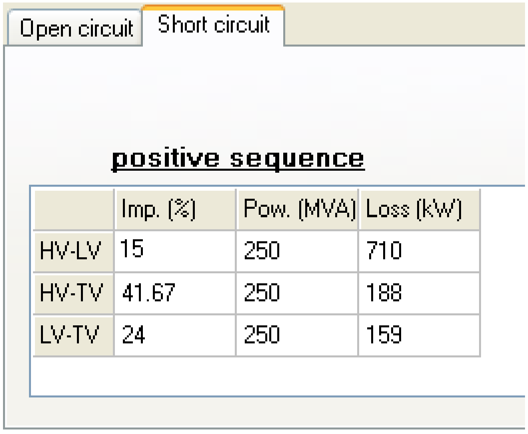
Under the *Open circuit* tab the user can specify where the factory test has been performed and where to connect the excitation branch. In case of a three winding transformer one can choose between the HV, LV, and the TV winding. Normally the lowest voltage is preferred, but stability problems for delta-connected 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.



The *Short circuit* data can be specified as shown in Fig. 5.51. 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. 5.51 - 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.

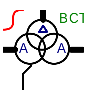
3. Short circuit test data

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/(Vri^2)*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/(Vri^2)*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 data specified in Fig. 5.50 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 Hybrid Transformer, XFMR

This component called XFMR was added to version 4.2 of ATPDraw in June 2005. The model is then improved in several steps by extensive debugging. The XFMR component is an implementation and extension of the work performed by Prof. Bruce Mork at Michigan Tech and his co-workers Fransisco Gonzalez-Molina and Dmitry Ishchenko. This project called "Parameter Estimation and Advanced Transformer Models for EMTP Simulations" was sponsored by Bonneville Power Administration. A series of report documents this work and his here used as references MTU4, MTU6 and MTU7. The implementation in ATPDraw was also funded by BPA.

5.7.1 Overview

The principle of the modeling is to derive a topologically correct model with the core connected to an artificial winding on the core surface. Individual magnetizing branches are established for the yokes and legs dependent on their relative length and area (normally a value within limited range). A key feature is that magnetization is assumed to follow the Frolich equation which is fitted to Test Report data (using the Gradient Method optimization). This improves extreme saturation behavior since linear extrapolation above the Test Report data is avoided. The leakage inductance is modeled with an inverse inductance matrix (A-matrix), following the BCTRAN approach as documented in the Theory Book p. 6.21. Shunt capacitances and frequency dependent winding resistance is also considered.

The transformer model consists of four parts (as shown in Fig. 5.52) :

- Inductance. Leakage reactance - > A-matrix
- Resistance. Winding resistance -> R(f)
- Capacitance. Shunt capacitance - C-matrix
- Core. Individual magnetization and losses for legs and yokes.

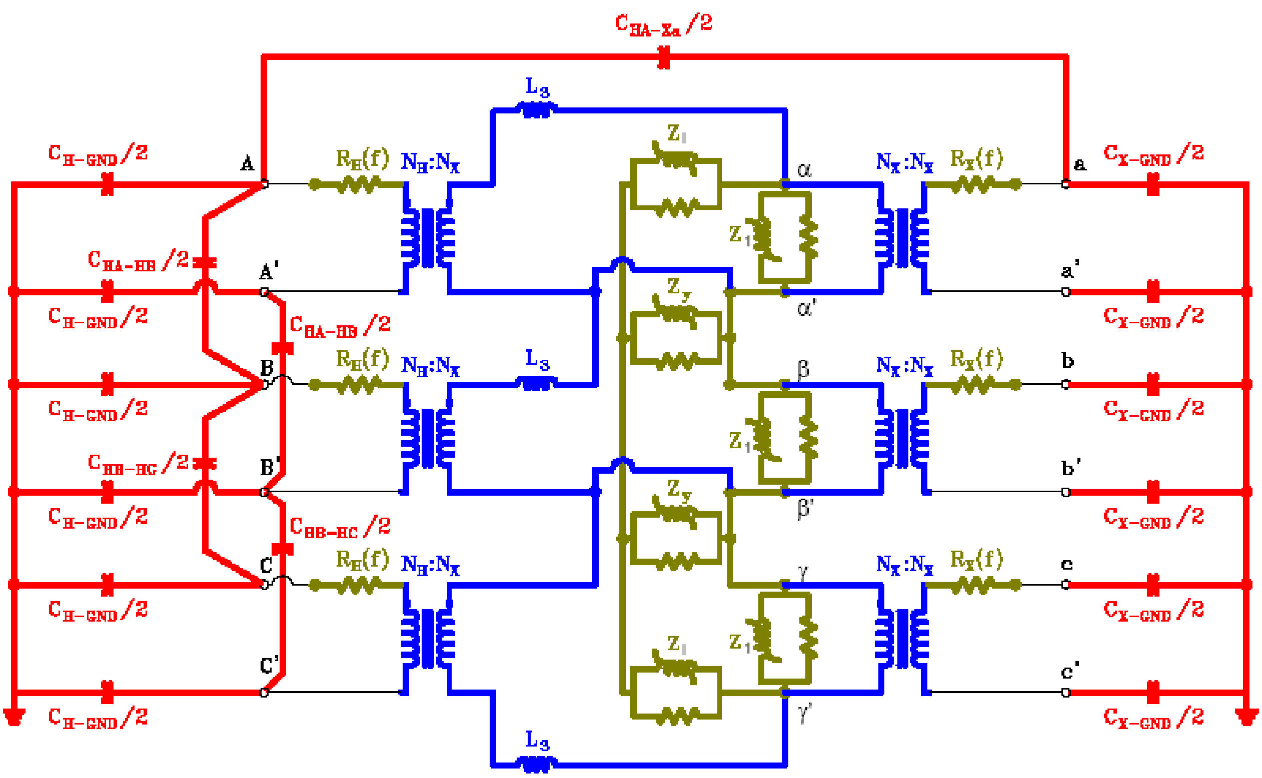


Fig. 5.52 - Duality model for a 3-phase, two-winding transformer from MTU4.

The XFMR component support three sources of data:

- Design parameters. Winding and core geometry and material properties.
- Test report. Standard Test Report data like in BCTRAN. Capacitances and frequency dependent resistance added.
- Typical values. Typical text book values based on transformer ratings. Be careful with this as both design and material properties have changed a lot the last decades.

The overall node structure of the XFMR component in the final ATP file is shown in Fig. 5.53.

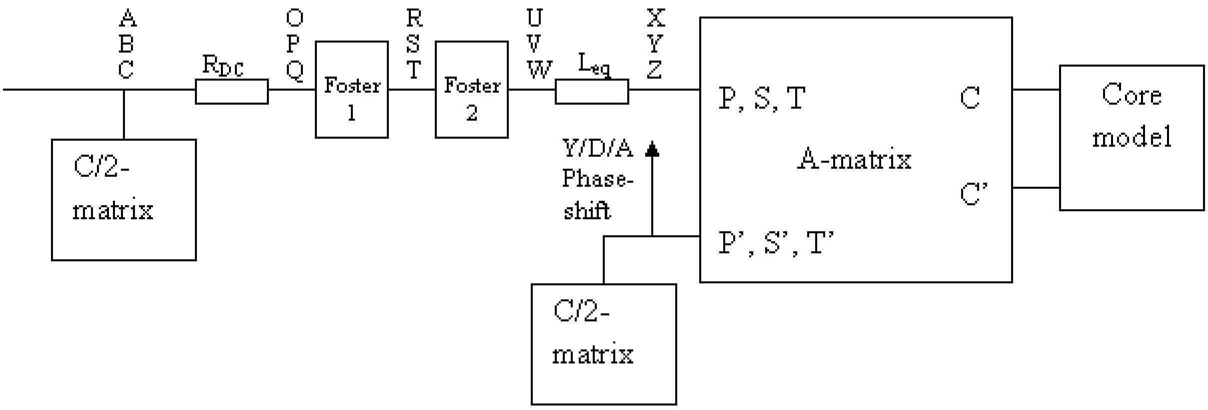


Fig. 5.53 - Node structure in the ATP-file.

This component can be connected as any other component in the circuit with the following exceptions.In both these cases switches should be used in order to maintain unique node names.

- It is not legal to ground nodes directly
- It is not correct to connect several components to the same bus.

5.7.2 XFMR dialog box

The advance Hybrid Transformer component, XFMR, is found under *Transformers* in the selection menu. The model support 3-phase transformers with two or three windings coupled as Wye, Delta, or Auto. All possible phase shifts are supported. Triplex (single phase bank), 3- and 5-legged stacked cores and shell form cores are supported. The dialog box is shown in Fig. 5.54.

All the input fields in the dialog box change dynamically with the user's selection of the number of windings and type of core.

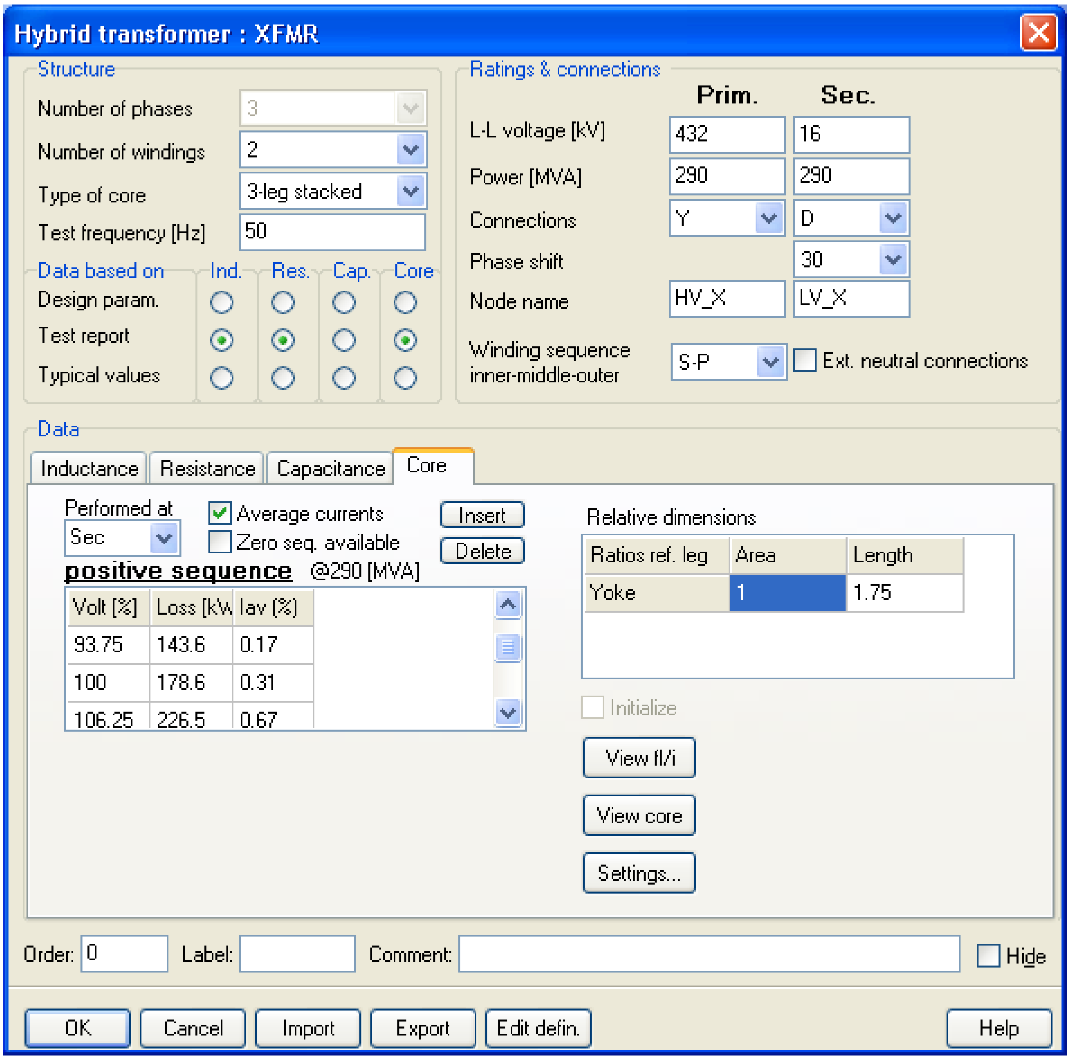


Fig. 5.54 – The XFMR component dialog box

When the user presses *OK* the electrical model data (A and C matrices, R, and Core) are calculated and stored internally. The calculation of the core model might take up to one minute and a progress bar is shown (the user can press ESC to stop the calculation). The data can be exported (*Export* button) to an external library file (`.xfm`) for later import, but also copied between projects. Using the Import button it is possible to load a priviously created `.xfm` file.

Twelve radio buttons are available under *Structure* and *Data based on* that enables the user to set the source of data individually for each part of the model. Click the right mouse button to omit the part completely (inductance can not be omitted). *Inductance*, *Resistance*, *Capacitance* and *Core*. Under *Type of core* the user can select the core configuration. Triplex (single phase bank), 3- and 5- legged stacked, and shell form cores are supported. The type of core will influence the structure and calculation process of the core model. A 5-legged core will have a saturation characteristic also for the outer legs, while in the case of a 3-legged core this is replaced by a constant inductance representing the zero sequence behaviour.

Under *Ratings & Connections* the user must specify the the line-to-line voltage in [kV], the rated power of the transformer [MVA] and the type of coupling and phase shift for each winding. These settings all refere to the Primary (P), Secondary (S), and Tertiary (T) notation. P is on the left side, S on the right side, and T on the top side of the transformer icon. There is no restrictions on the voltage levels here.

The phase-shift refered to the primary winding is specified in the drop down list. Only possible phase-shifts are listed. Other phase shift would require ZigZag couplings not supported here (use the Saturable Transformer component).

The sequence of the winding on the core leg is set in the combo box *Winding sequence*. This is used establish the artificial winding where the core should be connected. If this sequence is unknown then remember that the inner winding usually has the lowest voltage. When the *Ext. neutral connections* button is checked, all neutral points become 3-phase nodes that the user has to connect manually.

For design data the user must input the geometry and matrial data of the winding and core. For the core the user must choose a magnetic material. The list of available material data is very limited and only relatively new characteristisc are included. This means that a modeling of an old transformer using this approach would result in too low core losses. Uncertain aspects of the design data are the core losses and the zero-sequence data especially for 3-legged transformers.

For test report data ATPDraw has an embedded BCTRAN-like routine for calculation of the A-matrix and winding resistance R. The core model is established by fitting the measured excitation currents and losses. The user can specify 9 points on an excitation characteristic. Some *Insert* and *Delete* buttons are available. ATPDraw will also sort the points by increasing voltage level. If the current and core loss do not increase with voltage an error message is displayed.

For typical values some estimation is made based on textbook tables using the rated voltage and power. In the Typical data page there is a button *Edit reactances*, *Edit resistances*, *Edit capacitances*, or *Edit magnetization*. When the user check this button, ATPDraw calculates the typical values based on the rated quantities and display the typical values. The values are then locked. To update the values based on a new setting of rated values the user must uncheck the button. There are basically two levels of sophistication available.

- The default level requires no user input at all; the inductance, resistance, capacitance and core data is calculated based on typical values from tables. The user is allowed to specify a few data to improve the guessing; type of cooling for inductances (unknown=forced air), coupling factor for capacitances, and rated magnetic field intensity Bmax, loss density Pmax, and basic insulation level for core modeling. The user can examine the internally calculated data by checking an Edit button this also enables the second level. Once the button is checked the data are no longer updated when the rated voltage or power is changed.
- At the second level the user can directly specify the data.

Some buttons are available for viewing the winding and core design. If these buttons are checked a separate on-top window pops up with the information required to specify the input correctly. The Configuration image changes with the number and type of winding and the core type. The figures are fixed and are not scaled with the user specified dimensions.

Click on the *Settings* button on the core page to set some parameters for the core model. This will bring up the Advanced core settings dialog. An important setting is the #points in saturation; the internal core model based on the Frolich equation (2 or 3 parameter option) is fitted to the test report with a fast Gradient optimization method by minimizing the different between the measured and calculated rms currents. This is then converted to a piecewise linear characteristic (type 93 or 98 inductors) assuming a certain number of points. Type 96 hysteretic inductors are also supported, and in this case half the core loss is assumed to be hysteresis losses and the core loss is in general assumed to be proportional to the square of the flux density. Initialization is challenging for the type 96 inductors and ramping up the power supply with a controlled source might be necessary at lest for a 5-legged core. A very important parameter for inrush studies is the final slope inductance La. Design parameters are required here and $L_a = \mu_0 \cdot N^2 \cdot A_{leg} / l_{leg}$.

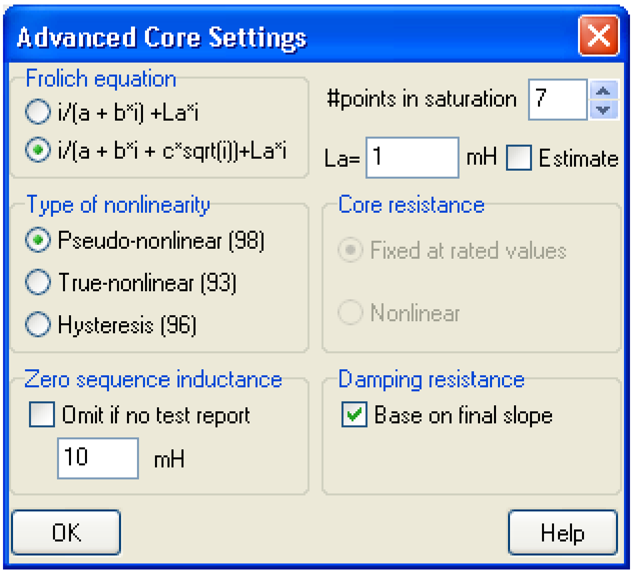


Fig. 5.55 – The Advanced core settings dialog.

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 [illegible], and the number of nodes in the range of 0 to [illegible]. 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 6.3 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 6.5.2 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. 5.56 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. 5.56. 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.

ATPDraw writes all three names of a 3-phase node in the `$INCLUDE` statement. In this example only the core name of the 3-phase node is expected on the argument list, because the phase identifiers `A-B-C` are added internally in the DBM-file. This option requires the *Internal phase seq.* checked box be selected in the component dialog box of the HVDC_6 object, as shown in Fig. 5.59. If it is selected, ATPDraw writes only the 5-character long core names in the `$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.

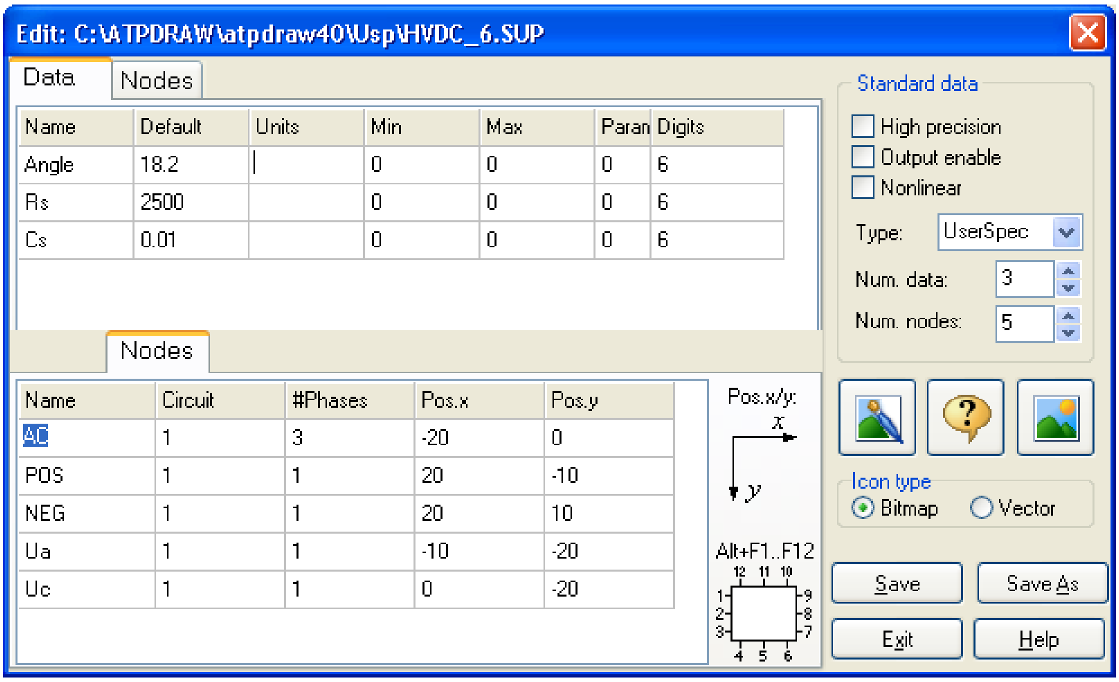


Fig. 5.56 - Properties of the new HVDC_6 object.

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. 5.57 shows an example file that can be associated with the user specified 6-phase rectifier bridge.

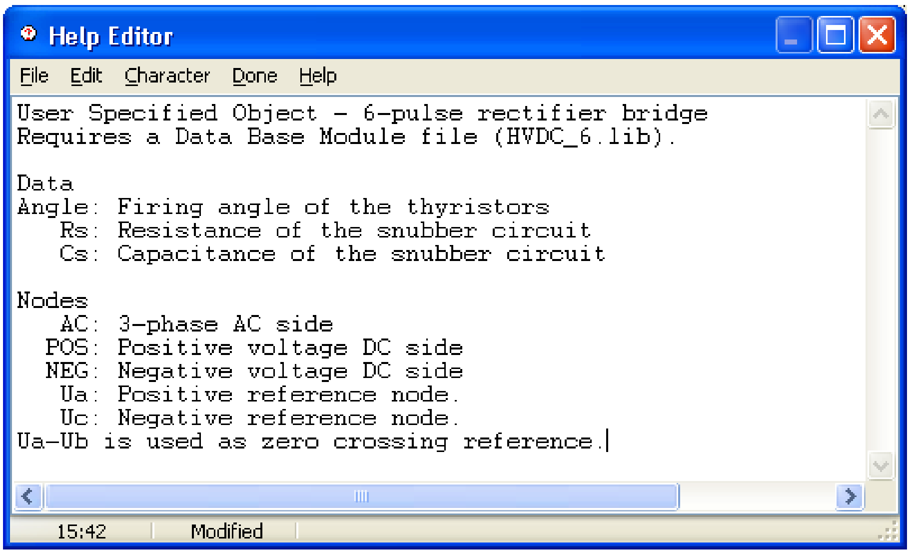


Fig. 5.57 - Help file of the HVDC_6 object.

Fig. 5.58 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. 5.56. The completed icon of the 6-pulse rectifier bridge is shown in Fig. 5.58.

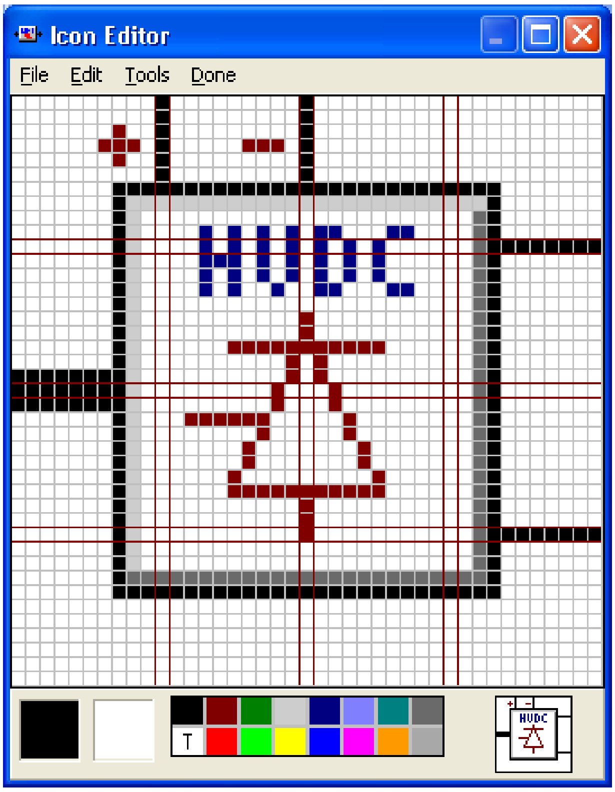


Fig. 5.58 - The icon associated with the new HVDC_6 object.

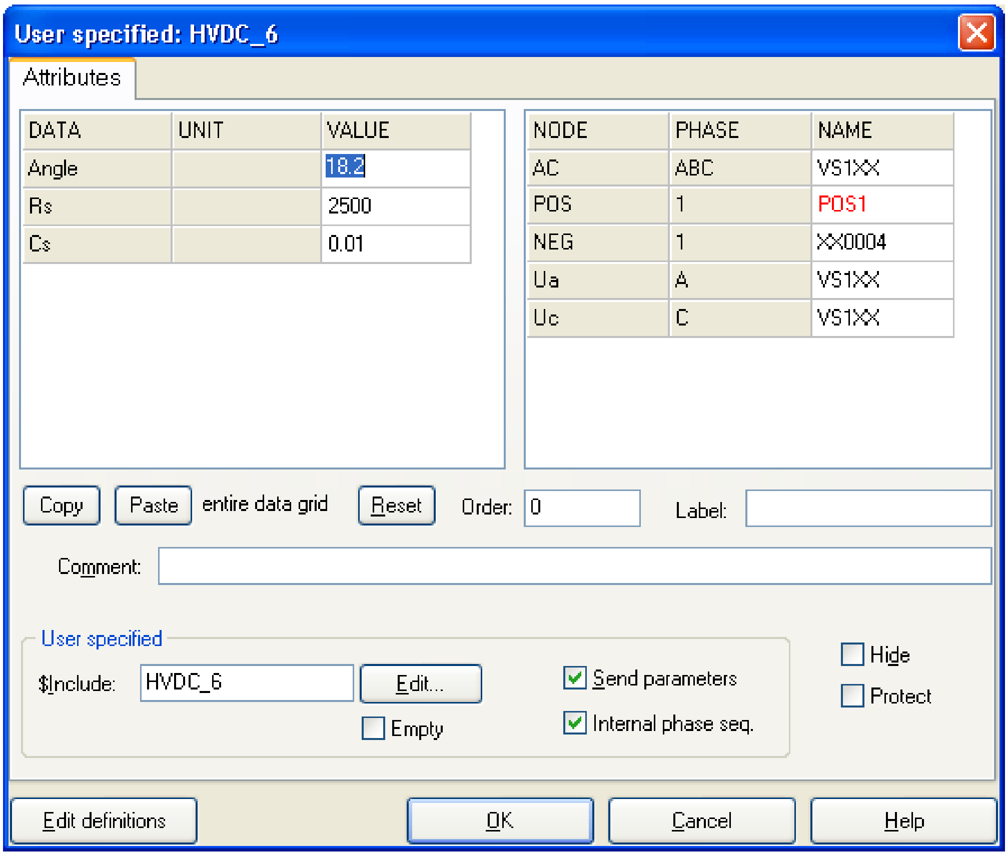


Fig. 5.59 - Component dialog box of the new user specified HVDC_6 object.

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. 5.59. 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.

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. 5.60.

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.

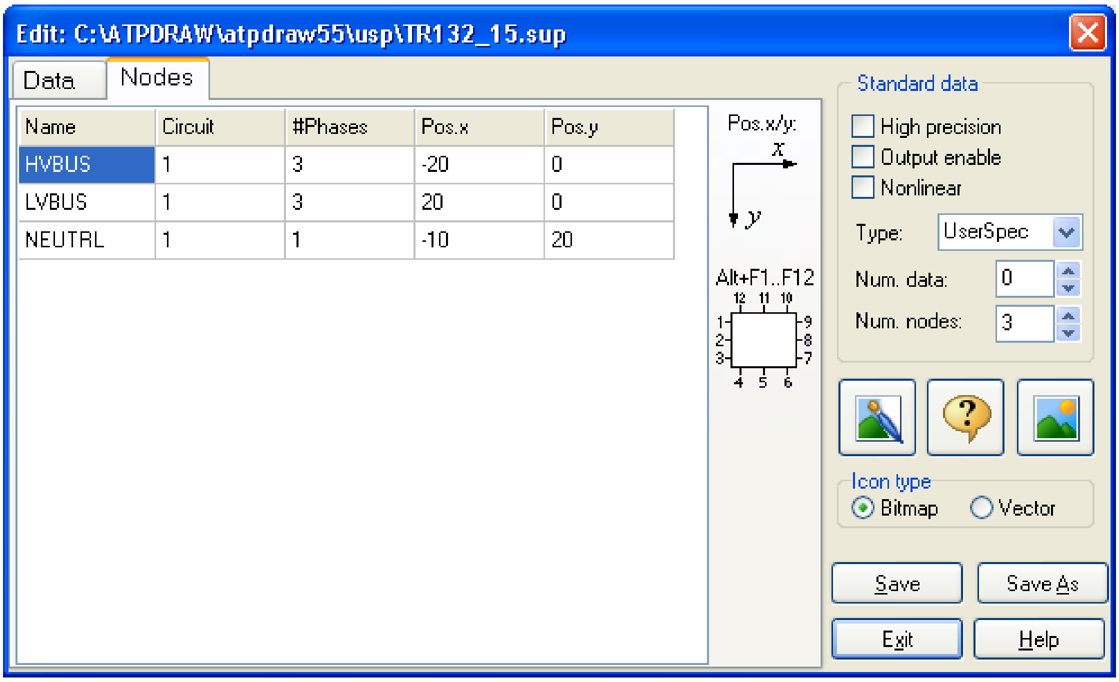


Fig. 5.60 – Creating support file for the new BCTRAN object.

The user specified components can be used in combination with the new grouping feature of ATPDraw as shown in Fig. 5.61. 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.

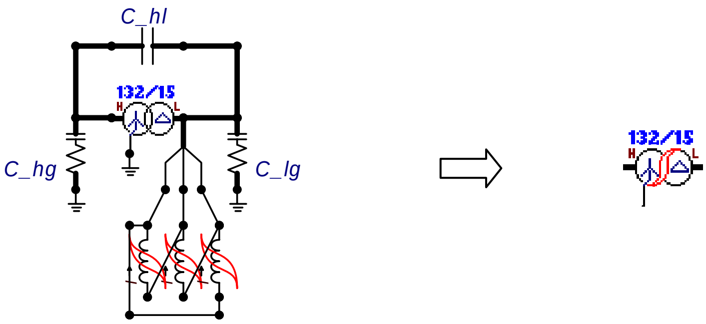


Fig. 5.61- Compressing the transformer model into a single object.

5.9 Vector graphic editor

In ATPDraw all icons of standard components are in vector graphic style. This enables better zooming and dynamic icon capabilities. A component can have either a bitmap or a vector icon, but not both. The building block of the vector graphic format is the Element (maximum 93). An Element has a Visible flag and can belong to a Layer, it is thus possible to easily turn on/off element as a respons to user settings. Further an element can either be a Shape or a Text. A shape can be of various standard Windows types (lines, rectangle, ellipses, poly-lines, polygons, arcs, pies, and Bezier curves), while a Text is simpler. A Shape can consist of maximum 255 points which is very beneficial for poly-lines, polygons and Bezier curves. The vector graphic editor has been developed from scratch utilizing an internal graphic format for fast drawings. The editor is shown in Fig. 5.62.

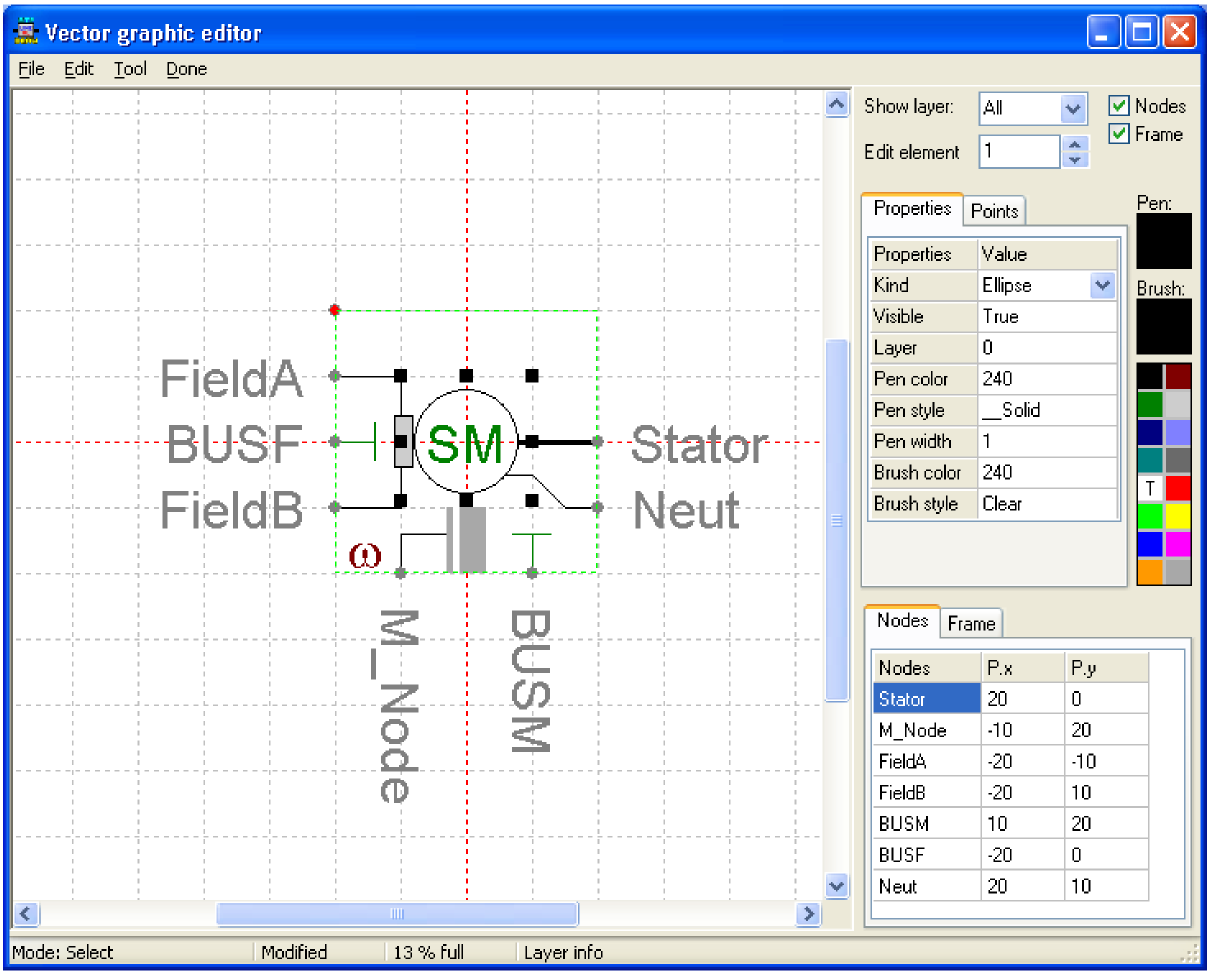


Fig. 5.62 – Vector graphic editor (400 % zoom).

An element can be selected by clicking in the icon window or by specifying the *Edit element* spin edit field to the top right. The selected element is shown with its properties below. In the *Properties* grid the pen and brush colors and styles can be selected. The colors are codes with numbers 0-255. The numbers 240-255 are used for the standard colors and lower values to match the closest possible color when selecting the full color palette. The present color of the pen (frame) and brush (fill) is shown by the squares to the right. A shortcut to the standard colors (240-255) is to click the palette to the right (r\left for pen, right for brush). This is the same is in the old Bitmap editor. In the Points grid the co-ordinates for the points are shown as well as rotation angle of rectangles and ellipses and rounding of rectangles.

Fig. 5.62 also shows the components *Nodes* and the *Frame*. These are turned on/off via the checkboxes in the very top right corner. The *Frame* is the selection area of the icon; mouse clicks inside this area in the circuit will select or open the component. A too large *Frame* will result in overlapping conflicts with other icons. The Frame is not changeable with the mouse; the user has to specify the coordinates in the *Frame* string grid. The External point drawn as red is used for branch output of some of the components. The Nodes are drawn as gray dots with their node names oriented relative to the Frame. The Node positions and name can be specified in the Nodes string grid. The nodes can also be moved with the mouse selecting *Tool|Move nodes*. The nodes have to be on the grid so the nodes are only moved in steps. The grid is also drawn in Fig. 5.62 with the red lines indicating the centre. The grid can be turned on/off via *Edit|Node grid*.

When the editing process is completed the user clicks on *Done*.

5.9.1 Properties

Fig. 5.63-Fig. 5.64 shows the properties grids. Most of the properties have combo boxes and pup-up dialogs attached as shown in Fig. 5.65 for selection of possible values.

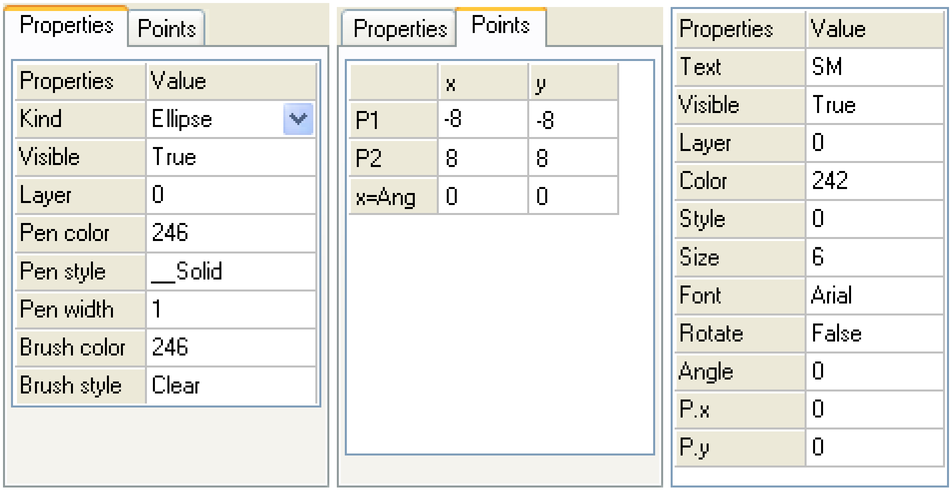


Fig. 5.63 – Properties grid. Left and centre: Shapes. Right: Texts.

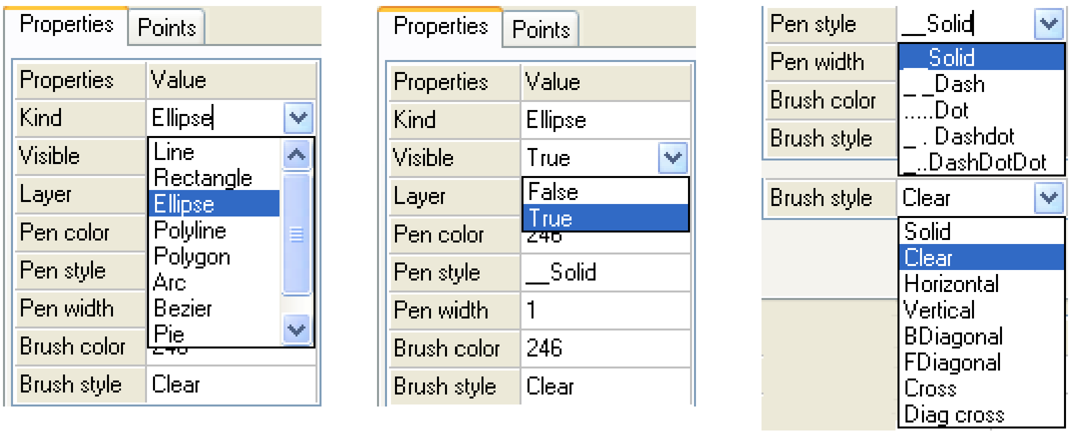


Fig. 5.64 – Shape properties alternatives.

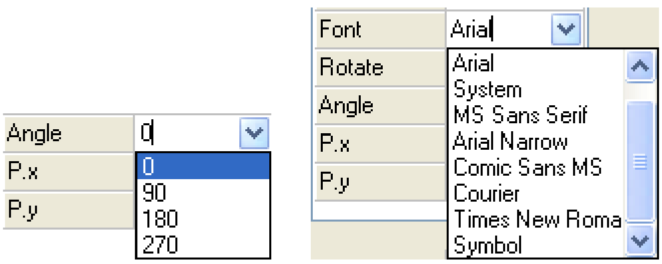
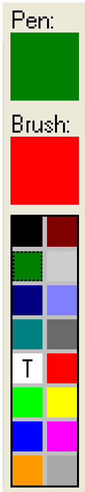


Fig. 5.65 – Text properties.

The *Shape* points are given in the co-ordinate system -128..127. The Text point P is specified in the centre of the text. The Node co-ordinates have to be rounded off to the nearest 10.

Colors are described by a code 0..255, where 240-255 are the old standard ATPDraw colors used in the bitmap styles icons. These colors are found in the color grid to the right. For the color properties there is a button available when pressed shows the basic color palette available in Delphi as shown in Fig. 5.66. The user can choose a color here which then is mapped to the closest color. The Custom color palette is the same as the standard ATPDraw colors. True 24 bit colors are not supported.

Fig. 5.66 – The Basic color palette.



Standard ATPDraw colors.

5.9.2 Editing: Selecting, moving, resizing and clipboard

An element is selected by clicking on it in the icon window. If the brush color is clear the user has to click on the visible border (does not apply to arcs and pies). Extensive code is added to support clicking on Bezier curves. If an element is already selected it is given priority in the selection process. Click in open space to unselect the element. Several elements can be selected by holding down the shift key or by clicking in open space and draw an enclosing rectangle. A single element or a group of elements can be moved clicking and holding down the left mouse key. Elements can be resized by clicking on one of the eight black marking squares (the mouse cursor changes style in this case). Also a group of elements can be resized.

It is possible to move all elements via the *Tool|Move all* menu, and this is the same as *Edit|Select all* + normal move.

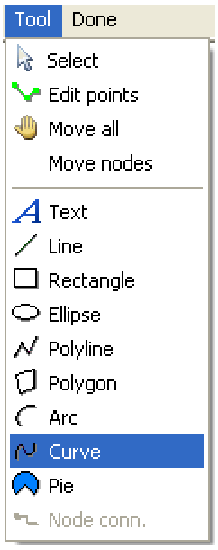
The position of elements can be fine tuned by holding down the shift key and use the arrow keys to move the selected group one pixel. The point position can also be typed directly into the points grid shown in Fig. 5.63.

The order of elements can be changed via the *Edit|Arrange* menu where the four choices, send up/down, send to back/front are available. Elements or groups of elements can also be rotated 90 deg. and flipped left to right or top to bottom via the *Edit|Flip&Rotate* menu.

It is possible to copy selected elements to the windows clipboard. This can then be pasted into other icons (or duplicated). To place the graphical content in metafile format on the clipboard select *Edit|Copy Graphics*.

5.9.3 Drawing new elements

A new element is drawn be selecting the proper tool under *Tools*. The following tools are available:



After selecting the tool click with the left mouse button to place points and with the right mouse button to place the final point. Line, rectangles, ellipses, arcs, and pies take a fixed number of points so the left/right clicking does not really matter in this case. For polylines, polygons, and (Bezier) curves the number of points can range up to 255 maximum. When drawing Bezier curves only the curve points follow the mouse clicks (point 1-4-7-10 etc.) while the intermediate control points (2-3, 5-6 etc.) are calculated internally.

Fig. 5.67 – Available modes and tools

The shape points can be edited later by entering the *Tool|Edit points* mode. The shape points are then drawn as green squares which can be moved directly. It is also possible to add or delete points by clicking the right mouse button and choose from the pop-up menu. Bezier curves are handled in a special way as shown in Fig. 5.68. The curve points are drawn in a lime color while the control points are drawn in read with a line to their curve points. The curve points lies on the curve while the control points sets the curve derivative. (In the drawing tool in Windows office (Word and Power points) the left and right control points are forced to lie on the same tangent and this will force a smooth curve). When points are added to or deleted from Bezier curves this directly affects the curve point while the control points are automatically added/removed. The Bezier curve can be closed by selecting Brush style solid.

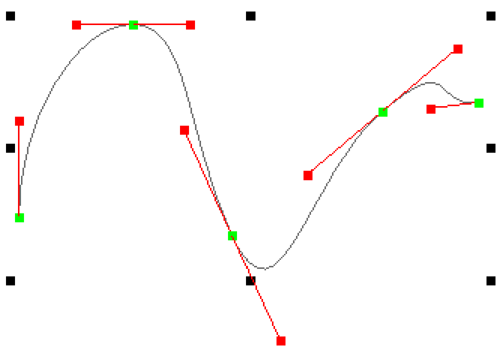


Fig. 5.68 – Bezier curve drawn in *Edit points* mode. Green squares: curve points, red squares: control points.

5.9.4 Layers and visible

Each element can belong to a specific layer as specified in the properties grid in Fig. 5.63-Fig. 5.64. The layers can be shown individually by changing the Show layer item in Fig. 5.62. Elements with Layer=0 are always drawn. The practical usage of this for user specified icons is limited to separation of elements in the drawing process. For standard elements though, the Layer property is used to turn on/off elements dynamically. This is hard coded in the source code of

ATPDraw an affects RLC elements, transformers, time controlled and statistical switches, TACs devices, sources (current/voltage), LCC transmission lines (overhead line, single core cables, enclosing pipe + length), and universal machines. The Layer information is used to control the Visible property. Elements with Visible=false are not drawn in the circuit window, but they are drawn in the icon editor.

5.9.5 Example of complex icons

In the new vector graphics editor quite complex icons can be created. There is however still a limit of the size of the icon (41x41 bytes inherited from the old bitmap icon). This restricts the size to 93 elements. The occupied space of the current icon is shown in the status bar at the bottom shown in fig. 8 (13 % full in this case).

One of the benefits with vector graphic icons is that it is possible to create larger and much more complex icons. Fig. 5.69 shows an example of a created windmill and transformer icons.

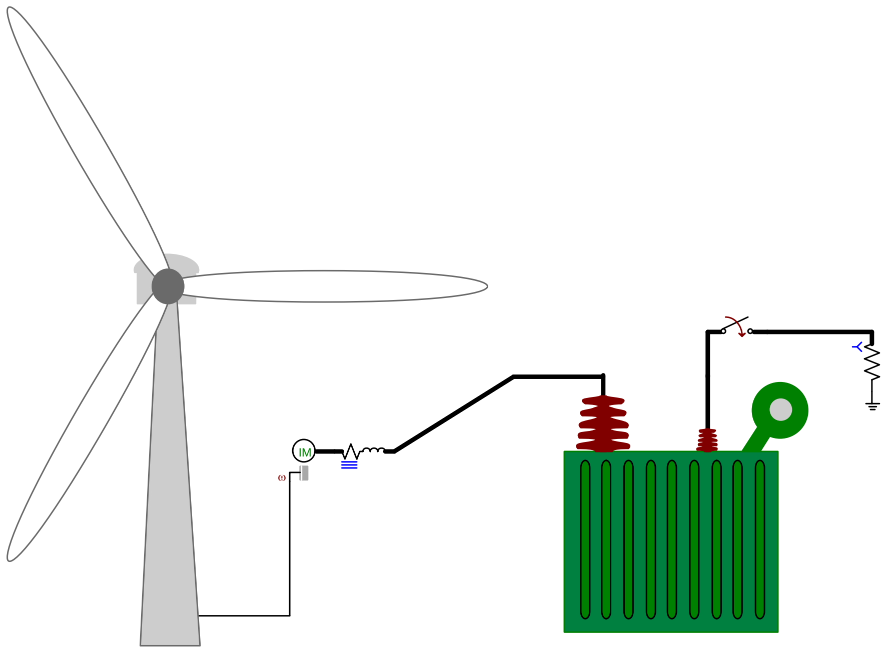


Fig. 5.69 – Windmill and transformer icon with connecting universal machine and load in standard size.

5.10 Bitmap background

It is possible to add a standard graphic background to any component in ATPDraw. This comes in addition to the icon itself. The graphic is included via the *Edit definitions* dialog shown in Fig. 5.70. This dialog in shown from the *Library* menu item in the main menu for support files on disk, or from the *Edit definition* button in all component's dialog box. A button for adding graphic background is shown as the rightmost speedbutton. This brings up the Graphic Background dialog as shown in Fig. 5.71 where a standard bitmap or metafile can be loaded and scaled (Width), positioned relative to the icon centre and be forced to rotate with the icon (only bitmaps can be rotated). This option must be used with care, as graphic backgrounds significantly increase the project file size and the redraw time of circuits.

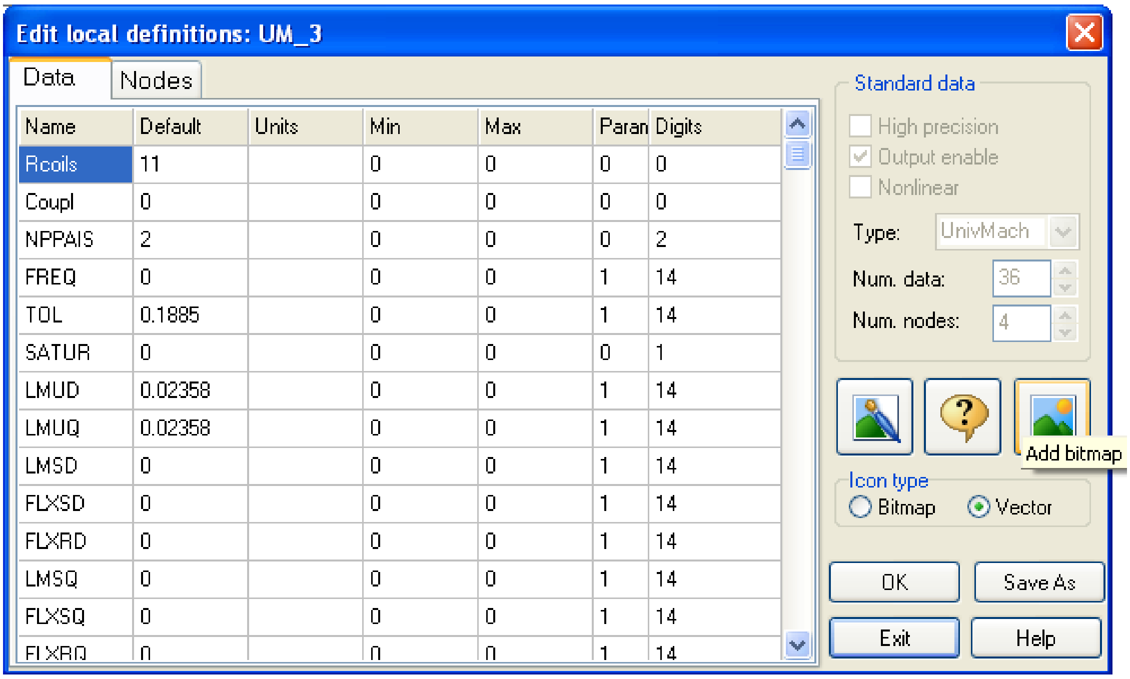


Fig. 5.70 – Edit definitions dialog.

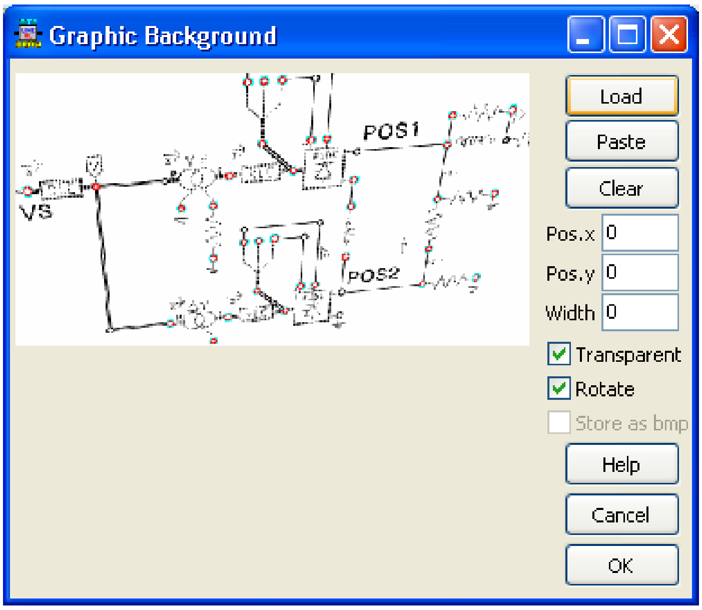


Fig. 5.71 – The graphic background dialog.

5.11 Optimization

This module was added to ATPDraw version 5.6 as part of a co-operation with Schneider Electric. The user has to add a cost function object found in the selection menu under *MODELS|Write Max/Min*. This component will extract a single value from the simulation. In addition variables must be assigned to data in the circuit. These variables can then be tuned to optimize the cost function. The optimization problem is defined as the minimization or maximization of the object function *OF* in *n* dimensions with variables *x*. $\underset{\min}{\overset{\max}{\cdot}} OF(x_1, x_2, \ldots, x_n)$

The variables *x* are can be selected by the user among the global variables.

5.11.1 Optimization routines

Three different optimization routines are supported:

The **Gradient Method** (GM) is the **L-BFGS-B** routine [16] (limited memory algorithm for bound constrained optimization) which is a quasi-Newton method with numerical calculation of the gradient. The gradient is calculated based on the two-point formula:

$$\frac{\partial f}{\partial x} \approx \frac{f(x+h) - f(x-h)}{2h}$$

where the discretization point h is calculated as $h = \max(|x|, 10^{-6}) \cdot dx$ where dx is a user selectable parameter (*delta X*).

If n is the number of variables in the optimization problem the cost function thus has to be evaluated $2n+1$ times for each solution point. This is calculated in a single ATP run utilizing PCVP. The iteration number is somewhat loosely defined in the Gradient Method. If the solution is poorer than the previous point the algorithm steps backwards along the gradient until an improved solution is found and only then the iteration number is incremented.

The **Genetic Algorithm** (GA) is based on the RiverSoft AVG package (www.RiverSoftAVG.com), but modified to better handle the variable constraints. This optimization routine might need further improvement and development. The evolvement of the solution with GA is to more or less randomly select solutions (individuals) and mate these to obtain new solutions. The selection process can be Random, Roulette (using cumulative distribution), Tournament (competition between a user selectable number of randomly selected rivals), Stochastic Tournament (combination of Roulette and Tournament), and Elitism (select only the user defined best percentage of the population). Tournament with 5-10 rivals is a reasonable starting point. The user has to select the size of the population (maximum 1000) and this is a critical parameter which depends on the problem and the number of variables. The user must also select the resolution with 8, 16 and 32 bits available. This part needs further development to allow integer values and arbitrary resolutions. Up to twenty cost function evaluations are performed in parallel using PCVP of ATP.

The **Simplex Annealing** (SA) method is implemented from Numercal Recipies [17]. It is based on the Nelder-Mead simplex algorithm with an added random behaviour gradually reduced (simulated annealing). The algorithm also uses a possible larger set of points (called population) and can support mutation. With all control parameters set to zero the algorithm simply reduces to the classical Nelder-Mead simplex method. The method relies only on function evaluations and POCKET CALCULATOR of ATP is thus not used. Since a single case is run through ATP for each cost function evaluation, the method thus has potential to be extended to include other variables than those defined within the global variables ($Parameter).

5.11.2 Cost function

A general purpose Cost Function in MODELS called WRITEMAXMIN is introduced in ATPDraw version 5.6. The idea is to extract a single value from a simulation and write this to the lis-file and read it back when the simulation is finished. The single value is either the maximum or minimum of the signal *xout* from time *Tlimit* and out to the end time of the simulation. The Model has one input but this can be expanded. The Model also takes in one DATA parameter *AsFuncOf* and if this is assigned to a variable WRITEMAXMIN writes output as function of this data parameter. If *AsFuncOf* is a number it is simply replaced by the simulation number.

WRITEMAXMIN supports multiple run though POCKET CALCULATOR. The selection of the component and its input dialog is shown in Fig. 5.72.

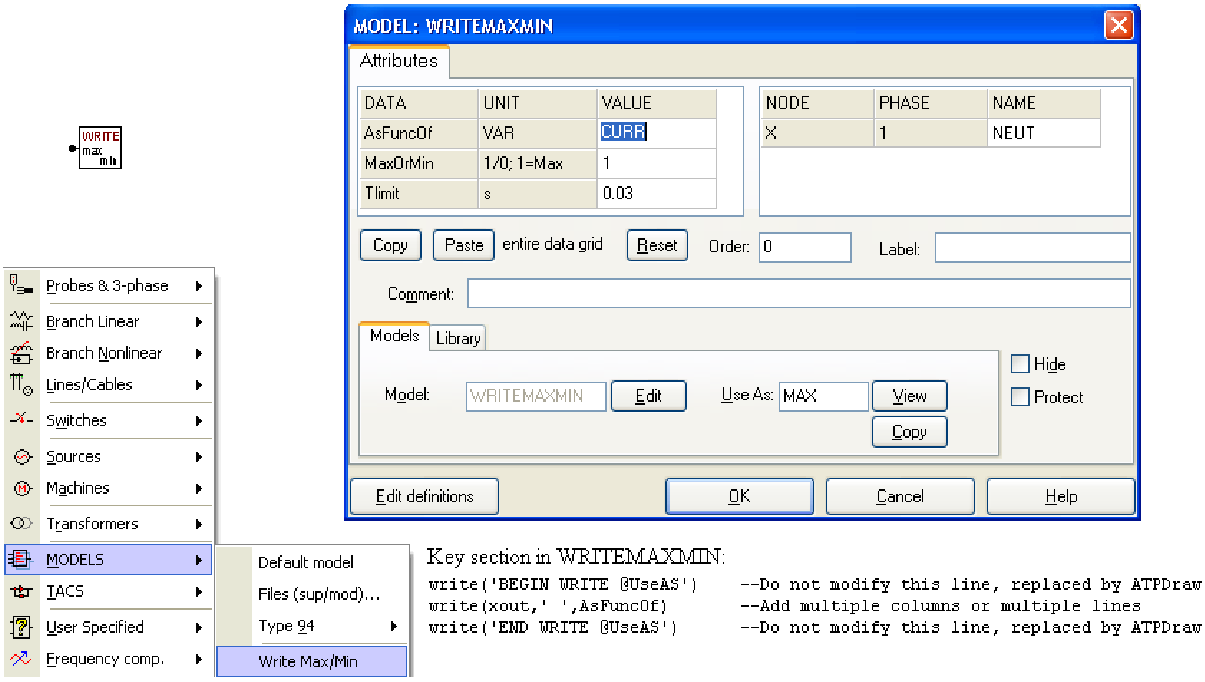


Fig. 5.72 - Cost Function WRITEMAXMIN.

5.11.3 Optimization dialog

The Optimization dialog is found under *ATP|Optimization*. The user has to set up the data case which is not stored with the project. The variables $x_1..x_n$ are chosen by clicking in the *Variables* column and selecting the available variable in the appearing combo box as shown to the left in Fig. 5.73. The user also has to specify the constraints Minimum and Maximum. The Object function must be selected among the available WRITEMAXMIN components in the circuit. The user can then select to minimize or maximize and select a solution method (Genetic Algorithm, Gradient Method or Simplex Annealing). The *Max iter* field is the maximum number of iterations in the solution algorithm.

For the Genetic Algorithm there are several, special selections. The size of the *Population* is a critical parameter. A low number will produce a degenerated result, while a too high number will waste computation time. The maximum allowed number is 1000. The required *Resolution* depends on the selected range (Max-Min). Since it anyhow is recommended to switch to the Gradient Method for fine tuning a 8-bit resolution (255 steps) is normally sufficient. The *Population* count and *Resolution* can not be changed in the optimization process (Continue). The *Crossover* probability should be set to a high number (<1) as the alternative is cloning. The *Inversion* and *Mutation* probabilities should be set to low numbers but this depends on the complexity of the problem. High numbers will slow down the convergence considerably. The *Rival count* for Tournaments should be set to a medium value (2-10). A large number here will approach strong elitism and possible degenerated solutions. The *Preserve fittest* option will simply copy the fittest individual to the next generation (weak elitism). The preferred *Selection method* is one of the Tournament types. Elitism can be selected towards the end of the optimization process.

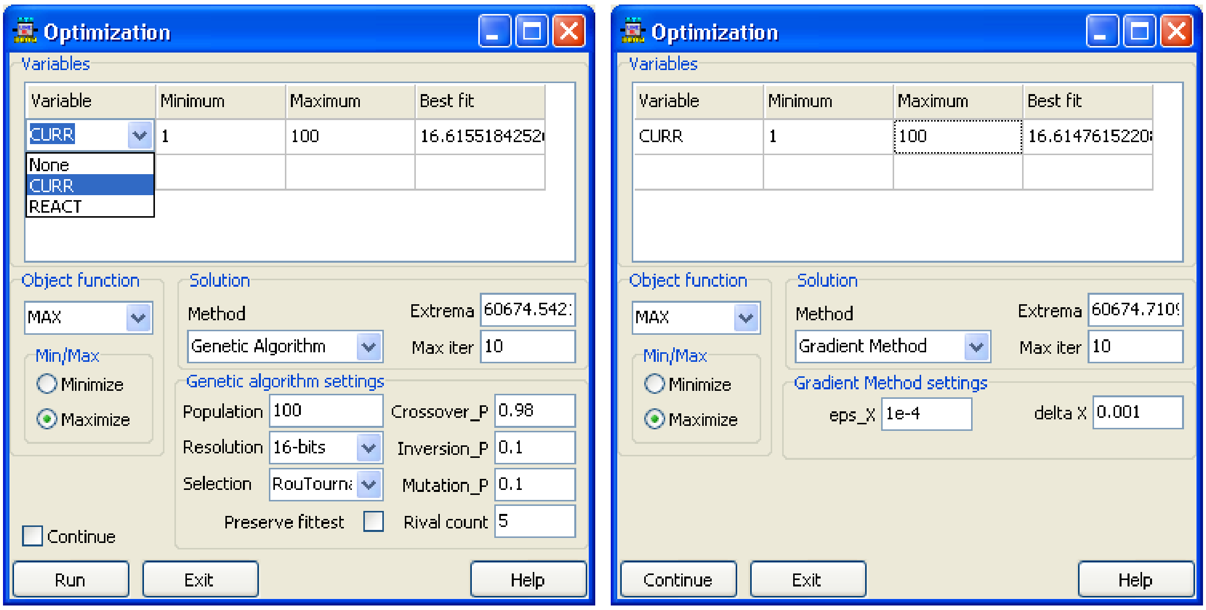


Fig. 5.73 - Optimization dialogs.

For the Gradient Method the user has to specify a convergence limit *eps_x* and a dicretization step in per unit *delta X*. Intermediate trial steps do not count as part of the *Max iter*. The user also has to specify the starting point in the *Best fit* column (if blank the average of Minimum and Maximum is assumed).

For the Simplex Annealing method the user has to choose the *Population* (number of points evaluated for membership in the simplex) which is internally restricted to Population=max (Population, n+1). The *Mutation* probability parameter controls if the new points in the simplex is found at random or with the classical methods reflection, expansion or contradiction. The *Max Climbs* parameter controls how many steps in a negative direction that is accepted by the method. This should be a moderate value 0-3. The parameter *beta* (<1) controls annealing schedule (temperature reduction), and the parameter *ratio* (controls the annealing schedule when a local minima is found. For a rough surface with many local minima the beta and ration parameters needs to be increased. *Ftol* is the convergence criterion (the downside of this method). The iteration stops if $F_{tol} > \frac{2|f_{\max} - f_{\min}|}{|f_{\max}| + |f_{\min}|}$. With all the other parameters set to zero the Simplex Annealing method becomes equal to the Nelder-Mead simplex method.

The user can press ESC to stop the optimization algorithms. When the user clicks on Exit the result of the optimization are written back to the VALUE field in *ATP|Settings/Variables*.

5.11.4 Example: Resonance grounding (Exa_18.acp)

Fig. 4 shows a resonance grounding circuit which could be extended to any complexity. The variable REACT is assigned to the neutral inductor and the unit is set to ohms as XOPT is 50. An intermediate variable CURR is used in Fig. 5.75 to vary the current linearly between 1 and 20 Amps with the special syntax @LIN 1 20 as this is the standard way of quantifying a resonant grounding.

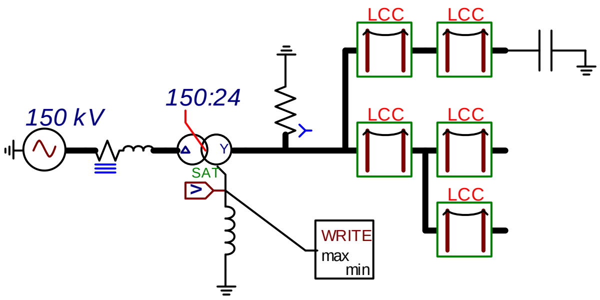


Fig. 5.74 - Resonant grounding circuit.

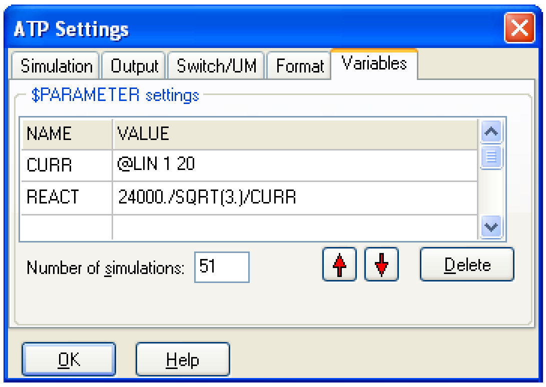


Fig. 5.75 - $Parameter selections @LIN

The new, special Model component WRITEMAXMIN is used to write the maximum value of the neutral voltage as function of the neutral current CURR for all the 51 simulations specified in Fig. 5.75. The input dialog of the Model component is shown in Fig. 5.76. It takes one input and writes the max or min value of this after an onset-time *Tlimit* to the lis-file. After the simulation the results are automatically read back from the lis file and a *View* button is available for charting the results as shown in Fig. 5.77.

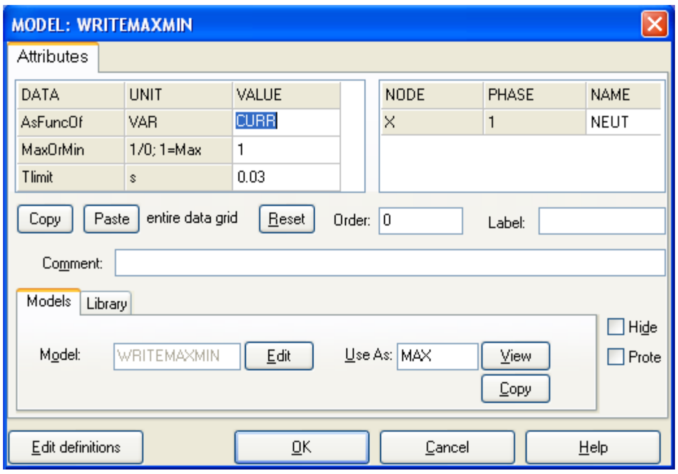


Fig. 5.76 - Input dialog of the new WRITEMAXMIN component.

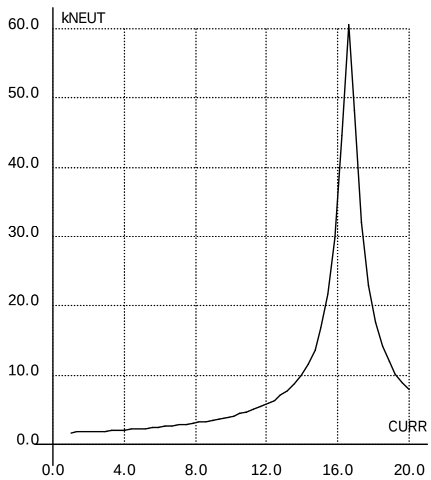


Fig. 5.77 Neutral voltage as function of neutral current.

The exact value of current that corresponds to resonance can be found via the new Optimization module of ATPDraw. This is obtained under *ATP|Optimization* with an input dialog as shown in Fig. 5.73. Fig. 5.73 shows the optimum value found for the GA and GM solution methods. This case with a single variable involved, and a pure convex object function as shown in Fig. 5.77 is simple to solve.

6. Application Manual . . .

. . . .

ATPDraw™
for Windows
5.6

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 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.

6.1.1 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. 6.1.

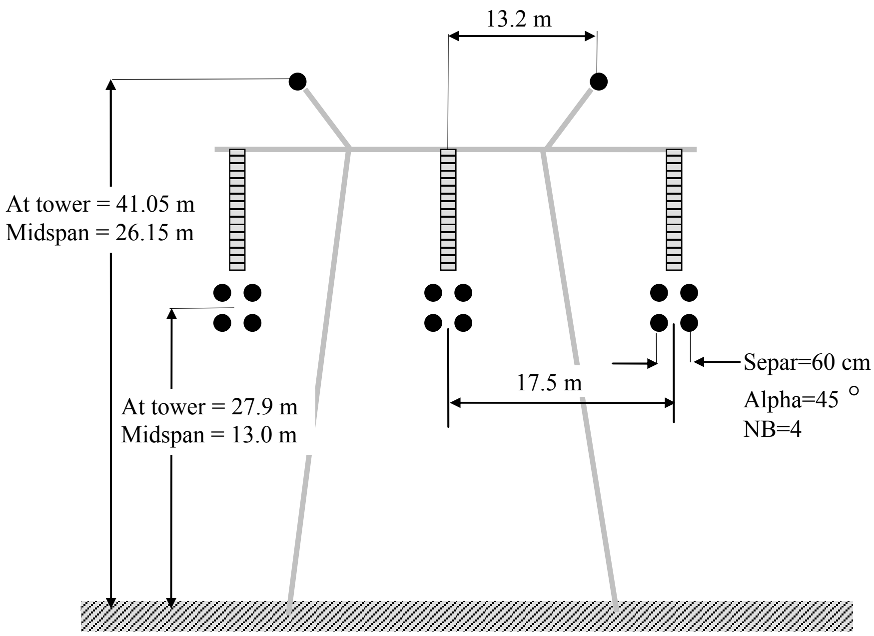


Fig. 6.1 - Tower configuration of the 750 kV line.

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.

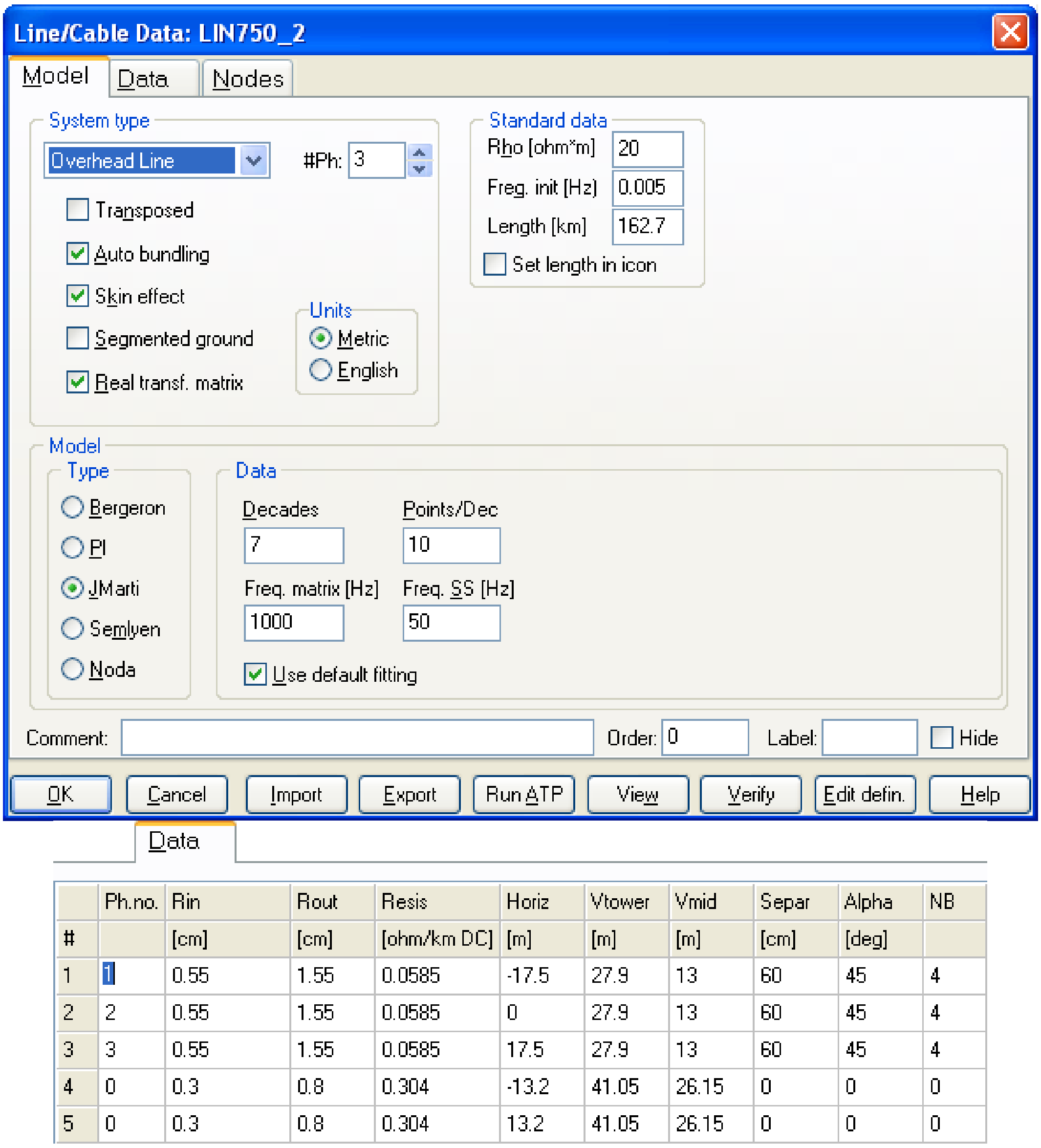


Fig. 6.2- LCC Model and Data tab of the 1st section of the 750 kV line.

```
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
```

6.1.2 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. 6.3. 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.

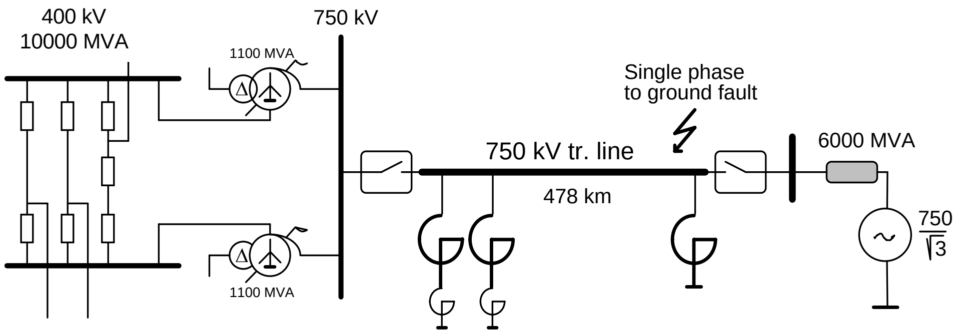


Fig. 6.3 - One line diagram of the faulted line.

The layout of the completed ATPDraw circuit is shown in Fig. 6.4. 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.

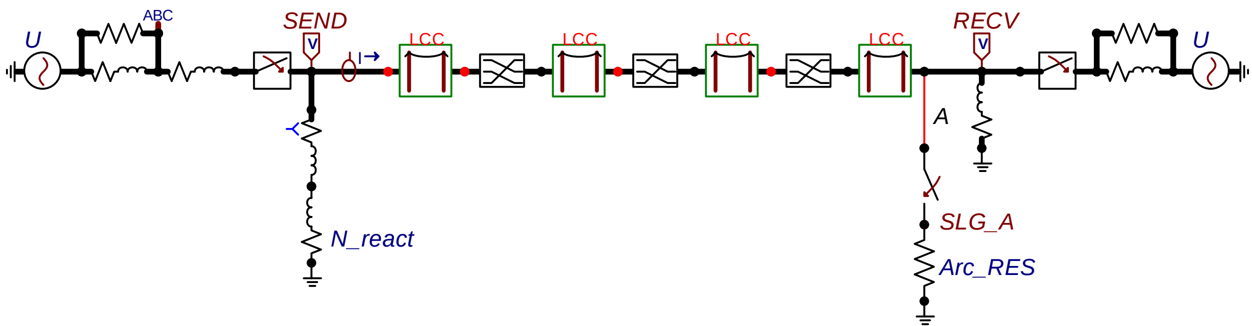


Fig. 6.4 - Line-to-ground fault study (`Exa_7a.acp`)

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  >
  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. 6.5 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. 6.6 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.

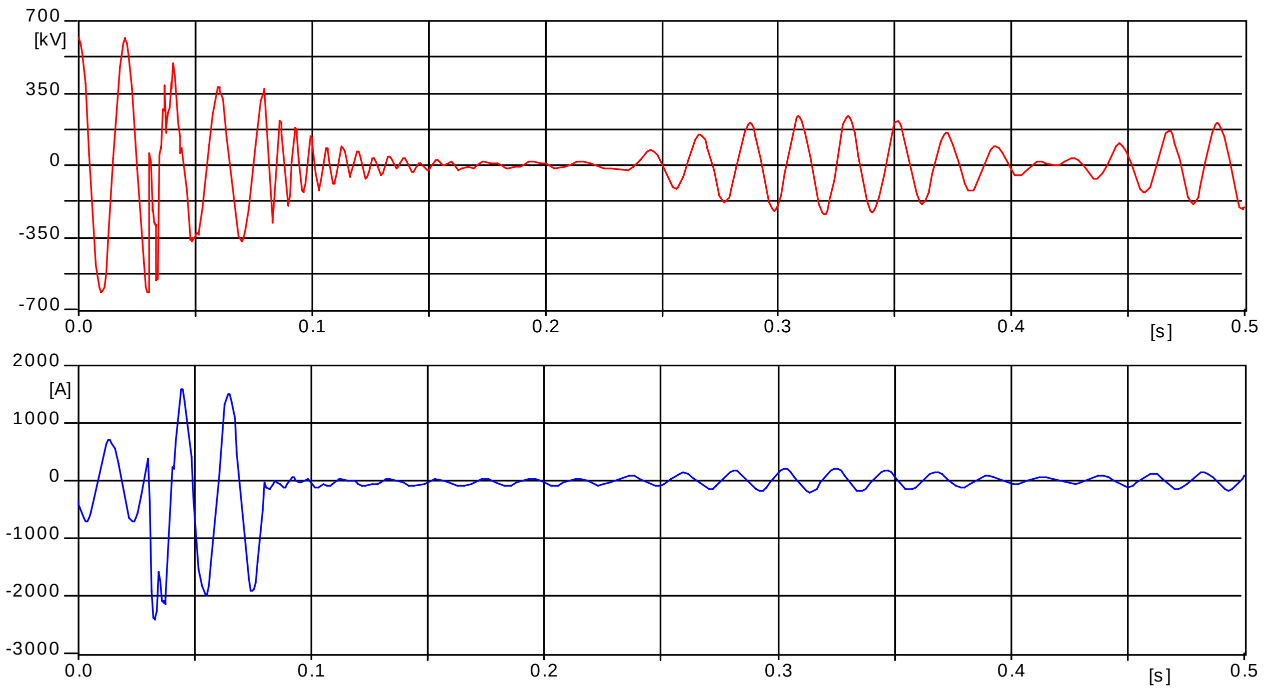


Fig. 6.5 - SLG fault and fault clearing transients (simulation).
upper curve: phase to ground voltage, lower curve: line current

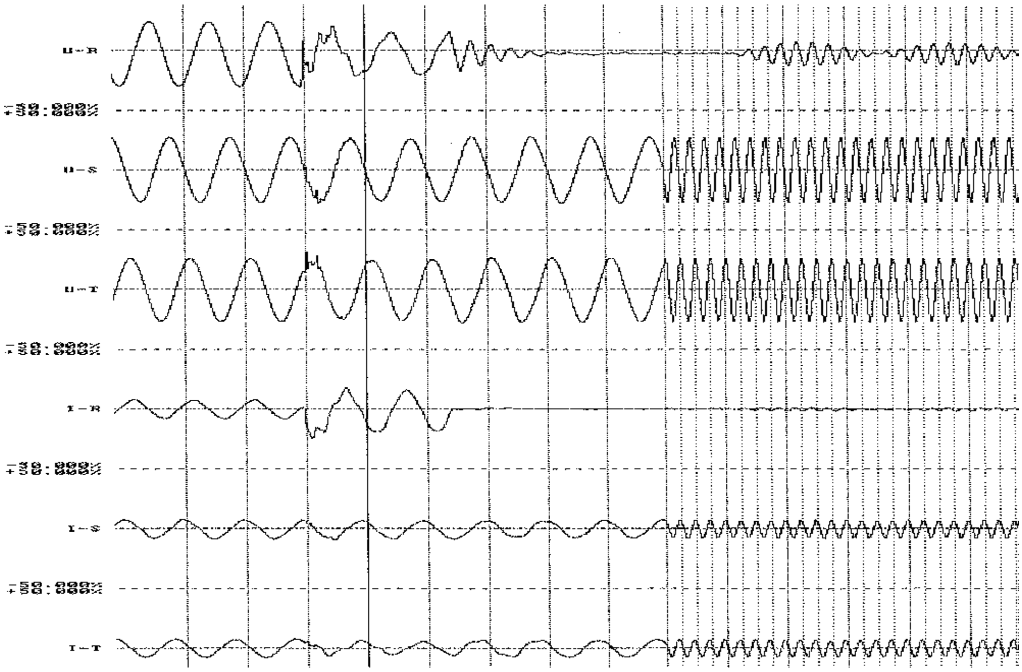


Fig. 6.6 - SLG fault and fault clearing transients. Phase currents and voltages recorded at a staged fault test by a variable sampling frequency disturbance recorder.

6.2 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. 6.7. 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.

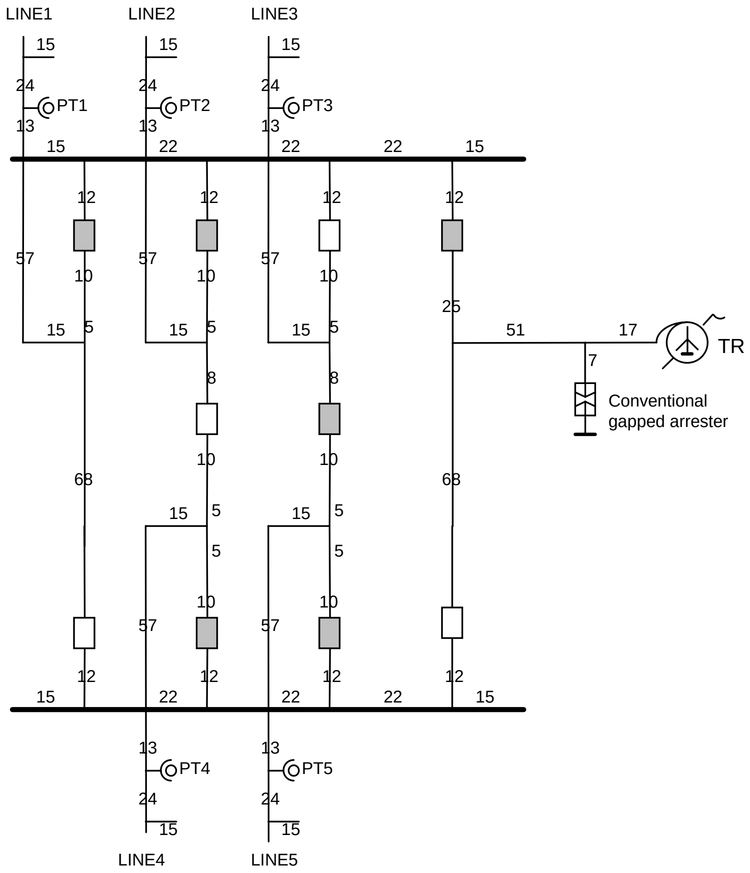


Fig. 6.7 - One-line diagram of the substation

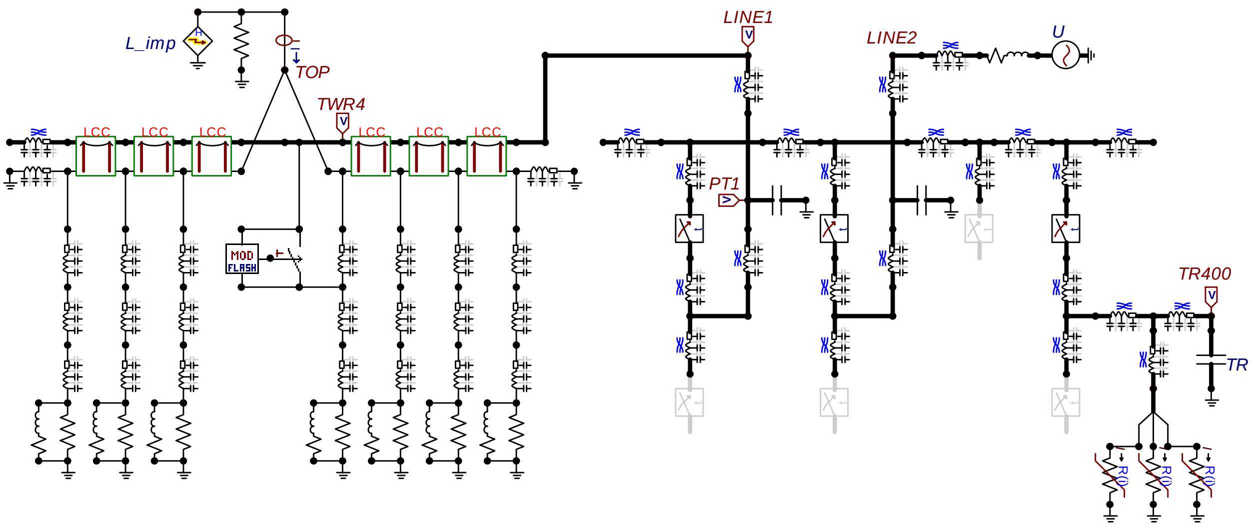


Fig. 6.8 - Example circuit (`Exa_9.acp`)

The ATPDraw circuit of the complete network (substation+incoming line) is shown in Fig. 6.8. 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        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
-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. 6.9. 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.

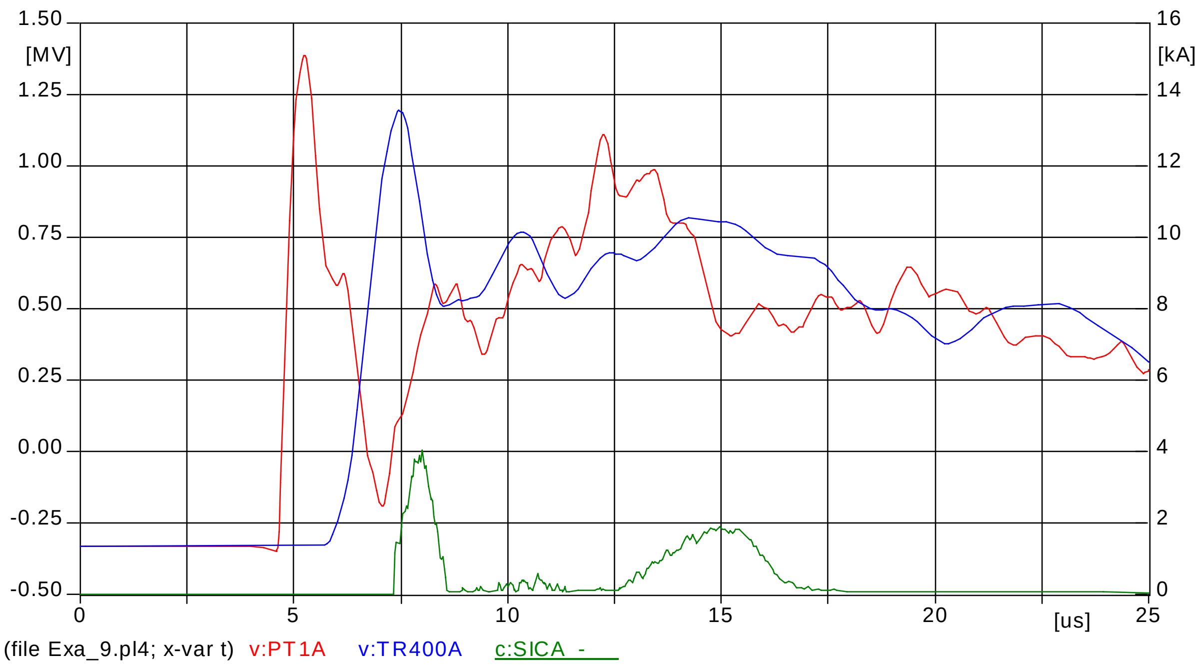


Fig. 6.9 - ATP simulation results. Red: incoming surge at the substation entrance. Blue: voltage stress at the transformer terminal. Green: arrester discharge current.

6.3 Modeling Rectifiers, zigzag transformers and analysis of Harmonics *(Exa_14.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 diode rectifier will be used instead and the focus shifted to harmonics in the supplying line currents. The case is an industrial plant consiting of AC/DC converters and consuming 55 MW for aluminium production. The plant is supplied by a 132 kV high voltage AC system and there are concerns about the harmonics in the current on the high voltage side. This example shows how to model an equivalent 24 pulse diode rectifier and calculate the harmonics in currents in Models. The harmonics could alternatively have been calculated as a part of a post-processing. Fig. 6.10 shows the example circuit.

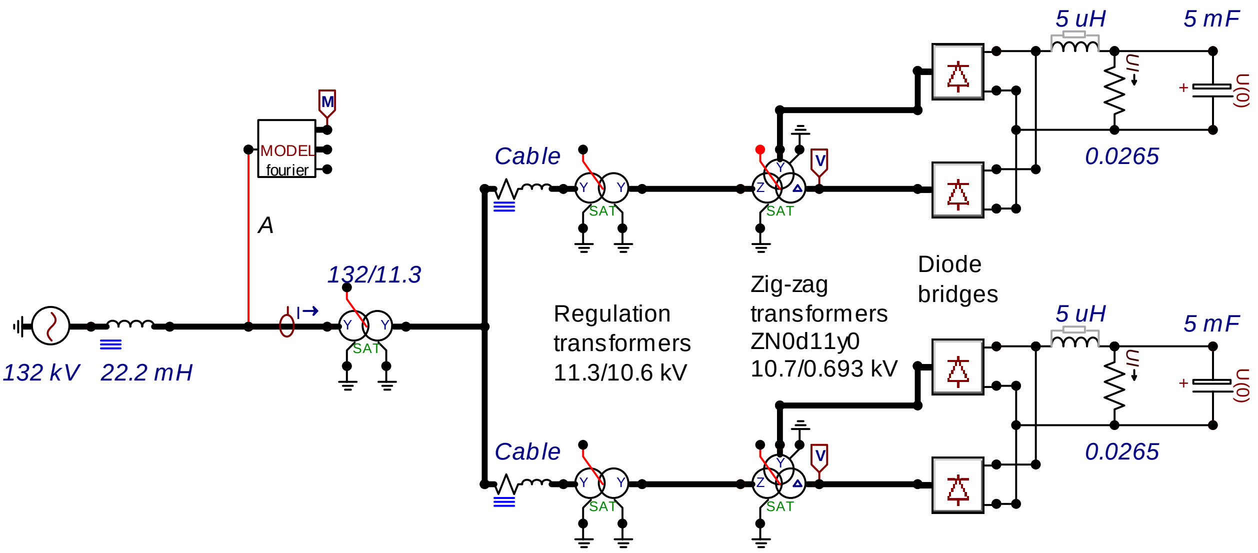


Fig. 6.10 – Example circuit (`Exa_14.acp`).

The diode bridge is modeled and compressed into a group as shown in Fig. 6.11. Note the need for small resistors (1 μΩ) to decouple the diodes and added snubber circuits. The R and C data for all six snubbers are added to the External parameter group, but will appear as only two parameters in the compressed object. A bitmap icon is created for diode bridge.

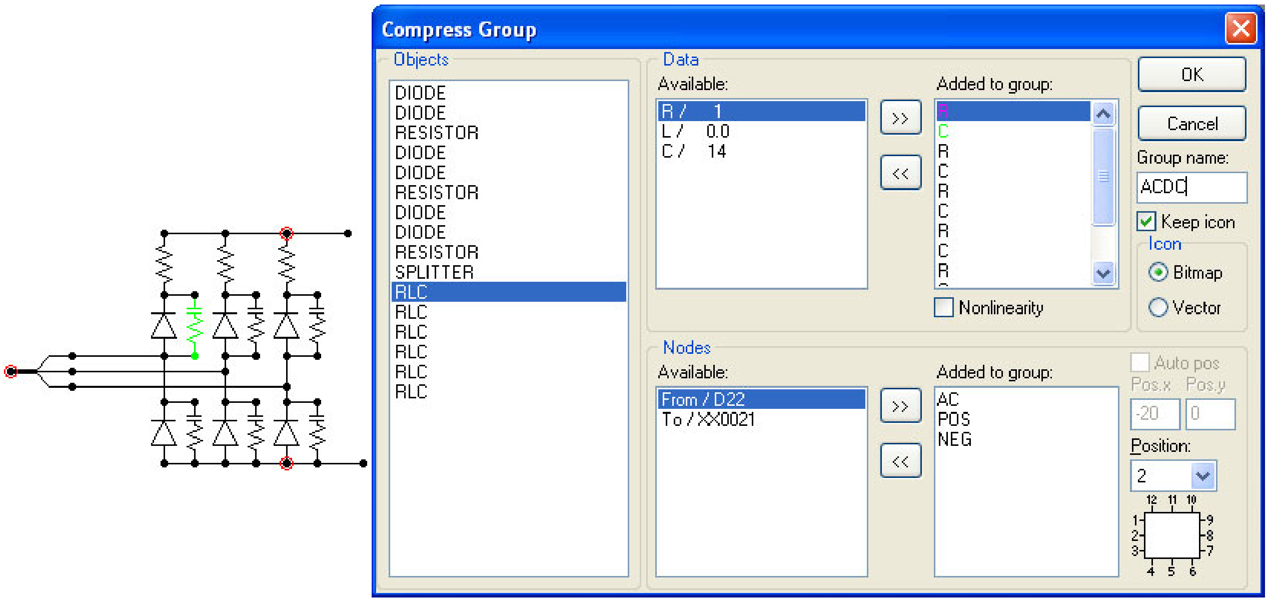


Fig. 6.11 – Compress a 3-phase diode bridge.

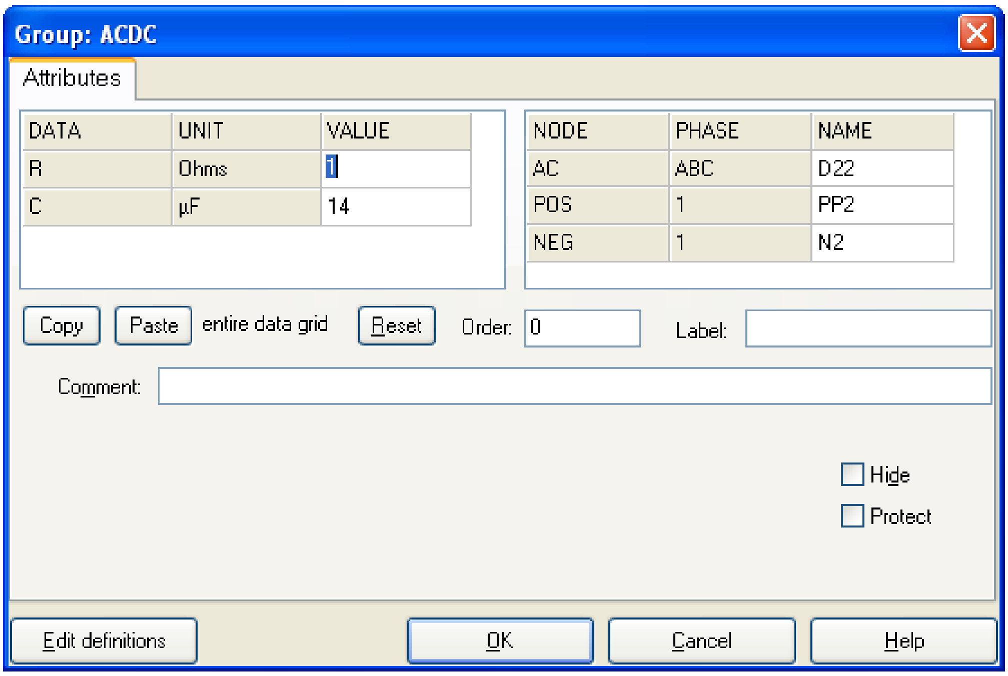
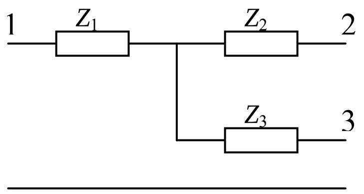


Fig. 6.12 – Component dialog of the compressed group ACDC.

The key unit to produce the 24-pulse system are the two supplying transformers phase shifted 15 degrees and with a Y and Δ coupling on the secondary side. This is accomplished by using the Saturable Transformer component with a zigzag coupling on the primary winding. The input dialog of the upper transformer is shown in Fig. 6.12. The Saturable Transformer requires direct input of electrical quantities so recalculation of Test Report data is required. The transformers had the following test report data:

Coupling:	ZN0d11y
Rated power:	24.8 MVA
Rated primary voltage:	10.735 kV
Rated secondary voltage:	693 V
Rated tertiary voltage:	693 V
Rated frequency:	50 Hz
Open circuit current:	0.0056 pu
Short circuit impedance 1-2:	0.0084 + j0.1015 pu
Short circuit impedance 1-3:	0.0084 + j0.1015 pu
Short circuit impedance 2-3:	0.0210 + j0.1887 pu
Phase shift Z (ref. 3):	±7.5 deg.

This will result in the standard per unit equivalent circuit for the short circuit impedances



$$
\begin{aligned}
Z_1 &= (Z_{12} + Z_{13} - Z_{23})/2 = -0.0021 + j0.00715 \ [pu] \\
Z_2 &= (Z_{12} + Z_{23} - Z_{13})/2 = 0.0105 + j0.0944 \ [pu] \\
Z_3 &= (Z_{13} + Z_{23} - Z_{12})/2 = 0.0105 + j0.0944 \ [pu]
\end{aligned}
$$

Fig. 6.13 – Per unit equivalent circuit of the 3-winding transformer.

Note the negative resistance in the primary winding. This could result in a stability problem in the simulations, but fortunately this didn't seem to be the case in this example. The input dialog of the Saturable transformer with the electrical parameters is shown in Fig. 6.14.

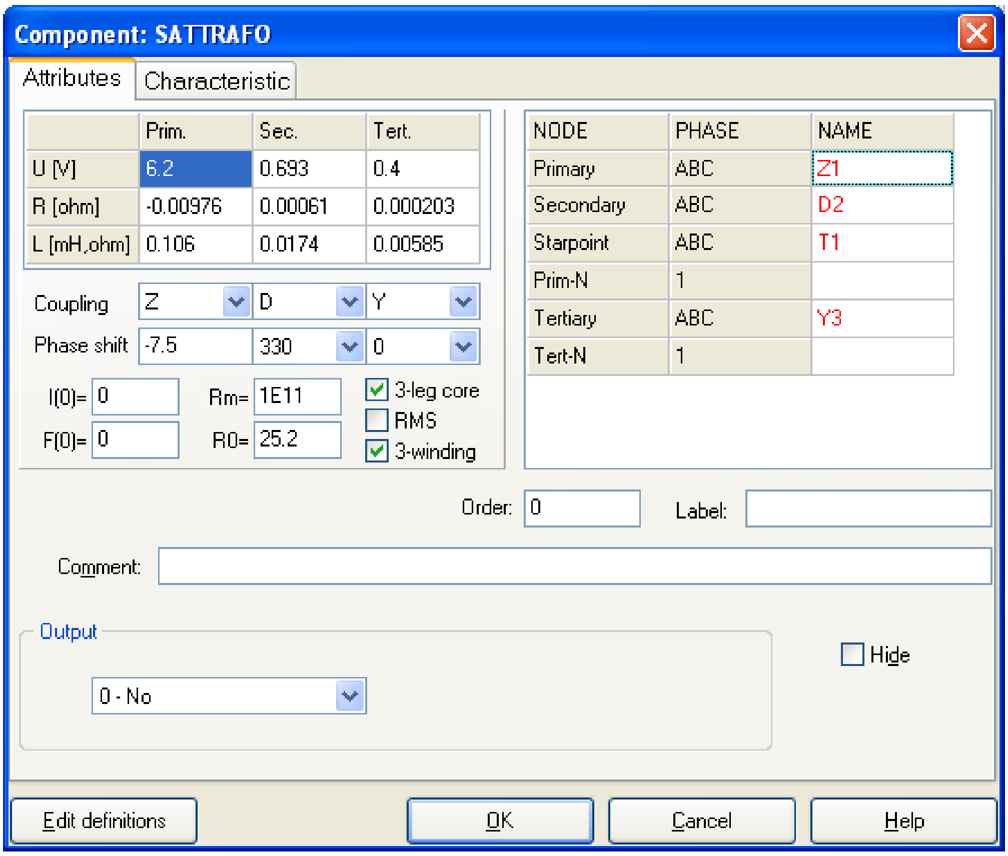


Fig. 6.14 – Component dialog of the Saturable Transformer component.

The total winding voltage is $U_A = 10.735/\sqrt{3}$ kV = 6.2 kV

The short circuit impedance is

$$Z_1 = \left(-0.0021 + j0.00715\right)\cdot\left(10.735\ \text{kV}\right)^2 / 24.8\ \text{MVA} = -0.00976 + j0.0332\,[\Omega]$$

The zigzag winding 1 is further split in Z and Y parts with $n = \dfrac{\sin(7.5^\circ)}{\sin(60-7.5^\circ)} = 0.165$.

The voltages across each winding part and the individual leakage impedances are automatically calculated by ATPDraw as:

$$U_{1z} = \frac{10.735/\sqrt{3}}{\cos(7.5^\circ) + 0.165\cdot\cos(60^\circ - 7.5^\circ)}\ \text{kV} = 5.68\ \text{kV}$$

$$U_{1y} = 5.68\ \text{kV}\cdot 0.165 = 0.934\ \text{kV}$$

$$R_{1z} = -0.00976\cdot\frac{1}{1+0.165}[\Omega] = \text{-}\,8.4\,[\text{m}\Omega]$$

$$R_{1y} = -0.00976\cdot\frac{0.165}{1+0.165}[\Omega] = \text{-}\,1.4\,[\text{m}\Omega]$$

$$L_{1z} = \frac{0.0332}{2\pi\cdot 50}\cdot\frac{1}{1+0.165^2}[\text{H}] = 0.103\,[\text{mH}]$$

$$L_{1y} = \frac{0.0332}{2\pi\cdot 50}\cdot\frac{0.165^2}{1+0.165^2}[\text{H}] = 2.79\,[\mu\text{H}]$$

If the HV winding 1 is chosen as the primary winding, the magnetizing branch will be added to the first winding part (Z) of the zigzag winding. This is probably not a good choice, and alternatively the magnetizing branch should be added to the low-voltage Y-coupled winding. This could be done externally or by choosing winding 3 as the primary.

The measured inductance is

$$L_m = \frac{1/\sqrt{3}}{2\pi\cdot 50\cdot 0.0056}\text{pu} = 0.328\ \text{pu} = 0.328\cdot\left(10.735\ \text{kV}\right)^2 / 24.8\ \text{MVA} = 1.52\ [\text{H}]$$

and the inductance that should be added to winding 1Z in ATP:

$$L_{mz}^{ATP} = \frac{L_m}{1+n+n^2} = 1.28\ [\text{H}]$$

Saturation is of no importance in this example and a single point is set on the characteristic page (i, λ)=(1, 1.28).

If a measurement of the zero sequence impedance is missing a reasonable assumption for this particular transformer is to set it to 2/3 of the positive sequence magnetizing current. Further, the zero sequence inductance added in ATP is one half of the real value. This gives

$$R_0 = \frac{U_{z0}^2}{3\cdot L_{0z}^{ATP}} \approx 2\cdot\frac{5.68^2}{2\cdot L_{mz}^{ATP}} = \frac{5.68^2}{1.28} = 25.2\,[\Omega]$$

The Delta- winding:

The total winding voltage is $U_{A2} = 0.693$ kV

The short circuit impedance is

$$Z_2 = \left(0.0105 + j0.0944\right)\cdot\left(\sqrt{3}\cdot 0.693\ \text{kV}\right)^2 / 24.8\ \text{MVA} = 0.61 + \text{j}5.48\,[\text{m}\Omega]$$

R_2=0.61 [mΩ] and L_2=17.5 [μH]

The Wye- winding:

The total winding voltage is $U_{A3} = 0.693/\sqrt{3} = 0.4$ kV

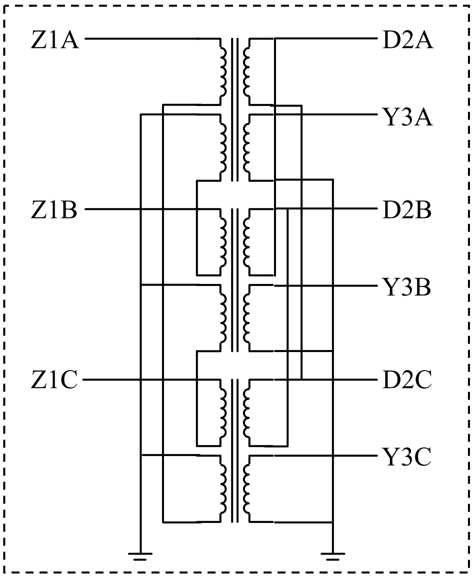
The short circuit impedance is

$$Z_3 = (0.0105 + j0.0944)\cdot\left(\sqrt{3}\cdot 0.4\text{ kV}\right)^2 / 24.8\text{ MVA } = 0.203 + \text{j}1.83\,[\text{m}\Omega]$$

R_3=0.203 [mΩ] and L_3=5.85 [μH]

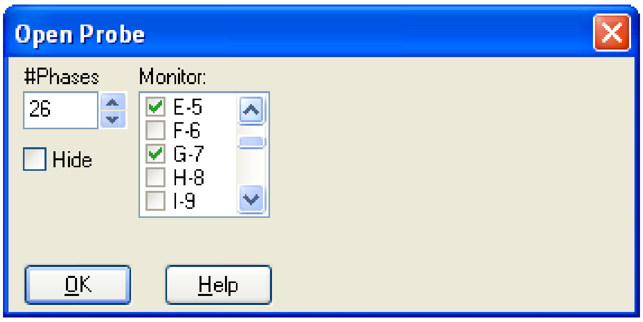
The ATP file format and connectivity of the transformer specified in is:

```
  TRANSFORMER THREE PHASE TX0001  25.2
  TRANSFORMER                                 T1A     1.E11
              1.            1.28
            9999
 1Z1A    T0002C                  -.0084.103215.6797
 2       T0002A                  -.0014.00279.93446
 3D2A    D2C                     .00061 .0174  .693
 4Y3A                             .0002.00585    .4
  TRANSFORMER T1A                             T1B
 1Z1B    T0002A
 2       T0002B
 3D2B    D2A
 4Y3B
  TRANSFORMER T1A                             T1C
 1Z1C    T0002B
 2       T0002C
 3D2C    D2B
 4Y3C
```



The example shown in Fig. 6.10 also includes a stepdown transformer and regulating transformer (regulation not modeled) that also are modeled as Saturable Transformer components. Alternatively the BCTRAN or Hybrid Transformer models could have been used as they have an internal conversion of test report data. These models do not support Zigzag transformers, however.

The harmonics are calculated by an algorithm in MODELS. This is shown in chapter 5.5.1 in this manual. The automatic approach is assumed. A default model is used and the Models text is typed in under Edit. The output of absolute value and angle are declared s 26-phase (ABSF and ANGF) while the input X is single phase. The user can selec the type of input (switch current in this case) by cliakin on the left input node of the model and select *Input Current* in the Node dialog box. The Model will output all harmonics 0..N (where N is a data parameter) as a function of time. The calcualtion is performed by integration of a sliding window of size 1/FREQ [sec]. The selection of variables to plot is made from a models probe connected to the ABSF node.



The probe is set to 26-phases and the the phases of special interest 1, 5, 7, 11, 13, 23, 25 are checked under Monitor.

Fig. 6.15 – Model probe dialog.

The line current in phase A at the 132 kV side is selected as input. A connection is drawn from the left 3-phase side of the switch an to the single phase Model input node. In the Connection dialog that then pops up phase A is selected. The simulated phase A current is shown in Fig. 6.16 and the 5^{th} 7^{th}, 23^{rd} and 25^{th} harmonics calculated in Models shown in Fig. 6.17.

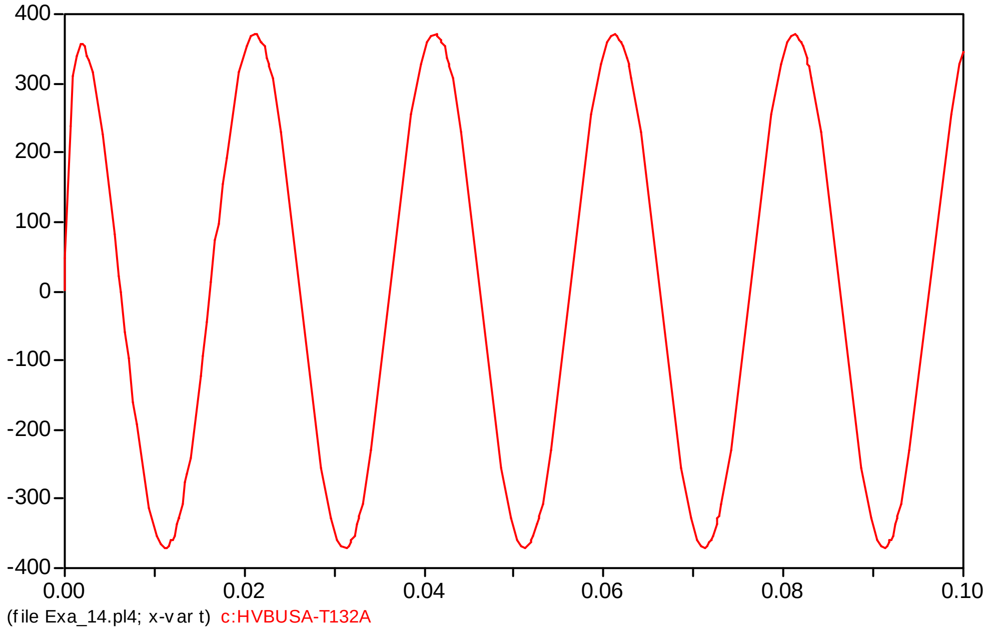


Fig. 6.16 – Simulated line current phase A at the 132 kV side.

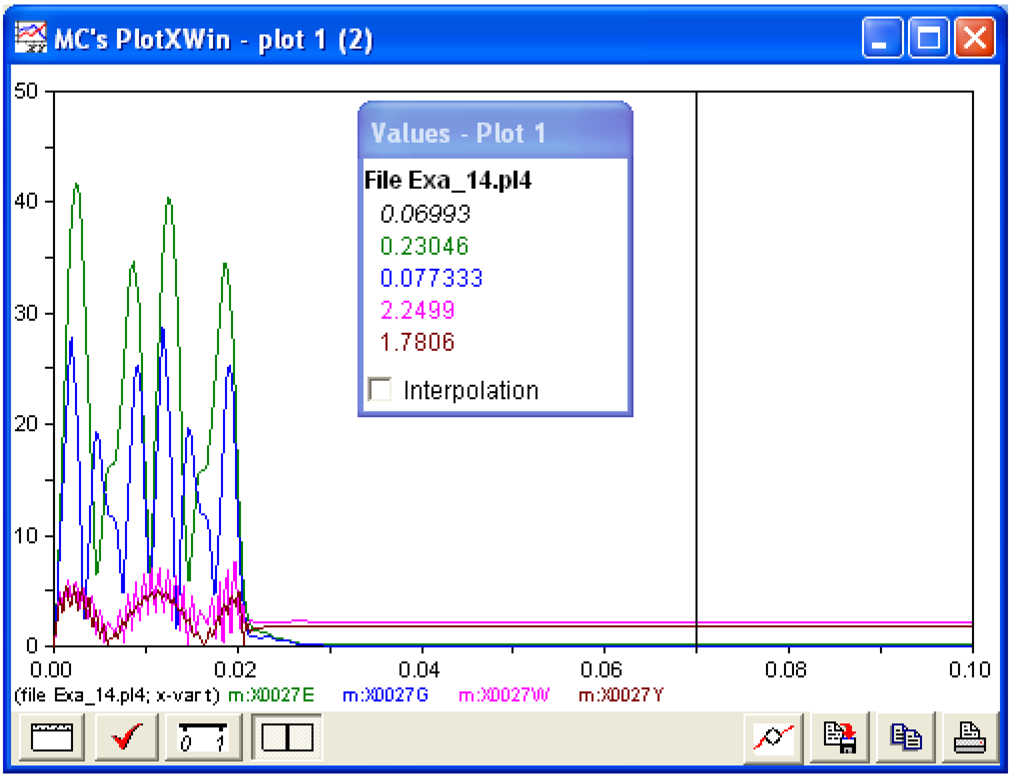


Fig. 6.17 – The harmonics of the current in Fig. 6.16.

The harmonics can also be calculated in for instance PlotXY as shjown in Fig. 6.18, but not as a function of time.

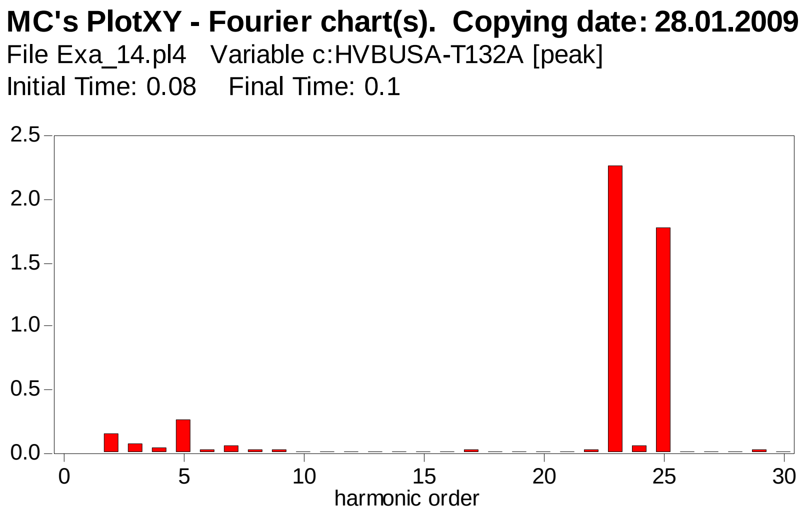


Fig. 6.18 – Harmonics calculated by FFT in PlotXY.

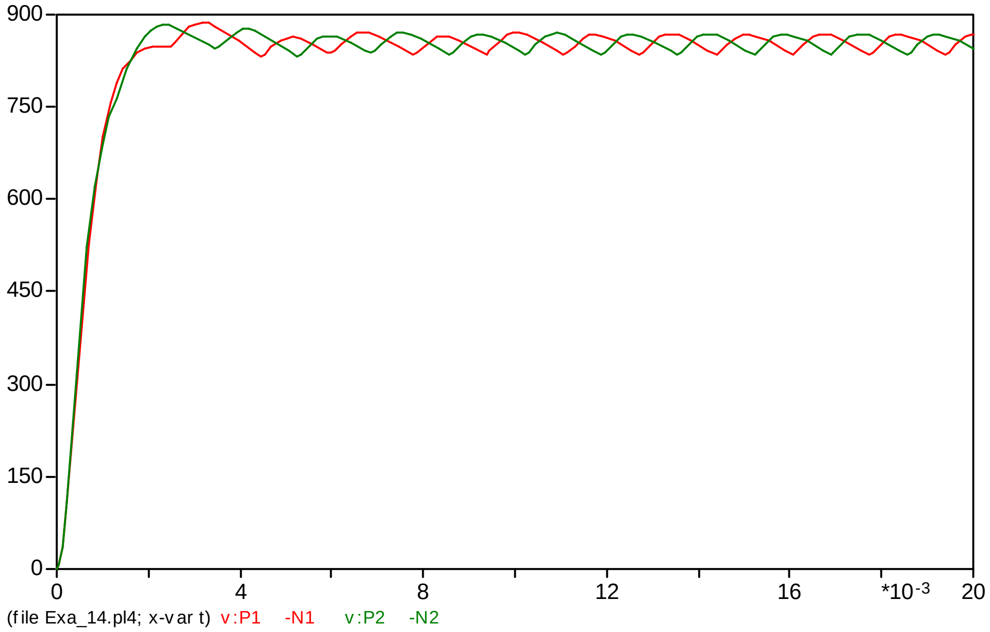


Fig. 6.19 – DC voltages on the LV side.

6.4 Modelling of electrical machines and controls

This section illustrate a few examples of machine and control modeling with emphasis on how to interphase the various component involved.

6.4.1 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. 6.20/b:

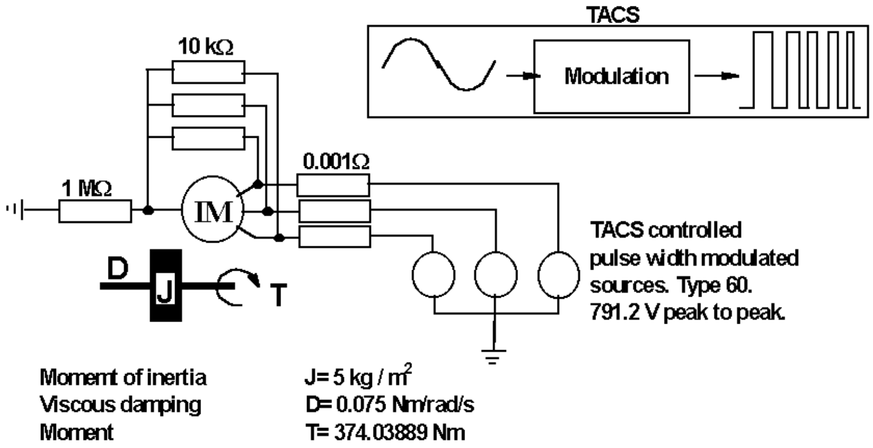


Fig. 6.20/a - Induction machine + TACS

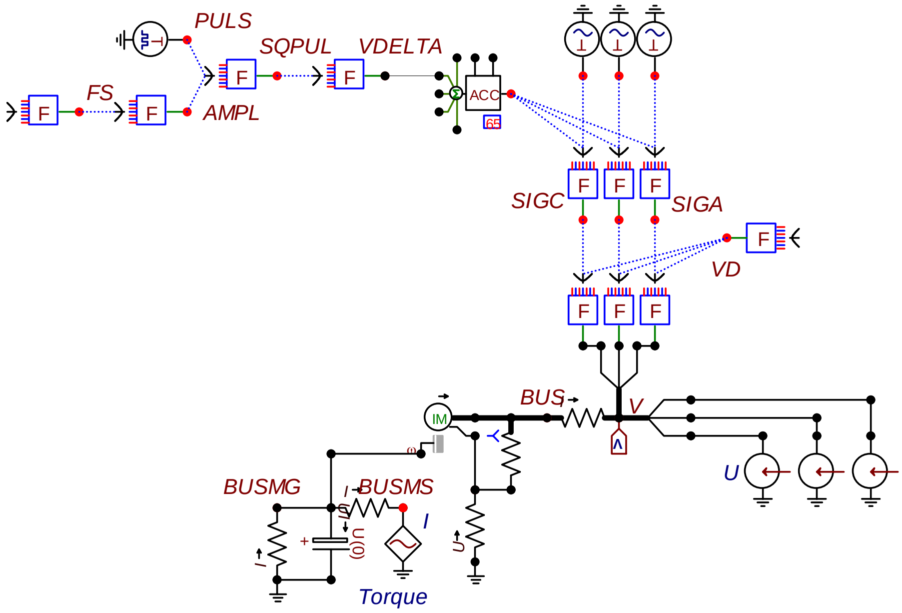


Fig. 6.20/b - ATPDraw scheme of the induction machine example (`Exa_4.adp`)

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. 6.21. This TACS object creates the armature voltage in phase *A* of the 3-phase node *V*.

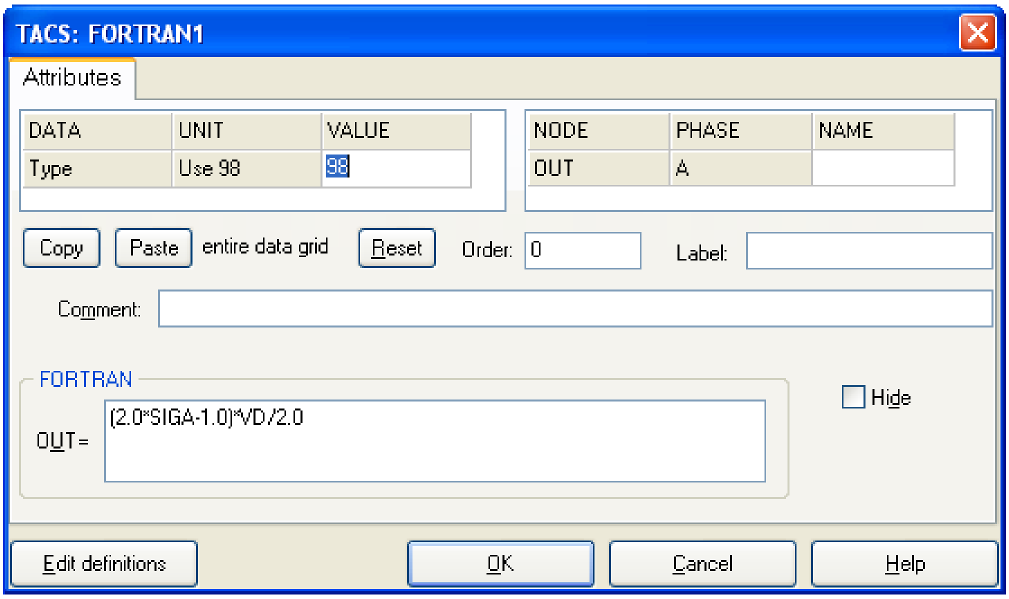


Fig. 6.21 - TACS Fortran input window

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. 6.22:

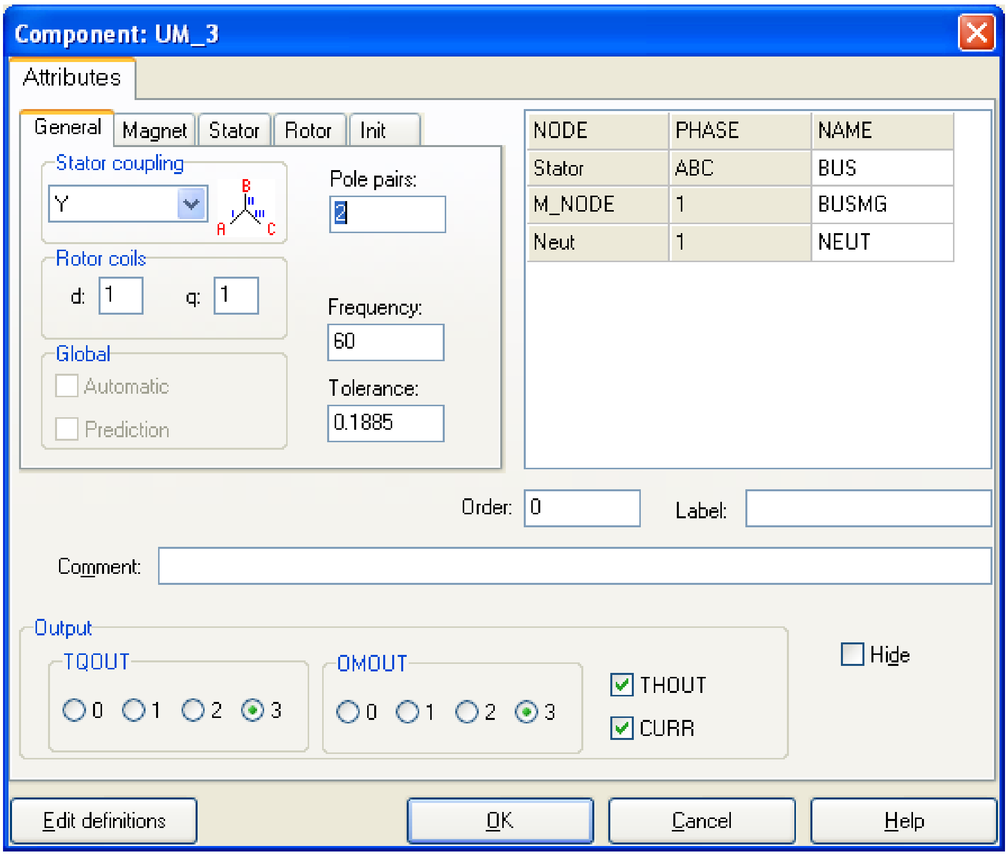


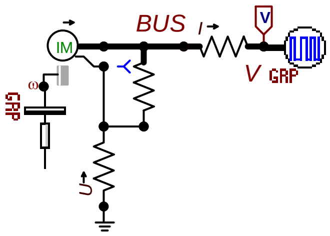
Fig. 6.22 - Induction machine input window

The numerical values in Fig. 6.22 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. 6.23 - PWM source and mechanical load compressed into a single icon.

6.4.2 Windsyn machine model

A challenge with the above example is to obtain the electrical data for the induction machine. The program WindsynATPDraw is integrated with ATPDraw and enable manufacturers data to be used instead of the electrical data. The input dialog of the Windsyn component is shown in Fig. 6.24. You can specify the data in this dialog, but you have to click on *Run Windsyn* to create the model.

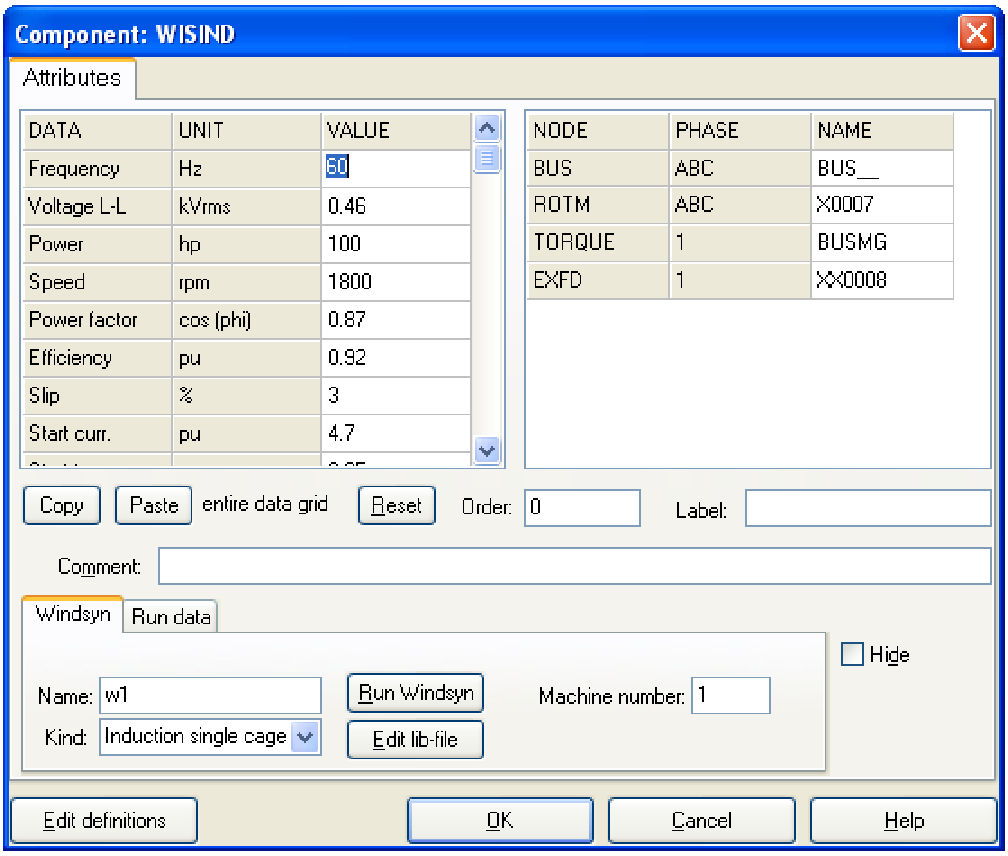


Fig. 6.24 – Windsyn input data. Induction machine, wound rotor.

Automatic initialization of the machine was chosen as set under *ATP|Settings/Switch&UM*. The required manufacturers data for producing the same electrical model as in `Exa_4.acp` were not available. The efficiency and starting current parameters were adjusted to reach relatively close to the data given in Exa_4.acp. Note that the mechanical network is included inside the Windsyn component and that the stator neutral is assumed directly grounded. This resulted in comparable stator current in steady state as shown in Fig. 6.25. Installation of `WindsynATPDraw.exe` is required to use this component. The link to the installed program is set under *Tools|Options/Preferences-Windsyn*. Note that Windsyn in the version used here resets the units of inertia to kWs/kVA each time. Besides this it was possible to simply click on *Continue-Continue-Create files|Save run data-Exit* in Windsyn. When you click on *Exit* in Windsyn the control goes back to ATPDraw (press ESC if Windsyn does not terminate properly) and the data files `atpdraw.pch` and `atpdraw.wis` are read into memory. The `pch` file is then run through ATP to produce the `'Name'.lib` used for $Include. This file (`w1.lib` in this case) is written to the same loacation as the final ATP file (Result Directory). You can inspect this 'file' by clicking the *Edit lib-file* button.

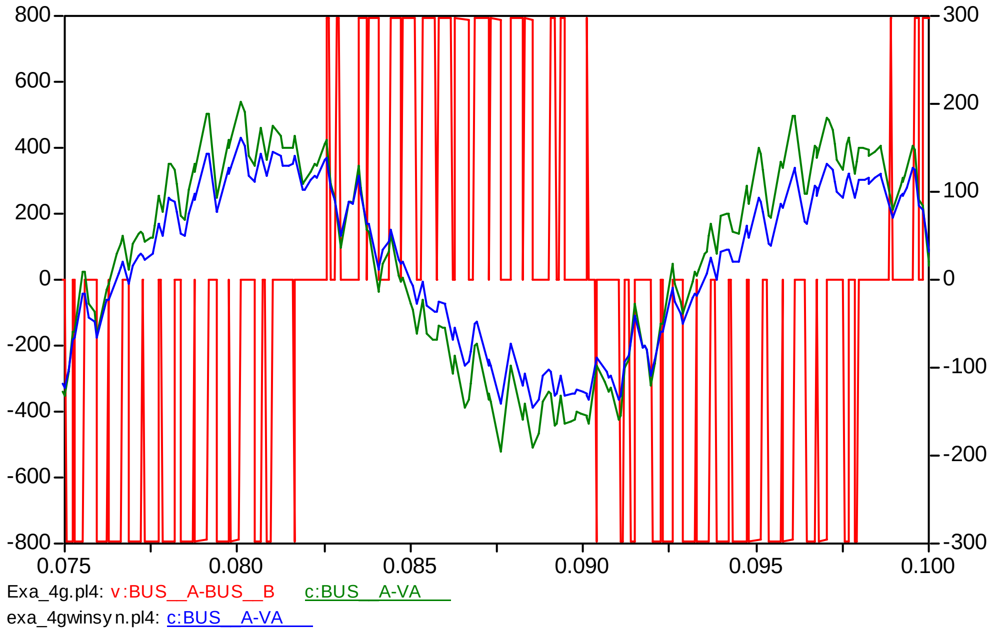


Fig. 6.25 – Simulated PWM line voltage and stator current in steady-state.

6.4.3 Machine control *(Exa_17.acp)*

Machine control is typically of minor importance in an electromagnetic transients program as the time constants involved are much larger than the electrical time constants. Newertheless is some situation it might be of interest. The Fig. 6.26 shows a simple example where the Windsyn synchronous machine model is being controlled by a governor and an exciter. The loads of the machine doubles at 2 seconds and goes back to the initial 500 kW at 10 seconds. The Windsyn generator is auto-initialized and this involves two sources hidden inside its lib-file. Initialization of the control units can thus be a challenge. To control the machine additional external sources must be adjusted. MODELS is here used for convenience, but TACS components will result in much master performance. The Windsyn component requires the special request card 'UM TO TACS' so be able to do calculation performance parameters in TACS. This is added as a *User Specified|Additional* component. The parameters used and the type of controls may certainly be discussed, but the point here is to illustrate the interface between machine and control.

The speed control takes as input the actual speed of the machine (voltage at the TORQUE node of the machine) and gives out the torque to an additional current source connected to the same node.

The voltage control takes as input the phase A voltage to ground and gives out the field voltage to an additional voltage source. The example shows how to get the field current and initial field voltage into the ST1A exciter model A separate model is used to calculate the rms value.

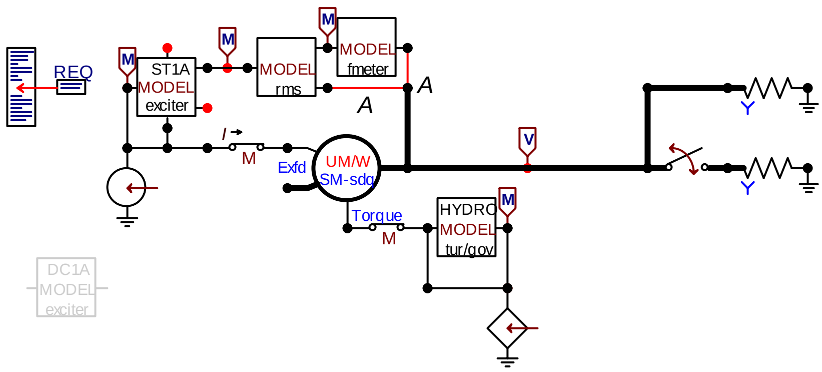
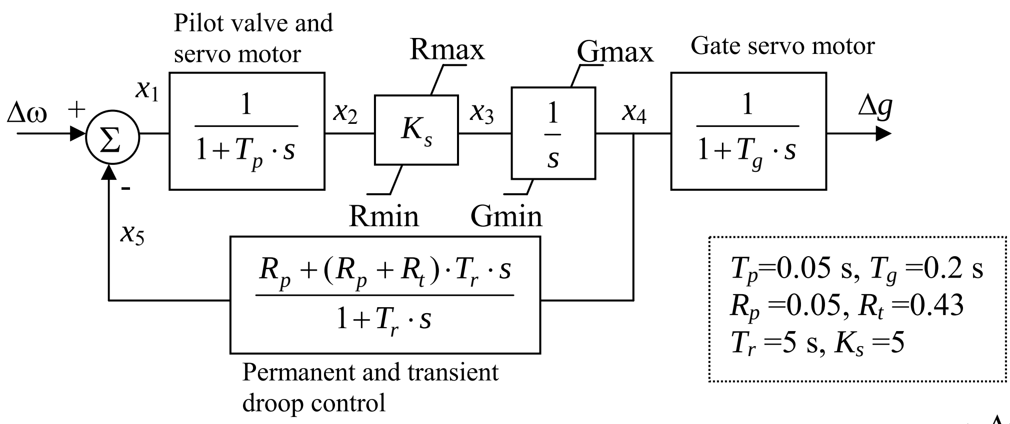


Fig. 6.26 – Machine control of Winsyn, autoinitialize synchronous machine (`Exa_17.acp`).

6.4.3.1 Hydro turbine governor

The gate opening limits must be adjusted to take the steady-state condition into account and Gmin=-1 is set in this case to allow 1 pu increase and reduction in torque. Also the initial head h0 is set to zero here.



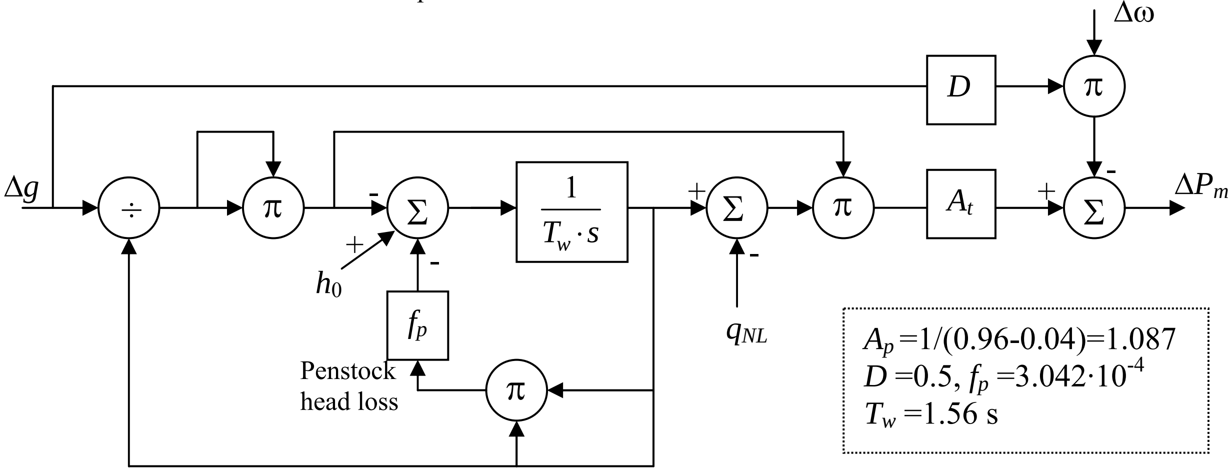


Fig. 6.27 – Hydro turbine governor

```
MODEL TUR_GOV
DATA Tw,D, gFL, gNL, fp, Rp,Tr,Rt,Tg,Tp,Ks
     Rmaxclose,Rmaxopen,Gmax,Gmin,MW,Wrated, Wref
INPUT W
OUTPUT Torque
VAR x1,x2,x3,x4,Pmech,At,x5,h,q,qNL,h0,s
     y1,y2,g,h1,Wrefpu,Wpu,torque
HISTORY
x1 {dflt:0},x2 {dflt:0},x3 {dflt:0},x4 {dflt:0}, x5 {dflt:0},
q {dflt:0},h {dflt:0}, y1 {dflt:0},y2 {dflt:0},g {dflt:0}
INIT
  h0:=0  --Initial head. Set to zero in case of auto-initiation of generator.
  At:=recip(gFL-gNL)
  qNL:=(gNL)*sqrt(h0)
  Wrefpu:=(Wref/Wrated)
ENDINIT
EXEC
  Wpu:=(W/Wrated)*(30/pi)
  --Governor hydraulic turbin
  x1:= Wrefpu-Wpu-x5
  cLaplace(x2/x1):=(1|s0)/(1|s0+Tp|s1)
  x3:=(Ks*x2) {min:Rmaxclose max:Rmaxopen}            --Gate opening/closing rate
  cLaplace(x4/x3) {dmin:Gmin dmax:Gmax}:=(1|s0)/(1|s1)  --Gate position
  cLaplace(g/x4):=(1|s0)/(1|s0+Tg|s1)                 --Gate servo motor
  cLaplace(x5/x4):=(Rp|s0+((Rp+Rt)*Tr)|s1)/(1|s0+Tr|s1) --Permanent and transient droop
  --Hydraulic turbin
  cLaplace(q/y1):=(1|s0)/(Tw|s1)                      --q=Flow
  h:=(q*recip(g))**2                                  --h=Head
  h1:=(q*q)*fp                                        --Penstock head loss
  y1:=(h0-h-h1)                                       --Change in head
  y2:=(q-qNL)*h                                       --Change in mechanical power
  Pmech:=At*y2+(g*D*(Wrefpu-Wpu))
  --Pmech:=g                                          --Uncomment to turn off turbine
  Torque:=(Pmech*recip(W))*MW*1e6
ENDEXEC
ENDMODEL
```

6.4.3.2 Exciter model

The Exciter is of type IEEE ST1A with inputs; terminal voltage V_T, field current I_{FD}, reference voltage V_{ref} and stabilizer signal V_S (all signals in pu). The Exciter IEEE DC1A is also implemented for comparison.

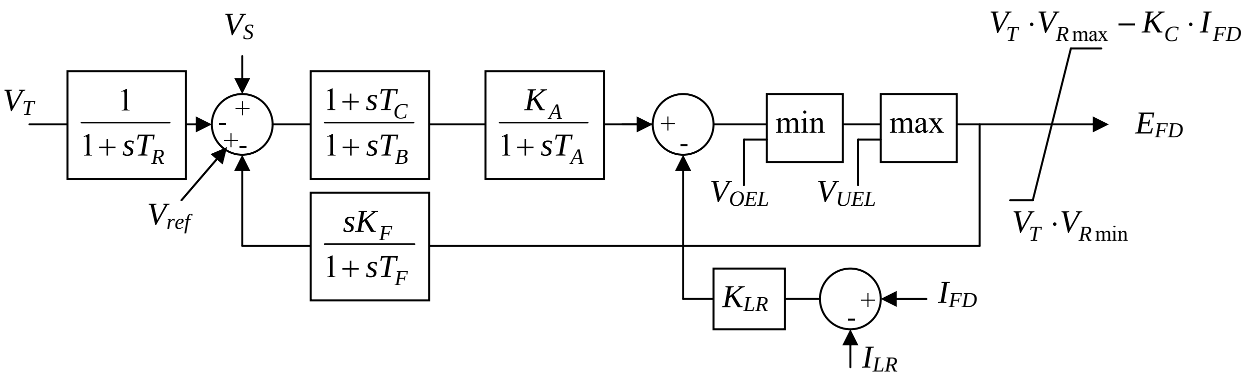


Fig. 6.28 – IEEE ST1A exciter. Parameters used; T_R=0.04, T_B=10, T_C=1, K_A=190, T_A=0, T_F=1, K_F=0, K_{LR}=0, I_{LR}=5, $V_{R\max}$=7.8, $V_{R\min}$=-6.7, K_C=0.08.

The exciter model ST1A requires the field current as input. This variable can be obtained directly from the Windsyn component as it is used there in the TACS section. The name of the TACS variable is 'IE1C*n*', where *n* is the machine number (1 in this case). To get the machine number, open the Windsyn component and read the machine number field (cannot be set). Then click on the IFD node of the exciter model and specify the node name IE1C1 and input type TACS as shown in Fig. 6.29.

Fig. 6.29 – How to get the field current into Models, and how to specify the Vs and VT nodes.

Windsyn does not allow field voltage regulation before 1 sec. The field connections are modeled as shown in Fig. 6.30 with 0.01 Ω separating resistors. The initial field voltage can be found by setting the the external field voltage to zero and then measure the current through it. This special trick is illustrated in the ST1A model, but not actually used in this example.

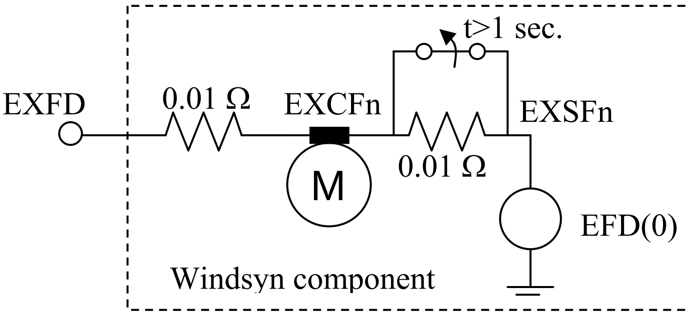


Fig. 6.30 – Internal field winding connections in Windsyn (n=Machine number)

```
MODEL  EX_ST1A
DATA   Vref,VTpu,Tr,Tc,Tb,Ka,Ta,Vuel,Voel,   Klr,   Ilr,   Kf,   Tf,   VRmax,VRmin,   Kc,
EFDref,IFDref
INPUT VT, Ifd, Vs, If0
OUTPUT Efd
VAR x1,x2,x3,x4,x5,x6, Efd,Vc,IFDpu,Efd0
HISTORY
x1 {dflt:0},x2 {dflt:0},x3 {dflt:0},x4 {dflt:0},x5 {dflt:0}, x6 {dflt:0},
Vc {dflt:0}, VT {dflt:0}
INIT
  Efd:=0
ENDINIT
EXEC
  if T<2*timestep then  --Special trick to obtain the initial field voltage
    Efd0:=-If0*0.01
  else
    IFDpu:=-IFD/IFDref
    --Vc:=VT/(1+Tr)
    cLaplace(Vc/VT):=(1/VTpu|s0)/(1|s0+Tr|s1)
    cLaplace(x6/x5):=(Kf|s1)/(1|s0+Tf|s1)
    x1:=Vref-Vc-Vs-x6
    cLaplace(x2/x1):=(1|s0+Tc|s1)/(1|s0+Tb|s1)
    cLaplace(x3/x2):=(Ka|s0)/(1|s0+Ta|s1)
    x4:=x3-(IFDpu-Ilr)*Klr
    x5:=max(x4,Vuel)
    x5:=min(x5,Voel)
    Efd:=x5 {min:VT/VTpu*VRmin max:VT/VTpu*VRmax-Kc*IFDpu}
    Efd:=Efd*EFDref+0*Efd0                    --Efd0 not used here
  endif
ENDEXEC
ENDMODEL
```

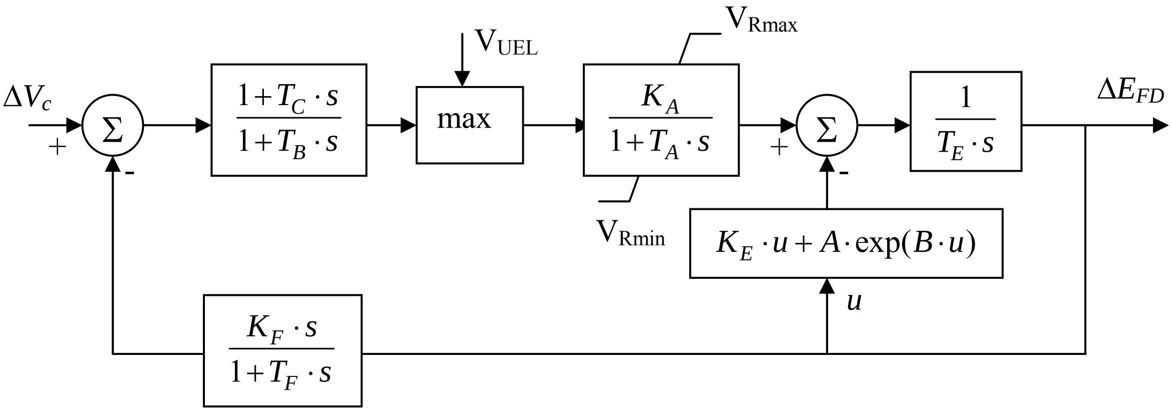


Fig. 6.31 – IEEE DC1A exciter. Parameters used; T_B=0.06, T_C =0.173, K_A=400, T_A =0.89, T_E =1.15, K_E =1, A =0.014, B = 1.55, K_F =0.058, T_F=0.62.

```
MODEL EX_DC1A
DATA Vref,Tc,Tb,Ka,Ta,VRMAX,VRMIN,Kf,Tf,Te,Ke,Vuel,A,B, Efdbase
INPUT Vc
OUTPUT Efd
VAR x1,x2,x3,x4,x5,x6, Vfe,Vf,Efd,Vcpu
HISTORY
x1 {dflt:0},x2 {dflt:0},x3 {dflt:0},x4 {dflt:0},x5 {dflt:0}, x6 {dflt:0}, Vfe {dflt:0}
Vf {dflt:0}
INIT
  Efd:=0
ENDINIT
EXEC
  Vcpu:=(Vc*sqrt(3)/(Vref*1000))  --Phase voltage measured so scale to line voltage
  x1:=(1-Vcpu-Vf)
  cLaplace(x2/x1):=(Tc|s1+1|s0)/(Tb|s1+1|s0)
  x3:=max(x2,Vuel)
  cLaplace(x4/x3) {dmin:VRMIN dmax:VRMAX}:=(Ka|s0)/(Ta|s1+1|s0)
  x5:=(x4-Vfe)
  cLaplace(x6/x5):=(1|s0)/(Te|s1)
  Vfe:=x6*Ke+A*exp(B*x6)
  cLaplace(Vf/Vfe):=(Kf|s1)/(Tf|s1+1|s0)
  Efd:=x6*Efdbase
ENDEXEC
ENDMODEL
```

6.4.3.3 RMS value calculation

The RMS value is calculated by a standard models provided by Laurant Dube. Since the speed of the generator changes the frequency is calculated by another model. The *MODELS|Default model* option was used and the text simply pasted into the Model component. *Edit|Flip* was used to switch the input and outputs. As this model gives its output to another model it must be written first to the ATP file. This is managed by giving it a lower *Order* number than the receiving model and then choose *ATP|Settings/Format – Sort by Order*. In the receiving model the input node must be set to *Input MODEL*.

```
MODEL rms_meter
  DATA   freq      -- base frequency
         xrms_ini {dflt:-1} -- initial rms value
  INPUT  x         -- monitored signal
  VAR    xrms      -- rms value of monitored signal
         x2        -- internal, x*x
         ix2       -- internal, integral of x2
         period    -- 1/freq
  OUTPUT xrms
```

```
  DELAY CELLS(ix2): 1/freq/timestep +1
  INIT
    period       := recip(freq)
    histdef(ix2) := 0
    integral(x2) := 0
    IF xrms_ini <0 THEN xrms:=0 ELSE xrms:=xrms_ini ENDIF
  ENDINIT
  EXEC
    x2  := x*x
    ix2 := integral(x2)
    IF t>period THEN
      xrms:= sqrt((ix2 - delay(ix2, period))/period)
    ENDIF
  ENDEXEC
ENDMODEL
```

The frequency is calculated by another model based on zero-crossing detection.

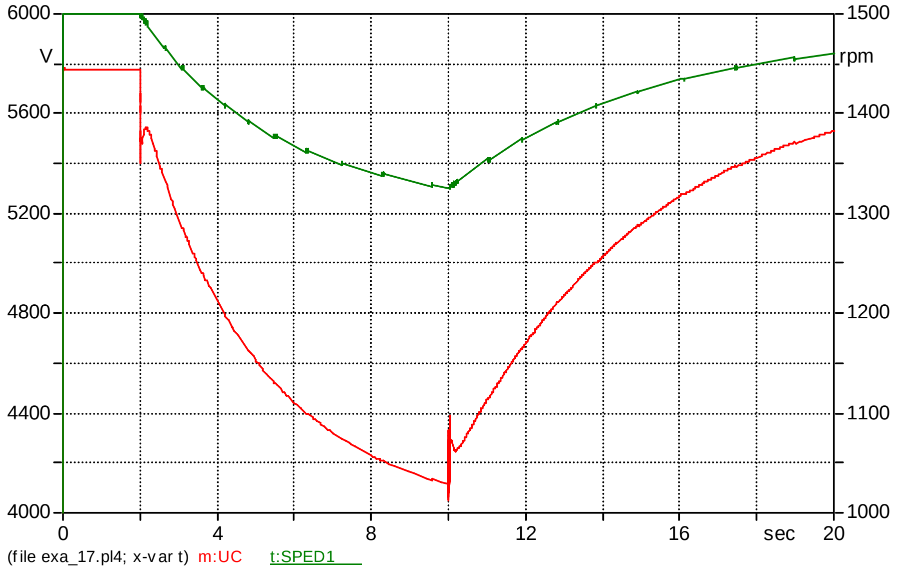


Fig. 6.32 – Machine response with no regulation

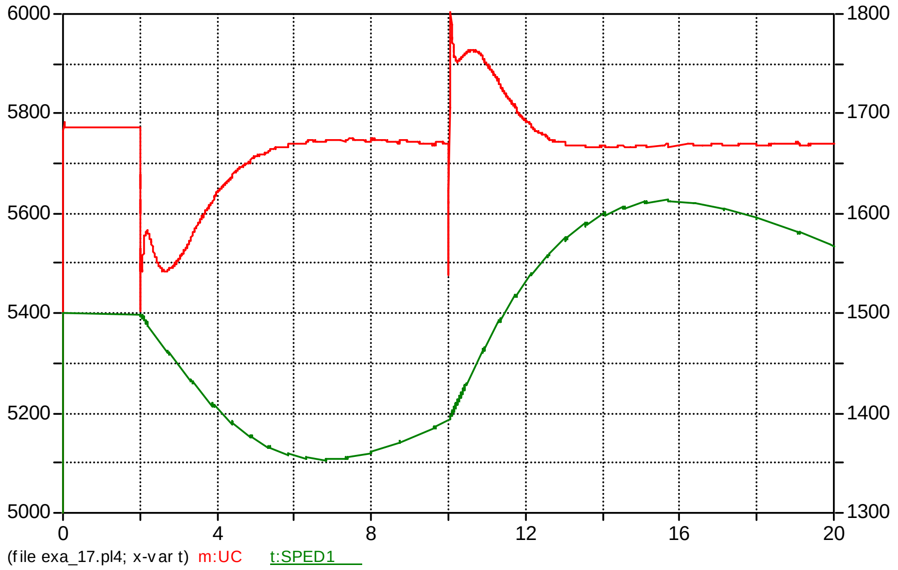


Fig. 6.33 – Machine response with exciter (DC1A) and governor (no hydro turbine).

6.5 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.

6.5.1 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. 6.34, the corresponding ATPDraw circuit is depicted in Fig. 6.35.

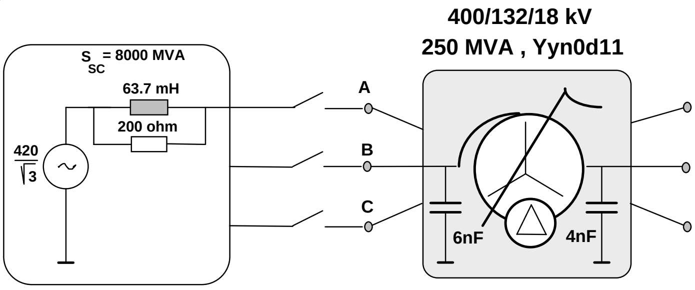


Fig. 6.34 - One-line scheme of the transformer and the 400 kV source.

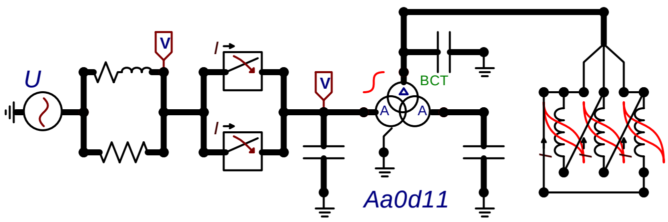


Fig. 6.35 - ATPDraw circuit (`Exa_10.acp`).

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. 6.36). 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.

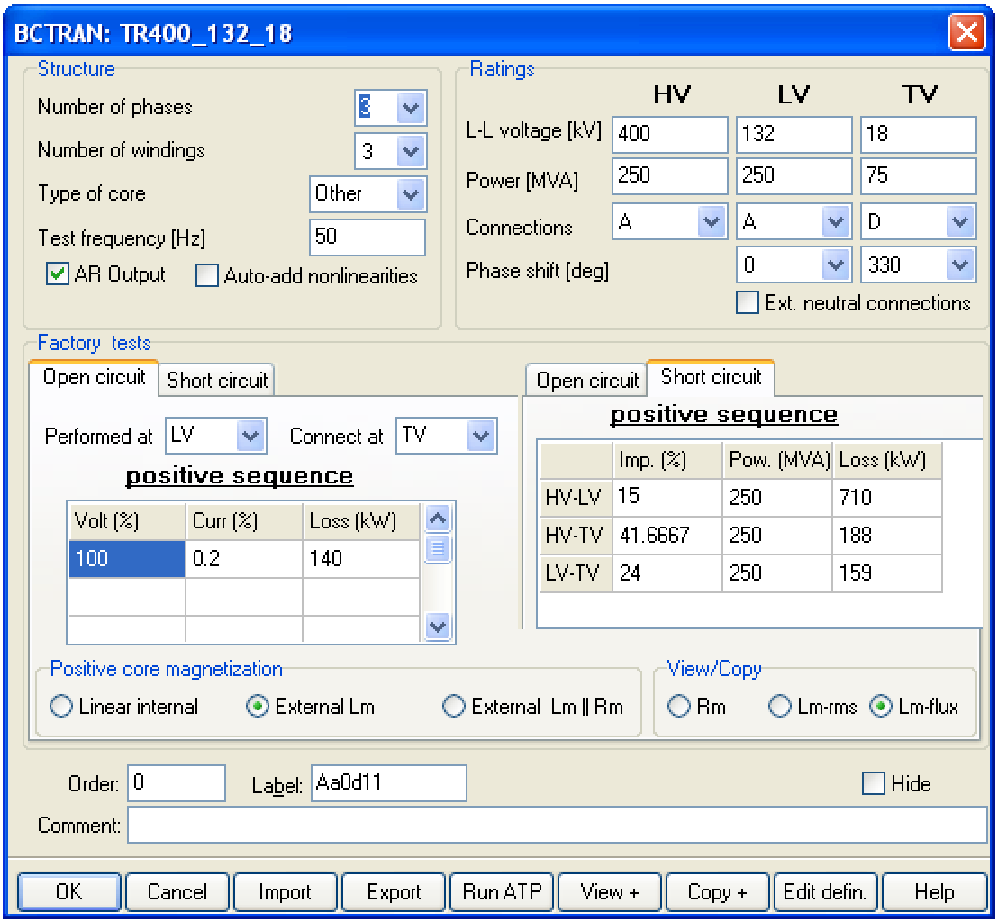


Fig. 6.36 - BCTRAN dialog box of the 400/132/18 kV transformer.

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. 6.37 shows the hysteresis loop of the Itype-1 material of ATP and of the magnetic core of the transformer.

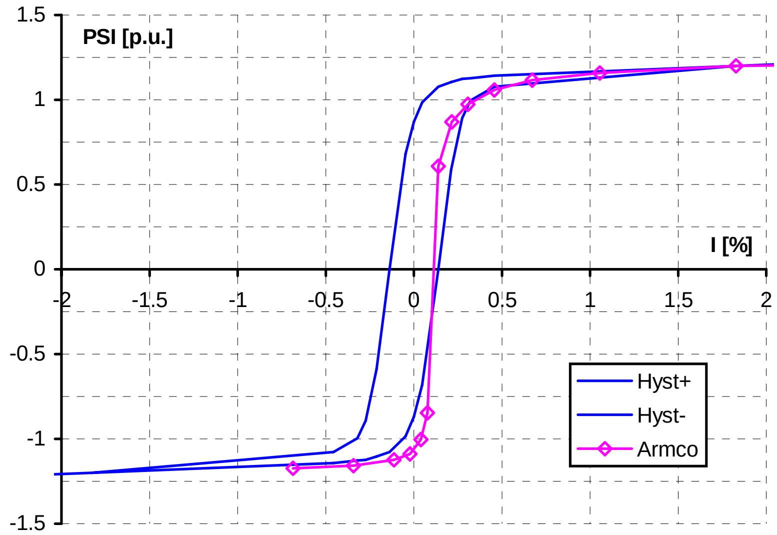


Fig. 6.37 - The shape of the hysteresis loop of the transformer magnetic core compared with the material type 1 of ATP's HYSDAT supporting routine.

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. 6.38. 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 64. No such limit exists for the included nonlinear characteristics.

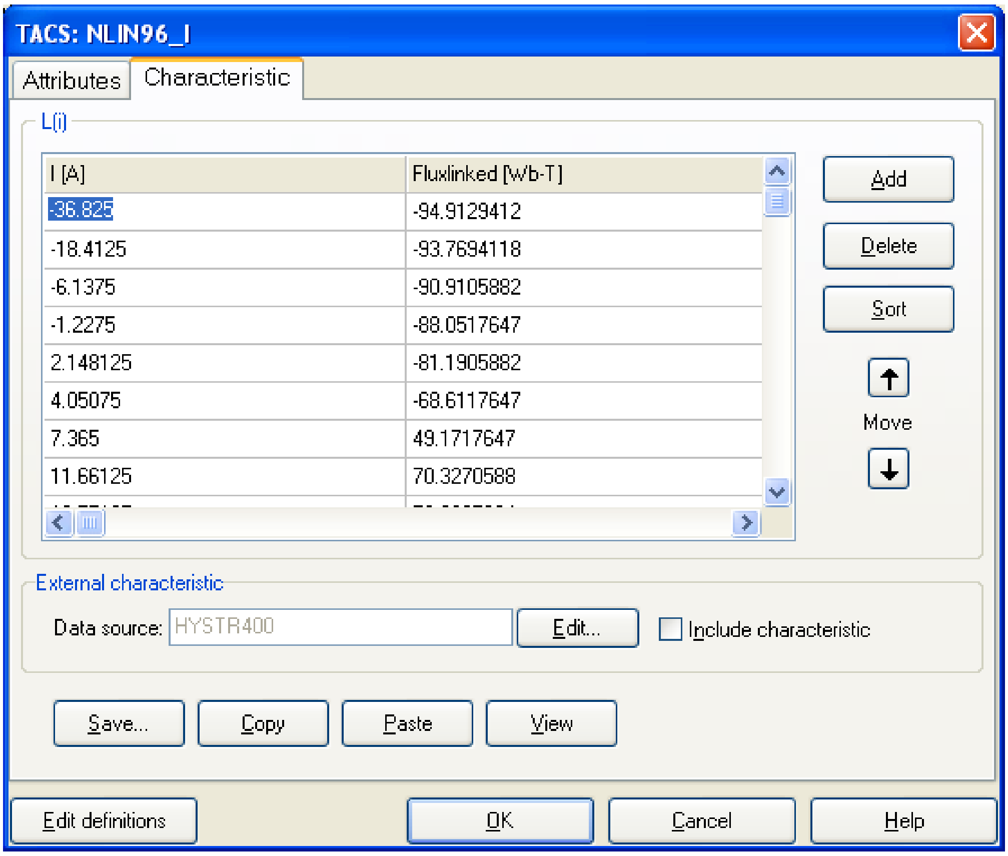


Fig. 6.38 - Importing the nonlinear characteristic from a HYSDAT punch-file.

The complete ATP input file generated by ATPDraw for this study case is listed next:

```
BEGIN NEW DATA CASE
C 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  >
        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. 6.39. 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.

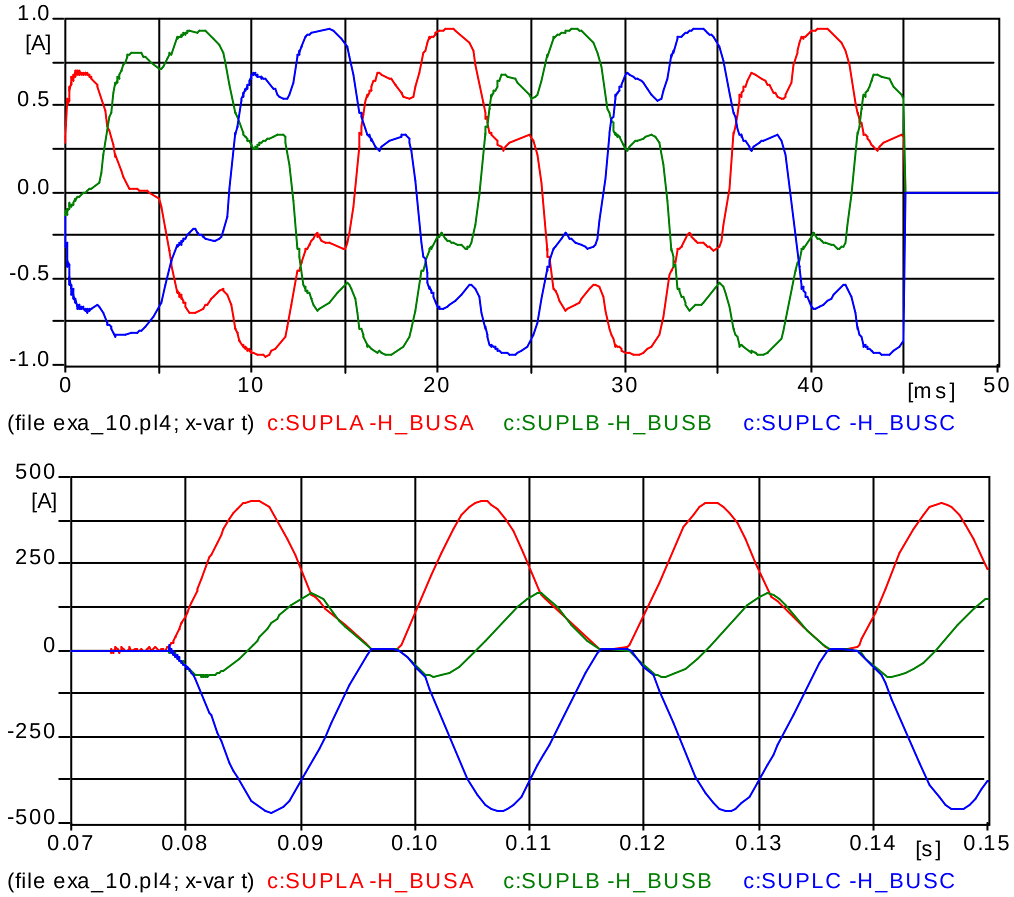


Fig. 6.39 - Steady-state magnetizing current (upper curves) and the inrush current (lower curves) at a subsequent energization.

6.5.2 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. 6.40.

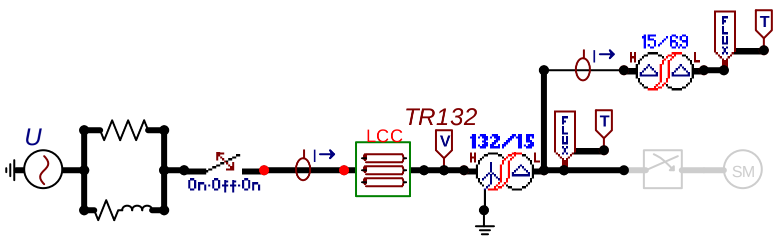


Fig. 6.40 - ATPDraw circuit (`Exa_11g.adp`)

Fig. 6.40 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. 6.40 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. 6.41. Therefore, you can see how the grouping feature makes the circuit more readable.

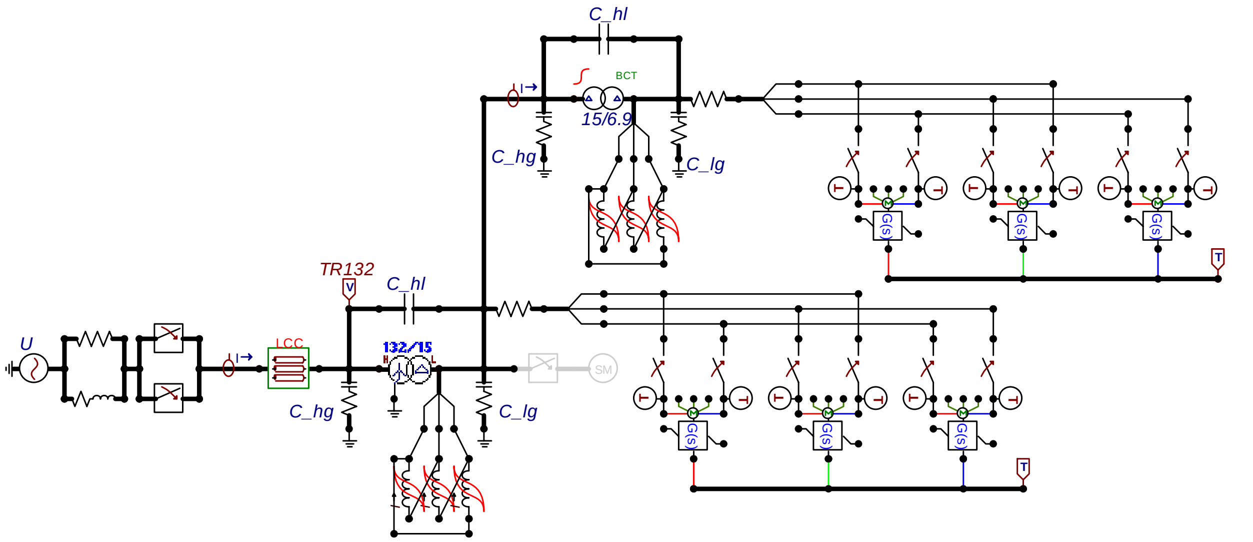


Fig. 6.41 - ATPDraw circuit without using compress (`Exa_11.adp`).

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 has been 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. 6.42.

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. 6.43 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. 6.44).

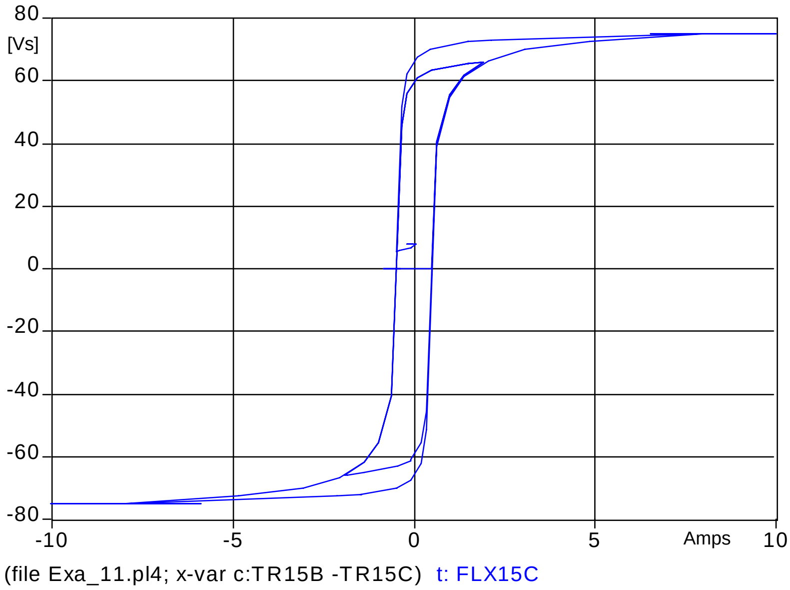
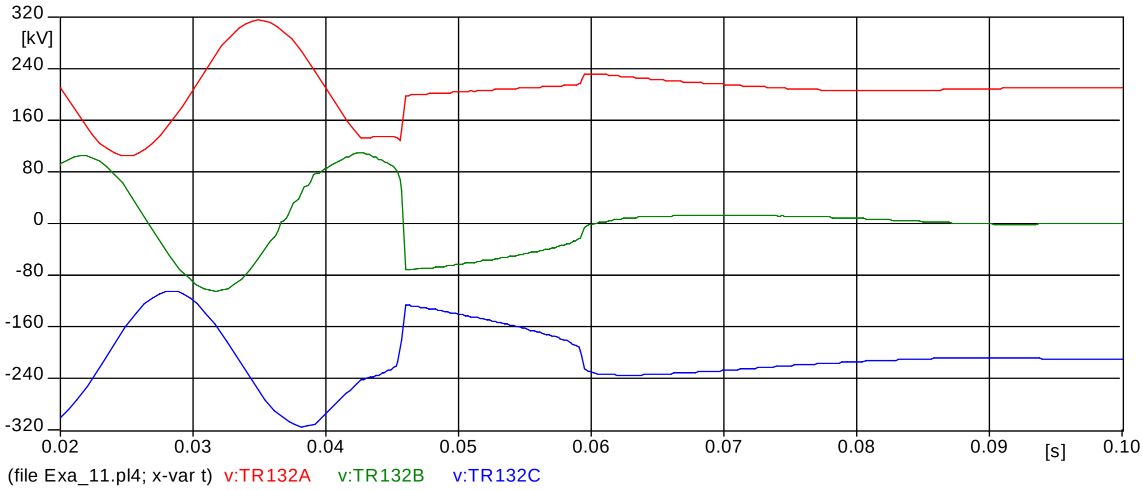


Fig. 6.42 - Roaming of the operating point on the hysteresis loop in steady-state and during the subsequent non-sinusoidal oscillations at transformer de-energization.



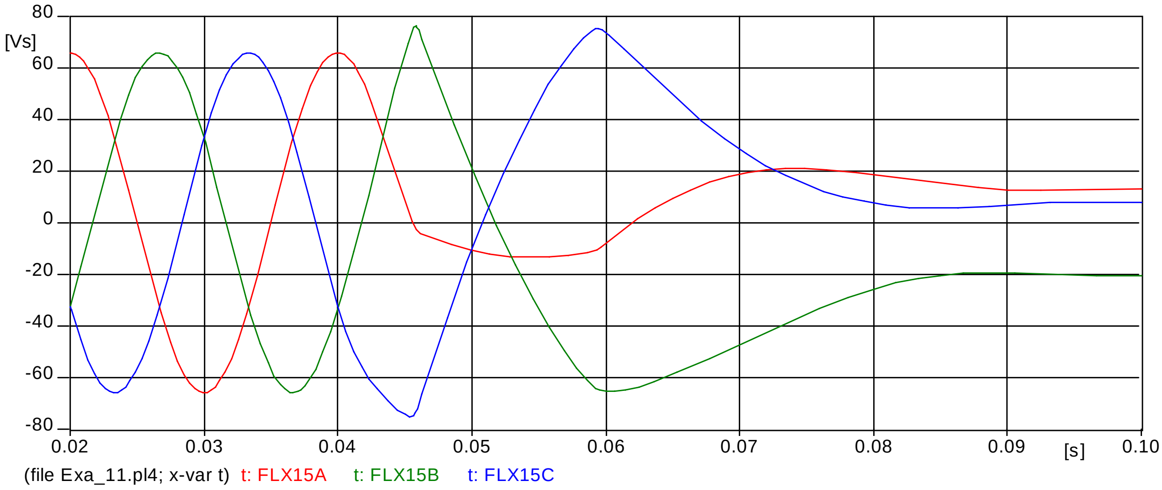


Fig. 6.43 - Non-sinusoidal voltage oscillations appear after de-energizing the step-up transformer (upper curves). The residual flux is less then 30% in each phases (lower curves).

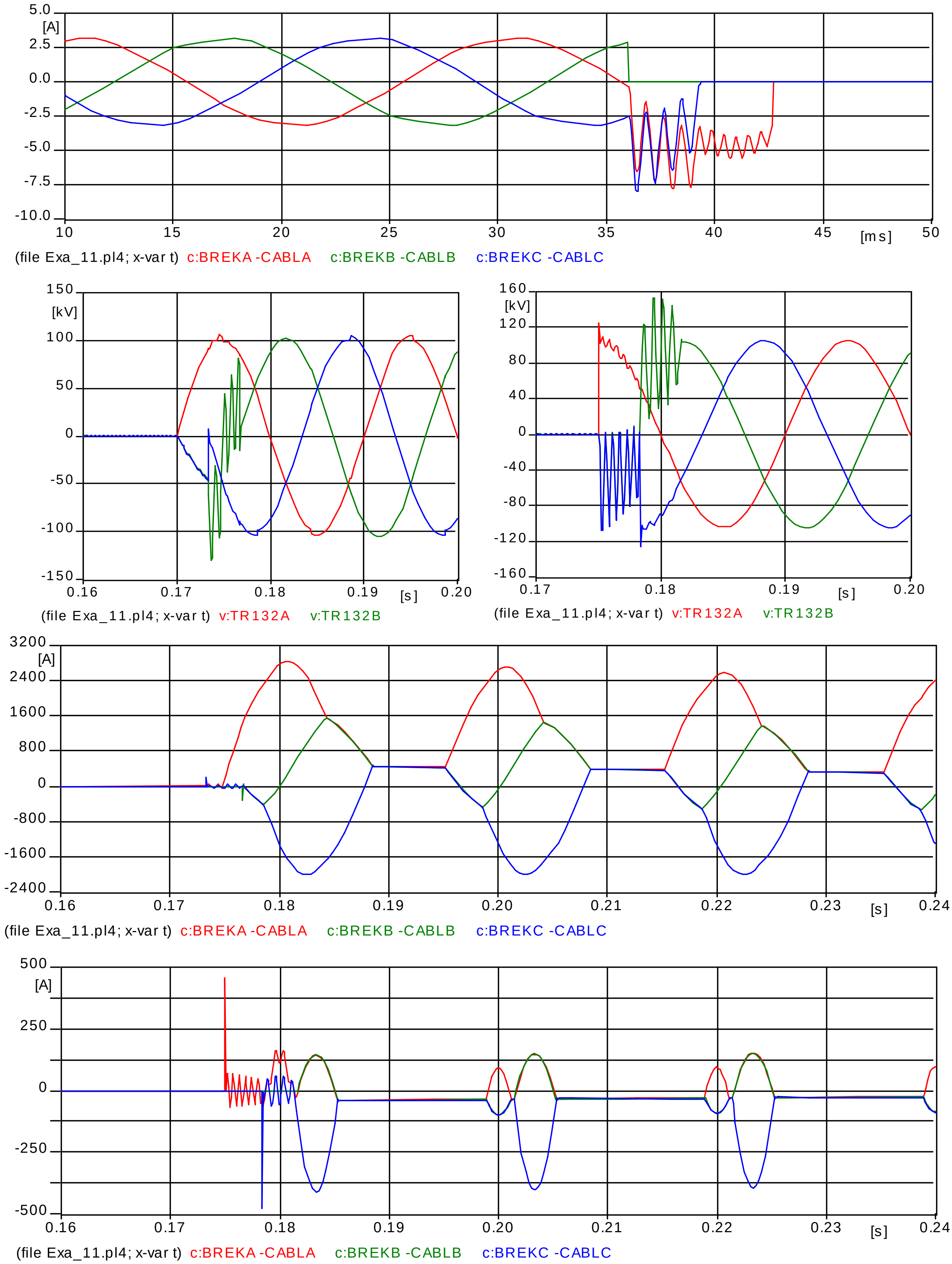


Fig. 6.44 - Interrupting the steady state no-load current of the step-up transformer (upper curves) and the inrush current amplitude (below) when energizing the first pole of the breaker: a) at the voltage zero crossing, b) close to the voltage peak.

6.5.3 Using the Hybrid Transformer component *(Exa_16.acp)*

The Hybrid Transformer component (XFMR) provides a topologically correct core model with individual saturation characteristics in legs and yokes calculated based on relative core dimensions. Further the saturation characteristic is based on the Frolich equation with an additional, optional air-core inductance thus improving the response above the last test report value. This is of great importance when it comes to over-excitation situations like inrush current simulations. The XFMR component in version 5.6 offers type 96 inductances even if these are not recommended for transient studies. This gives on the other hand residual flux in the core after de-energization. In general advance Models controlled hysteretic inductors are needed to give good inrush current predictions.

Fig. 6.45 shows the XFMR input dialog for the example `Exa_16.acp`. A 3-legged stacked core is selected and this requires relative yoke dimensions to be given under Core data. A Triplx core (single phase units) does not require relative dimensions. Under *Inductance* and *Core* the short and open circuit test report data are given, respectively (*Resistance* automatically follow *Inductance* for Test Report data). The *Winding sequence* is set with the low-voltage winding as the inner. The XFMR dialog can work test report data directly. Creation of the saturation characteristics is automized (for type 96 half of the core losses is assigned to hysteresis losses with a Steimetz coefficient *n*=2, and a uniform width of the hysteresis).

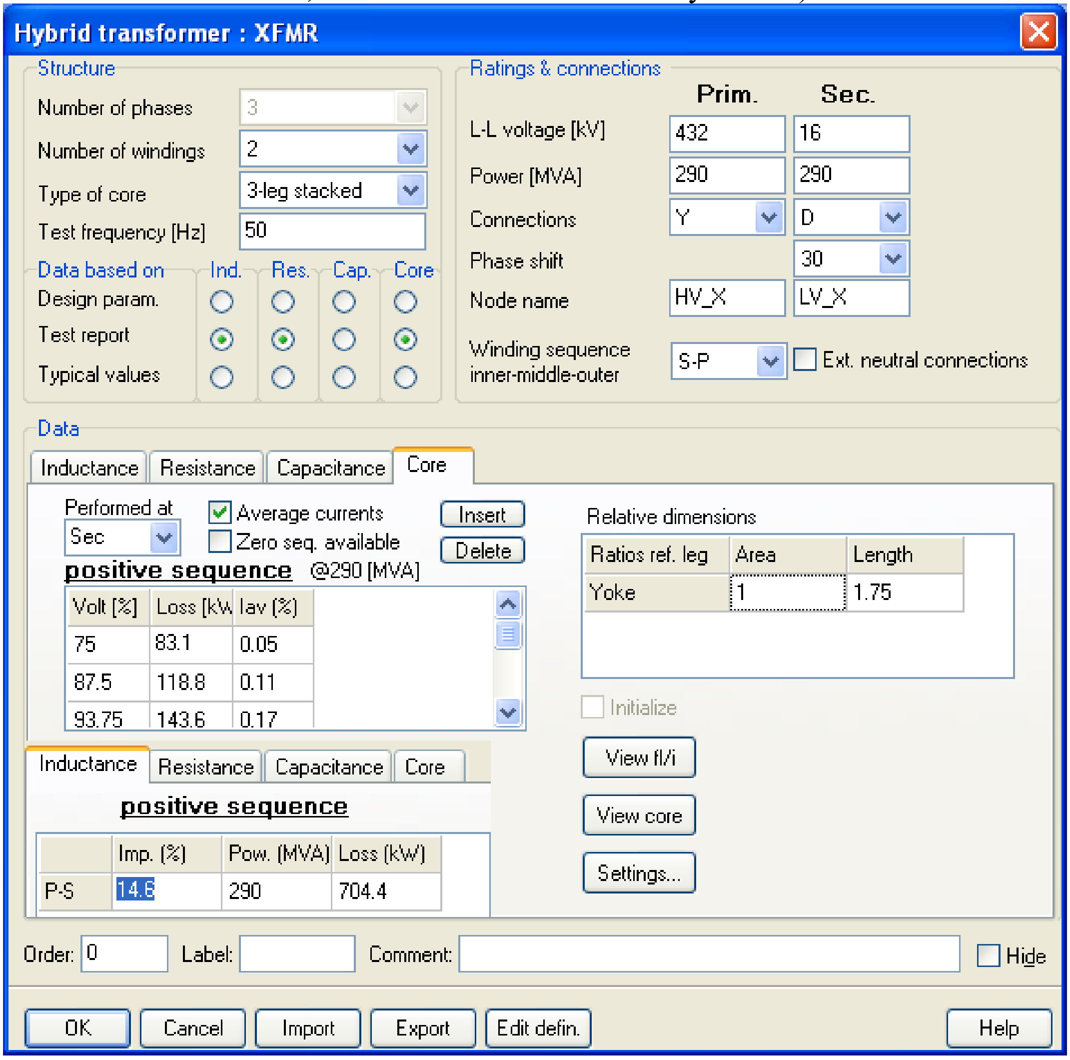
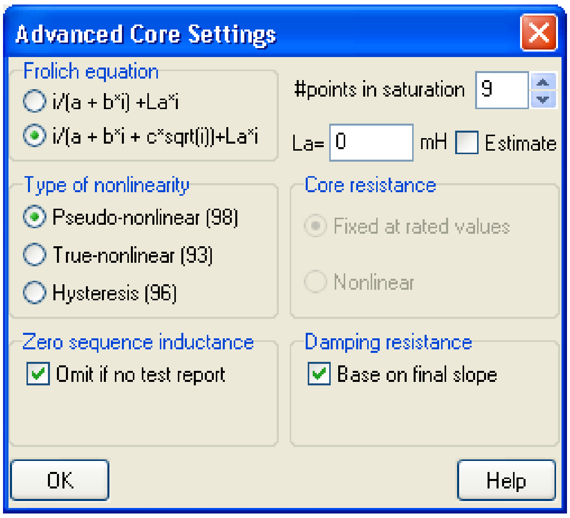


Fig. 6.45 – XFMR model example `Exa_16.acp`

When the user clicks on OK ATPDraw performs an internal calculation of the leakage inductance in the same way as BCTRAN. The winding resistances are connected outside the A-matrix, however. The core model is fitted to the Test Report rms values by a Gradient Method optimization routine.



The user should also click on the *Settings…* button on the *Core* page to select the type of nonlinear inductance (98, 93, or 96) and the number of points on linearized Frolich equation (maximum 9). A high number is required to get good inrush current estimates. The final slope inductance (part of the air-core inductance) is set to zero in this case. Design data really required to estimate it. Using the *Estimate* check box will estimate La=$\mu_0 \cdot 6/a'$ where the factor a=6 is typical for core material M4 and a' is found from the optimization (with '=0)

Fig. 6.46 – Core settings.

Fig. 6.47 shows a simulated inrush currents switching in a 290 MVA transformer from the 16 kV side with zero residual flux. The same transformer is modeled both in BCTRAN and XFMR and the comparison shows that the XFMR gives about four times higher inrush currents. This is because the BCTRAN model incorrectly assumes linear extrapolation of the magnetization characteristic above the Test Report data. In addition the currents into the XFMR model have more reasonable waveshapes and attenuation.

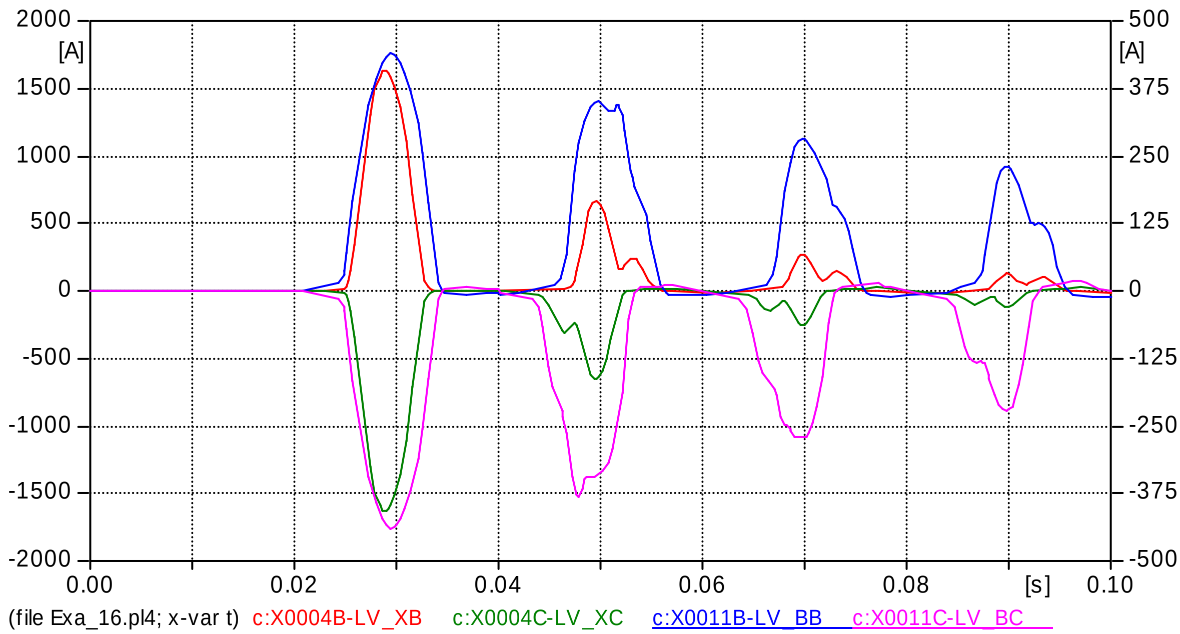


Fig. 6.47 – Comparizon of inrush currents (zero residual flux) for a 290 MVA transformer modeled in BCTRAN and XFMR.

6.6 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. 6.48 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.

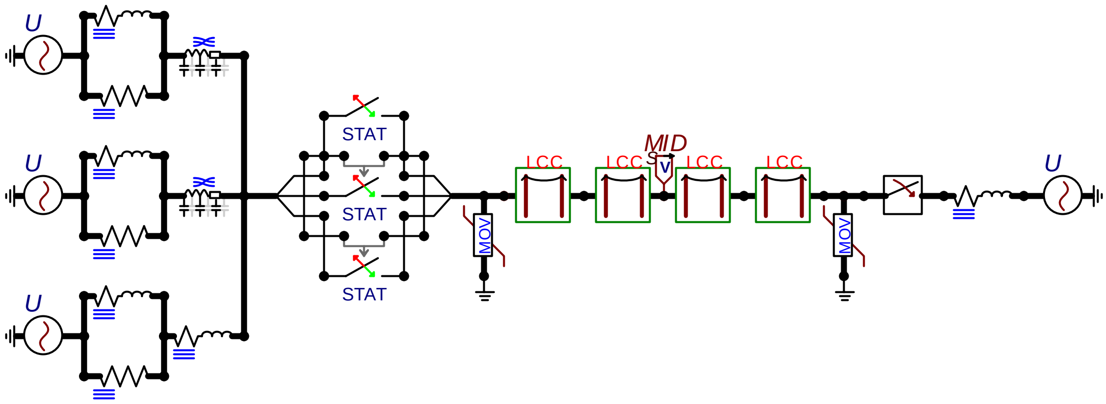


Fig. 6.48 - ATPDraw circuit for the statistical switching study (`Exa_12.adp`).

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.

6.6.1 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. ATPDraw ensured automatically this ordering. This is why the closing of the dialog box of a master switch is somewhat delayed.

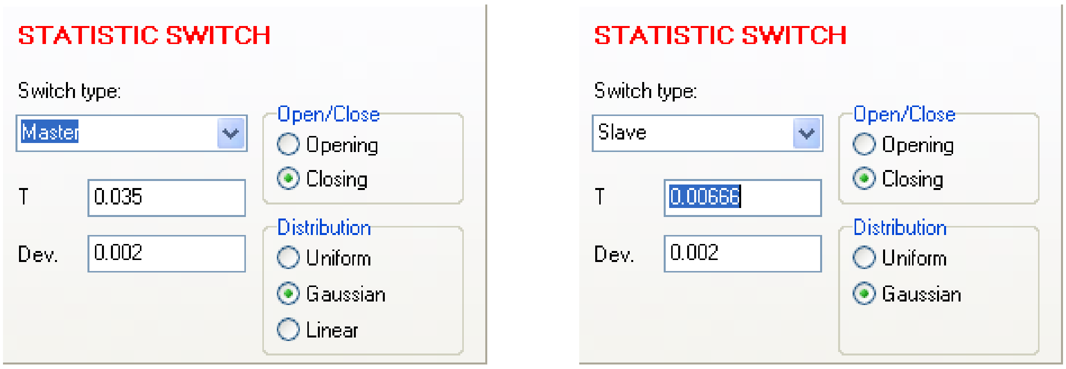
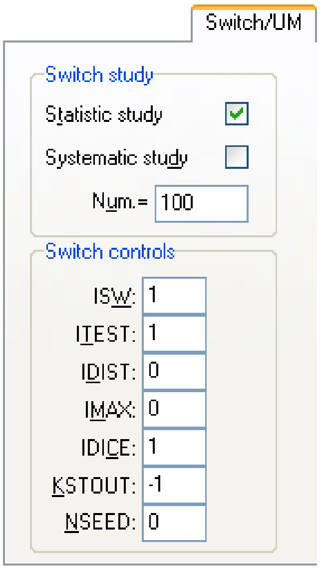


Fig. 6.49 - Input parameters of master and slave statistical switches.



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.

Fig. 6.50 - Setting the parameters of the statistical study.

The ***Output Manager*** found under *ATP|Output Manager (F9)* enables the user to select those output requests to be added to the statistical tabulation. The user can also group and scale the output requests. Example 12 requests as default only output of the MID voltage, but the terminal voltages and for instance surge arrester energy can be added. The selection of alternative statistical tabulation is shown in Fig. 6.52.

6.6.2 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 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. 6.51 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.

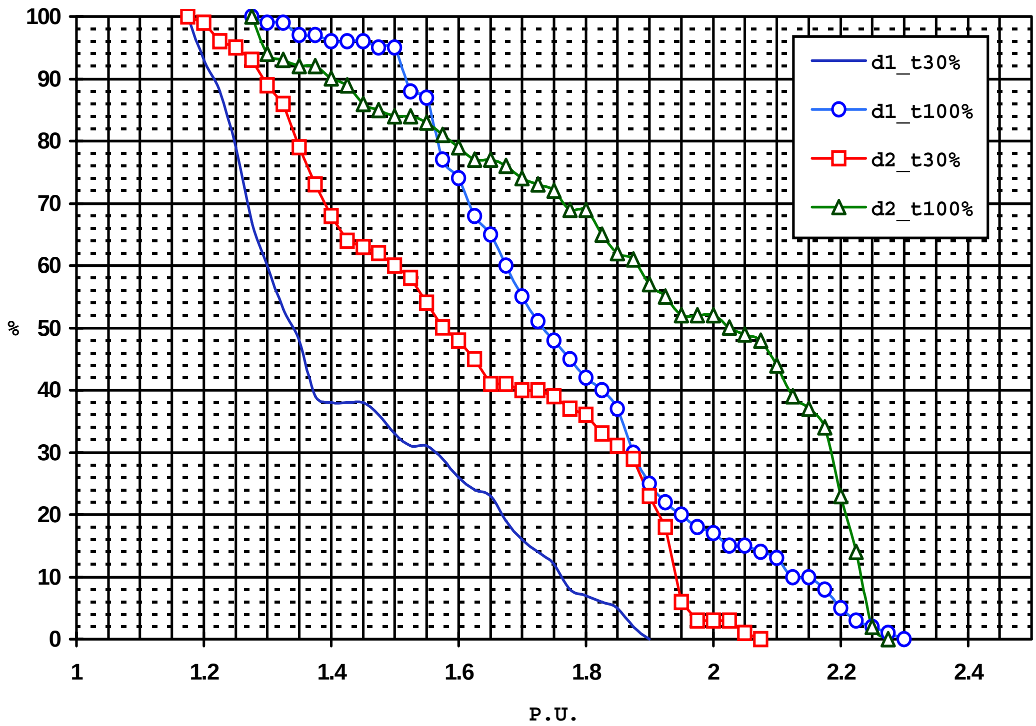
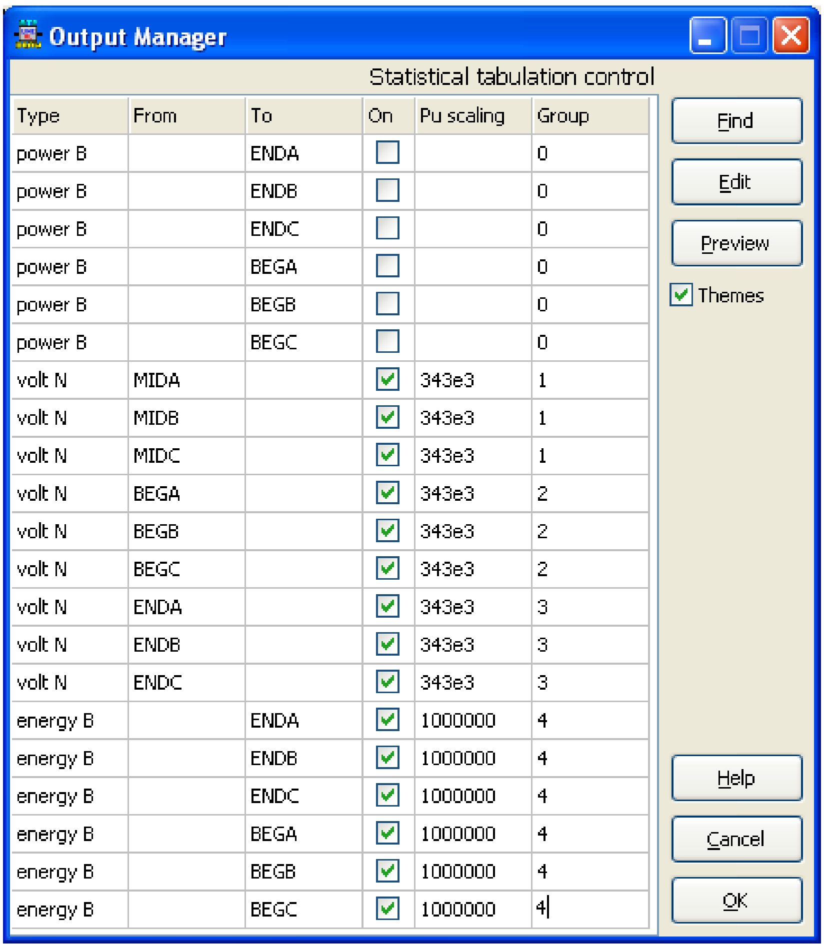


Fig. 6.51- Probability distribution function of the 3-phase reclosing overvoltages.



Type	From	To	On	Pu scaling	Group
power B		ENDA	☐		0
power B		ENDB	☐		0
power B		ENDC	☐		0
power B		BEGA	☐		0
power B		BEGB	☐		0
power B		BEGC	☐		0
volt N	MIDA		☑	343e3	1
volt N	MIDB		☑	343e3	1
volt N	MIDC		☑	343e3	1
volt N	BEGA		☑	343e3	2
volt N	BEGB		☑	343e3	2
volt N	BEGC		☑	343e3	2
volt N	ENDA		☑	343e3	3
volt N	ENDB		☑	343e3	3
volt N	ENDC		☑	343e3	3
energy B		ENDA	☑	1000000	4
energy B		ENDB	☑	1000000	4
energy B		ENDC	☑	1000000	4
energy B		BEGA	☑	1000000	4
energy B		BEGB	☑	1000000	4
energy B		BEGC	☑	1000000	4

Fig. 6.52 – Output Manager and alternative request of Statistical Tabulation.

7. Appendix . . . .

ATPDraw™
for Windows
5.6

7.1 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

$$Q_+ = \frac{-V^2 \cdot n}{4 \cdot \sin^2(\pi / n)} \mathrm{Im}\left(I_+\right)$$

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):

$$Q_+ = \frac{-V^2 \cdot 2}{n}\left(Q_1 + Q_2 + \cdots + Q_n\right)$$

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

$$Q_0 = \frac{-V^2 \cdot n}{4 \cdot \sin^2(\pi / n)} \operatorname{Im}(I_0)$$

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

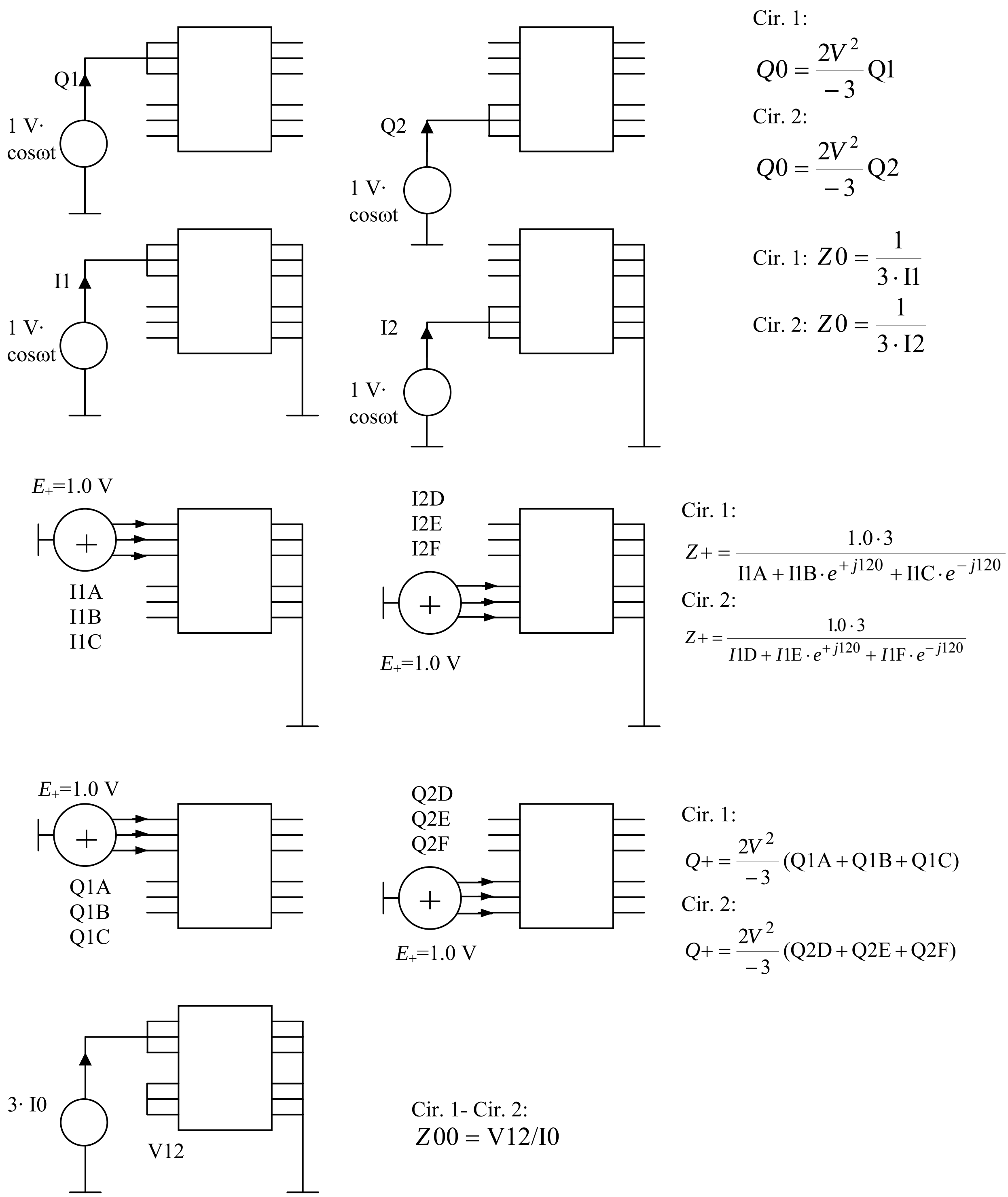
$$Q_0 = \frac{-V^2 \cdot 2}{n}(Q)$$

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:



Zero sequence short circuit impedance: (real and imaginary part). $Z_0 = R_0 + jX_0$.

Fig. 7.1 – LCC-Verify; Power Frequency Calculations.

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.

7.2 Line Check

When performing transient analysis of power systems, high frequency models of overhead transmission lines and underground cables must be developed. In this process, parameters like ground and conductor conductivity, cross-section geometry, and average overhead line height could be uncertain and questionable. Very often the only reliable benchmark data are sequential parameters at power frequency. It is thus of great interest to be able to verify the developed line/cable model at power frequency before simulating and analyzing transients. The present version of ATPDraw has in the LCC-module a built in option to verify a line segment [1]. This is done by calculating the short circuit input impedances and the open circuit reactive power consumption. In addition a frequency scan is supported. However, data for each line segment is rarely available, and in addition one would prefer to verify an entire line/cable length including the effect of transpositions. Instead of calculating the short circuit input impedance and the open circuit reactive power consumption it would be better to obtain the serial impedance and the shunt admittance along with the average mutual impedance and admittance between circuits in 6-phase and 9-phase cases. The new module integrated in ATPDraw involves an improved handling of the equivalent mutual coupling between circuits.

7.2.1 Single phase systems

Initially, consider a single-phase circuit of length l with frequency domain distributed series impedances and shunt admittances, as shown in Fig. 7.2. The line is spited in segments of length dx.

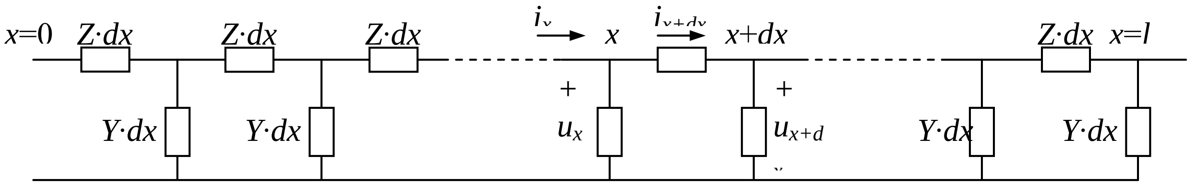


Fig. 7.2 – Single phase representation of transmission line. $Z = R + j\omega L$ [Ω/m], $Y = G + j\omega C$ [S/m].

The currents and voltages at the sending and receiving ends will not be equal. The idea is further to use the measured quantities at both terminals to obtain the series impedance and shunt capacitance. Current balance at point x results in $-\frac{\partial i}{\partial x} = Y \cdot u$. The voltage drop between x and $x+dx$ gives $-\frac{\partial u}{\partial x} = Z \cdot i$. These two equations result in the wave equation $\frac{\partial^2 u}{\partial x^2} = Z \cdot Y \cdot u$ with the solution $u(x) = A \cdot e^{\gamma \cdot x} + B \cdot e^{-\gamma \cdot x}$, where the constants A and B are determined from the boundary conditions and $\gamma = \sqrt{Z \cdot Y}$. The current is $i(x) = -Z^{-1} \cdot \frac{\partial u}{\partial x} = -Z^{-1} \cdot \gamma \cdot \left(A \cdot e^{\gamma \cdot x} - B \cdot e^{-\gamma \cdot x}\right)$

1. Short circuit case:

This is the typical configuration for obtaining the series impedance. A sinusoidal voltage or current is applied at the sending end while the receiving end is grounded.

$u(0) = U_0$ and $u(l) = 0$ gives

$A + B = U_0$ and $A \cdot e^{\gamma l} + B \cdot e^{-\gamma l} = 0$ which result in

$$u(x) = U_0 \cdot \frac{\sinh \gamma \cdot (l - x)}{\sinh \gamma \cdot l} \quad \text{and} \quad i(x) = U_0 \cdot Z^{-1} \cdot \gamma \cdot \frac{\cosh \gamma \cdot (l - x)}{\sinh \gamma \cdot l} \tag{1}$$

The currents at the terminals are

$$i(0) = U_0 \cdot Z^{-1} \cdot \gamma \cdot \frac{\cosh \gamma \cdot l}{\sinh \gamma \cdot l} \approx U_0 \cdot (Z \cdot l)^{-1} \cdot \left(1 + \frac{1}{3} \cdot (\gamma l)^2 - \frac{1}{45} \cdot (\gamma l)^4 \ldots\right) \quad \text{and} \tag{2}$$

$$i(l) = U_0 \cdot Z^{-1} \cdot \gamma \cdot \frac{1}{\sinh \gamma \cdot l} \approx U_0 \cdot (Z \cdot l)^{-1} \cdot \left(1 - \frac{1}{6} \cdot (\gamma l)^2 + \frac{7}{360} \cdot (\gamma l)^4 \ldots\right) \tag{3}$$

where the approximation comes from a series expansion of the hyperbolic functions.

The second quadratic term is eliminated in the following combination:

$$\tilde{\imath} = \frac{i(0) + 2 \cdot i(l)}{3} = U_0 \cdot (Z \cdot l)^{-1} \cdot \left(1 + \frac{1}{180} \cdot (\gamma l)^4 \ldots\right) \tag{4}$$

The total series impedance can thus be approximated by the following combination of the measured inputs and outputs:

$$Z_s = \left.\frac{3\cdot u(0)}{i(0)+2\cdot i(l)}\right|_{sc} = Z\cdot l\cdot\left(1-\frac{1}{180}\cdot(\gamma l)^4 ...\right) \approx Z\cdot l \quad [\Omega] \tag{5}$$

The same result is obtained if a current is applied at the sending end instead of a voltage.

2. Open circuit case:

This is the typical configuration for obtaining the shunt admittance. A sinusoidal voltage or current is applied at the sending end while the receiving end is left open.

$u(0) = U_0$ and $i(l) = 0$ gives

$A + B = U_0$ and $A\cdot e^{\gamma l} - B\cdot e^{-\gamma l} = 0$ which result in

$$u(x) = U_0\cdot\frac{\cosh\gamma\cdot(l-x)}{\cosh\gamma\cdot l} \text{ and } i(x) = U_0\cdot Z^{-1}\cdot\gamma\cdot\frac{\sinh\gamma\cdot(l-x)}{\cosh\gamma\cdot l} \tag{6}$$

The unknown terminal quantities are:

$$i(0) = U_0\cdot Z^{-1}\cdot\gamma\cdot\frac{\sinh\gamma\cdot l}{\cosh\gamma\cdot l} \approx U_0\cdot Y\cdot l\cdot\left(1-\frac{1}{3}\cdot(\gamma l)^2+\frac{2}{15}\cdot(\gamma l)^4 ...\right) \text{ and} \tag{7}$$

$$u(l) = U_0\cdot\frac{1}{\cosh\gamma\cdot l} \approx U_0\cdot\left(1-\frac{1}{2}\cdot(\gamma l)^2+\frac{5}{24}\cdot(\gamma l)^4 ...\right) \tag{8}$$

where the approximation again comes from a series expansion of the hyperbolic functions.

Similar to the short circuit case an equivalent voltage is defined as:

$$\tilde{u} = \frac{u(0)+2\cdot u(l)}{3} = U_0\cdot\left(1-\frac{1}{3}\cdot(\gamma l)^2+\frac{5}{36}\cdot(\gamma l)^4 ...\right) \tag{9}$$

The total shunt impedance can be approximated by the following combination of the measured inputs and outputs:

$$Y_s = \left.\frac{3\cdot i(0)}{u(0)+2\cdot u(l)}\right|_{oc} = \frac{Y\cdot l\cdot\left(1-\frac{1}{3}\cdot(\gamma l)^2+\frac{2}{15}\cdot(\gamma l)^4 ...\right)}{\left(1-\frac{1}{3}\cdot(\gamma l)^2+\frac{5}{36}\cdot(\gamma l)^4 ...\right)} = Y\cdot l\cdot\left(1-\frac{1}{180}\cdot(\gamma l)^4 ...\right) \approx Y\cdot l \quad [S] \tag{10}$$

The same result is obtained if a current is applied at the sending end instead of a voltage.

3. Comparison with input impedance/admittance

The short circuit input impedance and the open circuit input admittance (scaled to get reactive power in ATPDraw) is for comparison

$$Z_{in} = \left.\frac{u(0)}{i(0)}\right|_{sc} = Z\cdot l\cdot\left(1-\frac{1}{3}(\gamma l)^2+\frac{2}{15}(\gamma l)^4 ...\right) \text{ and} \tag{11}$$

$$Y_{in} = \left.\frac{i(0)}{u(0)}\right|_{oc} = Y \cdot l \cdot \left(1 - \frac{1}{3}(\gamma l)^2 + \frac{2}{15}(\gamma l)^4 \ldots\right) \tag{12}$$

In these expressions there is a quadratic term present, but for short transmission lines the two approaches will give similar results.

4. PI-circuits implications

So far only a distributed parameter model has been investigated. However, concentrated parameter models are often used. Besides, the distributed parameter models in ATP are replaced by PI-equivalents during steady state calculation. This sub-section briefly outlines the implications of this.

Fig. 7.3 shows a PI-equivalent under short- and open circuit testing.

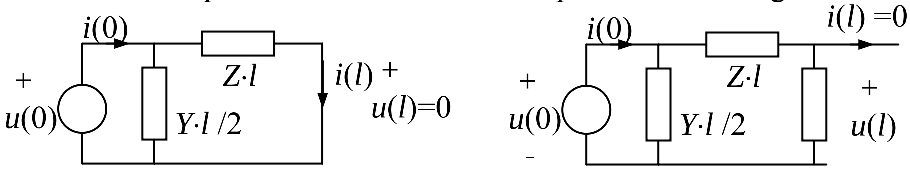


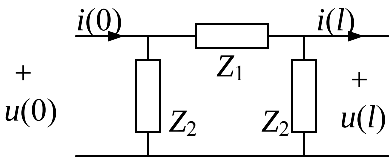
Fig. 7.3 – Testing a PI-circuit. Left: short circuit; serial impedance. Right: open circuit shunt admittance.

The procedure for calculation of the series impedance and shunt admittance in (5) and (10) will in this case result in

$$Z_s^{PI} = \left.\frac{3 \cdot u(0)}{i(0) + 2 \cdot i(l)}\right|_{sc} = \frac{Z \cdot l}{1 + (\gamma l)^2 / 6} \approx Z \cdot l \cdot \left(1 - \frac{(\gamma l)^2}{6}\right) \text{ and}$$

$$Y_s^{PI} = \left.\frac{3 \cdot i(0)}{u(0) + 2 \cdot u(l)}\right|_{oc} = Y \cdot l \cdot \frac{1 + (\gamma l)^2 / 4}{1 + (\gamma l)^2 / 6} \approx Y \cdot l \cdot \left(1 + \frac{(\gamma l)^2}{12}\right) \tag{13}$$

Due to the present quadratic term, the result in (13) will be less accurate than for distributed parameters models. Care must be taken to prevent wrong results for long transmission lines. For example by splitting the line up in smaller segments. In constant parameter distributed parameter line models the series resistance (R) is concentrated at each end ($R/4$) and at the middle of the line ($R/2$). This will result in some different formulations than in (13), with accuracy dependent on R. A solution to this problem is to request 'EXACT PHASOR EQUIVALENT' [2, 3] which prevents ATP from using lumped resistance. In such case the "exact pi" equivalent is used (as is also the case for frequency dependent transmission line models in ATP). The exact PI-equivalent is on the form shown in Fig. 7.4.



$$Z_1 = \sqrt{\frac{Z}{Y}} \cdot \sinh(\gamma l) \text{ and}$$

$$Z_2 = \sqrt{\frac{Z}{Y}} \cdot \frac{\sinh(\gamma l)}{\cosh(\gamma l) - 1}$$

Fig. 7.4 – Exact PI-equivalent

With reference to (13) the calculated series impedance and shunt admittance become

$$Z_s^{Exact-PI} = \left.\frac{3\cdot u(0)}{i(0)+2\cdot i(l)}\right|_{sc} = \sqrt{\frac{Z}{Y}}\cdot\frac{3\cdot\sinh(\gamma l)}{2+\cosh(\gamma l)} \approx Z\cdot l\cdot\left(1-\frac{(\gamma l)^4}{180}\right) \text{ and}$$
$$Y_s^{Exact-PI} = \left.\frac{3\cdot i(0)}{u(0)+2\cdot u(l)}\right|_{oc} = \sqrt{\frac{Y}{Z}}\cdot\frac{3\cdot\sinh(\gamma l)}{2+\cosh(\gamma l)} \approx Y\cdot l\cdot\left(1-\frac{(\gamma l)^4}{180}\right) \quad (14)$$

We see that the exact-pi equivalent gives the same result as the distributed parameter model.

7.2.2 3-phase systems

1. Positive and zero-sequence

A 3-phase circuit is tested with positive and zero sequence sources applied. In the positive sequence, phase number *i* is energized with a sinusoidal source with a phase angle -120º·(*i*-1). In the zero sequence system all phases are energized with a sinusoidal source with zero phase angle. In cases with several 3-phase circuits in parallel the other circuits are not energized and open. The series impedance and shunt admittance are calculated for each individual phase as deduced above. For example in phase *a*: $Z_{sa} = 3\cdot u_a(0)/(i_a(0)+2\cdot i_a(l))$.

2. Self-impedance/admittance

The self-impedance and admittance of the 3-phase circuit *j* is defined as the average of the values for each individual phase: $Z_{jj} = 1/3\cdot(Z_{sa}+Z_{sb}+Z_{sc})$ and $Y_{jj} = 1/3\cdot(Y_{sa}+Y_{sb}+Y_{sc})$ in either the zero- and positive-sequence system.

3. Mutual couplings

Mutual couplings are the equivalent impedance and admittance between circuits. The deduction of these quantities is base on an equivalent two-phase representation shown in Fig. 7.5. Each 3-phase circuit is equated by a single conductor with its self-impedance/admittance and with the average voltage and current distribution.

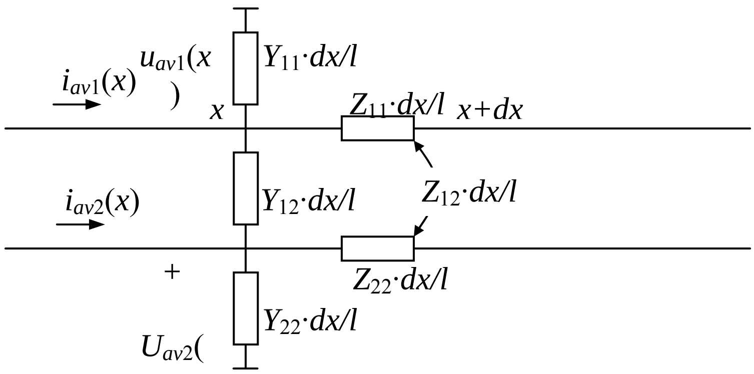


Fig. 7.5 – Two-phase representation

Similar to the single-phase case, matrix expressions are now developed and approximated by series expansions. The end-result is equal to the single-phase case:

$$u(0) = Z_s\cdot\tilde{i} \quad (15)$$
$$i(0) = Y_s\cdot\tilde{u}$$

with

$$Z_s = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12} & Z_{22} \end{bmatrix},\ Y_s = \begin{bmatrix} Y_{11} + Y_{12} & -Y_{12} \\ -Y_{12} & Y_{22} + Y_{12} \end{bmatrix}, \tag{16}$$

$$u(0) = \begin{bmatrix} u_{av1}(0) \\ u_{av2}(0) \end{bmatrix}, i(0) = \begin{bmatrix} i_{av1}(0) \\ i_{av2}(0) \end{bmatrix}, \tag{17}$$

$$\tilde{u} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} u_{av1}(0) + 2u_{av1}(l) \\ u_{av2}(0) + 2u_{av2}(l) \end{bmatrix}, \tilde{i} = \begin{bmatrix} \tilde{i}_1 \\ \tilde{i}_2 \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} i_{av1}(0) + 2i_{av1}(l) \\ i_{av2}(0) + 2i_{av2}(l) \end{bmatrix} \tag{18}$$

The unknown mutual impedance and admittance becomes

$$Z_{12} = \left(\frac{u_{av1}(0)}{\tilde{i}_1} - Z_{11} \right) \cdot \frac{\tilde{i}_1}{\tilde{i}_2} \tag{19}$$

$$Y_{12} = \left(\frac{i_{av1}(0)}{\tilde{u}_1} - Y_{11} \right) \cdot \frac{\tilde{u}_1}{\tilde{u}_1 - \tilde{u}_2} \tag{20}$$

In the positive sequence system the average currents and voltages tend to be very small, and for a perfectly symmetric and transposed systems exactly zero. In such situations the positive sequence coupling has no meaning. The typical test condition is to apply 1 pu current at both circuits with the other ends grounded to obtain the mutual impedance. For mutual admittance the test condition is to apply 1 pu at one and 0 (or -1) pu at the other circuit and leaving the other ends open.

7.3 Hybrid Transformer, XFMR

The modeling of the transformer is based on the magnetic circuit transformed to its electric dual [7, 8]. The leakage and main fluxes are then separated into a core model for the main flux and an inverse inductance matrix for the leakage flux. The copper losses and coil capacitances are added at the terminals of the transformer. The resulting electrical circuit is shown in Fig. 7.6. Only standard EMTP elements are used.

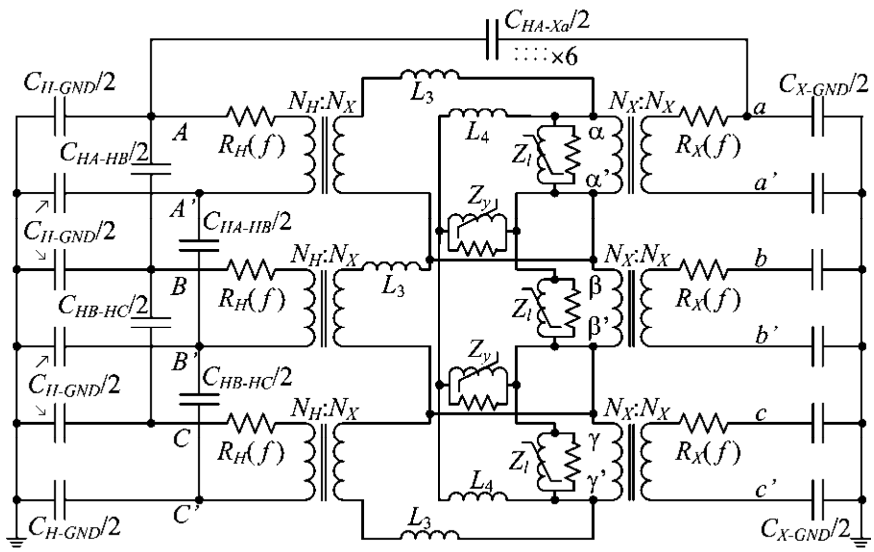


Fig. 7.6 – Electric model of the Hybrid Transformer [9],
2-windings (H and X), 3-phases, 3-legged core.

Transformer parameters can be based on three different data sources; typical values, test report, and design information. The three sources can be selected independently for resistance, inductance, capacitance, and core. Test report input is based on standard open- and short-circuits tests, with capacitance measurements as an additional option. This is the normal choice of data source for existing transformers. Design data requires the geometry and material parameters of the windings and the core. Such data are rarely available so this option is more for research purposes. The Typical value option uses available text book tabulated values of leakage impedance, copper and core losses, and magnetizing current to estimate model parameters. This is suitable when the transformer is not purchased yet, or data is unavailable in an initial study. However, such model must be used with caution.

7.3.1 Leakage inductance

The leakage inductance is modeled with an inverse inductance matrix (A-matrix). The matrix has dimension $(nw+1)\cdot np$ where nw is the number of physical windings, the core is connected to the $nw+1$ winding, and np is the number of phases [7-9]. The coupling (auto, Y, D), turns ratio, and phase shift are produced directly in the A-matrix. All possible phase shifts are supported. The A-matrix has the following structure for a three-winding, three-phase transformer:

$$A = \begin{array}{c} \begin{matrix} \mathrm{A} & \mathrm{B} & \mathrm{C} \end{matrix} \\ \begin{bmatrix} A_w & 0 & 0 \\ 0 & A_w & 0 \\ 0 & 0 & A_w \end{bmatrix} \end{array} \text{ where } A_w = \begin{array}{c} \begin{matrix} \mathrm{P} & \mathrm{S} & \mathrm{T} & \mathrm{C} \end{matrix} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \end{array} \qquad (1)$$

where ABC are the three phases and PSTC stands for primary, secondary, tertiary, and the core (nw+1) winding.

The A-matrix is assumed to have no mutual coupling between the phases. The entire zero-sequence effect is modeled in the attached core. The A_w-matrix is established according to the EMTP Theory Book [5] Section 6.4, and Section 5.2.4 p. 31 in [7].

7.3.1.1 Typical values

The leakage reactance is established from [11] using the lowest value in the typical range. In the case of a three-winding transformer the leakage reactance (in pu) between the inner and outer winding is approximated as the sum of the other two. In this case it is assumed that the medium voltage winding is the middle one.

7.3.1.2 Test report

The leakage reactance is calculated from the standard test report short circuit data (positive sequence).

$$X[pu] = \sqrt{Z[\%]^2 - (P[kW]/(10 * S[MVA]))^2}\ /100 \qquad (2)$$

In the case of an autotransformer the reactances are scaled according to the Theory Book [5] Section 6.7.

7.3.1.3 Design data

The leakage reactances are calculated according to classical MMF distribution theory as shown in [7, 8]. Both cylindrical and pancake windings are supported.

7.3.1.4 Handling of the core winding

The artificial core winding is related to the leakage channel between the inner physical winding and the core. A parameter $K=a_1/a_2$ is defined in [7, 10] where a_1 is the width of the inner leakage channel and a_2 is the width of the leakage channel between the inner and the outer/middle winding. A fixed value K=0.5 is used in ATPDraw. If the pu leakage reactances X_{ML}, X_{MH}, and X_{HL} (L=inner, M=middle, H=outer) for a three winding transformer are given then the leakage reactances to the core winding are assumed to be [7, 10]

$$X_{LC} \approx K \cdot X_{ML} ,\; X_{MC} \approx X_{LC} + X_{ML} = (K+1) \cdot X_{ML} \text{, and}$$

$$X_{HC} \approx X_{MC} + X_{HM} = (K+1) \cdot X_{ML} + X_{HM} \tag{3}$$

7.3.2 Winding resistance

The winding resistances are added externally at the terminal of the transformer (A-matrix). Optionally, the resistances can be frequency dependent.

7.3.2.1 Typical values

The typical winding resistances (at power frequency) are in principle based on [12]. A function (4) is established that takes in the parameter u [kV] and s [MVA] and returns the resistance in %. Data for a 290 MVA/ 420 kV transformer (Table I) were used to extend the data given in [12]:

$$R_w = 0.7537 \cdot \left(\frac{u}{15}\right)^{0.0859} \cdot s^{-0.2759} \quad [\%] \tag{4}$$

7.3.2.2 Test report

The test report data are given at power frequency. The per unit short circuit resistances are calculated from short circuit power losses in the test report (positive sequence). The winding resistance (in pu) is assumed to be equally shared between the windings in the case of a two-winding transformer. In the case of a 3-winding transformer the traditional star-equivalent approach is used.

In the case of an auto-transformer the short circuit resistances are recalculated according to the power balance used in [10]. The approach used for reactances (from the Theory Book [5]) did not work out for the resistances.

7.3.2.3 Design data

The user can specify the winding conductivity σ, the equivalent cross section A of each turn, the average length l of each turn, number of turns of the inner winding N. The DC resistance is normalized to the power frequency. If the resistance is assumed to be frequency dependent the conductor area must be specified in height and width (which determines the stray losses).

7.3.2.4 Frequency dependency

The frequency dependent resistance is calculated between 0.1 to 10 kHz. The typical values and test report resistances are assumed to follow $R_S(\omega) = R_0 \cdot \sqrt{\omega / \omega_0}$ where R_0 is the resistance at the

angular power frequency ω_0. This expression results in considerably lower values than suggested in Fig. 26 in [7]. This needs to be further investigated. The design data resistances are assumed to follow eq. (37) in [7].
The calculated $R(\omega)$ and ω value pairs are fitted to the function (two-cell Foster equivalent)

$$R(\omega) = R_0 + \frac{R_1 \cdot \omega^2 \cdot L_1^2}{R_1^2 + \omega^2 \cdot L_1^2} + \frac{R_2 \cdot \omega^2 \cdot L_2^2}{R_2^2 + \omega^2 \cdot L_2^2} \tag{5}$$

$$L(\omega) = \frac{L_1 \cdot R_1^2}{R_1^2 + \omega^2 \cdot L_1^2} + \frac{L_2 \cdot R_2^2}{R_2^2 + \omega^2 \cdot L_2^2}$$

with the resistances R_1 and R_2, and inductances L_1 and L_2 as unknowns. The fitting routine is based on a Genetic Algorithm implemented in ATPDraw. The object function is defined as OF= min ($|R(\omega)\text{-}R_S(\omega)|+|\omega\, L(\omega)|$) contrained to positive unknowns. A negative inductance L_0=-L_1-L_2 is added in series with the winding resistance to compensate for the inductance of the Foster cells. A constraint is put on the total inductance $|L_0| < L_w$ where L_w is the inverse of the diagonal A_w-matrix element, [7] section 5.4.2. The constraint is handled simply by setting L_1=L_2=$0.5\cdot L_w$ when the constraint is violated and then continue to obtain new optimized values for R_1 and R_2.

7.3.3 Capacitance

The C-matrix is split in two halves and connected to each end of the physical windings. The capacitance matrix C is based on the following two matrices:

$$C_w = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \text{ and } C_p = \begin{bmatrix} C_{AA} & C_{AB} & C_{AC} \\ C_{BA} & C_{BB} & C_{BC} \\ C_{CA} & C_{CB} & C_{CC} \end{bmatrix} \tag{6}$$

The C_w matrix contains the capacitances between windings 1-3 equal in all phases. The capacitance matrix C_w is built up like a nodal admittance matrix. The C_p matrix contains capacitances that are specific to phase A, B, or C. These are typically connected to the outer windings. The total C-matrix is then built on these two symmetrical matrices dependent on the type of winding (pancake/cylindrical). The concept “outer winding” will be different for pancake and cylindrical windings.

7.3.3.1 Typical values

A capacitive coupling factor K_c can be specified by the user with a default value of 0.3. The concept of transient recovery voltage (TRV) is used to calculate the effective capacitance when the inductance is known [13]. IEEE standard C37, Fig. B2 [14] is used to obtain the typical frequency of the TRV for a known voltage level and fault current.

$$C_{eff}(U,S,X_{pu},f) = \frac{f}{2\pi} \cdot \frac{3 \cdot I}{U \cdot \left(f_{TRV}(U,I)\right)^2} \ [\mu\text{F}] \tag{7}$$

with U in [kV], S in [MVA] and $I = \dfrac{S}{3 \cdot U \cdot X_{pu}}$ [kA]

In the case of typical values, the C_p matrix (between phases) is always set to zero for lack of any better choice. For a two-winding transformer the C_w matrix is calculated as

$$\begin{aligned} C_w[1,2] &= C_{PS} = K_c \cdot C_{eff}(U_S, S, X_{PS,pu}, f) \\ C_w[1,1] &= C_{PP} = C_{eff}(U_P, S, X_{PS,pu}, f) - C_w[1,2] \\ C_w[2,2] &= C_{SS} = (1/K_c - 1) \cdot C_w[1,2] \end{aligned} \tag{8}$$

For a three winding transformer the typical capacitance is more complicated with several coupling factors involved. Here a simple approach is used:

$$\begin{aligned} C_w[1,3] &= C_{PT} = 0 \\ C_w[2,3] &= C_{ST} = C_{eff}(U_S, S, X_{ST,pu}, f) - C_w[2,2] \\ C_w[3,3] &= C_{TT} = C_{eff}(U_T, S, X_{ST,pu}, f) - C_w[2,3] \end{aligned} \tag{9}$$

This approach could be further discussed and improved.

7.3.3.2 Test report

In the test report the capacitances between each winding and ground and between all windings is assumed to be directly specified while the C_p matrix is set to zero. All values must be specified per phase.

7.3.3.3 Design data

The calculation of design data capacitances are based on [7] chapt. 5.3, p. 33-42. The user has to specify the winding geometry as well as the various equivalent permittivities of insulation system. Standard formulas for calculating the capacitance between cylinders and for cylinders over planes are used with end effect and tank effect adjustments.

7.3.4 Core

The core model is connected to the "core winding" terminals of the *A*-matrix. Triplex (single phase cores), stacked cores with three and five legs, and shell form cores are supported. Basically the inductive and resistive core parts are treated independently, but this is a point that requires more research particularily for 3- and 5-legged cores where harmonics in the flux creates additional losses. The core losses are represented by a linear resistor and the nonlinear inductances are modeled by the Frolich equation (10). Each part of the core is modeled with its own core loss resistance and nonlinear inductance using information about their relative cross section and length to scale the values. Fig. 7.7 shows the core model for a 5-legged transformer.

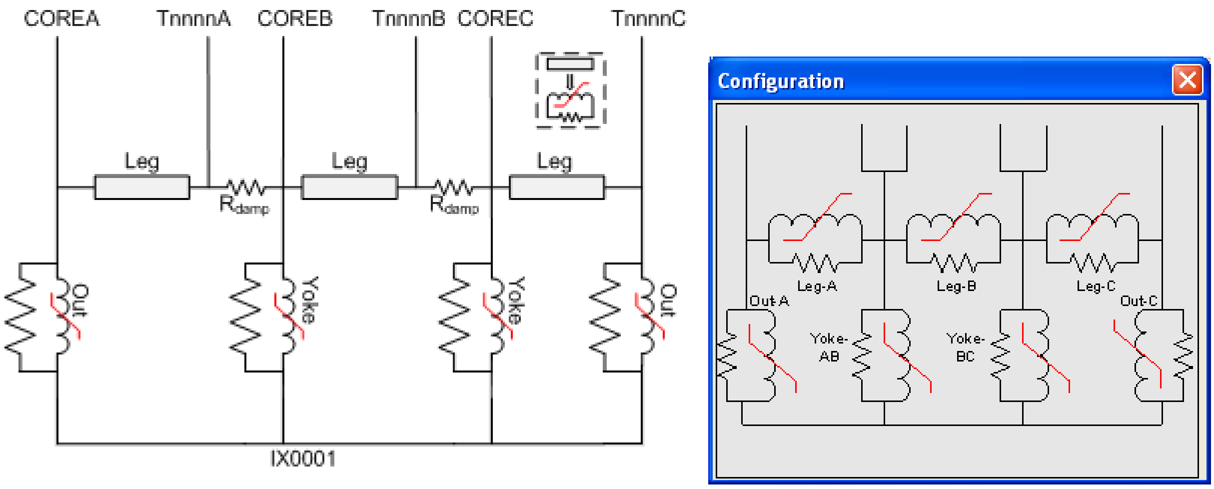


Fig. 7.7 – 5-legged stacked core model. The αβγ terminals are the *nw*+1 winding.
Left: Practical ATPDraw implementation. Right: Topologically correct model.

It is assumed that the magnetic material is characterized by five parameters *a*, *b*, c , *d* and *e*. A list of typical steel materials is developed based on fitting the manufacturer's data from state of the art catalogues. Older steel materials will have a different characteristic and the losses are typically higher. The material list is only used for design data and typical values.

The *B*/*H* relationship is assumed to follow the Frolich equation where the optional parameter *c* (introduced in [15]) improves the fitting to test report data around rated voltage

$$B = \frac{H}{a + b\cdot|H| + c\cdot\sqrt{|H|}} + \mu_0 \cdot H \tag{10}$$

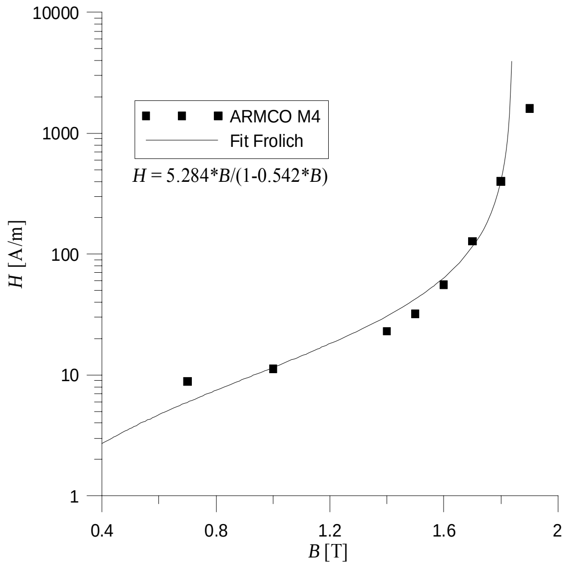
The specific loss is assumed to follow

$$P\,[W/kg] = \left(\frac{f}{50}\right)^{1.5} \cdot \left(d\cdot B^2 + e\cdot B^{10}\right) \tag{11}$$

where *f* is the power frequency.

The specific loss is traditionally (for instance Westinghouse T&D reference book, 1964) assumed to be $P = K_e \cdot (f \cdot t \cdot B_{\max})^2 + K_e \cdot f \cdot B_{\max}^x$ with x said to be 3 for modern materials in the year of 1964. In the above expression *t* is the thickness of the laminates. The traditional expression was tested on modern material data with little success.

Fig. 7.8 shows the fit of the specific losses and DC-magnetization curve of ARMCO M4 steel. The Frolich fitting is not very good, and in Fig. 7.8b fitting around the knee point (nominal flux) was preferred at the sacrifice of high field fitting (*B*=1.9 T). Similar fitting is performed for the other core materials.



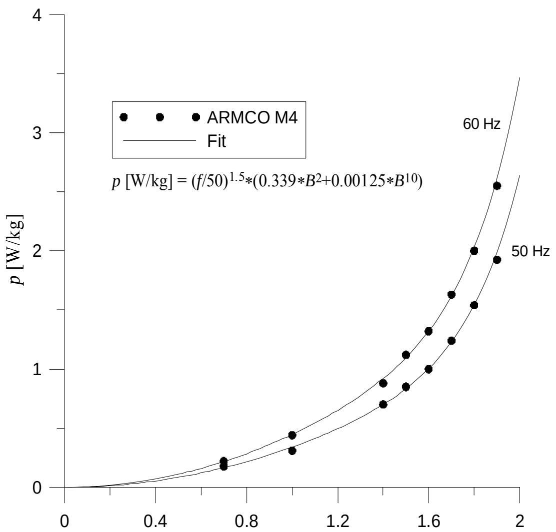


Fig. 7.8a) – Core loss curves (*c*=0)

Fig. 7.8b) DC-magnetization curve

7.3.4.1 Inductance modeling:

The basic Frolich equation in (10) is reformulated as a current flux-linkage characteristic by introducing the flux linkage $\lambda = B \cdot A \cdot N$ and the current $i = H \cdot l / N$ where *N* is the number of turns of the inner winding, *A* is the cross section, and *l* is the length of the involved core section. This gives

$$\lambda = \frac{i \cdot A \cdot N^2 / l}{a + b \cdot |i| \cdot N / l + c \cdot \sqrt{N / l \cdot |i|}} + \mu_0 \cdot i \cdot A \cdot N^2 / l \Rightarrow \frac{\lambda}{A_r} = \frac{i / l_r}{a' + b' \cdot |i| / l_r + c' \cdot \sqrt{|i| / l_r}} + L_a \cdot i / l_r \qquad (12)$$

where the constants $a' = a \cdot l_L / (N^2 \cdot A_L)$, $b' = b / (N \cdot A_L)$ and $c' = c \cdot \sqrt{l_L / (A_L^2 \cdot N^3)}$, based on the absolute length (l_L) and cross section area (A_L) of the core leg, are determined in an optimization process; $\min OF(a', b', c') = \sum_{i=1}^{n} \left(I_{meas,rms}(U_{i,rms}) - I_{calc,rms}(U_{i,rms}, a', b', c') \right)^2$ for n excitation levels.

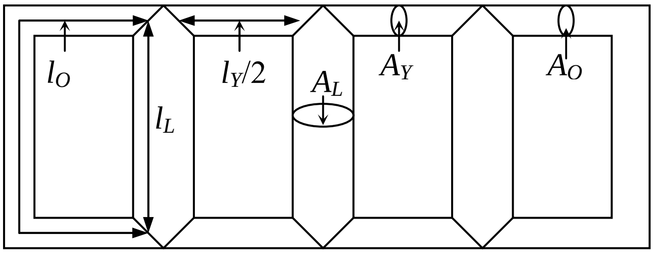


Fig. 7.9 Core dimensions, 5-legged stacked core. The user must provide A_Y/A_L, A_O/A_L, l_Y/l_L, l_O/l_L

The final characteristics are determined by using the relative (to the main leg) dimensions for the corresponding section, A_r and l_r. The nonlinear inductances are implemented as optional type 98, 93, or 96 inductances in ATP.

7.3.4.2 Core loss modeling

The core loss is split in parts associated with individual core sections. It is assumed that the core loss is proportional to the core volume and to the square of the rms voltage across each section of the electric dual. The voltage, U_o, in the neutral point in Fig. 7.7 (node IX0001) is the time derivative of the neutral flux found during the Frolich optimization described above. Is is assumed that the inductive current components determine the voltage distribution. For a 5-legged core

$$P_{loss} = 3P_{leg} + 2P_{yoke} + 2P_{out} = p \cdot \left(3 + 2 \cdot V_{ry} \cdot (U_y / U)^2 + 2 \cdot V_{ro} \cdot (U_o / U)^2 \right) \qquad (13)$$

where V_{ry} and V_{ro} are the relative volumes of the yoke and outer legs respectively.
and where U_y and U_o are the rms value of the voltage across the sections.

For a 3-legged core the outer leg volume is zero and for triplex and shell form core the loss distribution is straight forward and determined only by the main leg voltage.

In the type 96 modelling, half of the loss is included as hysteresiss loss scaled by a Steinmetz coefficient of 2. The hysteresis has a uniform width.

7.3.4.3 Typical values

The estimation of the magnetizing current (I_m) is based on [12]. Some fitting of the data is performed which results in

$$I_m = 0.73 \cdot \left(\frac{BIL}{350} \right)^{0.2933} \cdot \left(\frac{s}{20} \right)^{-0.2154} \quad [\%] \qquad (14)$$

when the basic insulation level (BIL) is known and

$$I_m = 0.855 \cdot \left(\frac{u}{150} \right)^{0.2283} \cdot \left(\frac{s}{20} \right)^{-0.2134} \quad [\%] \qquad (15)$$

when *BIL* must be estimated. *BIL* is in [kV], *u* is the rated voltage in [kV], and *s* is the rated power in [MVA].

For a typical core model the user has to specify the maximum *B*-field (normally 1.5-1.7 Tesla) and the maximum core loss density. First a core material has to be guessed and this gives the *a* and *b* values in the Frolich equation (and possibly also the *c* and *d* values that would replace *p*).
The following relationships are then assumed:

$$\lambda_{\max} = \frac{\sqrt{2}\cdot U_{rms}}{\omega} = B_{\max}\cdot A\cdot N \Rightarrow A\cdot N = \frac{\sqrt{2}\cdot U_{rms}}{\omega\cdot B_{\max}} \tag{16}$$

$$\begin{aligned} H_{\max} &= \frac{a\cdot B_{\max}}{1-b\cdot B_{\max}} \approx \sqrt{2}\cdot i_{rms}\cdot\frac{N}{l} \\ \Rightarrow \frac{N}{l} &= \frac{a\cdot B_{\max}}{(1-b\cdot B_{\max})\cdot\sqrt{2}\cdot i_{rms}} \end{aligned} \tag{17}$$

which simplistically assumes a sinusoidal magnetizing current.

This gives the parameter of the fluxlinkage-current characteristic:

$$\begin{aligned} a' &= a\cdot\frac{l}{A\cdot N^2} \approx \omega\cdot(1-b\cdot B_{\max})\cdot\frac{i_{rms}}{u_{rms}}\ , \\ b' &= b\cdot\frac{1}{A\cdot N} \approx b\cdot\frac{\omega\cdot B_{\max}}{\sqrt{2}\cdot u_{rms}} \qquad \text{and} \\ c' &= 0 \end{aligned} \tag{18}$$

We see that the expressions for *a'* and *b'* are independent of the magnetic material property *a*. The typical value of *b* seems to be fairly constant for standard core materials and a value of 0.5 is assumed in ATPDraw.

The core loss is estimated as

$$P_{loss} = p\cdot\rho\cdot A\cdot l = p\cdot\rho\cdot\frac{(1-b\cdot B_{\max})\cdot 2\cdot u_{rms}\cdot i_{rms}}{\omega\cdot a\cdot B_{\max}^2} \tag{19}$$

where *p* [W/kg] and ρ [kg/m^3] are given and the volume $A\cdot l$ is estimated from (16) and (17).

7.3.4.4 Test report

The user specifies the excitation voltage in [%], the current in [%] and the core loss in [kW]. The core loss is used directly as explained above to obtain the core resistances. For now the core resistances are assumed to be linear and the core loss value at 100 % excitation is used.

The inductive magnetizing current for each point is calculated as

$$I_{rms} = \sqrt{I_0[\%]^2 - \left(\frac{P[kW]}{10\cdot S[MVA]}\right)^2}\ [\%] \tag{20}$$

This results in a sequence of excitation points (U_{rms} and I_{rms}). The magnetic circuit in Fig. 7.7 assuming sinusoidal fluxes is solved and the rms values of the line currents are calculated and compared to measured ones. Optimized values of *a'*, *b'* and *c'* (optional) in (12) are found by a Gradient Method implemented in ATPDraw. If a single point is specified the core model is linear.

7.3.4.5 Design data

For design data the user directly specifies the core material with its B-H relationship (*a* and *b* values in (10)). The absolute core dimensions and the number of inner-winding turns *N* are known, so the inductances can be found directly from (12). Based on manufacturer data the core losses can be established from (11) with $B = \frac{\sqrt{2} \cdot U_{rms}}{\omega \cdot A \cdot N}$ and known values of the core weight (volume and density) the core loss can be estimated.

7.4 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] B. A. Mork, F. Gonzalez, and D. Ishchenko: “Parameter estimation and advancements in transformer models for EMTP simulations. Task MTU-7: Model performance and sensitivity analysis”, Bonneville Power Administration, Portland, OR, 2004.

[8] B.A. Mork, F. Gonzalez, D. Ishchenko, D. L. Stuehm, J. Mitra.” Hybrid Transformer Model for Transient Simulation-Part I: Development and Parameters”, *IEEE Trans. Power Delivery*, Vol. 22, pp. 248-255, 2007.

[9] B.A. Mork, F. Gonzalez, D. Ishchenko, D. L. Stuehm, J. Mitra, “Hybrid Transformer Model for Transient Simulation-Part II: Laboratory Measurements and Benchmarking”, *IEEE Trans. Power Delivery*, Vol. 22, pp. 256-262, Jan. 2007

[10] B. A. Mork, F. Gonzalez, D. Ichshenko, “Leakage inductance model for Autotransformer transient simulation”, in *Proc. Int. Conf. on Power System Transients*, paper 248, 2005.

[11] J. J. Graininger and W. D. Stevenson: *Power System Analysis*, McGraw-Hill 1994.

[12] A. Greenwood: *Electrical Transients in Power Systems*, Wiley, 1991.

[13] IEEE Working Group 15.08.09, Editors: A. M. Gole, J. Martinez-Velasco, A. J. F Keri, *Modeling and analysis of power system transients using digital programs*, IEEE 99TP133-0, pp. 4.12-4.13, 1998.

[14] *IEEE Guide for Transient Recovery Voltage for AC High-Voltage Circuit Breakers Rated on a Symmetrical Current Basis*, ANSI/IEEE Standard C37.011-1994.

[15] N. Chiesa, “Power Transformer Modelling: Advanced Core Model”, M. SC. Thesis, Politecnico di Milano, Italy, 2005.

[16] C. Zhu, R.H. Byrd and J. Nocedal : *L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization*, ACM Transactions on Mathematical Software, Vol 23, Num. 4, 1997 Page(s): 550 – 560.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P, Flannery: *Numerical recipes*, 2nd Ed. 1992, Cambridge University Press.

7.5 Index

N

O

P

R

S

T

U