Simulation of Electromagnetic transients in Power systems: a comparison between ATP/ATPDRAW and SIMULINK/PSB environments

M. Ceraolo, S. Barsali
University of Pisa
Via Diotisalvi, 2, 56126 Pisa - Italy   m.ceraolo@ing.unipi.it

Abstract - **This paper presents a first comparison between two modern tools for Electro-magnetic transient simulations:**

- **the set ATP/AtpDraw that, based on the historical EMTP from BPA, has recently improved its usability mainly as a consequence of the introduction of the AtpDraw pre-processor**
- **the Power System Blockset addition to the well-known MATLAB/SIMULINK computing environment.**

The comparison between the two products is performed based mainly on the following items:

- **features available**
- **user interface**
- **integration with other products**

In addition, some sample circuits are analysed with both tools:

- **a very simple example just to give an idea of the general graphical arrangement of the circuit and simulation parameters**
- **a simple circuit showing a typical numerical instability arising when using one of the two tools, correctly simulated using the other**
- **a larger example that can give an idea of the usability of the two programs in case of circuits having some complexity.**

I. INTRODUCTION

Nowadays one of the main activities of the power system designer is to simulate the system behaviour at different levels.

In several cases, especially in which no sinusoidal shape can be assumed for electrical quantities (mainly voltages, currents) simulations must be performed by means of the so-called electro-magnetic transients programs.

Several of them exist, with different characteristics. One of the most widely used electromagnetic transients program is the Alternative Transient Program (ATP), based of the "historic" program EMTP (Electro-Magnetic transients Program) from Bonneville Power Administration.

This product has, in comparison with the majority of competitors, the advantage of being a fully featured product, distributed at virtually no cost. Recently ATP has become much easer to use by the addition of a powerful graphical pre-processor, called AtpDraw.

On the other hand, recently a new electro-magnetic transients simulation tool has been delivered: the Power System Blockset based on the worldwide known Matlab/Simulink commercial simulation package [2, 7]. One of the main advantage of this option is its tight integration with Matlab/Simulink environment, so users that already know Simulink are virtually ready to use PSB.

It is felt that these two simulation tools are very appealing for the electrical engineer that needs to perform electromagnetic transient simulations reliably and fast.

It is then deemed useful to have adequate information available to make a choice between them.

The main aim of this paper is to supply, at least partially, this information.

II. A HANDS-ON PRESENTATION OF THE TWO TOOLS

A. The ATP set

The traditional user interface of the ATP program implies the preparation of one or more text files containing all the information needed to perform the simulation: circuit data, simulation parameters, output options.

This file because of its historical background, has a complex appearance that discourages the beginner (fig. 1). It is also error-prone, since the data are position-based, so a simple one-character horizontal shift of a data item can change dramatically the meaning of the program input.

Nevertheless, it is not too bad for the experienced users, that, once they have prepared the file/s for the first time, need only to change some number or piece of text somewhere, and can repeat several simulations in a short time.

```
BEGIN NEW DATA CASE
C   A  sample case
  10.e-6   0.080
   10000       1
  DCLOADDCMENO                      11.56115.60
  DCMENO                            1.E06
  DCPIU                             1.E06
  TRANSFORMER                                   T1
            9999
 1TAP    STARP                      3.840 554.538105.
 2TAS                               .0001 00001 1496.
```

Fig. 1: The first rows of a sample input deck of ATP (some data present at the right part of the figure is not displayed).

However, times have come that users more and more refuse to tackle with nasty user interfaces and they want to have higher productivity graphical user interfaces.

The ATP program has therefore been integrated with graphical pre- and post- processors.

The graphical pre-processor is called AtpDraw, and allows to create automatically the ATP input deck starting from a graphical representation of the circuit to be simulated.

Fig. 2 shows the representation of a simple R-L circuit fed by a sinusoidal source.

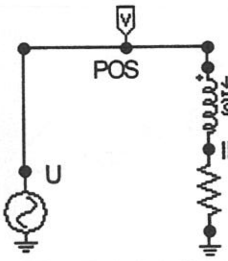


Fig. 2: AtpDraw representation of a simple R-L circuit fed by a sinusoidal source.

The circuit has only four objects: a voltage source, a linear inductor, a linear resistor, a voltage probe. Double clicking on each of these objects causes a corresponding dialog box to be displayed, that allows to set all the object data: numerical values, node names, choice of output some of the object quantity (e.g., voltage or current), etc.

After the simulation is performed the results can be viewed by one of the several plotting programs available. This paper takes as reference the PlotXY plotting program, that is designed from the beginning for the Win32 environments, and therefore integrates in the Microsoft Windows' operating systems smoothly. Since it has been created as a demonstrator of the various Man-Machine interface issues of a plotting program [8], it has also a strong usability, as several users acknowledge.

The set composed by AtpDraw-ATP-PlotXY will be called in this paper shortly the *ATP set*.

A simple transient that can be simulated in a simple R-L circuit is its energisation starting from the initial condition of i(L)=0; voltage and current plots versus time of this simulation performed with the ATP set are shown in figure 3.

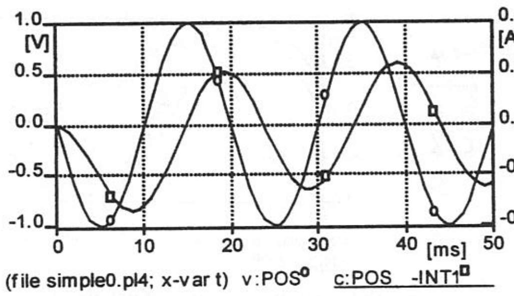


Fig. 3: POS node voltage (v:POS, Left scale) and inductance current (c:POS -INT1) for the circuit of fig. 2.

Very often the user of an electro-magnetic transients program wishes to start its simulation starting from an automatically-calculated steady-state phasor solution. This can be done easily with the ATP set, but the inductance component has to be changed from the "IND_I0" component to the "INDUCTOR" one (fig. 4).

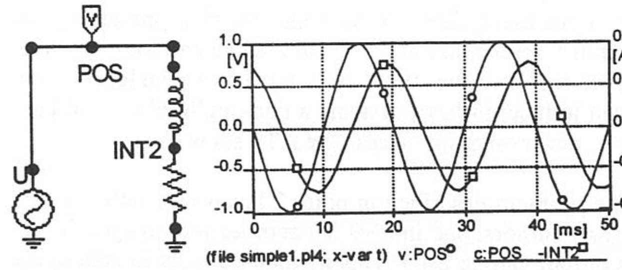


Fig. 4: Simulation of a simple R-L circuit starting from phasor solution.

B. *The PSB Set*

The so called *Power System Toolbox* (PSB) is an addendum of the MATLAB/SIMULINK environment [7]. In the remainder of this paper the whole package composed by Matlab, Simulink, and PSB will be referenced to as the *PSB set*.

The same circuit analysed in the previous section using the ATP set can be analysed using the PSB set. A pictorial description of the circuit according to the PSB set symbology, is shown in fig. 5. The resulting plots are showed in fig. 6.

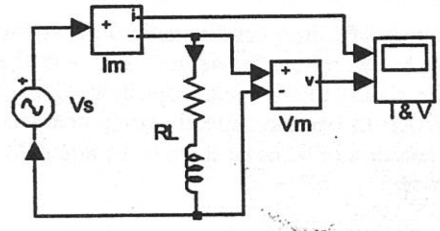


Fig. 5: PSB representation of a simple R-L circuit fed by a sinusoidal source.

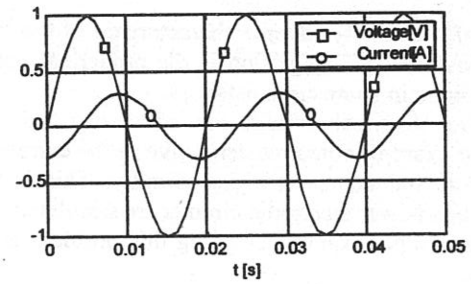


Fig. 6: Voltage current obtained simulating the simple circuit of 2.

Two main differences with the ATP set are apparent:

1. all the wires must carry an arrow to show their orientation. This is a direct consequence of the fact that the sketch is actually a Simulink drawing, and Simulink mainly relates to technical diagrams for which the orientation is mandatory. Just to have an idea, take as reference a Laplace transfer function: a clear definition of which of the signal is the "input" and which is the "output" is mandatory.

The need of using always oriented connections, however,

does not create excessive overhead to the program user: just the need to add special blocks to manage situations of wires converging into a single node.
2. to store some signal for plotting or further processing the circuit description has to be altered since blocks such as the "Im" and the "Vm" ones have to be included. This can induce, in large systems a non-negligible visual hassle, not present in the case the ATP set is used.

The problem described in point 2 has been tackled by the program authors, and in PSB 2.0 another way to specify signals output can be used. This method consists in specifying output quantities directly in the circuit elements (sources, R-L-C blocks, etc) and then access them by a device called "multimeter" (in figure named "I&V"). Although the simplification obtained in this case is modest, it is easily understood that on more complex cases it can be much larger.

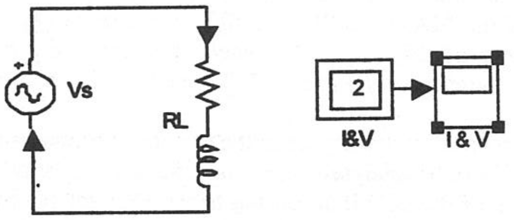


Fig. 7: Use of multimeter for outputting of signals.

The selection of initial conditions in PSB is Straightforward, via a block called "powergui" that lets the user to chose all the state variables (e.g. capacitor voltages and inductor currents) to be automatically computed according to the phasor solution or some or them to be manually overridden by the user.

III. THE PROBLEM OF NUMERICAL OSCILLATIONS

One of the mostly adverse characteristic of the fixed-step trapezoidal rule of integration is the numerical oscillations that can arise in some circumstances.

In case electrical circuits are concerned, this problem arises for example when the derivative of the current flowing through an inductance changes abruptly. This is very frequent when power electronic circuits are simulated.

A very simple example showing this problem is reported in fig. 8.

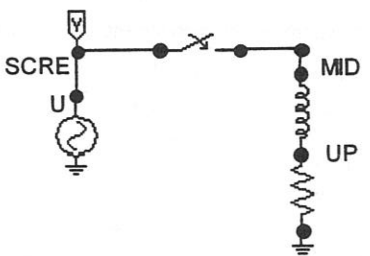


Fig. 8: Simple circuit for showing the problem of numerical oscillations.

The switches receives the order of opening at 10 ms, but it actually performs this operation when it detects a zero-crossing of its current. Therefore the example is also indicative of what could happen if instead of the switch there was a diode (or SCR) in the circuit.

The fig. 9 contains the result of the simulation of this circuit.

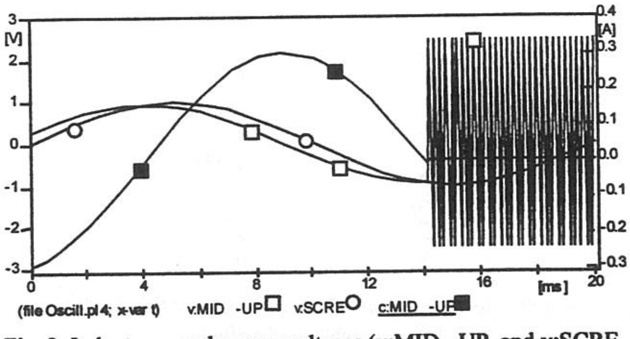


Fig. 9: Inductance and source voltages (v:MID -UP, and v:SCRE, left scale) and inductance current (c:MID -UP, right scale) for a simulation performed on the circuit of fig. 8

A numerical oscillation is started at t= 14 ms, i.e., when the switch actually opens. This kind of numerical oscillation has been thoroughly studied in the past, and for example discussed in [3] and [4] that propose effective countermeasures[1], but no "algorithmic" solution has been introduced in the ATP.

Therefore all ATP users should be aware of this problem, and be able to manage modifications of the circuit to be simulated in such a way so that the results are not significatively affected, except for the fact that the numerical oscillations are suppressed. The typical solution of the problem of numerical oscillations shown in figs. 8 and 9, is to put in parallel with the inductance a large resistors able to "damp" these oscillations. To make the oscillation last just one time step this "damping" resistor should fulfil the requirement: $R_d < \frac{2L}{\Delta t}$ where R_d is the damping resistor and Δt is the integration timestep (the demonstration is rather trivial and is omitted for brevity).

The modified circuit and plots are reported in fig. 10.

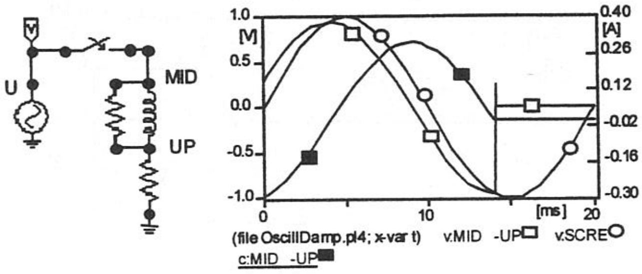


Fig. 10: Circuit and plots showing damping of oscillations of fig. 9.

Nevertheless the relative simplicity with which the numerical oscillations can be suppressed with a good understanding of

[1] These countermeasures have been implemented, for example, in the well known DCG/EPRI and Microtran EMTP programs.

the phenomenon, it requires additional skill from the human operator, and, for complex cases can be time consuming.

To perform a comparison with respect to this problem of numerical oscillations let the circuit of the fig. 11, that is equivalent to the one of fig. 8, be considered.
This can be executed using different discretisation algorithms. The author's have considered the `ode15s` and `ode23tb` that are suggested by the PSB authors[2]. The results using these two algorithms were coincident, and are reported in the right side of fig. 11.

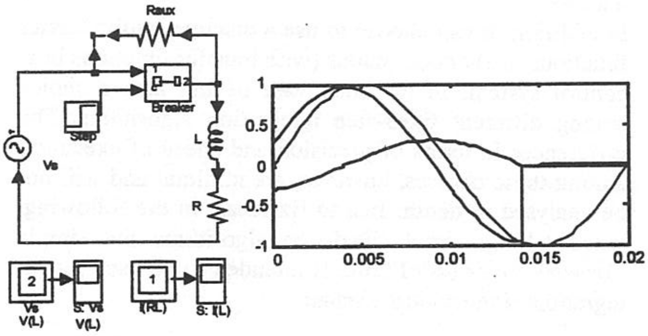


Fig. 11: Circuit and plots corresponding to fig. 8, related to PSB.

It is rather obvious that the better numerical stability shown in this example by the PSB with respect to the ATP set cannot be immediately and totally generalised. However, since the example is so simple and the problem of currents being cut by breakers or diodes or SCRs is so frequent, in the authors' opinion the example shows clearly that the problem of numerical oscillations or instability, if present, is much smaller with PBS than ATP.

From this examples it can be drawn the conclusion that with respect to the problem of numerical oscillation the PSB set is better that the ATP set.

IV. COMPLEX CIRCUITS SIMULATION

When complex circuits are to be simulated a simulation package should have particular characteristics not required for simpler ones.

One of these is the possibility of defining some *subcircuits*: parts of a circuit that can be treated as single components and freely replicated.

As an example take a Three-phase thyristor Graetz bridge: while this system has an inner structure composed by thyristors, snubbers, and, possibly some part of the valve control, it is useful to treat it as a single component to simplify the visual representation of the circuit, and greatly ease the circuit set-up in case multiple instances of the component are required.

[2] The ode15s algorithm is a modified version of the Gear's formulas; the ode23tb, has a first stage that is a trapezoidal rule step and a second stage that is backward differentiation formula of order two.

Let the circuit of fig. 12 be considered as an example. It contains an simple AC system feeding a three-phase bridge that in turn feeds a load.

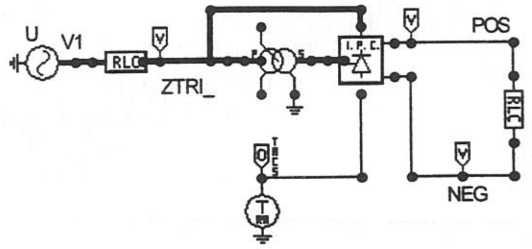
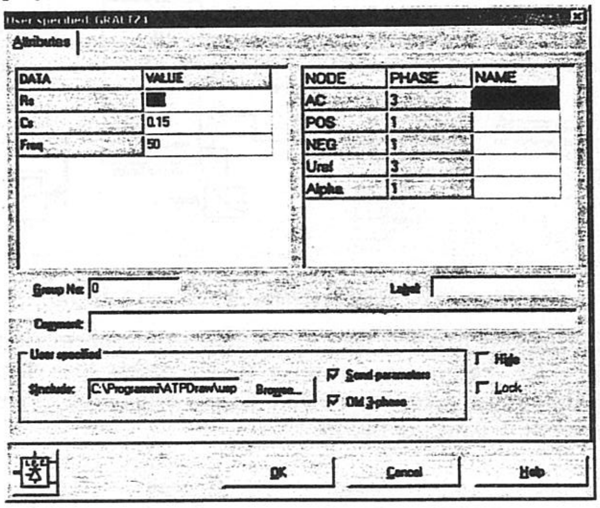


Fig. 12: An ATP example showing the use of User Specified Objects.

The block having as icon a thyristor and the label "I.P.C." is a user-specified component and encapsulates:
a full three-phase thyristor bridge (six thyristors)

- the thyristor snubbers
- an Individual Phase Control technique to identify the voltage zero-crossings that define the times at which the delay angle alpha begins to be computed.

If one double clicks on this block, the following dialog box is displayed:



The user has only to specify the DATA fields, in this case Rs, Cs, Freq (respectively snubber resistance and capacitance and frequency). The NODE NAME fields are automatically filled by the program when the builds the ATP input deck while executing the command ATP|Make File.

From this example it is clear that using user-defined components is very useful and simple in the ATP set.

However, creating them is much more difficult, and especially breaks the main reason for which AtpDraw was created: to be a layer between the user and the complex file structure of the "pure" ATP. Indeed, to create a User Specified Model the programmer should know the ATP syntax to a good degree.

In Fig. 13 the DC voltage and current obtained simulating the circuit of fig. 12 in which the delay angle alpha is linearly raised from 0 to 90 degrees is shown.

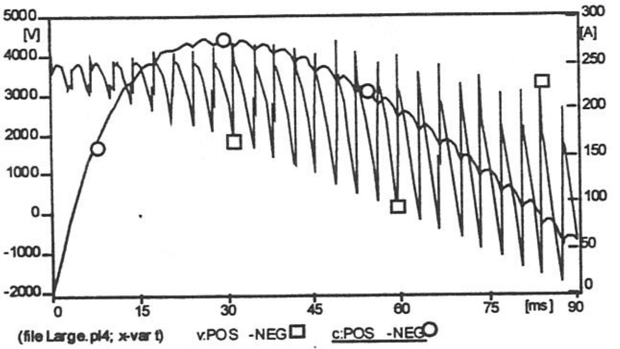


Fig. 13: DC voltage (v:POS -NEG) and current (c:POS NEG) of the circuit of fig. 12 with delay angle varying linearly from 0 up to 90 deg.

To perform a comparison with PSB, the circuit of fig. 14, that is equivalent to the one of fig. 12, is uses. The corresponding result is nearly identical to the one shown in fig. 13; the only differences lie in the overshoots during snubber transients, that are more accurately described by the PSB transient, because of its ability of automatically adjusting the time-step size.

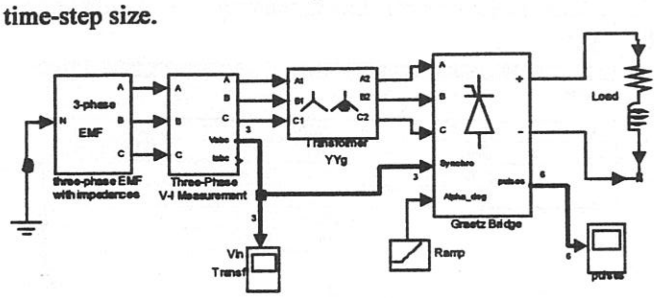


Fig. 14: A PSB circuit equivalent to the one of fig. 12.

PSB allows to easily peek at the details of a subcircuit inside block. For instance, with a simple access to a menu item, the inner description of the Graetz bridge can be evidenced. It is shown in fig. 15.

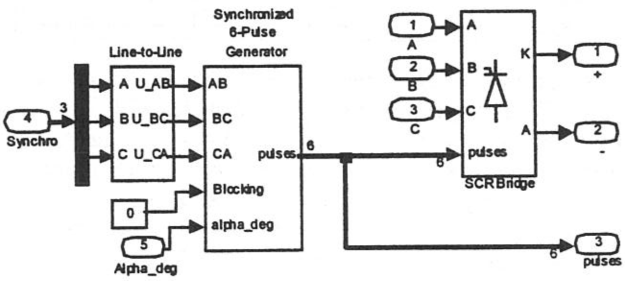


Fig. 15: Inner structure of the Graetz Bridge of fig. 14.

In turn, the synchronised 6-pulse generator has inner details that can be easily evidenced.

The circuits proposed in previous figures can be used to make some cpu-consumption comparisons.

To perform this comparison, it must be considered that. differently from ATP, and from PSB rel. 1.0, PSB 2.0 has two main simulation options:

- Variable-step algorithms (see. footnote N. 2). These offer the maximum quality of the results, since the timestep is automatically right-sized keeping into account the simulation characteristics; in particular an integration step is automatically added in correspondence to any circuit topology modification (e.g. caused by opening or closing of switches, electronic switches, diodes, etc.), and this enhances a lot the quality of final results.
- Fixed-step algorithms. To use fixed-step algorithms with PSB 2.0 the user has to require an automatic discretisation conversion of the electric network with an algorithm that *The Mathworks* says to be equivalent to the trapezoidal rule.
  In addition, it can choose to use a discrete (with transfer functions in z) or continuous (with transfer functions in s) control system. In the latter case he has also a choice among different fixed-step integration algorithms. The differences in terms of precision and speed of execution among these choices, however, are minimal and will not be analysed in depth. Just to fix ideas, in the following, when talking about fixed-step algorithms the simple *Forward Euler* (ode1) rule is intended to be used for integration of the control system.

To perform speed comparisons four cases are simulated (Table I). The corresponding cpu consumptions, obtained on a Pentium II 400 MHz machine running Windows 2000 are indicated in tab. I. Since they are reported just for giving a rough indication of the computation burden of the different considered cases all details of the PC configuration and of the condition under which the tests were performed are not reported here.

Tab. I: Comparison of cpu consumption for some of the considered cases.

Case	Set	timestep	cpu [s]
1	PSB	variable	10.3
2	PSB	fixed to 20 µs	2.16
3	ATP	fixed to 20 µs	1.50
4	ATP	fixed to 2 µs	5.66

The results obtained in the three cases are non easily distinguishable considering global plots such as those of fig. 16. However, a close up makes these differences more visible, as shown in fig. 16.

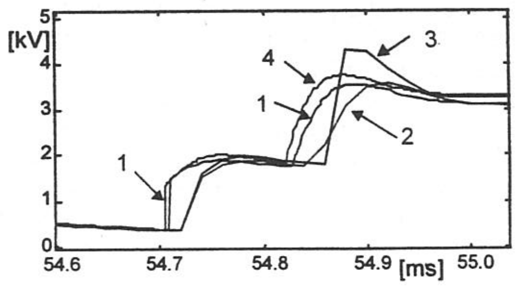


Fig. 16: A close-up of the output DC voltage showing small differences on the cases considered in Tab. I.

The following considerations can be drawn:

1. while the variable-step algorithm gives the maximum

confidence on the precision of the results, it is much slower than comparable fixed step results;
2. at equal timesteps ATP is faster but less accurate than PSB;
3. for this circuit, if the details of the commutation process are not required to be simulated with the maximum degree of precision, a fixed timestep of 20μs is adequate. This causes strong reductions of the simulation time over the solution PSB-variable timestep, that was the only available with PSB 1.0.

V. SIMPLIFIED COMPARISON OF THE TWO TOOLS

This paper was written to give a general idea of the main differences of the two simulation sets considered. Therefore a thorough comparison of the two programs is out of the scope of the paper.

However, it is deemed useful to give a compact, and not exhaustive, list of the main advantages of the two programs, based on the authors' experience.

A. *Advantages of the ATP set*

1. This is not a commercial program. It is distributed free-of charge, given that the user signs the so-called "License Form" (cf. site www.eeug.de for details).
2. It has a larger set of objects than PSB. In particular, the ATP set comes with the so-called "supporting routines": a set of tools to compute from geometry, or lab measures, electrical parameters of several components such as overhead and cable lines, saturable or hysteretic reactors, zinc-oxide surge arresters.
3. the ATP users constitute a User group that from years exchanges valuable information on ATP related problems.

B. *Advantages of the PSB set*

1. Since it is an addendum to the general MATLAB/ SIMULINK simulation environment, it can take advantage to a great deal of already programmed Matlab routines and SIMULINK blocks;
2. the PSB algorithms appear to be more precise and not to suffer from the problems of numerical stability, frequent in ATP as detailed in previous paragraph;
3. its learning-curve is steeper, mainly because with the ATP set, as currently available the user cannot do all its work within the AtpDraw/PlotXY graphical programs, but often has to resort to the old-style ASCII-file programming
4. modularisation of large circuits is much easier.

VI. CONCLUSIONS

This paper has discussed on several points-of view the different characteristics of the two modern programs for simulating electromagnetic transients: ATP and PSB.

Although a comprehensive comparison of the two products was out of the scope of this paper, a sufficient set of examples and items of discussion was presented to be able to draw a list of pros and cons of the two programs that can constitute a basis for the choice

VII. REFERENCES

[1] *ElectroMagnetic Transients Program (EMTP) Rule Book*, M39 version, Bonneville Power Administration, Division of System Engineering, P.O. Box 3621, Portland, Oregon, USA

[2] *Power System Blockset User's Guide*, The Mathworks, Natik, MA 01760-1500 `www.mathworks.com`.

[3] J. R. Marti, J Lin: "Suppression of numerical oscillations in the EMTP", *IEEE Transactions on Power Systems*, Vol. 4 N. 2, May 1989.

[4] J. Lin, J. R. Marti: "Implementation of the CDA procedure in the EMTP", *IEEE Transactions on Power Systems*, Vol. 5, No 2, May 1990.

[5] *Alternative Transients Program Rule Book*, European EMTP-ATP Users Group e.V (EEUG), `www.eeug.de`.

[6] *ATPDraw User's Manual*, freely available on the site `ftp://ftp.ee.mtu.edu/pub/atp/atpdraw/Manual`.

[7] *MATLAB, The Language of Technical Computing*, `www.mathworks.com`

[8] M. Ceraolo: "Get a Clear View Simulated and Measured Data", *IEEE Computer Applications in Power*, Oct 2000