An Integrated Simulation and Control Implementation Environment

G. Wild H. Messner A. Moosburger
Siemens,
Erlangen, GERMANY

M.H.Xie A.M. Gole
University of Manitoba,
Winnipeg, MB., CANADA

D.P. Brandt
Brandt Consultants, Inc.
Winnipeg, MB., CANADA

Abstract: **A control system for the control of HVDC, SVC and other FACTS devices can be developed, tested and debugged using an electromagnetic transients simulation program. The paper presents a procedure for the direct conversion of this developed emtp-type model into the control language for a real world implementation. The procedure is validated by comparing the simulated emtp-type output with that from the real- world control system automatically derived from this code. The procedure has been implemented on the PSCAD/EMTDC emtp-type program and produces control software that can be loaded on to the SIMADYN-D control system. The benefits of this capability are to achieve faster design of complex controls with greater accuracy and reduced engineering costs.**

Keywords: HVDC, SVC, FACTS, Digital Control Systems, Electromagnetic Transients Simulation

I. INTRODUCTION

Simulation using electromagnetic transient programs (emtp) is gaining wider application in studies concerning the performance of Power-Electronics and Controls. Such studies include High Voltage Dc Transmission Static VAR Compensation Systems and Flexible Ac Transmission Systems (FACTS). Several earlier authors have developed detailed models for the simulation of HVDC and SVC Control Systems [1,2,3]. One reason for this trend is the increasing confidence in the models for typical control blocks available in the real system. It is indeed possible to have an exact representation of the manufacturers' control blocks in an emtp-type program. One such implementation is that of the Siemens SIMADYN-D control library which has been exactly represented in the emtp-type program PSCAD/EMTDC [7].

Present day methodology requires that after the successful completion of the control and protection simulation studies, the optimized control topology as well as the controller parameters have then to be included in the actual hardware implementation. This requires great care, particularly when the control systems are very large and have an extremely large number of gains, limits, time constants and so on. It would therefore be more convenient if the actual hardware installed in the real plant be directly set from the emtp-type program. This is particularly possible today, because of the use of digital controls which are programmable via software. The authors report here on a prototype system that has been developed which allows the generation of software for the Siemens SIMADYN-D Control System from PSCAD/EMTDC. Thus, when the desired control system design has crystallized after the completion of several simulation studies, the designed control system can be directly implemented in the real controls.

The actual implementation concept is shown in Fig. 1. The SIMADYN-D system is programmed using a language called STRUC-L in which the controls are defined. This language is then compiled into assembly within SIMADYN-D and the resulting assembly code used in the DSP based control hardware. For ease of data entry, SIMADYN - D provides a graphical interface (STRUC-G) to STRUC-L. The simulation software PSCAD/EMTDC, on the other hand, also has a graphical interface (DRAFT) to generate the data files and the FORTRAN code necessary for running the actual EMTDC program. The run itself is fully interactive via the RUNTIME module, in a manner similar to analog simulators[4].

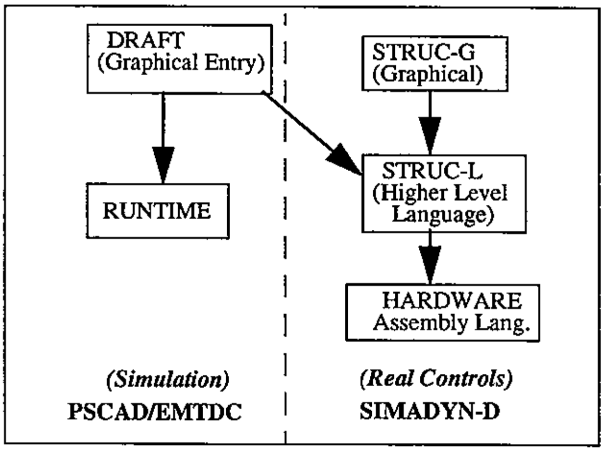


Fig. 1. Interfacing of PSCAD/EMTDC and SIMADYN-D

The link between the real controls and the simulation software is made between DRAFT and STRUC-L. The DRAFT program already contains detailed models of the blocks of the SIMADYN-D library. These blocks, when compiled, generate FORTRAN code for PSCAD/EMTDC. An additional control switch can easily be incorporated into these blocks, so that instead of generating FORTRAN code, they generate STRUC-L. The connection between blocks is already defined by the connections shown on the DRAFT palette and is the same for the simulation as well as the real implementation. In a similar manner, STRUC-L code could also be read from the real controls and implemented in PSCAD/EMTDC, although this feature is not yet implemented.

The paper will show a simple implementation of this concept using basic control building blocks.

II. DESCRIPTION OF SIMADYN-D CONTROL SYSTEM

The SIMADYN D control system [5] is a multi-processor system for fast closed-loop control and arithmetic operations, open-loop control and monitoring and for signalling and logging. The programmable SIMADYN D control system is used by Siemens for all HVDC, SVC and FACTS applications [6] where it is utilized both for control and protection functions.

A. Hardware

The multi-processor control system is made up of various plug-in boards optimally configured in a rack for HVDC and SVC type applications. Several different processor boards can be used depending the application and peripheral interface requirements with up to eight processor boards in one sub-rack. Each processor board has its own program and data memory and processes its allocated tasks independent of other processors. The program memory sub-module inserted in each processor board contains the system and application software. The parallel processors communicate via a local bus through a communication buffer board. Input/output boards interface the processors to external systems and signals with normalized input and output levels. Interface sub-modules which can be inserted in the processor board provide standardized serial interface (20 mA, V.24 or EIA 485) for data exchange with other equipment and systems. Separate Communication boards are also used for serial data exchange if required.

B. Software

The programming of control or protection functions, including all control parameter settings, is done in a high level programming language called STRUC-L, with function blocks available from an extensive library of control functions. The library consists of control blocks, converter specific blocks, arithmetic blocks, logic and switch blocks, and input/output blocks. Execution time step of these function blocks from 100 μs and up are assigned according to the response time requirements of the particular control loop or protection application. Up to five different sampling times can be assigned to each processor. The valve firing system ("trigger set") and extinction angle measurement are specially created function blocks for HVDC applications with corresponding specific high-speed processors that ensure firing pulse and extinction angle accuracy of ±0.1°. The software also includes the communication functions to handle data transfer within the SIMADYN D system and with external systems. Diagnostic software identifies errors/faults in the SIMADYN D system and provides external signals to allow the appropriate contingency actions such as switch-over to a redundant control channel. The software is self documenting and the control structure is directly entered using a graphical user interface (called STRUC-G) as in Fig. 2. On compiling, STRUC-L code is produced as shown in the sample listing in Fig. 3.

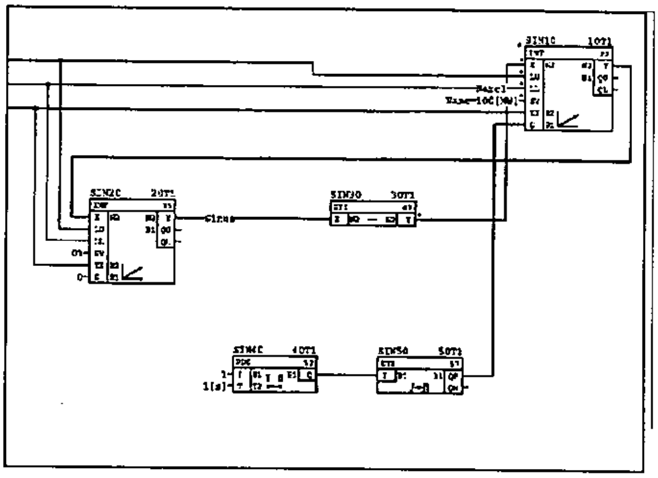
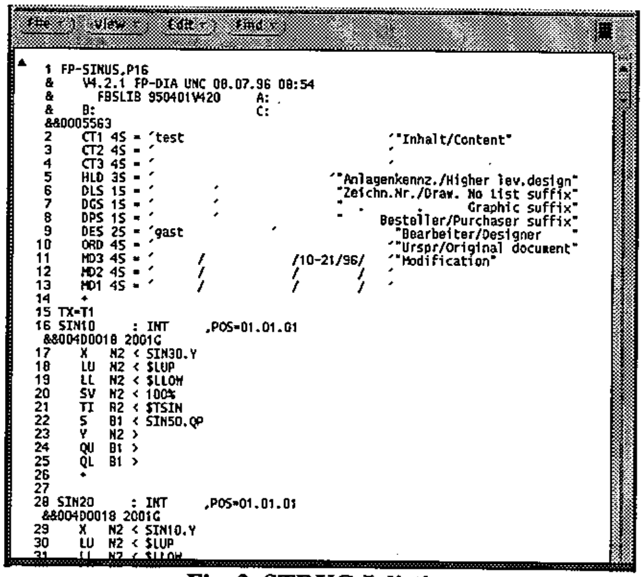


Fig. 2. SIMADYN-D Graphical User Interface



```
1 FP-SINUS.P16
&    V4.2.1 FP-DIA UNC 08.07.96 08:54
&      FBSLIB 950401V420       A:
&    B:                        C:
&&0005563
2     CT1 4S = 'test                      '"Inhalt/Content"
3     CT2 4S = '                          '
4     CT3 4S = '                          '
5     HLD 3S = '                    '"Anlagenkennz./Higher lev.design"
6     DLS 1S = '        '           "Zeichn.Nr./Draw. No list suffix"
7     DGS 1S = '        '           "                 Graphic suffix"
8     DPS 1S = '        '           "        Besteller/Purchaser suffix"
9     DES 2S = 'gast          '        "Bearbeiter/Designer     "
10    ORD 4S = '                        '"Urspr/Original document"
11    MD3 4S = '     /          /10-21/96/  '"Modification"
12    MD2 4S = '     /          /        /  '
13    MD1 4S = '     /          /        /  '
14    *
15 TX=T1
16 SIN10     : INT     ,POS=01.01.01
 &&004D0018 2001C
17    X   N2 < SIN30.Y
18    LU  N2 < $LUP
19    LL  N2 < $LLOW
20    SV  N2 < 100%
21    TI  R2 < $TSIN
22    S   B1 < SIN50.QP
23    Y   N2 >
24    QU  B1 >
25    QL  B1 >
26    *
27
28 SIN20     : INT     ,POS=01.01.01
 &&004D0018 2001C
29    X   N2 < SIN10.Y
30    LU  N2 < $LUP
31    LL  N2 < $LLOW
```

Fig. 3. STRUC-L listing

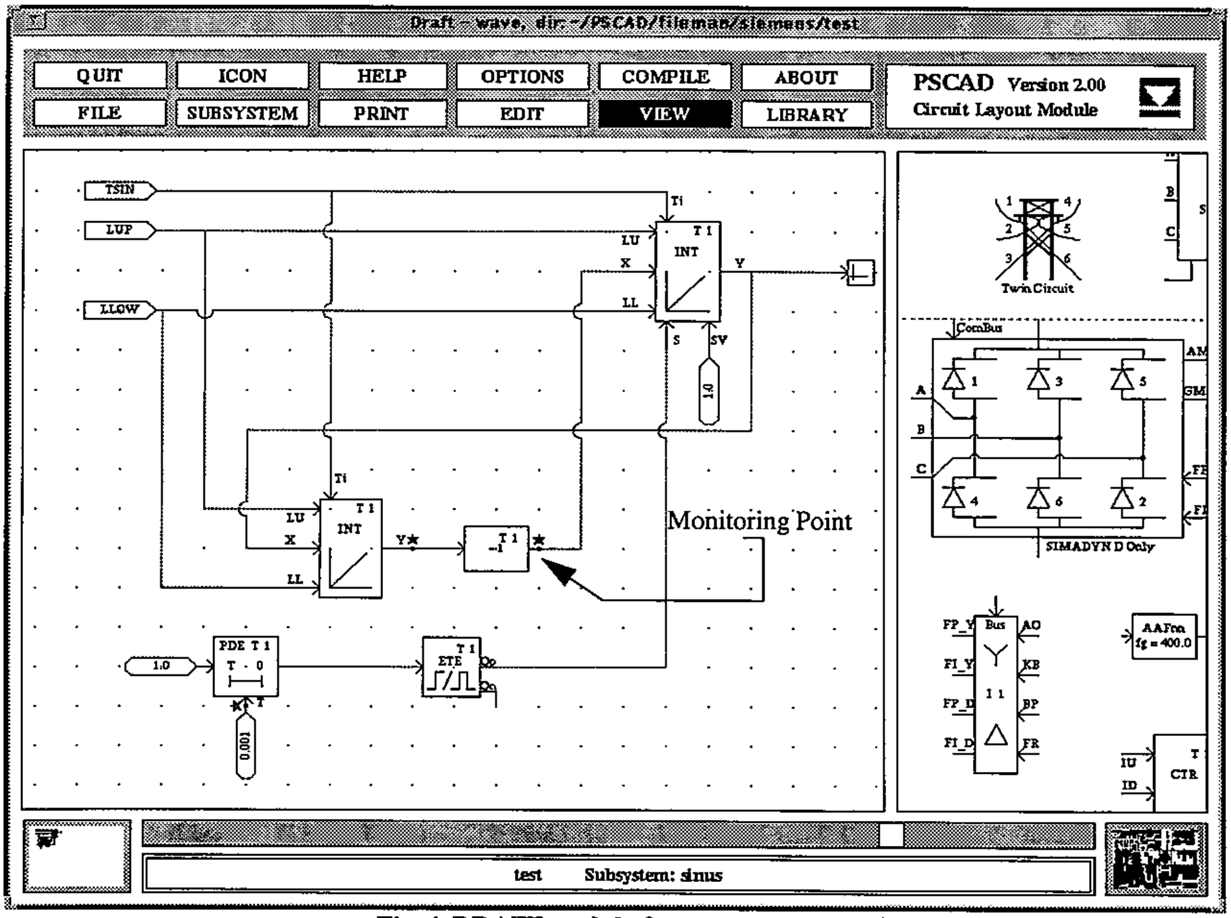


Fig. 4. DRAFT module for wave-generator circuit

III. DESCRIPTION OF GRAPHICAL SIMULATION TOOLS

PSCAD is a graphical front-end for the emtp-type simulation program EMTDC. It allows the user to make schematic drawing of control circuits using one of its modules called DRAFT. In DRAFT, the complete system (network as well as controls) can be divided into many subsystems. This allows one to represent the controls in a modular fashion, with each subsystem representing one specific functionality. Every sub-system is a drawing board on which many PSCAD components can be drawn as in Fig. 4. Each PSCAD component has a definition file defining its graphical appearance and underlying FORTRAN simulation code. The data for the component can be entered by filling in the popup menu that appears when the component is clicked on with the mouse.

In addition to the vendor-supplied components, users can also compose their own components [4]. Using this approach a comprehensive library has been developed and tested for the SIMADYN D system [7]. Fig. 4 shows one subsystem of the PSCAD DRAFT module, in which a SIMADYN-D wave generator is simulated.

The circuit developed in DRAFT is compiled and run through an operator console type interface called RUNTIME. This interface allows the user to make changes to the run while it is in execution, in much the same manner as a operator at a system control console[4]. Gains and set-point can be varied on line, switches opened and closed, system quantities can be read on meters or plotted on graphs and so on.

IV. DEVELOPMENT OF CONTROL GENERATION PROGRAMS

One of the files internally generated by DRAFT (referred to as the DRAFT file) contains the position information and parameter table of each component. Although not originally intended for this purpose, this file now provides a convenient starting point for generating SIMADYN-D control code instead of the FORTRAN code that is used in an EMTDC simulation. A PSCAD-SIMADYN D compiler was developed to translate this DRAFT file to SIMADYN D's STRUC-L language.

A. Analysis of the two systems

Several similarities as well as differences exist between the representation of the system in the simulation program and the actual field implementation. The similarity exists mainly in the aspects of overall system behavior and the control circuit structure. However, in the real-world implementation, there are usually additional details, such as analog to digital conversion interfaces, redundant modules and input/output functions, which are simplified in simulation. On the other hand, in the simulation system, there can be several functions mainly used to monitor or control the simulation which do not need to be included in the real-world control system. Thus, it is very important to define a border in the simulation system, to distinguish those functions which are to be compiled into real controls from those that do not need to be compiled. Fig. 5 shows the relationship between a simulation system and a real control system.

It was found out that an effective method to realize this is to first partition the simulation circuit into different sub-systems. Any subsystems that do not need to be compiled into real controls, are identified and marked as "non-compiled subsystems". The PSCAD-SIMADYN D compiler program filters out these subsystems generates the real-world control code from the remaining.

Thus only the part of the controls relevant to the real-world implementation are selectively translated to SIMADYN D control software for downloading to the controller boards.

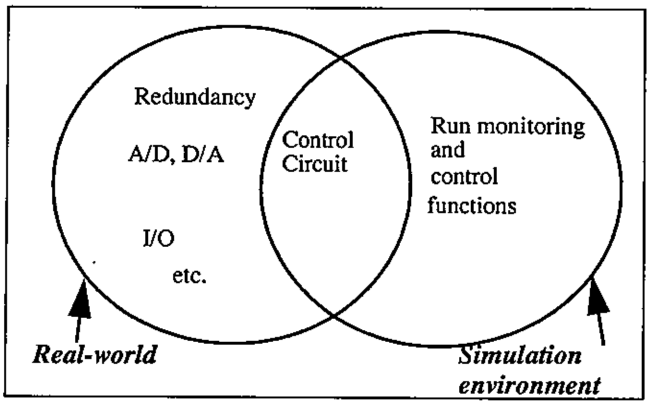


Fig. 5. Control Representations in Simulation and in the Real-world

B. Treatment of Special Control blocks not Being Simulated

As mentioned above, there are several features in SIMADYN-D that are not necessary for the PSCAD/EMTDC simulation. For example, a monitoring point may be required in the field implementation for on-site debugging which is of no consequence in the simulation. Since PSCAD is also generating the real-world code, it would be convenient if this point could be specified in DRAFT. No data would be generated for the EMTDC simulation; however when the control code for SIMADYN D is generated, this point would be made available. Such points are indicated in DRAFT with the asterisk label as in Fig. 4, and are entered via a specially developed PSCAD component called "modifier" . Fig. 6 illustrates the pop-up menu form for this component.

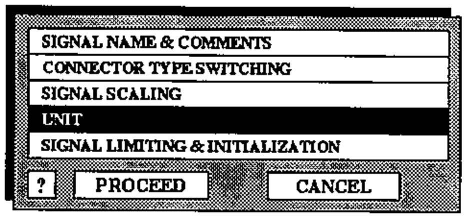


Fig. 6. Input Form for modifier

C. Treatment of Special Simulation Components

On the other hand, some features in PSCAD are very convenient for simulation purposes, but do not need to be compiled literally into the real controls system. Such features include interactive components such as on-line gain setting, push-buttons and many others monitoring features such as meters and plots.

There are also some small differences in the simulation models and the real-world models which arise mainly from conventions used. These include the interpretation of wire cross-over, input/output notation and so on. In such cases a separate subroutine has to be written for each PSCAD component so that the circuit compiler reproduces the appropriate features for the type of implementation (simulation or real-world) desired.

In this project, Object-Oriented Programming is utilized everywhere. A generic PSCAD component is defined as the base class for all components, and the above mentioned special components are derived form the generic component, they inherit many important features from the generic PSCAD component (such as file format and position information), but also have their own special functions to inform the compiler program how to convert them into real control function blocks.

D. Development of Compilation Library

In order to generate real controls from the simulation, the PSCAD-SIMADYN code compiler has to look up a library that associates the corresponding SIMADYN D block and STRUC-L code function blocks with a given PSCAD block. This library can be generated at development time with great ease with the use of a specially developed Windows 95 based assignment program, which runs independently of PSCAD/EMTDC. The program interface can be seen in Fig. 7, where the upper (right) window shows the PSCAD component and the bottom window shows the corresponding SIMADYN D component. The correspondence between the two icons is then established graphically. Once this library file is developed, all further development can be carried out in the PSCAD environment without any further reference to the assignment program.

Another benefit of using library files is that different versions of SIMADYN D can easily be incorporated. Such need arises, for example, when upgrading the controls to a different processor type (such as from a 16 bit to a 32 bit SIMADYN D processor).

For the most part there is one PSCAD component for each SIMADYN D component, with the two having an identical graphical icon. However sometimes it is convenient to have a simple PSCAD representation for a class of SIMADYN D functions. For example the summing junction component in PSCAD (called sumjcts2_s) can be configured to select he SIMADYN D components for addition ADD2, ADD24 (which has more accuracy); as well as the components for subtraction, namely SUB and SUB04. This convenience arises from the fact that functionally all four blocks perform a similar action, and incorporating these into a single class results in a smaller well organized PSCAD library. This feature of allowing different SIMADYN D representations for variants of the same PSCAD block has also been constructed into the assignment program. The tree structure shown in the left hand side window of Fig. 7 shows several SIMADYN D blocks that can be selected from the same PSCAD block.

E. Overall structure

The design cycle for the controls can thus be summarized as follows:

1. Several simulation study cycles of the controls and power system network are carried out in PSCAD/EMTDC and the final control strategy is defined.
2. The subsystems that do not need to be compiled for SIMADYN D are identified and marked.
3. Additional modifier components are added as necessary (optional).
4. The DRAFT to STRUC-L conversion program is invoked.
5. STRUC-L code is flashed onto the boards via an EEPROM (Electrically Erasable Programmable Read Only Memory) programmer.

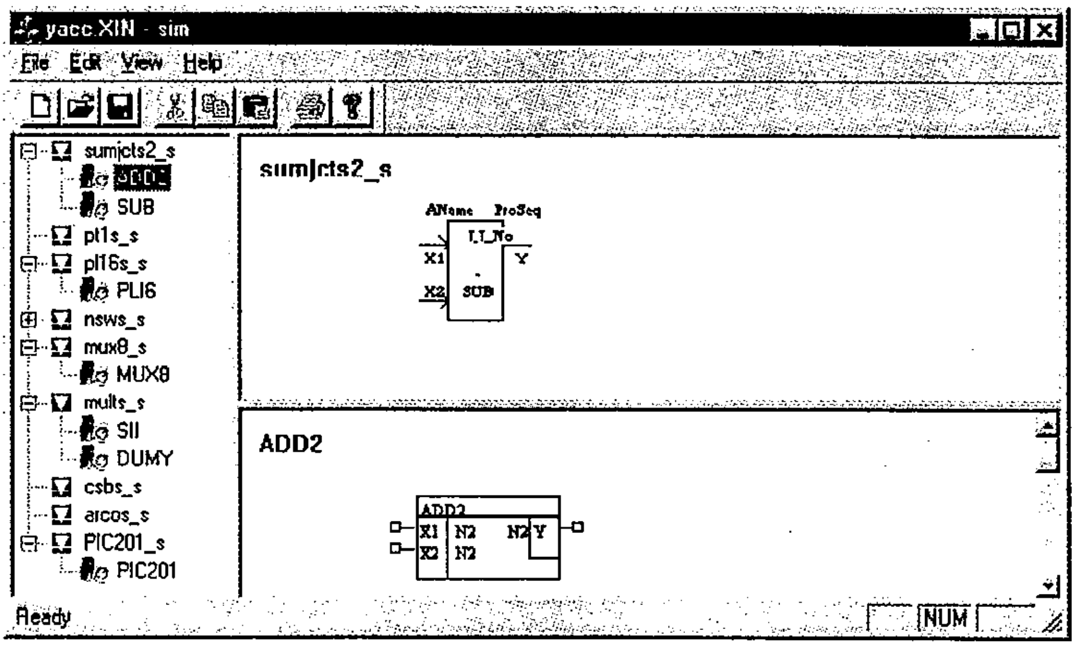


Fig. 7. Graphical application for library construction

V. VALIDATION OF THE CONTROL GENERATION PROGRAMS

A first test of the program is a self-contained wave generator, which is first simulated using PSCAD/EMTDC. Then the control software is generated using above programs and library, the results are then uploaded into the SIMADYN-D control hardware and the final output is produced. Fig. 8 shows a comparison between the simulation result and the real control output plot. An exact match is found between these two outputs, which validates the correctness of the compiler program.

At this stage tests have been conducted on simple test control systems. The next stage already under way, will evaluate the development approach by considering a detailed HVDC control system.

VI. CONCLUSIONS

It is possible to develop an integrated environment for simulation and control development. In this environment, the proposed control system can be thoroughly analyzed through digital simulation, and the crystallized design readily converted to the software code for the real-world controller.

The features available for control modelling in PSCAD/EMTDC can be exploited to develop this technique. An off-line assignment program can make the task of assigning SIMADYN D code to the PSCAD icon fairly straightforward.

The use of the integrated development/simulation environment will speed up the design and implementation of complex controls and reduce the engineering costs.

VII. REFERENCES

[1] A. Hammad et al: "Controls Modelling and Verification for the Pacific Intertie HVdc 4-Terminal Scheme", IEEE Transactions on Power Delivery, Vol. 8, No. 1, January 1993.

[2] P. Kuffel, K.L. Kent, G.B. Mazur and M.A. Weekes "Development and Validation of Detailed Controls Models of the Nelson River Bipole I HVdc System". IEEE Transactions on Power Delivery, Vol. 8, No. 1, January 1993.

[3] G. Morin et al: "Modelling of the Hydro Quebec - New England HVdc System and Digital Controls with EMTP", IEEE Transactions on Power Delivery, Vol. 8, No. 2, April 1993.

[4] A.M. Gole, O.B. Nayak. T.S. Sidhu, M.S. Sachdev, "A Graphical Electromagnetic Simulation Laboratory for Power Systems Engineering Programs", IEEE Trans. on Power Systems, May 1996

[5] SIMADYN D Digital Control System, Standard Function Blocks, Siemens AG 1988.

[6] D. Knittler and L.Huegelschaefer," Experience with a Digital Fully Redundant Control System for HVdc-Plants', EPE, Firenza, 199 1.

[7] K. Sadek, G.Wild, L. Huegelschaefer, A. Gole, X. Jiang, D. Brandt "Modelling of Digital HVDC Control Systems using a Graphical Electromagnetic Simulation Program", IPST, Sept. 1995, Lisbon, Portugal.

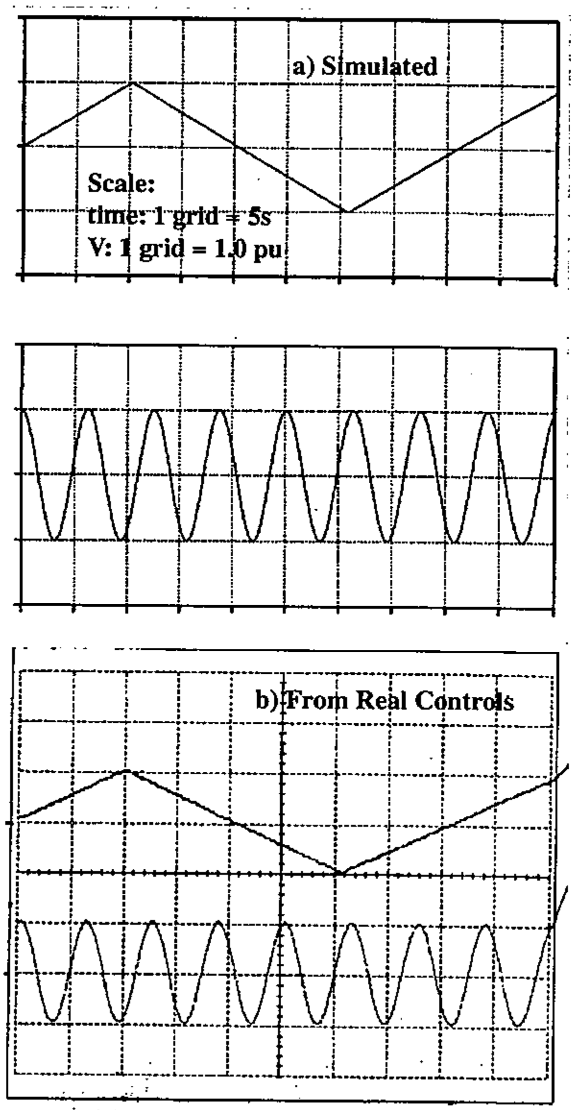


Fig. 8. Comparison of the Results for Waveform-generator