

HIL Simulation of a Mixed Islanded Power Network with External DSP Regulator

Nicolas Junod¹, Philippe Allenbach¹, Sylvain Robert², André Hodder², Gyorgy Banyai¹ and Basile Kawkabani¹

1. Ecole Polytechnique Fédérale de Lausanne, Electrical Machinery Laboratory, Lausanne CH-1015, Switzerland

2. Ecole Polytechnique Fédérale de Lausanne, EPFL-STI-VDFORM, Lausanne CH-1015, Switzerland

Received: May 25, 2011 / Accepted: September 23, 2011 / Published: July 31, 2012.

Abstract: The present paper deals with the development of a modular, flexible and structured block to block approach for the study of regulators by implementing the different blocks on a DSP (digital signal processor). The proposed low-cost approach has been applied and validated by the implementation of an industrial regulator in a real time hardware-in-the-loop simulation of a mixed islanded power network including precise models of the hydraulic system. The studied network is constituted of three different types of electrical power generation systems and a consumer.

Key words: DSP (digital signal processors), RTS (real time systems), power system simulation, PWM (pulse width modulation), regulators, HIL (hardware-in-the-loop simulation), DLL (dynamic link library).

Nomenclature

| | |
|-----------------|-------------------------------------|
| Coeff1 | Coefficient 1 |
| Coeff2 | Coefficient 2 |
| C_p wind | Turbine power coefficient [-] |
| C_{wind} wind | Velocity [m/s] |
| Ext | External |
| GEN | Generator |
| lambda | Tip speed ratio [-] |
| N | Generator speed [rpm] |
| P | Power [W] |
| t | Time [s] |
| TC | Computation time [s] |
| TE | Read/write time [s] |
| Theta | Blade pitch angle [deg] |
| T_s | Simulation data exchange period [s] |
| $T's$ | External device sampling period [s] |
| ΔT | Step time [s] |
| uc | Set point voltage [V] |
| um | Measured voltage [V] |
| U | Voltage [V] |

Indexes

| | |
|-------|-----------|
| hydro | Hydraulic |
| therm | Thermal |
| wind | Wind |

| | |
|------|--------------------------|
| load | Load |
| ex | Excitation |
| l | Line |
| rms | Root mean square |
| soft | Software |
| dsp | Digital signal processor |
| dll | Dynamic link library |

1. Introduction

During the last few years, many studies have been dedicated to the development of real-time digital simulators employed in the simulation of power systems and converters. The state-of-the-art in interfacing issues related to such real-time digital simulators as well as HIL (hardware-in-the-loop) interfacing for controller hardware and power apparatus hardware are presented in details in Ref. [1]. HIL simulations are already used in different applications to test critical functionalities. An overview of existing work in different domains is given in Refs. [2-7] (electric ship [2, 3], motor drive systems [4], power electronic converters [5], distributed generation systems [6], test platform for hybrid electric power systems [7]). Real-time HIL testing approaches for power systems modeling and

Corresponding author: Basile Kawkabani, Ph.D., senior scientist, research fields: modeling of power systems, power system stability and control. Email: basile.kawkabani@epfl.ch.

simulations are presented in Refs. [8-10]. These HIL systems are often based on expensive hardware or operate only for a given hardware device.

A HIL extension, called SIMSEN-RT, of an existing simulation software SIMSEN [11] running on a standard PC has been developed in the laboratory [12]. This HIL system can be used for fine-tuning regulator parameters and for testing of regulators in different operating conditions (islanded network, emergency shutdown...). The main aim of this study is to develop and validate a modular, flexible and structured block to block approach for the study of different regulators by implementing the different blocks on a DSP (digital signal processor). This low-cost approach has been applied and validated by the implementation of an industrial voltage regulator in a DSP (Fig. 1).

An event such as the decrease of wind in a mixed islanded power network described in Ref. [13] has been used as a test case for this development. The interest of this case is to be realistic and to be at the limit of the computing power of a standard personal computer, in real time applications.

The studied power network represented in Fig. 2, is constituted of three different types of electrical power generation and a consumer:

- Hydraulic (300 MW);
- Thermal (1300 MW);
- Wind (200 MW).

This paper is organized as follows: The communication between the simulation software and the external hardware regulator is described in Section 2. Section 3 describes the block structure of the regulators as well as their implementation in DLL (dynamic link library). The studied system is presented in Section 4. The comparison between DSP/DLL and software regulators is given in Section 5. Finally, conclusions are given in Section 6.

2. Description of the HIL Structure

2.1 Signal Path

At each simulation time step a state value is sent by

the software (SIMSEN-RT) to the DSP regulator through interface cards needed for AD (analog/digital) translation as well as voltage adaptation. Fig. 3 shows the path of the signal representing a simulation value. The simulation software first sends digital signals to the acquisition card which converts it to an analog voltage through a DAC (digital/analog converter). This analog signal is then sent to an external regulator (a DSP regulator in this case). The DSP regulator performs its computation and outputs a PWM signal (because the used DSP has no analog output). This signal is then converted to an analog voltage through a

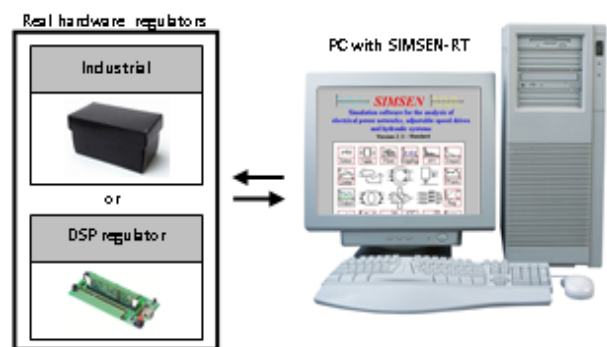


Fig. 1 HIL principle.

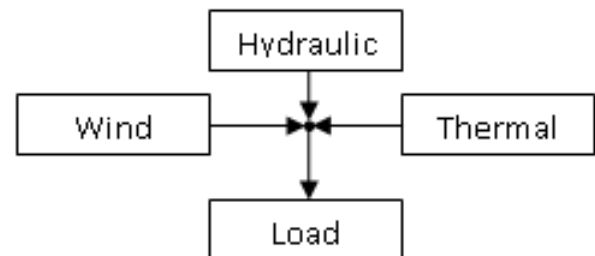


Fig. 2 Network diagram.

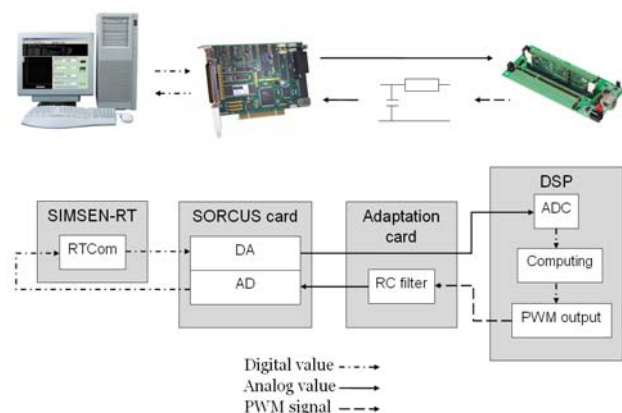


Fig. 3 Signal path.

RC filter and sent back to the acquisition card that converts it to a digital simulation value with an AD converter.

2.2 Signal Adaptation

In each step of the signal path there is a need to adapt the signal to the range of values accepted by the different elements. As an example, Fig. 4 shows how to convert a simulation value to an analog voltage. The external voltage range is between zero and 3 V due to the AD converter of the DSP. The RMS (root mean square) line voltage range is between 0 and the double of the nominal voltage (17,500 V in this case).

2.3 HIL Exchange Timing

Fig. 5 represents the communication scheme used to realize a HIL simulation. The simulation and the tested hardware are independent and each one has its own sampling period. The simulation sets its output to a new value every T_s seconds and reads its inputs at the same time. The tested hardware regulator sets its output every $T's$ seconds and reads its inputs at the same time.

To perform a HIL simulation, it is required that the simulation runs in real time. This means that each integration step (ΔT) of a simulation has to be computed faster than the real time to let time spared for the communication between the simulation and the tested hardware. This is resumed in the following requirement:

$T_E + T_C \leq T_s$. T_E is the time necessary for reading and writing the data. The computation time (T_C) is influenced by the topology of the simulated installation. The computation time of a basic installation is lower than for a complex installation. The integration step has a big influence on the computation time. To get a short computation time, the integration step has to be large. The integration step is constrained by the smallest time constant of the physical phenomena of the simulated installation. A typical integration step is 1 ms for an electrical installation

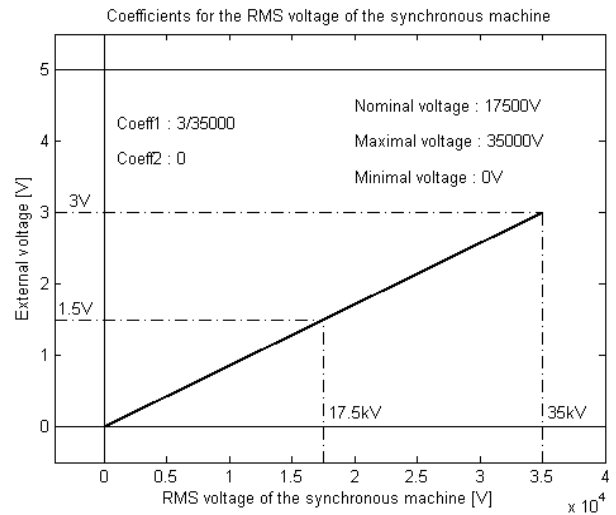


Fig. 4 Coefficients.

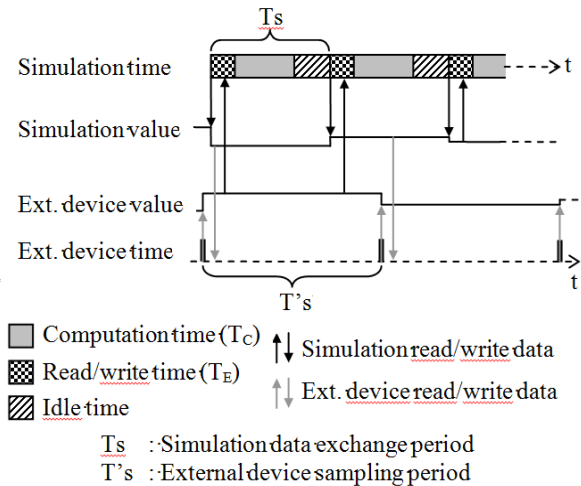


Fig. 5 HIL communication principle.

with electrical machines and without power electronics. The simulation software used in this work is able to check whether the real time condition is respected or not. A tolerance value can be chosen so to allow for small discrepancy between real time and simulation time. If the difference is too large, the simulation stops.

3. Regulator Code Structure

In this work the code structure has been designed such as to allow the composition of the regulator in a modular way and an easy transposition in the different implementations (inside the simulation program, in an external program, in a DSP).

3.1 Block Structure of Regulators

In the simulation software a regulation scheme can be built using basic blocks; a C code structure is created to make easy to transpose such a block diagram to a C code. When the block diagram of an industrial regulator is known, it is easy to implement it in the DSP. Each module of a block diagram has its C code equivalent.

Fig. 6 shows some basic blocks used to create a regulator. The code is designed in an object oriented manner with a set of fundamental elements and their copy for each block of a regulator scheme.

The program has two distinct parts, one defining basic block structures and one using these structures to create blocks. These blocks have their parameters filled and are then connected to the other blocks of the regulator scheme. Here is an example of the C code line corresponding to the use of the basic “reg” block.

3.2 Implementation of Regulators in DLL

A DLL is a kind of program that is meant to stay in memory and be called by other programs. In this work it has been taken advantage of so as to be able to develop and test the code of regulator programs on the computer before implementing it on the DSP. The block structure of the regulator code has been put to full use to ease its transfer between the DLL and the DSP.

```
f_reg (reg1, setpoint, value);
```

The output of the regulator reg1 is then accessible by reading its output parameter: reg1.out.

Fig. 7 shows how to translate a DSP C code for a DLL. The DSP has to be initialized with a specified C code.

This part of C code is no more needed for the DLL. It means that this part of the code has to be removed to generate a DLL file.

The DSP C code used to read an analog voltage through an AD converter has to be modified for the DLL C code as there is no AD converter used by the DLL. The digital simulation value is directly sent to the DLL without any conversion.

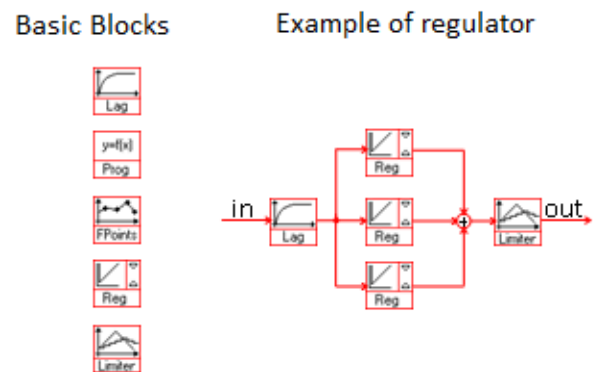


Fig. 6 Block diagram of a regulator.

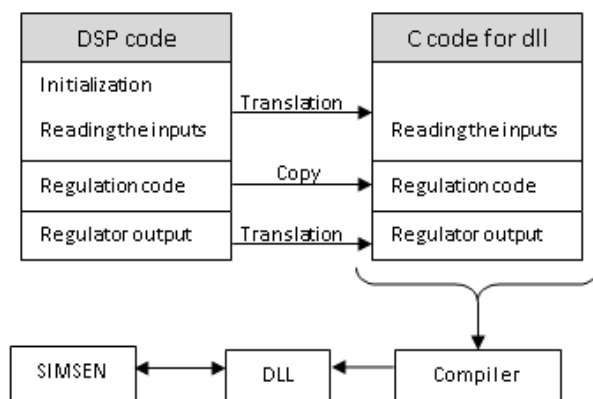


Fig. 7 DLL generation.

Most importantly, the part of the C code used for the regulation can be simply copied from the DSP C code to the DLL C code.

The output DSP C code has to be adapted for the DLL because there is no more PWM signal. The digital output value of the DLL is directly sent to the simulation software without any conversion.

After these modifications, the new C code is compiled by a DLL compiler to generate a DLL file that can be used by the simulation software.

3.3 Implemented Voltage Regulator

In this study an industrial voltage regulator is implemented in C code. The full structure comprises around 150 basic blocks.

4. Simulated Installation

This section describes the mixed islanded power network simulated in this study such as illustrated in Fig. 8. The thermal power plant produces a constant

electrical power, the wind turbine installation produces variable electrical power depending on wind speed; the hydraulic installation is used to compensate variations in wind and load consumption. The lines voltages of the three generators are set by three voltage regulators. The wind turbine speed is set by a speed regulator that changes the pitch angle of the blades. If the wind is too slow or too fast, the wind turbine stops. The speed of the hydraulic turbine is set by a speed regulator that changes the guide vane opening of the turbine. The speed of the thermal installation is not set by a specified regulator. A constant torque is applied on the rotor of the thermal power plant and the speed is determined by the two previous speed regulators.

The voltage regulator of the hydro-electrical installation is replaced by an external hardware regulator (DSP regulator). The “VReg” block represents the software regulator, the “RTCom” block represents the link to the external DSP regulator and the “Extern” block represents the link to the DLL regulator. These three regulators are represented in Fig. 8. Only one of them is used during a simulation, the two others must be disabled. They are just three equivalent implementations of the

same regulator.

5. Tests and Results

The behavior of the mixed islanded power network is tested with wind speed changes. Fig. 9 shows the speed evolution of the wind.

The wind turbine increases its power generation during the wind speed increase. The thermal power and the load power consumption stays at the same value so the variation of wind power is compensated by the hydro-electrical installation.

5.1 Comparison between DSP and Software Regulators

Figs. 10-13 show the comparison of the active power evolution using a software regulator or a DSP regulator. The power evolution is the same in both cases (curves are superimposed).

Table 1 shows the power balance evolution. The wind speed increases between 100 sec and 170 sec with some fluctuations and the power generated by the wind turbine increases too. The increase of wind power (Fig. 12) is compensated by a decrease of hydraulic power generation (Fig. 10). The load power consumption is given in Fig. 13.

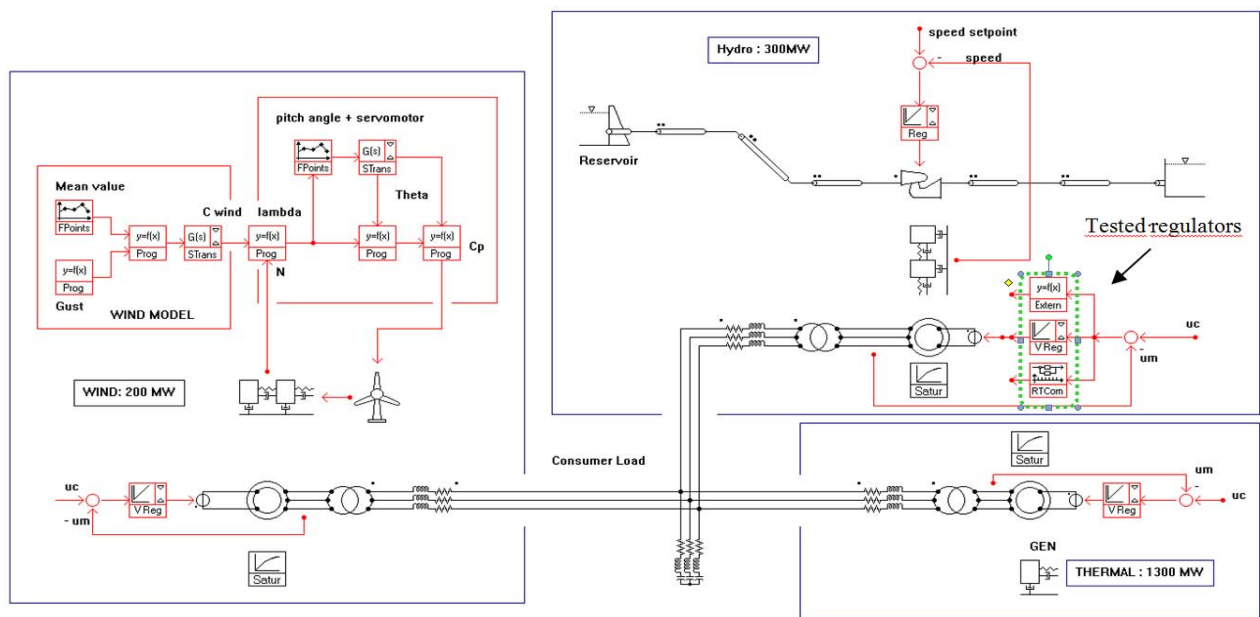


Fig. 8 Mixed islanded power network.

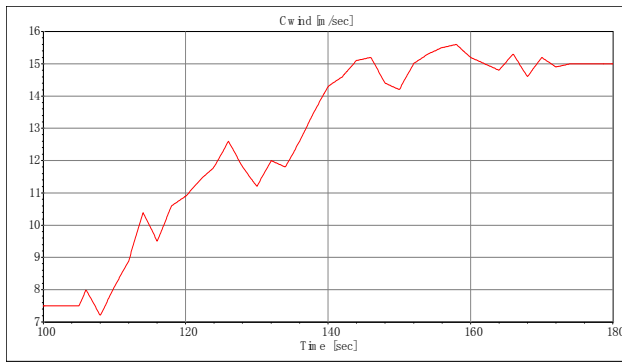


Fig. 9 Wind speed.

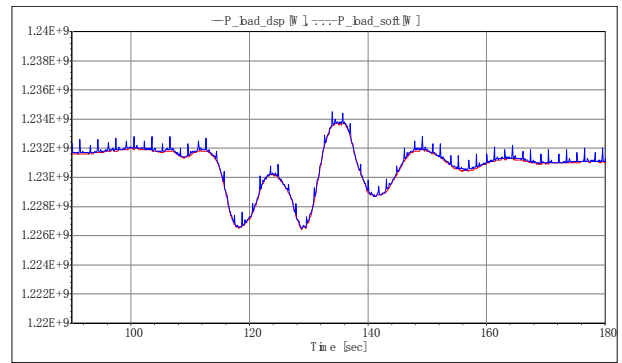


Fig. 13 Load power consumption (DSP-software).

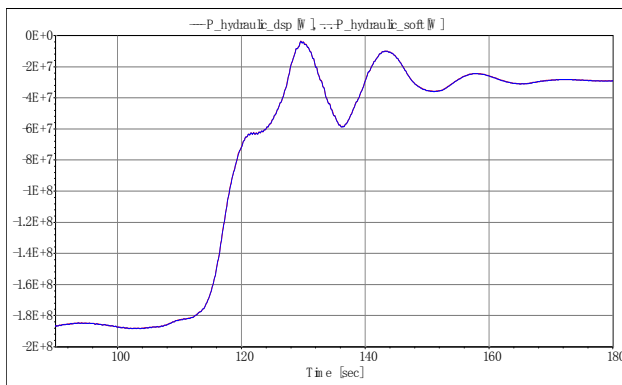


Fig. 10 Hydraulic power (DSP-software).

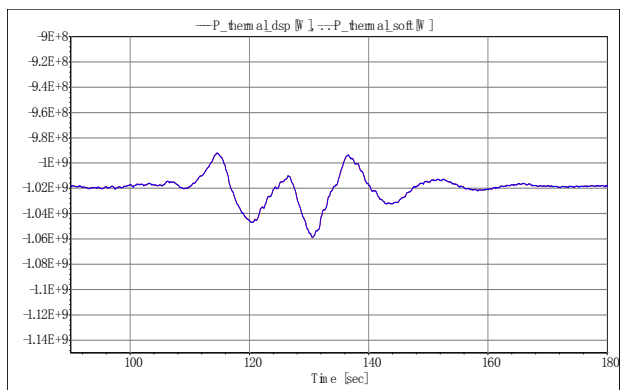


Fig. 11 Thermal power (DSP-software).

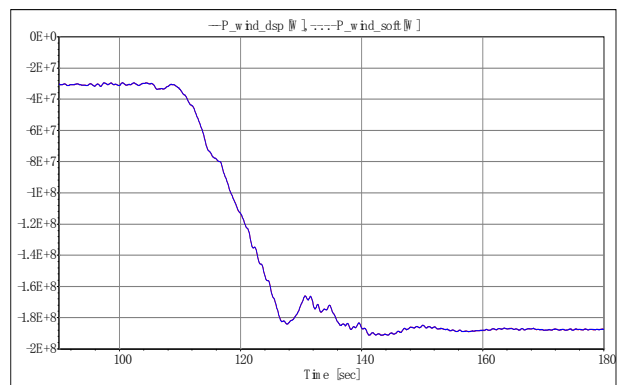


Fig. 12 Wind power (DSP-software).

Table 1 Power balance.

| Element | Active power (MW) | |
|-------------------------------|-------------------|------------|
| | At 100 sec | At 180 sec |
| Hydraulic | -187 | -29 |
| Thermal | -1,018 | -1,018 |
| Wind | -30 | -188 |
| Load | 1,232 | 1,232 |
| Losses (lines & transformers) | 3 | 3 |

Table 1 shows that the power consumed by the thermal power (Fig. 11) and the load (Fig. 13) is the same before and after the wind speed increases. The no-load excitation voltage corresponding to the nominal line voltage is equal to 209 V.

Fig. 14 shows a comparison between the excitation voltage given by the software regulator and by the DSP regulator. The two curves follow the same evolution but the DSP output contains noise that is mainly added by the computations performed in the DSP. The nominal RMS line voltage of the synchronous machine of the hydro-electrical installation is 17,500 V.

Fig. 15 represents the evolution of the RMS voltage of the synchronous machine used in the hydraulic installation. The RMS voltage follows the same evolution when it is set by the software regulator or by the DSP regulator. A little offset of 1 V is nevertheless existing due to the AD and DA (digital/analog) conversions.

5.2 Comparison between DLL and Software Regulators

As mentioned before, another part of this project was to generate DLL programs with a slightly modified

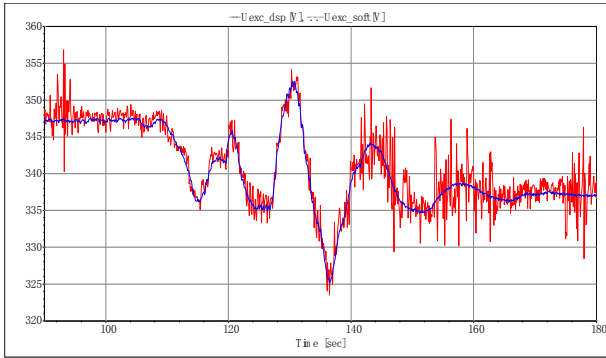


Fig. 14 Excitation voltage (DSP-software).

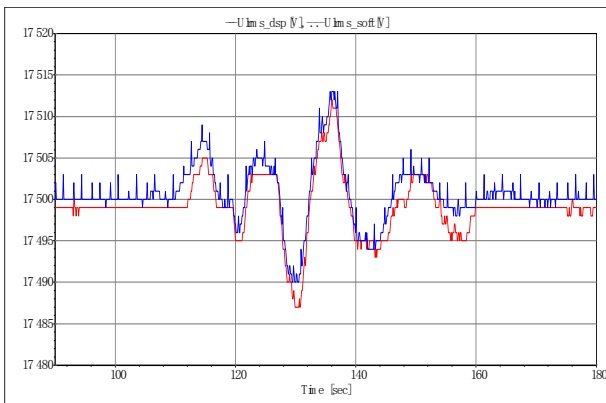


Fig. 15 Line voltage (DSP-software).

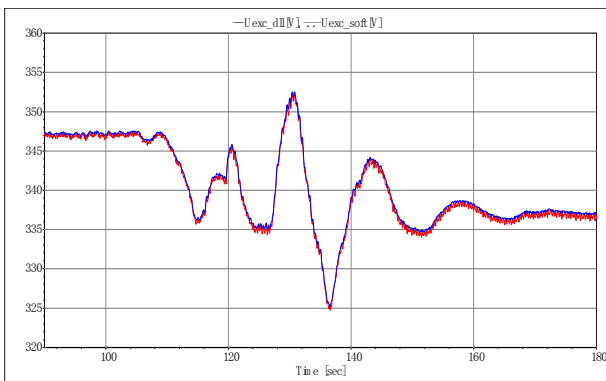


Fig. 16 Excitation voltage (DLL-software).

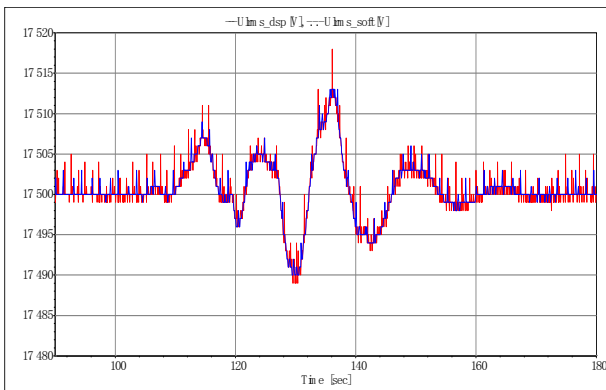


Fig. 17 Line voltage (DLL-software).

version of the DSP code. Figs. 16 and 17 show a comparison between the software regulator and the DLL regulator. The excitation voltage and the line voltage represented in these figures have the same evolution with the software regulator and the DLL regulator, which validates the proposed approach.

For this mixed islanded power network, the frequency remains in an acceptable range of variation (49.85 Hz; 50.55 Hz) for the studied case in both implementations (DLL, DSP).

6. Conclusions

The main aim of this study was to perform a HIL simulation with a voltage regulator implemented in a DSP. The results obtained are convincing. This project has illustrated the usefulness of a modular structure to implement regulators and translated them in different environments. A block diagram of a regulator can be easily translated in C code implemented in DLL files, then tested and in the end transferred to a DSP. Finally this project shows that it is possible to perform HIL simulations of relatively complex systems on a standard computer.

Acknowledgments

The authors would like to thank Dr. Christophe Nicolet for his contribution to the simplification of the hydraulic part of the simulated system to decrease the computation time without any loss of accuracy regarding the physical behavior.

References

- [1] W. Ren, M. Sloderbeck, M. Steurer, V. Dinavahi, T. Noda, S. Filizadeh, et al., Interfacing issues in real-time digital simulators, *IEEE Transactions on Power Delivery* 26 (2) (2011) 1221-1230.
- [2] W. Ren, M. Steurer, S. Woodruff, Applying controller and power hardware-in-the-loop simulation in designing and prototyping apparatuses for future all electric ship, in: *IEEE Electric Ship Technologies Symposium, USA, 2007*.
- [3] M. Steurer, S. Woodruff, T. Baldwin, H. Boenig, F. Bogdan, T. Fikse, et al., Hardware-in-the-loop investigation of rotor heating in a 5 mw hts propulsion

- motor, *IEEE Transactions on Applied Superconductivity* 17 (2) (2007) 1595-1598.
- [4] C. Dufour, H. Blanchette, J. Bélanger, Very-high speed control of an FPGA-bases finite-element-analysis permanent magnet synchronous virtual motor drive system, in: 34th Conf. IEEE Industrial Electronics Society, Florida, 2008.
- [5] L. Gregoire, K. Al-Haddad, G. Nanjundiah, Hardware-in-the-loop (HIL) to reduce the development cost of power electronic converters, in: India International Conference on Power Electronics (IICPE), 2011, pp. 1-6.
- [6] M. Mauri, F.C. Dezza, G. Marchegiani, Hardware in the loop (HIL) test bench for small-scale distributed generation systems, in: IEEE International Symposium on Industrial Electronics ISIE, 2008, pp. 2177-2182.
- [7] J.M. Timmermans, J.V. Mierlo, P. Lataire, F.V. Mulders, Z. McCaffree, Test platform for hybrid electric power systems: Development of a HIL test platform, in: European Conference on Power Electronics and Applications, 2007, pp. 1-7.
- [8] R. McNeal, M. Belkhat, Standard tools for hardware-in-the-loop (HIL) modeling and simulation, in: IEEE Electric Ship Technologies Symposium, 2007, pp. 130-137.
- [9] B. Lu, X. Wu, H. Figueroa, A. Monti, A low-cost real-time hardware-in-the-loop testing approach of power electronics controls, *IEEE Transactions on Industrial Electronics* 54 (2) (2007) 919-931.
- [10] J. Wu, Y. Cheng, A. Srivastava, N.N. Schultz, H.L. Ginn, Hardware in the loop test for power system modeling and simulation, in: IEEE Power Systems Conference and Exposition, 2006, pp. 1892-1897.
- [11] SIMSEN Home Page, <http://simsen.epfl.ch>.
- [12] A. Béguin, P. Allenbach, S. Keller, J.J. Simond, S. Brausewetter, J. Koutnik, Hardware-in-the-loop simulation software for regulator tests and optimization, in: IEEE Industry Applications Conference, 2007, pp. 2422-2428.
- [13] C. Nicolet, Y. Vaillant, B. Kawkabani, P. Allenbach, J.J. Simond, F. Avellan, Pumped storage units to stabilize mixed islanded power network: A transient analysis, in: Hydro 2008 Conference, Processing World Hydro Development, Ljubljana, 2008.