An HIL-Based Reconfigurable Platform for Design, Implementation, and Verification of Electrical System Digital Controllers

Shahram Karimi, Philippe Poure, Shahrokh Saadate

A HIL-based reconfigurable platform for design, implementation and verification of electrical systems digital controllers

Shahram Karimi, Philippe POURE and Shahrokh Saadate

Abstract— **This paper presents a Top-Down design flow for design, implementation and verification of digital controllers associated with electrical systems. In the proposed design flow, the functional description of the studied system and the detailed electronic hardware design and validation of the digital controller are performed using a HIL-based reconfigurable platform in a unique design environment. The way of using this design flow and the reconfigurable HIL platform is analyzed through a fault tolerant shunt active power filter application. The experimental results obtained with a laboratory prototype fault tolerant active filter demonstrate the performances and the efficiency of the proposed design flow and HIL-based reconfigurable platform.**

Index Terms— **Hardware-In-the-Loop (HIL), Field Programmable Gate Array (FPGA), digital controller, active filter.**

I. Introduction

The use and the complexity of digital controllers associated with electrical systems are nowadays widely increasing. Therefore, more efforts are devoted to the design, simulation and verification of such controllers. Recent advances in digital technologies and software design tools allow the designers to develop complex and compact high-performances digital controllers. With VLSI (Very Large Scale Integration) devices currently available, such as FPGA (Field Programmable Gate Array) and ASIC (Application Specific Integrated Circuit), fully digital controllers can be realized [1], [2]. Thus, a control system which was previously implemented in an electronic board can be now designed in a single chip [3]. In addition, the use of reconfigurable components, such as FPGA, allows the development and the rapid prototyping without significant hardware modification [4], [5].

The development of digital control techniques for electrical drives and power converters provides the possibility of carrying out flexible control systems that can be easily adapted to different applications with as little modifications as possible or varying only the software. However, a flexible platform, that involves both hardware and software design methodologies and tools, must be used to perform a reliable design and to realize satisfactory "time to market" [2], [4].

The traditional model-based simulation has the disadvantage of being unable to exactly replicate real operational conditions. It does not take into account the limitations of the digital controller, like saturation of values in fixed point format during the intermediate calculations and the finite resolution of registers [6]. Moreover, the use of the designed architecture in the actual electrical system before hardware validation can even cause irreversible damage. Hardware-In-the-Loop (HIL) testing in the development cycle of a control system is an effective solution to reduce the risks of discovering an error in the very last stage of the on-the-field testing and assembling [7], [8].

In real-time HIL platforms the control system is connected to a real-time simulator which simulates the other parts of the actual system. Although a platform based on real time HIL simulation is very useful to test and debug the digital controllers, such platforms require specific Computer Aided Design (CAD) tools and/or software platforms for design, configuration and target implementation. Furthermore, due to required accuracy and precision of accounting the gating signals coming from the digital controller, only the real-time simulators that use the correction algorithms can simulate, with acceptable accuracy, the power electronic systems including converters with high switching frequency [9], [10]. The correction algorithms are mostly interpolation/extrapolation and variable simulation time-step [9].

Recently, FPGA-based real-time digital simulator is proposed for real-time simulation of the complete power electronic and drive systems [11]–[13]. In this real-time simulation approach, the actual system is modelled and written in Very high speed integrated circuits Hardware Description Language (VHDL) and implemented on a FPGA. The simulation runs directly in FPGA and the small time-step (the order of a few nanoseconds) allows a highly detailed and precise accounting of gating signals [10]. Although FPGA-based real-time digital simulator has some advantages such as freedom from reliance on complicated correction algorithms and detailed representation of the device switching characteristics, however a library of the models (in VHDL) for the power elements used in electric drives and power system applications is not yet commercially available. This lack of library can increase the design and development time

This paper presents a new HIL-based reconfigurable platform for design, implementation and verification of digital controllers for electrical systems. A Top-Down design methodology is used to study step by step the digital adaptation, the modelling and the fully digital implementation and hardware verification of the control algorithm. In this approach, the mathematical aspects of engineering systems (functional/behavioral description) and the detailed electronic hardware design and validation are performed in a unique design environment. A development board which comprises a FPGA chip is used to realize the reconfigurable platform. The control algorithm is implemented in the FPGA target that communicates, in closed loop, with the simulated power plant. The applicability of using the proposed HIL-based reconfigurable platform to design digital controllers is analyzed through a fault tolerant shunt active power filter

application case. The performances and the efficiency of the proposed design flow and HIL-based reconfigurable platform are evaluated using a laboratory prototype fault tolerant active filter.

II. Electrical Systems

Conventional electrical systems consist of a complex assemblage of individual components with a variety of different functionalities. They are mostly made up of power elements (plant), controller, sensors and interfaces (see Fig. 1). Source, converter and load are the major power elements. Sensors mostly consist of electrical (voltage or current) or mechanical sensors. The plant and the controller are usually connected in closed loop by means of sensors sending analog signals from the plant to the controller. An interface levels the signals sent from the controller to the power switches (Firing pulse unit, gate drives, …). A computer-based link between the operator and the digital controller allows setting up and reconfigurating the controller.

Multiple such sub-systems may be inter-connected to realize a single complex system. Typical AC motor drive using two converters for grid connection, hybrid electric vehicle power drive, traction motor, DC-DC converters and AC-DC converters are some examples [13].

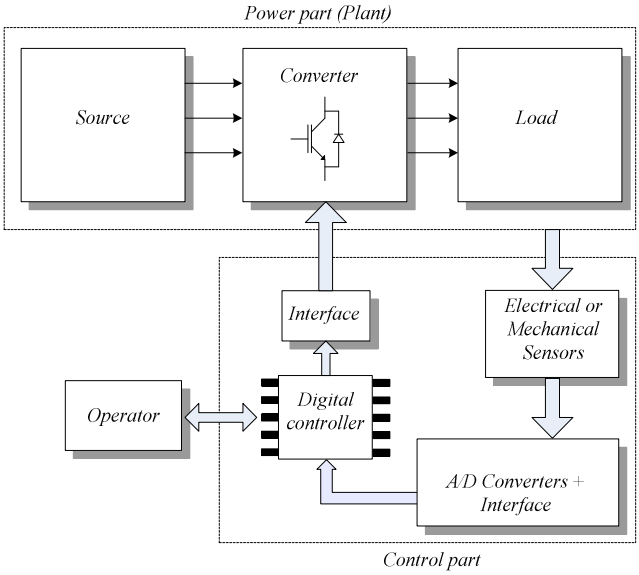


Fig. 1. Structure of a conventional electrical system.

III. Software Platform

Digital Signal Processing (DSP) system design in Programmable Logic Devices (PLDs) such as FPGA requires specific development tools supporting both high-level modeling and hardware description language (HDL), such as VHDL or Verilog. To verify the functionality of the HDL description and the interaction between the plant (analog part) and the digital controller, a software platform is required. There are different alternatives to simulate such mixed (analog and digital) systems.

One way consists in using mixed-signal simulation tools, such as ADVance MS (Mentor Graphics), AMS Designer (Cadence), SaberHDL (Synopsys), SIMPLORER (Ansoft) or SMASH (Dolphin Integration). These CAD tools are mainly used for the functional verification of mixed-signal System-on-a-Chip (SoC). They can simulate any combination of SPICE or SPICE-like models and mixed-signal hardware description languages (MS-HDL), such as VHDL-AMS. VHDL-AMS is a superset of VHDL and full VHDL syntax and semantics is supported. It supports the modelling of discrete-time and continuous systems as well as the modelling at different abstraction levels in pluri-disciplinary energy domains (electrical, thermal, mechanical, etc.).

The co-simulation using Saber-Modelsim or Matlab-Modelsim CAD tools is an other alternative. In this approach, a synchronous interface allows the digital controller designer to concurrently simulate the VHDL model of the controller in the ModelSim HDL simulator with a model of the controlled electrical system in Saber or Matlab environment. The synchronous interface exchanges data between both simulators at fixed interval and performs an analog-digital conversion between the plant model and the VHDL model of the digital controller.

Recently, FPGA manufacturers have designed software packages, such as DSP Builder (Altera) and System Generator (Xilinx), which enable both the simulation and the automatic translation into hardware of a design inside the Matlab-Simulink environment. More recently, such software can support the HIL simulation for effective testing of digital circuits design.

In this paper, we use the DSP Builder software to make our HIL-based reconfiguration platform. This software integrates HDL development tools by combining the algorithm development, simulation and verification capabilities of the Matlab/Simulink system-level design tools with the VHDL design flow including Quartus II (Altera) software. Quartus II software provides a comprehensive design, synthesis, and analysis environment for PLDs applications. DSP Builder allows creating the hardware representation of the required digital signal processing functions using Matlab/Simulink user-friendly algorithm-development environments, for shorter design and implementation cycles. Matlab functions and native Simulink blocks can be combined with DSP Builder library blocks to create FPGA designs which can be simulated under Simulink. For automated design flow, the “Signal Compiler” block allows generating VHDL code and scripts for Quartus II-based synthesis and fitting from within Simulink. Furthermore, the DSP Builder HIL block enables chip programming for HIL verification and validation.

IV. Top-Down Design Flow

The designer can proceed using the Top-Down design flow depicted in Fig. 2, first performing conceptual studies using functional simulation, and then continually refining the digital controller design using mixed simulation and finally testing the implemented control algorithm via a HIL-based reconfigurable

platform. Fig. 2 shows this proposed design flow. Let's now discuss and detail the three stages of this flow.

A. Functional simulation

To define the system specifications and evaluate the functionality of the controller, functional simulation is used. In this stage, the power circuit (plant) and the sensors are modelled using Matlab/SimPower Systems and the interfaces and the controller are modelled using Simulink. At this step of the design flow, the studied system is first modelled in continuous-time mode and then in discrete-time mode.

Non-linear switching topologies such as IGBT inverter model with switching frequency of several kHz need very tiny time-steps to attain acceptable accuracy. Thus, time-step for the plant model must be chosen sufficiently small to ensure accuracy. A time-step equal or smaller than 1µs is usually recommended to simulate the power system including the converters with high switching frequency. Note that using the correction algorithms such as interpolation method, the power converters can be modelled with time-steps of 10% to 20% of the PWM carrier period [9], [13]. The sampling periods for the interfaces and the controller models are chosen depending on digital control system specifications, mostly the A/D conversion time and the computational complexity of the control algorithm. Note that if the A/D conversion time is greater than the computation time, the sampling period can be chosen equal to the A/D conversion time to have higher bandwidth and minimum calculation delay. The computation time can be calculated from the timing report after compilation stage (see section C).

B. Mixed simulation

After having checked and validated the control part in the previous step, it must be modelled with binary synthesizable format (see Fig. 2). In this step, the power plant, the sensors and the interfaces models remain the same as the ones used in the functional simulation, previouly described. However, the Simulink blocks, used for the controller modelling, are removed and replaced by DSP Builder blocks, in the same Simulink environment. In this case, Simulink signals must be converted to fixed-point binary signals. A fixed-point binary format, noted [s, m_i, m_d], is defined with s the sign bit, m_i the bit number of the integer part, and m_d the bit number of the decimal part. It should be noted that some complex functions and Simulink blocks do not have corresponding blocks in the DSP Builder library. As a result, these functions or blocks should be described by the designer himself from the basic blocks of the DSP Builder library.

Since the ultimate goal is to implement the control algorithm into a FPGA target, the number of bits of the FPGA input signals depends on the resolution of the A/D converters that will be used. Note that the number of bits does not depend on the interface modules that are used to reduce the value of analog signals in the range [-1, 1]. However, the total number of bit used for intermediate calculations should be chosen lower than 51 because of the restriction imposed by the DSP Builder software. It should be noted that the small bit-widths increase the calculation errors and can lead to the saturation whereas the large bit-widths increase execution time and use high FPGA chip resources. So, the best trade-off between the fixed-point format size of each control variable and the respect of the control specifications is needed. The "Bus Probe" block, available in DSP Builder library, can be placed at any node of the model to know the maximum number of bits (integer part) required during the mixed simulation.

At this step of the digital controller design, the time-step and the sampling period used for the models are the same as the ones previously used in the functional discrete-time mode simulation.

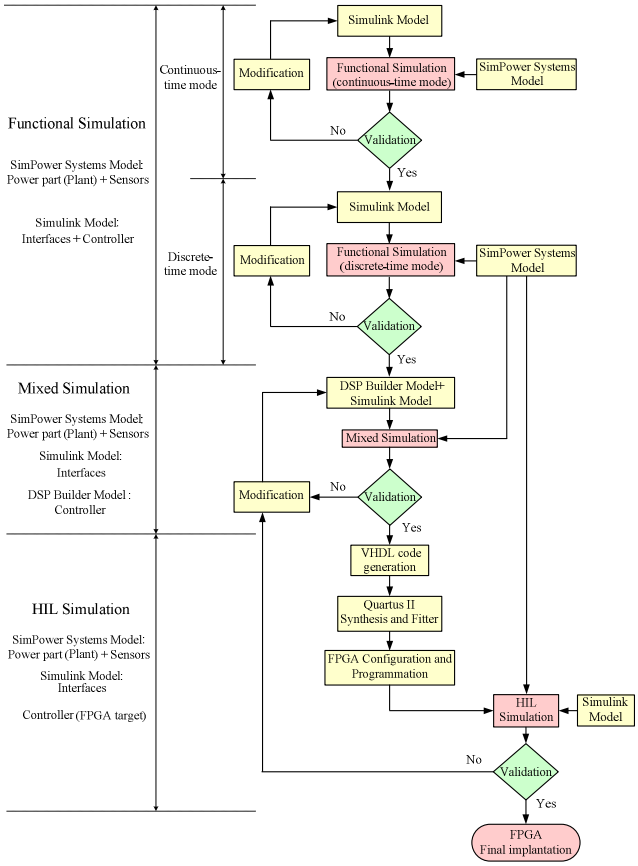


Fig. 2. Top-Down design flow.

C. HIL simulation

The final step of the design flow (see Fig. 2) is dedicated to the hardware implementation, the verification and the validation of the digital control algorithm in a HIL-based reconfigurable platform. To achieve these objectives, the digital controller, modelled in the previous step using DSP Builder blocks, is firstly converted into a synthesizable VHDL model at Register Transfer Level (RTL). The VHDL files can be generated "automatically" using the "Signal Compiler" block available in DSP Builder library. After the generation of the RTL VHDL files, the synthesis is performed using a synthesis tool, such as Quartus II (Altera), Synplify

(Synplicity), Precision RTL or LeonardoSpectrum (Mentor). However, the Quartus II software is mandatory to perform the physical synthesis for Altera's FPGA family. The "Signal Compiler" block generates the Tcl scripts based on the selected synthesis tool. The Quartus II optimization option, that is available on the "Signal Compiler" settings page, allows the designer to specify whether the compiler should optimize, during the mapping and fitting stages, speed or area, or perform a "balanced" optimization. After synthesis, the fitting stage is performed. The Quartus II Fitter places and routes the design. It matches the logic and timing requirements of the desgin with the available resources of a target FPGA device. It assigns each logic function to the best logic cell location for routing and timing, and selects appropriate interconnection paths and pin assignments.

At this step of the procedure, the so-called HIL block, available in the DSP Builder library, is added to the design environment. Then, this HIL block replaces all the DSP Builder blocks which must be removed from the design environment. After the connection of the Simulink input and output signals to the HIL block, the Quartus II project, created through the compilation step, is imported into the HIL block and then the various parameters, such as the input and output pin caracteristics, are specified. After this stage, the HIL block is compiled to create a programming object file that will be used for HIL simulation. Finally the development board that contains the target FPGA can be programmed and used in the HIL simulation.

The development board is linked to a PC via a Joint Test Action Group (JTAG) connexion. This interface performs communication between the digital control algorithm (implemented into the target FPGA) and the plant (emulated by the PC with Matlab/SimPower Systems). In the proposed HIL-based reconfigurable platform, at each time-step, the plant model is simulated using Matlab/SimPower Systems and then the output signals are exported to the target FPGA placed in the development board. When the FPGA receives signals from Simulink, it executes the implemented program for one sample interval. The FPGA returns control signals, computed during this step, to Simulink. At this point, one sample cycle is performed.

If the results obtained from the HIL simulation are not correct or the goals are not met, the mixed simulation and the HIL simulation steps must be performed again with the modified digital controller model, according to the Top-Down design flow shown in Fig. 2.

V. Application Example: Fault Tolerant Shunt Active Power Filter

This application example illustrates the design procedure, the hardware implementation and the verification using the proposed HIL-based reconfigurable platform for a fault tolerant shunt active power filter. More, the results obtained from the HIL simulation are compared with experimental results to validate the efficiency of the HIL platform. In this example, a digital algorithm for power switch fault detection is implemented into a FPGA to detect and compensate a faulty switch. The fault tolerant topology is achieved by adding one redundant leg to the classical voltage source inverter used in shunt active power filter.

A. System description and fault tolerant topology

Fig. 3 shows the fault tolerant topology, composed of four legs. The fourth leg formed by the switches S_7 and S_8 is a redundant leg. This leg will replace the faulty one of the three other legs. When a fault occurres in one of the semi-conductor switches (S_1-S_6), the fault detection scheme detects the fault occurrence and isolates the faulty leg. Then, the suited bidirectional switch is triggered to connect the faulty phase to the midpoint of the redundant leg.

The output currents of the active filter are controlled to provide the harmonic currents generated by the non-linear load to ensure filtering. The non-linear load is a three-phase diode rectifier feeding (R, L) load. The specifications of the power system are given in the Table 1.

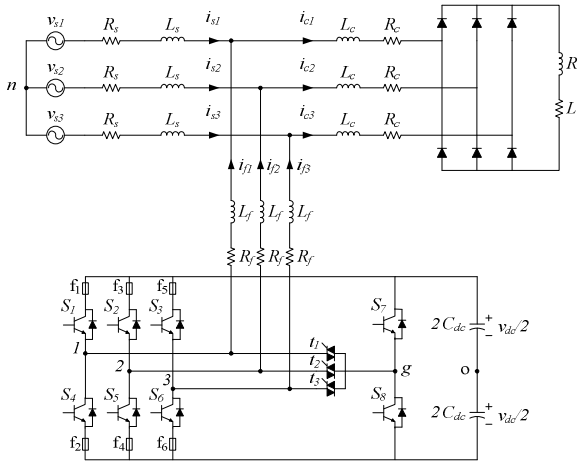


Fig. 3. Fault tolerant shunt active power filter.

TABLE 1
POWER SYSTEM SPECIFICATION

Mains voltage	400 V
Mains frequency	50 Hz
Mains inductance (L_s)	0.1 mH
Mains resistance (R_s)	0.2 mΩ
Load ac inductor (L_c)	0.8 mH
Load ac resistance (R_c)	0.27 mΩ
Load inductor (L)	40 mH
Load resistance (R)	48.6 Ω
Filter inductor (L_f)	3 mH
Filter resistance (R_f)	5 mΩ
DC capacitor (C_{dc})	1100 µF
DC voltage (V_{dc})	700 V

B. Control strategy

The control strategy used to compensate the harmonic currents generated by the non-linear load consists in a reference currents generator and a current controller. The reference currents generator determines the reference currents which must be generated by the active filter. The reference

currents are identified using a modified version of the instantaneous active and reactive power method, developed in our laboratory and based on the work presented by H. Akagi [14]. This new strategy is based on two digital voltage and current selective band-pass filters [15]. Thanks to these selective band-pass filters, the reference currents are correctly generated, even in unbalanced voltage source conditions, by directly extracting the fundamental components of the load currents without Phase Locked Loop (PLL) [15]. The current controller is used to control the switches (S_1-S_6) of the voltage source inverter. The controller we have used is a modulated hysteresis current controller which performs a fixed switching frequency [16].

C. Fault detection scheme

The power switch fault detection is based on the comparison between the measured and estimated pole voltages, v_{ko} (k = {1, 2, 3}), respectively noted v_{kom} and v_{koes} [17]. The estimated pole voltages are expressed by:

$$v_{koes} = (2\delta_k - 1) \times \frac{v_{dc}}{2} \quad (1)$$

Where δ_k = {0, 1} is the switching pattern of the top switch of the leg number k and v_{dc} is the DC bus voltage. The fault occurrence can be detected by analysing the difference between the measured and estimated pole voltages. This voltage error is given by:

$$\varepsilon_{ko} = v_{kom} - v_{koes} \quad (2)$$

With ideal switches consideration, the measured and estimated pole voltages are equal in normal operation and thus the voltage error ε_{ko} is zero. However, in real case, because of the dead time generated by the switches' drivers and the delay time introduced by the A/D converters and the voltage sensors used for the pole voltages measurement, the voltage error ε_{ko} is not null and constituted of peaks during switching times. To avoid false fault detections due to power semi-conductors switching, we have proposed to use simultaneously a "time criterion" and a "voltage criterion" [17].

The time error signal is computed for each phase by first taking the absolute value of the voltage error signal (see Fig. 4). Then, the output is applied to a comparator with a threshold value h. The output of the first comparator, c_k, is equal to zero if $|\varepsilon_{ko}| \leq h$ and equal to one if $|\varepsilon_{ko}| > h$. Thus, the output of this comparator is a repetitive square waveform because of power semi-conductors switching. The up-counter is enabled and starts to count when the output of the first comparator becomes equal to one. The output of the up-counter, n_k, is equal to the number of clock pulses when the output of the first comparator is one, if counting is initialised to zero after each square waveform. By considering the up-counter clock frequency, the output of this counter is directly linked to the time during which v_{kom} and v_{koes} are different. Consequently, the fault occurrence is detected using simultaneously a "time criterion" and a "voltage criterion". To do this, the up-counter output is applied to a second comparator with a threshold value N_t. This threshold value should be chosen greater than the maximum delay time introduced during switching. By this way, we avoid false fault detection due to semi-conductors switching while the fault condition can be detected. The resulting signal f_k from the fault detection scheme is used to isolate the faulty leg and trigger the suited bidirectional switch t_k.

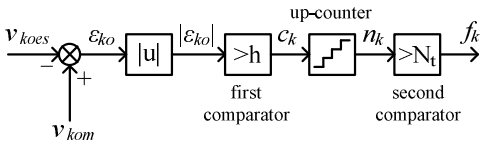


Fig. 4. Block diagram of the fault detection scheme.

D. Functional simulation

At this first stage of the design flow, the fault tolerant topology (plant) shown in Fig. 3 and the sensors are modelled using Matlab/SimPower Systems. The interfaces, the control strategy and the fault detection scheme are modelled using Simulink. The studied system is first modelled in continuous-time mode and in discrete-time mode. In the functional simulation, the up-counter, shown in Fig. 4, is modelled by an integrator. The values of the thresholds N_t and h are chosen respectively equal to 25 (corresponding to 5µs) and 20V.

Fig. 5 presents functional simulation results (continuous-time mode) for an open-circuit fault. The fault is introduced in the switch S_3 at t = 135.5 ms. The mains current i_{s3}, the filter current i_{f3}, the switching pattern δ_3 and the fault detection signal f_3 are illustrated in Fig. 5. This figure shows that the failure has no effect on the mains and filter current waveforms. Moreover, the fault tolerant shunt active power filter preserved the main performance features after fault compensation, i.e. a THD value equal to 2.2%. Fig. 6 shows zoomed functional simulation results for duration of 500 µs. One can see that δ_3 is equal to zero when the fault is introduced at t = 135.5 ms. So, the leg number 3 operates correctly and the fault must effectively not be detected until the switching pattern δ_3 becomes one. Therefore the fault detection is achieved 5 µs after the time that δ_3 becomes one.

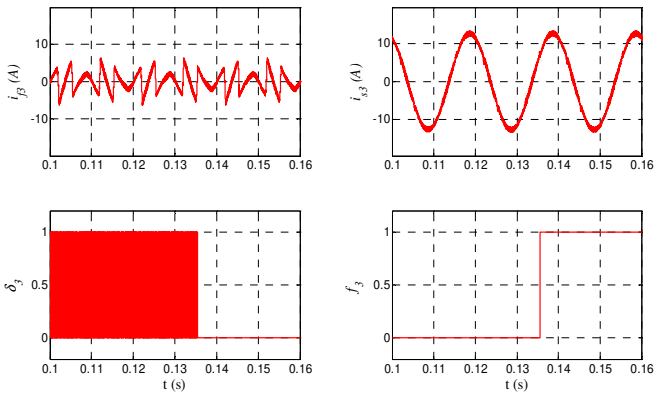


Fig. 5. Functional simulation results (continuous-time mode) when an open-circuit fault is introduced in the switch S_3 at t = 135.5 ms.

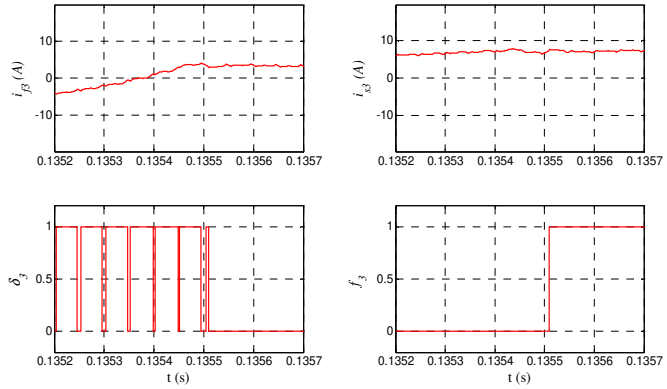


Fig. 6. Zoomed functional simulation results (continuous-time mode).

After the validation of the studied system models by functional simulation in continuous-time mode, the plant and the sensors are modelled with Matlab/SimPower Systems in discrete-time mode with a time-step equal to 0.2 µs. The interfaces, the control strategy and the fault detection scheme are modeled with Matlab/Simulink. The sampling period for the modulated hysteresis current controller and the fault detection scheme is chosen equal to 0.2 µs. For the reference currents generator, the sampling period is chosen equal to 30 µs. This value corresponds to the minimum sampling period that we can choose when the reference currents generator is implemented in a dSPACE system associated with a DS1104 development board (we will use this board to generate the reference currents in the fault tolerant active filter prototype realized in our laboratory).

Since the results obtained for the functional simulation (discrete-time mode), for the mixed simulation and for HIL simulation are identical, we will only present HIL simulation results.

E. Mixed simulation

Since, in this paper, only the fault detection scheme will be implemented into the FPGA target, only this part is modeled with DSP Builder and the control strategy and the plant models remain the same as the ones used in the functional simulation (discrete-time mode). Fig.7 illustrates this mixed simulation.

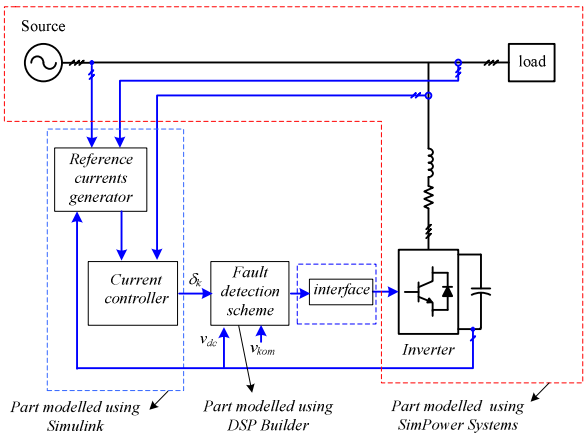


Fig. 7. Principle of the mixed simulation for the fault tolerant active filter.

The input signals of the fault detection scheme (see Fig. 4 and equations 1 and 2) are the following: the DC bus voltage v_{dc}, the measured pole voltages v_{1om}, v_{2om} and v_{3om} and the switching patterns δ_1, δ_2 and δ_3. Considering the A/D converters we used with 12-bit resolution, the format of the input voltage signals is chosen equal to [1, 0, 11], i.e. 1 bit for the sign bit and 11 bit for the decimal part. Note that interface modules reduce the value of analog signals in the range [-1, +1]. Because the switching patterns δ_k are logic signals and take only two values, 0 or 1, a single bit is used to encode these signals. The output signals from the fault detection scheme are the 8 switching patterns δ_1-δ_8, and the 3 fault detection signals f_1, f_2 and f_3. These output signals are also logic signals, so the logical format is used.

We used a Moore Finite State Machine (FSM) to model the fault detection scheme. Fig. 8 shows the state diagram and the control unit of this fault detection scheme. The up-counter is clocked by a clock pulse with a period of 0.2 µs.

F. HIL simulation

We now validate the implementation of the digital fault detection scheme using the HIL simulation. The used reconfigurable platform is based on a DSP S80 development board, which comprises the Altera Stratix EP1S80B956C6 FPGA chip. This FPGA contains 79,040 programmable logic elements, 679 user I/O pins, 12 reconfigurable fast PLLs and 7.4 Mb RAM. Also, this development board has an on-board 80-MHz oscillator.

To implement the fault detection algorithm into the FPGA target, the following steps are performed:

1 - Generation of the synthesizable VHDL code;

2 - Logical synthesis;

3 - Fitting.

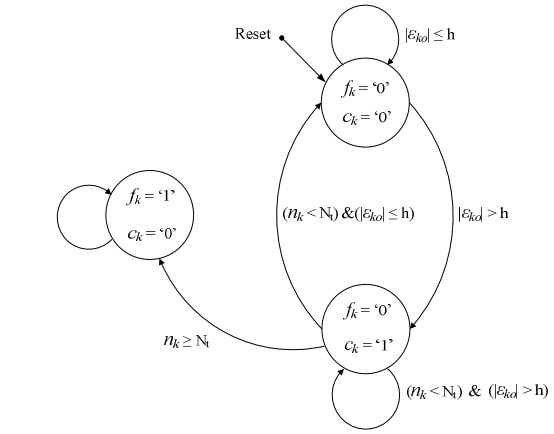
At this step, all DSP Builder blocks are replaced in the Matlab/Simulink environment by a single "HIL block" corresponding to the digital fault detection scheme implemented into the FPGA. The connection between the development board and the computer is done by using a ByteBlasterMV cable also used for FPGA programming.

In this case, at each time step (0.2 µs), the FPGA target receives the signals, v_{dc}, v_{1om}, v_{2om}, v_{3om}, δ_1, δ_2 and δ_3. Then it runs the implemented digital fault detection scheme and transmits back to Simulink the switching patterns δ_1-δ_8, and the fault detection signals f_1, f_2 and f_3.

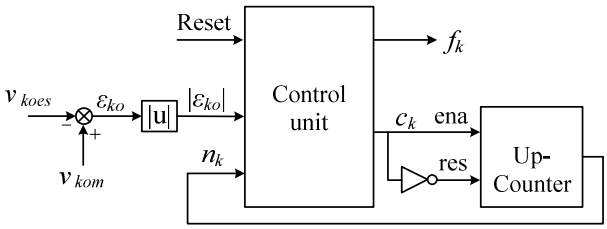
To compare the results obtained using the HIL-based reconfigurable platform with the results obtained by functional simulation in continuous-time mode (presented in figures 5 and 6), we consider the same fault, i.e. an open-circuit fault introduced in the switch S_3 at t = 135.5 ms. Fig.9 presents the HIL simulation results. We note again that the mains and filter currents waveforms are not affected by the fault. However, one can see that the small notches appear on the mains current waveform. The origin of these notches is the delay (30µs)

between the harmonic currents generated by the voltage source inverter and the harmonic currents absorbed by the load.

In this case, the THD is equal to 2.7% while it was equal to 2.2% with the functional simulation in continuous mode (without delay). These THD values show that the compensation delay degrades the performance of the shunt active power filter.



(a)



(b)

Fig. 8. (a) State diagram of the fault detection scheme. (b) Control unit of the fault detection scheme.

Fig.10 presents zoomed HIL simulation results for duration of 500 µs. This figure shows that δ_3 is equal to one when the semi-conductor S_3 becomes faulty. As a result, the fault is detected 5µs after its occurrence. This result confirms the validity of the modeling and the digital implementation of the fault detection scheme.

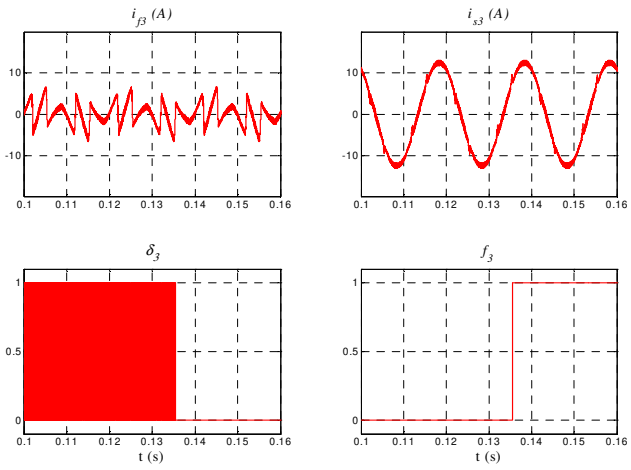


Fig. 9. HIL simulation results when an open-circuit fault is introduced in the switch S_3 at t = 135.5 ms.

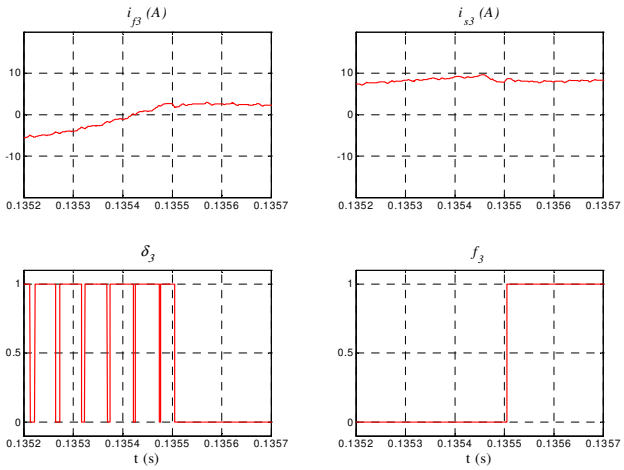


Fig. 10. Zoomed HIL simulation results.

VI. Experimental Results

After having validated the digital implementation of the fault detection algorithm using the HIL-based reconfigurable platform, the digital algorithm can now be used in the FPGA-based controller. Fig. 11 presents the principle of our fault tolerant shunt active power filter set up.

The reference currents are generated using a Dspace DS1104 development board with a sampling period equal to 30µs. The switching patterns are generated using an analogue card which realizes the modulated hysteresis current controller. To connect the DSP S80 development board to the experimental system, an interface card has been designed. The A/D converters used in this interface card are ADS7810U components which convert the analog signals every 1.25 µs.

For our laboratory experimental system, the maximum delay time between the measured and estimated pole voltages during semi-conductors switching is about 14 µs. To avoid false fault detection due to semi-conductors switching, the threshold value of the second comparator, N_t, shown in Fig. 4, was fixed to 150 (corresponding to 30µs).

Figures 12, 13 and 14 present the experimental results. The open switch fault was generated in the top switch driver of the leg number 3 by cutting its connection with the current controller output. Fig. 12 shows, for duration of 0.04 s, the mains phase current i_{s3}, the filter phase current i_{f3}, the fault detection signal f_3 and the fault signal, corresponding to the time that the fault is generated. Fig. 13 shows zoomed results for duration of 400 µs. Fig. 14 shows the switching pattern δ_3, the fault detection signal f_3 and the fault signal for duration of 200 µs. To generate these figures, two 4 channels oscilloscopes, simultaneously triggered by the fault signal, were used. As one can see, the fault was introduced when the switching pattern δ_3 is equal to one, thus the fault detection was achieved 30 µs after the time that the fault is generated. One can notice that the fault effect does not appear in the mains and filter current waveforms.

The results obtained using the laboratory experimental system confirms the performances and the efficiency of the proposed HIL-based reconfigurable platform for design, digital implementation and verification of digital controllers for electrical systems.

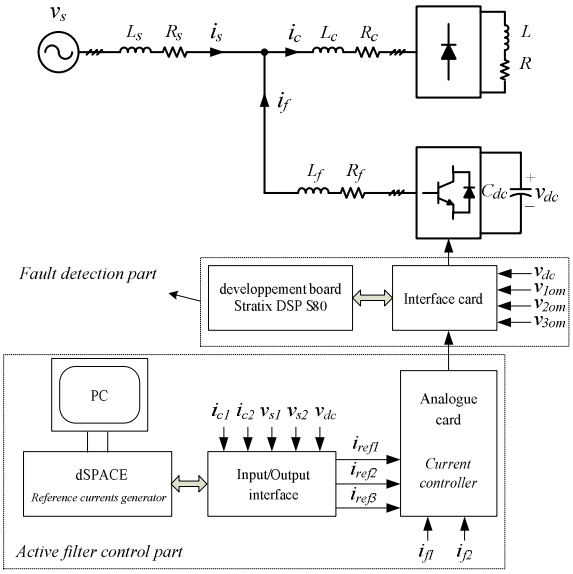


Fig. 11. Principle of the laboratory prototype fault tolerant active filter.

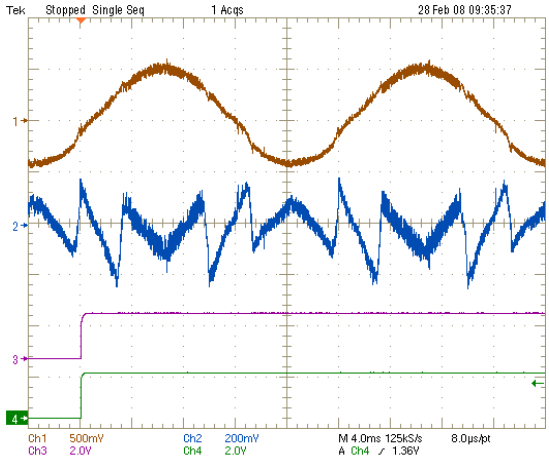


Fig. 12. Experimental results when an open-circuit fault is introduced in the switch S_3. From top to bottom: mains phase current i_{s3} (15A/div), filter phase current i_{f3} (6A/div), fault detection signal f_3 (4V/div) and fault signal (4V/div). Time scale: 4ms/div.

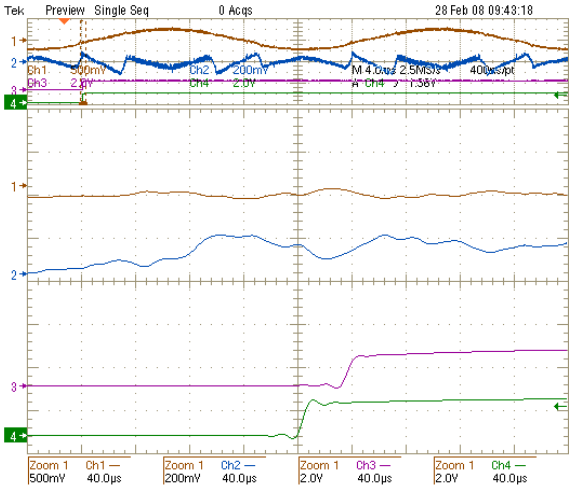


Fig. 13. Zoomed experimental results. Time scale: 40µs/div.

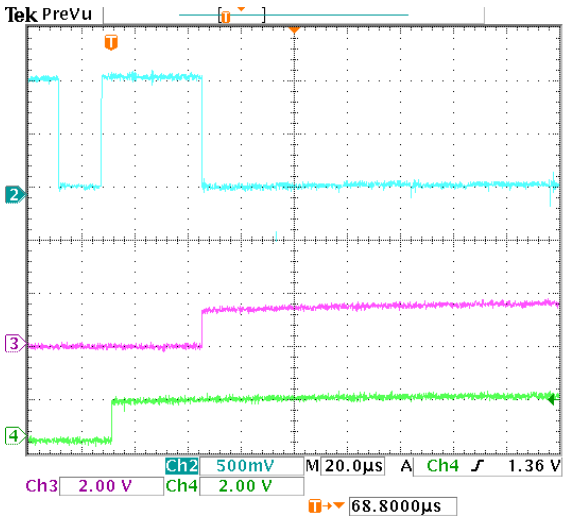


Fig. 14. Detailed study of the fault detection.
From top to bottom: Switching pattern δ_3 before and after the fault occurrence (7V/div), fault detection signal f_3 (4V/div) and fault signal (4V/div). Time scale: 20µs/div.

VII. Conclusion

We have presented a Top-Down design flow for design, implementation and verification of digital controllers for electrical systems. We either used a unique environment for modelling, simulation and verification of the digital controller. In the proposed design flow, the digital algorithm validation is performed using a HIL-based reconfigurable platform that eliminates the risk of damaging the actual plant. The proposed Top-Down design flow allows fast design and short time to market.

A fault tolerant shunt active power filter has been used to illustrate and validate the proposed design flow. The results obtained using this laboratory experimental system confirm the performances and the efficiency of the proposed design flow and HIL-based reconfigurable platform.

REFERENCES

[1] M.-W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja, N. Patin, "FPGA-based current controllers for AC machine drives—a review", *IEEE Transactions on Industrial Electronics,* vol. 54, no. 4, pp. 1907-1925, Aug. 2007.

[2] J. J. Rodriguez-Andina, M. J. Moure, M. D. Valdes, "Features, design tools, and application domains of FPGAs", *IEEE Transactions on Industrial Electronics,* vol. 54, no. 4, pp. 1810-1823, Aug. 2007.

[3] A. Alvarez, J. Doval-Gandoy, F. Freijedo, A. Nogueiras, A. Lago, C. Penalver, C. M., "Comparison of the FPGA Implementation of Two Multilevel Space Vector PWM Algorithms", *IEEE Transactions on Industrial Electronics*, April 2008, Volume: 55, Issue: 4, pp. 1537-1547.

[4] E. Monmasson, M. N. Cirstea, "FPGA design methodology for industrial control systems—a review", *IEEE Transactions on Industrial Electronics,* vol. 54, no. 4, pp. 1824-1842, Aug. 2007.

[5] X. Lin-Shi, F. Morel, A. Llor, B. Allard, J.M. Retif, "Implementation of Hybrid Control for Motor Drives", *IEEE Transactions on Industrial Electronics*, Aug. 2007, Volume: 54, Issue: 4, pp. 1946-1952.

[6] S. Vamsidhar, B.G. Fernandes, "Hardware-in-the-loop simulation based design and experimental evaluation of DTC strategies", *IEEE Power Electronics Specialists Conference*, vol. 5, pp. 3615-3621, June 2004.

[7] 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,* vol. 54, no. 2, pp. 919-931, Apr. 2007.

[8] A. Bouscayrol, X. Guillaud, P. Delarue, B. Lemaire-Semail, "Energetic Macroscopic Representation and Inversion-Based Control Illustrated on a Wind Energy Conversion Systems Using HIL simulation", *IEEE Transactions on Industrial Electronics*, Accepted for future publication.

[9] M. Harakawa, H. Yamasaki, T. Nagano, S. Abourida, C. Dufour and J. Bélanger, "Real-time simulation of a complete PMSM drive at 10 µs time step", *International Conference on Power Electronics*, 2005.

[10] G. Parma, V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives", *IEEE Transactions on Power Delivery,* vol. 22, no. 2, Apr. 2007.

[11] Y. Shi, A. Monti, "FPGA-based fast real time simulation of power systems", *IEEE Power and Energy Society General Meeting*, pp. 1-5, July 2008.

[12] V. Dinavahi, M.R. Iravani, R. Bonert, "Design of a real-time digital simulator for a D-STATCOM system", *IEEE Transactions on Industrial Electronics,* vol. 51, no. 5, pp. 1001-1008, Oct. 2004.

[13] S. Abourida, C. Dufour, J. Bélanger, "Real time and hardware in the loop simulation of electric drives and power electronics: process, problems and solutions", *International Conference on Power Electronics*, 2005.

[14] H. Akagi, Y. Kanazawa, A. Nabae, "Generalized theory of the instantaneous reactive power in three-phase circuits", *International power electronics conference*, pp. 1375-1386, Japan, 1983.

[15] S. Karimi, P. Poure, S. Saadate, "High performances reference current generation for shunt active filter under distorted and unbalanced conditions", *IEEE Power Electronics Specialists Conference,* pp. 195-201, June 2008.

[16] M.A. Shamsi-Nejad, S. Pierfederici, J. P. Martin, F. Meibody-Tabar, "Study of an hybrid current controller suitable for DC–DC or DC–AC applications", *IEEE Transactions on Power Electronics,* vol. 22, no. 6, pp. 2176-2186, Nov. 2007

[17] S. Karimi, A. Gaillard, P. Poure, S. Saadate, "FPGA-based real time power converter failure diagnosis for wind energy conversion systems", *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4299-4308, December 2008.