Implementation of an FPGA-based wind turbine HIL model

Laszlo Schaffer
Department of Technical Informatics
University of Szeged
Szeged, Hungary
schaffer.laszlo@stud.u-szeged.hu

Zoltan Kincses
Department of Technical Informatics
University of Szeged
Szeged, Hungary
kincsesz@inf.u-szeged.hu

Abstract*—Wind energy has become one of the most economical types of renewable energy technology, which provides a secure and sustainable energy supply. Nowadays the use of wind energy is increasing, but many countries still have considerable wind resources untapped. Building a real physical wind turbine is much more expensive and complex than modelling it with software or hardware (Hardware-In-The-Loop (HIL)) system. Therefore it is necessary to implement an efficient wind turbine model to easily study the operation of the system or analyze the effect of different parameter changes. In this paper a three-bladed horizontal variable pitch wind turbine model is implemented in a Digilent Nexys-4 board, which contains a Xilinx Artix-7 Field-Programmable Gate Array (FPGA) as a HIL device using Xilinx System Generator. In this model the wind speed, the pitch angle of the blades of the wind turbine, the radius of the blade, and the density of the material in which the blade is spinningare adjustable parameters. These parameters can be set-up using a PC via serial communication with a LabVIEW based Graphical User Interface (GUI). This FPGA-based HIL system with the PC software can be used not only for experimental but even for education purposes. Students can observe the impact of the input parameter changes, or implement some control equipment to realize an efficient wind turbine system.

Keywords— wind turbine, FPGA, MATLAB/Simulink, Xilinx System Generator, HIL

I. Introduction

Renewable energy systems are becoming more and more important as non-renewable energy sources run out. The most widely used renewable energy systems are wind turbine, hydropower, hydroelectric, photovoltaics, concentrated solar power, biomass, biofuel and geothermal power. According to [1] about 337 GW wind power generation capacity has been set up in the worldin June 2014, which indicates a growing trend comparing to the previous years. Nowadays wind energy is used in more than 100 countries as an electrical power source. Due to this, wind energy can play the most significant role to reduce the effect of global warming. Wind energy became a mainstream electric energy resource.

Usually, a wind turbine consists of a rotor with blades, a shaft and gearbox, and a generator. Wind turbines convert the kinetic energy of the wind into electrical energy and wind farms generate electricity for public service.

Reviewing the literature of modelling wind turbine systems, many various wind turbine, shaft and gearbox, and generator models have been analyzed. A fixed-pitch angle wind turbine simulator was presented in [2], which was built in MATLAB/Simulink. The goal of the simulator was to verify the effectiveness of the fixed-pitch angle wind turbine model with an induction generator when the applied rotational speed was higher than the nominal. A fix-speed wind turbine model was introduced in [3], including the model of the aerodynamic, mechanical and electrical components of the turbine. Using this model a wind farm was developed in Power System Computer Aided Design/Electromagnetic Transients including Direct Current (PSCAD/EMTDC) to study the operation and power grid integration issues of wind turbines. In [4] a comparative study of various methods for mathematically modelling wind turbines was presented. In this work the wind turbine was modelled based on the fundamental equations of wind power and a presumed shape of the power curve. The first model does not correctly substitute the physical wind turbine, and neither the second model can reach the desired accuracy for lower annual average wind speeds. A review of commonly used Variable Speed Wind Turbine (VSWT) power curve equations were presented in [5] from the data of 200 commercial VSWTs ranging from 225 to 7500 kW. In [6] a model used for representing all types of VSWTs in power system dynamics simulations was presented. The Power System Simulator for Engineering (PSS/E) software package was used to simulate the model and compare the results with real measurements. Results showed that the simulation and the measurements were correlating.

In the literature of Field-Programmable Gate Array (FPGA) based Hardware-In-the-Loop (HIL) systems numerous applications have been made, but none of them dealt with wind turbine model implementation. Examples for FPGA based HIL applications implemented in LabVIEW were presented in [7] where HIL simulation and model validation of thermocouples,linear variable differential transformers and resolvers were made. Results showed that high-performance prototypes and test systems can be developed easily, without prior experience in FPGA hardware design. In [8] a dynamic but not HIL system model of Doubly Fed Induction Generator (DFIG) wind turbine system was implemented on an FPGA. The FPGA-based and the MATLAB/Simulink simulation results were compared to each other. The results showed that significant simulation speed-up can be reached by using FPGA circuits.

Based on the literature research it was reasonable to implement a dynamically adjustable FPGA-based HIL device for wind turbine modelling. Due to the great flexibility, high computational capacity and boundless reconfiguration possibilities of FPGAs, FPGA-based HIL devices can be used to easily implement various types of wind turbine systems.

Furthermore, the HIL approach is well applicable in education of wind turbine based systems, because students can analyze the operation of the wind turbine, and study the effect of different parameter changes, without the risk of damaging or endangering the real physical system.

II. Field-Programmable Gate Arrays

FPGAs are programmable semiconductor devices which are built up from a matrix of programmable logic elements connected through programmable interconnections,and can be used to implement almost any kind of digital circuits or systems. The basic structure of a Xilinx FPGA is shown in Fig. 1. It consists of Configurable Logic Blocks (CLBs), a Programmable Interconnection (PI) network, configurable I/O Blocks (IOBs), Block RAM memories (BRAMs), Digital Clock Management blocks (DCM), and Digital Signal Processor (DSP) blocks. The CLBs are the main digital processing blocks configured to perform combinational as well as sequential operations. Look-Up Tables (LUTs) are employed for combinational operations, and D-Flip-Flops are used for sequential operations. The DSP blocks allow complex arithmetic operations, while the BRAM resources can be used for internal data storage to increase the processing speed. Current FPGA devices include high-speed data transceiver blocks, and numerous data communication protocols are supported like USB, Ethernet, CAN, PCI, SPI, I2C, UART, etc. FPGA architectures can contain embedded soft or hard core microprocessors (MicroBlaze, ARM processor cores) and additional peripherals. [9**]**

A. *Artix-7 FPGA architecture*

The wind turbine model was implemented on an Artix-7 XC7A100T FPGA. The general and dedicated resources of this device are shown in Table 1. One CLB of the Artix-7 FPGA contains two Slices. A Slice contains four 6-input LUTs, eight registers, multiplexers and carry logics as general logic resources. A single LUT can be configured as a 6-input LUT, a 32-bit shift register or a 64-bit distributed RAM. The Artix-7 architecture also contains dedicated resources like 36kbit dual port BRAM memories, and 25x18bit DSP Slices.

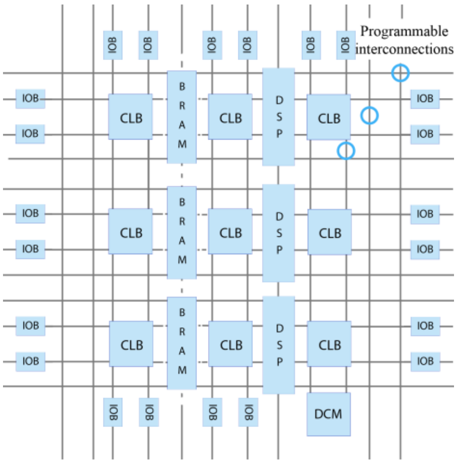

Fig. 1. Basic FPGA architectures

TABLE II. Artix-7 XC7A100T general and dedicated resources

Resource Type	***Name***	***Quantity***
Generalresources	Slices	*15850*
	6-input LUT	63400
	Flip-Flops	126800
Dedicated resources	36Kbit BRAM	135
	DSP48E1 Slice	240

The BRAMs can be configured as two independent 18kbit SRAM, or a FIFO memory due to the dedicated logic of the BRAM. The DSP can be used for 25x18bit signed multiplication, 48bit addition/subtraction or logic operations. [10]

III. Digilent Nexys-4 Board and Accessories

A Digilent Nexys-4 board was used to set-up the FPGA-based wind turbine HIL system. This board contains all the circuitry for configuring the Artix-7 XC7A100T FPGA, and also the required peripherals for implementing the HIL device. The PMOD interfaces can be used to connect low frequency, low pin-count peripheral modules to the Nexys-4 board. Several types of PMOD compatible modules are available from simple push buttons to more complex modules like accelerometers, gyroscopes, thermometers, ADC and DAC converters [11].

The adjustable parameters of the wind turbine model can be configured by using the USB-UART Bridge of the Nexys-4 board. Since the input of the HIL device should be an analogue voltage proportional to the wind speed and the output should be an analogue voltage proportional to the torque, AD and DA conversion is required. Using this configuration real miniature wind sensor and generator can be connected to the HIL.

The analog to digital conversion was performed by the PMOD AD1 and the digital to analog conversion was performed by the PMOD DA2 module. The PMOD AD1 module uses a 6-pin connector header and it is built up from 2 Analog Devices AD747612bit ADCs [12]. It canconvert signals at a maximum sampling rate of 1MS/s. The PMOD DA2 also uses a 6 pin connector header and the digital-analog conversion was performed by using 2 National Semiconductor DAC121S101 12bit DACs [13]. Each module uses SPI interface for the communication with the FPGA, and converts the 0-3.3V voltage into digital signals and vice-versa [11].

IV. Wind Turbine Model

Two different and still similar models have been implemented in MATLAB/Simulink and on the Artix-7 FPGA. The first model is based on the normalized wind turbine model of the SimPowerSystems/Specialized Technology/Renewable Energy Systems library of MATLAB [14], which has a generator connected to the turbine. The output of the system is the torque applied to the generator shaft in per unit of the generator ratings. The second model is based on a standalone wind turbine model [15], without the generator. This is a non-normalized model, so the output is the torque on the shaft of the turbine. The mathematical description of the models is similar, so they can be considered as one model, in different aspects.

A. *Mathematical Model*

The frequently used mathematical models are based on the kinetic energy of the wind, because in that way a dynamically configurable system is achievable. The kinetic energy contained in a mass *m* (kg) of the moving airE_k can be calculated using (1).

$$E_K = \frac{1}{2} m v^2 , \tag{1}$$

Where *v* denotes the wind speed (m/s). From (1) the total mechanical power P_T (*W*) in the mass of moving air can be obtained as follows (2):

$$P_T = \frac{1}{2} \rho A v^3 , \tag{2}$$

whereρ represents air density (kg/m^3) and *A* is the area swept by the blades (m^2). The amount of energy extracted from the wind depends on the aerodynamics of the rotor blade. The ratio between the total mechanical power and the extracted mechanical power is defined by the power coefficient C_P. From this the extracted mechanical power *P* can be obtained as (3).

$$P = \frac{1}{2} \rho A v^3 C_P , \tag{3}$$

The power coefficient is given by the manufacturers and it is a function of the rotational speed of the turbine, wind speed and pitch angle of the blades. Generally the rotational speed and the wind speed are combined into the Tip Speed Ratio (TSR) represented as λ(4).

$$\lambda = \frac{R\omega}{v} , \tag{4}$$

Where *R* represents the length of the rotor blades (m) and ω denotes the mechanical angular velocity (rad/s), what can be calculated as (5).

$$\omega = \frac{2\pi n}{60} , \tag{5}$$

Where *n* is the rotational speed (r/min). *A*can be substituted by $R^2\pi$. Therefore the equation for the torque *T* can be obtained as (6) by dividing (3) with (5).

$$T = \frac{\frac{1}{2} \rho \pi R^2 v^3 C_P}{\omega} , \tag{6}$$

Another approach for the torque is (7).

$$T = \frac{1}{2} \rho \pi R^3 v^2 C_q , \tag{7}$$

Where $C_q$represents the torque coefficient defined as (8).

$$C_q = \frac{C_P}{\lambda} , \tag{8}$$

The power coefficient is the function of the blade pitch angle represented as β and the TSR, and can be obtained as follows (9):

$$C_P(\lambda, \beta) = c_1 (c_2 - c_3 \beta - c_4) e^{-c_5} + c_6 \lambda , \tag{9}$$

The general parameters are [16]:

- $c_1 = 0.5176$
- $c_2 = \frac{116}{\lambda_i}$
- $c_3 = 0.4$
- $c_4 = 5$
- $c_5 = \frac{21}{\lambda_i}$
- $c_6 = 0.0068$

Where $\lambda_i$can be written as follows (10):

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} , \tag{10}$$

The theoretical maximum of the power coefficient is 0.59 referred as the Betz limit, but the practical values vary from 0.2 to 0.5.The normalized wind turbine model is built from (3) and can be obtained as follows (11) [14]:

$$P_{pu} = k_p v_{pu}^3 C_{P\_pu} , \tag{11}$$

where:

- $P_{pu}$is the power of nominal power for particular values of air density and swept area in per unit
- C_{P_pu} is the power coefficient of the maximum value of $C_P$in per unit
- $v_{pu}$is the wind speed of the base wind speed in per unit. The base wind speed is the mean value of the expected wind speed in m/s.
- $k_p$is the power gain for C_{P_pu} = 1 per unit, and v_{pu} = 1 per unit. The k_p is less than or equal to 1.

Equation (7) is used for the second, non-normalized model.

B. *MATLAB/Simulink Model*

First the mathematical models were implemented in MATLAB/Simulink to verify their usability. Table 2 shows the inputs and outputs of the corresponding models.To avoid damage and unnecessary energy loss, wind turbines have mechanical breaks to stop the rotor when wind speed is lower than the cut-in speed and higher than the cut-out speed. The cut-in and cut-out wind speed thresholds and the angular velocity of the turbine are predefined inner parameters in the non-normalized case.

C. *Xilinx System Generator*

Engineers who use model-based design to target Xilinx FPGAs can use the Xilinx System Generator for DSP as a Simulink plug-in. This tool provides an easy graphical way to implement complex models using FPGA specific blocks collected into a Xilinx Library of the Simulink. This library contains general (add/sub, multiplication, divider, MATLAB-code, ROM block …) and Xilinx specific (AXI-Stream, PicoBlaze, Vivado…) blocks. If the model based on the Xilinx blocks is completed, the System Generator either can be used to simulate the behavior of the FPGA-based model on the PC or co-simulate it on the real FPGA hardware. In addition to this, the model can be converted into a (Hardware Description Language) HDL-based native Xilinx ISE Development Kit project for further development or analysis [17]. The Simulink can be used to feed data to and from the System Generator model. Since the Simulink uses double precision fixed-point numbers which are not efficiently implementable on FPGAs, fixed-point data conversion is required. This can be done by using the Gateway In and Gateway Out blocks of the Xilinx Library.The wind turbine models were implemented using the Xilinx library blocks and also the standard Simulink blocks. Using the co-simulation possibilities the outputs of the models were compared to each other to verify the correctness of the FPGA-based hardware model.

V. The Wind Turbine HIL Model

The architecture of the FPGA-based HIL system connected to a PC is shown in Fig. 2.The system is built up from a *Wind Turbine Core*, an *AD Core*, a *DA Core*, and an *UART&Comm Ctrl Core*.

The *Wind Turbine Core* is responsible for the real time computation of the output torque based on the normalized or the non-normalized models according to the input parameters. The input parameters denotedby red are used in both models and parameters denoted by blue are used only in the non-normalized model.

TABLE III. Input/Output Variables of Models

	Variable	*Description*	*Unit*
Normalized Model	*v*	Wind Speed	m/s
	beta	Blade pitch angle	degrees
	$omega_{gen}$	Angular velocity in per unit of base generator speed	rad/s
	T_{pu}	Generator shaft torque in per unit of the generator ratings	Nm
Non-Normalized Model	*v*	Wind Speed	m/s
	β	Blade pitch angle	degrees
	R	Blade Radius	m
	ρ	Air Density	kg/m^3
	T	Turbine torque	Nm

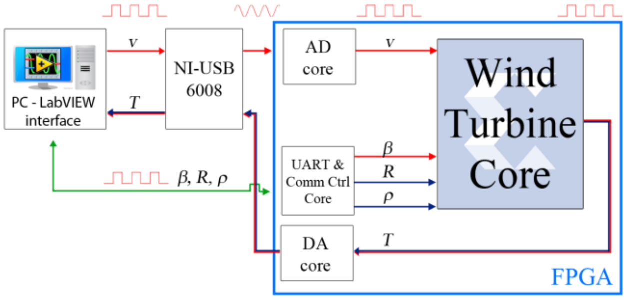

Fig. 2. The FPGA-based HIL system architecture

The *AD* and *DA Core* communicate with the PMOD AD1 and PMOD DA2 modules via SPI communication, which send and receive the input wind speed and output torque using analogue signals. The *UART&Comm Ctrl Core* enables an 8-bit serial UART communication to send digital commands and data to, or receive digital data from the FPGA HIL device. This core provides dynamic input parameter changes while the FPGA is operating. An UART state machine was built in the *UART&Comm Ctrl Core* in VHDL (Very High Speed Integrated Circuit Hardware Description Language), to control the simple UART RX/TX VHDL modules. The communication begins with an exclamation mark, and then the type of the request is sent together with the register address and the data from the LabVIEW interface. The 8-bit request can be a write or read register, the register address is 1 byte, and the data are 2 byte fix-point type.

A. *Wind Turbine Core*

The architecture of the normalized and non-normalized wind turbine models are shown in Fig. 3 and Fig. 4.

The main parts of the models are:

- Computation of λ
- Computation of C_P
- Computation of the torque

The computation of the Tip Speed Ratio (TSR) is made in the λ part. It is important, because the functions in the C_P part require the value of λ.At the end of the architecture the value of the torque is computed in both cases.In the normalized model it is necessary to multiply the output of the λ, $C_{P,}$ and torque parts with constants in order to get the output torque in per unit of the generator parameters.

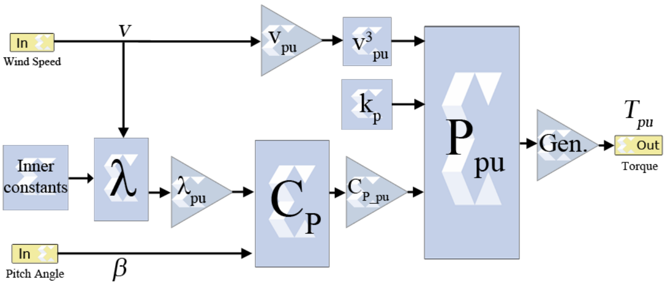

Fig. 3. The Normalized Wind Turbine model architecture

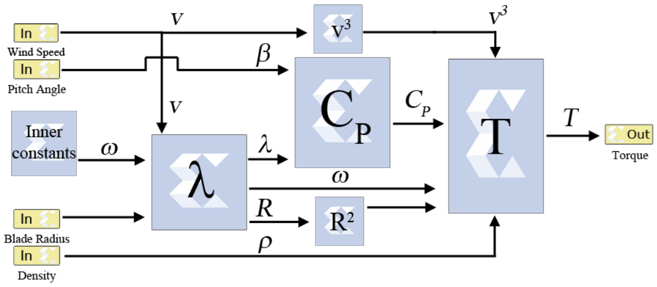

Fig. 4. Non-Normalized Wind Turbin model architecture

Because of the limited resources of the FPGA it is required to exactly define the input parameter ranges as it can be seen on Table 3. The input ranges define the computation time, the resource requirement and the latency of the architecture. During the implementation of the *Wind Turbine Core*s the following three problems have to be solved:

- Synchronization of the divider blocks
- Exponential function
- Timing errors

Either in the normalized or in the non-normalized model division is required. Using this operation in MATLAB or Simulink is simple, but to implement it in an FPGA circuit is more complicated. The divider block in the Xilinx System Generator uses the High Radix algorithm to compute the result of the division, which is an iterative method. Due to this, the output of a divider block is fluctuating till the appropriate result has been computed. In case of one or two dividers this problem can be solved with a well-planned latency value configured in the divider core, but using two or more dividers the system is going to be useless. The best solution for the problem is to use the divider blocks with the enable and the data ready ports. With these ports the data transfer between the blocks can be synchronized. Another problem appears when the input word length of the divider block is too large, since the sum of the length of input word must be less or equal to 64-bit. So, it is required to optimize the inputs data width.In case of the wind turbine model, allof the inputs are fix-point 8-bitintegers with 4 fraction bits, so the size of the inputs is 12 bits.

In case of both models the exponent function of the C_P part should be computed, but this function is not implemented for FPGAs in the Xilinx System Generator. This problem can be solved with using a ROM block with pre-calculated exponential function values.The ROM block automatically reserves the required amount of BRAM memory resources of the FPGA, depending on the required precision and interval.

TABLE IV. INPUT RANGES

Inputs	*Input Name*	*Unit*	*Range*	*Word length (bit)*
v	Wind Speed[5]	m/s	0-15	10, (fraction 4)
β	Pitch Angle	degree	0-30	10, (fraction 4)
R	Blade Length[5]	meter	0-64	12, (fraction 4)
ρ	Air Density	kg/m^3	0-1000	15, (fraction 4)

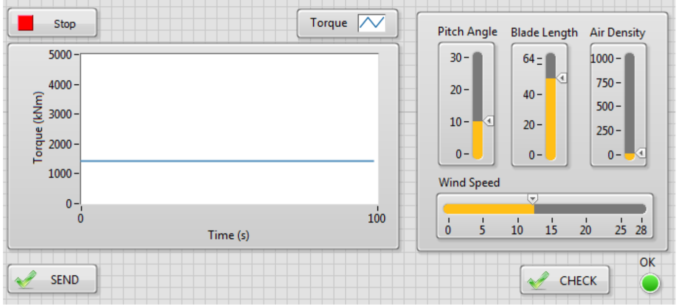

Fig. 5. The LabVIEW interface

It is not achievable to store all the possible exponential values with any precision, therefore the required precision and interval should be determined and optimized to lower the memory requirement of the design.

During the implementation phase of the designs the timing constraints are not met, because the design contains high latency data paths in case of the divider and multiplier blocks. To eliminate this error, delay registers should be added to the critical paths too.

B. *LabVIEW interface*

An interface was built in LabVIEW to realize analog and digital communication between the PC and the FPGA-basedHIL device. The interface enables the use of UART serial communication with the built-in VISA driver, and analog communication via the NI-USB6008 device. Using this interface the pitch angle, the blade length, the air density, and the wind speed can be set using 4 different sliders, and the output torque of the model can be examined on a waveform graph. The interface is shown in Fig. 5.

VI. Results

The normalized and the non-normalized model were tested either using Simulink-based simulation or real FPGA hardware. During the test the pitch angle was 0°, the blade length was 50m, the air density was 1.25kg/m^3, the angular velocity was 1.46rad/s and the wind speed was changed linearly between 3m/s and 15m/s.

The best way to describe the behavior of a wind turbine is the C_p–TSR characteristic, which plots the extracted power from the wind in the function of the TSR. The comparison of the C_p – TSR curves in case of the Simulink and FPGA-based models can be seen in Fig. 6. The characteristics of the FPGA-based models are denoted by red, while the characteristics of the Simulink simulation are denoted by blue. During the tests the wind speed was increased with 0.01 steps. The FPGA-based curves are stepped, because the divider blocks use the High Radix algorithm.In Fig. 6 it can be seen the FPGA and the Simulink-based characteristics are very similar, so the HIL system works correctly for the two models.Below 3 m/s wind speed none of the models were working correctly, but in general the wind turbine systems work with 3 m/s cut-in wind speed.

The resource requirement of the models is shown in Table 4. The BRAM and the DSP Slice requirements are the main bottlenecks of the implementation.

TABLE V. RESOURCE REQUIREMENTS OF THE MODELS

Resource Type	*Name*	*Normalized*	*Usage*	*Non-Normalized*	*Usage*
General resources	Slices	2405	3%	2306	3%
	6-input LUT	5959	9%	5574	8%
	Flip-Flops	4484	3%	4380	3%
Dedicated resources	36Kbit BRAM	84	62%	84	62%
	DSP48E1 Slice	170	70%	164	68%

The DSP Slice requirement is the function of the number of the add/sub and multiplication/divider blocks, while the BRAM requirement is the function of the required precision and interval of the exponential function. The bottleneck of the implementation is the number of BRAM and DSP Slices.

Improving the output precision or the input parameter range of the model it is required to increase the precision or the interval of the stored e^x function which increase the BRAM requirement. So an optimum should be found between the precision and the input parameter range to fit the design into the Artix-7 FPGA-based Nexys-4 board. This can be found if the wind speed is changing between 3m/s and 15m/s. In this case an acceptable computing precision can be reached.

VII. CONCLUSION AND FUTURE WORK

In this paper a horizontal three-bladed variable pitch wind turbine model waspresented and implementedon a Digilent Nexys-4 board containing Xilinx Artix-7 FPGA as a HIL device using Xilinx System Generator. In the normalized model the wind speed is an analog adjustable input parameter converted to digital data with the AD core and interface, while the pitch angle of the blades of the wind turbine is a digital adjustable input parameter. The output is the torque applied to the generator shaft in per unit of the generator ratingsconverted with the DA core and interface into analog voltage. The non-normalized model has another two changeable digital input parameters: blade length, and air density. These parameters can be setup with serial and analog communication using a LabVIEW interface.

This FPGA-based HIL system with the LabVIEW-based PC software can be used not only for experimental but for education purposes as well. Students can analyze the operation of the wind turbine, and study the effect of different parameter changes, without the risk of damaging or endangering the real physical system.

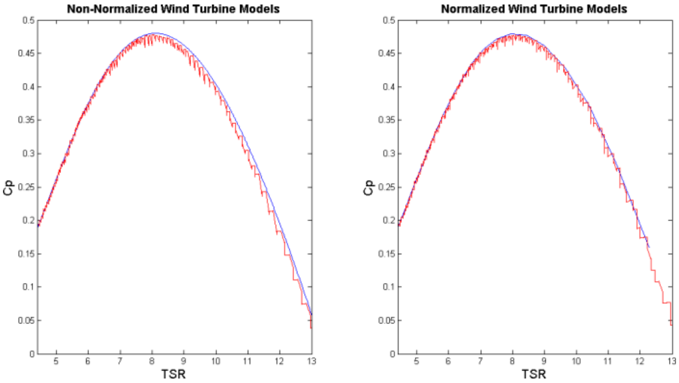

Fig. 6. Cp-TSR graphs of the models

The characteristics of the PC and FPGA-based models were compared and it can be concluded that the output of the FPGA-based models is similar compared to the Simulink-based simulations. The results showed that a three bladed variable pitch wind turbine can be implemented on an FPGA and used as a real-time dynamically adjustable HIL system. In the future, this HIL system could be used for power output estimation of various generator models, comparison of different controlling methods oreven a real miniature wind sensor and generator could be connected to it.

ACKNOWLEDGMENT

This research project has been supported by the HUSRB HUSRB/1203/214/248, Renewable Energy Virtual Laboratory (REVLAB) project.

This document has been produced with the financial assistance of the European Union. The content of the document is the sole responsibility of University of Novi Sad and can under no circumstances be regarded as reflecting the position of the European Union and/or the Managing Authority.

REFERENCES

[1] WWEA 2014 half-year report: http://www.wwindea.org

[2] P. Jansuya, Y. Kumsuwan, “Design of MATLAB/Simulink Modeling of Fixed-Pitch Angle Wind Turbine Simulator”, Energy Procedia, Volume 34, 2013, pp. 362–370

[3] S. Santoso, H.T. Le, “Fundamental time–domain wind turbine models for wind power studies”,Renewable Energy, Volume 32, Issue 14, November 2007, pp. 2436–2452

[4] V. Thapara, G.Agnihotrib, V. K. Sethic, “Critical analysis of methods for mathematical modelling of wind turbines”,Volume 36, Issue 11, November 2011, pp. 3166–3177

[5] C. Carrillo, A.F. Obando Montaño, J. Cidrás, E. Díaz-Dorado, “Review of power curve modelling for wind turbines”,Renewable and Sustainable Energy Reviews,Volume 21, May 2013, pp. 572–581

[6] J.G Slootweg., S.W.H. de Haan, H. Polinder, W.L. Kling, “General Model for Representing Variable Speed Wind Turbines in Power System Dynamics Simulation”,Power Systems, IEEE Transactions on Volume:18, Issue: 1, Feb 2003, pp. 144 – 151

[7] C. Washington, J. Dolman, “Creating Next Generation HIL Simulators with FPGA Technology”,Autotestcon, IEEE,Sept. 2010, pp. 1 – 6

[8] H. Chen, S. Sun, D.C. Aliprantis, J. Zambreno, “Dynamic Simulation of DFIG Wind Turbines on FPGA Boards”, Power and Energy Conference at Illinois (PECI), IEEE, 2010, pp. 39 – 44

[9] H. Yang, J. Zhang, J. Sun, L. Yu, “Review of Advanced FPGA Architectures and Technologies”Journal of Electronics (China),Volume 31, Issue 5 , pp 371-393

[10] Xilinx website: http://www.xilinx.com

[11] Digilent website: http://www.digilentinc.com

[12] AD7476 datasheet: http://www.analog.com

[13] DAC121S101 datasheet: http://www.ti.com

[14] Implement model of variable pitch wind turbine: http://www.mathworks.com

[15] F. Iov, A. D. Hansen, P. Sorensen, F. Blaabjerg,”Wind Turbine Blockset in Matlab/Simulink”,Aalborg University, 2004, ISBN 87-89179-46-3

[16] Siegfried Heier, "Grid Integration of Wind Energy Conversion Systems," John Wiley & Sons Ltd, 1998, ISBN 0-471-97143-X

[17] Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP: http://www.mathworks.com