Dynamic Simulation of DFIG Wind Turbines on FPGA Boards

Hao Chen, *Student Member, IEEE,* Song Sun, *Student Member, IEEE,*
Dionysios C. Aliprantis, *Senior Member, IEEE,* and Joseph Zambreno, *Member, IEEE*

***Abstract*—This paper presents the implementation of a dynamic simulation of a doubly fed induction generator (DFIG)-based wind turbine on a field-programmable gate array (FPGA) board. The explicit fourth-order Runge–Kutta numerical integration algorithm is used to obtain the system dynamic response. The FPGA simulation results and speed improvement are validated versus a Matlab/Simulink simulation. Using FPGAs as computational engines can lead to significant simulation speed gains when compared to a typical PC computer, especially when operations can be efficiently parallelized on the board.**

I. Introduction

A field-programmable gate array (FPGA) is a reconfigurable digital logic platform, which allows for the parallel execution of millions of bit-level operations in a spatially programmed environment. Research has been under way on the modeling and real-time simulation of various electrical power components using FPGAs as computational [1]–[6] and non-computational [7], [8] devices. Herein, the goal is to implement an entire dynamic simulation of a doubly fed induction generator (DFIG) wind turbine system on a single FPGA board as fast as possible (i.e., without being constrained by the requirement of real-time simulation).

The individual mathematical operations required by numerical integration algorithms are generally simple in terms of required logic (additions and multiplications). Hence, hardware implementations can be used to increase efficiency by reducing the overhead introduced by software, thus leading to simulation speed gains of two orders of magnitude when compared to PCs. Moreover, complex systems requiring the simultaneous solution of numerous differential equations for simulation are inherently conducive to a parallel mapping to physical computational resources. Therefore, an FPGA becomes an attractive choice for simulating complex electrical power and energy systems. Herein, a DFIG wind turbine system model is designed using very high speed integrated circuit hardware description language (VHDL), synthesized and verified using Xilinx integrated software environment (ISE). The basic steps of designing an explicit fourth-order Runge–Kutta (RK4) numerical ordinary differential equation (ODE) solver on the FPGA platform are outlined.

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA. Email: {chenh, sunsong, dali, zambreno}@iastate.edu.

This project was financially supported by the "CODELESS: COnfigurable DEvices for Large-scale Energy System Simulation" project, funded by the Electrical Power Research Center (EPRC) at Iowa State University.

978-1-4244-5901-8/10/$26.00 ©2010 IEEE

II. Modeling and Control

A. Induction Machine and Wind Turbine Model

The fifth-order induction machine model in the stationary reference frame is given by [9]:

$$v_{qs}^{s} = R_s i_{qs}^{s} + p(L_s i_{qs}^{s} + L_m i_{qr}^{\prime s}) \quad (1)$$

$$v_{ds}^{s} = R_s i_{ds}^{s} + p(L_s i_{ds}^{s} + L_m i_{dr}^{\prime s}) \quad (2)$$

$$v_{qr}^{\prime s} = R_r^{\prime} i_{qr}^{\prime s} - \omega_r(L_r^{\prime} i_{dr}^{\prime s} + L_m i_{ds}^{s}) + p(L_r^{\prime} i_{qr}^{\prime s} + L_m i_{qs}^{s}) \quad (3)$$

$$v_{dr}^{\prime s} = R_r^{\prime} i_{dr}^{\prime s} + \omega_r(L_r^{\prime} i_{qr}^{\prime s} + L_m i_{qs}^{s}) + p(L_r^{\prime} i_{dr}^{\prime s} + L_m i_{ds}^{s}) \quad (4)$$

$$p\omega_r = \tfrac{P}{2J}(T_e - T_m - B\tfrac{2}{P}\omega_r) \quad (5)$$

$$T_e = 0.75PL_m(i_{qs}^{s} i_{dr}^{\prime s} - i_{ds}^{s} i_{qr}^{\prime s}) \quad (6)$$

where $p = \frac{d}{dt}$ is the differentiation operator; R_s and $R_r^{\prime}$ are the stator and rotor resistances; L_s and $L_r^{\prime}$ are the stator and rotor inductances; L_m is the magnetizing inductance; v_{qs}^{s}, v_{ds}^{s}, i_{qs}^{s}, and i_{ds}^{s} are the *qd*-axes stator voltages and currents; $v_{qr}^{\prime s}$, $v_{dr}^{\prime s}$, $i_{qr}^{\prime s}$, and $i_{dr}^{\prime s}$ are the *qd*-axes rotor voltages and currents; ω_r is the rotor angular electrical speed; T_m and T_e are mechanical and electromagnetic torque; P is the number of poles; J is the total rotor inertia; and B is a friction coefficient.

The wind turbine model is based on the relation between the upstream wind speed v_w and the mechanical power P_m extracted from the wind [10]. The pertinent equations are

$$P_m = \tfrac{1}{2}\rho\pi R_w^2 c_p(\lambda,\beta)\, v_w^3 \quad (7)$$

$$c_p(\lambda,\beta) = 0.5176\left(\tfrac{116}{\lambda_i} - 0.4\beta - 5\right)e^{\frac{-21}{\lambda_i}} + 0.0068\lambda \quad (8)$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \quad (9)$$

where ρ is the air density; R_w is the wind turbine radius; $c_p(\lambda,\beta)$ is the performance coefficient; β is the pitch angle in degrees; and λ is the tip-speed ratio given by $\lambda = \omega_w R_w / v_w$, where ω_w is the wind turbine rotor speed. Note that the relation between ω_w and ω_r is determined by the gearbox ratio. For $\beta = 0$, the performance coefficient attains its maximum value $c_p^{\max} = 0.48$ for an optimal $\lambda_{op} = 8.10$. The mechanical torque applied to the generator shaft is $T_m = \frac{P}{2}\frac{P_m}{\omega_r}$.

B. DFIG Control

An essential characteristic of DFIG control strategy is that the generated active and reactive powers can be controlled independently. It is common to use the air-gap flux oriented vector control [11] or the stator flux oriented vector control [12]–[14], under the assumption of negligible stator resistances. In particular, it has been shown that stator flux

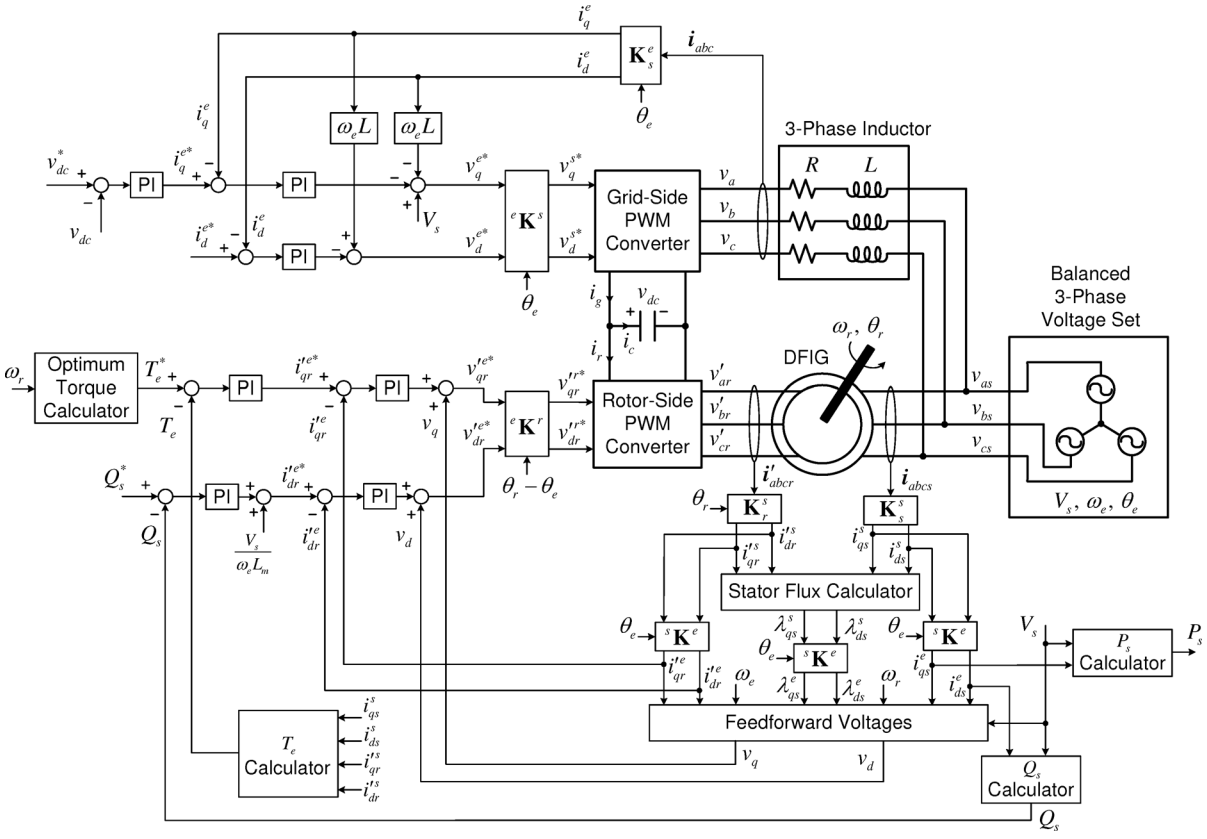

Fig. 1. Control block diagram for DFIG

orientation can cause instability under certain operating conditions [15]. Herein, a stator voltage oriented vector control without the assumption of negligible stator resistances is presented. The overall control structure of back-to-back pulse-width modulated (PWM) converters is shown in Fig. 1. The reference frame transformation matrices $\mathbf{K}_s^s$, $\mathbf{K}_r^s$, $\mathbf{K}_s^e$, ${}^s\mathbf{K}^e$, ${}^e\mathbf{K}^s$, and ${}^e\mathbf{K}^r$ are defined in [9].

1) Rotor current control: Aligning the stator voltage vector with the q-axis, the induction machine voltage equations in the synchronous reference frame can be written as

$$V_s = R_s i_{qs}^e + \omega_e \lambda_{ds}^e + p\lambda_{qs}^e \tag{10}$$
$$0 = R_s i_{ds}^e - \omega_e \lambda_{qs}^e + p\lambda_{ds}^e \tag{11}$$
$$v_{qr}'^e = R_r' i_{qr}'^e + (\omega_e - \omega_r)\lambda_{dr}'^e + p\lambda_{qr}'^e \tag{12}$$
$$v_{dr}'^e = R_r' i_{dr}'^e - (\omega_e - \omega_r)\lambda_{qr}'^e + p\lambda_{dr}'^e \tag{13}$$

where V_s is the stator voltage amplitude, and ω_e is the stator voltage angular frequency. The rotor flux linkage equations are

$$\lambda_{qr}'^e = L_r' i_{qr}'^e + L_m i_{qs}^e = (L_m/L_s)\lambda_{qs}^e + \sigma L_r' i_{qr}'^e \tag{14}$$
$$\lambda_{dr}'^e = L_r' i_{dr}'^e + L_m i_{ds}^e = (L_m/L_s)\lambda_{ds}^e + \sigma L_r' i_{dr}'^e \tag{15}$$

where $\sigma = 1 - L_m^2/L_s L_r'$. Substituting (14) and (15) into (12) and (13) yields

$$\begin{aligned} v_{qr}'^{e*} = v_{qr}'^e &= R_r' i_{qr}'^e + \sigma L_r' p i_{qr}'^e + v_q \\ &= K_q\left(1 + \frac{1}{\tau_q s}\right)(i_{qr}'^{e*} - i_{qr}'^e) + v_q \end{aligned} \tag{16}$$

$$\begin{aligned} v_{dr}'^{e*} = v_{dr}'^e &= R_r' i_{dr}'^e + \sigma L_r' p i_{dr}'^e + v_d \\ &= K_d\left(1 + \frac{1}{\tau_d s}\right)(i_{dr}'^{e*} - i_{dr}'^e) + v_d \end{aligned} \tag{17}$$

where v_q and v_d are compensating feedforward voltages given by

$$v_q = \tfrac{L_m}{L_s} p\lambda_{qs}^e + (\omega_e - \omega_r)[\sigma L_r' i_{dr}'^e + \tfrac{L_m}{L_s}\lambda_{ds}^e] \tag{18}$$
$$v_d = \tfrac{L_m}{L_s} p\lambda_{ds}^e - (\omega_e - \omega_r)[\sigma L_r' i_{qr}'^e + \tfrac{L_m}{L_s}\lambda_{qs}^e] \tag{19}$$

and K_q, τ_q, K_d, and τ_d are parameters of two PI current controllers. The stator flux linkages (λ_{qs}^e, λ_{ds}^e) are computed from the stator and rotor current measurements. The derivatives $p\lambda_{qs}^e$ and $p\lambda_{ds}^e$ are obtained from (10) and (11).

2) Torque and power control: The optimal electromagnetic torque reference T_e^* shown in Fig. 1, after compensating for the friction losses, is given by [12], [13]

$$T_e^* = \frac{P}{2}\frac{K_1^3 K_2\, c_p^{\max}\omega_r^2}{\lambda_{op}^3} - B\frac{2}{P}\omega_r \tag{20}$$

where $K_1 = 2R_w/GP$, $K_2 = \frac{1}{2}\rho\pi R_w^2$, and G is the gearbox ratio.

3) Grid-side converter control: The purpose of the grid-side converter is to regulate the DC-link voltage [12]. The vector control approach shown in Fig. 1 is used. Aligning the stator voltage vector with the *q*-axis, the voltage equations in the synchronous reference frame can be written as

$$v_{qs}^e = V_s = Ri_q^e + Lpi_q^e + \omega_e Li_d^e + v_q^e \tag{21}$$

$$v_{ds}^e = 0 = Ri_d^e + Lpi_d^e - \omega_e Li_q^e + v_d^e \tag{22}$$

where R and L are the resistance and inductance of the current's filter inductors, and v_q^e, v_d^e, i_q^e, and i_d^e are the *qd*-axis converter input voltages and currents. From (21) and (22), the converter voltage references v_q^{e*} and v_d^{e*} are

$$\begin{aligned} v_q^{e*} = v_q^e &= -\left(Ri_q^e + Lpi_q^e\right) + V_s - \omega_e Li_d^e \\ &= -K_{qg}\left(1 + \frac{1}{\tau_{qg}s}\right)(i_q^{e*} - i_q^e) + V_s - \omega_e Li_d^e \end{aligned} \tag{23}$$

$$\begin{aligned} v_d^{e*} = v_d^e &= -\left(Ri_d^e + Lpi_d^e\right) + \omega_e Li_q^e \\ &= -K_{dg}\left(1 + \frac{1}{\tau_{dg}s}\right)(i_d^{e*} - i_d^e) + \omega_e Li_d^e \end{aligned} \tag{24}$$

where K_{qg}, τ_{qg}, K_{dg}, and τ_{dg} are parameters of two PI current controllers. Herein, i_d^{e*} is arbitrarily set to zero in order to set the stator-side reactive power to zero, but this is not always necessary in practice.

III. FPGA Implementation

A. Simulation Architecture

The transient response of the system is obtained by the RK4 numerical integration algorithm [16]. This is a fixed-step explicit integration algorithm, which is based on simple numerical calculations (additions and multiplications), and is thus straightforward to implement on the FPGA. The RK4 method for the initial value problem $(p\mathbf{x} = \mathbf{f}(t, \mathbf{x}), \mathbf{x}(t_0) = \mathbf{x}_0)$ is described by:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \tfrac{h}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right) \tag{25}$$

$$t_n = t_{n-1} + h \tag{26}$$

where $\mathbf{x}_n$ is the RK4 approximation of $\mathbf{x}(t_n)$ (i.e., the exact solution), h is the time step, and

$$\mathbf{k}_1 = \mathbf{f}\left(t_{n-1}, \mathbf{x}_{n-1}\right) \tag{27}$$

$$\mathbf{k}_2 = \mathbf{f}\left(t_{n-1} + 0.5h, \mathbf{x}_{n-1} + 0.5h\mathbf{k}_1\right) \tag{28}$$

$$\mathbf{k}_3 = \mathbf{f}\left(t_{n-1} + 0.5h, \mathbf{x}_{n-1} + 0.5h\mathbf{k}_2\right) \tag{29}$$

$$\mathbf{k}_4 = \mathbf{f}\left(t_{n-1} + h, \mathbf{x}_{n-1} + h\mathbf{k}_3\right). \tag{30}$$

The ODEs representing the entire DFIG system, expressed in the form $p\mathbf{x} = \mathbf{f}(t, \mathbf{x})$, are derived by combining the induction machine model, wind turbine model and DFIG control strategy. The state variables are i_{qs}^s, i_{ds}^s, $i_{qr}'^s$, $i_{dr}'^s$, ω_r, the integrators of four PI controllers for the rotor-side converter (x_6, x_7, x_8, and x_9), the three-phase RL circuit *qd* currents in the stationary reference frame (i_q^s and i_d^s), the DC-link voltage v_{dc}, and the integrators of three PI controllers for the grid-side converter (x_{13}, x_{14}, and x_{15}). The input variables are v_{qs}^s, v_{ds}^s, the wind speed v_w, the pitch angle β, the reactive power reference Q_s^*, the DC-link voltage reference v_{dc}^*, and the three-phase RL circuit d-axis current reference in the synchronous reference frame i_d^{e*}. The output variables are the stator-side active and reactive power.

As shown in Fig. 2, four functional modules are used to establish the entire system. The "Stator Voltage Input" module is responsible for the generation of v_{qs}^s and v_{ds}^s. The "ODE Function" and "Vector Update" modules constitute the RK4 solver. The "Output" module implements the calculation of the stator-side active and reactive power (P_s and Q_s) and the machine rotor *qd*-axes currents in the synchronous reference frame ($i_{qr}'^e$ and $i_{dr}'^e$). These modules have been developed using VHDL in ModelSim, which is a verification and simulation tool for VHDL designs. All variables and parameters are represented as signed fixed-point numbers with 13 bits representing the integral part, and 32 bits representing the fractional part. This provides a numerical range that can accommodate every variable involved in the simulation, with a resolution of 2^{-32}. For economizing FPGA resources, the per unit system is used in order to decrease the necessary number of bits (because variables are expected to be close to 1.0).

Every RK4 iteration shown in Fig. 3 consists of six steps. The "ODE Function" module executes the evaluation of $\mathbf{f}(t, \mathbf{x})$. The "Vector Update" module is responsible for the alteration of $\mathbf{x}$ in $\mathbf{f}(t, \mathbf{x})$ during steps 2, 3, and 4, as well as the calculation of (25) in step 5. Since v_{qs}^s and v_{ds}^s are dependent on the time t, the "Stator Voltage Input" module should generate the appropriate v_{qs}^s and v_{ds}^s for the "ODE Function" module. Specifically, $v_{qs}^s(t_{n-1} + 0.5h)$ and $v_{ds}^s(t_{n-1} + 0.5h)$ are generated during step 1 and stored for the usage of the "ODE Function" module in step 2 and step 3, while $v_{qs}^s(t_{n-1} + h)$ and $v_{ds}^s(t_{n-1} + h)$ are generated during step 3 and stored for the usage of the "ODE Function" module in step 4 and step 1 of the next iteration. Note that the "Stator Voltage Input" module and the "ODE Function" module are executed in parallel in step 1. A similar parallel execution is also performed in step 3. On the other hand, the "ODE Function" module and the "Vector Update" module have to be executed in serial pattern because the inputs of one strictly depend on the outputs of the other.

To design a sinusoidal function involved in the 'Stator Voltage Input' module, a look-up table approach is followed [17]. The evaluation of the exponential function involved in (8) is based on the following identities [18]:

$$\begin{aligned} e^x &= 2^{x\log_2 e} \\ &= 2^{x_i}2^{x_f} = 2^{x_i}e^{x_f \ln 2}, \quad x > 0 \end{aligned} \tag{31}$$

$$\begin{aligned} e^x &= 2^{x\log_2 e} \\ &= 2^{x_i - 1}2^{x_f + 1} = 2^{x_i - 1}e^{(x_f + 1)\ln 2}, \quad x < 0 \end{aligned} \tag{32}$$

where x_i and x_f are the integer and fractional part of $x\log_2 e$, respectively.[1] Since $0 < x_f \ln 2 < 1$ in (31) and $0 < (x_f + 1)\ln 2 < 1$ in (32), the 32 bits representing these decimal fractions can be divided into 3 sections: bits 2^{-1} to 2^{-8} (x_H), bits 2^{-9} to 2^{-16} (x_L) and bits 2^{-17} to 2^{-32}

[1] For example, if $x\log_2 e = 2.3$, then $x_i = 2$ and $x_f = 0.3$; if $x\log_2 e = -2.3$, then $x_i = -2$ and $x_f = -0.3$.

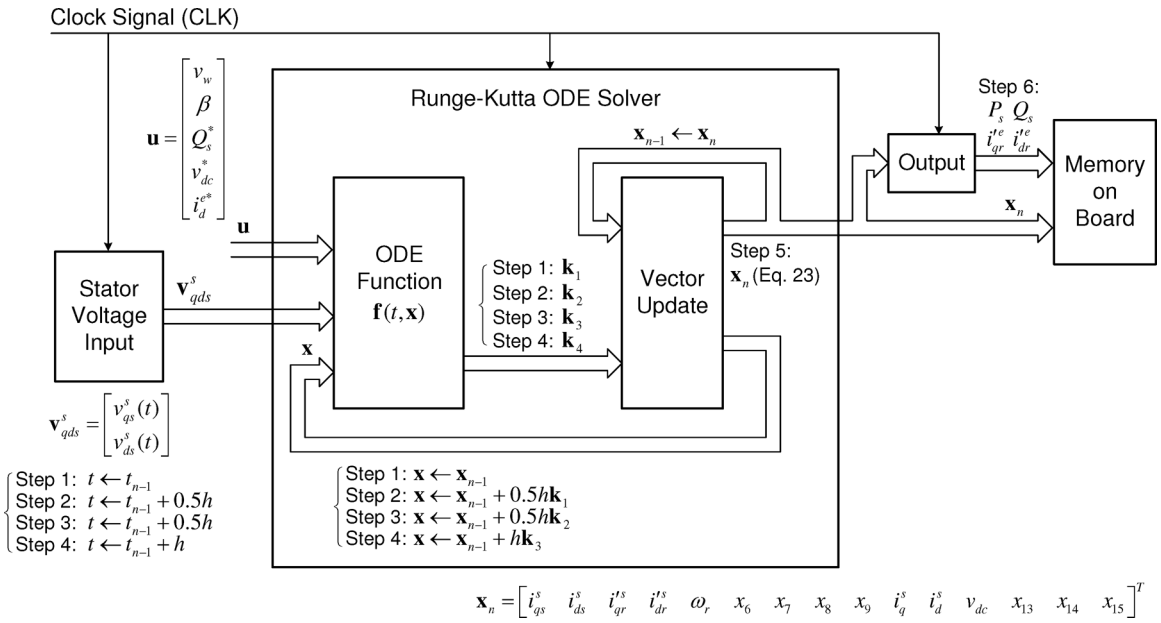

Fig. 2. FPGA implementation of DFIG wind turbine system

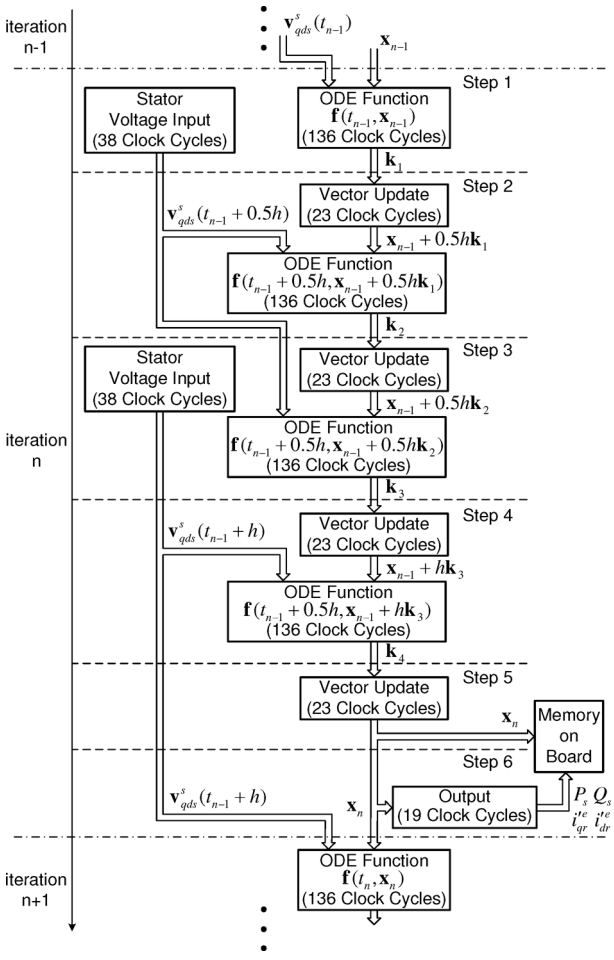

Fig. 3. RK4 iteration process

(x_T). e^{x_H} and e^{x_L} are obtained using two exponential look-up tables (named as 'high 8 table' and 'low 8 table'). Each of them contains 256 elements. e^{x_T} is calculated by Taylor series expansion ($e^{x_T} \approx 1 + x_T$). Thus,

$$e^x = 2^{x_i} e^{x_H} e^{x_L} (1 + x_T), \quad x > 0 \quad \text{or}$$
$$e^x = 2^{x_i - 1} e^{x_H} e^{x_L} (1 + x_T), \quad x < 0 .$$

The multiplication by 2^{x_i} or $2^{x_i - 1}$ is executed by a bit-shifting operation.

B. Synthesis and Implementation

After the functionality and results of all modules designed using VHDL were validated in the ModelSim environment, the Xilinx ISE was used to develop, synthesize, and verify the substantial top-level wrapper module together with the DFIG wind turbine system model. The target FPGA device was Xilinx Virtex-5 XC5VLX330. The post-place and route report presented the FPGA hardware resources usage as shown in Table I, and the maximum frequency of the clock signal that can be applied is 221.533 MHz. Generally, the consumption of FPGA hardware resources increases with the model complexity. Note that the entire design for the DFIG wind turbine system must fit within the resource limitation of the target FPGA device. Otherwise, an FPGA device with more hardware resources should be chosen or the model should be redesigned in order to meet the requirement of the FPGA device. The final system will be integrated on a development board that features the XC5VLX330 device—for example, Xilinx Virtex-5 and DDR2 SDRAM multi-application platform board. The simulation output data will be stored in the memory embedded on the development board.

IV. Simulation Results

The simulation parameters are shown in Table II. The moment of inertia J was set to 2 kg·m^2 (an unrealistically low value) in order to reduce the simulation time required to reach

TABLE I
XILINX VIRTEX–5 XC5VLX330 RESOURCES USAGE SUMMARY

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	86288	207360	41%
Number of Slice LUTs (Look Up Tables)	80997	207360	39%
Number of LUT-FF (Flip Flop) pairs	91913	207360	44%

a steady-state operating condition. An average-value model is used to represent the rotor-side and grid-side converters shown in Fig. 1. The ModelSim clock frequency was (arbitrarily) set to 200 MHz, a value less than the maximum clock frequency (221.533 MHz) in the post-place and route report of the Xilinx ISE. The simulation time-step h was 10^{-4} s.

The exact same DFIG wind turbine system was also implemented in Matlab/Simulink. The verification of the results coming from ModelSim was performed versus a Simulink simulation using the ODE23tb solver with a maximum time step of 10^{-5} s. Fig. 4 shows the machine stator and rotor qd-axes currents in the synchronous reference frame (i^e_{qs}, i^e_{ds}, i'^e_{qr}, and i'^e_{dr}), the stator-side active and reactive power (P_s and Q_s), the rotor angular electrical speed ω_r, and the DC-link voltage v_{dc}. The wind speed v_w was stepped down from 7 m/s to 5 m/s at $t = 1$ s, and the reactive power reference Q^*_s was stepped up from 10 kVAR to 50 kVAR at $t = 3$ s. Note that in the per unit system, the value of i^e_{qs} is equal to that of P_s and the value of i^e_{ds} is equal to that of Q_s. The ModelSim waveforms are superimposed on the Simulink waveforms, but they are so close that differences cannot be distinguished.

To compare simulation speed, we ran the simulation using the ODE45 and ODE23 integration algorithms of Simulink with maximum step size of 10^{-4} s (typically the two "simplest" available solvers), because they are implementations of the explicit Runge–Kutta algorithm, albeit of a variable-step nature. The simulation speed was further increased by using the "Accelerator" mode of Simulink, which replaces normal interpreted code with compiled code. The simulation times on an Intel Core2 Duo 2.2 GHz computer were 6.7 s for ODE45 and 4.7 s for ODE23. The FPGA simulation time predicted by ModelSim was 0.166 s, which represents a 40x speed gain. The simulation time will be further decreased if the clock frequency can be set to a higher value, if the simulation time step h is increased, or if a lower-order integration algorithm (e.g., the trapezoidal algorithm) is used.

V. CONCLUSION

This paper presented the FPGA implementation of a DFIG wind turbine system dynamic simulation, using the RK4 numerical integration algorithm. The entire system has been developed using VHDL, synthesized using the Xilinx ISE, and will be implemented on an FPGA board. An optimal VHDL design should be sought for the purpose of economizing FPGA hardware resources, especially when the model has high complexity. A comparison between the simulation results from FPGA and Simulink demonstrates the validity of this implementation. The 40x simulation speed gain demonstrates the performance advantage of FPGAs compared to PC-based simulations.

FPGAs represent an interesting possibility for simulating more complex electrical power and power electronics-based systems because of their flexibility, high processing rates and possibility to parallelize numerical integration computations. In principle, FGPAs could be coupled with other simulation platforms to perform multi-rate co-simulation of complex systems. To accelerate the dynamic simulations, FPGAs would simulate faster subsystems that require smaller integration time steps. However, it has been observed that the data exchange rate with an FPGA can be a critical bottleneck for developing such co-simulation applications, especially when it is required to achieve real-time simulation speeds. On the other hand, a pipeline VHDL design [19], [20] of a DFIG wind energy conversion system can potentially enable the dynamic simulation of entire wind farms (containing hundreds of turbines) on a single FPGA board.

REFERENCES

[1] M. Martar, M. Adbel-Rahman, and A.-M. Soliman, "FPGA-based real-time digital simulation," in *Int. Conf. Power Syst. Transients*, Montreal, Canada, Jun. 2005.

[2] P. Le-Huy, S. Guérette, L. A. Dessaint, and H. Le-Huy, "Real-time simulation of power electronics in power systems using an FPGA," in *Canadian Conf. Electr. Comp. Eng.*, May 2006, pp. 873–877.

[3] ——, "Dual-step real-time simulation of power electronic converters using an FPGA," in *IEEE Int. Symp. Ind. Electron.*, Montreal, Canada, Jul. 2006, pp. 1548–1553.

[4] J. C. G. Pimentel and H. Le-Huy, "Hardware emulation for real-time power system simulation," in *IEEE Int. Symp. Ind. Electron.*, Montreal, Canada, Jul. 2006, pp. 1560–1565.

[5] J. C. G. Pimentel, "Implementation of simulation algorithms in FPGA for real time simulation of electrical networks with power electronics devices," in *IEEE Int. Conf. Reconfig. Comp. & FPGA's*, Sep. 2006, pp. 1–8.

[6] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235–1246, Apr. 2007.

[7] T. Maguire and J. Giesbrecht, "Small time-step ($< 2\mu$Sec) VSC model for the real time digital simulator," in *Int. Conf. Power Syst. Transients*, Montreal, Canada, Jun. 2005.

[8] C. Dufour, J. Bélanger, S. Abourida, and V. Lapointe, "FPGA-based real-time simulation of finite-element analysis permanent magnet synchronous machine drives," in *IEEE Power Electron. Spec. Conf.*, Jun. 2007, pp. 909–915.

[9] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*, 2nd ed. IEEE Press, 2002.

[10] S. Heier, *Grid Integration of Wind Energy Conversion Systems*, 2nd ed. Chichester, England; Hoboken, NJ: Wiley, 2006.

[11] M. Yamamoto and O. Motoyoshi, "Active and reactive power control for doubly-fed wound rotor induction generator," *IEEE Trans. Power Electron.*, vol. 6, no. 4, pp. 624–629, Oct. 1991.

[12] R. Pena, J. C. Clare, and G. M. Asher, "Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation," *IEE Proc. Elec. Power Appl.*, vol. 143, no. 3, pp. 231–241, May 1996.

[13] A. D. Hansen, P. Sørensen, F. Iov, and F. Blaabjerg, "Control of variable speed wind turbines with doubly-fed induction generators," *Wind Energy*, vol. 28, no. 4, pp. 411–434, Jun. 2004.

[14] R. Fadaeinedjad, M. Moallem, and G. Moschopoulos, "Simulation of a wind turbine with doubly fed induction generator by FAST and Simulink," *IEEE Trans. Energy Convers.*, vol. 23, no. 2, pp. 690–700, Jun. 2008.

[15] A. Petersson, L. Harnefors, and T. Thiringer, "Comparison between stator-flux and grid-flux-oriented rotor current control of doubly-fed induction generators," in *IEEE 35th Annual Power Electr. Spec. Conf.*, vol. 1, 20-25 Jun. 2004, pp. 482–486.

[16] W. Gautschi, *Numerical Analysis: An Introduction*. Boston: Birkhäuser, 1997.

TABLE II
SIMULATION PARAMETERS [14]

R_s	$1.4 \times 10^{-3}\ \Omega$	P	4	β	0°	τ_{T_e}	0.0158	$K_{V_{dc}}$	0.5773
R'_r	$9.92 \times 10^{-4}\ \Omega$	ω_e	$2\pi 60$ rad/s	C	4000 μF	K_{Q_s}	0.0024	$\tau_{V_{dc}}$	0.02
L_s	1.616 mH	J	2 kg·m^2	R	0.1 Ω	τ_{Q_s}	0.0158		
L'_r	1.608 mH	P_{rated}	2 MW	L	1.75 mH	K_q, K_d	0.0474		
L_m	1.526 mH	R_w	35 m	v^*_{dc}	700 V	τ_q, τ_d	0.0135		
V_s	$690\sqrt{2/3}$ V	G	120	i^{e*}_d	0 A	K_{qg}, K_{dg}	1.3		
B	0.01 N·m·s	ρ	1.25 kg/m^3	K_{T_e}	0.4488	τ_{qg}, τ_{dg}	0.0023		

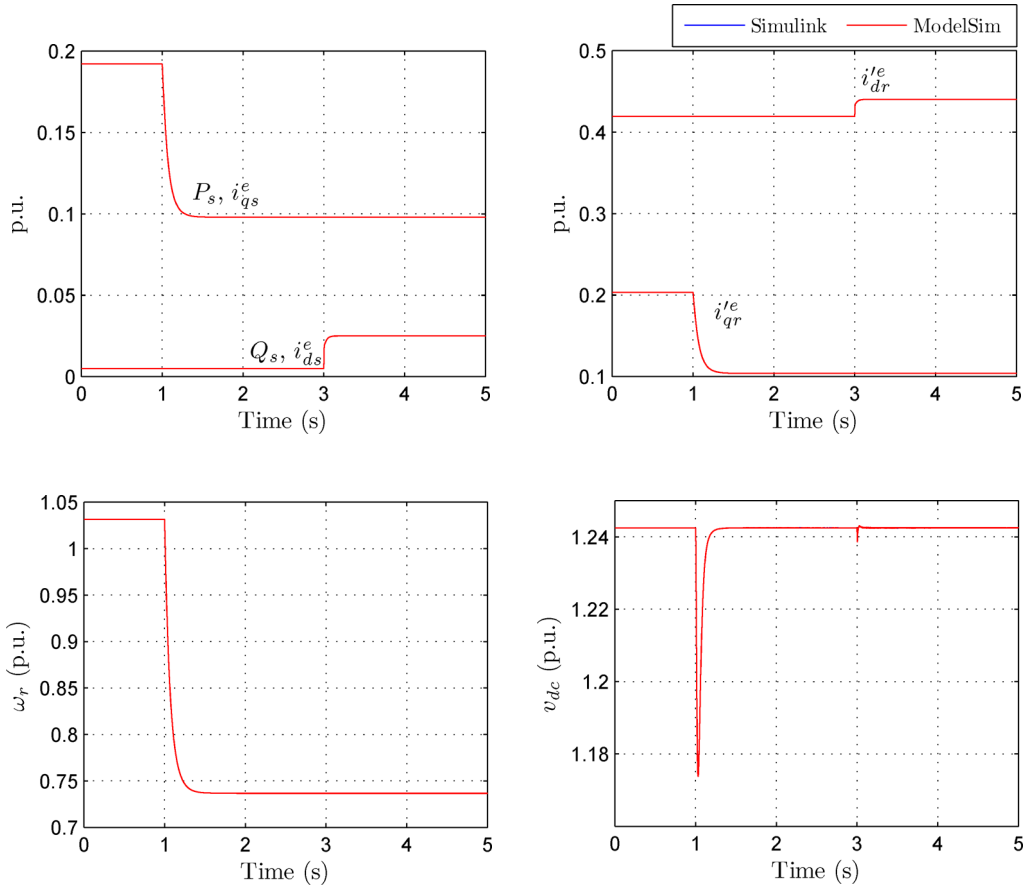

Fig. 4. Simulation results

[17] H. Chen, S. Sun, D. C. Aliprantis, and J. Zambreno, "Dynamic simulation of electric machines on FPGA boards," in *IEEE Int. Electr. Mach. and Drives Conf.*, May 2009, pp. 1842–1847.

[18] E. Jamro and K. Wiatr, "FPGA implementation of 64-bit exponential function for HPC," in *IEEE Int. Conf. Field Programmable Logic and Applications*, Aug. 2007, pp. 718–721.

[19] P. P. Chu, *RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability*. Hoboken, New Jersey: Wiley-Interscience, 2006.

[20] J. Cavanagh, *Verilog HDL: Digital Design and Modeling*. Boca Raton, Florida: CRC Press, 2007.

Hao Chen received the B.E. and M.S. degree in Electrical Engineering from Xi'an Jiaotong University, China, in 2002 and 2005. He is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. His research interests include power electronics, electric machine drives, and wind energy conversion.

Song Sun is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. His research interests include designing efficient hardware architecture for data mining applications.

Dionysios C. Aliprantis (SM'09) received the Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 1999, and the Ph.D. from Purdue University, West Lafayette, IN, in 2003. He is currently a Litton Industries assistant professor of Electrical and Computer Engineering at Iowa State University, Ames, IA, USA. His interests include the modeling and simulation of electric machines and power systems, power electronics and controls, and renewable energy applications.

Joseph Zambreno is an assistant professor of Electrical and Computer Engineering at Iowa State University. His research interests include computer architecture and compilers, the use of reconfigurable computing as a general enabling technology, and the use of design automation to address various aspects of security and trust. He has a BS, MS, and PhD in electrical and computer engineering from Northwestern University.