FPGA-based Design and Implementation of Direct Torque Control for Induction Machines

Mohammad A. Zare*, Rajesh G. Kavasseri*, Cristinel Ababei†,
*Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND.
{rajesh.kavasseri}@ndsu.edu
†Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI.
{cristinel.ababei}@marquette.edu

***Abstract*—We present a field programmable gate array (FPGA) based implementation for direct torque control (DTC) of induction motor drives. The proposed design utilizes several improvements to execute the functional blocks in DTC that reduce the execution time and improve the sampling frequency. The FPGA system is implemented on a Xilinx Virtex-5 board using VHDL code assembled from scratch and the DSP based solution is implemented using dSPACE DS 1104. Both systems are validated experimentally with hardware-in-the-loop, on a small 200 W, 3 phase induction machine. Experimental results indicate that the proposed design enables a far higher sampling frequency (up to 800 kHz), compared to typical digital signal processors (DSP) based solutions which are limited to 20kHz. The higher sampling frequency helps mitigate torque ripple which is a well known limitation of DTC. Additionally, the short execution times suggest the possibility of extending the use of such FPGA implementations to serve auxiliary motor diagnostic functions.**

***Index Terms*—FPGA, dSPACE, Direct Torque Control (DTC), Induction Motor.**

I. Introduction

The demand for high performance electric drives is steadily growing with the increasing emphasis for electrification of the transportation industry. For induction motors, direct torque control (DTC) [1], [2] and Field Oriented Control (FOC) [3] are two popular torque control methods and a relative comparison of their performance is presented in [4]. While FOC is considered industrially mature but more complex, DTC, which is relatively recent, is rapidly becoming a popular strategy because of its simplicity (no need for mechanical shaft mounted torque transducers) and good dynamic response. However, minimizing the high torque ripple that is inherent in DTC schemes continues to pose a significant challenge to Digital Signal Processor (DSP) based implementations. Because DTC uses hysteresis torque and stator flux comparators, the ripple needs to be contained within the hysteresis bands which calls for higher sampling frequencies. Current DSP platforms such as dSPACE DS1104 or TMS C2000 can support sampling frequencies upto 20 kHz. However, with this resolution, the hysteresis controllers cannot be still driven to reach the performance provided by its analog counterparts. These limitations have motivated FPGA-based implementations where the parallelism can be exploited to achieve shorter execution times that can be beneficial in high performance applications. FPGA applications to achieve drastically shorter simulation times of motor control algorithms have been proposed in [5]–[8]. FPGAs have also been proposed for hardware-in-the-loop (HIL) simulation of power electronic converters - which exhibit very fast switching dynamics. In such cases, FPGAs can provide the latencies needed for real-time simulation of fast switching transistors. FPGAs have been used for modeling and simulation [9]–[12] of power electronic components. For actual implementation applications of motor control algorithms, as pointed in [13] (and shown in in Fig. 1), timing constraints and algorithm complexity are the two main factors governing the choice - DSP or FPGA. Implementing

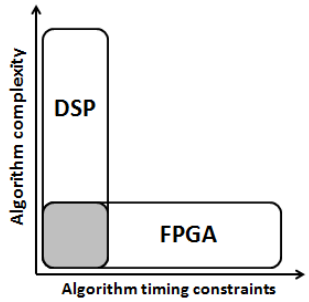

Fig. 1. DSP and FPGA domain of use based on algorithm complexity and timing constraints, [13]

DTC with FPGAs introduces challenges because the algorithm involves several complex arithmetic operations. For example, estimating the torque and stator flux involves numerical integration and square root evaluations and the stator flux vector (magnitude and angle) needs to be computed. Thus, finding clever implementations of these functions is the major research focus for FPGA based implementations. The work in [14] suggests improved methods for torque and stator flux estimators, sector selectors, and the data format, with the overall design goal of maximizing the sampling frequency and while minimizing hardware resource utilization and errors. The results in [14] achieve a sampling frequency of 200kHz.

This paper presents a hardware implementation of DTC on a 200W, 3 phase, 4 pole induction motor with FPGAs (on a Xilinx Virtex-5 board) and the DSP counterpart (on a dSPACE DS1104 platform). Specifically we use the following design features: (i) We employ a look-up table to implement the sector selection block. This requires only comparison operators and while similar to [14], our implementation is achieved in 4 clock cycles and thus it is much simpler than

978-1-4799-5944-0/14/$31.00 ©2014 IEEE

evaluating the *arctan* function with the CORDIC algorithm ([15]) (ii) Custom hardware is designed for signal conditioning, A/D conversion and generating the gating signals driving the inverter, (iii) Operating the hysteresis comparators and sector selector in parallel which requires 7 clock cycles for execution. Experimental results show that the control loop is executed in 66 clock cycles at a maximum clock frequency of 54MHz. This implies that the maximum inverter switching frequency can be potentially raised up to approximately 800kHz; compared to 20kHz achieved with DSP implementation and 200 kHz achieved in [14]. This feature of the FPGA-based implementation is immensely beneficial because as explained earlier, reducing the torque ripple requires a high sampling frequency considering the bandwidths of the hysteresis comparators.

The rest of the paper is organized as follows. In section II, the DTC scheme is briefly explained. Sections III and IV describe the dSPACE and FPGA based system components and design respectively. In section V, experimental results of both dSPACE and FPGA-based systems are presented and compared. Section VI concludes this paper.

II. Description of the Direct Torque Control Scheme

The DTC scheme as shown in Fig. 2 contains the following main components: 3-phase induction motor, 3-phase voltage source inverter, estimation block, switching table, torque and stator flux comparators and a sector selector. These blocks are described in the following subsections. It should be noted that except the induction motor and inverter which appear as physical hardware-in-the-loop, the rest of the blocks are implemented digitally.

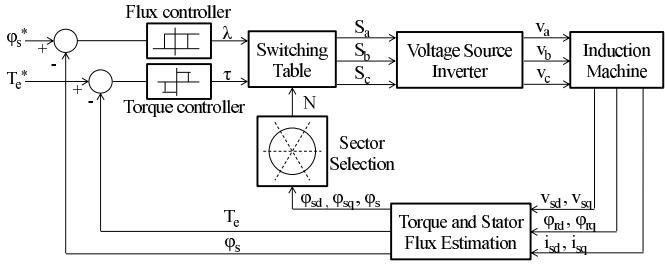

Fig. 2. Block diagram of Direct Torque Control (DTC).

A. Induction Machine

The 3-phase induction motor is modeled by the following differential equations in the synchronous reference frame using dq components.

$$v_{sq} = R_s i_{sq} + \frac{d\phi_{sq}}{dt} + \omega_e \phi_{sd} \tag{1}$$

$$v_{sd} = R_s i_{sd} + \frac{d\phi_{sd}}{dt} - \omega_e \phi_{sq} \tag{2}$$

$$0 = R_r i_{rq} + \frac{d\phi_{rq}}{dt} + (\omega_e - \omega_r)\phi_{rd} \tag{3}$$

$$0 = R_r i_{rd} + \frac{d\phi_{rd}}{dt} - (\omega_e - \omega_r)\phi_{rq} \tag{4}$$

where ω_e is the stator angular electrical frequency; ω_r is the rotor angular electrical frequency; R_s and R_r are stator and rotor resistances; i_{sd} and i_{sq} are the stator current dq components; ϕ_{sd}, ϕ_{sq}, ϕ_{rd} and ϕ_{rq} are the dq axes stator and rotor flux components, respectively. The flux variables are related to the machine currents and inductances as:

$$\phi_{sq} = L_s i_{sq} + L_m i_{rq},\ \phi_{sd} = L_s i_{sd} + L_m i_{rd} \tag{5}$$

$$\phi_{rq} = L_r i_{rq} + L_m i_{sq},\ \phi_{rd} = L_r i_{rd} + L_m i_{sd} \tag{6}$$

where L_s and L_r are the stator and rotor inductances, and L_m is the magnetizing inductance.

The 3-phase voltages of the stator, transformed to dq components are obtained from the positive sequence dq transformation matrix:

$$\begin{bmatrix} v_{sd} \\ v_{sq} \\ v_{s0} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -0.5 & -0.5 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} \tag{7}$$

The electromechanical (intertial) dynamics are given by:

$$\frac{d\omega_m}{dt} = \frac{1}{J}(T_e - T_L) \tag{8}$$

where ω_m is the angular mechanical speed, T_e is the electromechanical torque, T_L is the load torque and J denotes the machine inertia.

B. Estimation Block

The torque and stator flux estimation blocks (see Fig. 2) estimate the motor electromechanical torque T_e and stator flux linkage ϕ_s described by the following equations.

$$T_e = \frac{3}{2}\frac{P}{2}(\phi_{sd} i_{sq} - \phi_{sq} i_{sd}) \tag{9}$$

$$\phi_{sd} = \int (v_{sd} - R_s i_{sd})dt,\ \phi_{sq} = \int (v_{sq} - R_s i_{sq})dt \tag{10}$$

$$\phi_s = \sqrt{\phi_{sd}^2 + \phi_{sq}^2} \tag{11}$$

The evaluation of ϕ_{sd} and ϕ_{sq} from equation (10) is done numerically by:

$$\phi_{sd} = \phi_{sd_{old}} + T_s(v_{sd} - R_s i_{sd}) \tag{12}$$

$$\phi_{sq} = \phi_{sq_{old}} + T_s(v_{sq} - R_s i_{sq}) \tag{13}$$

where T_s is the integration time step.

C. Torque and Flux Comparators

The values estimated by the estimation block are compared with the torque and stator flux command values using three and two-level hysteresis torque and flux comparators, respectively, as shown in Fig.2. The output of the comparators are used along with the sector selector block output to determine the proper switching signals that should be applied to the inverter.

D. Sector Selector

The dq coordinate plane is divided into 6 sectors. This block determines in what sector number the stator flux vector is located at a given sampling instant. This is done by comparing the flux magnitude (ϕ_s) and the flux dq components (ϕ_{sd}, ϕ_{sq}).

E. Switching Table

The inverter switching signals are decided based on the switching table shown in Table I. The table consists of zero and non-zero (active) vectors. Zero vectors are (0,0,0) and (1,1,1) that stop the field vector, reducing the torque as the result. On the other hand, all other six vectors known as active vectors which advance the field forward resulting in torque increase [16].

TABLE I
SWITCHING TABLE FOR DTC.

(S_a, S_b, S_c)							
ϕ, τ, N		N = 1	N = 2	N = 3	N = 4	N = 5	N = 6
ϕ = 1	τ = 1	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)
	τ = 0	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)
	τ = -1	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)
ϕ = 0	τ = 1	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)
	τ = 0	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)
	τ = -1	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)

F. Inverter

A 3-phase inverter is used to provide the required voltage for the stator of the induction machine. The switching signals applied to the inverter are determined based on the estimated torque, estimated stator flux and the stator flux vector sextant number.

III. DSPACE-BASED IMPLEMENTATION DESIGN OF DTC

The hardware block diagram of dSPACE-based (DS 1104) DTC is shown in Fig. 3. The system includes induction motor, dSPACE interface board, 3-phase inverter and two hall effect current sensors. The interface is connected to a desktop computer which controls the entire system and also collects data through ADC and DAC converters.

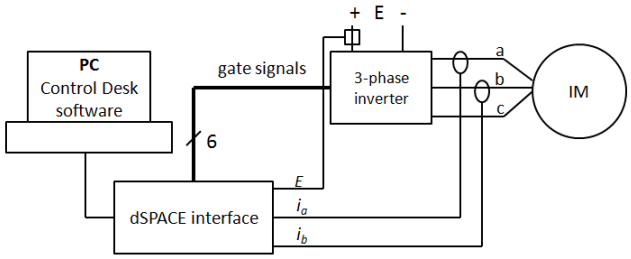

Fig. 3. Block diagram for hardware implementation of dSPACE-based DTC scheme.

The dSPACE-based parameter estimation system is implemeted in SIMULINK models that are compiled to C code using Real-Time Interface (RTI). This code is executed by the dSPACE processor.

IV. FPGA-BASED DTC DESIGN

In this section, we present the specific design measures untertaken to further improve the FPGA-based DTC design, hardware units and the experimental setup.

A. FPGA design for DTC

Fig. 4 shows the outputs and inputs of FPGA when it is used as the processing unit for DTC. The FPGA receives two phase currents (i_a, i_b), the DC link voltage (E) and three switching signals (S_a, S_b, S_c). Based on these inputs, it estimates stator flux (ϕ_s) and torque (T_e) values and outputs the inverter switching signals.

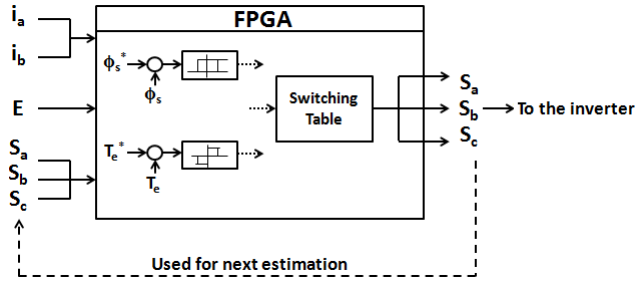

Fig. 4. Inputs and outputs of FPGA as the digital signal processor for DTC.

The non-integer values present in FPGA calculations are represented by 32-bit single precision floating point numbers ($IEEE-754$). This provides adequate dynamic range and sufficient precision. On the other hand, this is not the most economical choice from a hardware resource requirements perspective. It should also be noted that this choice requires floating point operators including adders and multipliers. In comparison, [17] uses variable length fixed-point while [14] uses two's complement fixed point format. In implementing the torque and stator flux estimator on FPGA, there are challenges related to implementing the required functions - mainly the square root function. To handle such operations, several Xilinx-provided floating point compatible IP cores namely: *Square Root*, *Greater than or equal*, *Smaller than or equal* were used. The last two cores were specifically used for implementation of the two and three-level hysteresis operators. Note that our variables are defined by single precision floating point format, therefore we can not use simple *if-statements* to realize these comparisons. The IP core for square root is used to calculate stator flux magnitude (ϕ_s) based on (11). Also to convert the motor phase current values from fixed point to floating point format a *Block Memory Generator* IP core was used as a look-up table.

The sector selector block is also implemented using a look-up table. The output of the look-up table is determined based on comparison between ϕ_{sd}, ϕ_{sq}, $\frac{\sqrt{3}}{2}\phi_s$, and 0 which is similar to [14], [18]. However, as we show later,our design executes this block in 4 clock cycles. Moreover, this method is significantly simpler compared to other FPGA-based implementations for DTC which use the *arctan* of the angle or CORDIC algorithm [15], [13].

We used 6 look-up tables to implement the switching table block. A look-up table is associated with each of the 6 sectors in the flux vector plane. Therefore the output of the switching table block is determined by the output of the sector selector block and the 3 bits provided by the torque (2 bits) and flux (1 bit) comparators.

B. FPGA-based hardware setup design

The hardware block diagram of FPGA-based DTC is shown in Fig. 5.

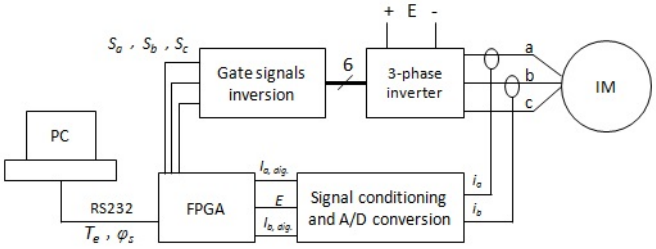

Fig. 5. Block diagram for hardware implementation of FPGA-based DTC scheme.

Apart from the basic elements, some other hardware units were needed to accommodate the signals present in the setup. These blocks are explained next.

1) Signal Conditioning and Analog to Digital Conversion: The signals coming from the Hall effect current sensors are analog signals that need to be applied to the FPGA board which is a digital device. Therefore analog to digital conversion is needed. To maximize the conversion resolution, proper signal conditioning is essential. Considering the sensor current to voltage ratio, we condition (amplify and shift) the current signal to match the input voltage range of the ADC (typically 5 V). A circuit to achieve this is shown in Fig. 6. The first op-amp circuit is a non-inverting amplifier with a gain of 25 and the second one is a differential amplifier with a unity gain and acts as a voltage shifter by 2.5V. Considering the maximum instantaneous phase current to be 10A and resistance of the current sensors to be 10mΩ, this circuit makes the output change from 0 to 5V as the motor phase current changes form -10A to +10A. An 8-bit ADC ($MAX150$) with a maximum sampling frequency of 500 ksps is used to provide the required resolution while enjoying the high speed processing merits provided by the FPGA in our DTC scheme.

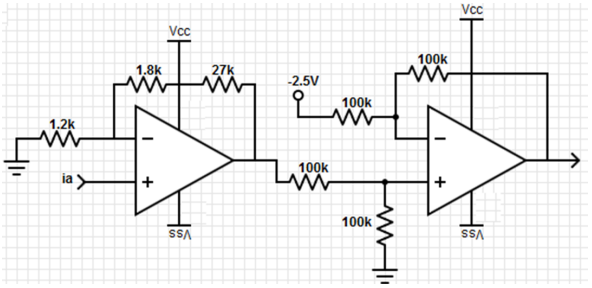

Fig. 6. Op-amp amplifier and voltage shifter to accommodate current sensor output for the ADC.

2) Gate signals inversion: The FPGA outputs 3 bits of switching signal commands for the three phase legs of the inverter. Each bit needs to be inverted to be applied to the low-side switch of each phase leg as a complementary signal. This is done with the circuit shown in Fig. 7 consisting of two MOSFETs and one buffer. At the same time, this circuit changes the CMOS voltage level of 3.3V to 5V considering that the original FPGA voltage may not be high enough to drive the inverter switches.

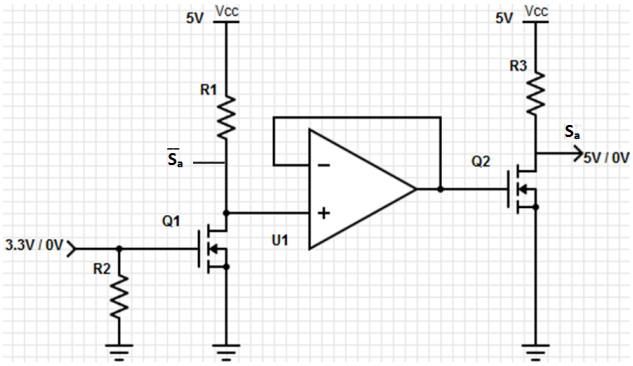

Fig. 7. FPGA switching signal output voltage inverting circuit to be applied to the inverter.

V. Experimental Results

In this section, experimental results of both dSPACE-based and FPGA-based DTC are presented and compared. We implemented the DTC scheme on dSPACE DS 1104 [19] as well as Xilinx Virtex-5 FPGA [20]. The parameters of the 200W, 3-phase, 4-pole induction machine used in our experiments are shown in Table II. The induction motor is coupled with a DC generator with a resistive load to provide variable torque on the motor shaft.

TABLE II
Parameters of the Induction Motor

R_s	0.17 Ω
R_r	0.17 Ω
L_s	6.02 mH
L_r	6.03 mH
L_m	5.33 mH
J	0.000225 kg.m^2
P	4

The experimental setup of dSPACE-based DTC is shown in Fig. 8.

The stator flux reference value is 0.04Wb and the torque reference value is considered to be a step command of 0.5Nm to -0.5Nm. The response waveforms of the setup for torque (T_e), stator flux dq components (ϕ_{sd}, ϕ_{sq}) and the motor phase a and b currents (i_a, i_b) are shown in Fig. 9, Fig. 10 and Fig. 11, respectively.

The stator flux (ϕ_s) waveform for the torque command is shown in Fig. 12. The flux value maintains its reference value of 0.04Wb as expected.

The experimental setup of FPGA-based DTC is shown in Fig 13. The ADCs are $MAX150$.

We used a global counter register to control and synchronize different blocks of the system. The design flow is shown in Fig. 14. Whenever the output of each block becomes valid, it is sampled and applied to the next block at a certain counter

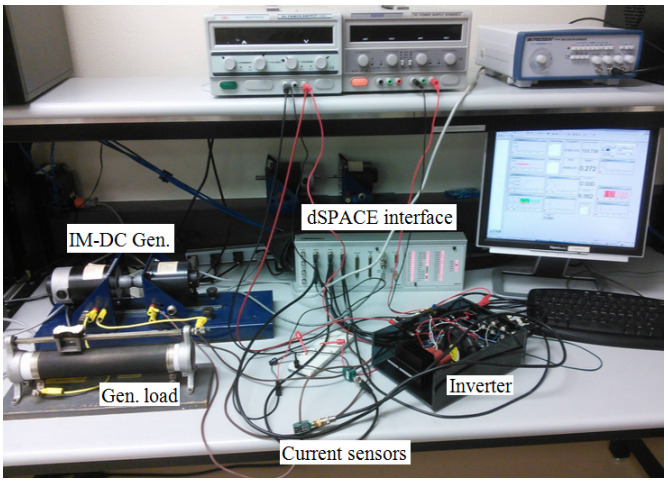

Fig. 8. Experimental setup for dSPACE-based DTC scheme.

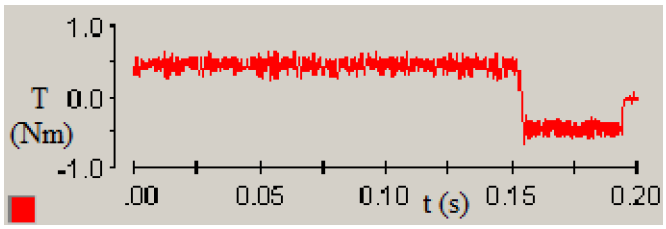

Fig. 9. Induction motor estimated torque for a step torque command of 0.5Nm to -0.5Nm.

number. The current interface block needs 7 clock cycles to concatenate parallel bits coming from the ADC and make an 8-bit word. Then using a Memory Block Generator IP core working as a 256-line look-up table, the single precision word is determined and finally the configured value of the current is calculated using a floating point multiplier and adder. The estimation block calculates stator flux and torque values in 48 clock cycles. The hysteresis comparators and and sector selector block work in parallel and need 7 clock cycles to output a valid value. Finally the switching table that decides about the inverter gate signals needs only 4 clock cycles.

Table III shows the implementation results from Xilinx ISE synthesis report. In our implementation, the design occupies 91% of the available slice LUTs and 12% of available slice registers on Xilinx Virtex-5. The designer should consider

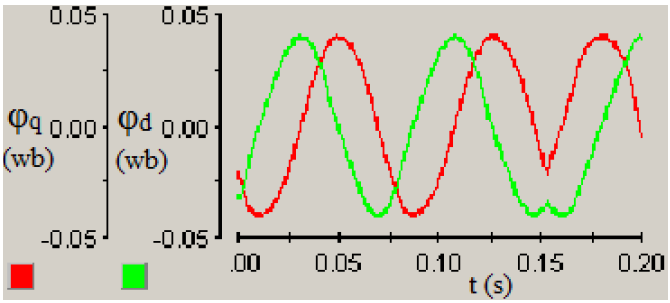

Fig. 10. Induction motor dq flux linkages for a step torque command of 0.5Nm to -0.5Nm.

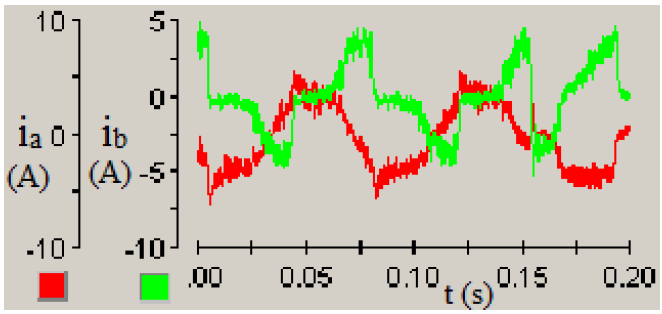

Fig. 11. Induction motor phase currents for a step torque command of 0.5Nm to -0.5Nm.

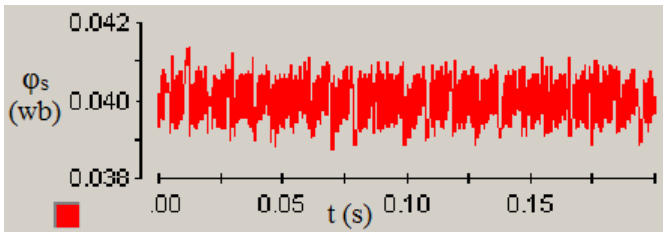

Fig. 12. Induction motor estimated stator flux for a step torque command of 0.5Nm to -0.5Nm.

optimization techniques to reduce the used hardware resources as much as possible.

TABLE III
IMPLEMENTATION RESULTS OF THE DTC SCHEME ON XILINX VIRTEX-5 FPGA

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2502	28800	8%
Number of Slice LUTs	21758	28800	75%
Maximum Clock Frequency	54 MHz		

As stated earlier, FPGAs can handle calculations in shorter time intervals compared to DSP (dSPACE) by virtue of its inherent parallelism. Based on Table III, in our FPGA-based im-

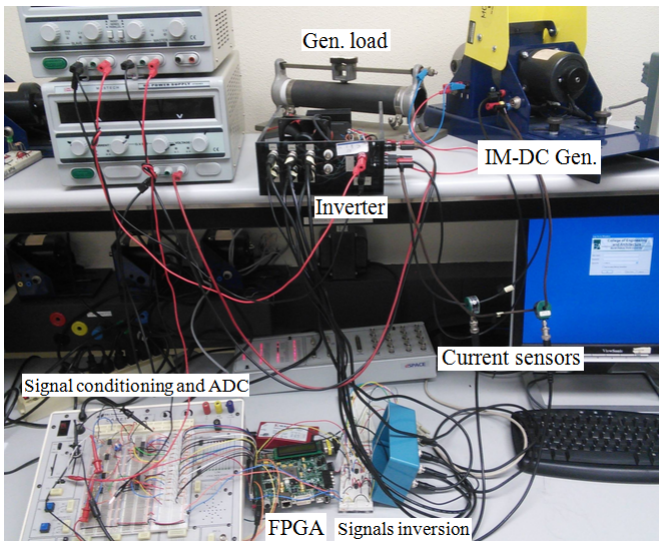

Fig. 13. Experimental setup for FPGA-based DTC scheme.

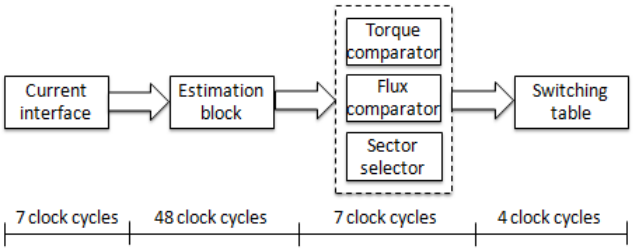

Fig. 14. Number of clock cycles required by the building blocks for the proposed DTC estimator. It takes 66 clock cycles for each iteration of the feedback loop.

plementation, the maximum clock frequency is about 54MHz. Executing one loop of the calculation requires 66 clock cycles. This suggests that the inverter switching frequency can be potentially up to about 800KHz. However, one should note that the switching frequency is also limited by the ADCs and the switching devices used in inverters. Overall, this improvement leading enabling higher sampling frequencies can dramatically reduce the torque ripple - which is one of the main limiting factors for DTC.

VI. Conclusion

The inherent parallelism in FPGAs can be exploited in the design of high performance controllers for electric drives where execution time is critical. This paper presents the design and hardware implementation of one such controller - namely, Direct Torque Control (DTC) for induction motor drives. The implementations are realized on both the FPGA and DSP (using dSPACE) platforms. The proposed FPGA design uses a look-up table to implement the sector selection block which is operated in parallel with the hysteresis comparators. The estimation block which requires numerical integration and square-root function evaluations takes the longest time (48 cycles) for execution compared to the other functional blocks. These design improvements reduce the overall execution time to 66 clock cycles. While the dSPACE-based system has a sampling frequency limit of 20kHz imposed by the system ADCs, the proposed FPGA-based design allows a potential maximum sampling frequency of 800kHz. This has significant implications because higher sampling frequencies in the hysteresis controllers reduce the torque ripple - which continues to be a challenge for DTC. The design choice of 32-bit floating point representations provides sufficient accuracy while increasing hardware utilization. These benefits, along with short execution times suggests that FPGAs primarily employed to implement the control can additionally be utlilized for auxiliary tasks such as fault analysis and diagnosis, and closely monitor motor operation. The development of such add-ons will be the subject of our future work.

VII. Acknowledgement

Support for this work in part from John Deere, Inc and the Department of Electrical and Computer Engineering at North Dakota State University, Fargo is gratefully acknowledged.

References

[1] I. Takahashi and T. Noguchi, "A new quick-response and high efficiency control strategy of an induction motor," *IEEE Trans. on Industry Application*, vol. 22, no. 5, pp. 820827, Sept./Oct. 1986.
[2] M. Depenbrock, "Direct Self Control (DSC) of Inverter Fed Induction Machine," *IEEE Trans. on Power Electronics*, vol. 3, no.4, pp.420-429, Oct. 1988.
[3] F. Blaschke, "A new method for the structural decoupling of A.C. induction machines, in. *Conf. Rec. IFAC*, Duesseldorf, Germany, Oct. 1971, pp. 115.
[4] D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: Two viable schemes for induction motors torque control," *IEEE Trans. on Power Electonics*, vol. 17, no. 5, pp. 779-787, Sept. 2002.
[5] H. Kia, M. Zare, R. Kavasseri, and C. Ababei, "Dynamic simulation of direct torque control of induction motors with FPGA based accelerators," *Int. Conference on Reconfigurable Computing and FPGAs (ReConFig)* 2013, Cancun, Mexico.
[6] H. Chen, S. Sun, D.C. Aliprantis, and J. Zambreno, "Dynamic simulation of electric machines on FPGA boards," *IEEE Int. Electric Machines and Drives Conference (IEMDC)*, 2009.
[7] H. Chen, S. Sun, D. C. Aliprantis, and J. Zambreno, "Dynamic simulation of DFIG wind turbines on FPGA boards," *Power and Energy Conference*, 2010.
[8] S. Tola and M. Sengupta, "Real-Time simulation of an induction motor in different reference frames on an FPGA platform," *IEEE Int. Conference on Power Electronics, Drives and Energy Systems*, 2012.
[9] M. Martar, M. Adbel-Rahman, and A.-M. Soliman,"FPGA based real-time digital simulation," *Int. Conference on Power Systems Transients*, 2005.
[10] G. Parma and V. Dinavahi,"Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. on Power Delivery*, vol. 22, no. 2, pp. 12351246, Apr. 2007.
[11] J. Pimentel,"Implementation of simulation algorithms in FPGA for real time simulation of electrical networks with power electronics devices," *IEEE Int. Conference on Reconfigurable Computing and FPGAs*, 2006.
[12] P. Le-Huy, S. Gu erette, L. A. Dessaint, and H. Le-Huy,"Real-simulation of power electronics in power systems using an FPGA," *Canadian Conference Electrical and Computer Engineering*, 2006.
[13] E. Monmasson and M. Cirstea, "FPGA Design Methodology for Industrial Control SystemsA Review," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1824-1842, Aug. 2007.
[14] T. Sutikno, N. Idris, A. Jidin, and M. Cirstea, "An improved FPGA implementation of direct torque control for induction machines," *IEEE Trans. on Industrial Informatics*, Oct. 2012.
[15] C. Kowalski, J. Lis, and T.O Kowalska, "FPGA implementation of DTC control method for the induction motor drive," *Int. Conference on Computer as a Tool (EUROCON)*, 2007.
[16] M. P. Kazmierkowski and G. Buja, "Review of direct torque control methods for voltage source inverter-fed induction motors," *IEEE Industrial Electronics Society (IECON)*, 2003.
[17] S. Ferreira, F. Haffner, L. Pereira, and F. Moraes, "Design and prototyping of direct torque control of induction motors in FPGAs," *Proceedings of the 16th Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2003.
[18] T. Sutikno, A. Jidin, and N. Idris, "New approach FPGA-based implementation of discontinuous SVPWM," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 18, pp. 499-514, 2010.
[19] http://www.dspace.com/
[20] http://www.xilinx.com/