An Improved FPGA Implementation of Direct Torque Control for Induction Machines

Tole Sutikno, *Member, IEEE*, Nik Rumzi Nik Idris, *Senior Member, IEEE*, Auzani Jidin, *Member, IEEE*, and Marcian N. Cirstea, *Senior Member, IEEE*

***Abstract*—This paper presents a novel direct torque control (DTC) approach for induction machines, based on an improved torque and stator flux estimator and its implementation using field-programmable gate arrays (FPGA). The DTC performance is significantly improved by the use of FPGA, which can execute the DTC algorithm at higher sampling frequency. This leads to the reduction of the torque ripple and improved flux and torque estimations. The main achievements are: 1) calculating a discrete integration operation of stator flux using backward Euler approach; 2) modifying a so called nonrestoring method in calculating the complicated square root operation in stator flux estimator; 3) introducing a new flux sector determination method; 4) increasing the sampling frequency to 200 kHz such that the digital computation will perform similar to that of the analog operation; and 5) using two's complement fixed-point format approach to minimize calculation errors and the hardware resource usage in all operations. The design was achieved in VHDL, based on a MATLAB/Simulink simulation model. The Hardware-in-the-Loop method is used to verify the functionality of the FPGA estimator. The simulation results are validated experimentally. Thus, it is demonstrated that FPGA implementation of DTC drives can achieve excellent performance at high sampling frequency.**

***Index Terms*—Direct torque control (DTC), field-programmable gate arrays (FPGAs), induction machine, VHDL.**

I. Introduction

DIRECT torque control (DTC) of machine drives has gained popularity since it can provide fast instantaneous torque control with simple control structure. The original DTC scheme was proposed by Takahashi in 1986 [1] and uses hysteresis controllers to control independently both the stator flux and the torque. Ideally, the error or ripple of the torque (or flux) is restricted within the hysteresis band, so that the output torque (or flux) will satisfy its demand. However, in practice, as the hysteresis controller follows a discrete computation approach, this is impossible to achieve due to the delay between the torque sampling instant and the instant the corresponding switching status is passed to the inverter [2]. The ripple might exceed beyond the hysteresis bands and hence tends to select the reverse voltage vector that causes rapid increase/decrease of the torque [3]. Consequently, this will produce larger torque ripples and slightly degrade the performance of DTC. Several methods were proposed to minimize the output torque ripple. These include the use of space vector modulation (SVM) [4]–[7], the injection of dithering signal [8], the use of constant carrier frequency [3], and, recently, the hysteresis-based DTC with predictive control [9]–[12]. All of these methods require knowledge and modifications of machine parameters which will complicate the simple DTC structure and will increase its control sensitivity. Moreover, the same effectiveness in minimizing the output torque ripple using those methods can be achieved if a higher switching frequency is applied, with a high-speed processor.

Traditionally, the DTC algorithm is executed using a digital signal processor (DSP) [13]–[15], with code written using C programming or a graphical programming approach appropriate for rapid prototyping. It should be noted that the sampling frequency of the processor depends on the computational burden. For the basic DTC algorithm, normally the sampling frequency of the DSP (e.g., DSPACE 1104 or TMS C2000 series) can reach up to 20 kHz. However, this is still insufficient to operate the discrete hysteresis controller, similar to that of analog/continuous hysteresis system, so that the output torque ripple can be restricted within the band, even when it operates at the worst conditions (i.e., at very low speeds that cause extreme torque slope).

Some works used a combination of DSP and field-programmable gate arrays (FPGA), reducing the DSP's computational burden by distributing some DTC algorithm tasks (lookup table (LUT), blanking time generator, and hysteresis controllers) to the FPGA. Thus, the sampling period to execute the overall DTC algorithms can be minimized to reduce the output torque ripple [16]–[18]. However, the combination of controllers increases the cost and complexity of the interfacing circuit and is not a practical solution for commercialization purposes. Some attempts [19], [20] implemented entire DTC algorithms on a single FPGA but the HDL coding there was generated using third-party packages, i.e., MATLAB/Simulink, with the Xilinx System Generator Fixed-point toolbox, which is not fully optimized to achieve fast sampling frequency. In [20], a significant increment in the sampling frequency to twice of that obtained with a DSP (which is 40 kHz) is reported.

This paper presents an effective way to design, simulate, and implement the flux and torque estimations for hysteresis-based

Manuscript received December 19, 2011; revised June 14, 2012, August 01, 2012; accepted September 17, 2012. Date of publication October 03, 2012; date of current version August 16, 2013. This work was supported in part by the Universiti Teknologi Malaysia under Grant VOT 78584 and the Ministry of Higher Education of the Malaysian government. Paper no. TII-11-1037.

T. Sutikno is with the Electrical Engineering Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia, and also with Universiti Teknologi Malaysia, Johor 81310, Malaysia (email: tole@ee.uad.ac.id).

N. R. N. Idris is with the Energy Conversion Department, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia (e-mail: nikrumzi@ieee.org).

A. Jidin is with the Power Electronics and Drives Department, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia (e-mail: auzani@ieee.org).

M. Cirstea is with the Computing and Technology Department, Anglia Ruskin University, Cambridge CB1 1PT, U.K. (e-mail: marcian@ieee.org).

Digital Object Identifier 10.1109/TII.2012.2222420

DTC utilizing FPGAs. The main contribution of this paper is the development of the flux and torque estimators using an optimized VHDL code on the FPGA (i.e., from scratch), to achieve a sampling frequency of 200 kHz. With the highest sampling frequency, it is therefore possible for the torque ripple to be restricted within its hysteresis band and hence minimize the ripple by reducing the band size. Moreover, the performance of flux estimation as well as the inherent current control in DTC system can be improved. Taking this into account, the estimations in DTC are the main parts to be implemented using FPGA, as they involve complex calculations (e.g., integrals, square-root, multiplication and precise current scaling factor). The optimized VHDL code design will be based on the MATLAB simulation model, where the type of data, number of bits (resolution), sampling time, and scaling factor performed in simulation are similar to that of FPGA implementation. The estimations of stator flux and torque in the DTC of the induction machine will be presented in Section II. The equations of stator flux and torque in discrete form and sector identification will be given in Section III. Section IV will present the description of the estimations using MATLAB simulation and Modelsim Altera simulation. Finally, the simulation and experimental results are compared to verify code/design effectiveness at the highest sampling frequency.

II. Major Problem in Hysteresis-Based DTC

Despite its simplicity, the DTC based on hysteresis controller causes some major problems such as variable inverter switching frequency, high torque ripple and high sampling requirement for digital implementation [3]–[8]. These problems are briefly described as follows.

A. Variable Inverter Switching Frequency

In hysteresis-based DTC, the switching frequency of a VSI is mainly governed by the switching of the torque hysteresis comparator. The slope of the torque waveform, which directly affects the switching of the hysteresis comparator, vary with the operating conditions (rotor speed, stator and rotor fluxes, dc link voltage) [5]. This can be seen from the discrete form of the torque equation given by

$$\begin{aligned}\text{Torque slope} &= \frac{T_{e,n+1} - T_{e,n}}{\Delta t}\\ &= -T_{e,n}\left(\frac{1}{\sigma\tau_s} + \frac{1}{\sigma\tau_r}\right)\\ &\quad + \frac{3}{2}P\frac{L_m}{\sigma L_s L_r}[(v_{s,n} - j\omega_r\varphi_{s,n}) \cdot j\varphi_{r,n}]\end{aligned} \tag{1}$$

where

T_e	electromagnetic torque;
Δt	small value;
σ	total flux leakage factor;
τ_s	stator time constant;
P	number of pole pairs;
L_m	mutual inductance;
L_s	stator self-inductance;
L_r	rotor self-inductance;
τ_r	rotor time constant;
φ_s	stator flux linkage space vectors in stationary reference frame;
φ_r	rotor flux linkage space vectors in stationary reference frame;
ω_r	rotor electrical speed in rad/s;
v_s	stator voltage space vector in stationary reference frame;
J	moment of inertia.

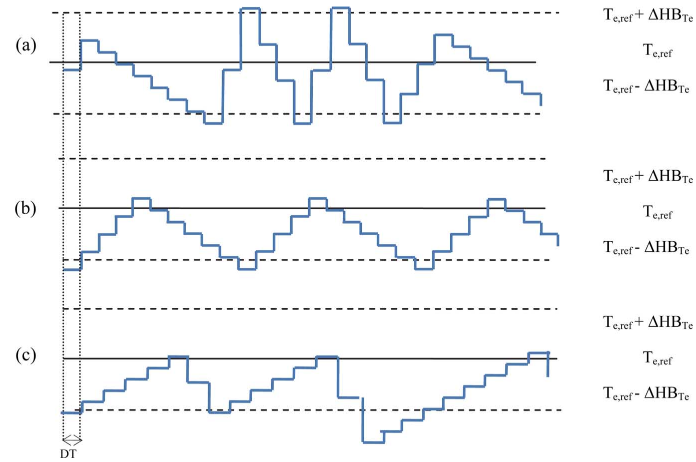

Fig. 1. Waveforms of output torque sampled at DT in the hysteresis comparator for (a) low speed, (b) middle speed, and (c) high speed.

To illustrate this, waveforms of discretized electromagnetic torque under three different steady-state operating conditions are shown in Fig. 1. These are diagrammed so that only the effects of motor speed and the applied voltage are considered. During the positive torque slope, the active voltage vector is applied; otherwise, the zero voltage vector is selected. It can be noticed that the torque slopes (for positive and negative slopes) vary with the operating speed. As a result, the torque switching frequency, and hence the VSI switching frequency, also vary with operating conditions. Thus, it is common practice to select the device with switching capability based on the worst case of operating conditions.

B. High Torque Ripple

In digital implementation, the output torque is calculated, and the appropriate switching states are determined at fixed sampling time (DT in Fig. 1). However, this causes a delay between the instant the variables are sampled and the instant in which the corresponding switching status is passed to the inverter, therefore, the torque ripple cannot be restricted exactly within the hysteresis band. If the band is set to be too small, the overshoot of the torque beyond the hysteresis band could cause a reverse active voltage vector selection, instead of a zero

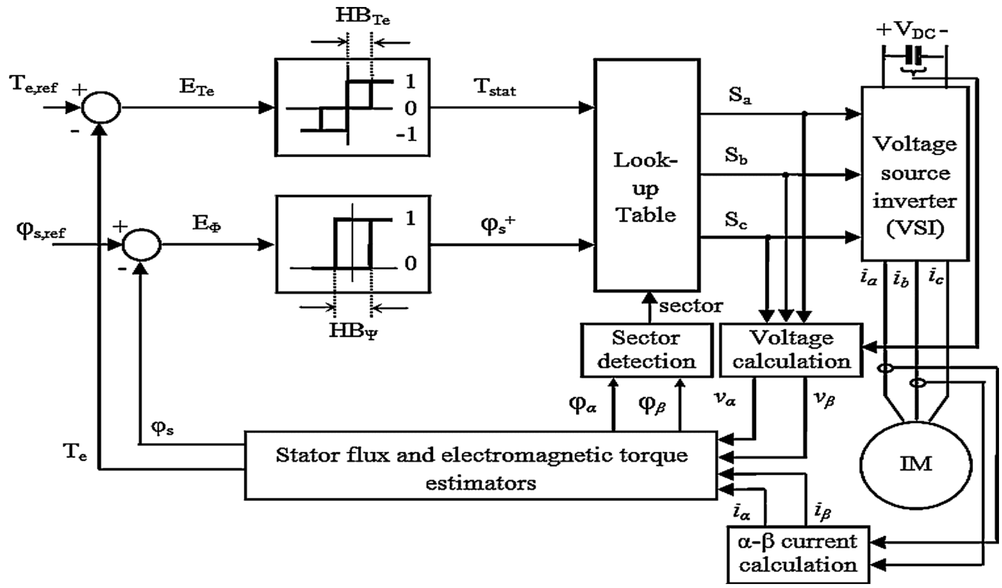

Fig. 2. Control structure of DTC-based induction machine.

voltage vector selection. The selection of the reverse voltage vector causes the torque to decrease rapidly and as a result the torque ripple increases [8], [18], [21]–[24]. This situation is illustrated in Fig. 1(a).

C. Need for a High-Speed Processor

Reducing torque ripple by lowering the bandwidth of the hysteresis comparator would be fruitless when the processor used has a limited sampling frequency. The problem of high torque ripple can be eliminated if a high-speed processor is utilized, where the discrete hysteresis controller performs closer to the operation of an analog based comparator. As shown in Fig. 1(a) and discussed in Section II-B, the rapid decrease of torque due to the selection of reverse voltage vector can be avoided if the sampling time (DT) is sufficiently reduced.

III. Proposed DTC

Fig. 2 shows a simple structure of hysteresis-based DTC by Takahashi [1]. A decoupled control of torque and flux was established to permit fast instantaneous control. The stator flux is controlled using a two-level hysteresis comparator, while the electromagnetic torque is controlled using a three-level hysteresis comparator. The outputs of the comparators, along with sector flux information, are used to index the LUT, to select the appropriate voltage vectors to control simultaneously both the stator flux and the torque. The most significant element that can guarantee a satisfactory DTC performance is the estimation of the stator flux and the torque.

In order to estimate the stator flux and the electromagnetic torque, several parameters need to be determined. The mathematical model to be used is tailored to the needs of controlled drives [25]. First, the stator currents from the motor I_a and I_b, are transformed into α–β coordinates [26], which are adequately suited to the DTC algorithm as follows:

$$I_\alpha = I_a \tag{2}$$

$$I_\beta = \frac{\sqrt{3}}{3}(I_a + 2I_b). \tag{3}$$

At the same time, by using the switching status (S_a, S_b and S_c) produced by the switching table, the stator voltages in the α–β reference frame are determined as

$$V_\alpha = \frac{V_{\mathrm{dc}}}{3}(2S_a - S_b - S_c) \tag{4}$$

$$V_\beta = \frac{\sqrt{3}}{3}V_{\mathrm{dc}}(S_b - S_c) \tag{5}$$

Then, using the calculated $I_\alpha, I_\beta, V_\alpha$ and V_β, the estimation of the stator flux in α–β coordinates is performed as follows:

$$\varphi_\alpha = \varphi_{\alpha_{\mathrm{old}}} + (V_\alpha - R_S I_\alpha)T_s \tag{6}$$

$$\varphi_\beta = \varphi_{\beta_{\mathrm{old}}} + (V_\beta - R_S I_\beta)T_s. \tag{7}$$

Finally, the equation

$$\varphi_s = \sqrt{\varphi_\alpha^2 + \varphi_\beta^2} \tag{8}$$

calculates the flux magnitude by using a square root calculation, whereas the electromagnetic torque is estimated through

$$T_e = \frac{3}{4}P(I_\beta \varphi_\alpha - I_\alpha \varphi_\beta). \tag{9}$$

The original scheme is based on hysteresis controllers, where the output status from the controllers, together with the sector flux information, are used to select the optimized voltage vectors from the LUT to satisfy simultaneously both flux and torque references. The flux vector is controlled to form a circular flux shape.

IV. Design of Torque and Stator Flux Estimator

A. Proposed Method to Improve Torque and Stator Flux Estimator

This paper presents an improved FPGA-based torque and stator flux estimator for DTC induction motor drives, which permits very fast calculations. The improvements are performed by: 1) calculation of the discrete integration operation of the stator flux using backward Euler approach; 2) reducing the sampling time down to 5 μs—to avoid saturation due to dc offset present in the sensed currents, the low-pass filter (LPF) is applied; 3) modifying the nonrestoring method to calculate complicated square root operation of the stator flux; and 4) introducing a new method to determine the sector. In all operations of FPGA implementation, the two's complement fixed-point format approach is used in order to minimize calculation errors and the hardware resources usage.

1) Fixed-Point Arithmetic: A fixed-point variable consists of a binary pattern which is encoded in two's complement number, and a binary point. It is a way to encode negative numbers into ordinary binary. The size of the binary pattern and the location of the binary point are specified using three parameters, namely: sign bit, integer word length (IWL) and fraction word length (FWL). The total number of binary pattern bits is well-known as word length (WL). The approach can represent numbers in the range $[-2^{\mathrm{IWL}}, 2^{\mathrm{IWL}}]$ with a step size of $2^{-\mathrm{FWL}}$. When using this arithmetic, the most important aspect is always to consider the binary point location for every variable. VHDL has supported the fixed-point arithmetic operations, and designers have some manipulation flexibility to improve performance.

2) Backward Euler Approach: The discrete backward Euler formula is $y(n) = y(n-1) + k.T.u(n)$. This is simpler for FPGA hardware implementation, compared with the forward Euler and Trapezoidal method in that they require the register to store the previous value of $u(n-1)$ function. The backward Euler integration method can also maintain the system stability in the large step size. Therefore, the discrete backward Euler integration method is chosen to calculate the quadrature flux (φ_α and φ_β).

3) LP Filter: Notice that R_s is the estimated stator resistance, while T_s is the implementation sampling time. The works [27], [28] suggested that a low-pass filter should be added to the integrator in the practical implementation to avoid integration drift problem due to the dc offset in the sensed currents. The stator flux equations are

$$\varphi_\alpha = (\varphi_{\alpha_{\mathrm{old}}} + (V_\alpha - R_S I_\alpha)T_s)\,(1 - \omega_c * T_s) \qquad (10)$$

$$\varphi_\beta = (\varphi_{\beta_{\mathrm{old}}} + (V_\beta - R_S I_\beta)T_s)(1 - \omega_c * T_s) \qquad (11)$$

where ω_c is the cutoff frequency of the filter. For this implementation, the cutoff frequency is chosen as 5 rad/s.

4) Nonrestoring Square Root Algorithm: In DTC drives, the stator flux (φ_s) is calculated as square root of the quadrature flux magnitude. To calculate the stator flux (φ_s), the non-restoring square root algorithm, proposed by [29], is modified as below ($D = \mathrm{radicand}, q = \mathrm{quotient}, r = \mathrm{remainder}$, and $n = \mathrm{half\ of\ the\ radicand\ word\ size}$).

TABLE I
Karnaugh Map of the Proposed Simpler Identification of the Sectors

INPUT			OUTPUT
$\varphi_\alpha > 0$	$\varphi_\alpha > \sqrt{3}\varphi_\beta$	$\varphi_\alpha > -\sqrt{3}\varphi_\beta$	(sector)
0	0	0	101
0	1	0	110
0	0	1	100
0	1	1	ddd
1	0	0	ddd
1	**1**	**0**	**001**
1	0	1	011
1	1	1	010

$r_0 = -1$ ($n/2 + 2$ bits)

$q_0 = 0$ ($n/2 + 1$ bits)

For $i = 0$ to $n - 1$ do:

If $r_i \geq 0$ then

$$r_{i+1} = (4r_i + D_{(n-2i)-1}D_{(n-2i)-2}) - (4qi + 1)$$

else

$$r_{i+1} = (4r_i + D_{(n-2i)-1}D_{(n-2i)-2}) + (4qi + 3)$$

If $r_{i+1} \geq 0$

$$q_{i+1} = 2q_i + 1$$

else

$$q_{i+1} = 2q_i$$

The square root result is $q_n(n/2 - 1\ downto\ 0)$, coded in $n/2$ bits.

5) New Sector Identification: The present work introduce a simple method to determine the sectors of the flux vector, based on a comparison between $\varphi_\beta, \sqrt{3}\varphi_\alpha, -\sqrt{3}\varphi_\alpha$ and 0, which is modified from [30]. With the comparison, it is simpler to determine the sector of the voltage vector, compared to the conventional methods of using arc tan of angle, three stages comparison based on $\varphi_\alpha, \varphi_\beta$ or determination of angle using CORDIC algorithm [31].

Table I shows the Karnaugh map of the proposed sector identification. Through the simplification, it will be possible to get simpler logic of the sector analysis for FPGA implementation through VHDL gate level coding; each sector is represented on 3 bits.

B. Design Flow

The validation of the designed torque and flux estimators was performed by using the Hardware-in-the-Loop (HiL) simulation. The DTC MATLAB/Simulink model is simulated and then, the same data I_a, I_b, S_a, S_b, and S_c obtained from the simulation, are copied from the MATLAB workspace to VHDL codes, along with the inputs for the targeted FPGA. The VHDL codes are simulated in ModelSim-Altera before being synthesized and implemented in FPGA.

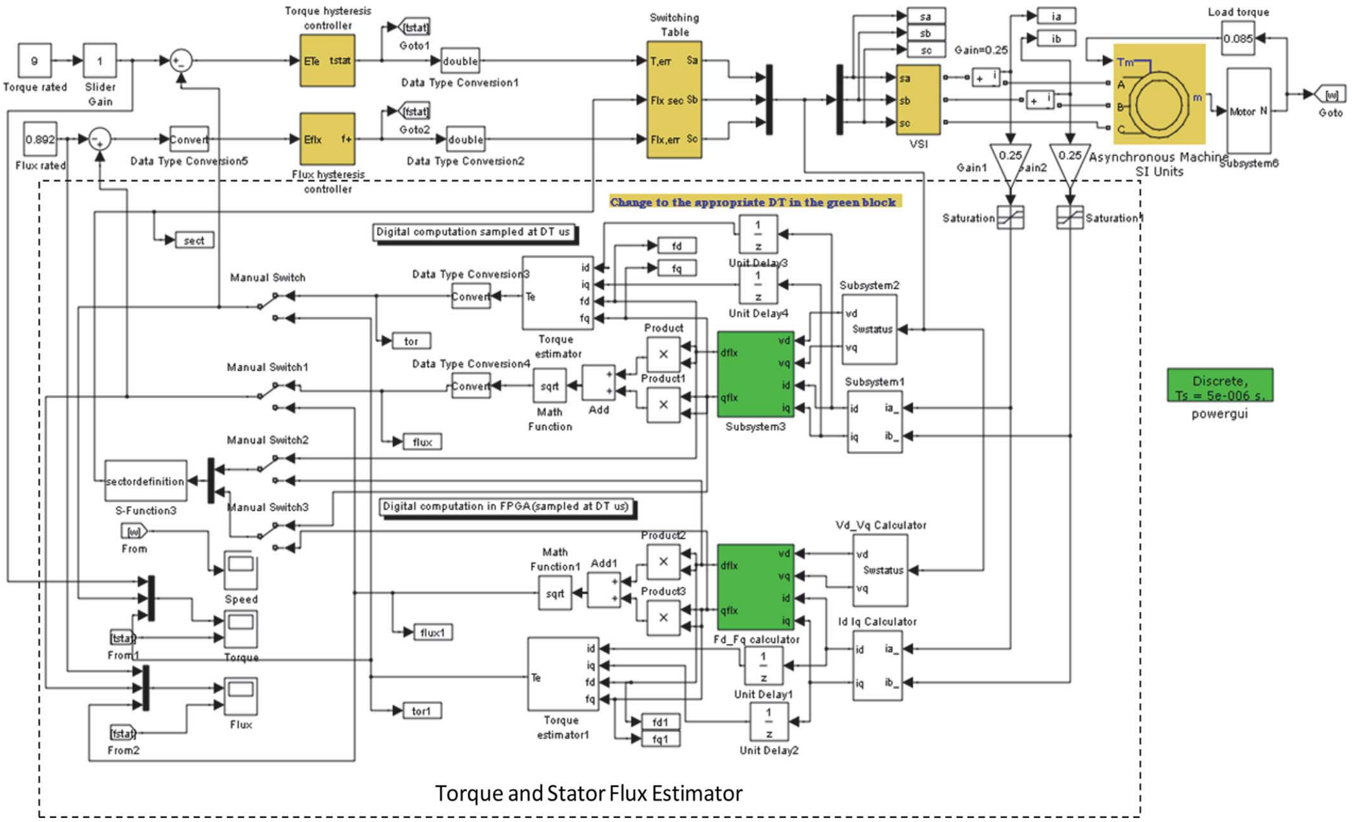

Fig. 3. Control structure of DTC-based induction machine.

V. MATLAB AND MODELSIM-ALTERA SIMULATIONS

In order to verify the torque and stator flux estimator models, a comprehensive DTC simulation is conducted. in MATLAB/Simulink (Fig. 3). The upper model is a standard model (which is not ready yet to be implemented in FPGA), and the lower model is generated as one ready to be implemented in FPGA. The simulations of the DTC model, which perform double-precision calculations, are used as references to digital computations executed in FPGA implementation.

The standard Simulink models are not ready as direct FPGA design input, the designer must prepare them as the FPGA programming will be conducted in two's complement. In principle, the procedure is similar with the one in [19], [32], [33], which is aimed to use minimum number of operators that process a maximum number of operations.

The DTC model is simulated in MATLAB/Simulink and then the same data (I_a, I_b, S_a, S_b and S_c) obtained from the simulation was copied from the MATLAB workspace to VHDL code, as well as the inputs for the targeted FPGA. The VHDL codes were simulated in ModelSim-Altera before being synthesized and implemented in FPGA. However, the stage is optional. From MATLAB simulation, the designers can go the to FPGA implementation stage, without using ModelSim-Altera simulation stage. Quartus simulation environment can be used to verify the design.

VI. FPGA IMPLEMENTATION OF THE TORQUE AND FLUX ESTIMATORS

The algorithm of torque and flux estimation is implemented in an architecture consisting of six main blocks, as shown in Fig. 4. This architecture has six inputs: two 21-b currents (I_a and I_b), 12-b high-voltage dc-supply (V_{dc}) and three switching statuses S_a, S_b, and S_c. At the end, it produces three outputs: the estimation values of torque (T_e), flux (φ_s), and sector. The sampling time chosen is 5 μs, which is limited by the ADC used.

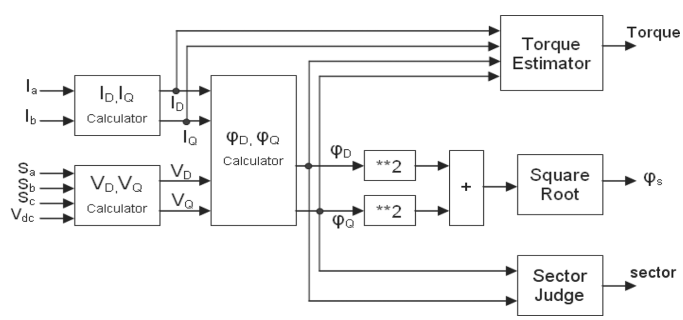

Fig. 4. Block diagram of torque and flux estimators.

A. Architecture of Torque and Flux Estimator

All of the equations modeling the motor's behavior are implemented in a two-stage-pipelined architecture, shown in Fig. 5. Several mathematical operations are performed in parallel. At the first stage, stator currents and voltages in $\alpha\beta$-coordinates are calculated in parallel, so that those results can be used to estimate the stator flux in the same stage. The resulted currents and flux are used to determine the flux magnitude and the torque estimation in the second stage. A 62-b nonrestoring square root is implemented in order to compute the flux magnitude.

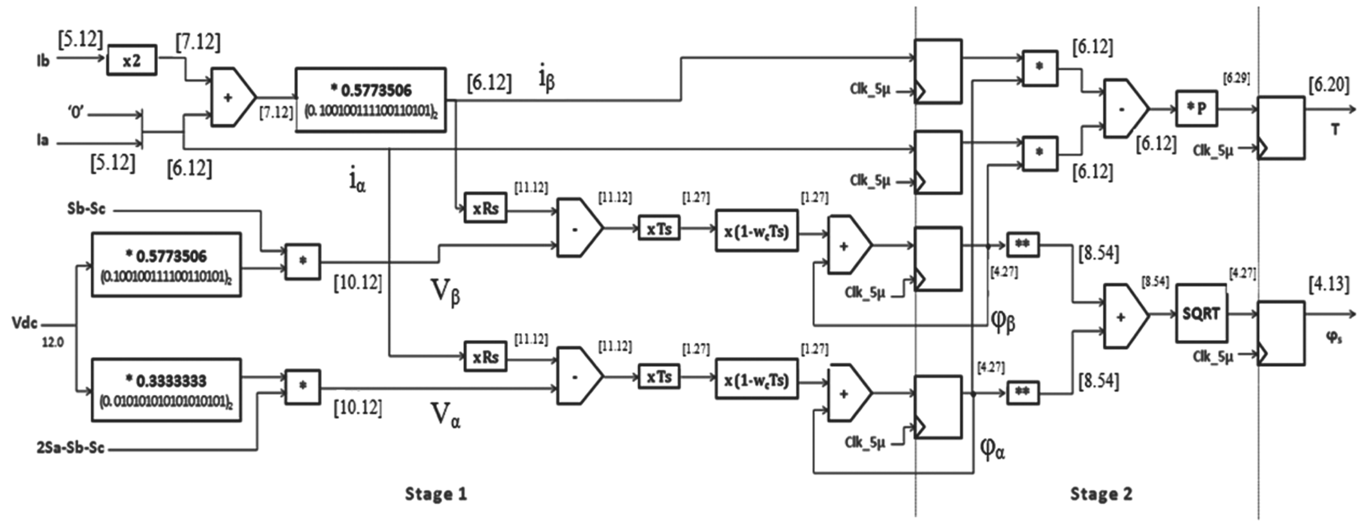

Fig. 5. Architecture of torque and flux estimators.

The work in [20] proposed that a three-stage-pipelined architecture should be implemented in this module by separating the computation of stator currents and voltages from the estimation of the stator flux. However, the former can be considered as an immediate calculation, and, therefore, those calculations can be merged into a single stage. As a consequence, the latency of the estimator is reduced from 15 to 10 μs.

B. Digital Properties of the Torque and Flux Estimators

To achieve a good implementation, several digital properties need to be considered when designing these estimators. Adopted binary format, quantization, and sampling time are amongst the key factors.

1) Binary Format Representation: In this implementation, two's complement fixed-point representation is used during all of the operations, except for the square root calculation. In this particular case, unsigned fixed-point representation is applied, since its operand and its results are always positive.

2) Quantization: The determination of word size (word length) is one of the critical parts in FPGA implementation. On one hand, the use of an insufficient number of bits may reduce the precision or cause a calculation error, which can destabilize the whole system. On the other hand, the use of larger words may increase the hardware implementation area.

3) Sampling Time: The sampling time T_s is limited to 5 μs by the ADC used. Therefore, all of the operations involved in this model are performed within this sampling time.

C. VHDL Design of the Torque and Stator Flux Estimators

The algorithm of stator flux and torque estimator is implemented in an architecture consisting of seven blocks.

1) I_α and I_β Calculation: The function of this block is to transform the stator phase currents I_a, I_b and I_c into the stationary α–β coordinates (I_α and I_β) refer to (2) and (3). In this design, the values (I_a, I_b) and (I_α and I_β) are represented on 17-b and 18-b two's-complement fixed-point format [5.12 b] and [6.12 b], respectively, to get the precise values. As shown in Fig. 5, to avoid overflow that the result calculation of "$I_a + 2I_b$"

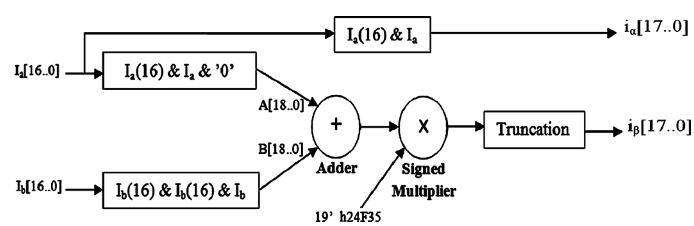

Fig. 6. Block arithmetic unit of the I_α and I_β calculations.

and $(1/3)\sqrt{3}$ of (2) are represented each on 19 b, as [7.12] and [1.18], respectively. The part of $(1/3)\sqrt{3}$ of (2) is represented as 151349 (i.e., $(1/3)\sqrt{3}x2^{18}$). In Fig. 6, the value of $(1/3)\sqrt{3}x2^{18}$ is represented as "19' h24F35" (19 is the number of bits, and the h24F35 is value of 151349 in hexadecimal). The output of the signed multiplier is represented on 38 b, as [8.30]. However, the I_β is only represented on 18 b as [6.12] to minimize hardware resource, so the 38 b [8.30] is truncated to become 18 b [6.12]. Based on the evaluation result, the 18 b has been considered suitable to represent I_β precisely. Here, the "tailor made" experience of designers is very important in order to develop the VHDL code effectively. The efficient implementation of the algorithms largely depends on the designer's experience [34]. Therefore, the paper offers a simpler arithmetic concept based on two's complement fixed-point format for VHDL programming.

2) V_α and V_β Calculation: The function of this block is to calculate the stator voltages in α-β components—refer to (4) and (5). The input is 12-b high-voltage dc-supply and three switching status. The output voltages are represented in 22-b two's-complement fixed-point format [10.12]. The RTL viewer of the calculation is shown in Fig. 7. The numbers "*19' h24F35*" and "*19' h15555*" are to represent $1/3$ and $\sqrt{3}/3$ in (4) and (5), respectively. In these cases represent 87381 (i.e., $(1/3)x2^{18}$) and 151349 (i.e., $(\sqrt{3}/3)x2^{18}$), respectively, in binary 19 b, each as [1.18]. It is important to note that the results (before truncated) of the V_α and V_β are allocated each to 34 b, as [16.18], but

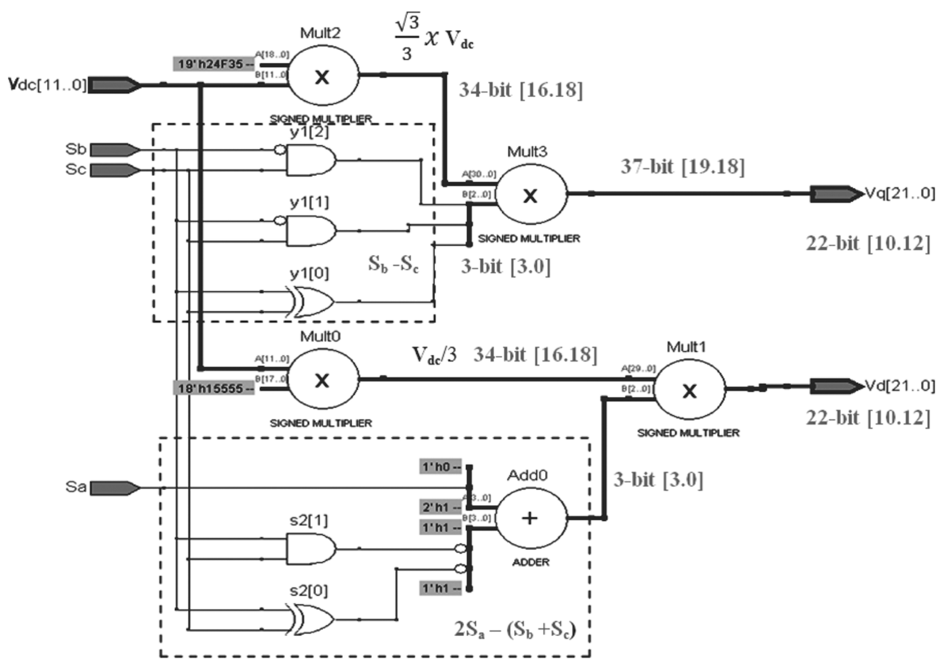

Fig. 7. RTL viewer of the V_α and V_α calculations.

the final values of the V_α and V_β are only 22 b. The most significant 9 b and least significant 6 b of each the V_α and V_β 34 b are truncated, so only 27th–6th bits are used to represent the final values of each the V_α and V_β as 22 b, i.e., [10.12]. The truncations are conducted to minimize hardware resources, while still retaining sufficient precision. Once again, the tailoring and adaptation made by the designers are very important here.

3) φ_s Calculation:

a) φ_α and φ_β Calculation: After the calculation of α-β components of current and voltage, the α-β flux is calculated in this block [refer to (6) and (7)]. The other input, $\mathrm{R_s}$, is represented on 10 b (5.5 b). The output α-β components of the stator flux are represented in 31-b two's-complement fixed-point format [4.27]. In this paper, the sampling time (T_s) is 5 μs. The value of T_s is represented in [1.27] as "*28' h000029F*" $(=671)$, and therefore the sampling time of 5 μs will be calculated as 4.99934 μs $(671/2^{27} \approx 0,00000499934$ s). Consider the $(1 - \omega_c * T_s)$ filter part of (9) and (10), the part is selected: 0.999975. In this case, the value is represented in [1.22] as "*23' h3FFF97*" $(=4194199)$, so 0.999 974 966 will be obtained to represent the 0.999 975 filter. The filter is designed to overcome the problem of integration drift. Therefore, the low-pass filter is used to replace the pure integrator with appropriate cutoff frequency (5 rad/s).

b) Magnitude Calculation: This block is designed to calculate the magnitude of the stator flux. The inputs of the block are the α–β components of stator flux, and the output is their magnitude, which is represented in 62-b fixed-point format (8.54 b). The RTL viewer for the magnitude calculation is shown in Fig. 8.

c) Square Root Calculation: This block is designed to calculate stator flux (φ_s) using modified nonrestoring square-root algorithm. The first output of the block is represented in 31-b fixed-point format, and then it is truncated to 17 b (4.13 b). The

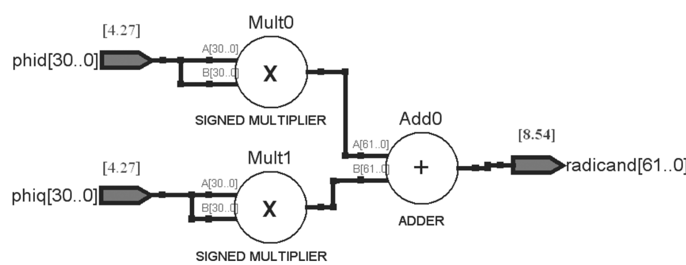

Fig. 8. RTL viewer of the magnitude calculator.

principle of the calculation is based on the powerful improved method presented in [35], which is created by authors originally, and can be used in general applications.

4) Determination of Sector: The work has introduced a simple method to determine the sector of the flux vector, based on a comparison as shown in Table I, which is modified from [30]. By using Karnaugh map simplification, it only involves two comparisons (as apposed to three comparisons). The RTL viewer of the sector determination is shown in Fig. 9.

5) Torque Calculation: The function of this block is to calculate the torque as given by (8). The output is represented on 55 b [14.41] fixed-point format, and then it is truncated to 26 b [6.20]. The RTL magnitude calculation viewer is shown in Fig. 10.

D. Synchronizations

In the proposed architecture of the torque and flux estimators, synchronizations are conducted in two stages. The first stage is used to synchronize the output of α-β stator currents (i_α and i_β) and α-β stator fluxes (φ_ε and φ_β), and the second stage is to synchronize the flux magnitude (T_e) and the electromagnetic torque (φ_s).

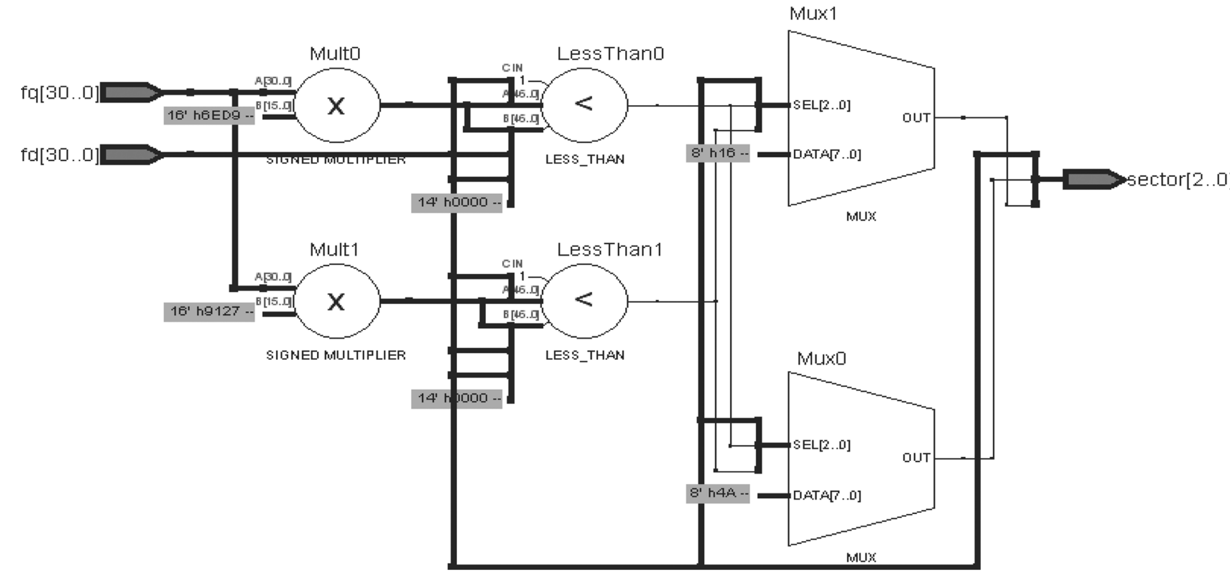

Fig. 9. RTL viewer of the sector judgment.

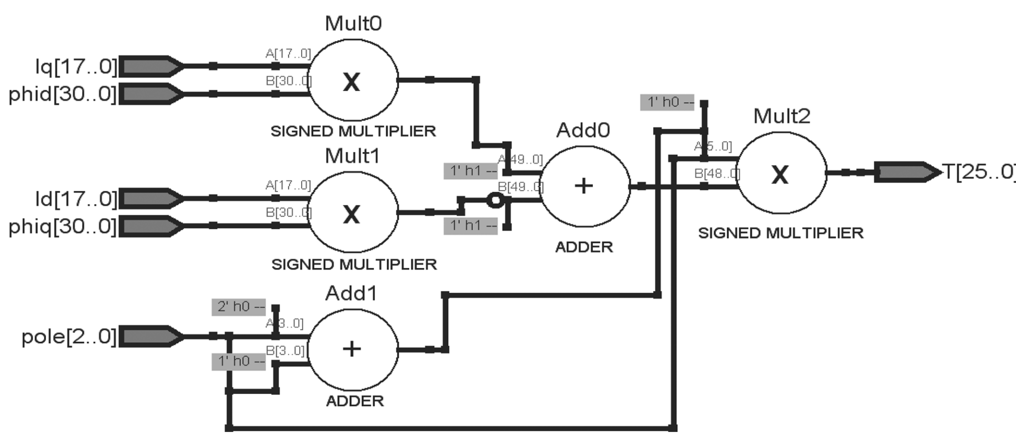

Fig. 10. RTL viewer of the torque calculator.

The synchronizations are designed in one cycle of the sampling time (in this case, 5 μs). By using the low sampling period (high sampling frequency), the torque ripple can be reduced significantly. In other words, the undesired overshoot or undershoot in torque can be minimized by employing a faster sampling time. The 5 μs sampling time (in the flux and torque estimators) can only be achieved by employing FPGA. With the DSPs and microprocessors available in the market today it may not be possible to implement such high sampling frequency. The one cycle synchronizations also functions as a buffer, so that the parameters can be loaded to the buffer in each clock cycle. The similar data-path and buffering concept have been introduced in [36] for application to an automatic speech recognition system based on FPGA.

VII. Results and Discussion

As discussed in Section II, the torque ripple can be reduced by increasing the sampling frequency. The sampling time used in this implementation is 5 μs and by doing so, the torque ripple is reduced to 0.2 Nm, as shown in Fig. 11. The figure shows the experimental results obtained using two different sampling time. With a much lower sampling time the torque can be limited within its hysteresis band since the oversoot (or undershhot) beyond the band is avoided. Eventually, the ripple can be reduced by reducing the hysteresis band.

The experiments were conducted on Altera APEX EP20K200EFC484-2x and consumes 2093 logic elements for the implementation. The comparisons of the area (LEs) consumption between the results of the research presented in this paper, with other works, are shown in Table II.

As an alternative solution to the implementation, a low cost FPGA devices, such as from Cyclone family, can be used. For example Altera DE2 board which offers a rich set of features is suitable for sophisticated digital systems implementation. APEX EP20K200EFC484-2X was used for our implementation due to the availability of this device/board in our laboratory during the development of the system.

To test on the effectiveness of the FPGA-based estimators and DTC controller, experiment based on hardware-in-the-loop (HiL) simulation is conducted. In the HiL simulation setup, the induction motor is simulated using an FPGA device. The induction motor and inverter are modeled using a LUT which

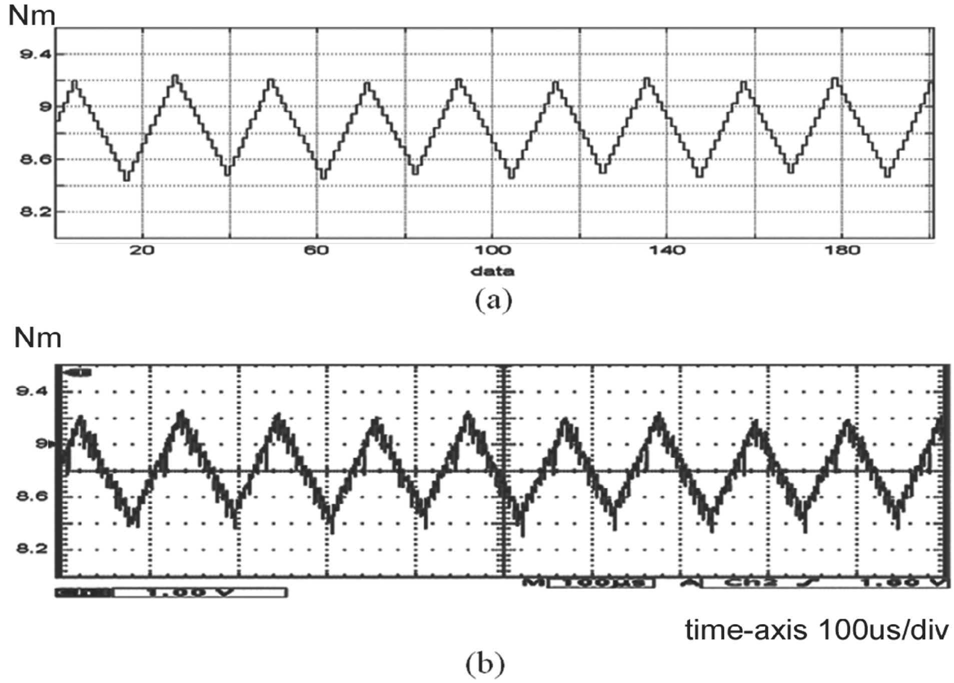

Fig. 11. Comparison between MATLAB\Simulink simulation and experimental result for torque estimation. (a) MATLAB\Simulink simulation. (b) FPGA-based experimental result.

TABLE II
COMPARISON OF THE LE CONSUMPTION

No	Reference	LEs consumption	DTC sampling period (kHz)
1	Ferreira [20]	4,100	40
2	Llor [37]	3,901	40
3	Utsumi [38]		
	- Type A controller	3,737	20
	- Type B controller	5,622	40
4	Bossoufi [39]	3,166	20
5	**Proposed**	**2,093**	**200**

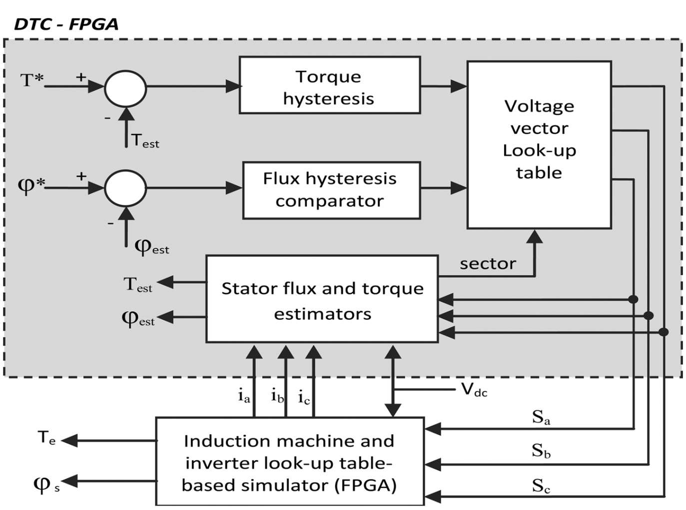

Fig. 12. HiL implementation of DTC.

is constructed based on the results obtained from the offline MATLAB/SIMULINK simulation run earlier. The HiL set up is illustrated in Fig. 12, and the parameters used in the HiL simulation are listed in Table III. Thereafter, the results were

TABLE III
INDUCTION MACHINE PARAMETERS

Parameters	Type or values
Rotor type	Squirrel-cage
Nominal power	2425 VA
Voltage (line to line)	400 V
Frequency	50 Hz
Stator resistance, R_s	5.5 ohm
Rotor resistance, R_r	4.45 ohm
Stator self inductance, L_s	0.0149 H
Rotor self inductance, Lr	0.0149 H
Mutual inductance, L_m	0.299 H
Combined inertia, J	0.00925 kg-m^2
Combined viscous friction, B	0.006 N.m.s
Number of pole pairs, P	2

compared to the validated MATLAB/Simulink simulations (carried out in double precision). Figs. 11 and 13 show some comparison results between MATLAB/Simulink offline simulations and the ones obtained from the real-time HiL simulation. In Fig. 13, the hysteresis band is reduced to approximately 0.7 Nm. Due to the very small sampling time of 5 μs, the torque ripple is mostly contained within the band, with very small overshoot and undershoot. Similarly, owing to the small sampling time, it is also possible to reduce the flux hysteresis band to a very small value of 0.00446 Wb (0.5% of the rated flux). As a result, Fig. 13 shows the locus of the flux of almost a perfect circle, with very small ripple. Consequently, one will expect almost a sinusoidal stator currents generated, with very small harmonic contents. It is important to note that the experimental outputs are displayed through a 12-bit DAC. So, all outputs are truncated within 12 b. Regardless of this, the offline simulation from Matlab/SIMULINK shows a very close agreement with the results obtained from the HiL real-time simulation as demonstrated by both Figs. 11 and 13. The results have

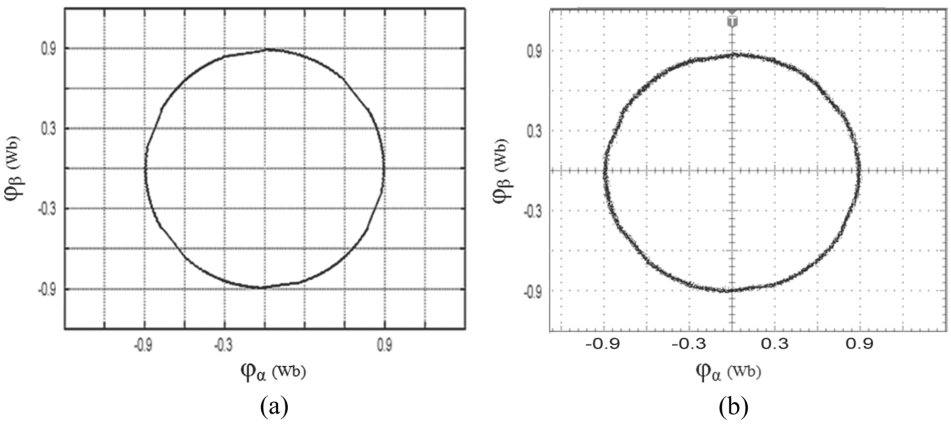

Fig. 13. Comparison between MATLAB/Simulink simulation and the experimental result for flux locus. (a) MATLAB/Simulink simulation. (b) FPGA-based experimental result.

proved that the proposed FPGA implementation of the torque and stator flux estimators is successful. All units in the system have been designed in fully generic VHDL code, independent of the target implementation technology, without the need for third party products or special FPGAs. Given that most of the DTC research solutions have limitations on the performance of the implementation of the torque and flux estimator, obviously this contribution has been eagerly awaited by researchers to support, enable, and take forward their DTC improvements.

VIII. CONCLUSION

This paper has achieved the reduction of the sampling time (to increase the sampling frequency) by using FPGAs, so that the width of the band of the hysteresis controller can be used to directly control the torque ripple. The technique retains the simple control structures of the DTC drive. The paper presented an effective way to design, simulate and implement hysteresis-based DTC utilizing FPGAs. All modules in the system have been designed in fully generic VHDL code, which is independent of the FPGA target implementation technology. All calculations in the modules are conducted in two's complement fixed-point arithmetic with appropriate word sizes. The choice of word sizes, the binary format and the sampling time used are very important in order to achieve a good implementation of the estimators. To get simpler implementation and fast computation, several methods were introduced: 1) the backward Euler approach to calculate the discrete integration operation of stator flux; 2) the modified non-restoring method to calculate complicated square root operation of stator flux; and 3) a new sector analysis method; the simulation results of the DTC model in MATLAB/Simulink, which performed double-precision calculations, are used as references to digital computations executed in FPGA implementation. The Hardware-in-the-loop (HiL) method is used to verify the minimal error between MATLAB/Simulink simulation and the experimental results. The design, which was coded in synthesizable VHDL code for implementation on Altera APEX20K200EFC484-2x device, has produced very good estimations, giving minimal errors when being compared with MATLAB/Simulink double-precision calculations.

REFERENCES

[1] I. Takahashi and T. Noguchi, "A new quick-response and high-efficiency control strategy of an induction motor," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 820–827, Sep./Oct. 1986.

[2] A. Jidin *et al.*, "Torque ripple minimization in DTC induction motor drive using constant frequency torque controller," in *Proc. Int. Conf. Electr. Machines Syst.*, 2010, pp. 919–924.

[3] N. R. N. Idris and A. H. M. Yatim, "Direct torque control of induction machines with constant switching frequency and reduced torque ripple," *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. 758–767, Aug. 2004.

[4] A. Tripathi, A. M. Khambadkone, and S. K. Panda, "Torque ripple analysis and dynamic performance of a space vector modulation based control method for AC-drives," *IEEE Trans. Power Electron.*, vol. 20, no. 2, pp. 485–492, Mar. 2005.

[5] D. Casadei, G. Serra, and K. Tani, "Implementation of a direct control algorithm for induction motors based on discrete space vector modulation," *IEEE Trans. Power Electron.*, vol. 15, no. 4, pp. 769–777, Jul. 2000.

[6] C. Lascu, I. Boldea, and F. Blaabjerg, "A modified direct torque control for induction motor sensorless drive," *IEEE Trans. Ind. Appl.*, vol. 36, no. 1, pp. 122–130, Jan.–Feb. 2000.

[7] T. G. Habetler, F. Profumo, M. Pastorelli, and L. M. Tolbert, "Direct torque control of induction machines using space vector modulation," *IEEE Trans. Ind. Appl.*, vol. 28, no. 5, pp. 1045–1053, Sep.–Oct. 1992.

[8] T. Noguchi, M. Yamamoto, S. Kondo, and I. Takahashi, "Enlarging switching frequency in direct torque-controlled inverter by means of dithering," *IEEE Trans. Ind. Appl.*, vol. 35, no. 6, pp. 1358–1366, Nov. –Dec. 1999.

[9] J. BeertenJ. VerveckkenJ. Driesen, "Predictive direct torque control for flux and torque ripple reduction," *IEEE Trans. Ind. Electron.*, vol. 57, no. 1, pp. 404–412, Jan. 2010.

[10] J. Kley *et al.*, "Performance evaluation of model predictive direct torque control," in *Proc. IEEE Power Electron. Specialists Conf.*, 2008, pp. 4737–4744.

[11] G. Papafotiou, J. Kley, K. G. Papdopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control—Part II: Implementation and experimental evaluation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1906–1915, Jun. 2009.

[12] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control—Part I: Concept, algorithm, and analysis," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[13] B. K. Bose and P. M. Szczesny, "A microcomputer-based control and simulation of an advanced IPM synchronous machine drive system for electric vehicle propulsion," *IEEE Trans. Ind. Electron.*, vol. 35, no. 4, pp. 547–559, Nov. 1988.

[14] L. Lianbing *et al.*, "A high-performance direct torque control based on DSP in permanent magnet synchronous motor drive," in *Proc. 4th World Congress Intell. Control Automat.*, 2002, vol. 2, pp. 1622–1625.

[15] S. M. A. Cruz *et al.*, "DSP implementation of the multiple reference frames theory for the diagnosis of stator faults in a DTC induction motor drive," *IEEE Trans. Energy Convers.*, vol. 20, pp. 329–335, 2005.

[16] A. Jidin *et al.*, "Simple dynamic overmodulation strategy for fast torque control in DTC of induction machines with constant switching frequency controller," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 2010, pp. 1–8.

[17] N. R. N. Idris, L. T. Chuen, and M. E. Elbuluk, "A new torque and flux controller for direct torque control of induction machines," *IEEE Trans. Ind. Appl.*, vol. 42, no. 6, pp. 1358–1366, Nov.–Dec. 2006.

[18] A. Jidin, N. R. N. Idris, A. H. M. Yatim, T. Sutikno, and M. E. Elbuluk, "An optimized switching strategy for quick dynamic torque control in DTC-hysteresis-based induction machines," *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3391–3400, Aug/ 2011.

[19] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems: A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.

[20] S. Ferreira *et al.*, "Design and prototyping of direct torque control of induction motors in FPGAs," in *Proc. 16th Symp. Integr. Circuits Syst. Design*, 2003, pp. 105–110.

[21] J. W. Kang and S. K. Sul, "Analysis and prediction of inverter switching frequency in direct torque control of induction machine based on hysteresis bands and machine parameters," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 545–553, Jun. 2001.

[22] A. Jidin *et al.*, "Extending switching frequency for torque ripple reduction utilizing a constant frequency torque controller in DTC of induction motors," *J. Power Electron.*, vol. 11, pp. 148–155, 2011.

[23] R. K. Behera and S. P. Das, "High performance induction motor drive: A dither injection technique," in *Proc. Int. Conf. Energy, Autom., Signal*, 2011, pp. 1–6.

[24] N. R. N. Idris and A. H. M. Yatim, "Reduced torque ripple and constant torque switching frequency strategy for direct torque control of induction machine," in *Proc. 15th Annu. IEEE Appl. Power Electron. Conf. Expo.*, 2000, vol. 1, pp. 154–161.

[25] A. Saleem *et al.*, "Hardware-in-the-loop for on-line identification and control of three-phase squirrel cage induction motors," *Simul. Modelling Practice Theory*, vol. 18, pp. 277–290, 2010.

[26] A. Barakat *et al.*, "Analysis of synchronous machine modeling for simulation and industrial applications," *Simul. Modelling Practice Theory*, vol. 18, pp. 1382–1396, 2010.

[27] N. R. N. Idris and A. H. M. Yatim, "An improved stator flux estimation in steady state operation for direct torque control of induction machines," in *IEEE Ind. Appl. Conf. Rec.*, 2000, vol. 3, pp. 1353–1359.

[28] N. R. N. Idris and A. H. M. Yatim, "An improved stator flux estimation in steady-state operation for direct torque control of induction machines," *IEEE Trans. Ind. Appl.*, vol. 38, no. 1, pp. 110–116, Jan.–Feb. 2002.

[29] S. Samavi *et al.*, "Modular array structure for non-restoring square root circuit," *J. Syst. Architecture*, vol. 54, pp. 957–966, 2008.

[30] T. Sutikno *et al.*, "New approach FPGA-based implementation of discontinuous SVPWM," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 18, p. 6, 2010.

[31] C. T. Kowalski *et al.*, "FPGA implementation of DTC control method for the induction motor drive," presented at the EUROCO, Sep. 9–12, 2007, pp. 1916–1921.

[32] E. Monmasson *et al.*, "FPGAs in industrial control applications," *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 224–243, 2011.

[33] B. Alecsa *et al.*, "Simulink modeling and design of an efficient hardware-constrained FPGA-based PMSM speed controller," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 554–562, 2012.

[34] J. J. Rodriguez-Andina *et al.*, "Features, design tools, and application domains of FPGAs," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1810–1823, Aug. 2007.

[35] T. Sutikno *et al.*, "A simple strategy to solve complicated square root problem in DTC for FPGA implementation," in *Proc. IEEE Symp. Ind. Electron. Appl.*, 2010, pp. 691–695.

[36] O. Cheng *et al.*, "Hardware-software codesign of automatic speech recognition system for embedded real-time applications," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 850–859, Mar. 2011.

[37] A. Llor *et al.*, "Comparison of DTC implementations for synchronous machines," in *Proc. IEEE 35th Annu. Power Electron. Specialists Conf.*, 2004, vol. 5, pp. 3581–3587.

[38] Y. Utsumi *et al.*, "Comparison of FPGA-based direct torque controllers for permanent magnet synchronous motors," *J. Power Electron.*, vol. 6, pp. 114–120, 2006.

[39] B. Bossoufi *et al.*, "FPGA-based implementation by direct torque control of a PMSM machine," in *Proc. 7th Int. Conf. Workshop, Compatibility and Power Electron.*, 2011, pp. 464–469.

Tole Sutikno (M'08) received the B.Eng. degree from Diponegoro University, Indonesia, and the M.Eng. degree from Gadjah Mada University, Indonesia, in 1999 and 2004, respectively, both in electrical engineering. He is currently working toward the Ph.D. degree at the Universiti Teknologi Malaysia, Malaysia.

Since 2001, he has been a Lecturer with the Electrical Engineering Department, Universitas Ahmad Dahlan, Indonesia. His research interests include the field of power electronics, motor drive systems and FPGA applications.

Nik Rumzi Nik Idris (M'97-SM'03) received the B.Eng. degree in electrical engineering from the University of Wollongong, Australia, in 1989, the M.Sc. degree in power electronics from Bradford University, West Yorkshire, U.K., in 1993, and the Ph.D. degree from Universiti Teknologi Malaysia, Malaysia, in 2000.

He is an Associate Professor with the Universiti Teknologi Malaysia. His research interests include ac drive systems and DSP applications in power electronic systems.

Prof. Idris is an administrative committee member of the IAS/PELS/IES Joint Chapter of the IEEE Malaysia Section.

Auzani Jidin (M'09) received the B.Eng., M.Eng., and Ph.D. degrees in power electronics and drives from Universiti Teknologi Malaysia, Malaysia, in 2002, 2004, and 2011, respectively.

He is a Lecturer with the Department of Power Electronics and Drives, Universiti Teknikal Melaka Malaysia, Malaysia. His research interests include the field of power electronics, motor drive systems, FPGAs, and DSP applications.

Marcian N. Cirstea (M'97–SM'04) received the Degree in electrical engineering from the Transilvania University of Brasov, Brasov, Romania, and the Ph.D. degree from Nottingham Trent University, Nottingham, U.K., in 1996.

He is currently a Professor of industrial electronics and Head of the Computing and Technology Department, Anglia Ruskin University, Cambridge, U.K., after previously working with De Montfort University, U.K. He has coauthored several technical books and over 100 peer-reviewed papers, three of which have received awards. His research is focused on digital controllers for power electronics. He has delivered five international tutorials on VHDL design for power electronic systems modeling and FPGA controller prototyping.

Dr. Cirstea is founder and past Chairman of the Electronic Systems-on-Chip Technical Committee of the IEEE Industrial Electronics Society, Fellow of IET, and Chartered Engineer (CEng). He is an associate editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He was General Chair of ISIE Conference (Cambridge, 2008). He coordinated a European renewable energy project consortium.