FPGA Based Torque and Flux Estimator of Direct Torque Control for Induction Machine Drives

Aiman Zakwan JIDIN[#1], Tole SUTIKNO*[2],Auzani JIDIN•[3], Nik Rumzi Nik IDRIS□[4]

[#1] *Ecole Supérieur d'Ingénieur en Electronique et Electrotechnique Paris (ESIEE Paris), France*
[#1]jidina@esiee.fr
[*2]*Universitas Ahmad Dahlan (UAD), Indonesia*
*[2]tole@ee.uad.ac.id
[•3] *Universiti Teknikal Malaysia Melaka (UTeM), Malaysia*
•[3]auzani@ieee.org
[□4]*Universiti Teknologi Malaysia (UTM), Malaysia*
□[4]nikrumzi@ieee.org

***Abstract*— This paper presents a new design of the torque and stator flux estimators for Direct Torque control (DTC) for Field Programmable Gate Array (FPGA) implementation, which permit very fast calculations.An alternative variable word-size approach in two's complement fixed-point format is used for the implementation, in order to minimize calculation errors and the hardware resource usage. The simulation results of DTC model in Matlab, 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 simulation and the experimental results, and thus the well-functionality of the implemented estimators.**

I. INTRODUCTION

Direct Torque control (DTC) is an alternative for controlling induction machines, which was introduced by Takahashi (1986) [1] and Depenbrock (1988) [2]. It become famous owing to its simple structure and good behavior such as its high efficiency and low losses, no separate modulator is needed, the coordinate trans-formation is not necessary, while the position encoder and the PI current controller are not required [3-7].

The DTC algorithm is frequently implemented in a Microcontroller or Digital Signal Processing (DSP) [8-10]. However, serial calculations are performed and therefore, they cannot execute very fast computation without any losses. As an adequate solution, fast calculations are performed by using FPGA[11-12]. Moreover, its high sampling frequency allows the minimization of torque ripple [13-16].

However, it is not easy to implement DTC in FPGA hardware. One of the hardest parts in the DTC implementation is the torque and flux estimations [17].Complex digital computations are involved, such as binary multiplications and also a square root calculation targeted for FPGA implementation, the difficulties of which have been addressed in several researches [18-20].

This paper presents a new design of the torque and flux estimator in DTC for FPGA implementation, using two's complement fixed-point representation with variable words' sizes through the estimation process. The validation of the implementation is done by verifying the experimental results during the motor's steady state.

II. DIRECT TORQUE CONTROL

Figure 1 represents the topology of a DTC drive:

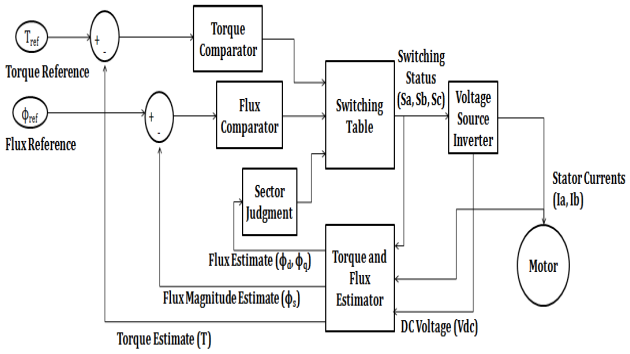

Figure 1DTC topology

The estimated flux magnitude and torque are compared with their references values. Torque and flux comparators are consisted of three and two-level hysteresis respectively. Besides, the sector judgment evaluates the position of the stator flux vector in DQ coordinates.

The switching table produces the switching status according to the output of torque and flux comparators and the sector judgment. Those switching status are connected to the inverter, which is connected to the motor. They are also used as the input for torque and flux estimation.

In order to estimates the stator flux and the electromagnetic torque, several parameters need to be determined. Firstly, the stator currents from the motorIa and Ib, are transformed into DQ coordinates, which are adequate to DTC algorithm, as follows:

$$Id = Ia \quad (1)$$

$$Iq = \frac{\sqrt{3}}{3}(Ia + 2Ib) \quad (2)$$

At the same time, by using the switching status (Sa, Sb and Sc) produced by the switching table, the stator voltages in DQ components are determined:

$$Vd = \frac{Vdc}{3} * (2Sa - Sb - Sc) \quad (3)$$

$$Vq = \frac{\sqrt{3}}{3} * Vdc * (Sb - Sc) \quad (4)$$

Then, using the calculated Id, Iq, Vd and Vq, the estimation of the stator flux in DQ coordinates are performed as follows:

$$\varphi_d = \varphi_{d_{n-1}} + (Vd - Rs * Id) * Ts \quad (5)$$

$$\varphi_q = \varphi_{q_{n-1}} + (Vq - Rs * Iq) * Ts \quad (6)$$

Notice that Rs is the estimated stator resistance, while Ts is the implementationsampling time. In addition, equation (5) and (6) correspond to the integration using Back Euler Method. As a matter of fact, [21-22] suggested that a filter should be added to the integrator in the practical implementation. Thus, equation (5) and (6) become:

$$\varphi_d = \left(\varphi_{d_{n-1}} + (Vd - Rs * Id) * Ts\right) \times (1 - \omega_c Ts) \quad (7)$$

$$\varphi_q = \left(\varphi_{q_{n-1}} + (Vq - Rs * Iq) * Ts\right) \times (1 - \omega_c Ts) \quad (8)$$

Finally, equation (9) calculates flux magnitude by using a square root calculation, whereas the electro magnetic torque is estimated in equation (10).

$$\varphi_s = \sqrt{\varphi_d^2 + \varphi_q^2} \quad (9)$$

$$T = \frac{3}{4} * P * (Iq * \varphi_d - Id * \varphi_q) \quad (10)$$

III. TORQUE AND FLUX ESTIMATOR ARCHITECTURE

The algorithm of torque and flux estimation is implemented in anarchitecture consisted of five main blocks, as shown in Figure 2. This architecture has six inputs: two 21-bit currents Ia and Ib and three switching status Sa, Sb and Sc. At the end, it produces four outputs: the estimation values of torque φ_d ,φ_q ,and φ_S.The sampling time chose is 5 µs, which is limited by the ADC used.

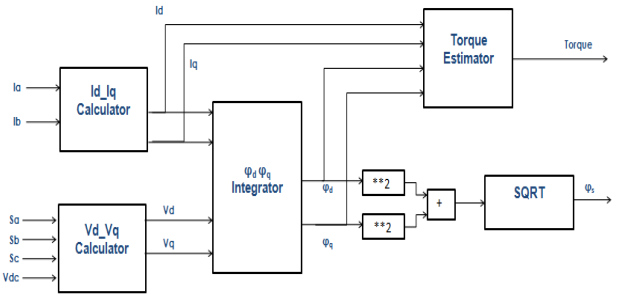

Figure 2 Block Diagram of torque and flux estimators

All the equations which are modeling the motor behavior are implemented in a two-stage-pipelined architecture, as presented in Figure 3.Several mathematical operations are performed in parallel. At the first stage, stator currents and voltages in DQ-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-bit non-restoring square root is implemented in order to compute the flux magnitude.

As the matter of fact, [17] proposed that 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 thus, those calculations can be merged into one single stage. As a consequence, the latency of the estimator is reduced from 15 µs to 10 µs.

To achieve a good implementation, several digital characteristics need to be considered when designing this estimator. Adopted binary format, quantization and sampling time are among those key factors.

A. Binary Format

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

Recent DTC implementation generally used 32-bit format where some bits might be left unused, while16-bit format is not appropriate to achieve good DTC implementation [23]. Therefore, variable word-size approach is adopted for this implementation and so, all

the redundant bits can be eliminated by truncating process to minimize the hardware resources usage.

B. Quantization

The determination of word size is one of the critical parts in FPGA implementation. On one hand, insufficient number of bits used may reduce the precision or cause the calculation error, which can unstabilize the whole system. On the other hand, larger words used may increase the hardware area used for the implementation.

Since two's complement fixed-point format is used for the implementation, at least 2 things that need to be verified. Firstly, the size of the integer must be properly chosen to avoid the problem of overflow. Secondly, the number of the fractional bits used must be sufficient in order to minimize the quantization errors.

For example, due to the fact that the input currents Ia and Ib are varied from -10A to 10A, at least 5 bits are necessary for theinteger bits. While 16 fractional bits used can result in a very good precision, since the resolution is very small ($\simeq$15 µA).

One of the critical parts in this architecture is the stator flux estimation, where the integration is performed. This operation caneasily produce errors if the sampling time Ts is not properly scaled.

In this case, Ts = 5 µs = 0.000005 s. In fact, a minimum of 21 bits isnecessaryto represent Ts. In this case, Ts= 0.00000476837 s $(0.00000000000000001010)_2$. However, 27-bit representation is chosen to have a better precision and thus, Ts= 0.00000499934 s $(0.00000000000000001010011111)_2$.

Figure 5 shows the estimated torque taken during the steady state. From figure 5, it shows that the torque estimation for 21-bit Ts is imprecise, compared to Matlab double precision estimation, which is the ideal case.

In fact, the number of bits is increasing after each operation in order to avoid calculation errors or imprecision.This will result in the rising of the hardware area used. Therefore, truncation process must be performed avoid the excessive increase of the number of bits used.

In the example shown in Figure 4, when Ib is multiplied by 2, the result should be in 6.16 bits (6 integer bits plus 16 fractional bits). Nevertheless, it is stored in 7.16 bits to avoid overflow, which may happen during the addition operation. Next, when the addition result is multiplied by, which coded in 1.18 bits, Iq should be represented in 8.34 bits. But, it is truncated to 6.16 bits.

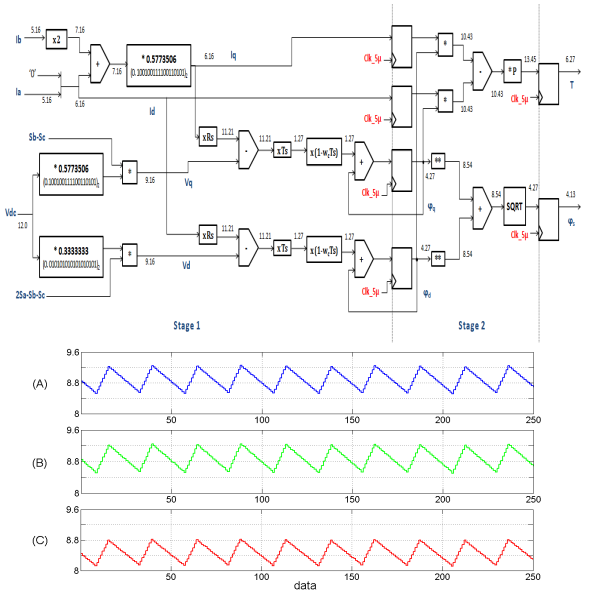

Figure 4 The torque estimation during steady state. (A) Estimated torque for Matlab double precision; (B) Estimated torque for Ts in 27 bits; (C) Estimated torque for Ts in 21 bits.

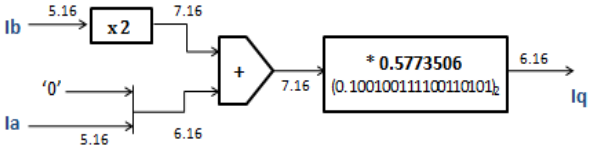

Figure 5 The example of Iq calculation

C. Sampling Time

The sampling time Ts is 5 µs. Therefore, all the operations involved in this model were performed within this period.

Notice that the use of high sampling frequency is important in DTC implementation, for the purpose to minimize the torque ripple. The sampling time used for DSP implementation is normally much bigger than Ts, which is not less than 50 µs.Therefore, it is reduced by a factor of 10 for this FPGA implementation and thus, lower torque ripple is produced, as shown in Figure 6.

IV. EXPERIMENTAL RESULTS

The validation of designed torque and flux comparators was performed based on Hardware-in-the-Loop (HIL) method. The DTC model in Matlab Simulink was simulated and then, the same data Ia, Ib, Sa, Sb, and

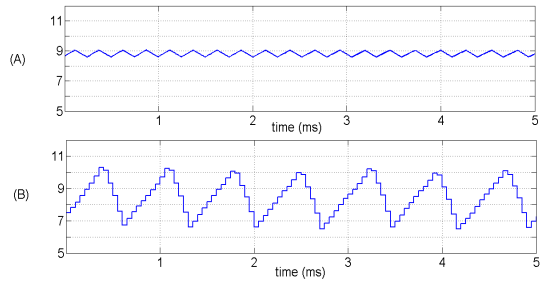

Figure 6 (A) Estimated torque for Ts = 5 µs; (B) Estimated torque for Ts = 50 µs.

Sc used for the simulation were copied from Matlabworkspace to VHDL codes, as the inputs for the targeted FPGA.

The VHDL codes were simulated in Modelsim before being synthesized and implemented in Altera EP2C35F672C6. The test design flow is presented in Figure 7.

The experiments were executed for three different motor's speeds: low speed, middle speed and high speed. All the experimental results were compared to the validated Matlab simulation results. Since the hardware resources in FPGA are limited to store all the inputs and the outputs, the tests were taking place only during certain periods of motor's steady state. The implementation results were observed on the oscilloscope.

Figure 8 presents the input of the high-speed test while Figures 9 to 11 shows the comparisons between Matlab simulations and the experimental results.

The experimental results shown were corresponded well with the simulation in Matlab, which was done in double-precision computation. Besides, the implementation of the algorithm was also validated for different motor's speed.It can be observed by comparing the form of the estimated torque triangles or the number of flux waveform complete cyclevisualized within the same period, for example.

V. CONCLUSION

FPGA is an alternative for the realization of a high-performance DTC implementationowing to its high processing frequency, which cannot be obtained by any DSP application. The choice of words' sizes, the binary format and the sampling time used are very important in order to achieve a good implementation of the estimators.

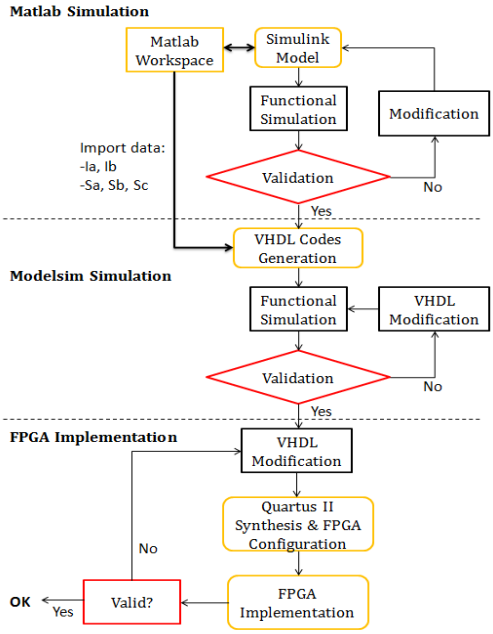

Figure 7 Top-down test design flow

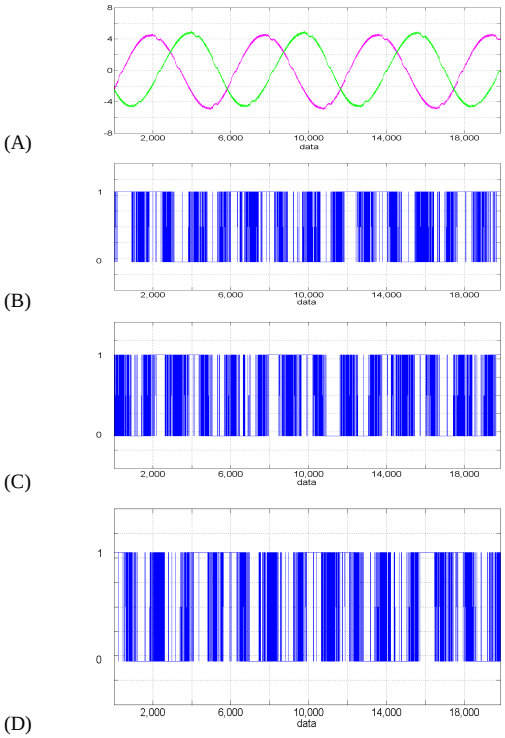

Figure 8 The inputs for high-speed test: (A) The stator currents Ia and Ib; (B) Sa; (C) Sb; (D) Sc.

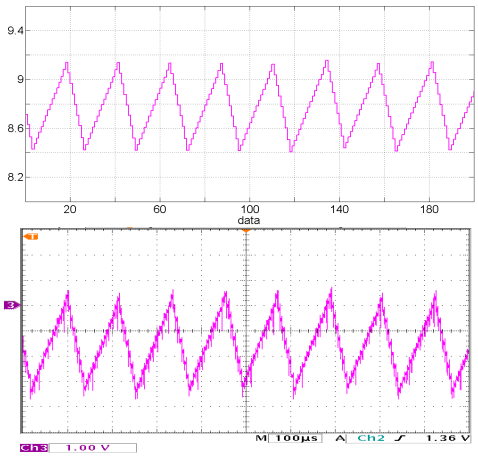

Figure 9 Comparison between Matlab simulation and the experimental result for torque estimation in high-speed test.

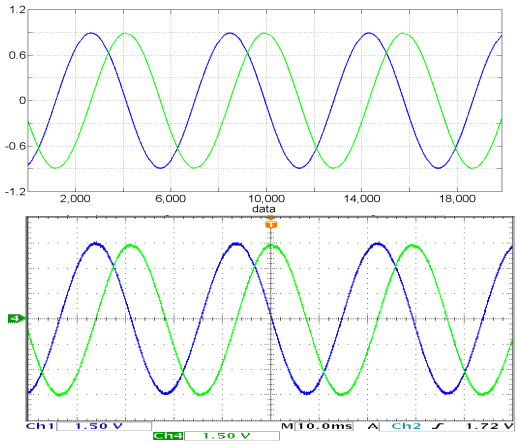

Figure 10 Comparison between Matlab simulation and the experimental result for φ_s and φ_r estimations in high-speed test.

The design, which coded in synthesizable VHDL,utilized 2093 logic elements for the implementation on Altera EP2C35F672C6 device, producing very precise estimations with minimal quantization and calculation errors. Therefore, FPGA is a reliable device to be utilized for any digital signal processing algorithm, which requires high-speed execution.

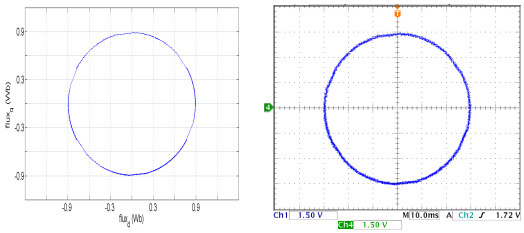

Figure 11Comparison between Matlab simulation and the experimental result for flux locus in high-speed test.

REFERENCES

[1] I. Takahashi and T. Noguchi, "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," IEEE Transactions on Industry Applications, vol. Vol.IA-22, No.5, pp. 820-827, Sept/Oct 1986.

[2] M. Depenbrock, "Direct self control (DSC) of inverter-fed induction machine," IEEE Trans. on Power Electronics, vol. 3 (4), pp. 420–429, 1988.

[3] T. G. Habetler, et al., "Direct torque control of induction machines using space vector modulation," Industry Applications, IEEE Transactions on, vol. 28, pp. 1045-1053, 1992.

[4] L. Zhong, et al., "Analysis of direct torque control in permanent magnet synchronous motor drives," Ieee Transactions on Power Electronics, vol. 12, pp. 528-536, May 1997.

[5] C. French and P. Acarnley, "Direct torque control of permanent magnet drives," Industry Applications, IEEE Transactions on, vol. 32, pp. 1080-1088, 1996.

[6] L. Yong, et al., "Direct torque control of brushless DC drives with reduced torque ripple," Industry Applications, IEEE Transactions on, vol. 41, pp. 599-608, 2005.

[7] L. Yong, et al., "Commutation-Torque-Ripple Minimization in Direct-Torque-Controlled PM Brushless DC Drives," Industry Applications, IEEE Transactions on, vol. 43, pp. 1012-1021, 2007.

[8] L. Lianbing, et al., "A high-performance direct torque control based on DSP in permanent magnet synchronous motor drive," in Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on, 2002, pp. 1622-1625 vol.2.

[9] L. Weijie, "Implementation of Direct Torque Control for Permanent Magnet Synchronous Motor with Space Vector Modulation Based on DSP," in Signal Processing, 2006 8th International Conference on, 2006.

[10] 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," Energy Conversion, IEEE Transactions on, vol. 20, pp. 329-335, 2005.

[11] E. Monmasson and M. N. Cirstea, "FPGA Design Methodology for Industrial Control Systems: A Review," Industrial Electronics, IEEE Transactions on, vol. 54, pp. 1824-1842, 2007.

[12] C. T. Kowalski, et al., "FPGA Implementation of DTC Control Method for the Induction Motor Drive," in EUROCON, 2007. The International Conference on Computer as a Tool, 2007, pp. 1916-1921.
[13] M. Mizuochi, et al., "Multirate Sampling Method for Acceleration Control System," Industrial Electronics, IEEE Transactions on, vol. 54, pp. 1462-1471, 2007.
[14] R. Morales-Caporal and M. Pacas, "A Predictive Torque Control for the Synchronous Reluctance Machine Taking Into Account the Magnetic Cross Saturation," Industrial Electronics, IEEE Transactions on, vol. 54, pp. 1161-1167, 2007.
[15] F. Morel, et al., "A predictive current control applied to a permanent magnet synchronous machine, comparison with a classical direct torque control," Electric Power Systems Research, vol. 78, pp. 1437-1447, 2008.
[16] C. L. Toh, et al., "Constant and high switching frequency torque controller for DTC drives," Power Electronics Letters, IEEE, vol. 3, pp. 76-80, 2005.
[17] S. Ferreira, et al., "Design and prototyping of direct torque control of induction motors in FPGAs," in Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings. 16th Symposium on, 2003, pp. 105-110.
[18] L. Yamin and C. Wanming, "Implementation of Single Precision Floating Point Square Root on FPGAs," in IEEE Symposium on FPGA for Cusom Computing Machines, Napa, California, USA, 1997, pp. 226-232.
[19] K. Piromsopa, et al., "An FPGA Implementation of a fixed-point square root operation," presented at the Int. Symp. on Communications and Information Technology (ISCIT 2001), ChiangMai, Thailand, 2001.
[20] S. Lachowicz and H. J. Pfleiderer, "Fast Evaluation of the Square Root and Other Nonlinear Functions in FPGA," in Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on, 2008, pp. 474-477.