FPGA-Based Control for Electric Vehicle and Hybrid Electric Vehicle Power Electronics

WP-01210-1.0 White Paper

FPGAs are gaining acceptance in high-performance power electronics control systems due to their speed, flexibility, and integrated design tools. This white paper describes the benefits of using FPGA-based control in a hybrid electric vehicle (hybrid EV) or electric vehicle (EV) drive system comprised of a variable-voltage control (VVC) or bidirectional DC-DC converter, 3-phase inverters, and interior permanent magnet (IPM) motor or generators.

- *Simplified Power Control Architecture for Hybrid EVs*—Integrates control functions of both VVC converter and IPM motor inverters into one FPGA for consolidated control hardware, and off-loads the computation of faster control loops from microcontroller units (MCUs).
- *High-Frequency VVC Converter*—Allows the use of smaller, lower cost reactive components and increasingly available high-speed switching silicon carbide (SiC) MOSFETs
- *IPM Motor Control with DTFC-SVM*—Reduces torque ripple pulsations through control of IPM motor with direct torque and flux control with space vector modulation (DTFC-SVM)

The white paper also describes the implementation of the VVC converter and the motor inverter control as follows:

- Implementation of control algorithms in The MathWorks simulation environment
- Automated design for synthesizing to FPGA using Altera® DSP Builder
- FPGA performance and resource metrics

Introduction

Analog control has given way to digital methods that have improved the performance and quality of power converters. (1) Most power electronics today are controlled by MCUs. This is mainly due to the low-cost nature of these devices and high level of integration of peripherals such as analog-to-digital converters. MCUs are typically programmed in C or Assembly languages, which can be outside the core expertise of power electronics engineers. MCUs are well-suited to algorithms that are executed sequentially with a rate within the MCU processor's capability. Faster sample rates and more complex algorithms are creating challenges with this traditional approach.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

© 2013 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

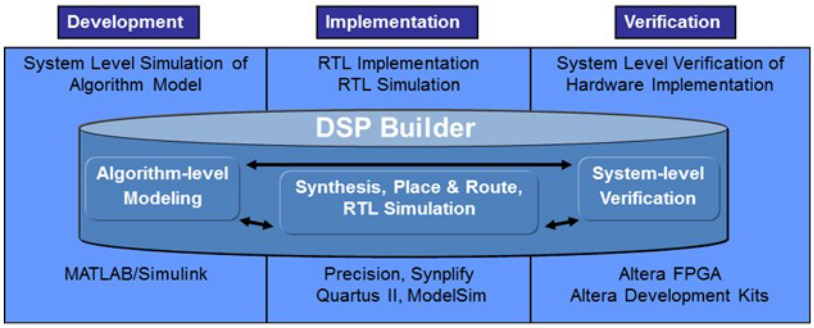
ISO 9001:2008 Registered

Parallel processing offers a solution, specifically by using an FPGA. FPGAs are well-suited for hybrid EV and EV drive system applications such as VVC and motor control due to their parallel architecture and ability to handle multiple complex algorithms simultaneously in hardware. The FPGA is programmed by connecting the gates together to form multipliers, registers, adders, FIFOs, memory-mapped registers, and other functions. This can be done using HDL, which places a dedicated resource for the task and allows for parallel operation.

Altera DSP Builder Introduction

The complexity of HDL coding can be a barrier for power electronics engineers. Altera's DSP Builder tool provides MathWorks Simulink design blocks and the ability to auto-generate HDL code. It allows the same model used to simulate the system to be directly implemented into the FPGA. In addition, it allows the designer to leverage a rich library of power electronics components when constructing the testbench or system simulation model. The use of DSP Builder in a Simulink environment can provide a holistic system model, which gives valuable understanding of the design prior to building hardware.

Figure 1. Altera DSP Builder Integrated Design Environment



This approach integrates the algorithm development, hardware implementation, and in-system verification steps into a common toolflow using the same source. Changes made during system verification can be immediately verified against simulation models.

The DSP Builder toolflow allows the designer to remain within the MathWorks environment and eliminates the need for HDL coding expertise. Moreover, the toolflow creates optimized FPGA implementations, with $\mathbf{f_{MAX}}$ and logic usage similar to hand-coded HDL. In addition, the tool provides the option to implement in either fixed point or floating point, something that is generally not feasible using a HDL approach. This frees the designer from numerical overflow, underflow, and saturation concerns in the algorithmic design phase. Multiple floating-point precisions are available, as the FPGA is not constrained to standard widths chosen to work with memory sizes. This allows the designer to trade off precision to FPGA logic and other resources.

Table 1 lists the floating-point precision options with Altera DSP Builder.

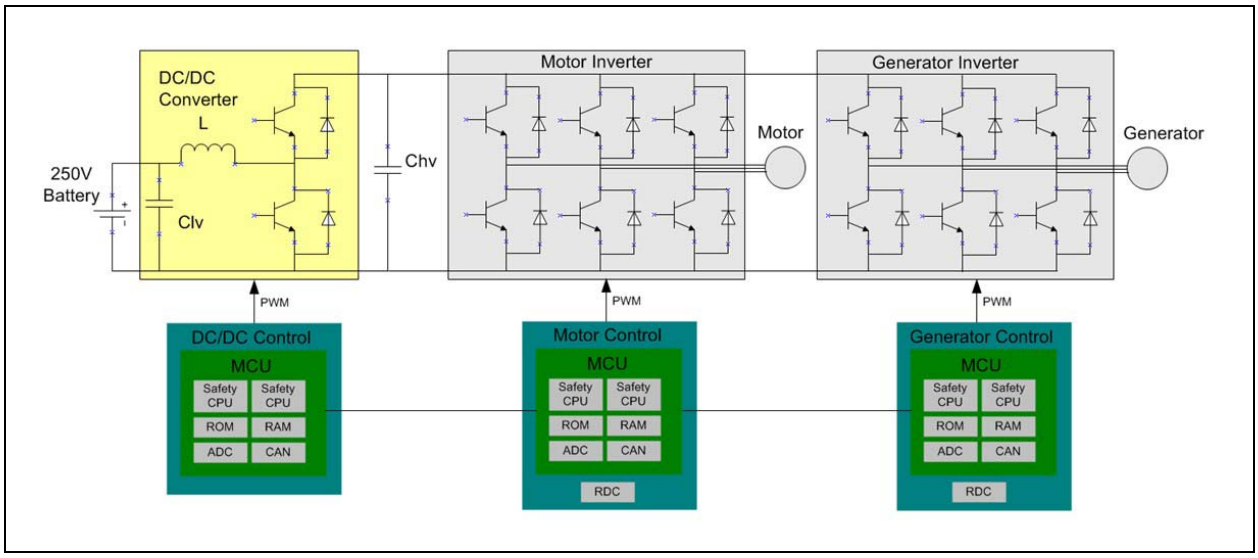
Table 1. Floating-Point Precision Options with Altera DSP Builder

Selectable Floating-Point Word Width (bits)	Mantissa Word Width (bits)	Logic Usage Relative to Single Precision
16 (“half” word)	10	0.3
26	17	0.6
32 (single precision)	23	1.0
35	26	1.4
46	35	2.2
55	44	3.4
64 (double precision)	52	4.6

Simplified Power Control Architecture for a Hybrid EV

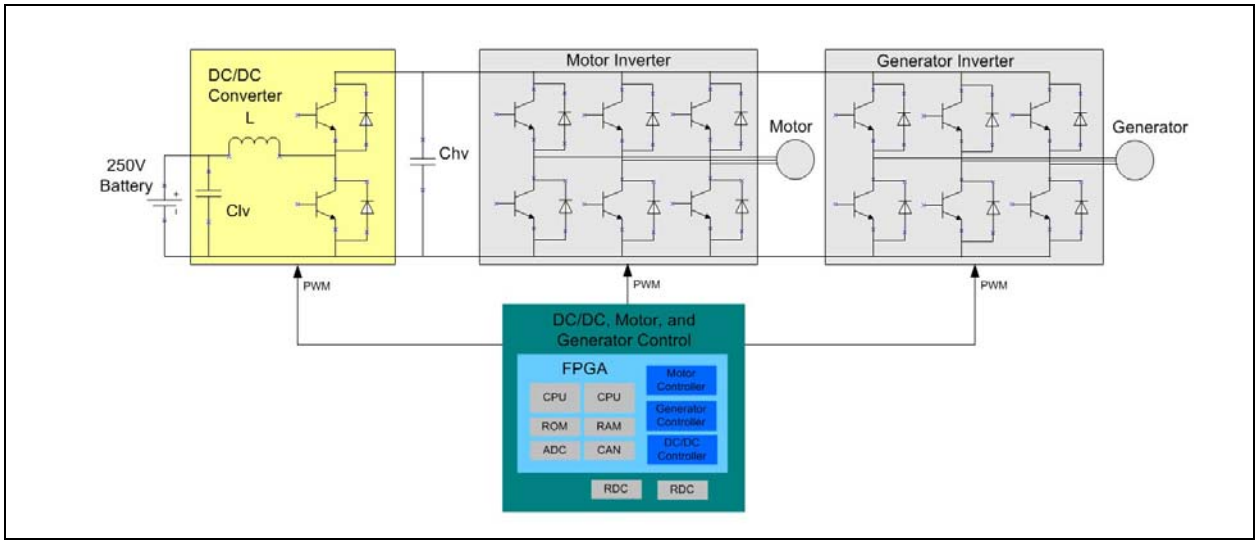
Figure 2 shows a common hybrid EV architecture that utilizes two independent motor or generators (MG) connected electrically through a DC link. The DC link is also connected to a 250 V battery though a VVC or bidirectional DC-DC converter comprised of an Insulated Gate Bipolar Transistor (IGBT) half-bridge and boost inductor. (2)

Figure 2. Standard Hybrid EV Power and Control Architecture



Each function (MGs and VVC) require sophisticated control circuits that are presently implemented with separate MCUs. FPGA technology allows multiple control functions to run in parallel on one device without the bottleneck of a single processor. Figure 3 shows a new architecture that integrates MG and VVC (DC-DC) control functions into a single FPGA.

Figure 3. Simplified Hybrid EV Power Control Architecture with Single FPGA



In addition to the benefits of a reduced number of parts, the new architecture reduces the number of hardware and firmware interfaces. It also provides opportunities for complete system simulation and auto-code generation not possible with the existing architecture. The development and implementation of these algorithms are described in the following sections.

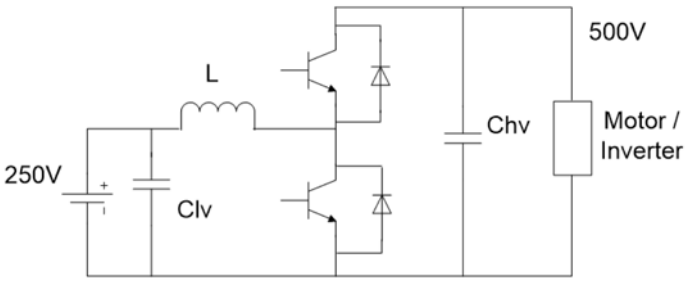
High-Frequency VVC (DC-DC) Converter

Introduction to High-Frequency VVC Converter

The VVC converter provides bidirectional power flow between the battery and the MG inverters. A standard design uses an IGBT half bridge with a 200 µH inductor, where the lower transistor is switched to "boost" the voltage from the battery to the motor inverter. Conversely, to charge the battery, the upper transistor is switched to "buck" the voltage from the motor inverter to the battery. The "boost" mode is analyzed below.

The battery is 250 V and the VVC can provide up to 650 V at 50 kW peak. The battery has a high-frequency capacitor Clv (400 µF) in parallel and there is another high-frequency capacitor Chv (2,000 µF) at the VVC boost output as shown in Figure 4.

Figure 4. VVC or Bidirectional DC-DC Converter



During boost mode, the lower switch is modulated with the duty cycle D resulting in the voltage gain as shown in Equation 1.

Equation 1.

$$\frac{V_{hv}}{V_{lv}} = \frac{1}{1-D}$$

Given the following operating conditions:

P_{OUT} = 50 kW

V_{lv} = 250 V

V_{hv} = 500 V

f_{switch} = 10 kHz

We get

$$\frac{V_{hv}}{V_{lv}} = 2$$

and D = 0.5

Equation 2 shows the average current in the inductor:

Equation 2.

$$I_{Lave} = \frac{P_{OUT}}{V_{lv}} = \frac{50\ kW}{250\ \text{V}} = 200\ A$$

Equation 3 shows the peak to peak ripple current in the inductor:

Equation 3.

$$I_{\Delta p-p} = \frac{V_L}{L}\frac{D}{f_{switch}} = \frac{250\ V}{200\ \mu H}50\ \mu\text{s} = 62.5\ A$$

Equation 4 shows the output voltage ripple:

Equation 4.

$$\Delta V_{OUT} = \frac{I_{hv}}{C_{hv}} \frac{D}{f_{switch}} = \frac{100\ A}{2000\ \mu h} 50\ \mu s = 2.5\ V$$

Proposed Higher Frequency Design

A trend in the power electronics industry is faster switching, which allows reduction of inductance and capacitance values to achieve equivalent voltage and current ripple. One barrier to switching faster is increased transistor switching losses. Application of IGBTs optimized for lower switching losses or MOSFETs can mitigate these losses, but the result will usually be some increase in transistor losses. SiC MOSFETs, with dramatically reduced switching losses are becoming available and will remove this barrier. While price is still an issue for SiC, the general trend of cost reductions is expected to continue to where SiC devices will compete with standard silicon.

Another barrier to higher frequency switching is the higher bandwidth needed for acceptable current control. This increased bandwidth is a challenge for MCU-based solutions, especially if multiple functions are to be implemented with one processor. FPGA control can easily provide the bandwidth required for this application, even if multiple control functions are implemented on one device. Proposed is a f_{switch} = 50 kHz switching frequency (5X increase). This results in a proportional reduction in inductance and capacitance values to get the same ripple current and voltage. Along with this reduction in component value, there is a similar reduction in size and cost. The existing and proposed values are tabulated in Table 2:

Table 2. Comparison of VVC 10 kHz and 50 kHz Designs

Component	f_{switch} = 10 kHz	f_{switch} = 50 kHz	Size Reduction	Potential Cost Reduction
L	200 µH	40 µH	5X	$200-> $100
C_{hv}	2,000 µF	400 µF	5X	$300->$100
C_{lv}	400 µF	80 µF	5X	$100->$50
I_{pp}	62.5 A	62.5 A	-	
V_{pp}	2.5 V	2.5 V	-	
Transistor Losses				
IGBT Losses	500 W	1,100 W	-	-
Si MOSFET Losses	600 W	750 W	-	-
SiC MOSFET Losses	150 W	250 W	2X	Higher cost but declining

Figure 5 shows the simulation of 10 kHz and 50 kHz VVC designs in Simulink, and its waveforms are shown in Figure 6. Simulated here is the response to a 500 V voltage command. The waveforms show equivalent current and voltage ripple with the two designs. In addition to the benefit of lower reactive component values, much quicker response in output voltage for 50 kHz switching design is also evident.

Figure 5. VVC Simulink Simulation

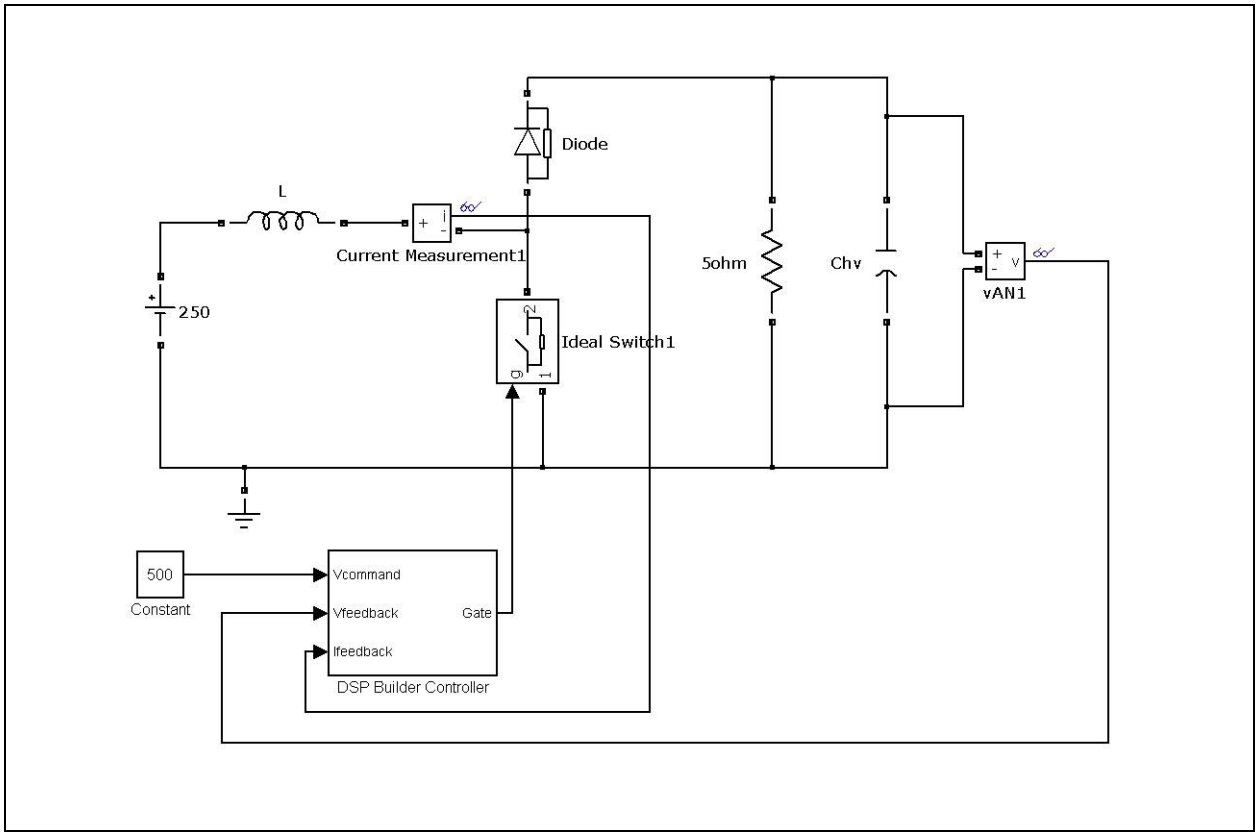


Figure 6. VVC with 10 kHz and 50 kHz Switching

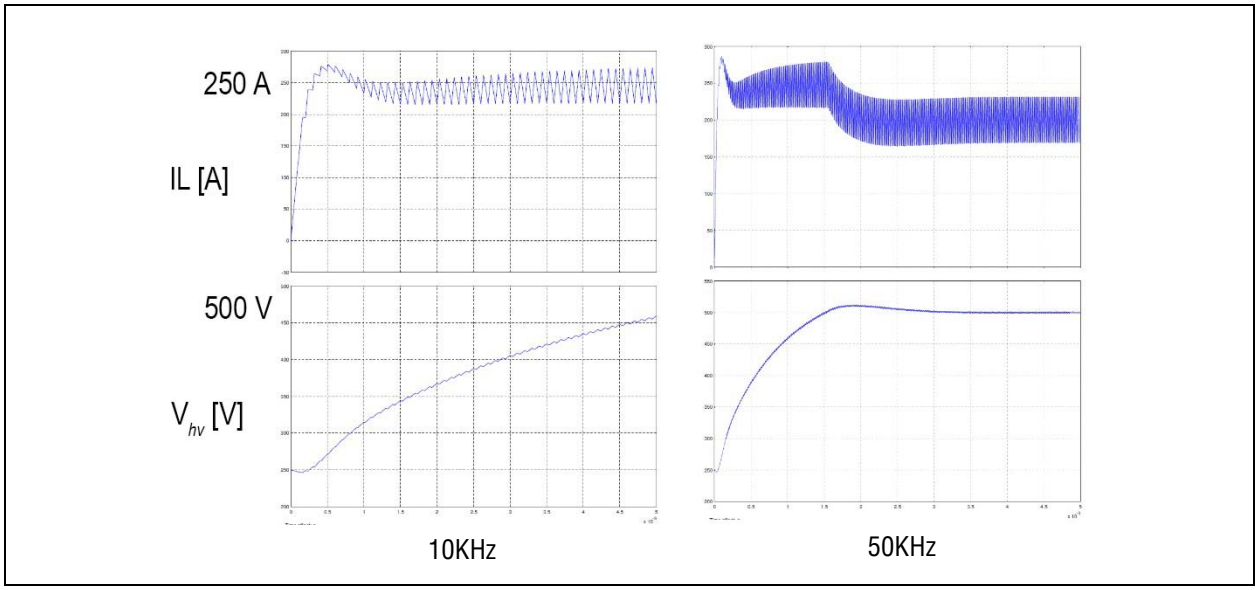
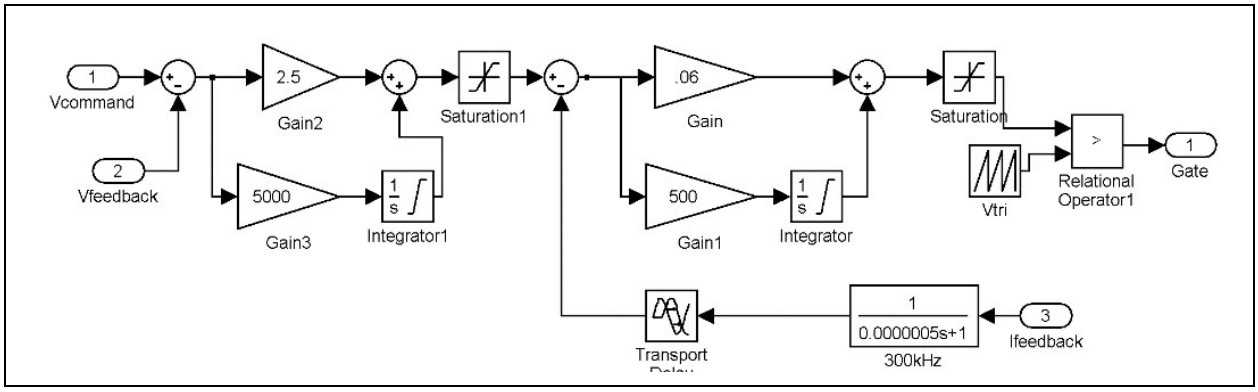


Figure 7 shows the controller design that uses an inner current loop and outer voltage loop.

Figure 7. VVC Current and Voltage Controller (Simulink Model)



For the 10 kHz design with larger 200 µH inductor, the bandwidth of the current loop is only f_{RW} = 1 kHz. The 50 kHz design with the smaller 40 µH inductor requires higher bandwidth to properly control the current and f_{RW} = 5 kHz was chosen for the current loop bandwidth. To maintain good stability, a rule of thumb states that the maximum control delay T_{delay} is given in Equation 5.

Equation 5.

$$T_{delay} \leq 0.1 \times \frac{1}{f_{BW}}$$

Where the 10 kHz design only requires $T_{delay} \leq 100$ µs, the 50 kHz design requires $T_{delay} \leq 20$ µs, which may be problematic for a MCU-based design, especially if it is controlling multiple functions.

FPGA Implementation with Altera DSP Builder

Figure 8 shows the digital controller for the VVC converter that was implemented with the Altera DSP Builder.

Figure 8. VVC DSP Builder Control

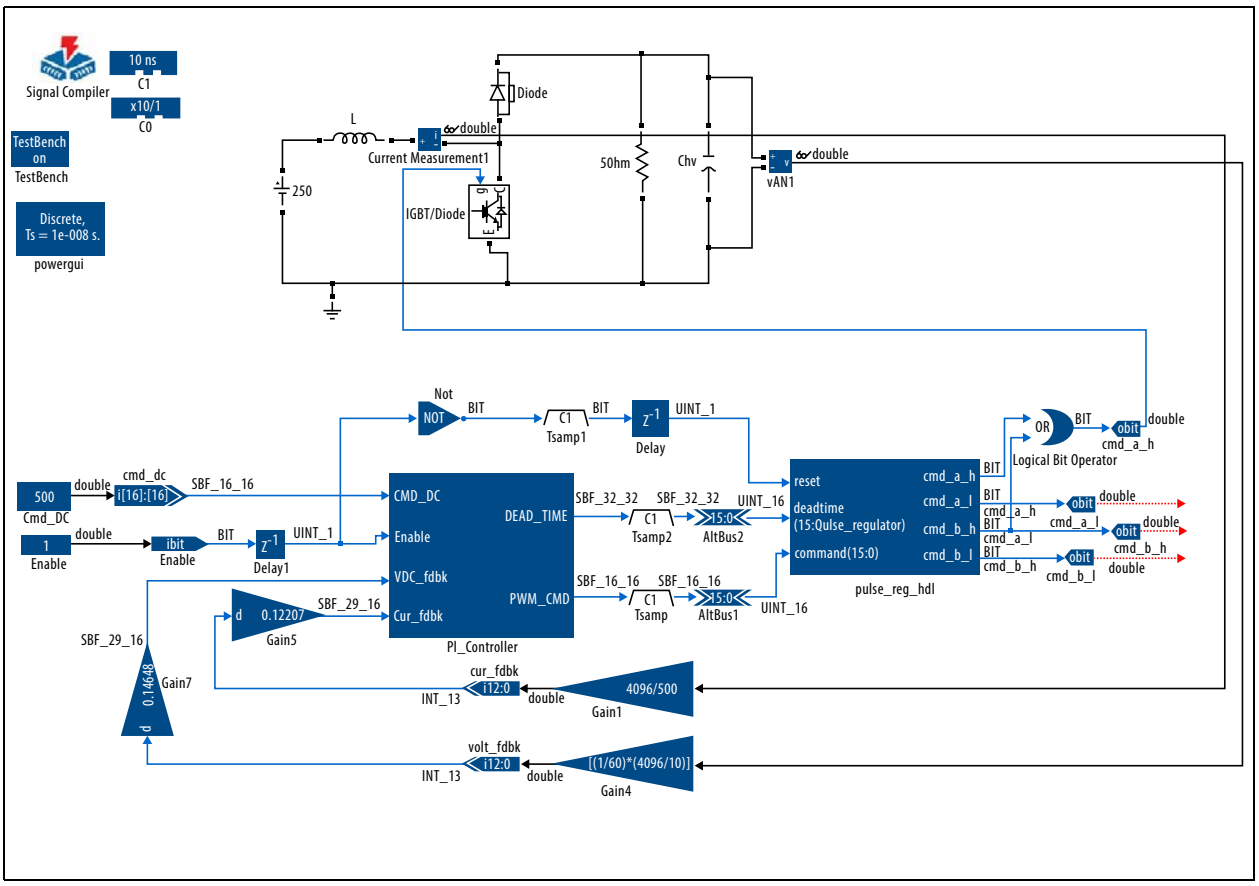
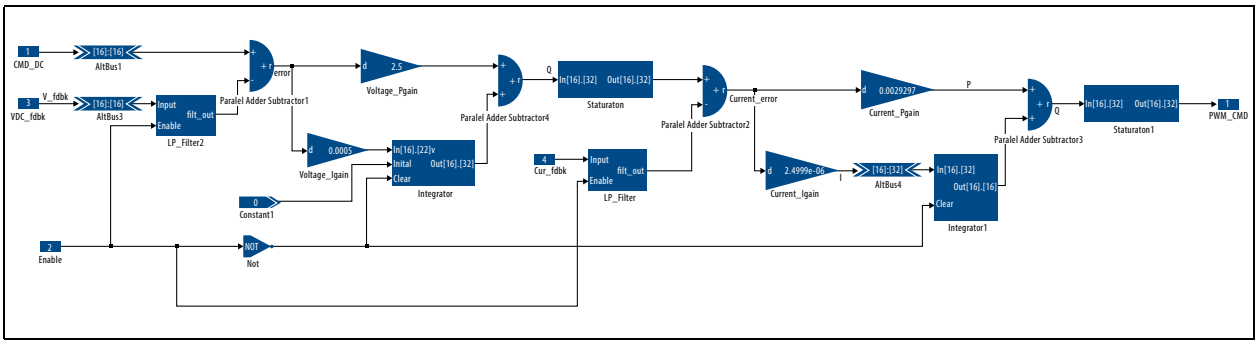


Figure 9 shows the current and voltage proportional integral (PI) regulators.

Figure 9. DSP Builder Current and Voltage PI Controllers



This DSP Builder design uses a 100 MHz clock for the pulse-width modulation (PWM) pulse generator and 10 MHz clock for the PI controllers. The design has three control delays in the 10 MHz clock region, resulting in a total control delay of T_{delay} = 0.3 µs, which has no effect on the control response as shown in Figure 10. Figure 11 shows the unstable response with simulated MCU delay of 20 µs.

Figure 10. VVC Control DSP Builder

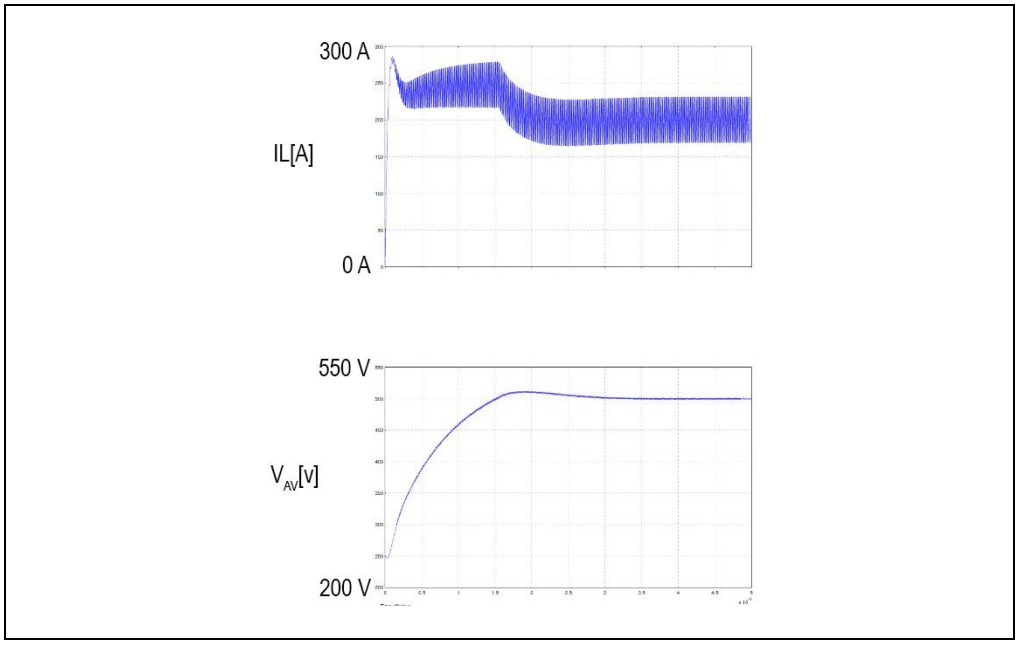
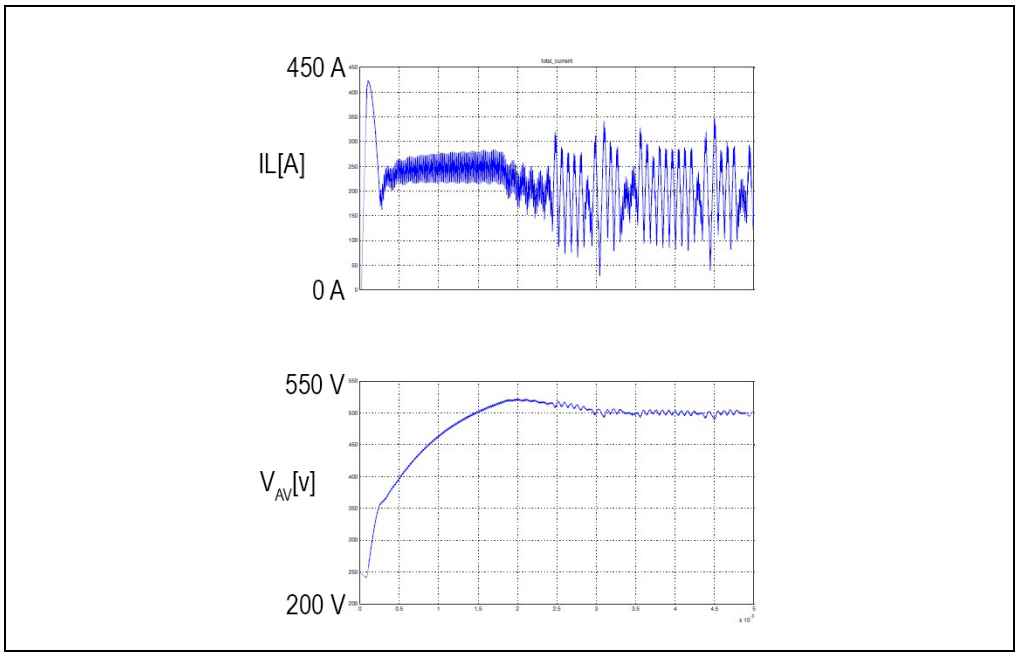


Figure 11. Simulated 20 µs MCU Delay



FPGA Implementation Resources

The VVC design uses a low-cost, automotive-grade Altera Cyclone® IV FPGA. The design resources are tabulated in Table 3.

Table 3. VVC FPGA Resources

VVC FPGA Controller	Fixed Point 26 Bit Word
LEs (logic elements)	2,344
Registers	970
Multipliers (18 x 18)	34

IPM Motor Control with DTFC-SVM

Introduction to Motor Control with DTFC-SVM

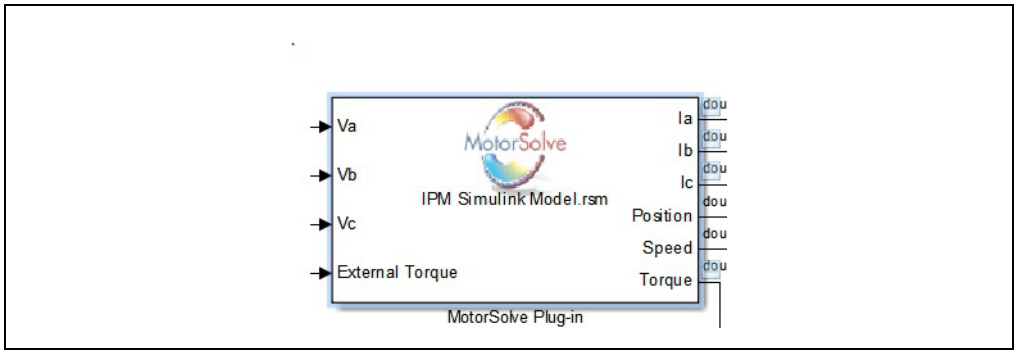
Use of IPMs is common in both electric and hybrid electric vehicles due to their ruggedness and increased torque capability. (2) However, an undesirable characteristic of IPMs is their inherent torque ripple pulsations that can degrade performance and reliability. (3) (4) We evaluated the torque ripple with an IPM motor model created by Infolytica Corporation that can be run in real time on an FPGA. DTFC-SVM is developed and compared to standard field-oriented control (FOC) or direct quadrature (DQ) control. When DTFC-SVM implemented with DSP Builder is used with the torque estimator of the IPM model, it reduces torque ripple. This section also discusses control performance of an FPGA versus an MCU implementation.

IPM Motor Model

The IPM motor model developed for a popular hybrid electric vehicle uses finite element analysis (FEA) and response surface modeling (RSM) for use in The MathWorks' MATLAB or Simulink software. The model accurately predicts the torque ripple of the IPM, which can be used as feedback in a control system to reduce the magnitude of the ripple.

Figure 12 shows the IPM model that may be used in Simulink as a "MotorSolve" plug-in. It is also available as a VHDL file that can be implemented on an FPGA or imported into the Altera DSP Builder software using the tool's **HDL Import** feature.

Figure 12. IPM Motor Model



FOC Control

Figure 13 shows how we simulated the IPM motor model in Simulink with a standard FOC (or DQ) control [9] to assess torque response and ripple. To produce torque, we apply a current reference to the quadrature (I_q) input of the DQ current controller.

Figure 13. IPM Motor with FOC Control

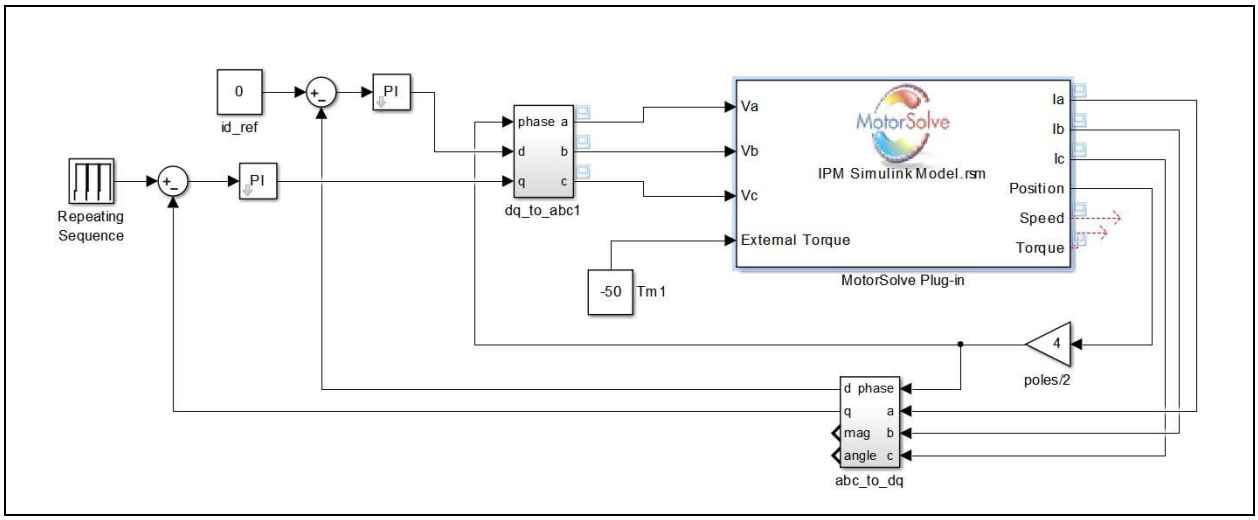
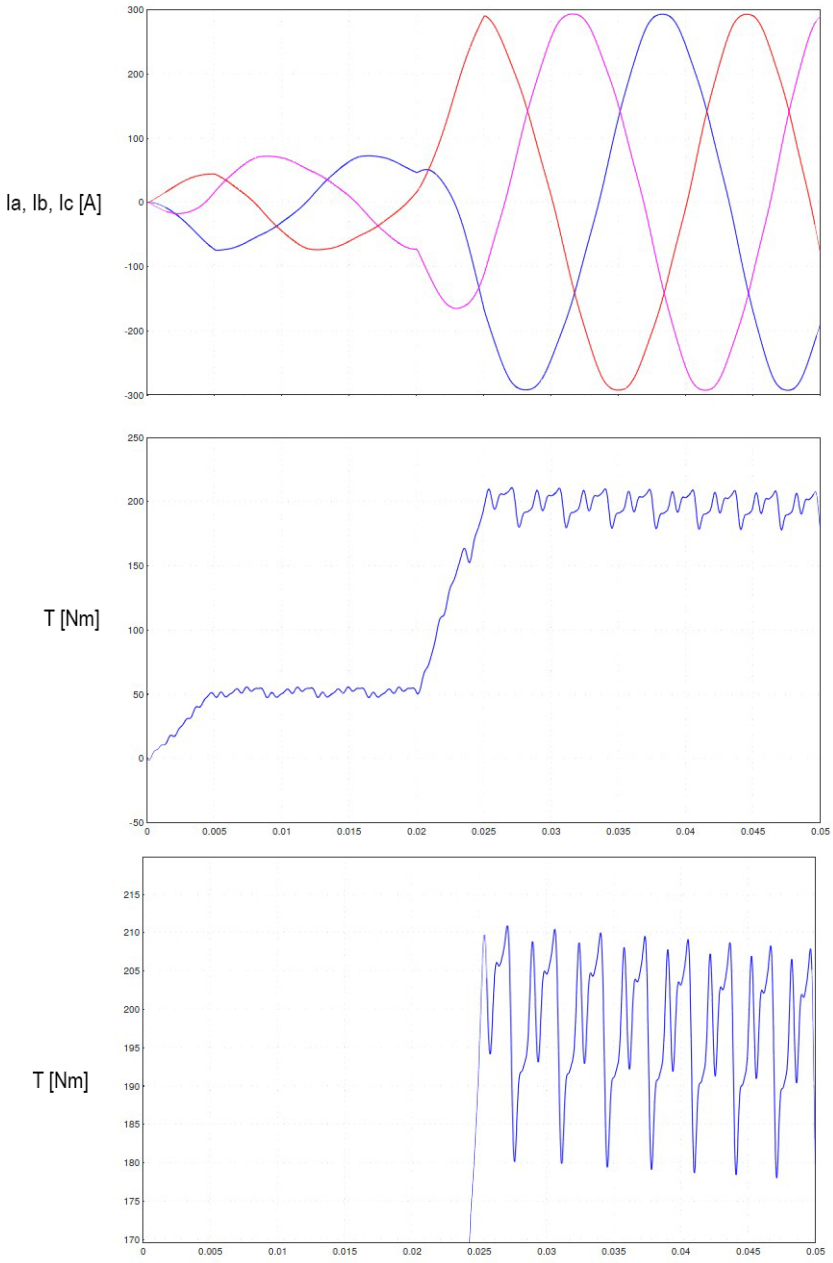


Figure 14 shows the vector current reference steps from 89 A to 356 A to produce 50 Nm and 200 Nm torque, respectively. Also shown is the torque on a magnified scale showing torque ripple of $\Delta T_{p\text{-}p}$ = 30 Nm. It should be noted that this torque ripple is inherent in the machine and not due to control effects (for example, DTC hysteresis control [8]).

Figure 14. Current and Torque with FOC Current Control

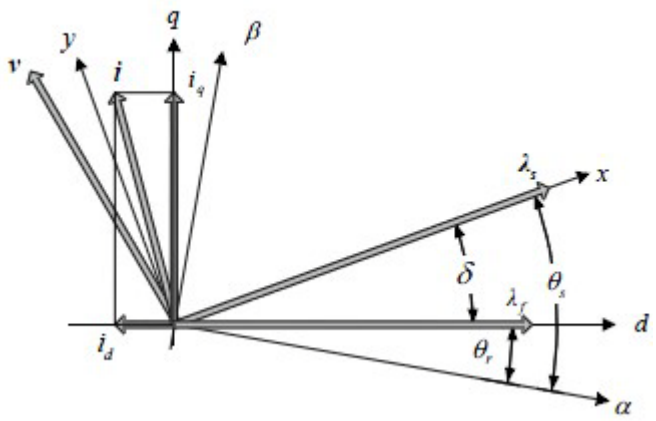


DTFC-SVM Control

DTFC-SVM has recently been shown to improve torque output and response. SVM is used versus standard PWM because of its benefits, including lower harmonics and switching losses. Implemented here is a "modified" DTFC-SVM [5] [6] [7] that eliminates high-frequency torque and flux ripple due to hysteresis control.

The stator flux linkage vector φ_s and the rotor (magnet) flux linkage vector φ_f can be drawn in the rotor flux (dq), stator flux (xy) and stationary (αβ) reference frames as shown in Figure 15. The angle between the stator and rotor flux linkages δ is the load angle.

Figure 15. Stator and Rotor Flux Linkages in Different Reference Frames



The well-known machine equations [5] for an IPM with pole saliency ($L_d \neq L_q$) are shown in Equation 6:

Equation 6.

$$\varphi_d = L_d i_d + \varphi_f$$
$$\varphi_q = L_q i_q$$
$$v_d = R_s i_d + p\varphi_d - \omega_r \varphi_q$$
$$v_q = R_s i_q + p\varphi_q - \omega_r \varphi_d$$
$$T = 3/2(p(\varphi_d i_q - \varphi_q i_d))$$

Where L_d and L_q are the direct and quadrature inductances, ω_r is the electrical rotor speed, p is the number of pole pairs, and T is the electromagnetic torque. Furthermore, it can be shown that the torque T of an IPM in the xy reference frame is given by Equation 7:

Equation 7.

$$T = \frac{3}{2} p |\varphi_s| i_y$$

The voltages in the stator flux (xy) reference frame, with $\varphi_y = 0$, can be in Equation 8:

Equation 8.

$$v_x = R_s i_x + \frac{d\varphi_s}{dt}$$
$$v_y = R_s i_y + \omega_r \varphi_s$$

This shows that the amplitude of the stator flux vector can be regulated by the x component of the stator voltage and the torque can be indirectly regulated by the y component of the stator voltage.

Equation 9 shows that the stator flux linkage and torque are estimated in the stationary (αβ) frame:

Equation 9.

$$\varphi_\alpha = \int (v_\alpha - R_s i_\alpha) dt + \varphi_{\alpha 0}$$

$$\varphi_\beta = \int (v_\beta - R_s i_\beta) dt + \varphi_{\beta 0}$$

$$|\varphi_s| = \sqrt{\varphi_\alpha^2 + \varphi_b^2}$$

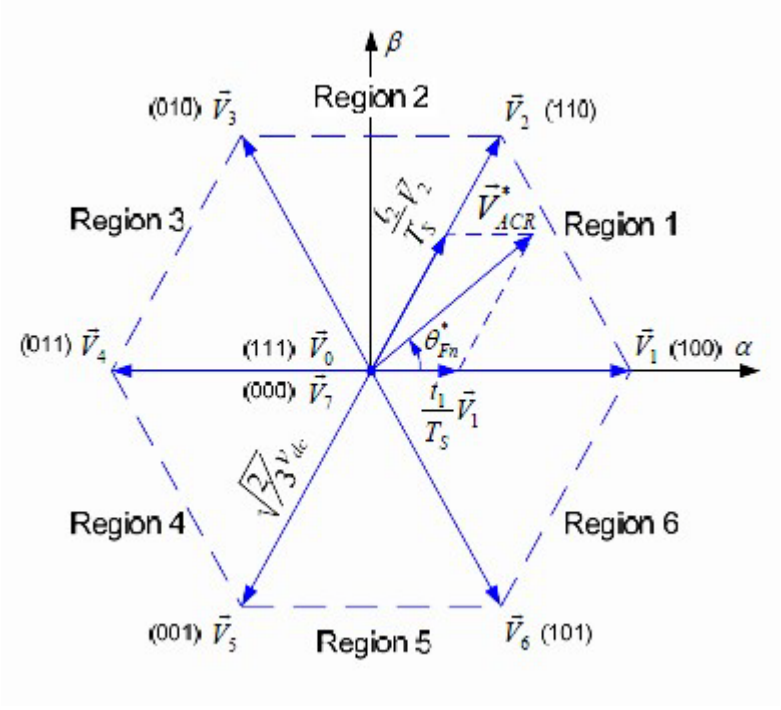
$$\theta_s = \tan^{-1}\left(\frac{\varphi_\beta}{\varphi_\alpha}\right)$$

$$T = \frac{3}{2} p(\varphi_\alpha i_\beta - \varphi_\alpha i_\beta)$$

SVM Description

Unlike PWM, SVM does not have modulators for each phase. SVM uses a reference voltage vector to calculate modulation times for active and zero vectors. Figure 16 shows the vector sequence order that is chosen to minimize commutations and current ripple.

Figure 16. Space Vector Modulation



DTFC-SVM Simulation

Figure 17 shows how we simulated the DTFC-SVM control with the IPM model in Simulink. The Simulink model has separate PI controllers for torque and flux. The torque and flux estimator implements the previously discussed equations. Simulink functions for coordinate transformations and SVM were used. The DTFC control can use either the torque estimator or the torque output from the IPM model. The IPM model torque is used here because it accurately models the torque ripple.

Figure 17. IPM motor with DTFC-SVM Control

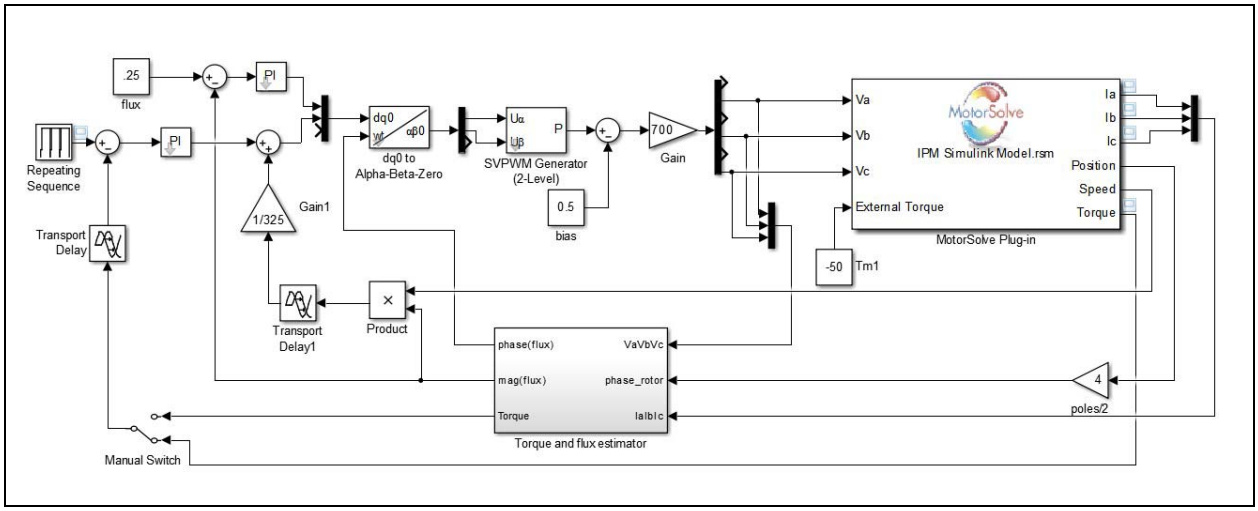
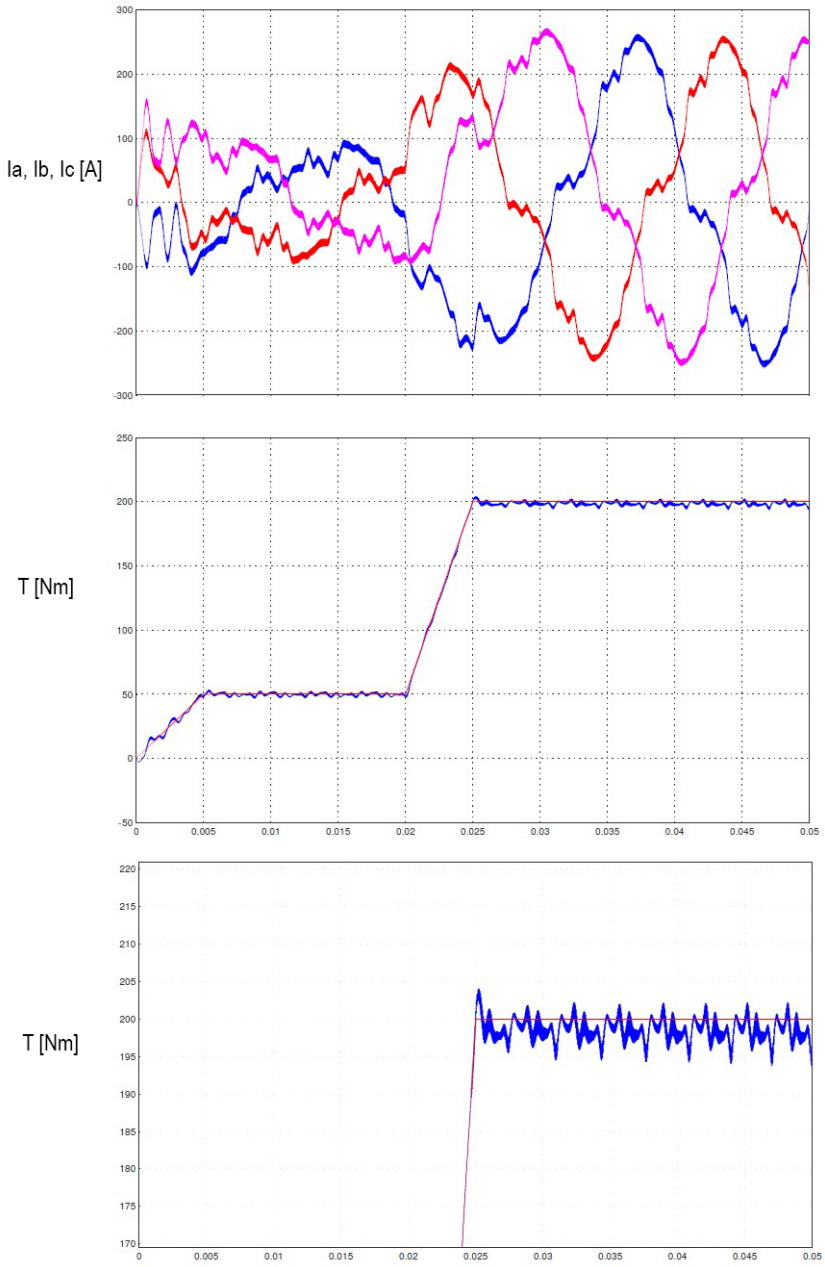


Figure 18 shows currents and torque (blue) in response to a change of torque command (red) from 50 Nm to 200 Nm. The magnified torque shows reduced ripple $\Delta T_{p\text{-}p}$ = 5 Nm giving a 6X reduction of torque ripple. This reduction in torque ripple is due to both the accurate IPM model and implementation of a high- bandwidth control loop that is discussed in the following section.

Figure 18. Current and Torque Response for DTFC-SVM Control



DTFC-SVM Implementation with Altera DSP Builder

Figure 19 shows the DTFC-SVM controller using the Altera DSP Builder. The IPM motor model is available as a Simulink block or VHDL file. Figure 20 shows the DTFC PI control block. Figure 21 shows the torque and flux estimator block, where DSP Builder math functions, such as sine, cosine, vector arctan, and vector magnitude were utilized.

Figure 19. DTFC-SVM Implemented with Altera DSP Builder

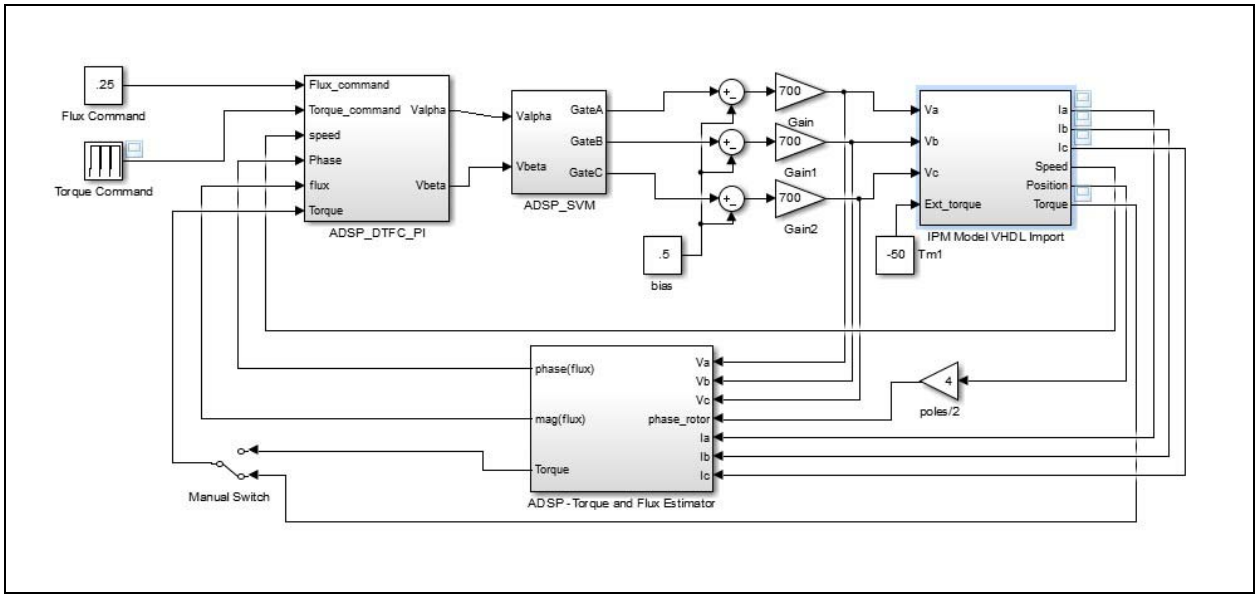


Figure 20. DTFC PI Controller Implemented with Altera DSP Builder

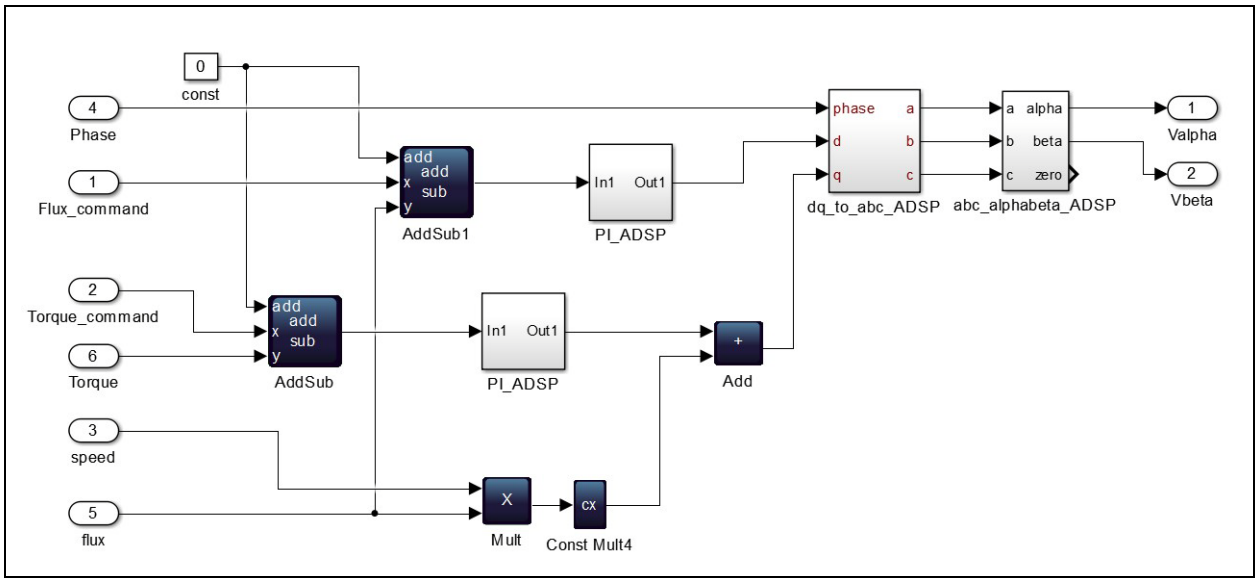


Figure 21. DTFC Torque and Flux Estimator Implemented with Altera DSP Builder

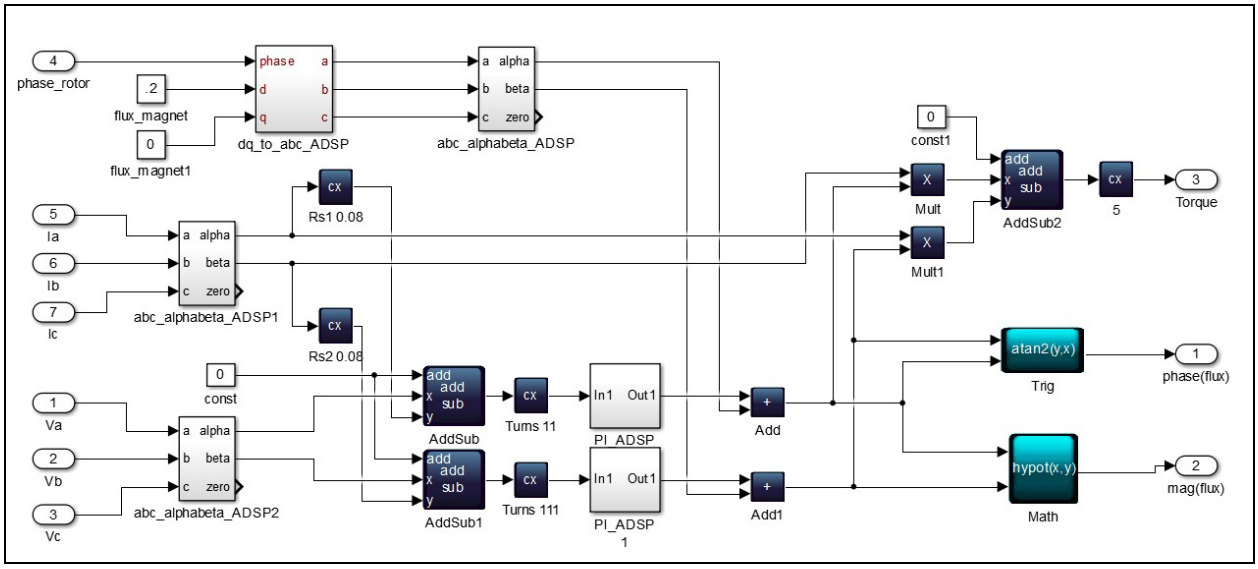
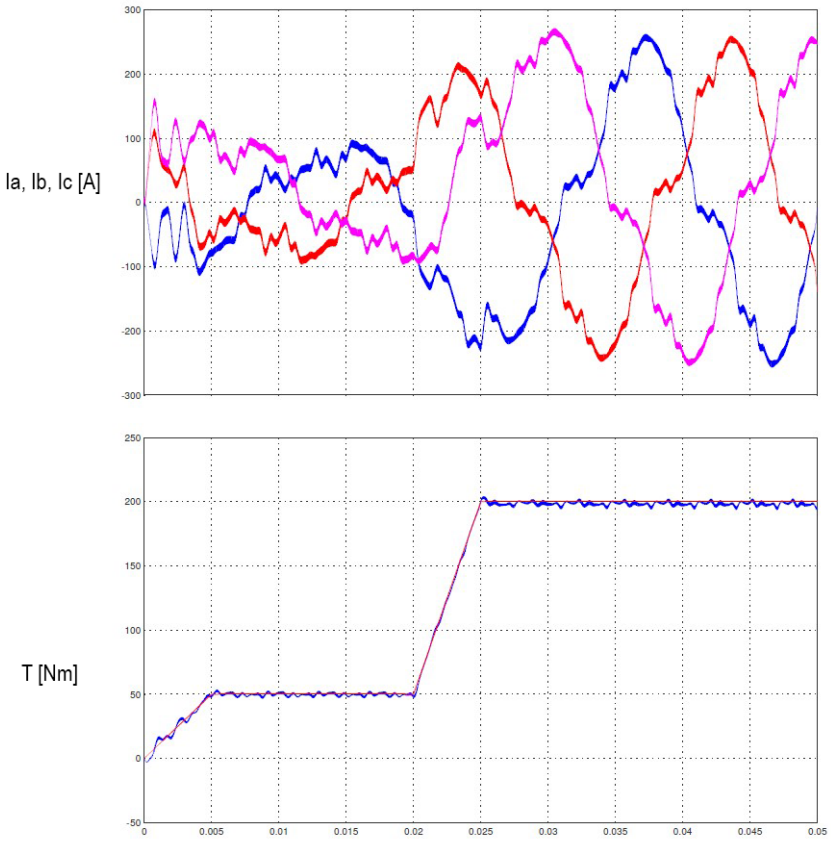


Figure 22 shows the torque response and ripple are equivalent to the linear Simulink model developed previously. The added delay of control response from the VHDL implemented is <5 µs, which includes DTFC, SVM, torque and flux estimator, and VHDL implementation of the IPM model. This delay (equivalent to a control update rate of 200 kHz) adds a negligible phase lag to the control loop and therefore has no impact on stability.

Figure 22. Altera DSP Builder DTFC Control



Comparison with MCU-Based Implementation of DTFC-SVM

If the DTFC-SVM algorithm, which includes the IPM model is implemented in a MCU, it is estimated that the delay to execute one control calculation would be approximately 70 µs phase delay (30 µs for DTFC-SVM and 40 µs for IPM motor model). To estimate the impact of this delay, a transport delay is added to the DTFC-SVM control loop, as shown in Figure 23. Figure 24 shows simulation results where it is evident that the added control delay is increasing the amount of torque ripple. This performance would be even more limiting if multiple control functions are required to implement this algorithm in the MCU.

Figure 23. Added Control Delay to Simulate MCU

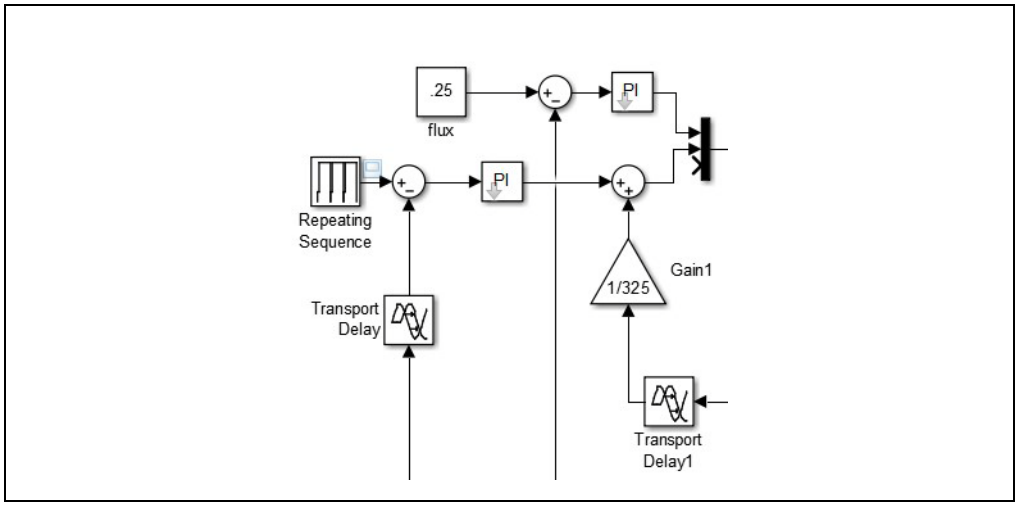
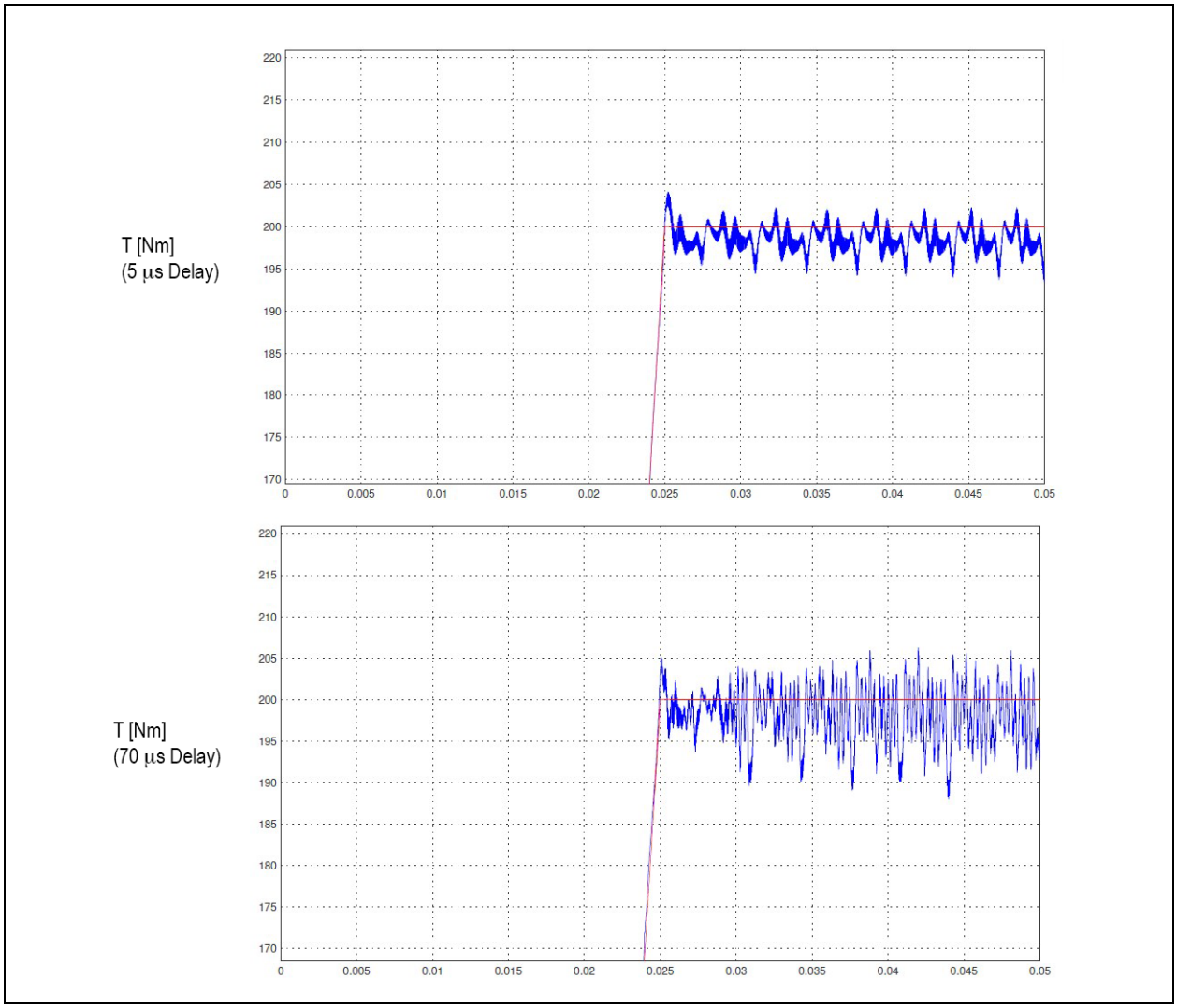


Figure 24. Effect of MCU Control Delay on Control Stability



FPGA Implementation Resources

The DTFC-SVM design uses a low-cost, automotive-grade Cyclone IV FPGA. The design resources are tabulated in Table 4.

Table 4. DTC-SVM Controller FPGA Resources

DTFC-SVM FPGA Controller	Single-Precision Floating Point
LEs	19,560
Registers	59
Multipliers (18x18)	27

FOC Generator Controller

The generator controller design uses the FOC design example available with DSP Builder. The design is a fairly standard FOC algorithm, and therefore not discussed in detail in this white paper. The design resources are tabulated in Table 5.

Table 5. FOC Generator Controller FPGA Resources

FOC Generator FPGA Controller	Single-Precision Floating Point
LEs	7,976
Registers	20
Multipliers (18x18)	27

Simplified Hybrid EV Power Control FPGA Resources

The complete design (including VVC, DTFC-SVM IPM motor control, and standard FOC generator control) use a low-cost, automotive-grade Cyclone IV FPGA (EP4CE40). The design resources are tabulated in Table 6.

Table 6. Complete Hybrid EV Power Control FPGA Resources

	VVC Controller	DTFC-SVM IPM Motor Controller	FOC Generator Controller	Complete Design	Cyclone FPGA Resources
LEs	2,344	19,560	7,976	29,880	39,600
Multipliers (18 x 18)	34	59	20	113	116
M9K Memory	—	27	17	44	126
f_{MAX}	—	101 MHz	98.4 MHz	98.4 MHz	—

Conclusion

This paper investigates the benefits of FPGA control for automotive power electronics. We have developed a simplified control architecture in which multiple motor control functions and VVC DC-DC converter control are implemented in one FPGA instead of multiple MCUs.

We have developed a controller for a high-frequency VVC DC-DC converter that makes possible smaller magnetics and capacitors. In addition, we have developed a high- performance DTFC-SVM motor controller to reduce torque ripple pulsations inherent in IPM motors. Both the VVC and DTFC-SVM controllers utilize high-bandwidth control that is enabled by the speed and parallel capability of FPGA control.

The VVC and DTFC-SVM controllers use the Altera DSP Builder, taking advantage of the following benefits—an integrated simulation environment with Simulink and DSP Builder, use of multiple floating-point functions to simplify the controller design, and auto-generation of code.

References

1. D. Maksimovic, et. al., *"Impact of Digital Control in Power Electronics,"* IEEE International Symposium on Power Semiconductor Devices and ICs" Kitakyusha, Japan, pp. 13-22, May 2004.
2. M. Olszewki, "*Evaluation of the 2007 Camry Hybrid Synergy Drive System,"* Oak Ridge National Laboratory, 2008.
3. H. Goto et al, *"Simulation of IPM Motor by Nonlinear Magnetic Circuit Model for Comparing Direct Torque Control with Current Vector Control,"* 2008 13th International Power Electronics and Motion Control Conference (EPE-PEMC 2008).
4. R. Cao, et al, *"Quantitative Comparison of Flux-Switching Permanent-Magnet Motors with Interior Permanent Magnet Motor for EV, HEV, and PHEV Applications,"* IEEE TRANSACTIONS ON MAGNETICS, VOL. 48, NO. 8, August 2012.
5. G. Foo, et al, *"Analysis and Design of the SVM Direct Torque and Flux Control Scheme for IPM Synchronous Motors,"* Conf. Rec.IEEE-PESC, June 2008, pp. 50-56.
6. A. Daghigh, et al, *"A Modifid Direct Torque Control of IPM Synchronous Machine Drive with Constant Switching Frequency and Low Ripple in Torque,"* Proceedings of ICEE 2010, May 11-13, 2010.
7. G. D. Andreescu et al, *"Combined Flux Observer with Signal Injection Enhancement for Wide Speed Range Sensorless Direct Torque Control of IPMSM Drives,"* IEEE Transactions on Energy Conversion, Vol 23, No 2, June 2008.
8. L. Tang et al, *"An Investigation of a Modified Direct Torque Control Strategy for Flux and Torque Ripple Reduction for Induction Machine Drive System with Fixed Frequency,"* IEEE Proceedings, 2002.
9. D. Novotny, T. Lipo, *"Vector Control and Dynamics of AC Drives,"* Clarendon Press – Oxford, 1997.

Acknowledgements

- Jason Katcha, President, All Digital Power, LLC
- Michael Parker, Sr. Manager, Altera DSP Product Planning
- Daisuke Yoshida, Strategic Marketing Manager, Altera Automotive Business Unit

- Ben Jeppesen, Motor Control Specialist, System Solutions Engineering, Altera Industrial and Automotive Business Unit
- Clive Davies, Automotive System Architect, Altera Automotive Business Unit

Document Revision History

Table 7 shows the revision history for this document.

Table 7. Document Revision History

Date	Version	Changes
December 2013	1.0	Initial release.