Use of FPGAs to develop energy-saving DC-DC control

Ben Jeppesen, Intel FPGA, UK, ben.jeppesen@intel.com
Ge Gao, Imperial College, London, UK, ge.gao13@imperial.ac.uk
Jason Katcha, Half Moon Ventures, USA, jason@presentpowersystems.com

Abstract

FPGAs provide complete flexibility in the digital control of switching power converters. This paper extends an existing DC-DC converter reference design with PWM control to create a new DC-DC converter controller combining PI control for load voltage control with hysteresis control for inductor current control. Model-based design is used to develop from offline simulation to real-time implementation with automatic code generation. The hysteresis controller is shown to reduce switching losses, improve inductor current and voltage control, and use fewer FPGA resources compared to the original PWM controller.

1. Introduction

1.1. DC-DC converters, hysteresis control and use of FPGAs

Switch-mode power converters are ubiquitous in modern electronic devices. Most are small buck converters, stepping down an unregulated input supply to a reliable DC voltage supply. However, increasingly, such DC-DC converters are being used to control higher-power systems in applications like grid energy storage or energy flow in electric vehicles. These systems may be operating more of the time, have widely varying loads and bidirectional operation, requiring more sophisticated controllers to achieve sufficient voltage regulation and energy efficiency.

Most converters use voltage-mode control, where the measured error in output voltage is input to a controller which produces the duty-cycle signal for PWM output signals, which drive the gates of MOSFETs (low power systems) or IGBTS (higher-power, but slower response). Current-mode controllers use an inner current control loop which uses the current feedback value to trigger the second switching event per PWM cycle, instead of a PWM carrier signal [1]. The term hysteresis control is used to describe a controller where the switching transistors are triggered directly by the voltage error crossing pre-set thresholds [1][2]. While this seems an ideally simple controller, there are difficulties in practice, including a varying switching frequency, making it more difficult to apply EMC measures [1]. Such hysteresis controllers can operate at high frequencies and are used in some low-power power supplies for microprocessors [2]. Due to the low-cost nature of such low-power supplies, the controllers are often fixed and use analogue electronics.

FPGAs are a programmable logic technology that enables complex digital circuits to be realized, such as custom 'system-on-chips' with interfacing IP, DSP functions and a soft or hard processor in the chip. FPGAs are often used in research of power converters because the control functions can be executed faster in dedicated logic than in standard microprocessors.

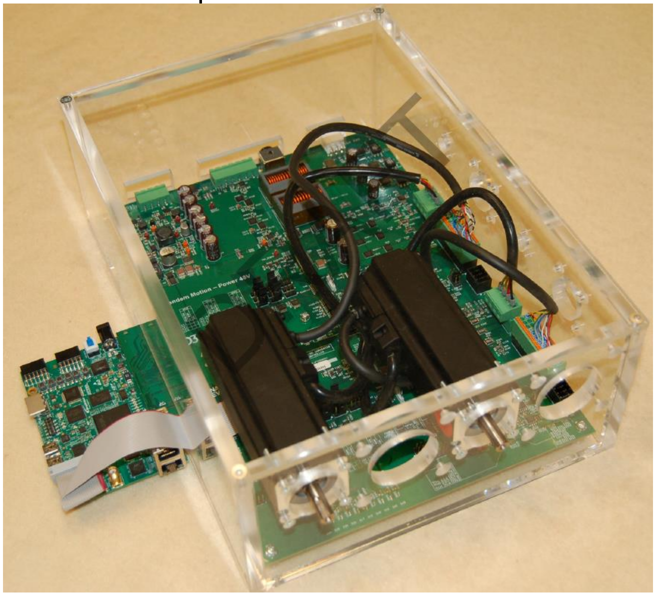

Fig. 1. MAX 10 FPGA with Tandem Motion-Power 48 V

Power converters often use dedicated control chips based on PI control with PWM outputs. This limits the power conversion efficiency by restricting the control algorithms to standard types. An FPGA enables more specialized control algorithms to be implemented. For example, an FPGA can be used to prototype the control logic for a power control chip intended to combine the benefits of voltage hysteresis control and a fixed converter switching frequency [3].

An FPGA can also be used to simulate in real-time both the control logic and a model of the power electronics and load, using a model-based design to generate the FPGA code automatically [4][5].

A typical power converter chip acts alone, without direct knowledge of the load. Using an FPGA, the DC-DC control can be one part of a larger FPGA-based controller and share its information with other parts of the system. This brings the potential for saving energy directly through more advanced converter algorithms and indirectly through prediction of load requirements. Intel FPGA have produced a ‘Drive-on-Chip’ reference design to demonstrate how a single FPGA can control multiple motors and a two-phase bidirectional DC-DC converter simultaneously [6][7] (Fig. 1).

Intel FPGA’s DC-DC converter reference design uses a voltage PI controller to drive an inductor current PI controller, which sets the duty cycle for a PWM waveform generator [5]. In this paper, we explore the benefits of using the FPGA to replace the inner PI current control loop with a hysteresis current controller. This has potential benefits of improving the inductor current response time, enabling faster voltage control, and improving efficiency through reduced switching events.

1.2 Aims of the paper

The paper contains two main sections, ‘2. Method’ and ‘3. Results and Discussion’.

The Method section describes the design process, starting from a simulation of the electronics, then adding and optimizing realistic sensor signal processing, tuning PWM controllers for 10kHz and 50kHz converters, then introducing the hysteresis controller to replace the current PI controller and PWM generator. Measures of performance and efficiency are described. The method of automatic code generation and important parameter settings in the DSP Builder tool are reported, targeting a small MAX 10 FPGA.

The Results and Discussion section reports numeric results from each stage of the process and comments on the outcomes.

Conclusions and Further Work sections complete the paper.

2. Method

2.1. Electronic simulation

The simulation of power electronics (Fig. 2.) is based on the model introduced in [4] and [5]. It models a 250V battery pack in an electric vehicle, connected to a 500V DC link that would be connected to inverters for motors and generators. The converter is a bi-directional boost/buck converter to enable recuperation of energy from a generator as well as driving of motors. It is designed for power flows up to 50kW, in which case the average inductor current is 200A and the load current is 100A. The load current is created by an ideal current source to enable controlled load current patterns to be applied.

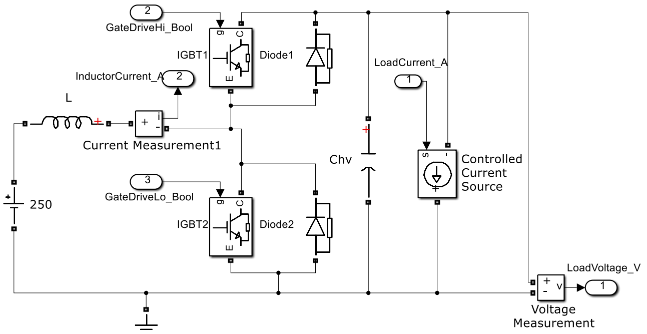

Fig. 2. Single-phase bi-directional DC-DC converter simulation using SimPowerSystems.

Compared to the models in [4] and [5], the simulation has been made more realistic by the introduction of IGBT models instead of ideal switches, with parallel diodes. The IGBT models are from SimPowerSystems in MATLAB 2013a, with default parameters (Table 1.).

Table 1. IGBT and diode parameters

Parameter	IGBT	Diode
Resistance Ron (Ohms)	0.001	0.001
Inductance Lon (H)	0	0
Forward voltage Vf (V)	1	0.8
Current 10% fall time Tf (s)	1e-6	n/a
Current tail time Tt (s)	2e-6	n/a
Snubber resistance Rs (Ohms)	1e5	500
Snubber capacitance Cs (F)	inf	250e-9

Passive component parameters are chosen in two alternative sets, designed for PWM based control at 10kHz and 50kHz respectively (Table 2). In both cases, in steady state operation with 50kW output, current and voltage ripple are calculated to be 62.5A and 2.5V respectively, peak to peak [4].

The capacitors are initialized to 250V. The simulation is run with a fixed step time of 5e-8s (20MHz) to be compatible with the simulation of FPGA control with 20MHz clock. SimPowerSystems uses a Tustin solver while Simulink uses ode3 (Bogacki-Shampine).

Table 2. DC-DC converter passive components

Parameter	**10kHz**	**50kHz**
Inductance L (H)	200e-6	40e-6
Load capacitance Chv (F)	2000e-6	400e-6

2.2. Sensor signal processing

A practical limitation in real control systems is the phase lag and delay (latency) caused by filtering of feedback signals. However, filtering is usually necessary to reduce noise. In digital control, filtering is required to reduce aliasing of higher-frequency analog signal content into the limited digital bandwidth.

When using FPGAs to control power electronics, it is convenient to use sigma-delta ADCs on the high-voltage side, and transmit the high frequency 1-bit digital signal over isolated connections to the FPGA on the low-voltage side. The FPGA is ideal for implementing digital filtering to resample the signal at a lower rate but higher resolution.

Intel FPGA's Drive-on-chip reference design uses sigma-delta ADCs that generate an output at 20MHz. Sinc3 filter IP in the FPGA is used to generate a 16-bit signal for use in control software [7]. Sinc3 filters are recommended by ADC manufacturers, and their theory and even FPGA code can be found in some datasheets [8].

A single Sinc filter is an integrator running at the input frequency followed by resampling at 1/N of the original frequency and then a differentiator. N is usually a power of 2, for convenience.

Each output sample is the average of the last N 1-bit inputs. A Sinc3 filter is 3 integrators at the input sample rate followed by resampling at 1/N and three differentiators, but mathematically is the same as 3 Sinc filters in series. Each Sinc filter contributes an average delay of N/2 clock cycles to the data it is processing. A Sinc3 filter outputs a weighted average of the last 3N samples, giving stronger filtering and better resolution than a Sinc1 filter with the same N and output sampling frequency, at the expense of 3 times the signal delay.

It will be assumed that similar Sinc filter IP should be used to process incoming 20MHz sigma-delta signals in the DC-DC converter control model. The order (oversampling ratio, N) of the filter must still be chosen.

For PWM-based control, the controller needs to choose a PWM duty cycle for the next PWM period based on the average behavior over at least the last full PWM cycle. Ideally, N should be chosen to give the average over an integer number of previous PWM cycles. Given the starting point of 10kHz and 50kHz PWM, and 20MHz input, there are no powers of 2 for N that work exactly. A Sinc3 filter averages over a longer period than a Sinc1 so a mismatch of the window size will have less effect. Sinc3 filters are chosen with the parameters in Table 3 for voltage and current signals for PWM control at 10kHz and 50kHz:

Table 3. Sinc3 filters for PWM-based control

PWM frequency	**10kHz**	**50kHz**
Input sampling frequency	20 MHz	20 MHz
Oversampling ratio N	2048	512
Output sampling frequency	9.77kHz	39.1 kHz

For hysteresis control, the requirements are different. There is no PWM period and control decisions should be made as quickly as practical. The best control should result from minimizing the error over recent time from two sources:

1. Resolution: The resolution of a Sinc1 filter is N times better than the input resolution, while Sinc3 filter output resolution is N^3 times better.
2. Latency: the average delay produces an average error of the signal gradient multiplied by the latency.

Hysteresis control will be applied to current control only. The input current measurement range will be assumed to be +/-800A, which will be converted to 1-bit. The output resolution error will therefore be (1600/N) A or (1600/N^3) A respectively for Sinc1 and Sinc 3 filters. The current signal gradient in general will be +/-250V (difference between source and load voltage) divided by the inductance (200 µH or 40 µH), giving 1.25e6 A/s or 6.25 A/s respectively. Based on this, the total error can be calculated for Sinc1 and Sinc3 filters with different N.

2.3. PWM controller

The PWM controller uses cascaded PI controllers for load voltage and converter inductor current respectively (Fig. 3.). Saturation can be applied to the integrators and controller output to avoid integrator wind-up and limit current demand and duty cycles.

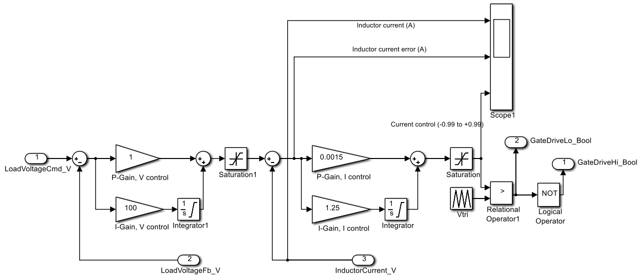

Fig. 3. PWM voltage and current controller, gains shown for 10kHz

The controllers were tuned independently for 10kHz and 50kHz power electronics. The procedure was:

1. Tune current controller: Set load current to zero and introduce fixed current demand to current controller instead of voltage controller output. Choose maximum P-Gain to maintain good damping (few oscillations), then increase I gain as much as possible while keeping good damping.
2. Check whether the controller is similarly stable for current demands at several levels within the operating range, in this case between -200 A and +200 A. Some reduction in the P and I gains may be necessary to achieve this.

Achieving good current control with the PI controller is difficult because the PWM duty cycle does not affect average inductor current directly, but rather the rate of change of average inductor current, and the response is affected by the source and load voltages, and the load current. Integral control is needed to compensate for the continuous effect of such unknown disturbances.

Once the current controller has been tuned, the voltage controller can be tuned by reconnecting the voltage controller to the current controller. A similar method of tuning the P-gain followed by the I-gain was used. Saturation of the voltage controller output was not used, however, so as not to mask the behavior of the PI controller.

To be able to compare the overall performance of difference controllers, the same test was used in all cases:

- The simulation starts with 250V initialized on the output capacitor, and zero load current. The simulation starts by showing how well the voltage controller raises the output voltage to 500VDC.
- Changing load currents are introduced as step inputs, in the order: +10 A, -10 A, +20 A, -20 A, +50 A, -50 A, +100 A, -100 A. The disturbance to the load voltage on each step, and other performance measures like current ripple and efficiency, can be made at each value of load current.

2.4. Hysteresis current controller

The hysteresis current controller replaces the PI current controller and PWM generator. It uses very simple logic to switch the power electronic switches when the inductor current error exceeds upper and lower bounds (Fig. 4.). To match the current ripple of 62.5 A peak-peak expected from the PWM control, upper and lower bounds of +30 A and -30 A can be set. To explore the effect on control and efficiency, other bounds were tried, up to +/-150 A.

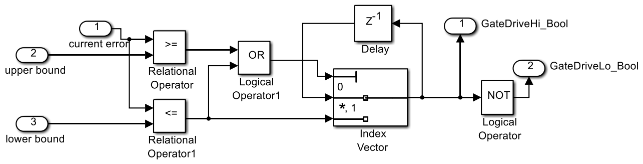

Fig. 4. Hysteresis current controller

2.5. Complete simulation model

The 500V command is connected to the controller, which in turn is connected to the power electronic simulation and then sensor signal conditioning (Fig. 5). For PWM control, both voltage and current signals were fed through the same Sinc3 filters, appropriate to the 10kHz or 50kHz PWM frequencies. For hysteresis control, the voltage signal conditioning was the same as for PWM, but the current signal conditioning used the faster acting filter derived later. A gate switch counter was added in addition to the efficiency calculations described next, as a check on average and total switching events.

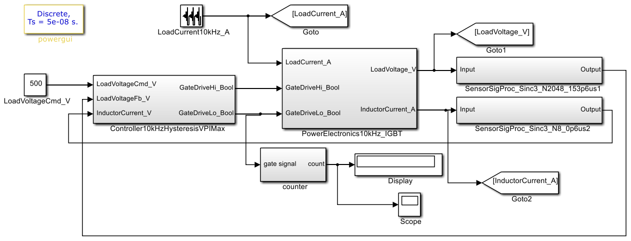

Fig. 5. Complete simulation model

2.6. Efficiency measurements

Measuring conversion efficiency is not easy when the simulation load currents are changing regularly. To reduce the effects of transients on the calculations, 1st order filters with 200Hz cut-off (time constant 0.8ms) were applied to the feedback signals. The calculations also take account of the changing polarity of the power flow to divide the output power by the input power. Fig. 6 shows the calculations and display.

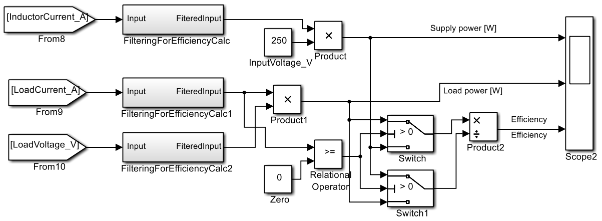

Fig. 6. Efficiency calculations

2.7. Prototyping on FPGA

The method follows the approach described in [4] and [5]. To generate hardware description language code automatically for the FPGA, the Intel FPGA tool 'DSP Builder Advanced' was used. This is a Simulink library available with Intel Quartus programming tools. Advantages include the abilities to:

- simulate the design in Simulink continuing model-based design
- quickly regenerate code with different high level parameters
- use HDL libraries optimized for Intel FPGAs
- automatically add pipelining registers to achieve the target FPGA clock frequency
- support various floating and fixed point number formats
- gives the option to automatically 'fold' the design to reduce FPGA resources
- automatically generates test benches in Modelsim using the Simulink model results
- opens a Quartus project automatically based on the generated code, so the code can be compiled and FPGA resources calculated.

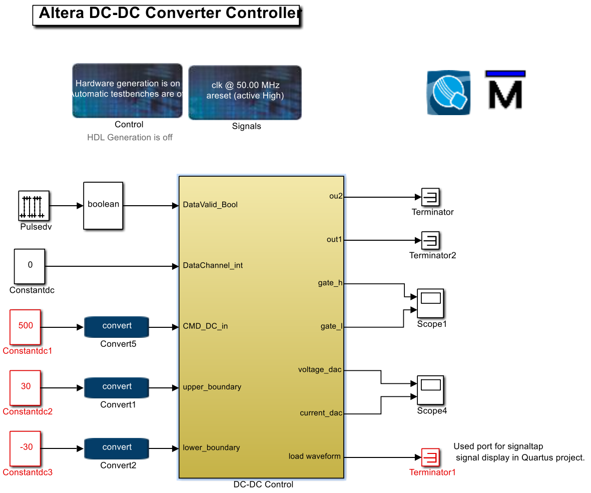

Fig. 7. Top-level of DSP Builder Advanced model

To use DSP Builder Advanced, the original Simulink representation must be translated to DSP Builder blocks. Simulink blocks often have DSP Builder equivalents, but the resulting DSP Builder model specified additional parameters guide the code generation process. Fig. 7 shows the top level of the model for the hysteresis controller including code generation control blocks, and blocks to start Modelsim and Quartus.

DSPB Builder 16.0 was used for this work. Some high-level code generation parameters specified for this were:

- Target FPGA device: MAX 10 with 8 kLE (8000 logic elements)
- Target clock frequency = 50MHz; a 50MHz clock is used in this case for consistency with the existing DC-DC Converter Reference Design [5].
- In the top-level DSP Builder code generation 'Control' block, the Logic/DSP Threshold was set to 200. This guides the code generator to use more logic and fewer DSP blocks than the default setting, enabling the resulting design to fit more easily within the resources of the chosen MAX 10 device.

3. Results and Discussion

3.1. Choice of input filter for hysteresis control

Following the method of section 2.2., results for total current measurement error are set out in Tables 4 and 5, for Sinc1 and Sinc3 filters with different N, at for both 10kHz and 50kHz converter electronics.

Table 4. Sinc1 filter error as a function of N

N for Sinc1 filter	Total error (A), 10kHz electronics	Total error (A), 50kHz electronics
1 (no filter)	1600.03125	1600.15625
2	800.0625	800.3125
4	400.125	400.625
8	200.25	201.25
16	100.5	102.5
32	51	55
64	27	35
128	16.5	32.5
256	14.25	46.25

Table 5. Sinc3 filter error as a function of N

N for Sinc1 filter	Total error (A), 10kHz electronics	Total error (A), 50kHz electronics
2	200.1875	200.9375
4	25.375	26.875
8	3.875	6.875
16	1.8906	7.8906
32	3.0488	15.0488

The optimum Sinc 1 filter has N=256 for 10kHz and N=128 for 50kHz, but the absolute errors in both cases are much higher than with the Sinc3 filters. A Sinc3 with N=16 is best for 10kHz or N=8 for 50kHz. It was chosen to implement N=8 and use for both 10kHz and 50kHz hysteresis controllers for simplicity, as N=8 minimizes the worst-case error between the two.

When implementing the filter in Simulink, it was found that numerical instability would result with floating point numbers due to the repeated integration and differentiation of closely-spaced values. To avoid this, the Simulink filter followed the pattern of a Sinc filter in hardware, working only on unsigned integer values.

3.2. PWM control

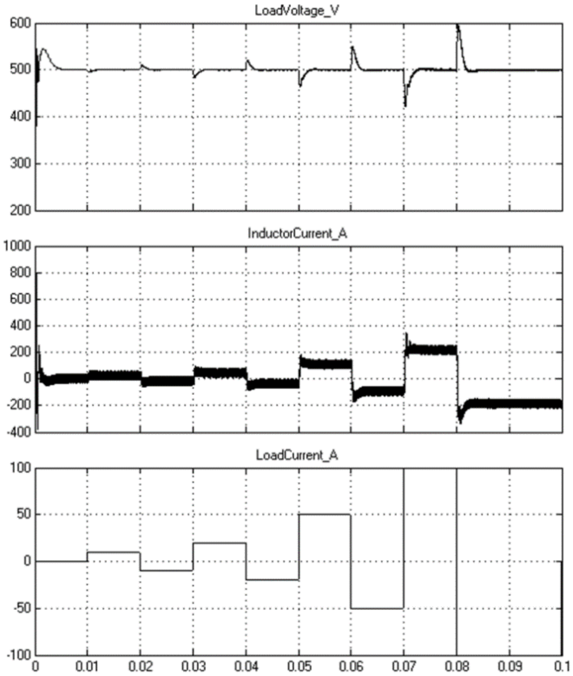

Fig. 8. PWM control, 50kHz

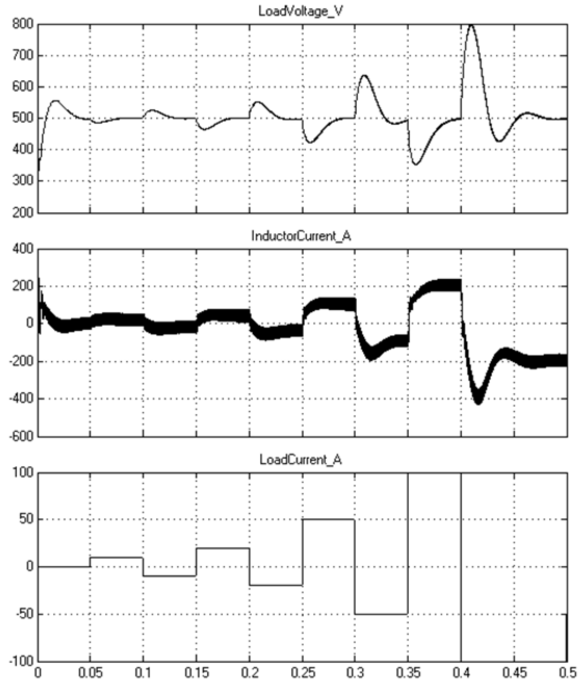

Fig. 9. PWM control, 10kHz

The controller responses after following the tuning procedure in 2.3. are shown in Fig. 8 and Fig. 9. The 10kHz current controller was more difficult to tune, probably due to the combination of longer sensor signal processing and slower responding electronics. Fig. 9. for the 10kHz case uses a 5x longer time base but voltage disturbances are still worse than for the 50kHz case.

3.3. Hysteresis control

Introducing hysteresis control with no change to voltage control can increase voltage disturbances on transients as there is initial delay with a long switching period while the inductor current is driven to a new value. However, the reduced delay in reaching the final current enables voltage PI with stronger P-gain, reducing worst-case deviations. Fig. 10 and Fig 11. Show results after this retuning.

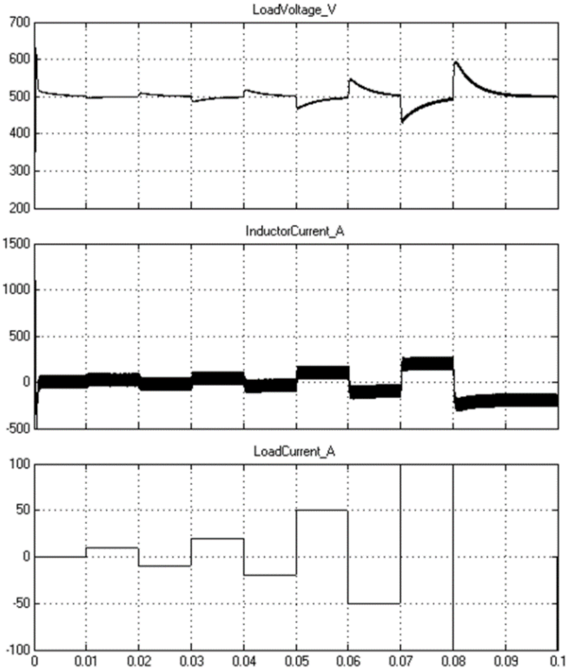

Fig. 10. Hysteresis control with improved VPI, 50kHz

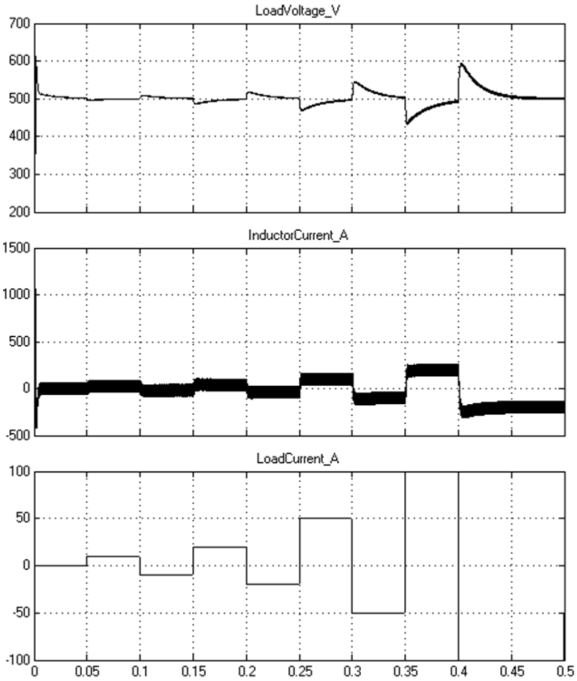

Fig. 11. Hysteresis control with improved VPI, 10kHz

3.4. Summary results, efficiency and disturbance rejection of controllers

Tables 6 and 7 summarize the simulation results with PWM and hysteresis controllers with different error bounds. The hysteresis controllers with improved voltage PI gains are shown with 'VPI+'. Efficiency µ is the average of values at +100 A and -100 A load currents. Iripple is peak-peak current ripple. Verror is the worst-case voltage error, seen on the transient from +100 A to -100 A load current. The gate Switch count is also included.

Using 60 A hysteresis bands improves efficiency significantly in the 50kHz case.

Table 6. Summary results, 50kHz electronics

Ctrl	µ (ND)	Iripple (A)	Verror (V)	Switch count
pwm50	0.92	62.5	98	10001
hys30	0.93	73	120	8391
hys60	0.955	130	120	4666
hys90	0.96	180	120	3194
hys120	0.96	240	120	2480
hys150	0.965	305	120	2038
hys60 VPI+	0.95	135	95	4670
Hys30 VPI+	0.93	73	94	8391

Table 7. Summary results, 10kHz electronics

Ctrl	µ (ND)	Iripple (A)	Verror (V)	Switch count
pwm10	0.97	62.5	300	1000
hys30	0.9725	62	320	9877
hys30 VPI+	0.9725	62	92	9932
hys60 VPI+	0.975	122	94	5083

3.5. Prototyping on FPGA

The PWM controller has been prototyped using DSP Builder in the DC-DC Converter reference design [5]. This model includes simulation of the power electronics in the FPGA.

Results from start-up with a +/-100A load waveform are shown in Fig. 12. This model shows better control than the previous results in this paper because it does not model sensor signal processing delays, and uses the voltage controller saturation feature to limit inductor current demand to +/-300 A, which reduces voltage overshoot on start-up.

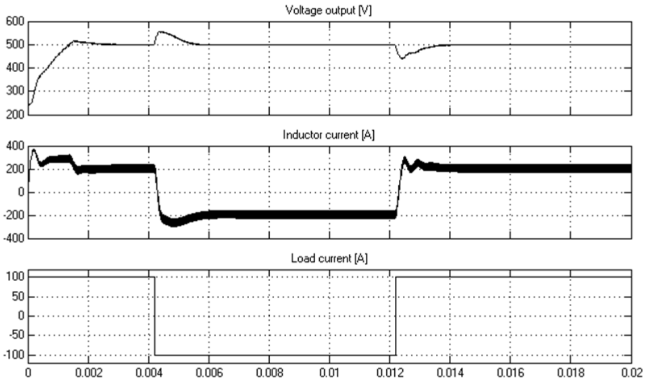

Fig. 12. Results from DSP Builder, PWM control

For a simple comparison, the original current PI+PWM controller was replaced by the hysteresis controller, leaving voltage control gains unchanged. Fig. 13. Shows that while worst-case voltage deviations are similar, inductor current is controlled better on start-up and on transients, leading to better-damped voltage control as well.

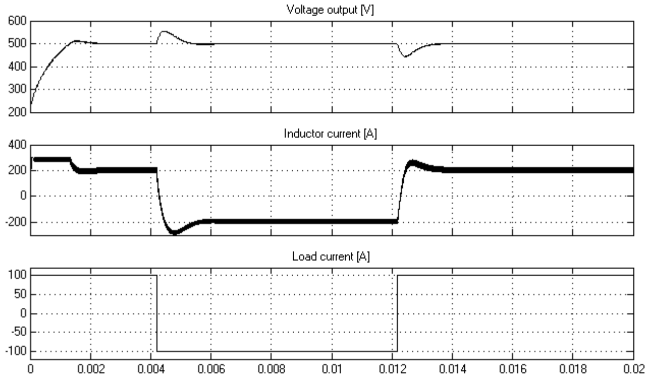

Fig. 13. Results from DSP Builder, Hysteresis control

The FPGA resources for the compiled code can also be compared for the two designs. Table 8 shows that the hysteresis controller design uses fewer resources than the original PWM controller, because the hysteresis controller requires simpler logic.

Table 8. MAX 10 FPGA with 8kLE, resources

Controller	**PWM**	**Hysteresis**
Logic Elements	3,225 (40%)	2,755 (34%)
DSP (9x9)	24 (50%)	18 (38%)
Memory (M9K blocks)	3 (7%)	0 (0%)

4. Conclusions

We have shown in this paper that:

- FPGAs enable non-PWM converter control
- Hysteresis inductor current control improves inductor current and voltage control and reduces FPGA resources compared to the original PWM controller.
- Hysteresis current error bounds determine the switching frequency and efficiency.
- Model-based control design is supported through automatic FPGA code generation.

5. Further work

The DC-DC Converter and Drive-on-Chip Reference Designs provide a platform for research in this area, with real-time simulation on FPGA and control of a real 2-phase bidirectional converter.

References

[1] Microsemi Technical Note TN-203, “Voltage-Mode, Current-Mode (and Hysteretic Control)”, 2012, https://www.microsemi.com/document-portal/doc_view/124786-voltage-mode-current-mode-and-hysteretic-control

[2] Texas Instruments, Rais Miftakhutdinov, “Synchronous buck regulator design using the TI TPS5211 high-frequency hysteretic controller”, 1999, http://www.ti.com/lit/an/slyt186/slyt186.pdf

[3] S. Saggini, et al, “Fully-digital hysteretic voltage-mode control for dc-dc converters based on asynchronous sampling”, APEC 2008. Twenty-Third Annual IEEE.

[4] Altera, White Paper 01210: “FPGA-Based Control for Electric Vehicle and Hybrid Electric Vehicle Power Electronics”, 2015, https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01210-electric-vehicles.pdf

[5] DC-DC Converter Reference Design: https://cloud.altera.com/devstore/platform/16.0.0/dc-dc-converter-control-with-adsp-builder/

[6] Jeppesen, Crosland, Chau: “An FPGA-based platform for integrated power and motion control”, IEEE IECON 2016, Florence, Italy, October 2016.

[7] Intel FPGA (formerly Altera), Application Note an773, “Drive-On-Chip Reference Design”, https://www.altera.com/en_US/pdfs/literature/an/an773.pdf

[8] Analog Devices, datasheet for AD7401A, http://www.analog.com/media/en/technical-documentation/data-sheets/AD7401A.pdf