Fast current controllers using FPGAs

Kjell Ljøkelsøy, Olve Mo
SINTEF ENERGY RESEARCH
Sem Sælands vei 11
Trondheim, Norway
Phone +47 73597200
Kjell.Ljokelsoy@sintef.no
Olve.Mo@sintef.no
http//www.energy.sintef.no

Keywords

Converter control, Active filters

Abstract

This paper describes a FPGA (Field Programmable Grid Array) based PI controller and Pulse Width Modulator for a three-phase inverter. The signal processing circuitry is designed as a distributed pipelined architecture, consisting of several small building blocks. It is designed to be functionally equivalent to an analog PI-controller/modulator circuit. By using a high sampling rate, 1,5 MHz, the effects of sampling and delay in the feedback loop are small compared to the process time scale. The signal processing can be considered to be approximated quite closely as a continuous process, in contrast to common processor based digital implementations that have sample rates of one or two times the switching frequency. This allows for higher feedback loop gain, and shorter time constants. This gives better performance in applications like active filters, where response time is important.

Introduction.

In applications like an active shunt filter, fast current controllers are required. Mains voltage distortion is reduced by injecting currents that cancel the harmonic currents drawn by nonlinear loads. The current controller of the active filter must respond quickly to changes in the load current [1].

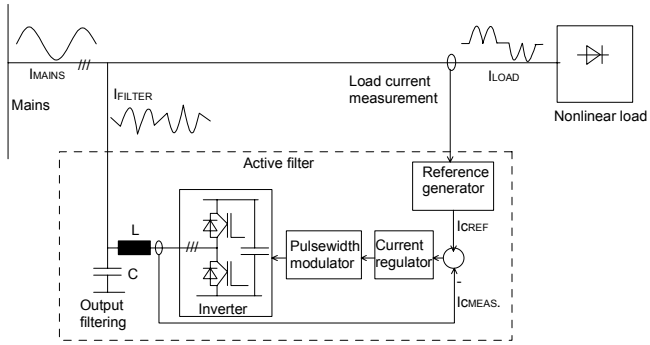

Figure 1: Active shunt filter.

PI current controllers and pulse width modulators are often chosen, because they give constant switching frequency. To obtain the bandwidth needed for good performance, high controller gain is required. It is easier to achieve good performance by using traditional analog controllers instead of microcontroller / DSP based digital controllers, due to the added delay in the feedback loop. The benefits of a digital system are, however, not easy to abandon, so a digital equivalent of the analog controller would be preferable. This could be obtained by a digital system running with high sampling rate and short calculation times compared to the switching frequency. This paper shows how a FPGA based digital equivalent to the analog current controller can be realized.

Signal delay.

The analog controller is used as reference for the design. The FPGA-based digital circuitry is designed to be nearly equivalent to the analog controller, giving similar performance.

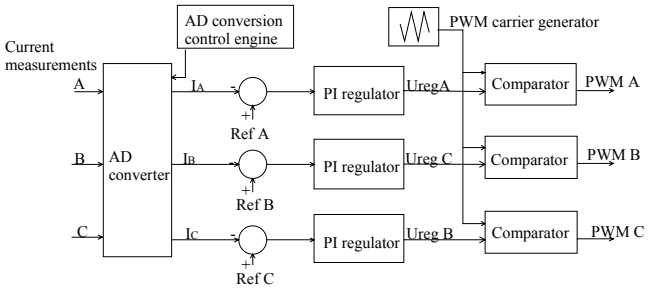

Figure 2: Digital current controller structure.

Some additional circuitry implemented in the FPGA, is omitted in this description for clarity.

Analog controller

An analog current controller works with continuous signals.

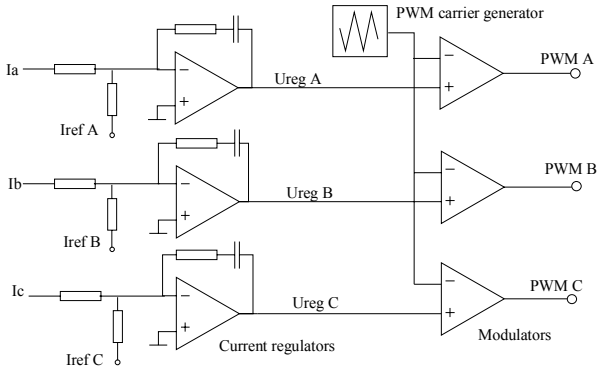

Figure 3: Analog current controller.

Apart from the sampling action of the pulse width modulation, bandwidth limitations and low pass filtering are the main causes of signal delay in the feedback loop.

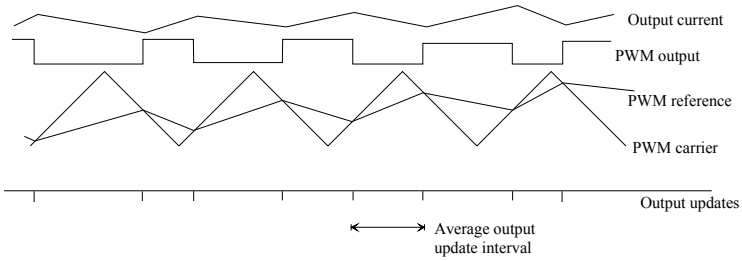

Figure 4: Signal pattern for a single phase analog current controller.

Only minor amount of filtering is applied, so the controller output signal is not smooth. It has a significant amount of ripple. This does not matter, as long as the controller output signal does not rise or fall faster than the triangular modulator reference wave. In the border region where the gain approaches the stability limit, the controller output signal sometimes rises faster than the triangle wave, giving unexpected extra pulses. This may in some cases be tolerated as long as it does not happen too often.

In a three phase converter, interaction between the phases gives somewhat more complex waveforms.

Signal delay in digital controllers

A microcontroller/ DSP implementation of a current controller has by nature a discrete behavior. Current values are sampled, calculations are performed, and the output signals are updated at a later instant. The time needed to do the calculations limits the sampling rate. Sampling rates tend to be in the same order as the switching frequency.

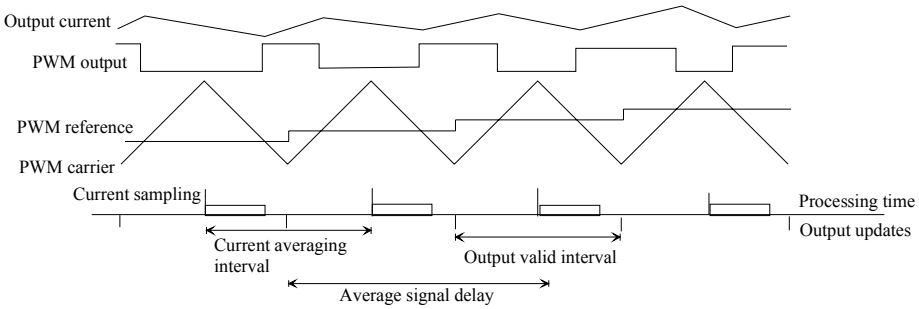

Figure 5: Signal pattern and event timeline for a processor based digital current controller.

Current and reference sampling occurs once or twice for each switching period. Sampling of the continuously flowing currents act as a signal delay. Processing takes some time, typically one half or one switching period. Output signals are updated at every half or whole period. The pulse width modulation also acts as discrete sampling, giving a delay.

The sum of these delays in the feedback loop imposes a severe restriction on the gain that can be allowed before the controller becomes unstable. It also limits the PI controller time constant. As a result, substantially better performance can be obtained by using an analog controller.

By sampling the current continuously at high speed, and use FPGA based parallel signal processing circuitry, instead of the sequential processing done by a processor, most of the added signal delay can be avoided.

Laboratory prototype.

Converter setup

To test the FPGA based current controller, a laboratory setup for an active rectifier, was assembled. Nominal rating: 20 kVA at 230V AC. The converter is connected to a 230V 3 phase grid. The LC output filter consists of series inductors of 0,82 uH and Y coupled capacitors of 100 uF.

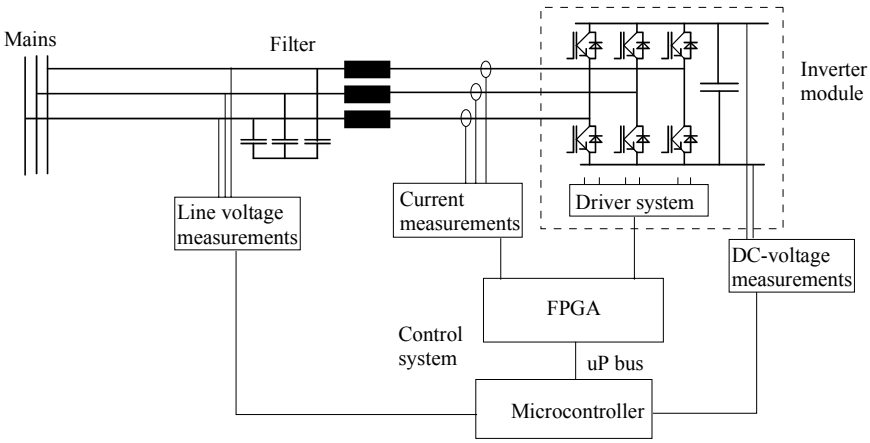

Figure 6: Laboratory setup.

The setup is built up around a 20 kW, general purpose three phase inverter building block, designed for prototype use. Oversized transistor modules, and a driver system with extensive protection functions gives it robustness in overload and control error conditions.

Interlock and turn-on delay circuits in the driver system provides shoot through protection. The inverter is shut down in case of cooling flange overtemperature, DC-link overvoltage and short circuit conditions.

The control electronics consist of a processor board with an Infineon C167CR microcontroller, running at 20 MHz, RAM, FlashEprom, EEPROM, and a 4 channel 12 bit DA-converter. A FPGA board contains a Xilinx Virtex XCV300 FPGA chip, a fast 4 channel 12 bit 6 MSPS AD-converter, a 4 channel 12 bit DA-converter and support circuits, as configuration FlashEprom [2].

The FPGA is connected to the microcontroller through the processor data bus.

Control system functions

The microcontroller software handles most of the control system tasks.

A monitor program running in the background loop, is used to set parameters and references and read variable values in the converter control system. The monitor program is controlled from a PC, connected via serial port.

The control program contains functions for:

- Phase locked loop, generating a stable angle reference for active and reactive current references. Line voltages are used as reference sources.
- Protection and interlock mechanisms, including diagnosis functions.
- DC-link voltage controller. It acts on the active current reference to govern the active power flow in and out of the converter.
- Park transform, converting current references from active-reactive current references to actual phase current references.

Basic structure of the FPGA design.

The FPGA is initially a sea of small, undedicated blocks of logic gates and registers. Before it can do something useful, the required functions must be designed as a digital circuit. This design is downloaded into the FPGA.

The FPGA design is written in VHDL. It is a synchronous design, driven by an 80 MHz global clock.

To make the design job manageable, the system is split into several smaller building blocks, each performing a simple operation. A number of these building blocks are then assembled to make the complete system. Some of these building blocks are designed to do dedicated special functions, while others are made as basic generic functions. These generic blocks are then used as components when larger blocks are being made.

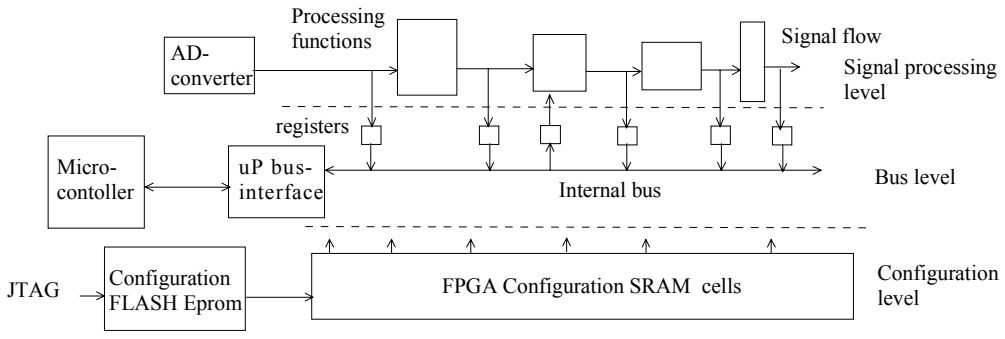

Figure 7 Basic FPGA circuitry structure.

The structure of the FPGA based system consist of three main levels:

- Configuration level.
- Bus level.
- Signal processing level.

The configuration level is actually not a part of the FPGA design, but contains the background system required to make the FPGA chip work. The main parts here are the configuration Flash Eprom, the configuration loading mechanism, and the configuration SRAM array inside the FPGA itself.

A bus structure is made inside the FPGA design. It is connected to the microcontroller bus, and gives access to signals inside the FPGA design. A number of registers act as interface between these signals and the internal bus. Each register has a dedicated address. Two register types exist: Write-registers that are used to set signal values, while Read only-registers that are used for reading signal values.

The blocks at the signal processing level do the actual job of the FPGA design.

Signal processing chain.

The signal processing chain consists of a number of smaller blocks performing simple operations. They read input data, process them and make output data. [3].

The AD-converter acts as a motor for the signal processing chain. A control block generates proper clock and read signals for the AD converter, using a set of counters and state machines. The AD-converter samples the four channels simultaneously every 0,66 μs.

When a new set of values has been read from the AD converter, a NewValue flag is pulsed high trigging the next blocks in the chain to start processing the new data.

Each signal processing block works in a cyclic operation:

- Idle. Waiting for start command.
- Start command flag appears.
- Input data is read. Processing starts.
- Processing. This can last for several clock cycles.
- Finished processing.
- Result is written to an output register. A New Value-flag is pulsed. It lasts one clock period.
- Back to Idle.

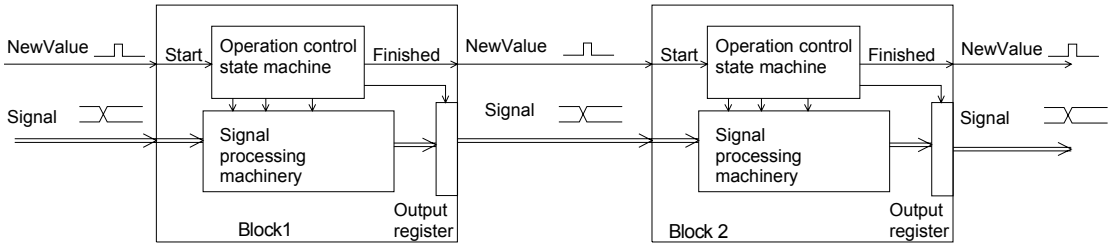

Figure 8 Dataflow between signal processing blocks.

This structure acts as a pipeline, where data flows from one block to the next as soon as they are available. Each signal is linked to a New Value flag.

Some processing blocks have more than one input signal, which not necessarily are updated simultaneously. The NewValue flag for the signal considered being the most time critical one is chosen as start signal for the operation.

The slowest block, the one that needs the largest number of clock pulses to do the processing, determines the maximum possible data rate. The signal delay is determined by the sum of clock pulses for the operations in the chain.

Some of the processing blocks in the design are not able to operate at full pipeline speed. They skip the values that arrive while they are working, and use only those values that appear when they are idling. As a result their operating speed becomes a fraction of the pipeline speed. As long as these blocks are not used as part of the most time critical signal path, this is not a problem.

The DA converter is a slow block. It needs about 7 μs to update all four DA-channels, so it only handles every 10thdata set. The DA converter is attached to some of the most interesting signals in the processing system, making them accessible on an oscilloscope during testing.

PI controller and modulator

The microcontroller sets the values in the current reference registers.

Current error, the difference between reference and measured value, is calculated once a new set of AD values comes. A common mode controller drift compensation signal is also added to the three references. It ensures that the average controller output signals stays at 50% duty cycle. (Not shown on the figures.) The error signal is processed in two parallel gain stages, one for the proportional gain, and one for the integrator time constant.

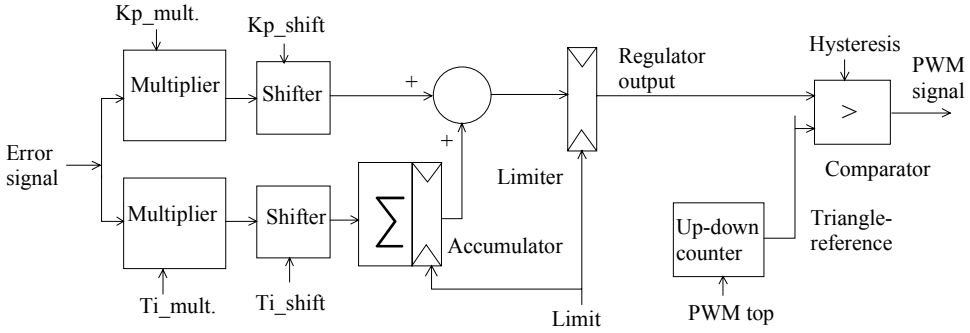

Figure 9 Digital PI controller and pulse width modulator.

The gain stages have two blocks, one multiplier block and one shifter block. The multiplier and the number of shifts determine the resulting gain. This allows gain values different from integer values. As example, a gain of 0,75 is obtained by first multiply the input by 3, and then divide the product by 4 by shifting it down two bits. The microcontroller sets the register values for these parameters.

Data width varies from stage to stage. A summing stage increases the data width by one bit, while the output width of a multiplier block becomes the sum of the width of its input signals. Surplus bits, either in the top or bottom of the signals, are truncated at carefully chosen points in the signal chain.

An accumulator performs the integration. It adds each integrator input signal to the accumulated sum. The large ratio between signal update rate and the time constant tends to give very small vales for the scaling factor for the integrator. To avoid the roundoff errors this would give, the accumulator datawidth is larger than the output signal, and the output is taken from its most significant bits. This acts as an additional shift stage, giving reasonable values for the input scaling. The least significant bits sum up fractions of an output bit over several cycles. A limiter is attached to the accumulator to avoid integrator windup or overflow.

The proportional gain and the integrator output signal is summed, and passed through another limiter stage. The limit levels are set to values just outside the operating range of the modulator.

The modulator stage is fairly simple. An up-down counter is the main part of the carrier wave generator block. It makes a triangular carrier wave. The top and bottom level of this counter determines the switching frequency.

The pulsewidth modulated output signals are made by comparing the controller output signals and the triangle carrier wave. The comparator stage runs continuously, toggling the PWM output signal whenever the controller output voltage crosses the triangle wave carrier wave. To avoid high frequency oscillations, a small amount of hysteresis is added to the comparator.

Results

Signal delay

The AD converter reads four channels, and updates the current values every 0,66µs. It has an internal conversion pipeline giving a signal delay of 5 samples, that is 0,83 µs delay. The signal delay in the processing pipeline is about 1 µs, giving a total delay through the signal processing system of about 2 µs.

This data update rate and signal delay should be sufficiently low to allow for values of controller gain and time constant approaching those of the analog controller. At 5 kHz switching frequency, the signals are updated about 300 times per switching period.

Modulator and controller signals

The converter was set to run at a typical operating point:

- Switching frequency was set to 5 kHz
- Input voltage is 230VAC.
- DC-link voltage is set to 400V DC.
- No load on the DC- link.

Reasonable values for the controller parameters were found by trial and error. Some signals inside the FPGA made available for oscilloscope measurements through the DA converter. (Slow update)

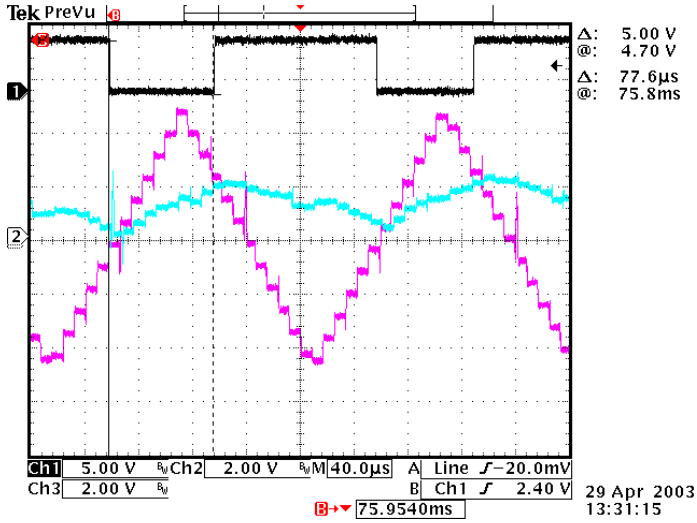

Figure 10: Typical modulator waveforms. PWM output signal (black), Controller output signal (blue), Modulator carrier wave (pink).

Signal waveforms at a typical operating point are shown in Figure 10. Controller proportional gain causes the current ripple to be mirrored on the controller output signal.

Note: The sampling steps in the oscilloscope waveforms are caused by the relatively slow update rate of the DA converter. The signals themselves are smoother, because they are updated about 10 times between each visible step. The DA converter also emits small spikes when it is updated.

Apart form the distortion made by the DA- converter, these waveforms has the same shape as in a similar analog circuitry. The PWM output signal changes as soon as the sawtooth reference crosses the controller output signal.

By increasing the gain substantially, the instability border area is reached. Some irregular acoustic noise from the converter indicates that the gain is too high.

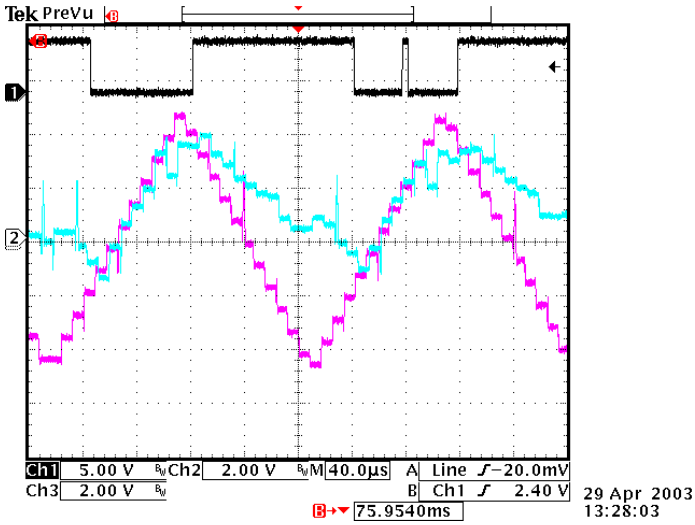

Figure 11: Too large gain. Multiple pulses. PWM output signal (black), Controller output signal (blue), Modulator carrier wave (pink).

The waveforms in figure 11 shows that extra switching pulses occur now and then. The current ripple now drives the controller output signal to change so fast after switching, that it catches up with the triangle wave, and crosses it in the opposite direction, before the top of triangle is reached. The controller loop is unstable, and tends to generate bursts of small pulses in parts of the switching period. Comparator hysteresis suppresses this behavior to some extent.

Gain limitations giving similar waveforms is also found in an analog controller.

Step response.

Step response gives a good picture of the performance of a controller system, both speed and stability. The current controller was given a reference step by changing the reactive current reference of the converter abruptly. The Park transformation then converts this into reference steps for each of the three current controllers.

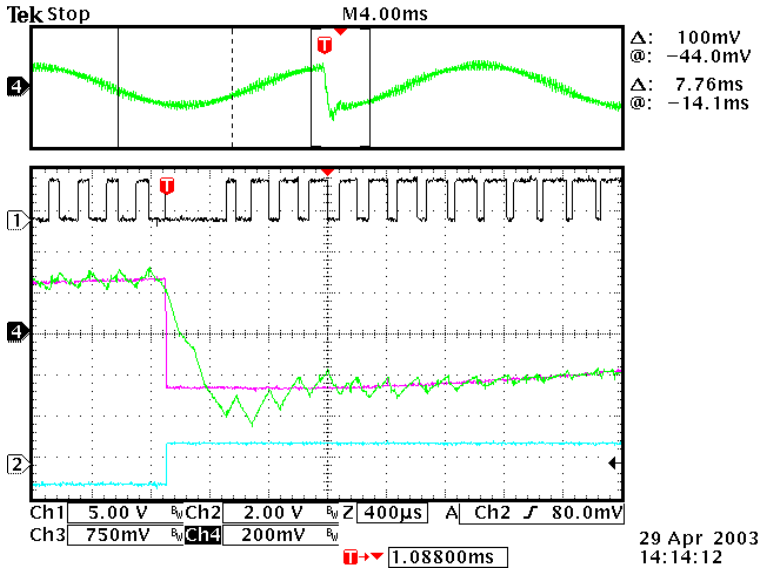

Figure 12 Response to current reference step.
PWM signal (black), Reactive current reference (blue),
Phase current reference (pink), Measured current [20A/div] (green).

When the reference step in figure 12 occurs, the PWM signal has just gone high. This pulse is aborted as the controller output is forced into saturation. It stays there, suppressing any more pulses, until the current has reached the reference again. A small overshoot follows then. The other two controllers also contribute to the resulting current shape.

In this case, the phase current catches up with its reference after about 400 us. That is two switching periods. Current change rate is mainly determined by the driving voltage and the filter inductance.

The step response waveforms shown in figure 12 are similar to signals in an analog controller.

Concluding remarks

The current controller described in this paper works in the same way as an analog PI-controller with pulsewidth modulator. It has similar performance, and the same limitations.

Other controller types can be made instead of the PI controller and pulsewidth modulator used here. Hysteresis controllers can be used to obtain faster response. Nonlinear functions as vector transformations can also be made.

By implementing the most time critical parts of a system in a FPGA design instead of software routines, the workload and speed demand for the processor can be reduced significantly.

A FPGA based design can be used to make pipelined signal processing circuitry that runs at update rates much faster than the fastest events occurring in the signals. Signal processing can then be considered to be nearly continuous, so digital equivalents to analog signal processing circuits can be made.

References

[1] S. Buso, L. Malesani,P. Mattavelli : Comparison of Current Control Techniques for Active Filter Applications. IEEE transactions on industrial electronics vol.45 no.5 (1998)

[2] Helge Kolstad: Control of an Adjustable Speed Hydro utilizing Field Programmable Devices. PhD Thesis, Norwegian University of Science and Technology (2002).

[3] J.Guo, S.h Edwards, D.Borojevic: Elementary Control Objects: Toward a Dataflow architecture for Power Electronics Software. IEEE-PESC Proceedings 2002.