

FPGA-based implementation of a cavity field controller for FLASH and X-FEL

Przemyslaw Fafara¹, Wojciech Jalmuzna¹, Waldemar Koprek¹,
Krzysztof Pozniak¹, Ryszard Romaniuk¹, Jaroslaw Szewinski¹
and Wojciech Cichalewski²

¹ Institute of Electronic Systems, Warsaw University of Technology, ul. Nowowiejska 15/17, Warsaw, Poland

² Department of Microelectronics and Computer Science, Technical University of Lodz, al. Politechniki 11, 90-924 Lodz, Poland

Received 22 December 2006, in final form 10 May 2007

Published 6 July 2007

Online at stacks.iop.org/MST/18/2365

Abstract

The subject of this paper is the design and construction of a new generation of superconducting cavity accelerator measurement and control system. The old system is based on a single digital signal processor (DSP). The new system uses a large programmable array circuit (FPGA) instead, with a multi-gigabit optical link. Both systems now work in parallel in the Free Electron Laser in Hamburg (FLASH). The differences between the systems are shown, based on the measurement results of the working machine. The major advantage of the new system is a bigger area of stability of the machine control loop.

Keywords: free electron laser, superconducting cavity controller, FPGA-based electronic feedback system, control loop stability, fast multi-channel measurement system

1. Introduction

The superconducting cavity simulator and controller (SIMCON) is a project of the reliable, fast and low latency digital controller [1–4], dedicated for the low level RF (LLRF) system [5] in FLASH [6] and x-ray free electron laser (XFEL) [7] machines. The SIMCON family of control systems is completely based on FPGA chips. The task of SIMCON is to stabilize the vector sum of fields in superconducting cavities of one or more cryo-modules in the linear accelerator. The device can also be used as a simulator of the cavity and as a test-bench for other accelerator sub-systems. The flexibility and computation power of this device allow the implementation of fast mathematical algorithms and component models [8–10]. The computation power of the practically applied SIMCON board far exceeds the possibilities of the existing old DSP system, which was used so far for cavity control purposes.

The laser beam quality depends on the stability of the accelerating field. Most of the controller parameters stem from a few basic data on the amplitude and phase stability of a high power, accelerating RF field. These requirements are 10^{-4} for the amplitude and 0.1° for the phase. Such values were never achieved in systems of the previous generation. The

presented results for the new controller promise the mentioned parameters and are a strong justification for further work on FPGA-based solutions.

A block diagram of the used LLRF control system for eight cavities in a single cryomodule is presented in figure 1. A broken rectangle embraces the digital part of the DSP and SIMCON controllers. This diagram shows signal flow in the loop and, thus, is the foundation for the design of the implemented control algorithm. The difference between the DSP and SIMCON devices is that all SIMCON parameters are programmable, including control algorithm exchangeability, while most of the DSP ones are determined for good by the initial controller configuration.

The stability margin of the control loop (figure 1) depends on loop amplification and latency. These parameters, for the existing DSP control solution, are around $5 \mu\text{s}$ for latency and around 1–2 for amplification. The FPGA controller aims at the following values: $1 \mu\text{s}$ for latency and around 100 for amplification. The estimated minimized loop latency (without the controller) is approximately 500 ns. The remaining 500 ns may be attributed to the controller. This value is not reachable by the DSP solution and is within reach for the FPGA-based controller. The control bandwidth is determined by

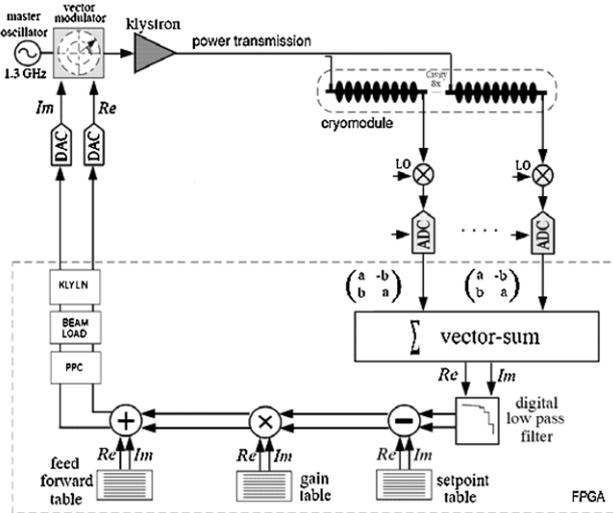


Figure 1. Block diagram of the LLRF control system for eight cavities of a cryomodule. The broken rectangle is the digital controller contained in the FPGA circuit.

Table 1. Parameters of the tested SIMCON cavity controllers.

	Simcon 2.1	Simcon 3.0	Simcon 3.1
FPGA chip	Virtex II 3000	Virtex II 4000	Virtex 2 Pro 50
ADC channels	1	8	10
DAC channels	2	4	4
Digital outputs	2	2	2
Digital inputs	2	2	2
Optical links	–	–	2
PPC	–	–	2
Interface	EPP/VME	VME/ETH/ RS232	VME/ETH/ RS232

the sampling frequency (SF), and intermediate frequency (IF), which are 1 MHz and 250 kHz for both controllers, and which will be 9 or 81 MHz and in the range of 1–10 MHz for a FPGA controller. The criteria of choice for these values are lowering the loop latency and noise mitigation.

Figure 1 shows schematically the signal path in the control loop. The signal from the cavity probe is down-converted in frequency from 1300 MHz to 250 kHz, sampled with 1 MHz and digitized. The digital processing part includes a rotation matrix, vector sum calculation, filtering, the application of a setpoint, gain and feed-forward tables, beam loading compensation and klystron control signal generation. The klystron is finally controlled by a complex analogue signal, via digital-analogue converters and vector modulator.

Several versions of the SIMCON system were designed, tested and applied practically to control a length of the then Tesla test facility (TTF) and now FLASH accelerator. The SIMCON system receives analogue signals directly from the field probes and downconverter (input) and returns analogue signals (output) for the vector modulator (figure 1). Table 1 gathers parameters and resources of the tested SIMCON families of controllers. In particular, SIMCON 3.1. has a FPGA chip and a DSP processor (Tiger-Sharc). The latter is for the off-line (i.e. between the accelerator pulse) calculations. The rest of calculations are done on-line (i.e. during the pulse).

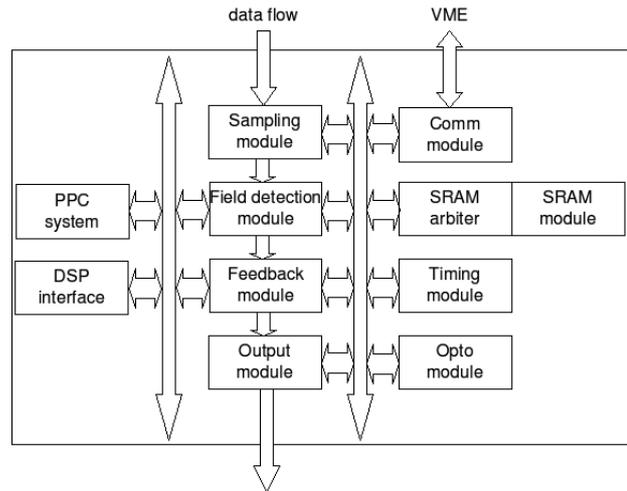


Figure 2. Structure of the SIMCON controller.

Table 2. Synthesis report for the SIMCON controller.

Number of slices:	8183 out of 13 696, 59%
Number of slice flip flops:	7933 out of 27 392, 28%
Number of 4 input LUTs:	13 050 out of 27 392, 47%
Number used as logic:	13 048
Number used as shift registers:	2
Number of bonded IOBs:	338 out of 644, 52%
IOB flip flops:	144
Number of BRAMs:	8 out of 136, 5%
Number of MULT18 × 18s:	45 out of 136, 33%
Number of GCLKs:	10 out of 16, 62%
Number of DCM_ADVs:	1 out of 8, 12%

2. Cavity field controller

The controller is meant as a universal platform for algorithm integration. It executes many algorithms related to field control quality and provides interfaces to the external block such as off-line calculation units. The set of functionalities is not yet closed, so the implemented solution will be constantly upgraded and extended. Therefore, it was necessary to design the internal structure of the controller in a way which makes these tasks as easy as possible. The controller should be able to efficiently use all available resources on the existing hardware platform and on the future platforms which are currently under development. A modular and parametrized design was chosen and the functions executed by each module were defined. The implemented structure is shown in figure 2.

The controller was implemented using VHDL language and synthesized using a Xilinx XST synthesizer. The synthesis report of the controller with a complete feature set for a xc2vp30 speed grade -6 chip is shown in table 2.

Access to the external memory is managed by the SRAM module. Precise timing is required for using one port external memory for three different purposes (DAQ system, control tables readouts and user communications). All the time dependences are managed by a ‘timing module’ and ‘SRAM arbiter’, while the low level memory access is managed by the ‘memory interface’ block.

Currently the controller software is executed on the Simcon 3.1 board and is used to control the first cryomodule of

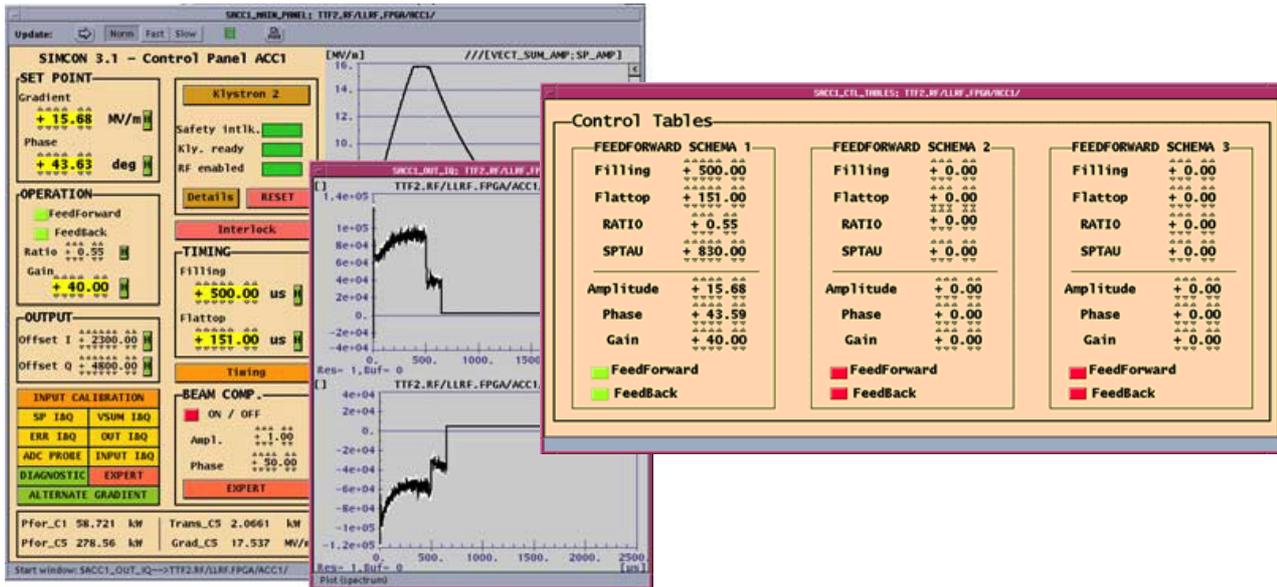


Figure 3. DOOCS control panels.

(This figure is in colour only in the electronic version)

the FLASH accelerator. To make efficient use of the resources available on the controller board, the data acquisition system is based on the external SRAM memory. It is also possible to use optical links to connect the controller to other systems.

The computation pipe which performs all the calculations is divided into two sub-modules: a field detection module and a feedback algorithm module. The first one returns a calculated vector sum of the fields in the cavities in a specified signal representation ($I-Q$, Amplitude- Q , Amplitude-Phase). The second one performs all the computations. The feedback module can be configured to include one or more of the computation modules listed below:

- error signal calculation calculates the difference between measured signal and expected value
- MIMO controller module with variable transmittance which performs actual control function
- Feed-forward module applies predefined control signal
- Beam loading compensation which improves the quality of the control when the beam is on
- Multiple correction modules for other purposes such as klystron linearization

In addition to the control algorithm, integration with the PowerPC processor system was performed. The computation power of this embedded processor can be used for the off-line calculations, which are executed between the pulses.

To provide integration with the existing systems, the interface with the DESY native control environment—distributed object oriented control system (DOOCS) was prepared. This is implemented as two cooperating software servers which communicate with the hardware using the VME bus. The synchronization between the servers and the

controller is provided using the VME bus interrupts. The controller asserts the interrupt at the end of every pulse. The task of the servers is to calculate all necessary operation parameters and to acquire all the data collected by the controller for monitoring and archival purposes. The control panels used to drive the ACC1 module of FLASH are presented in figure 3.

Simcon 3.1 programmability enables identification not only of the whole control loop but also of its individual components. This ability could be used for: beam loading compensation, klystron linearization, exception handling, microphonics and vibration mitigation, software based ionizing radiation hardening of electronics and others. Some of the representative examples are presented below.

3. Beam loading compensation

When the bunched electron beam goes through the accelerating module it extracts energy from the field in the cavities. The biggest effect of energy extraction is when the beam is positioned exactly on-crest of the accelerating field. This process is called ‘beam loading’.

Figure 4 presents how the beam loading effect is compensated. The vector V_{acc} represents an accelerating field in the resonance cavity. The vector V_{bl} is a beam loading, which causes the effective field in the cavity to be represented by the dashed vector. The beam compensation signal must be applied, in order to have the field vector the same as before the beam arrives. The vector V_{bs} represents a beam loading compensation mechanism. Such a compensation signal must be generated in the control system and added to the driving signals. A respective VHDL component was done, which follows a toroid signal and generates a compensation signal which is added to the I and Q components of the control signal.

Effective beam loading compensation requires fast and precise information about the bunches going through the

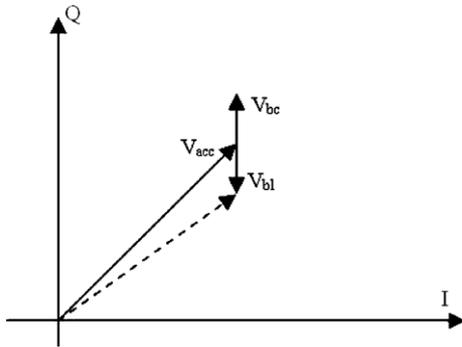


Figure 4. Basics of the compensation of energy extraction by the beam from the high power accelerating field (called ‘beam loading’).

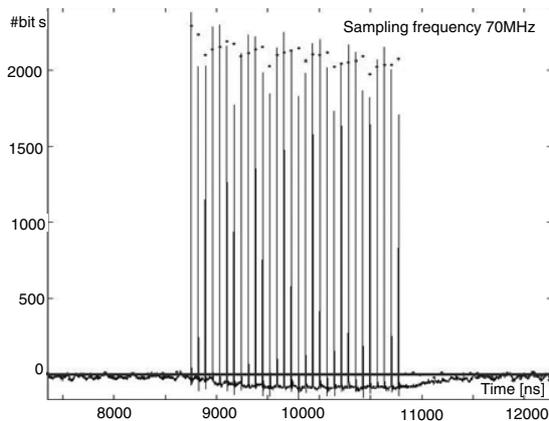


Figure 5. Measurement of 30 bunches in the toroid.

accelerating modules. Such information can be delivered by the toroid which is installed just after the accelerating module. When the bunches go through the module, every microsecond, and then through the toroid, they induce a voltage signal in it. Each bunch induces, in the toroid, a small pulse which lasts about 50 ns and the amplitude of the pulse is proportional to the bunch charge. The output signal from the toroid is connected to one of the A/D converters in the SIMCON board. The signal is sampled with 50 MHz frequency, which means that the number of samples from each pulse is 2 or 3. The SIMCON board has also connected 1 MHz strobe signal which is synchronized with the bunch arrival time.

The strobe signal is used to trigger the integration process in the beam loading compensation component. This integration process improves the stability of the measured values of the charge. Figure 5 presents the input and output of the beam loading compensation component. The beam consists of 30 bunches and the result is an input signal as measured by FPGA. The dots represent the result of the integration process. It is clearly visible that the variation of the integrated signal is smaller than the variation of the sampled signal.

The output signal of the integrator, which is a scalar, is used to create a compensation signal. The control signal is a complex number and the compensation signal must also be a complex number. In such a case the amplitude is proportional to the value of the integrated signal and the phase is found in

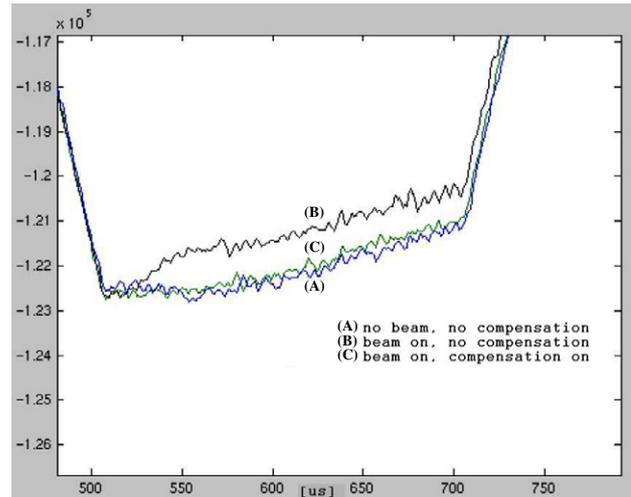


Figure 6. Beam loading compensation.

the calibration process. The calibration process of the phase relies on finding a minimal beam loading effect, which causes phase scan within a range of 100°.

Equation (1) presents how the compensation signal is calculated and added to the driving signal:

$$\begin{aligned} I' &= I + T * k_T * \cos(\alpha) \\ Q' &= Q + T * k_T * \sin(\alpha). \end{aligned} \quad (1)$$

New values of the driving signal (I' and Q' components) are sums of the driving signal from the previous stage I , Q of the controller and a signal generated by the beam loading compensation component, according to the following dependence $T * k_T * \cos(\alpha)$ for I and $T * k_T * \sin(\alpha)$ for Q , where T is the raw measured signal from the toroid generated every microsecond. The signal T is multiplied by a calibration factor k_T which regulates the amplitude of the compensation vector. Both signals, I and Q , are multiplied by a trigonometrical function of angle α , which corresponds to the angle between the field and beam vectors.

Verification of the beam loading compensation component was done with the beam passing through the module ACC1. Figure 6 presents how the beam loading is compensated. The lowest curve (blue) presents the amplitude of the sum vector of the field in eight cavities without the beam and any compensation. When the beam is injected into the module it extracts energy, which makes the amplitude of the field smaller. The beam is injected at a time position of 530 μ s and it consists of 30 bunches with 1 μ s spacing. The upper curve (black) presents the beam loading effect without the compensation. When the compensation is calibrated and it is on the crest of the field amplitude, shown by the middle curve (green), the output signal looks like without a beam, presented by the lowest curve (blue).

4. Adaptive feed-forward

To implement an adaptive feed-forward (AFF) control algorithm, it is required to filter out the noise from the error signal. A simple low-pass filter can be used to achieve stable results. This filter can be either of FIR or IIR type. The

IIR filter is preferred because it can be of lower order, in comparison with the FIR filter. It is important to minimize the order of the filter, because lower order means shorter pipeline, and lower latency in the data processing channels.

In the case of the adaptive feed-forward algorithm, a second-order IIR filter, with a pass band of 5 kHz, was used. Implementation of the IIR filter in FPGA uses a lot of resources, and this resource usage is not dependent on the timing requirement for this algorithm: the real-time online data processing should be executed during the accelerator pulse, while the off-line processes may be performed between the pulses (the pause between the pulses is 200 ms).

The adaptive feed-forward can be calculated between the pulses. It is convenient to place the calculation of the FF signal somewhere else, and use the saved FPGA resources for more urgent and time-critical calculations.

The SIMCON board is equipped with Xilinx VirtexIIPro FPGA chip (XC2VP30), which has two PowerPC405 CPU cores embedded. Such a processor is a relevant place to implement non-time-critical algorithms which process data between the pulses. In this case, the implementation of IIR/FIR filter can be comparatively simple—it can be in the form of two nested ‘for’ loops in C language, parametrized by a constant table of filter coefficients and the number of samples to be filtered.

To exchange data between the controller implemented in FPGA, and PowerPC CPU, it was necessary to implement dual port buffer memories, connected from one side (one port) to the OPB bus (where PowerPC can access its contents), and from the other side (the second port) connected directly to the controller core. Dual port memories have been generated by a Xilinx core generator (tool supplied with ISE software package). Each port of the memory has been set in one mode (read or write). The dual port memory core has been wrapped by an OPB bus slave core. There were two variants of such a combination:

- a core with memory writable by the controller and readable by PowerPC, data transferring for calculation in the CPU;
- a core with memory readable by the controller and writable by PowerPC, transferring the calculation results back to the controller core.

Dual port memories were needed to filter the error signal and obtain the result signal. The error signal and the result signal are represented by complex numbers. Each signal is represented by separate I and Q tables, as shown in figure 7.

The described solution has been implemented and tested with a cavity simulator (also implemented in FPGA). It worked as it was modelled. The achieved results were almost identical with *Matlab* simulations performed using off-line data. There were, however, some differences resulting from different numerical methods used in *Matlab* and C implementation of the filter. The PowerPC 405 is a fixed point processor, and has no hardware support for the floating point operations. All floating point operations are emulated by the compiler, by inserting an additional code. The code performs basic floating point operations (such as addition or multiplication) using selected fixed point numerical methods.

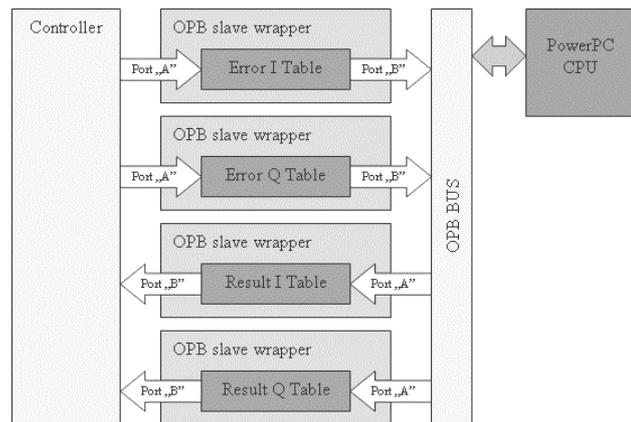


Figure 7. Interface between the controller and PowerPC CPU based on dual port memories.

Tests of the adaptive feed-forward algorithm have been performed on the cavity simulator (implemented in FPGA); the results are shown in figures 8 and 9. The control signal which normally should drive a klystron is shown on the left side, these plots have been recorded by an oscilloscope from the controller output (DAC). On the right side the vector sum is shown—a vector sum has been read out from the controller memory (directly from the simulator part). A standard feed-forward (without any adaptation) is shown in figure 8—vector sum (on the right) is far from the requested shape. In figure 9, the algorithm has been adapting for about a few hundred steps (defined by accelerator pulses). It is clearly visible that the result (vector sum) is much closer to the requested shape than in the previous case. The oscillations visible on the control signal after the adaptation are related to the parameters of the low pass IIR filter used to filter the error signal before applying the correction to the control (standard feed-forward) signal.

The most important result of the test was the time of execution, which in the case of two tables (I and Q) of 2048 samples (32 bit each), filtered by a second-order IIR low pass filter by the CPU running at frequency 300 MHz, was about 20 ms. It took about 10% of the time available between the accelerator pulses. This result qualifies this implantation of the adaptive feed-forward to be tested on the machine while controlling the accelerator.

Time measurement was done by reading the internal counter in the CPU (which is incremented on every rising edge of the clock). Reading the value of this counter was done by executing assembler instruction ‘mftbl’ (move from time base low) as an inline assembly in the C code.

To achieve the precise value, first an overhead of inserting an inline assembly in the C code was measured, by inserting two identical assembly inlines one after another (without any code between). The difference between the achieved values was 10 clock cycles. With this knowledge, every measured time of execution code segment between inline assemblies was decremented by 10, and it was a precise number of the clock cycles that the processor spent on executing the measured code. The number of clock cycles spent on the code execution and the knowledge of CPU clock frequency was enough to calculate the time of execution.

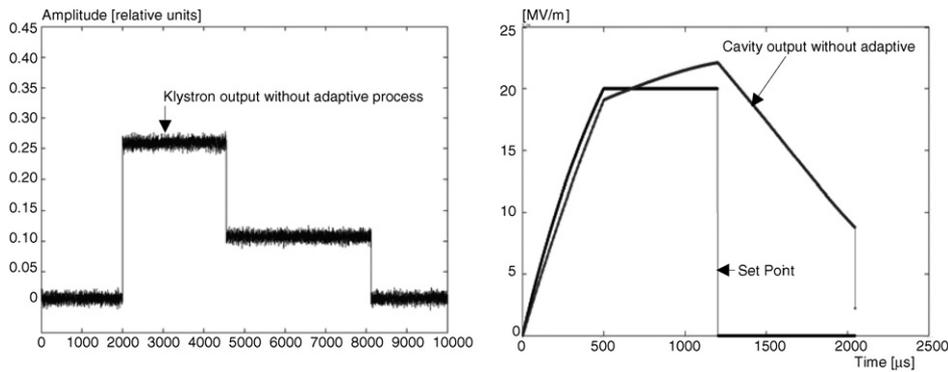


Figure 8. Control signal and vector sum without adaptive feed-forward.

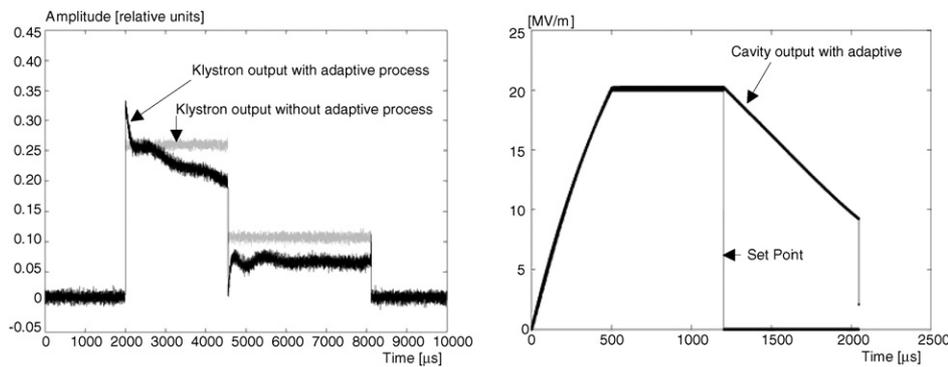


Figure 9. Control signal and vector sum with adaptive feed-forward.

5. Compensation of klystron nonlinearity

The nonlinearities of klystron and high power preamplifiers (when close to saturation) are the major phenomena that have negative influence on the whole control loop performance (figure 10). Playing the role of loop actuators, they are the main source of the electromagnetic field that is delivered to the cavities. Due to specific power requirements, most of their operational time they are working in the region where small signal gain significantly differs from the large signal one. Phase deviation that may even be at the level of several degrees is also present in that case. All these drawbacks cause power dependent loop gain phenomena that imply less effective LLRF feedback loop control.

To overcome the described difficulties, dedicated linearization algorithms have been implemented in the controller (figure 12). Basing on the amplifier characterization results, the digital predistortion approach based solution has been proposed and realized in the controller. The main responsibility of the algorithm is the distortion of the controller signal in such a way as to provide linear response of the amplifier chain.

The first step for nonlinear amplifier behaviour handling is characterization. Using the well-known constellation diagram method the static characteristic of the amplifier response for the applied test signal was measured (figure 10). Then amplitude and phase characteristics were calculated from the in-phase and quadrature representation of the response signal. Using this information a dedicated MATLAB set of functions was used to recalculate the correction coefficients. Afterwards calculated values were written to the LLRF feedback controller

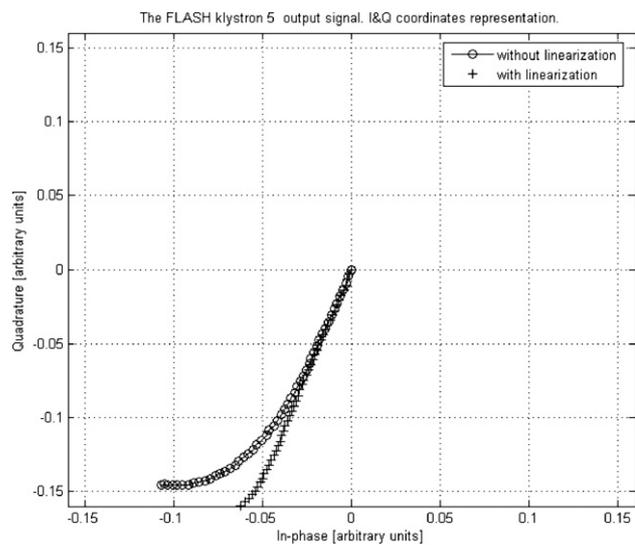


Figure 10. FLASH experiment klystron no 5 nonlinearities.

implemented in FPGA. The set of look-up tables (prepared before) was being filled with achieved values. Once the Simcon generated the control signal the linearization tool provided complex multiplication of the *I* and *Q* representations of the controller signal and correction signal with coefficients from the aforementioned tables. All mentioned operations provided real time amplitude nonlinearity cancellation and phase deviation compensation.

From the performed test, it can be concluded that the power amplifier linearization provides improvement in

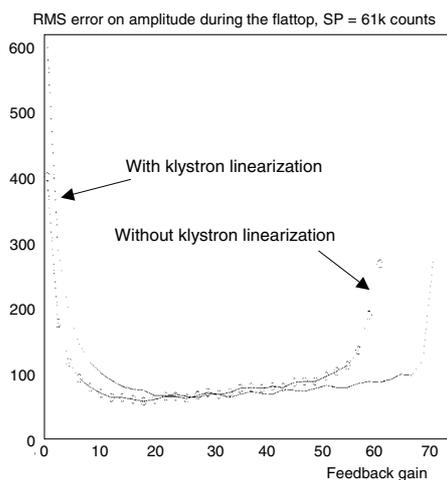


Figure 11. RMS of the field amplitude on the flattop with different gains.

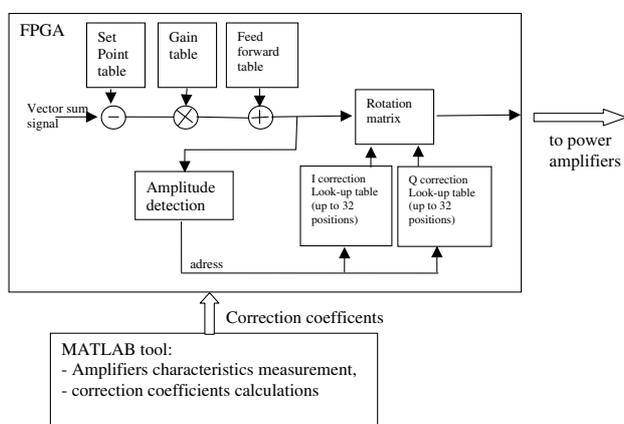


Figure 12. Overall structure of the linearization tool implementation in FPGA.

the loop performance level. As can be seen from the measurements (figure 11), the mean RMS error of the vector sum signal from the desired set point value remains on the lower level for a wider loop feedback range of gains. This gives an opportunity for the work with the high loop gain and without compromising between the output power level and feedback performance.

6. Conclusions

The SIMCON system, ver. 2.1, consisting of hardware and software, was practically introduced to control

superconductive TESLA cavities two years ago. Currently, the SIMCON version 3.1 is in operation in FLASH. The flexibility of the FPGA-based system enables testing of different control algorithms. The basic features of SIMCON (compared with the previous solution) are the following: greater processing power for measurement data calculation on-line; bigger resources (logic and memory block) in terms of performed metrological, monitoring, diagnostic and data acquisition functions; faster speed and, thus, lower latency; more parallel functional operation for a multichannel measurement system, such as on-line model linearization. In numbers the obtained measurement and control system parameters were as follows: the maximum allowed gain was over 100 and the SIMCON latency was below 500 ns. The beam loading compensation allowed us to test long train bunches and the klystron non-linearity compensation allowed us to achieve higher gain in the control loop. SIMCON is equipped in a multi-channel, multi-gigabit optical transceiver enabling replacement of the RF cables, in the previous system solution, with optical fibre links now. An optical network connects Simcon 3.1 boards with Simcon 4.0, a multi gigabit data concentrator [11]. A control system based on Simcon 4.0 board connected with 8 Simcon 3.1 boards provides service of up to 80 fast measurement channels. The system could control up to three cryomodules including the measurements of forward and reflected power from the individual cavities.

References

- [1] Czarski T *et al* 2006 Superconducting cavity driving with FPGA controller *Nucl. Instrum. Methods Phys. Res. A* **568** 854–62
- [2] Czarski T *et al* 2006 TESLA cavity modeling and digital implementation in FPGA technology for control system development *Nucl. Instrum. Methods Phys. Res. A* **556** 565–76
- [3] Czarski T *et al* 2005 Cavity parameters identification for TESLA control system development *Nucl. Instrum. Methods Phys. Res. A* **548** 283–97
- [4] Giergusiewicz W *et al* 2005 Low latency control board for LLRF system: SIMCON 3.1 *Proc. SPIE* **5948** 710–15
- [5] LLRF DESY web page <http://tesla.desy.de/LLRF>
- [6] FLASH web page <http://flash.desy.de>
- [7] XFEL web page <http://xfel.desy.de>
- [8] Pozniak K *et al* 2004 SIMCON 1.0 Manual, DESY Tesla-FEL Report 2004-04
- [9] Pozniak K *et al* 2005 SIMCON 2.1 Manual DESY Tesla Report 2005-02
- [10] Pozniak K *et al* 2005 SIMCON 3.0 Manual DESY Tesla Report 2005-20
- [11] Perkuszewski K *et al* 2006 FPGA-based multichannel optical concentrator SIMCON 4.0 for TESLA cavities LLRF control system *Proc. SPIE* **6347** 634708