Sub-LSB DAC Resolution Enhancement Applied to LLRF Control

Mariusz Grecki
MSK, DESY, Hamburg, Germany
mariusz.grecki@desy.de

Abstract—**The digital control systems use the ADCs/DACs as a front-end for analogue signals processing. The paper proposes the solution to enhance the DAC resolution by PWM. The modulation scheme is optimized to the parameters of of LLRF control in FLASH accelerator. For that purpose the genetic algorithm was implemented and applied. The distributed computing was used to speed-up the computations.**

Index Terms—**LLRF Control, DAC, Sub-LSB Resolution, Modulation, Genetic Algorithms, Parallel Computing**

I. Introduction

The main goal of LLRF control system (Fig. 1) in accelerator is to control field in the cavities with given accuracy [1]. Nowadays the LLRF system is usually implemented as a digital control system [2]. Digital systems have many advantages comparing to the analogue control systems but their application is always limited by the parameters of analogue front end (ADCs and DACs). One of the most important parameter of ADCs and DACs is resolution. It is expressed in number of bits that can be converted from and to analogue voltage.

The resolution can be improved (extended) by oversampling. In case of ADC it can be realized by averaging. In case of DAC the pulse-width modulation can be used. The modulation scheme must be simple enough to implement it in hardware. One of the simplest method to generate the PWM modulated signal is intersective method, which requires only a sawtooth modulation waveform and comparator (Fig. 2).

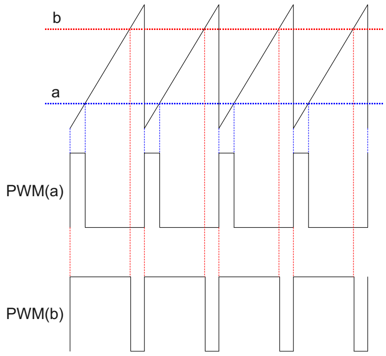

Fig. 2: PWM modulation with sawtooth signal

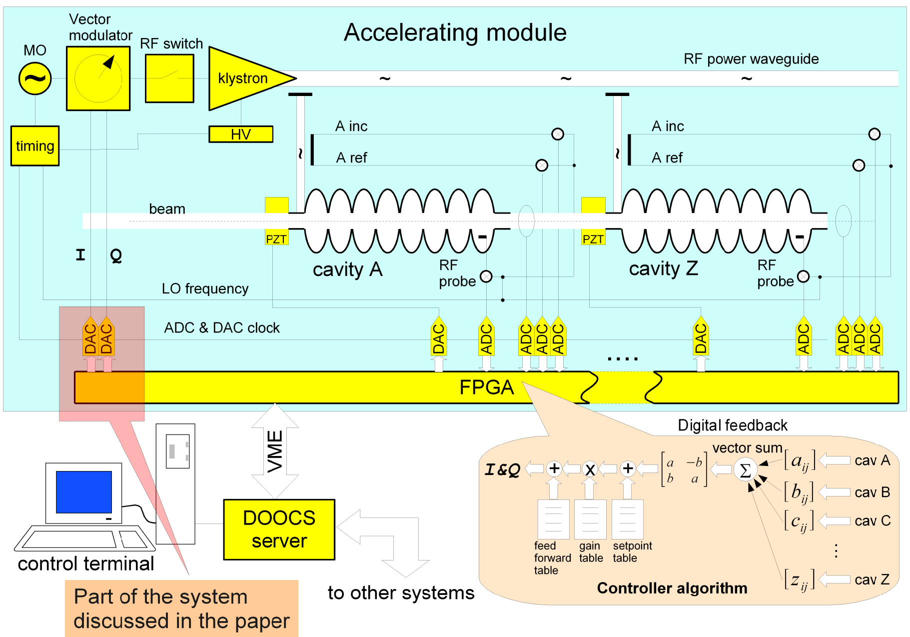

Fig. 1: Block diagram of LLRF system

As an example lets take into account the digital output of LLRF system used at FLASH accelerator (Fig. 1) [3]. It consist of two 14 bits DACs (AD9744 [4]) driving the two inputs (I and Q – **I**n Phase and in **Q**uadrature) of Vector Modulator. The internal calculation are performed using 18 bits arithmetic thus it make sense to try to extend the DAC dynamic range by additional 4 bits. The update rate of output signal is 1MHz thus a new sample appears every 1µs. The AD9744 can be clocked at maximum 165MHz [4]. Since the 81MHz frequency is easily accessible in the system it can be used as a modulation carrier. Taking all of that into account one could imagine the digital waveform consisting of 81 samples 4 bits each stored in a memory that will be combined with the calculated 18 bits wide output signal and will determine the digital output send to DAC (Fig. 3). The simplest form of the PWM modulation waveform would be a sawtooth presented in Fig. 2. The shape of the signal is simple for hardware implementation giving the output signal presented in Fig. 4 for the whole range of input numbers (0-15) but the frequency characteristics of the output is not optimal. In the Fig. 5 the high peak for low frequencies is clearly visible (around 5[th] harmonics). Using different shape of the PWM modulation waveform would it be possible to move the signal power density to the higher harmonics in order to easily filter them. Therefore the shape of PWM should be optimized with goal function penalizing the low frequency components of the output PWM signal for the whole range of input signals. This can be done applying optimization algorithm to the shape of PWM modulation waveform.

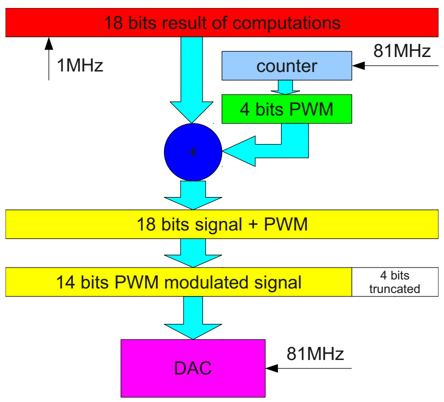

Fig. 3: PWM modulation by intersective method - implementation in LLRF controller

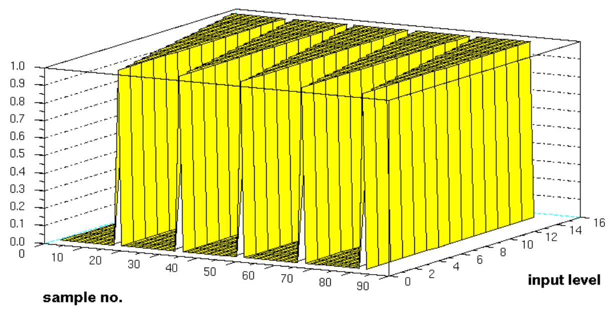

Fig. 4: Output PWM signal depending on the input level for the sawtooth signal

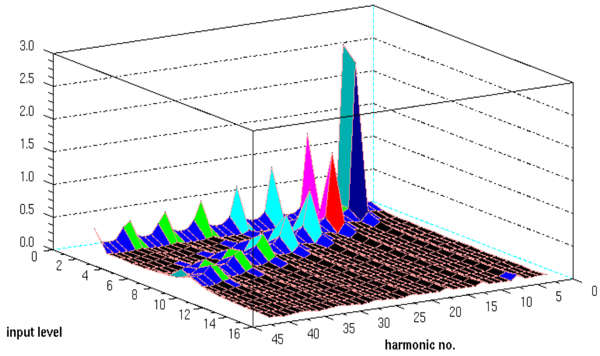

Fig. 5: Frequency spectrum of output signal for the sawtooth signal

II. Optimization of the PWM Modulation

The optimization of the PWM modulation scheme for the LLRF control system is straightforward to formulate. One should choose a set of 81 integer (p_j – j=1..81) numbers from the range of 0-15 in a way the PWM output signals fulfills following formal conditions:

$$\begin{aligned} &\overline{PWM(x)} \quad \propto \quad x \\ &\forall\, x : max(w_i h_{i,x}) \quad is\, minimum \end{aligned} \tag{1}$$

where:

$\forall$ - for all quantifier

$\overline{PVM(x)}$ – PWM output signal for x input (average)

$$PWM(x) = \begin{cases} 0 & if\ (x+p_j) \le 15 \\ 1 & if\ (x+p_j) > 15 \end{cases}$$

x – input level (least significant 4 bits of calculated output)

$h_{i,x}$ – ith harmonic of PWM signal for input level x

w_i – weight for i^{th} harmonic

However the classical optimization algorithms cannot be used efficiently since a big number of optimization variables (81) and their discrete character (integer numbers in the range 0-15). Therefore the genetic algorithm was applied to find the satisfactory solution. The fitness (goal) function fully corresponds to the conditions (1). It is defined as a sum of a measure of non-proportional character of average of PWM signal to the input level x plus the maximum of amplitude of weighted harmonics for all input levels x (2).

$$\begin{aligned} E = &we_1 \sum_x |\overline{PWM(x)} - x| + \ldots \\ &+ we_2 max_x (max|w_i \cdot h_{i,x}|) \end{aligned} \tag{2}$$

where:

we_1, we_2 - weight factors

max_x - maximum for all x levels

$$w_i = \begin{cases} 1 & i=2..8 \\ 1-0.999\cdot(i-8)/10 & i=9..18 \\ 10^{-3} & i=19..40 \end{cases} \qquad \text{(Fig. 6)}$$

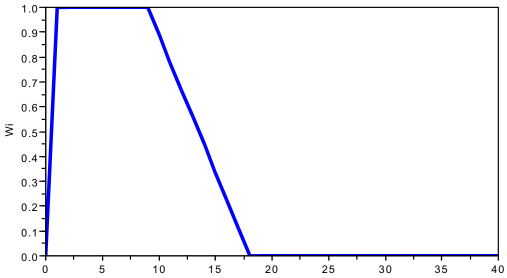

Fig. 6: The weights applied for goal function depending on harmonics number of the PWM modulated signal

III. Genetic Algorithm

The genetic algorithm is dealing with a group (population) of units characterized by genotype (chromosomes) that codes the properties of a single proposed solution for the problem. In the particular case of the optimization of the PWM modulation scheme the chromosome was defined as a sequence of 81 numbers of in range 0-15. The search space consist then of over $3 \cdot 10^{97}$ various elements. This is enormous variety of possible solutions.

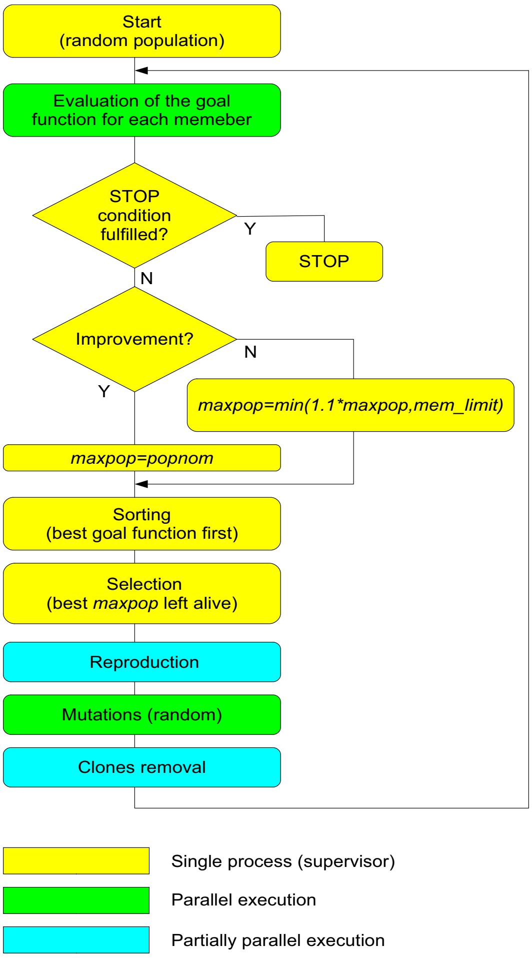

Fig. 7: Block diagram of genetic algorithm

The applied genetic algorithm has a typical structure (Fig. 7). It consist of looped sequence of elementary steps:

- evaluation of goal function (for each population member)
- selection of population part for reproduction
- reproduction
- mutations

The genetic operators (reproduction and mutations) were adjusted to the particular problem. Reproduction was made by creation of a new units from the genotypes of two existing units chosen to reproduce (Fig. 8). The selection for reproduction was made on the base of goal function. The population was sorted (least goal function value first) and then *maxpop* units were left alive (the *maxpop* was changing during computation accordingly to the obtained progress – if in last few steps there was no progress the population size was increased by 10% until the limit fixed by available memory was reached). Then from the surviving population the pairs of parents were selected (randomly) and children were reproduced. If the parents had genotypes enough different additional children were generated.

After the reproduction phase some units were being chosen (randomly) for mutations. If the unit was selected the part of its genome (with random position and size) was exchanged by random values (Fig. 9). Finally the whole population was searched for duplicated units (clones) and they were eliminated (only single copies of units were stored in population).

Then the entire process was repeated starting from evaluation of population members.

The algorithm was implemented in the Scilab script language [5]. The whole software package was written in a way that it is very easy to adjust it to various problems that can be formulated and solved using Genetic Algorithms. The implementation of the core of the algorithm is independent of the problem – only the goal function must be provided for other problems.

Since the genetic algorithms are rather processing-power hungry the parallel execution of the genetic algorithm were investigated. Unfortunately it is not possible to execute all the steps in parallel but some of them (in particular the one most time consuming – goal function evaluation) it is quite easy to distribute over several computer. For the framework of parallel computation the PVM system were chosen [6]. It allows to run the distributed programs on a computing cluster and provides the communication between the modules over the network. What was very important it has direct support by the Scilab scientific computational software package which was used for implementation of the algorithm.

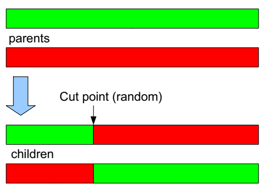

Fig. 8: Reproduction (cross-over) operator

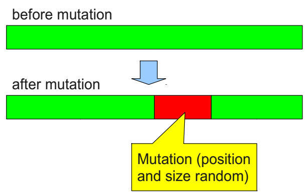

Fig. 9: Mutation operator

IV. Results

First the simplest PWM modulation scheme with sawtooth signal was evaluated for comparison (Fig. 10).

The main disadvantage of this scheme is a contents of low frequency harmonics in the output signal. The other problem is the slope of the transfer function which is not perfectly right since 16 is not a divider of 81.

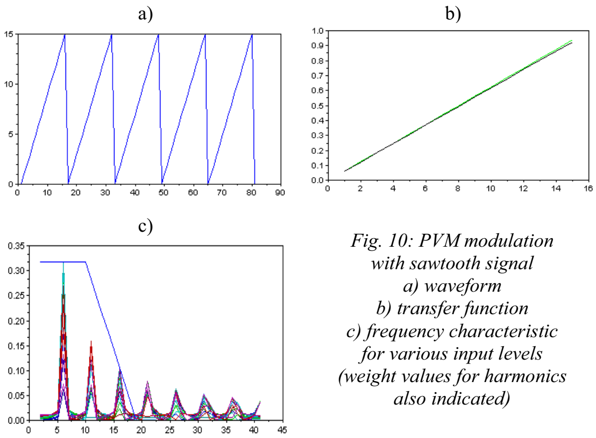

Fig. 10: PVM modulation with sawtooth signal
a) waveform
b) transfer function
c) frequency characteristic for various input levels (weight values for harmonics also indicated)

Then the optimization of the PWM modulation scheme has been performed using described distributed computational system with genetic algorithm. Starting from the random population the genetic algorithm has found the satisfying solution. Unfortunately it is not possible to check whether this is optimal one, however after reaching this point no further progress was observed even with large populations. The results for the best found solution is presented in Fig. 11.

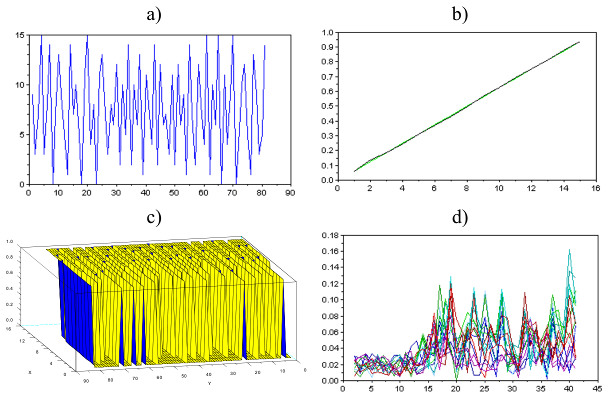

Fig. 11: Best solution of PVM modulation scheme found by genetic algorithm
a) waveform
b) transfer function
c) PWM output signal
d) *frequency characteristic (for input levels: 0-15)*

Comparing the results of optimized modulation scheme with the results of simplest sawtooth waveform it is clearly visible that the regular high peaks for harmonics *6n* were removed and the signal energy was distributed much more uniformly. The most optimized region (harmonics 1-8) are highly attenuated for all input levels. Small deviation from ideal transfer function can be observed but the slope is correct.

The history of calculation is presented in Fig. 12. After initial fast progress in goal function improvements (first 300 iterations of algorithm) the progress is very slow and finally has stopped completely, even with very big populations. Further execution of the algorithm did not improve the goal function even with manually modified probabilities of mutations and their size (this allows to tune the genetic algorithm).

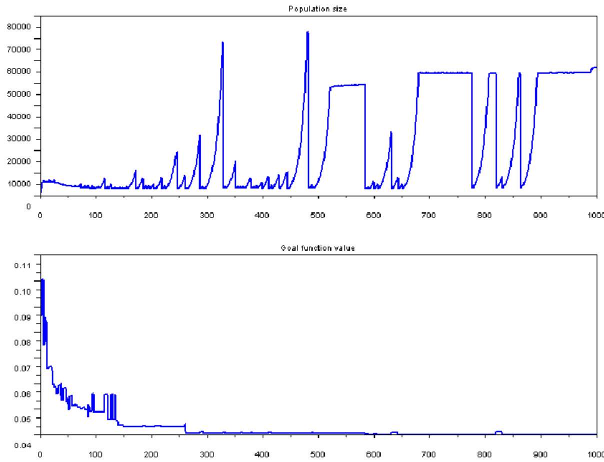

Fig. 12: History of calculations
(upper - population size, bottom – goal function)

V. Conclusion

In the paper the new method for DAC resolution enhancement using PWM modulation of the output signal from LLRF control system. The modulation scheme was optimized using Genetic Algorithms. Due to optimization the frequency spectrum of the PWM output signal contains only minimum of low frequency harmonics in particular comparing to simple PWM modulation using sawtooth signal. To speed-up the computation the optimization algorithm was implemented in the cluster of computers using computing using PVM as communication platform.

References

[1] Ayvazyan V., Petrosyan G., Rehlich K., Simrock S.N., Vetrov P." *RF Control System for the DESY VUV-FEL Linac,* Proc. Particle Accelerator Conference, Knoxville 2005, pp. 2899-2901

[2] Simrock S.N.: *State of the Art in RF Control,* Proc. of the 2004 LINAC Conference, Luebeck, Germany. pp.523-525

[3] Giergusiewicz W. et all, *Low latency control board for LLRF system: SIMCON 3.1,* Proc. of SPIE Volume 5948, Photonics Applications in Industry and Research IV, pp.710-715

[4] AD9744 data sheet:http://www.analog.com/static/ imported-files/data_sheets/AD9744.pdf

[5] Scilab Webpage http://www.scilab.org/

[6] PVM Webpage http://www.csm.ornl.gov/pvm/