Nuclear Instruments and Methods in Physics Research A 738 (2014) 74–81

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Synthesis of optimal digital shapers with arbitrary noise using simulated annealing

Alberto Regadío [a,b,*], Sebastián Sánchez-Prieto [a], Jesús Tabero [b]

[a] Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain
[b] Electronic Technology Area, Instituto Nacional de Técnica Aeroespacial, 28850 Torrejón de Ardoz, Spain

ARTICLE INFO

Article history:
Received 22 July 2013
Received in revised form
19 November 2013
Accepted 26 November 2013
Available online 10 December 2013

Keywords:
Spectroscopy
Noise
Shaping
Adaptive
Digital signal processing
Simulated annealing

ABSTRACT

This paper presents the structure, design and implementation of a new way of determining the optimal shaping in time-domain for spectrometers by means of simulated annealing. The proposed algorithm is able to adjust automatically and in real-time the coefficients for shaping an input signal. A practical prototype was designed, implemented and tested on a PowerPC 405 embedded in a Field Programmable Gate Array (FPGA). Lastly, its performance and capabilities were measured using simulations and a neutron monitor.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In spectroscopy, the information about incident particles can be extracted from the peak amplitude of the input pulses coming from particle detectors. This method is called Pulse Height Analysis (PHA) and provides a value proportional to the incident particle energy. Thus, identical particles with the same energy must generate identical peak values. The resolution of these systems is affected by noise. In spectroscopy, the noise is classified into three types: white series, white parallel and $1/f$ noise [1]. The $1/f$-parallel noise [2] is not considered in this work as the contribution is negligible in all electronic devices used in modern front-end electronics. On one hand, each type of noise has a spectral density that depends on the type of detector and the features of the spectroscopy system. In fact, some of these three noise types could be negligible depending on the type of detector and spectroscopy electronics. On the other hand, spectroscopy systems have filters at the output of particle detectors or pre-amplifiers called shapers. A basic feature of shapers is their capability to filter out noise. This capability is generally measured using noise indices [3]. Thus, the noise index for each type of noise must be considered. As such, spectroscopy systems are affected by both noise spectral density and the noise index of the selected shaper. The Signal/Noise Ratio (SNR) is generally measured using the Full Width at Half Maximum (FWHM) or the Equivalent Noise Charge (ENC).

For each time-invariant spectroscopy system, at least one optimal shaper exists. The optimal shaper depends on the spectral density of each noise type. There exist methods to calculate the optimal shaper, one of the most popular is described in Ref. [1]. However, the complexity of this method sometimes implies that optimal shapers were selected using other procedures (e.g. Ref. [4]). In this paper, a readily implementable optimal algorithm based on simulated annealing to find out automatically a shaper that filters the noise efficiently is developed.

This paper is structured as follows. Section 2 presents the fundamentals of the simulated annealing algorithm and the cost functions used in this work. Section 3 provides details of the FPGA platform, covering both, the hardware and the software. Section 4 presents the theoretical results of the simulated annealing. Section 5 presents the experimental results. Finally, Section 6 covers the conclusions and the future work.

2. Simulated annealing

Simulated annealing [5] is a technique for combinational optimization problems, such as minimizing functions of many variables. This technique was introduced by Kirkpatrick et al. [6]

* Corresponding author at: Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain. Tel.: +34 669013583.
E-mail addresses: aregadio@srg.aut.uah.es, aregadio@isdefe.es, alberto.regadio@edu.uah.es (A. Regadío), sebastian.sanchez@uah.es (S. Sánchez-Prieto), taberogj@inta.es (J. Tabero).

0168-9002/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nima.2013.11.099

and it was motivated by an analogy to the statistical mechanics of annealing in solids. To understand why such a physics problem is of interest, we may consider how to coerce a solid into a low energy state. A low energy state usually means a highly ordered state, such as crystal lattice. To reach this state, the material is annealed: heated to a temperature that permits many atomic rearrangements and then cooled slowly until the material freezes into a good crystal. Thus, simulated annealing offers an appealing physical analogy for the solution of optimization problems, and more importantly, the potential to reshape mathematical insights from the domain of physics into insights for real optimization problems.

Interest in such algorithms is intense because few important combinational optimization problems can be solved exactly in a reasonable time. For our purposes, a combinational optimization problem is one in which we seek to find some configuration of parameters that minimizes a given function which is usually referred to as the *cost function*. This function is a measure of goodness of a particular configuration of parameters. The election of an appropriate cost function is crucial for achieving good results using this algorithm.

The simulated annealing is an iterative algorithm. In each iteration, it generates some random perturbation, such as moving a particle to a new location. The random perturbations are proportional to a simulated temperature T. Thus, at higher temperatures, the probability of large moves in energy is large; at low temperatures the probability is small. If the cost function is reduced, the new configuration is accepted as the starting point for the next move.

2.1. Proposed simulated annealing algorithm

In order to obtain an optimal shaper using this algorithm, the following steps are to be taken:

1. Establish the sampling period T_s and the shaping time interval $\tau_{\text{range}} = \{N_{\min}T_s, \ldots, N_{\max}T_s\}$, where the set $\{N_{\min}, \ldots, N_{\max}\} \in \mathbb{N}$ and $N \in [N_{\min}, N_{\max}]$ is the shaper order equal to
$$N = \frac{\tau_s}{T_s} \tag{1}$$
2. Establish the number of temperature steps $T, \ldots, 0$ and the population P of individuals for each temperature step.
3. For each temperature step:
   (a) Generate a population of P individuals. In this work, and in order to reduce the processing time, we assume that individuals follow a monotonically increasing function until they reach the maximum, and then they follow a monotonically decreasing function. Thus, for each individual,
   $$I = \{x_1, x_2, \ldots, x_{N/2}\} \quad |0 \le x_1 \le x_2 \le \ldots \le x_{N/2} = 1 \tag{2}$$
   The shaper works as a digital Finite Impulse Response (FIR) filter. Thus x_n are the coefficients of the FIR filter.
   (b) Generate a shaper for each individual. In this paper, only symmetrical shapers are considered. Thus, the generated shaper is equal to
   $$\varsigma = \{x_1, x_2, \ldots, x_{N/2} = 1, \ldots, x_2, x_1\} \tag{3}$$
   (c) Combine ς with the current best shaper. The result will be S. The weight of ς with respect to the best shaper is proportional to T. If there is no current best shaper, S is not combined and $\varsigma = S$.
   (d) Evaluate S according to a cost function previously selected (see Section 2.2). If the cost function of the new shaper is lower than the cost function of the current best shaper, then the current best shaper is S.
4. At the end of the process, the optimal shaper will be the current best shaper.

For all the shapers considered, the flat-top duration is equal to T_s. When considering flat-tops with a duration of τ_t clock cycles, a number of ones equal to $L = \tau_t/\tau_s$ must be added in the middle of ς when attempting to generate the shaper using the individual. In this case,

$$\varsigma = \{x_1, x_2, \ldots, x_{N/2-L/2} = 1, \ldots, x_{N/2+L/2} = 1, \ldots, x_2, x_1\} \tag{4}$$

However, it is important to take into account that an increasing of the flat-top of a shaper implies an increase of parallel and $1/f$ noise.

The C-pseudocode of this algorithm is the following

```
N_range ⇐ {N_min…N_max};
bestDuration ⇐ meanValue(N_range);
bestShaper ⇐ {0…0};
bestMark ⇐ ∞;
for i=1 to T − for each temperature step
  for j=1 to P
    Ntmp ⇐ bestDuration + GenerateRandomNumber
   (N_min…N_max)/i;
    if Ntmp > N_max then
      N ⇐ Ntmp;
    else if Ntmp < 1 then
      N ⇐ 1;
    else
      N ⇐ Ntmp;
    end if;
    I ⇐ GenerateRandomIndividual(N);
    ς ⇐ Shaper(I);
    S ⇐ bestShaper +ς/i;
    if CalculateFunctionCost(S) < bestMark then
      bestMark ⇐ CalculateFunctionCost(S);
      bestDuration ⇐ N;
      bestShaper ⇐ S;
    end if;
  end for;
end for;
```

2.2. Cost functions

In this work, the cost function is the ENC for theoretical examples whereas for the real test, the cost function is the SNR.

2.2.1. ENC

As introduced in Section 1, the following three types of noise are considered for calculating the ENC: series noise, parallel noise and $1/f$ noise. Since the individual noise contributions are random and uncorrelated, they add in quadrature. Therefore, and according to Ref. [7], the ENC is equal to

$$\text{ENC}^2 = \frac{1}{2} i_n^2 N_S^2 + \frac{1}{2} v_n^2 C_i^2 N_\Delta^2 + \frac{1}{2} v_{fn}^2 C_i^2 N_F^2 \tag{5}$$

where v_n, i_n and v_{fn} are the spectral density of white series, white parallel and $1/f$ noise, respectively. C_i is the sum of all shunting capacitances of the input. Finally, N_S^2, N_Δ^2 and N_F^2 are the noise indices for white series, white parallel and $1/f$ noise, respectively, defined in Ref. [7] by $N_S^2 = 2F_i T_s$, $N_\Delta^2 = 2F_v/T_s$ and $N_F^2 = 2F_{vf}$.

Download English Version:

https://daneshyari.com/en/article/1822827

Download Persian Version:

https://daneshyari.com/article/1822827

Daneshyari.com