

Optimization Algorithms Testing and Convergence by Using a Stacked Histogram

Karel ZAPLATILEK¹, Miroslav TALPA², Jan LEUCHTER³
^{1,2,3}University of Defence, Kounicova 65, City of Brno 662 10, Czech Republic
 karel.zaplaticek@unob.cz, miroslav.talpa@unob.cz, jan.leuchter@unob.cz

Abstract—The article describes an original method of optimization algorithms testing and convergence. The method is based on so-called stacked histogram. Stacked histogram is a histogram with its features marked by a chosen colour scheme. Thus, the histogram maintains the information on the input digital sequence. This approach enables an easy identification of the hidden defects in the random process statistical distribution. The stacked histogram is used for the testing of the convergent quality of various optimization techniques. Its width, position and colour scheme provides enough information on the chosen algorithm optimization trajectory. Both the classic iteration techniques and the stochastic optimization algorithm with the adaptation were used as examples.

Index Terms—MATLAB, convergence, optimization, stacked histogram, stochastic system

I. INTRODUCTION

The field of optimization algorithms recently experienced a significant progress in development. Genetic and evolutionary techniques applicable in various areas of science are developed [1], [2]. These procedures have one feature in common. Without a few exceptions, they are the algorithms, the procedures of which are known or rateable in advance. Yet the case of optimization or regulation is mostly about searching for the function absolute extremes in a multidimensional optimization space [3], [4], [5]. Among the most significant parameters and features is a convergence of these methods. How the given algorithm converges to the correct result is as well important as its convergent quality. It depends on the algorithm's speed and trajectory. The whole process is rather complicated. Optimization and regulation algorithms differ from one another just in the convergent quality. Our solution is based on the so-called stacked histogram. The basic idea is as follows:

- The input sequence is marked by the chosen RGB colour scheme.
- The colour scheme is also used for the histogram.

All is noted in Fig. 1. Coloured, or stacked histogram has these basic advantages:

- Maintains the information about the order of the input sequence samples (thus enriches the histogram).

The research work is supported by the Grant Agency of the Czech Republic (project No. 102/09/0013) and University of Defence MATLAB Group (UDeMAG).

Digital Object Identifier 10.4316/AECE.2011.01002

- Identifies the statistical distribution probability density defects.
- Follows the histogram trajectory creation on the basis of the history of sequence creation.
- Distinguishes the stochastic processes with similar probability characteristics.

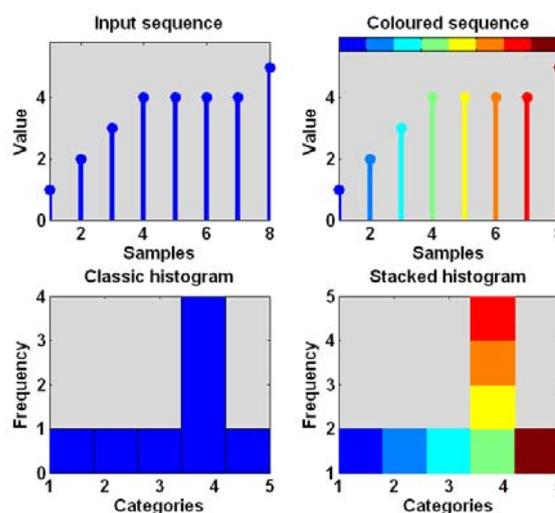


Figure 1. Stacked histogram illustration

To justify the use of the stacked histogram, we present two coloured histograms of two phase-shifted harmonic signals in Fig. 2. This simple picture demonstrates the aforementioned important feature of the stacked histogram: the distinction of the stochastic processes with similar probability characteristics.

The approach presented is suitable for monitoring of the optimization algorithms convergences with any strategy (deterministic, stochastic). In addition it serves for examination of the convergence quality, ergo the convergence trajectory. Stacked histogram in this role supplements the other ways used so far [6]. The examples and pictures presented in this article are 2D, but the method is not limited by the number of optimization space dimensions. In following chapters, we want to point out the stacked histogram ability for the mentioned purposes.

All of presented results and statements are verified by using a lot of optimization and analysis experiments. They can be generalized for higher-order dimensions of optimization spaces. In our department a special automated measuring workplace was established for digital data acquisition. Various stochastic systems can be explored and identified. The colour is suitable especially for systems operated by human.

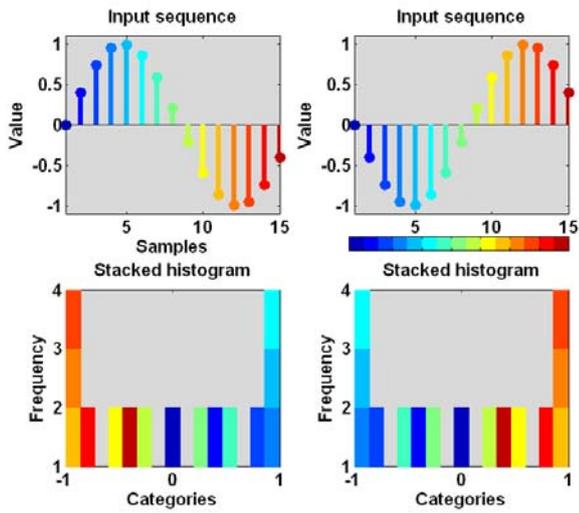


Figure 2. Harmonic function phase-shift distinction

II. STACKED HISTOGRAM OF AN OPTIMIZATION TRAJECTORY

The most common objective of an optimization is finding of the function roots or the absolute extremes. The optimization algorithm has a vector of parameters at its disposal, and searches their optimal values in order to meet the input requirements. If we mark the vector of parameters as $v_c(p_1, p_2, \dots, p_N)$, then the objective of the optimization is $v_c \rightarrow v_{c_optim}$ strategy. The N stands for the number of optimized parameters p_k and also for the optimization space dimension [3].

The way of finding the optimal parameters is called the optimization strategy (iteration, random). The quantitative evaluation of the distance to the optimum is then expressed by an optimization trajectory. Within the process of its determination, the purpose optimization function has the key role [4]. We will not repeat commonly known notions and names; we will focus directly on the essence. For the purposes of this article, we define a real function of one real variable by this formula:

$$y = 1 + e^{-0,3x} \sin\left(2\pi 0,6x + \frac{\pi}{4}\right) + 0,07 \sin(2\pi 3,6x).$$

The main task of the illustration algorithms mentioned below will be to evaluate the optimization quality and trajectory during the finding of the absolute minimum of given function in the interval of $x \in \langle 0,6;1,9 \rangle$. We will focus on the main principles with submitting that in practice the optimization space dimension is often very high. In general, the results are valid. The function's ripple can stand for superposed defects well-known from practice. The user can adjust the model according to his or her special usage. In the given interval, the function has only one absolute minimum. It was chosen for the purpose of this article's readability. In the case of multiple local extremes, the information above is valid with regard to the fact that stochastic methods are more complex [3]. We do not take an optimization purpose function to consideration because it is not necessary for understanding this principle. The purpose

function always regards concrete usage of the optimization techniques.

III. OPTIMIZATION TRAJECTORY OF ITERATION ALGORITHMS

The main objective is to display optimization trajectories while searching for the function's absolute minimum. A concrete implementation in practice gives always a physical meaning to the function. First, let us consider two iteration algorithms in Fig. 3. The first one is simple without lager intelligence. As a first step the so-called optimization is beginning task is solved, ergo we search for the vector $v_{c_ini}(p_{1_ini}, p_{2_ini}, \dots, p_{N_ini})$, then the coordinates of the function's absolute minimum. The user or the algorithm chooses only the solid iteration step or number of iterations without further limitations or changes. The stacked histogram is computed and rendered continuously. The colours of each point match with those of the histogram. The second iteration algorithm already has some intelligence. The system chooses the dynamic step on the X-axis and follows the changes in each step. The coloured histograms show the differences between the algorithms.

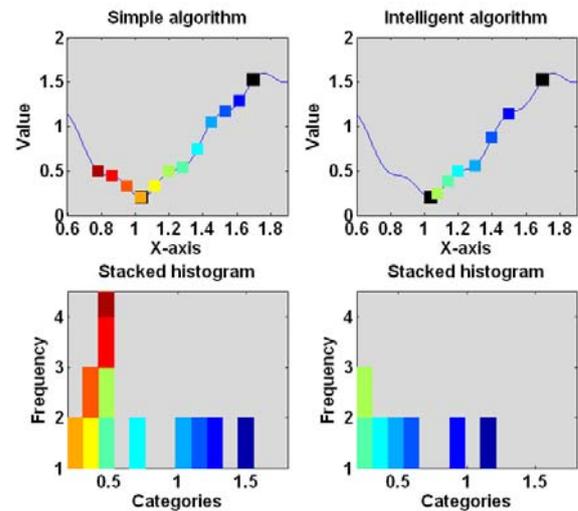


Figure 3. Optimization trajectory of iteration algorithms

After the evaluation of the histograms, following rules can be stated:

- The category on the very left is closest to the absolute minimum.
- The colour of the category on the very left corresponds to the convergence speed. The closer it is to the blue colour, the higher is the speed when using the classical RGB scheme.
- The histogram's width and scheme correspond to the convergence quality.
- The number of the categories is necessary to choose with regard to required accuracy.

Now we can formulate the ideal optimization iteration method requirements. They were formulated on the basis of practice experiments and experiences.

- The stacked histogram is rather narrow. It contains a

small number of colours and categories.

- The RGB colour scheme is fluent from blue on the right towards red on the left.
- The category on the very left represents the absolute minimum.

If the algorithm turns reverse, it can be easily revealed thanks to the colour scheme's regularity being disturbed. The warmer colours then spread the other way – left towards right. If the algorithm is divergent, it behaves according to the Fig. 4.

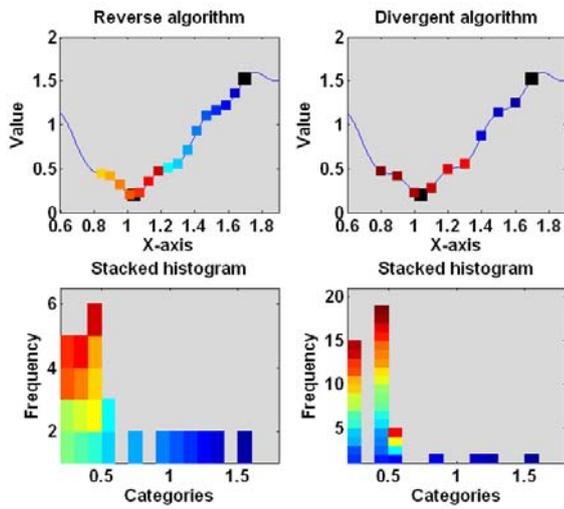


Figure 4. Reverse and divergent algorithms

The coloured histogram thus enables us to recognize the result as well as the trajectory to its reach. That is a very important statement. Let us repeat that in the experiments in practice, the situation is more complicated. In those cases, the stacked histogram plays an important part.

IV. OPTIMIZATION TRAJECTORY OF STOCHASTIC ALGORITHMS

Modern optimization techniques often employ a certain degree of randomness. The analyses show that it is an approach observed in Nature (sowing of cereals, irrigation etc.). These algorithms employ pseudo-random generators with different distribution of probability density and different parameters settings. At our working-site, we researched those systems more closely. The basic thought is good; however, there can be significant differences among algorithms concerning the convergence trajectory.

In Fig. 5 there is a comparison of two stochastic algorithms. The first one is a classic, one-pass with uniform probability density distribution according to the formula:

$$f_U = \frac{1}{x_2 - x_1} \text{ for } x \in \langle x_1, x_2 \rangle \text{ and } 0 \text{ outside, } x_2 \neq x_1, \quad (1)$$

where in this case $x_1 = 0,6$ and $x_2 = 1,9$ are the axis interval boundaries variable independently. The pseudo-random generator then generates the vector:

$$x_{unif} = 0,6 + (1,9 - 0,6) * rand_unif(1, Dim), \quad (2)$$

where Dim is a number of generated samples (20 in our case). We proved that the absolute extreme can be easily identified with the pseudo-random generator parameters chosen well [3]. The optimized vector parameters $v_{c_rand}(p_{1_rand}, p_{2_rand}, \dots, p_{N_rand})$ are tuned with regard to the given function's character. The histogram shows a good convergence quality. The passes of the 20 pseudo-random generator samples shows, however, that the optimization trajectory is not good. The system started to doubt in the extreme's surroundings rather than at the end. The adaptive stochastic system is of a higher quality. It employs a multiple-pass method. The generator parameters are set differently with each pass on the basis of the previous result. The picture shows the situation after the second pass. The higher accuracy of the minimum's finding is obvious. The histogram is narrow and the sample values are small. Their number was set to 8. The experience from practice revealed that 2-3 passes are sufficient to localize the extreme accurately, even in the cases of functions with local extremes. The method is very complex and independent from the initial guess of searched minimum's position. The number of the stacked histogram categories was chosen higher with regard to required accuracy.

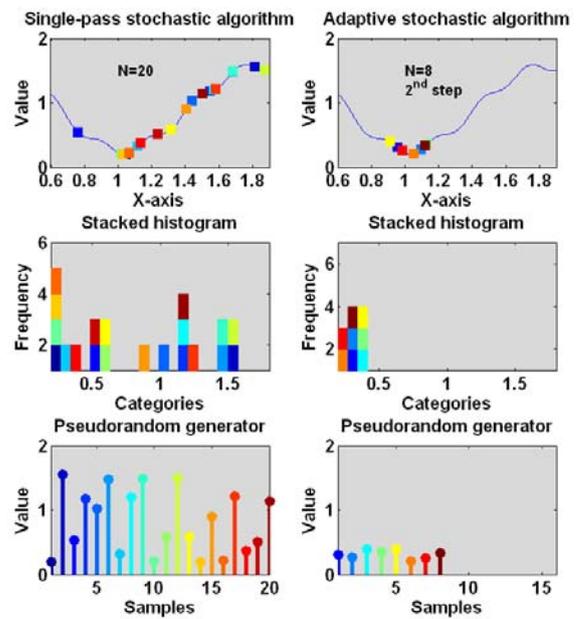


Figure 5. Classic and adaptive stochastic algorithms

In our case the distribution was uniform. In practice, it is suitable in some applications to choose steady distribution in the first step and the normal in the following steps according to the equation:

$$f_N(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}, \quad (3)$$

where σ is a standard deviation of the random variable x and m is a mean value. The system generates the vector:

$$x_{norm} = x_m + sqrt(D) * rand_norm(Dim), \quad (4)$$

where x_m is a mean value and D is a dispersion. The

choosing of the distribution type depends on the nature of the optimized problem. For example, when optimizing the frequency filters, all mentioned above is valid. More advanced algorithms can determine the parameters of the equation (4) automatically according to the previous steps' results and the extremes positions guess.

V. OPTIMIZATION PROCESS ADAPTATION ON THE BASIS OF CONTINUOUS HISTOGRAM

The use of colours in the histogram creation process enables both the visual and automatic evaluation of the optimization process. The stacked histogram in principle can be created using two methods:

- Creation after the optimizations or its part's ending (off-line).
- Continuous creation (on-line).

If the histogram is created continuously, it is possible to modify the algorithm settings and strategy within the optimization process. The use of histogram for the on-line algorithm's modification has not yet been published. Nevertheless, some of the recent techniques show a certain way of adaptation, however they are not common in the optimization process. The optimization system within its progress can follow these histogram parameters:

- The colour scheme's fluency (the minimum overpass, divergence, speed).
- The histogram's width (optimization interval, accuracy).
- The number of the categories (refers to the width, accuracy in the optimum's finding).
- Comparing of the two or more steps histograms (stochastic approaches), etc.

The only disadvantage can be a certain fall of the algorithm's speed, however in the most applications it is not a serious problem.

VI. THE STACKED HISTOGRAM'S CREATION IN THE MATLAB ENVIRONMENT

The common computing systems have not implemented the creation of the coloured histograms. It is possible in some but with difficulties and whit not user-friendly results (Microsoft[®] Excel[®], LabView[®]). In the MATLAB environment, there is an internal function *hist*, useable for the classic one-colour histogram with the chooseable number of categories [7]. Its syntax is intuitive: *hist(y,bin)* or *hist(y,x)*. For the stacked histogram is necessary to create the whole application, the so-called m-file. In Fig. 6 there is a real statement of the part of the stacked histogram creation source text using the *rectangle* function and the MATLAB programming language. The code is transparent and intelligible for the experienced MATLAB users. Redundant source rows and variable were removed. The presented list is universal a usable.

```
%-----source code for stacked histogram creation and drawing ---
M=length(Data); % data length
NoCat=15; % number of histogram categories
Color=colormap(jet(M)); % RGB colour scheme
Begin=Yini; % left limit of optim. interval
End=Yend; % right limit of optim. interval
Delta=(End-Begin)/NoCat; % one category length
Mat=zeros(M,NoCat); % auxiliary matrix
R=zeros(2,NoCat); % auxiliary vector
R(2,1)=Begin;
K=Begin;
%-----
for k=2:NoCat
    R(2,k)=K+Delta;
    K=R(2,k);
end
%-----
for k=1:M
    Inter=ceil((y(k)-Begin)/Delta);
    if Inter==0
        Inter=1;
    end
    row=M-R(1,Inter);
    Mat(row,Inter)=k;
    R(1,Inter)=1+R(1,Inter);
end
%--- stacked histogram drawing ---
subplot(3,2,3)
for k=1:NoCat
    for m=1:R(1,k)
        A=Mat(M-(m-1),k);
        P=[Begin+(k-1)*Delta m Delta 1];
        if Mat(M-(m-1),k)~=0
            rectangle('position',[P],'facecolor',Color(A),'edgecolor',Color(A))
        end
    end
end
end
%-----
```

Figure 6. Example of the stacked histogram creation in the MATLAB environment

The code includes the creation of the RGB colour scheme using the *colormap* function, auxiliary matrices and the *Mat* vector, and *R* for the right identification of colour in each part of histogram. The *Mat* matrix is important because its structure closely patterns the future histogram. The *rectangle* function determined for the histogram rendering uses a colour scheme in the *Color* variable. This variable, defined with the *colormap* function has $M \times 3$ dimension where M is a number of samples of the input sequence or its window. In the Fig. 6. source text there are the first two cycles '*for*'. Unfortunately, they cannot be replaced by the so-called vector approach, which is very useful in MATLAB, with regard to the code speed. The reason is the need to detect each sample's position in the final histogram due to the colour assign. If the input sequence length is large, the presented cycles enlarge the computing time. The advantage is the existence of the aforementioned *Mat* matrix which can be used even in the case when the histogram is not rendered. The possibilities of histogram creation in the MS Excel a Lab View environment are published in [8].

VII. PRACTICAL EXPERIMENTS

A. Analogue filter group delay optimization

The use of a coloured histogram in the optimization algorithms has not yet been published. It is used rather in the field of DSP, e.g. in the digital photography. We show the possible use of histogram on the example of the optimization of the 5th order Chebyshev type analogue filter group delay characteristic. The filter is optimized with a 1st order all-pass network connected in cascade. The main objective of the optimization is to find a value of ω_c , angular cut-off frequency for which the group delay ripple is minimal. The situation is shown in Fig. 7.

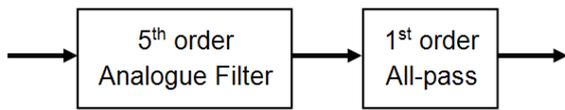


Figure 7. Main idea of the analogue filter group delay optimization

For the transfer function of the 1st order all-pass network applies this well-known formula:

$$\tau_{A1}(\omega) = \frac{2\omega_0}{\omega_0^2 + \omega^2} \quad (5)$$

The 1st order all-pass network is connected in cascade with the filter. The parameter which is optimized is only one, and that is ω_0 . The main strategy can be expressed with the equation: $v_{c_ini}(\omega_{0_ini}) \rightarrow v_{c_optim}(\omega_{0_optim})$. The optimization space $\Delta\tau_{T1} = f(\omega_0)$ is of the first order with one person independent variable. The $\Delta\tau_{T1}[s]$ sign stands for the group delay ripple. We will consider an angular cut-off frequency histogram. As mentioned before, the algorithm's task is to find the optimal value from the before-set interval $\omega_0 \in \langle 0, 1; 4 \rangle$, which will lead to the smallest $\Delta\tau[s]$ ripple. Fig. 8 shows the results. In the picture on the left, there is the group delay progress, both before and after the optimization. The picture on the right shows the stacked histogram of the filter-all-pass network cascade group delay characteristic.

It was found that the absolute minimum of the 1st order optimization space has a value of $\Delta\tau_{T1} = 2,3323[s]$ for the optimal value of $\omega_{0_optim} = 3,1667[rad.s^{-1}]$. In the case of the 2nd order all-pass network optimization, the optimized parameters are two: ω_r and Q . The optimization space is then of the 2nd order and there are two histograms in the output. In this case, the adaptive optimization algorithm was used. In the Fig. 8 picture shows the situation after the second step which was the final one. The whole optimization process was described more in detail in [3]. The picture obviously shows the pseudo-random generator's optimization trajectory. The frequency optimal value was found quite fast, at the end of the optimization cycle.

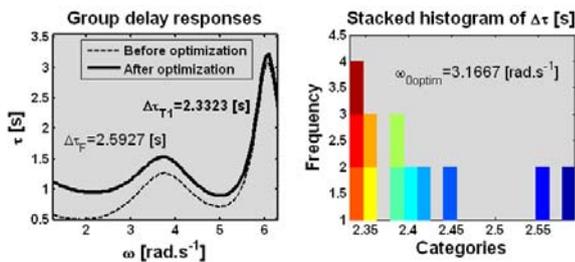


Figure 8. Stacked histogram of the $\Delta\tau$ ripple optimization

A lot of experiments were realized with various filter-order, various filter approximations and with a few all-pass networks. Next examples were published in [3].

This, almost ideal, behaviour was achieved with the careful tuning of the whole set of parameters in many previous experiments.

B. Power frequency instability evaluation

A power frequency was measured for a longer time. The total number of samples was 4096. The time response of the data is presented in Fig. 9. The nominal value is 50Hz in the Czech Republic. The frequency was measured by using the automated measuring workplace.

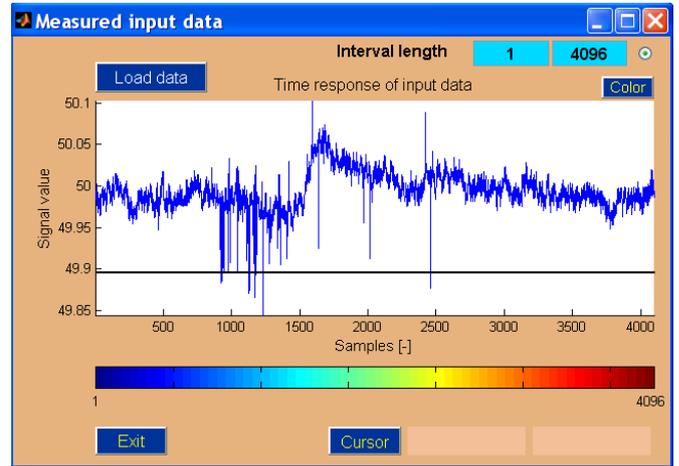


Figure 9. Measured power frequency (nominal value 50Hz)

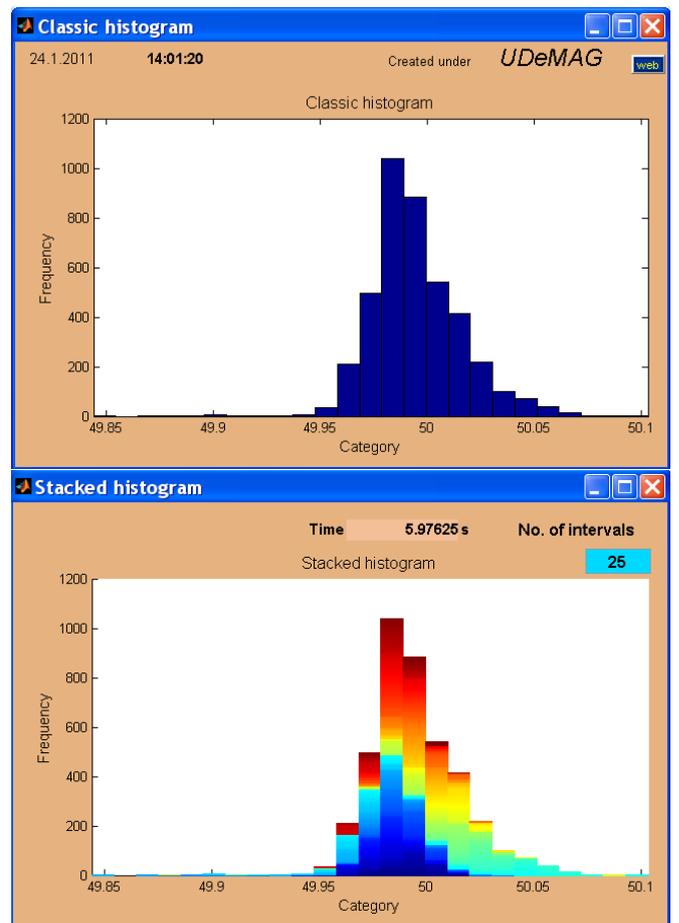


Figure 10. Classic and stacked histograms of power frequency

All of input samples are coloured by using RGB scheme (MATLAB command *Colormap*). Two histograms were

created: the classic one and the stacked histogram. Both of histograms are shown in Fig. 10.

Number of histogram categories was 25. The coloured histogram enables to study any inner stochastic processes. Both of histograms show statistic processing of whole input digital sequence. Our developed system enables to analyze not only whole signals but also its part by using the classic windowing. Whole data can be divided into a few windows. The coloured histograms analyses only defined window, see Fig. 11.

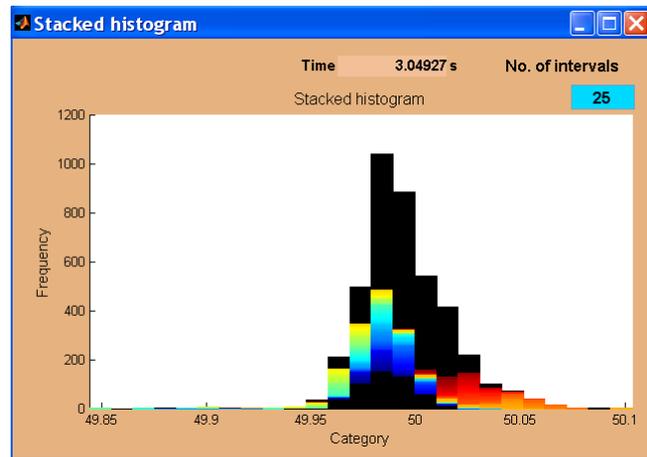


Figure 11. Stacked histogram of a signal window

The stacked histogram shown in Fig. 11 analyzed data from the interval 512-2048 samples (long data analysis).

C. Frequency generator stability

A stability of a generated frequency was explored and analyzed. The stacked histogram has a specific shape. There are a few discrete values along the category axis. It is a typical demonstration of the real stochastic system with a discrete distribution of a probability density. A chosen window of whole digital sequence can be also analyzed. The adjusted window can change its position. Then the stochastic system is analyzed using a floating stacked histogram. In case of a fast computer the whole described method works in real time.

VIII. CONCLUSION

This article describes the new method of the optimization algorithm trajectory evaluation. It maintains the information about the order of the input sequence values using the RGB colour scheme. That is why the algorithms qualities can be easily evaluated and compared with each other. The main advantage of this approach is the distinction of the random processes with similar probability characteristics. The stacked histogram can be used in both classic iteration techniques and the. Any similar approach has not yet been published at recent time. Many experiments were performed and it can be said that the stacked histogram is a suitable complement of the recent DSP methods and processes. The described methods can be extended and modified. The MATLAB offers the environment for similar experiments.

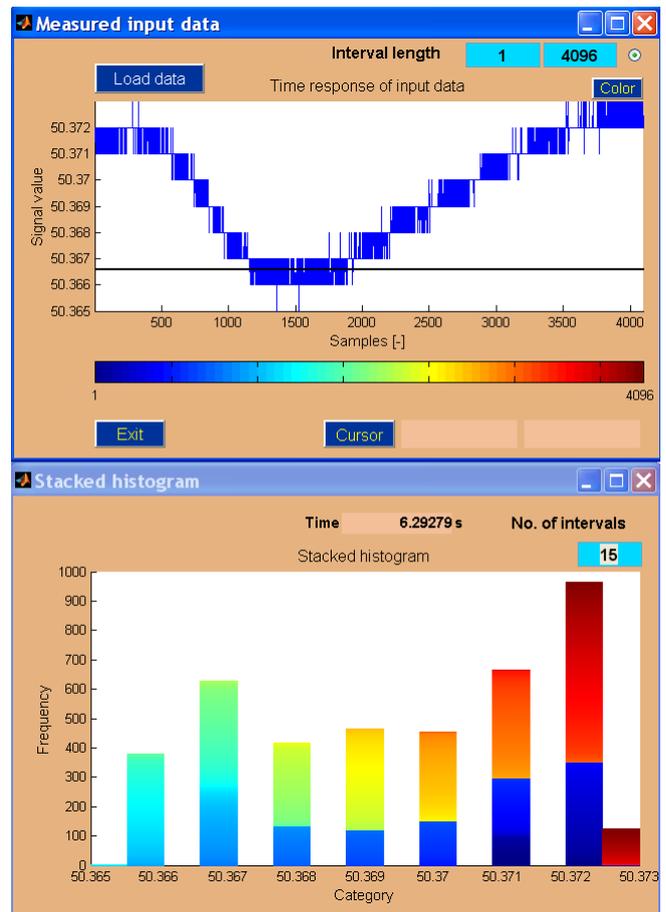


Figure 12. Stacked histogram of frequency stability

IX. ACKNOWLEDGMENTS

The research work has been supported by the Grant Agency of the Czech Republic (project No. 102/09/0013) and University of Defence MATLAB Group (UDeMAG).

MATLAB[®] is registered trademark of The MathWorks, Inc.

REFERENCES

- [1] P. Žiška, M. Laipert, "Analog Group Delay Equalizers Design based on Evolutionary Algorithm," *Radioengineering*, 2006, vol. 15, no. 1, pp. 1-5.
- [2] K. Deb, A. Anand, D. Joshi, "A computationally efficient evolutionary algorithm for real parameter optimization," *Evolutionary Computation Journal*, 2002, vol. 10, no. 4, pp. 371-395.
- [3] K. Zaplatilek, "Analogue Filter Group Delay Optimization Using a Stochastic Approach," *International Journal of Numerical Modelling*, 2010, vol. 23, no. 3, pp. 215-230.
- [4] K. Zaplatilek, P. Žiška, K. Hajek, "Practice Utilization of Algorithms for Analog Filter Group Delay Optimization," *Radioengineering*, April 2007, vol. 16, no. 1, pp. 7-15.
- [5] K. Zaplatilek, K. Hajek, "Efficient Algorithm for Group Delay Equalization of Analogue Filters," *Proc. of GSPx - International Signal Processing Conference*, April 2003, Dallas, TX, USA.
- [6] Q. Zhang, J. Sun, E.P.K. Tang, J.A. Ford, "Hybrid estimation of distribution algorithm for global optimization," *Engineering Computations*, 2003, vol. 21, no. 1, pp. 91-107.
- [7] J.H. Mathews, K.K. Fink, *Numerical Methods Using MATLAB*. Prentice-Hall Inc., 2004.
- [8] M. Talpa, K. Zaplatilek, "Histogram with higher information capacity," *Elektrorevue*, 2010, no. 62, 5p. Available: <http://www.elektrorevue.cz>. Published in Czech.