

# Robust probabilistic online change detection algorithm based on the continuous wavelet transform

Sergei Yendiyarov, Sergei Petrushenko

*Abstract*—In this article we present a change point detection algorithm based on the continuous wavelet transform. At the beginning of the article we describe a necessary transformation of a signal which has to be made for the purpose of change detection. Then case study related to iron ore sinter production which can be solved using our proposed technique is discussed. After that we describe a probabilistic algorithm which can be used to find changes using our transformed signal. It is shown that our algorithm works well with the presence of some noise and abnormal random bursts.

*Keywords*—Change detection, sinter production, wavelet transform.

## I. INTRODUCTION

CLASSICAL online change detection algorithms (e.g. sequential Wald analysis [7], CUSUM, Bayes-type, Filtered Derivative [2] et al.) depend upon the underlying distribution of a signal also they are heavily influenced by random abnormal bursts and noise. Moreover the overall performance of these algorithms decreases significantly when we have to deal with sampled signals with the given sampling interval. The more the sampling interval is the worse these algorithms can find change points in the signal.

In this article we consider a robust probabilistic change detection algorithm which is based on the continuous wavelet transform. Our proposed algorithm can cope with the above-mentioned difficulties efficiently. In this article we will give detailed information about how to transform our original signal for the purpose of change detection. A case study related to iron ore sinter production which can be solved using our proposed technique is discussed.

Now we briefly describe the necessary steps which we should follow. The first step is to find the continuous wavelet transform of the original signal. After the transformation has been made we take absolute maxima and minima values of the transformation at each translation. As a result of this operation we have two signals. Finally, we take both these signals and aggregate them in a new signal which we will call the aggregated signal. This signal we will use in our change point detection algorithm in order to find change points in our original signal. In the following sections we will give detailed descriptions for each of these steps.

## II. ORIGINAL SIGNAL PREPROCESSING

As we mentioned before first of all we have to find the continuous wavelet transform of the original signal. But before

S. Yendiyarov and S. Petrushenko are with the Department of computer science, Ural state mining university, 6201448, Yekaterinburg, Russia

that we have to preprocess our original signal in order to get the right results. Taking into account that every wavelet function has the following property (1) [4].

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (1)$$

This property means that in general the mean value of a wavelet function is equal to zero. Due to this property our aggregated signal will be seriously influenced at the beginning and at the end of it (i.e. only part of our wavelet function will be multiplied with our original signal). Therefore our aggregated signal will have sufficient spikes at the beginning and at the end of it.

To deal with this problem before transforming our original signal we need to extend it so we have to add additional element to the beginning of our signal and to the end of it. It can be done if we know compact support of a wavelet function.

In this article we will use the Gaussian wavelet function. The Gaussian wavelet function of  $n$ th order looks like (2) [3].

$$\Psi_n(x) = (-1)^{n+1} \frac{d^n}{dx^n} e^{-\frac{x^2}{2}} \quad (2)$$

In our case we will use first order the Gaussian wavelet function (3).

$$\Psi_1(x) = \frac{d}{dx} e^{-\frac{x^2}{2}} = -xe^{-\frac{x^2}{2}} \quad (3)$$

The first order Gaussian wavelet function has compact support on the interval  $x \in [-5; 5]$ . Therefore for this particular wavelet function necessary amount of elements  $L$  which should be added to our original signal can be calculated as follows (4).

$$N^+ = L \cdot R \quad (4)$$

Where  $R$  is width of the right or the left part of our wavelet function and  $L$  is the maximum value of the scaling factor. For example in the case of the Gaussian wavelet function we have  $R=5$  and for the given maximum scaling factor  $L=128$ , then amount of additional points which has to be added to our original signal is  $N^+ = 128 \cdot 5 = 640$ .

However this amount of points can be reduced considering the fact that the Gaussian wavelet function decreases very fast. Therefore in practice for the value of  $R$  we can use  $R=3$  without sufficient loss of precision.

Let  $U(t)$  be a value of the original signal at the point  $t$  then  $U(1)$  is some value of our signal at the point  $t=1$  and  $U(n)$  is some value of our original signal at the point  $t=n$  accordingly. Thus using our notation the original signal should be extended in the following way (5).

$$\tilde{U}(t) = U(0)_{N^+} + U(t) + U(N)_{N^+} \quad (5)$$

Where  $N$  is the length of the signal and expression  $U(0)_{N^+}$  means adding to the signal  $N^+$  elements of  $U(0)$ .

Using this modified signal we now can use our wavelet transform. General formula for the continuous wavelet transform looks like (6).

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t) \Psi\left(\frac{t-b}{a}\right) dt \quad (6)$$

Because in practice almost every signal is just a digitized version of a continuous signal which is represented by a sequence of digits. Therefore to find the continuous wavelet transform of some discrete signal we need to modify the formula of the continuous wavelet transform. The continuous signal can be represented in terms of some discrete signal  $\tilde{U}(t)$  as follows (7) [1].

$$\tilde{U}_\delta(t) = \tilde{U}(nT) = \sum_{n=-\infty}^{\infty} \tilde{U}(t) \cdot \delta(t - nT) \quad (7)$$

Where  $\tilde{U}_\delta(t)$  is the continuous signal which has nonzero values at the points  $\tilde{U}(nT)$ .

Substituting expression for  $\tilde{U}_\delta(t)$  into the formula for the continuous wavelet transform we will get (8).

$$\begin{aligned} W(a, b) &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [\tilde{U}(t) \cdot \delta(t - nT)] \cdot \Psi\left(\frac{t-b}{a}\right) dt = \\ &= \frac{1}{\sqrt{a}} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{U}(t) \cdot \delta(t - nT) \cdot \Psi\left(\frac{t-b}{a}\right) dt \end{aligned} \quad (8)$$

Taking into account properties of Dirac delta function, finally we will get (9).

$$W(a, b) = \frac{1}{\sqrt{a}} \sum_{t=0}^{N-1} \tilde{U}(t) \cdot \Psi\left(\frac{t-b}{a}\right) \quad (9)$$

Now when we know how to calculate the continuous wavelet transform of our modified signal we can find the values of maxima and minima of our wavelet transform.

Formulae for calculating absolute maxima and minima values of our wavelet transform look like (10) and (11).

$$\hat{W}_{\min}(b) = \left| \min_{1 \leq a \leq K} \{W(a, b)\} \right| \quad (10)$$

$$\hat{W}_{\max}(b) = \left| \max_{1 \leq a \leq K} \{W(a, b)\} \right| \quad (11)$$

Where  $K$  is the maximum scaling factor of our wavelet transform.

Then we should reduce the size of the signals  $\hat{W}_{\min}(b)$  and  $\hat{W}_{\max}(b)$  back to the original size. In order to do so we remove  $N^+$  elements from the beginning and from the end of these

signals. Let denote these reduced signals as:  $W_{\max}(b)$  and  $W_{\min}(b)$ .

After this manipulation has been made we can calculate the aggregated signal which will be used later. The aggregated signal is calculated as follows (12).

$$W_{\Sigma}(b) = \sum_{b=0}^{N-1} W_{\max}(b) + W_{\min}(b) \quad (12)$$

Where  $b \in [0, N-1]$  and  $N$  is the length of the original signal

Formulae for calculating the expected value and the variance of the signals  $W_{\min}(b)$  and  $W_{\max}(b)$  through all scaling factors are given by (13) and (14).

$$E[W_{\max}(b)] = \frac{\sum_{b=0}^{N-1} \max_{1 \leq a \leq K} \{W(a, b)\}}{N} \quad (13)$$

$$D[W_{\max}(b)] = \sum_{b=0}^{N-1} (W_{\max}(b) - E[W_{\max}(b)])^2 / N \quad (14)$$

The variance of the aggregated signal can be found from the expression (15).

$$D_{\Sigma}[W_{\Sigma}(b)] = D[W_{\max}(b)] + D[W_{\min}(b)] \quad (15)$$

Although the mean value of the wavelet coefficients is equal to zero this is not true for our case. Because after this transformations have been made the mean value of our aggregated signal is not equal to zero.

Due to this the expected value of our aggregated signal can be calculated similarly to the variance (16).

$$E_{\Sigma}[W_{\Sigma}(b)] = E[W_{\max}(b)] + E[W_{\min}(b)] \quad (16)$$

Initial values for the variance and the expected value are needed when we will deal with our change detection algorithm.

### III. CASE STUDY

Basically algorithms which can be used to find change points of a signal have broad range of applications. In our study we will consider iron ore sinter production plant. More precisely we will talk about the problem which arises at the ore stockpile where blending of iron ores takes place [6]. Different flows of ore and limestone from different sources usually come to the ore stockpile where these flows are being blended.

These flows usually have different chemical compositions (usually we are interested in the content of CaO in the limestone and Fe in the ore).

There are some major problems which are: 1) The chemical compositions of some flows are not controlled, 2) chemical analysis of a flow is carried out in discrete moments of time and the time period between two successive analyses is quite long (more than one hour) meanwhile the chemical composition of a flow is considered constant between successive analyses which is wrong (the chemical compositions of flows are fluctuating in a wide range of values), 3) The

operator of the ore stockpile calculates the average chemical composition of an ore pile using the chemical analyses of flows and consumptions of these flows. Then based on this average chemical composition, control of the sintering process is carried out. This average chemical composition can be far away from the chemical compositions of distinct parts of the ore pile.

Taking into account these problems we are considering the case when there is a sensor which is placed before the bins of a sinter machine. The sensor has abilities to measure the chemical composition of the blended ore every 10 minutes (which is relatively fast). Our task is to stabilize the chemical composition of the sinter to this end we should detect changes in the trend of the chemical composition (Fe and CaO) of the blended ore using abovementioned sensor.

We are assuming that because of the abovementioned problems our ore pile can be divided into some homogeneous parts with each part having the homogeneous chemical composition. If we can detect changes in the trend (i.e. moment of time when we move from one homogeneous part to another) we can change the chemical composition of the sinter by changing consumption of the flows.

#### IV. ROBUST ONLINE CHANGE DETECTION ALGORITHM

In the previous section we described the necessary transformations of the original signal. As a result of these transformations we have got our aggregated signal  $W_{\Sigma}$ .

In this section we are going to discuss an online algorithm of change detection which will work with our signal  $W_{\Sigma}$ . We construct this algorithm on the basis of statistical hypothesis testing which is the most common thing for different change detection algorithms (e.g. sequential Wald analysis, CUSUM et al.).

So in general we deal with two hypotheses: a null hypothesis  $H_0$  and an alternate hypothesis  $H_1$ . The null hypothesis means that there is no change in the mean of our signal. Conversely,  $H_1$  means that there is a change in our signal.

Let  $p_{u \text{ lim}}$  denote the probability of event  $H_1$  with respect to that  $\alpha = 1 - p_{u \text{ lim}}$  is the probability of event  $H_0$  accordingly. In accordance with the multiplication theorem for independent events we can define the joint probability as follows (17) [5].

$$P_{\Sigma}(t) = P(W_{\Sigma}(1)) \cdot P(W_{\Sigma}(2)) \cdot \dots \cdot P(W_{\Sigma}(t)) = \prod_{j=1}^t P(W_{\Sigma}(j)) \quad (17)$$

We should continue our inspection till  $P_{\Sigma}(t) > p_{u \text{ lim}}$ . To do our algorithm more robust with respect to the underlying distribution of the signal  $W_{\Sigma}$  we start our inspection when our signal  $W_{\Sigma}$  crosses given level  $H_U$ .

Parameter  $H_U$  can be a vector in case when we need to have more customization (18).

$$H_U^T = (h_0, h_1, \dots, h_k, \dots, h_n) \quad (18)$$

So we applied the technique which is used by control charts algorithms and we start to calculate our joint probability when  $W_{\Sigma}(t) \geq H_U(j)$ . Where  $H_U(j)$  denotes  $j$ 's element of the vector  $H_U$  (19).

$$H_U(j) = E[W_{\Sigma}(b)] + \Lambda(j) \cdot \sqrt{D[W_{\Sigma}(b)]} \quad (19)$$

Where  $\Lambda = (\lambda_0, \lambda_1, \dots, \lambda_n)$  is our vector of parameters. Then we calculate the probability that  $W_{\Sigma}(t) \geq H_U$  which can be found from the probability theory (20).

$$P(W_{\Sigma}(k) \geq H_U(j)) = 1 - \int_{-\infty}^{H_U(j)} f(W) dW \quad (20)$$

Where  $f(W)$  is the cumulative distribution function of our signal  $W_{\Sigma}$ .

In the case of the normal distribution this equation can be rewritten in this form (21)

$$P(|W_{\Sigma}(k) - E[W_{\Sigma}(k)]| < \varepsilon) = \Phi(\varepsilon/\sqrt{D[W_{\Sigma}(b)]}) \quad (21)$$

Taking into account that we have the one-sided condition  $W_{\Sigma}(t) \geq H_U$  finally we can write our expression in this form (22)

$$P(W_{\Sigma}(k) - E[W_{\Sigma}(k)] \geq H_U(j)) = \frac{1}{2} - \frac{1}{2} \Phi(H_U(j)/\sqrt{D[W_{\Sigma}(b)]}) \quad (22)$$

Where  $\Phi(x)$  is the Laplace function which is given by (23)

$$\Phi(x) = \frac{2}{\sqrt{2\pi}} \int_0^x e^{-x^2/2} dx \quad (23)$$

And  $\lambda_j$  are the tuning parameters and can be used to define some specific behaviour of the algorithm and they depend on both the signal characteristics and the type of changes.

For example, for the normal distribution we can set  $\Lambda$  to be  $\Lambda^T = (1, 2, 3)$ , in the case when the variance of our signal is known otherwise we can use t-table.

To do our algorithm more robust to some random abnormal bursts we should set  $P_{\Sigma}(t) = 1$  every time when  $W_{\Sigma}(t)$  becomes less than  $H_U$  (24).

$$W_{\Sigma}(t) < E[W_{\Sigma}(b)] + \Lambda(j) \cdot \sqrt{D[W_{\Sigma}(b)]} \quad (24)$$

After we have detected some change point we should have to turn off our detector and start only when we have gone down from the hill (see figure 1 for example).

For this reason we introduced a new parameter  $H_D$ . The parameter  $H_D$  can be set in the following way: after we have detected a change in our signal we should start our inspection again only when  $W_{\Sigma}(t) \leq H_D$ .

In order to do so we defined this parameter as follows (25).

$$H_D = E[W_{\Sigma}(b)] + \beta \cdot \sqrt{D[W_{\Sigma}(b)]} \quad (25)$$

Where  $\beta$  is the tuning parameter which along with the  $\lambda_j$  can be set depending on both the signal characteristics and the type of changes.

Therefore, in the case of the normal distribution we can put  $\beta = 0$ , this means that we wait until our random variable  $W_{\Sigma}(t)$  crosses its expected value. It is worth noting that we can also set our parameter  $\beta$  in the following way  $\Omega^T = (\beta_0, \beta_1, \dots, \beta_n)$ . Thus, we only need to set these parameters:  $p_{u \text{ lim}}$ ,  $H_U$  and  $H_D$  to use our algorithm.

## V. EXPERIMENTAL SETUP AND RESULTS

For testing our algorithm we used two signals (CaO and Fe) generated by the following formula (26).

$$S(t, n) = N(M(n), \sigma^2) + Z(t) \cdot \varepsilon \pm N(0, \tilde{\sigma}^2) \quad (26)$$

Where  $N(M(n), \sigma^2)$  is a normally distributed random variable with the mean  $M(n)$  and the variance  $\sigma^2$ ,  $Z(t) \cdot \varepsilon$  is an element which is used to introduce abnormal random bursts in the signal (where  $Z(t)$  is a random variable and it takes only two values 0 or 1 with the probability  $p=0.5$  and  $\varepsilon$  is also a random variable it takes values within the range  $a \leq \varepsilon \leq b$  and it has the uniform distribution),  $N(0, \tilde{\sigma}^2)$  is a normally distributed random variable with the zero mean and the given variance  $\tilde{\sigma}^2$  and it represents a noise which influences our signal (It is a noise which is introduced by our sensor).

The random variable  $N(M(n), \sigma^2)$  represents our real signal and the whole expression (26) represents the corrupted signal. Taking into account that our sensor can measure only every ten minutes before using our signal  $S(t, n)$  we should sample it every ten minutes. By doing so we get the signal which is being measured by our sensor i.e. the measured signal.

For simulation purposes we took following values for the abovementioned parameters:  $\tilde{\sigma}_{Fe}^2 = 0.5^2$ ,  $\tilde{\sigma}_{CaO}^2 = 0.3^2$ ,  $\sigma_{Fe}^2 = 1^2$ ,  $\sigma_{CaO}^2 = 0.5^2$ ,  $-5 \leq \varepsilon_{Fe} \leq 5$ ,  $-1 \leq \varepsilon_{CaO} \leq 1$ . We based on the real signals of Fe and CaO along with the information about the sensor.

Initial values for the means of our signals:  $M_{Fe}(0) = 60.7$  and  $M_{CaO}(0) = 3.71$ .

We introduced six change points in the mean of the every signal (i.e. we were changing  $M(n)$  at random moments of time).

For instance, the expression for the Fe signal which is introduced a change in its mean value (27).

$$M_{Fe}(n+1) = M_{Fe}(n) \pm Z(t) \cdot 2\sigma_{Fe} \quad (27)$$

After applying our algorithm to the signal of Fe we got the results which are shown in figure 1. From this figure it is clearly seen that our algorithm found too many redundant points.

## VI. MODIFIED VERSION OF THE ALGORITHM

The general cause of such behaviour is the presence of some noise and some abnormal bursts in our signal.

Let's consider this problem in more detail (see figure 2). From this figure you can see that our algorithm detected two points. First of all it detected one point (the left one) and then our algorithm turned off according to this condition  $W_{\Sigma}(t) \leq H_D$  until it crossed the expected value of the signal  $W_{\Sigma}(t)$  (in figure 2 blue asterisk line shows the expected value of  $W_{\Sigma}(t)$ ). It can be seen from this figure that the second point (the right one) was detected due to some abnormal burst which occurred in the signal.

In order to make our algorithm robust with respect to occasional random bursts we introduced the following parameter  $P_{\Sigma}^D(t)$ . It has almost the same meaning as our joint probability which was discussed earlier. There is only one difference

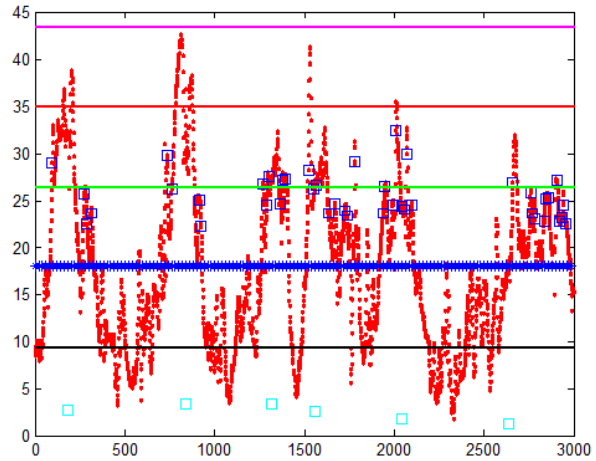


Fig. 1. Application of our algorithm to the signal of Fe (Blue squares are points which are detected by our algorithm, cyan squares are real changes of the signal; x-axis is time (in minutes) and y-axis is values of our aggregated signal  $W_{\Sigma}(t)$ ).

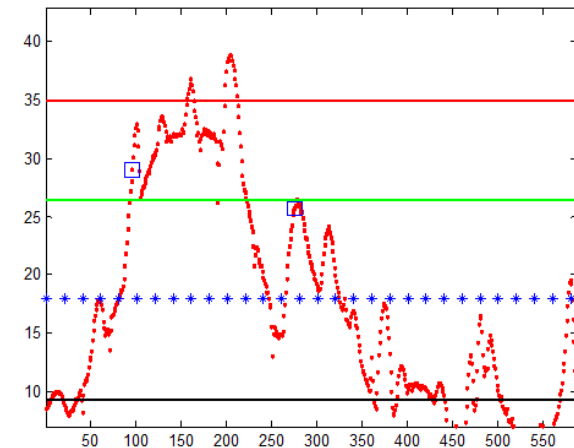


Fig. 2. Redundant point (the right one) which was wrongly detected by our algorithm due to some abnormal random burst in the signal.

between these two parameters, we start to calculate  $P_{\Sigma}^D(t)$  as soon as  $W_{\Sigma}(t) \leq H_D$ . Meanwhile we are calculating the probabilities that the random variable  $W_{\Sigma}(t)$  crossed the boundary value of  $H_D$ .

Analogously to the joint probability we should set  $P_{\Sigma}^D(t) = 1$  whenever  $W_{\Sigma}(t)$  becomes greater than  $H_D$  (28).

$$W_{\Sigma}(t) > E[W_{\Sigma}(b)] + \beta \cdot \sqrt{D[W_{\Sigma}(b)]} \quad (28)$$

Let  $p_{d\lim}$  denote the tolerance probability for this case. Our experience shows that in practice it can be set in the following way  $p_{u\lim} = p_{d\lim}$ . It is worth noting that  $H_D$  can be used as the tuning parameter. (For instance, in the case of close events (i.e. signal changes) it is recommended to increase the value of  $H_D$ ).

Figure 3 shows the results for the signal of Fe after modification of our algorithm. It is seen from the figure that our algorithm works well on our testing signal of Fe. To test

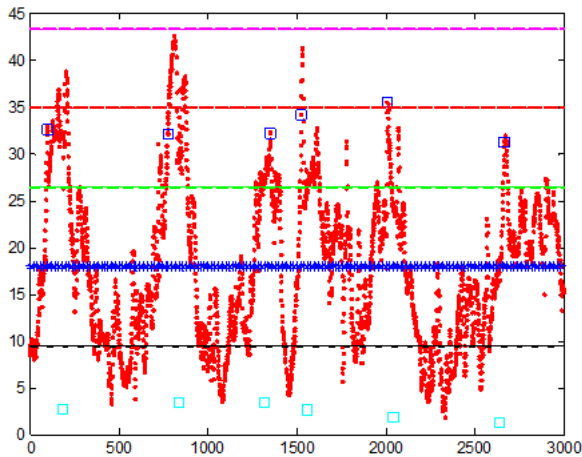


Fig. 3. Results for the signal of Fe (Blue squares are points which are detected by our algorithm, cyan squares are real changes of the signal)

TABLE I  
RESULTS OF OUR ALGORITHM ON TWO TESTING SIGNALS

Parameters for the signal of Fe	Detected points	Expected value	False detection	Variance
$\Lambda^T = (1.5, 2, 2.5, 3)$ $\beta = 0, p_{u\lim} = 0.005$	6	44,2	0	368,8
Parameters for the signal of CaO	Detected points	Expected value	False detection	Variance
$\Lambda^T = (1, 2, 2.5, 3)$ $\beta = 0, p_{u\lim} = 0.006$	6	41,8	0	350,4

our modified algorithm we generated signals for Fe and CaO many times and then we used our algorithm to detect change points. The algorithm shows good results for both the signal of Fe and the signal of CaO as it is able to detect all six change points in our signals.

Parameters we used to test our algorithm along with the estimations of the expected value and the variance of the detection are summarized in table 1.

## VII. CONCLUSIONS

In this article we described the robust online change detection algorithm which can be used to detect changes in signals with some presence of noise and abnormal bursts.

We considered the case study which is related to sinter production. In this article we assumed that the mean values and the variances of the both signals are constants if it is not true they can be calculated iteratively.

It is worth noting that despite the fact that in this article we used the normal distribution for calculating probabilities our algorithm can work well with different types of distributions because it has the tuning parameters and it uses the starting rule which is used by control charts.

Finally we presented the results for our testing signals of Fe and CaO which show that our algorithm works well and it is able to detect all change points, in spite of the fact that the signals were sampled every 10 minutes.

## REFERENCES

- [1] Popov A., Zhukov M., *Computation of continuous wavelet transform of discrete signals with adapted mother functions*, Proceedings of the SPIE, vol. 7502, pp. 75021E-75021E-6, 2009.
- [2] Bassville M., Nikiforov I., *Detection of abrupt changes. Theory and application*, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
- [3] Burrus C., Ramesh A., *Introduction to wavelets and wavelet transforms: A Primer*, Prentice-Hall, New Jersey, 1998.
- [4] Charles K., *An introduction to wavelets*, Academic Press, New York, 1992.
- [5] Degroot M., Schervish M., *Probability and Statistics*, Pearson Education, Carnegie-Mellon University, 2011.
- [6] German R., *Sintering theory and practice*, John Wiley and Sons, New York, 1996.
- [7] Wald A., *Sequential Analysis*, John Wiley and Sons, New York, 1947.