IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 36, No. 4, August, 2019, pp. 303-310
Journal homepage: http://iieta.org/journals/ts

Generating Random Numbers from Biological Signals in LabVIEW Environment and Statistical Analysis

Duygu Kaya[1*], Seda Arslan Tuncer[2]

[1] Department of Electrical-Electronics Engineering, Faculty of Engineering, Firat University, Elazig 23119, Turkey
[2] Department of Software Engineering, Faculty of Engineering, Firat University, 23119 Elazig, Turkey

Corresponding Author Email: dgur@firat.edu.tr

https://doi.org/10.18280/ts.360402

Received: 14 April 2019
Accepted: 21 July 2019

Keywords:
true random number generator (TRNG), biological signal, electromyographic (EMG) signal, LabVIEW, statistical test

ABSTRACT

This paper explores the generation of random numbers, using electromyographic (EMG) signals collected from arm, elbow and finger movements of healthy individuals. The original signals were extracted from the Ninaweb database. The author designed a new discretization algorithm to convert these signals from floating point numbers to discrete values, and proposed a true random number generator (TRNG) structure that obtain the EMG signals with human arm and finger movements as noise sources. The proposed algorithm was applied to obtain and process real-time signals in the LabVIEW environment, and verified through NIST, TestU01, Scale index and autocorrelation tests. The results show that the discretization algorithms in TRNGs faced a huge data loss (70 %), while the designed algorithm with our structure lost no data and achieved 100 % efficiency in number generation. The research results prove the possibility of generating random numbers from biological signals.

1. INTRODUCTION

Random number generators (RNG) are algorithms designed to produce number sequences that appear random. RNG's play an important role in cryptography, machine learning, simulation, game theory, industrial tests and labeling, and for games used in lotteries and gambling. Pseudo-random number generators (PRNGs) are deterministic methods that use mathematical algorithms. To obtain random number sequences from PRNG's, initial values known as seeds are required. If the seed is hidden and the algorithm is designed well then it is possible that the hidden number will be unpredictable. The advantage of PRNG's are their ability to easily produce numbers at a very low cost, especially in hardware such as mobile phones and computers [1, 2]. RNG's are expected to produce a great number of random numbers over a short period of time in a high quality way. Random numbers are needed in applications like stochastic simulation, flow passwords and online gambling, and for that reason, PRNG's are preferred in applications where the rate of number generation is an important parameter. However, there are true random number generators, which are non-deterministic, physical random number generators (TRNG's). Today, true random number generators are required in many diverse fields of everyday life, such as cryptography, mobile communication, e-mails, online payments, cash-free payments, ATM's, e-banks, e-sales, sales points and prepaid cards [3, 4].

Different things are expected from TRNG's compared to PRNG's. The most important difference is that numbers obtained from TRNG's are dependent upon noisy sources. In general, raw sequences obtained from physical sources do not show totally random behavior. For that reason, post-processing algorithms are needed to have equally distributed numbers of 0s and 1s. Even though the distribution of 0s and 1s are balanced out with post-processing algorithms, the number of useful bits get less and less, and this causes the efficiency of the generator to drop significantly. Another problem is that TRNG's are expensive and they require another hardware element, and they are especially slow when considering the applications that have been mentioned.

This study describes true random number generation using EMG signals obtained from the movements of fingers and wrists, from grasping and functional motions and from force patterns with the aid of sensors and without the use of post-processing algorithms. To realize the suggested system a discretization algorithm that will transform the continuous time signals into discrete time signals is suggested. Bit production efficiency is 100 % in this suggested system, since post-processing algorithms are not used.

The contributions of this study to the existing literature are shown below:

Different biological signals were used as physical sources.

A new algorithm was used to transform continuous time signals into discrete time signals.

Number generation efficiency was 100% and there was no loss of data.

The article is organized as follows.

Section 2 summarizes TRNG structures made from noise sources that exist in published literature, along with their advantages and disadvantages. In Section 3, presents the properties of the biological signals used. Section 4 explains how the random numbers are produced using the biological signals and how the discretization algorithm works. The statistical tests used to examine the randomness of the numbers obtained is explained in Section 4. In Section 5, the concluding part, the results obtained are presented and the advantages of the suggested system are discussed.

2. RELATED WORKS

One of the most important factors in true random number generation is the source of the noise. In cases where true random number generation is possible from raw number sequences obtained from noise sources, the statistical qualities of these random numbers are not always good enough. To eliminate this disadvantage, post-processing algorithms are applied. In the literature reviewed, the noise sources used have been thermal noise, clock jitter and nuclear decay. In addition, signals from human voices, videos, mouse movements, EEG measurements, ECG measurements, blood pressure and telegraph noise have all been used to generate random numbers. Hu et al. generated random numbers from user's mouse movements [5]. In their studies, they proposed 3 different chaos-based algorithms to eliminate the similar movement patterns of different users. This algorithm, which passed NIST tests, was successful with respect to effectiveness and efficiency. T-Chen et al. proposed an efficient random number generator that can be used with voice communication, where the voice was used as the noise source. In their studies, the random number generator required hardware such as a microphone or a cell phone. For that reason, random number generation was performed using common devices and without special equipment. A filter was applied to increase the success rate in NIST tests, and the success rate was increased [6]. In another study by T-Chen, a random number generator was produced by making use of white noise obtained from audio and visual (A/V) sources of high resolution, IPCAM, WEBCAM and MPEG-1 video files. To evaluate the statistical qualities of the numbers obtained, the NIST SP 800-22 test was applied, and a 98 % success rate was achieved. One of the greatest advantages of this test is that audio and visual sources can be found very easily, making the generator quantifiable, efficient and handy [7].

Iba et al. tried to answer the question: "Can We Behave as a Random Number Generator" [8]. After this study, they worked on random number generation using biological signals obtained from humans [9-15]. Schulz et al. requested 20 individuals to form a random number sequence ranging between 1 and 19. In the random sequence obtained, it was shown that individual characteristics could be identified [9].

Jokar et al. showed that there is no similarity between random numbers generated by different individuals and that these numbers can be used as biometric signatures [10]. Szczepanski et al. suggested the use of biological data to form random bit sequences. They showed that this new approach could help to produce seeds for pseudo-random number generators. They suggested a very basic algorithm based on the observation that biometric data shows randomness. The method was first applied to animal neurophysiological brain reactions and then tested on human galvanic skin reactions. To verify the cryptographic quality of biometric generators against the FIPS 140-2 standard, Maurer's universal test – which is commonly suggested – and the Lempel-Ziv complexity test – which guesses the entropy of the source – are used. Verification results, after choosing appropriate coding and experimental parameters, showed that the sequences obtained showed perfect statistical results and are true random number generators [11]. Petchlert et al. offered a new coding method to produce random numbers from EEG signals. They focused on true random number generators based on low-cost EEG signals that can be used in applications such as gaming, gambling and complex model simulations. In the verification method, only the least significant digits were taken into account and were transformed into a binary sequence. The binary sequence produced passed the NIST test package with a 99.47 % success rate [12]. Nguyen et al. offered a method that can use EEG signals and wavebands to generate random numbers. In tests, an EEG alcoholism data set was used, and it was shown how to use random number generation as seeds in cryptography and pseudo-random number generation. Numbers were put through the NIST test and the average success rate was shown to be 99.02 % in the gamma band [13].

Chen et al. showed whether an EEG signal can be used as a pseudo-number generator (PRNG) or not. Data used in the suggested method was obtained from both healthy and epileptic EEG signals. It was shown that all EEG signals have different standard deviations over a Gaussian distribution. EEG signals were converted into 5-bit numbers using modular arithmetic. It was shown how these 5-bit numbers were converted into 0s and 1s. The generated numbers passed almost all of the NIST tests with a few exceptions [14].

Tuncer et al. generated random numbers using bioelectrical and physical signals taken from humans. In their studies, they used EEG, electrooculography (EOG) and EMG as bioelectrical signals, and they used blood pressure, respiration and GSR (Galvanic Skin Response) as physical signals. To improve the statistical quality of their signals, they used logistic mapping in the sampling phase, and the exclusive OR (XOR) function in the post-processing phase. They showed that random number generation specific to each individual is possible by examining the statistical properties of the numbers using the NIST test [15].

3. METARIALS

Signals from the human body are characteristic to each individual. Signals that are obtained from DNA, the retina, an EEG and human movements are different for each person. The general definition and data gathering phase of EMG signals in the Ninaweb database, which are used to generate random numbers, are described below.

Muscle activation data are gathered using 12 wireless electrodes inserted on the arm. The placement of the electrodes is shown in Figure 1. Eight of these electrodes are placed at equal distances around the forearms, two electrodes are placed at the main activation points and the last two electrodes are placed at the main activation points of the biceps and triceps.

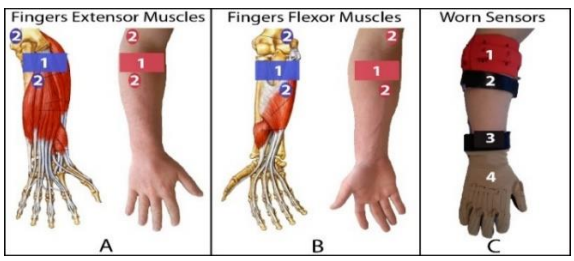

Figure 1. Placement of electrodes [16]

EMG signals were obtained by sampling at a frequency of 2 kHz. EMG signals were taken from 49 different movements over a 5-second recording time from both hands. Forty-nine different movements and the resting position is shown in Figure 2.

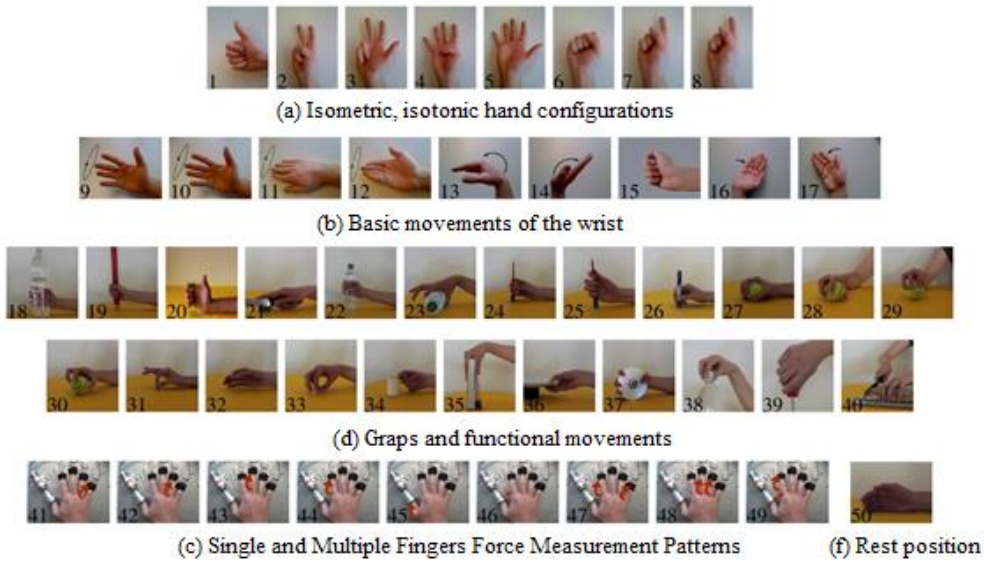

Figure 2. Resting and 49 movement patterns [16]

4. PROPOSED METHODOLOGY

In general, a TRNG structure consists of a physical source, digitization, circuit and post-processing units. Physical sources such as jitter, mouse movements, audio and visual signals, biological and physical signals, and radioactivity decay are used in TRNG structures.

Raw number sequences are produced after signals obtained from physical sources are digitized. But the statistical qualities and randomness of these number sequences are poor. A post-processing phase takes place to eliminate these weaknesses. In the literature reviewed, post-processing algorithms such as XOR, Von-Neumann, linear feedback shift register (LFSR) and Hash Function are very commonly used. In this case, the bit output rate decreased by 50 %.

In this study a TRNG structure is suggested which uses human arm and finger movements as noise sources to obtain the EMG signals. The suggested structure is shown in Figure 3.

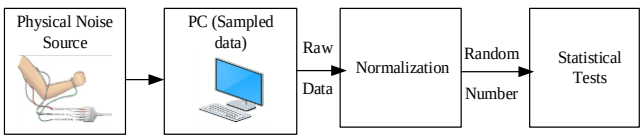

Figure 3. Block diagram of the suggested structure

LabVIEW environment is highly suitable for the online or offline evaluation of physical signals. EMG data taken from the Ninaweb database is converted into the TDMS format to be processed by LabVIEW environment. Data taken with a 2 kHz sampling frequency were subjected to a normalization procedure. Since the statistical qualities of the numbers obtained at the end of the discretization algorithm are good, there is no need for a post-processing procedure, and the bit production rate is 100 %. The statistical qualities of numbers obtained by this system has to be shown, so the NIST, Test U01, Scale index and autocorrelation tests are applied to numbers obtained in this way. The explanation of the TRNG structure summarized above, is given below.

4.1 Obtaining EMG signals

Data obtained from human arm and finger movements with the aid of sensors were used to obtain EMG signals. For finger and hand movements, data in the Ninaweb database, obtained by a 2 kHz sampling frequency, was used. Every sample taken was in the floating point form and every data obtained from a single electrode contained 1,771,800 samples.

4.2 Normalization

Fluctuations in the EMG signal in LabVIEW environment is given in Figure 4. These signals are in continuous time and are represented by the floating point number system. The process in the flow chart of Figure 5 is used to generate a random number in time t.

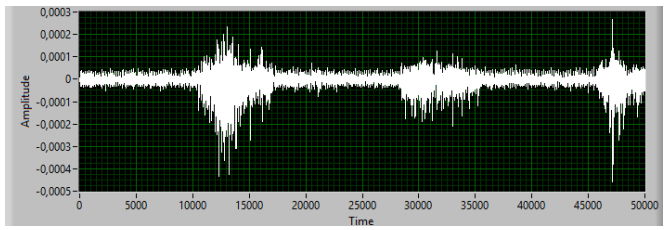

Figure 4. Fluctuations of the EMG signal in continuous time

Step 1: Take the voltage value of any of the EMG signals given in Figure 2.

Step 2: Convert this voltage value to an integer.

Step 3: Convert the continuous time integer value to the binary number system by finding its *t* value in mode 32.

Step 4: Find the number of bits with a value of 1 in the 5-bit number from the binary number system obtained.

Step 5: If the total number of 1s is odd generate a 1, if even, generate a 0 ($n_i=m_i \ mod \ 2$).

Step 6: If sufficient samples have been taken go to Step 7, if not increment *t* to equal $t + 1$ to get a new sample and return to Step 2.

Step 7: End

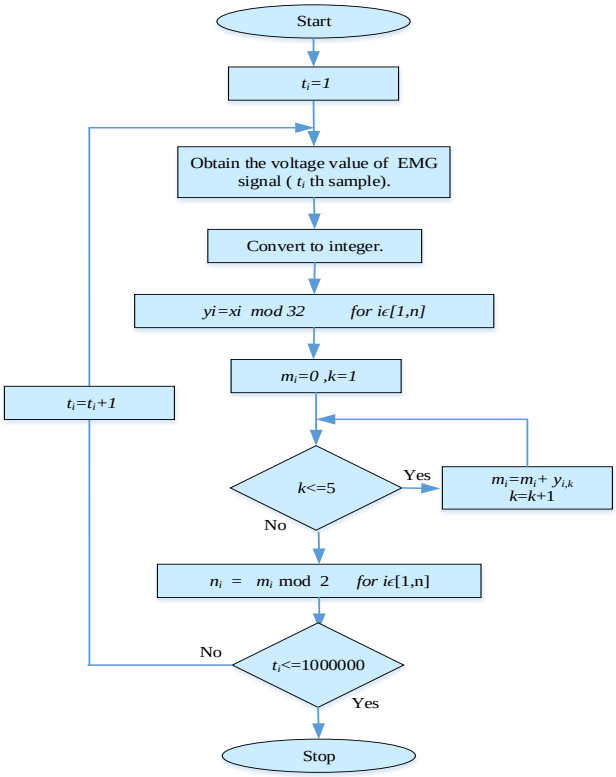

Figure 5. Flow chart of the normalization process

Let the values for *n* samples obtained from EMG signals be $x=(x_1,x_2, \ldots x_n)$. These values are converted into a positive integer by being multiplied by a constant *k*. Together with these converted numbers and equation 1, 5-bit numbers are produced [14].

$$y_i=x_i \bmod 32 \quad for\ i\epsilon[1,n] \qquad (1)$$

The bits which have a value equal to 1 in the produced 5-bit y_i sequence is summed up as (m_i). If the summation is an odd number a 1 is produced; if it is an even number a 0 is produced (n_i).

The discretization algorithm explained above and the number sequence (for *k* =1000) obtained from the EMG signals are shown in Table 1.

Table 1. Discretization process

x_i	0.179	0.084	0.272	0.096	0.107	0.149
$x_i=k*x_i$	179	84	272	96	107	149
y_i	19	20	16	0	11	21
y_i	10011	10100	10000	00000	01011	10101
m_i	3	2	1	0	3	3
n_i	1	0	1	0	1	1

The general view of the program flow chart for random number generation given in Figure 5 above was realized with a sub-program in LabVIEW environment and is given in Figure 6 below. The changes for the middle finger in the random number sequences obtained from the EMG signals are given in Figure 7.

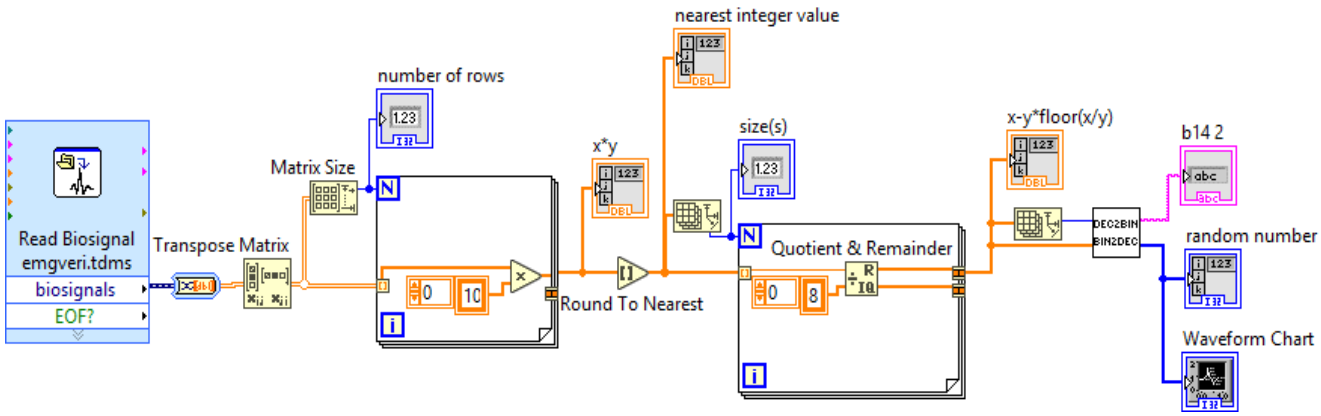

Figure 6. Random number generation in LabVIEW environment

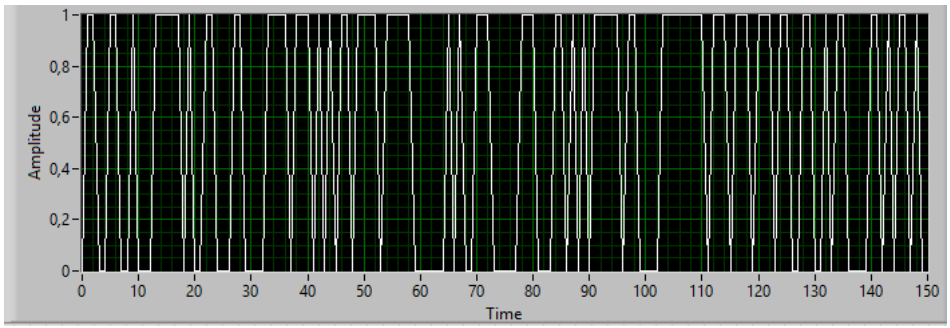

Figure 7. Random number obtained from the EMG signals

To show that the obtained number sequence can be used for cryptology, game theory, secure communication and games of chance, it was subjected to a variety of tests.

5. RESULTS

In literature reviews, tests such as NIST, Diehard and FIPS

are used to show whether 0 to 1 number sequences show randomness or not. Along with this – to show that these number sequences are non-periodic – a Scale-index test was used, and to observe the change of numbers in the number sequences, auto-correlation tests are used. In this study, the results obtained were subjected to NIST, TestU01, Scale index and autocorrelation tests.

5.1 NIST SP 800-22 test

The NIST Test Suite is a statistical package that consists of 15 tests and is produced by hardware- and software-based random number generators to test the randomness of sequences of 0s and 1s. Having a large number of samples in NIST tests (>1,000,000) is appropriate for asymptotic reference distributions. The most important parameter in this test is the p-value (shown in the tables below). Having this value greater than 0.01 shows that the test will be successful [17].

The NIST test results for EMG signals are taken from 12 sensors; the mod 8, 16 and 32 for NIST test results are given in Tables 2, 3 and 4.

Table 2. NIST test result (p-value) for mod 8

NIST test	Channel No											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.742	0.929	0.521	0.810	0.180	0.271	0.515	0.776	0.965	0.179	0.261	0.199
2	-	-	-	-	-	-	0.014	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	0.055	0.198	-	-	-	-
5	0.684	0.242	0.745	0.248	0.182	0.491	0.651	0.793	0.328	0.188	0.153	0.584
6	0.223	0.392	0.717	0.885	0.450	0.098	0.124	0.018	0.433	0.775	0.119	0.651
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	0.018	-	-	-	-	-	-	-	-
9	0.797	0.087	-	0.135	0.210	0.100	0.226	0.162	0.390	0.797	-	-
10	0.476	0.824	0.180	0.337	0.733	0.425	0.289	0.262	0.108	0.353	0.707	0.025
11	0.713/0.112	0.499/ 0.237	-/ 0.330	-/ -	-/ 0.053	-/ 0.019	-/ 0.496	0.015/ 0.295	-/ 0.034	-/ 0.226	-/ -	-/ 0.356
12	-	-	-	-	-	-	0.481	0.326	-	-	-	-
13	0.908	0.618	0.046	0.947	0.320	0.416	0.219	0.922	0.293	0.082	0.252	0.177

Table 3. NIST test results (p-value) for mod 16

NIST test	Channel No											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.859	0.702	0.416	0.864	0.245	0.265	0.162	0.318	0.498	0.119	0.612	0.340
2	0.053	0.064	-	0.211	0.013	-	0.313	0.899	0.081	0.175	-	-
3	-	-	-	-	0.013	-	0.595	0.987	-	0.063	-	-
4	0.338	0.416	0.050	0.730	0.359	0.172	0.679	0.161	0.214	0.035	0.217	0.065
5	0.293	0.909	0.988	0.346	0.023	0.512	0.824	0.461	0.470	0.784	0.533	0.314
6	0.353	0.303	0.277	0.296	0.542	0.621	0.642	0.324	0.504	0.191	0.788	0.959
7	0.060	-	-	0.814	-	0.027	0.265	-	-	0.715	-	0.382
8	0.441	0.707	-	0.465	0.496	0.080	0.093	0.360	0.373	0.526	0.316	0.048
9	0.093	0.370	0.017	0.789	0.907	0.506	0.612	0.079	0.386	0.964	0.137	0.993
10	0.626	0.633	0.960	0.501	0.977	0.903	0.803	0.725	0.979	0.030	0.151	0.167
11	0.499/ 0.237	0.677/ 0.692	0.614/ 0.739	0.237/ 0.708	0.012/ 0.674	0.011/ 0.348	0.634/ 0.984	0.715/ 0.577	0.054/ 0.439	0.157/ 0.414	-/ 0.917	0.187/ 0.949
12	0.882	0.400	0.077	0.729	0.324	0.565	0.587	0.333	0.687	0.094	-	0.339
13	0.569	0.598	0.070	0.812	0.280	0.124	0.147	0.570	0.780	0.032	0.395	0.350

Table 4. NIST test results for (p-value) mod 32

NIST test	Channel No											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.815	0.330	0.031	0.613	0.761	0.016	0.626	0.131	0.465	0.607	0.069	0.463
2	0.378	0.160	0.089	0.539	0.150	0.439	0.396	0.923	0.138	0.364	0.321	0.236
3	0.124	0.024	-	0.174	0.376	0.713	0.490	0.176	0.026	0.959	0.027	0.024
4	0.069	0.012	0.096	0.138	0.716	0.734	0.421	0.319	0.468	0.517	0.935	0.815
5	0.078	0.275	0.999	0.284	0.185	0.215	0.532	0.890	0.021	0.899	0.571	0.698
6	0.532	0.459	0.124	0.772	0.416	0.663	0.523	0.930	0.203	0.933	0.947	0.524
7	0.013	0.066	0.842	0.947	0.124	0.683	0.422	0.861	0.981	0.993	0.789	0.864
8	0.392	0.924	0.548	0.682	0.425	0.298	0.638	0.014	0.155	0.687	0.115	0.808
9	0.542	0.547	-	0.708	0.044	0.914	0.314	0.360	0.488	0.770	0.750	0.920
10	0.599	0.536	0.564	0.924	0.378	0.722	0.744	0.194	0.279	0.420	0.262	0.393
11	0.559/ 0.588	0.939/ 0.824	0.085/ 0.837	0.215/ 0.355	0.809/ 0.611	0.091/ 0.209	0.812/ 0.824	0.177/ 0.867	0.163/ 0.262	0.766/ 0.326	0.931/ 0.925	0.931/ 0.777
12	0.419	0.812	0.117	0.765	0.818	0.133	0.660	0.725	0.166	0.991	0.159	0.679
13	0.601	0.596	0.010	0.800	0.940	0.025	0.223	0.155	0.549	0.628	0.109	0.560

In Tables 2 and 3, even some of the NIST test results for mod 8 and mod 16 are successful, but in general, unsuccessful results are obtained. In Table 4, all of the p values are greater than 0.01 and the third and ninth test of the 3rd channel fails. These results show us that the numbers are random and that numbers obtained from EMG signals can be used in game theory, visual coding and secure communication.

5.2 Autocorrelation test

The autocorrelation test is the measure of fluctuation in 0s and 1s in the random number sequence.

If the |X5| parameter shown in equation 2 below is less than 1.6449, it shows that the autocorrelation test is successful [18, 19].

$$X5 = \frac{2[A(d)-(n-d)/2]}{\sqrt{n-d}} \quad (2)$$

Here (n) shows the length of the generated number sequence. The symbol d is an integer between [1, (n/2)].

The value of (d) is calculated as shown in equation 3:

$$A(d) = \sum_{i=0}^{n-d-1} b_i \oplus b_{i+d} \quad (3)$$

The ($\oplus$) symbol in equation (3) shows the XOR process, (bi) shows the number sequence.

Autocorrelation test results for mod 8, 16 and 32 and for different d values are given in Table 5, 6 and 7, respectively.

Table 5. Autocorrelation test results for mod 8

Channel No	d=4	d=10	d=16	d=25	d=40	d=50	d=125	d=250
1	−1.312	0.679	−1.698	−0.567	0.439	−1.865	−0.248	1.160
2	0.524	−0.041	−1.530	0.536	−0.120	0.373	0.162	−0.107
3	−1.268	−2.770	0.294	−0.615	−2.412	−1.454	−2.519	−0.104
4	−4.098	−1.619	−0.771	1.159	0.037	−0.158	0.498	−0.354
5	0.913	0.578	0.417	−0.716	−0.575	0.079	0.574	0.041
6	−0.547	0.022	0.044	0.482	0.306	0.600	−0.248	−0.895
7	−1.084	1.005	−1.157	−0.909	0.717	−1.204	−1.288	0.113
8	−1.802	−0.211	−0.581	2.041	1.758	−0.041	0.858	−0.537
9	−6.096	−1.362	0.626	−0.577	0.974	1.242	1.124	0.601
10	−0.809	−0.249	1.312	0.482	0	1.736	−0.241	0.474
11	−2.343	−0.622	0.461	−2.110	1.204	1.369	−0.390	−0.547
12	−0.771	0.509	1.391	−0.011	0.505	−1.944	−0.959	0.281

Table 6. Autocorrelation test results for mod 16

Channel No	d=4	d=10	d=16	d=25	d=40	d=50	d=125	d=250
1	0.031	−0.660	−1.464	3.224	0.234	1.419	1.418	0.158
2	−1.116	−0.471	−0.667	1.800	−0.702	−1.695	0.700	0.281
3	−0.442	−4.949	−1.315	−1.633	−1.103	−3.668	−2.092	−4.115
4	1.255	0.215	−0.739	−0.099	−1.948	−0.876	0.188	1.587
5	−0.790	−0.433	−1.068	0.627	−0.917	−1.837	−0.162	3.198
6	0.879	−0.796	1.271	−0.086	0.698	−1.125	−1.152	0.069
7	0.098	0.604	0.841	0.153	−0.094	0.670	1.295	1.679
8	1.302	0.664	1.176	−0.728	−1.350	−0.221	0.545	0.733
9	−1.815	−0.955	−1.941	1.089	0.382	1.034	−2.614	0.911
10	1.220	−2.191	0.246	−0.472	0.464	−0.082	1.197	−0.446
11	−0.341	0.977	−0.483	1.054	0.882	0.164	−1.383	0.775
12	0.382	−0.667	0.252	−0.520	0.306	−1.072	0.906	−0.351

Table 7. Autocorrelation test results for mod 32

Channel No	d=4	d=10	d=16	d=25	d=40	d=50	d=125	d=250
1	−0.660	−1.321	−0.948	−0.441	1.419	−0.983	0.336	−0.300
2	−0.942	0.559	1.245	−0.137	−1.400	0.018	1.491	1.524
3	−6.893	−7.225	−7.962	−6.500	−6.283	−7.703	−7.826	−7.142
4	0.869	0.958	−0.012	−1.149	0.167	0.170	−0.589	0.208
5	0.980	−1.252	0.012	0.105	0.562	−0.483	0.934	−0.389
6	1.138	0.338	1.647	−1.092	−0.189	0.354	−1.431	0.907
7	0.452	−0.047	0.622	0.083	0.025	−0.306	0.491	−0.060
8	−0.012	0.537	−0.724	0.105	−0.015	−1.616	−0.377	0.670
9	−0.351	0.597	−0.167	−1.023	0.164	0.262	−0.716	−0.752
10	−0.028	−0.638	0.898	−2.632	1.141	−0.056	−1.238	0.771
11	−0.553	1.359	−0.373	−0.102	0.395	−0.420	0.311	0.654
12	−0.531	−0.031	−1.359	1.231	−0.177	0.705	0.520	−1.391

Unsuccessful results were obtained in autocorrelation tests for mod 8 as many values for *d* were greater than |1.6449 |as shown in Tables 5, 6 and 7. Autocorrelation test results are relatively more successful for mod 16 as shown in Table 6. The highest success rate is obtained in mod 32 as shown in Table 7. This result shows us that there is no relationship

between the 0s and 1s in the random numbers generated.

5.3 Scale index

The Scale Index test shows the non-periodicity of a time series. In the published literature, the Scale Index test is used for random number generation [20], visual coding [21] and biomedicine [22]. The Scale Index test was devised by Benitez [23]. The Scale Index test gives values between 0 and 1, and being close to 1 means that the degree of non-periodicity of the series is high.

To define the Scale Index, the normalized inner scalogram is defined as in equation 4, sin being the inner scalogram:

$$\bar{S}^{in}(s) = \frac{S^{in}(s)}{(d(s)-c(s))^{\frac{1}{2}}} \quad (4)$$

$J(s) = [c(s), d(s)] \subseteq I$, for all u ϵ j(s), the maximal sub-interval includes I $\psi_{u,s}$ support in I.

Considering that the length of J(s) depends on the s scale, the values of the internal scalogram at different scales cannot be compared. Equation 5 indicates the scale index of *f* in the [s_0, s_1] scale interval. For a detailed proof see [23].

$$i_{scale} := \frac{S(s_{min})}{S(s_{max})} \quad (5)$$

The non-periodicity value of the random numbers generated for mod 8, 16 and 32 from the EMG signals taken from 12 sensors are shown below. Figure 8 shows the change in the Scale Index for each channel.

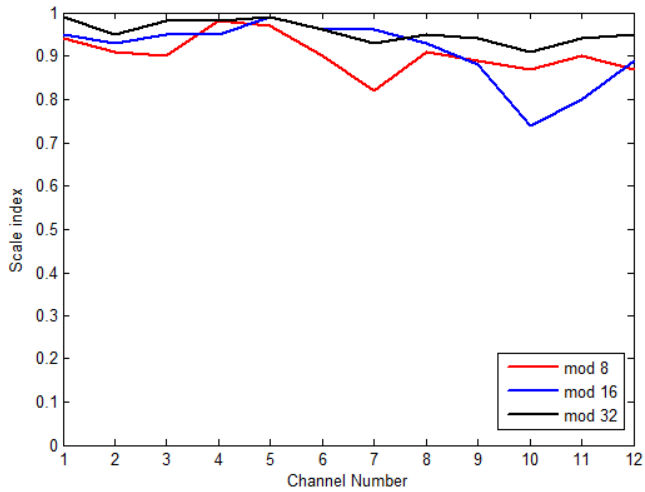

Figure 8. Scale Index change for mod 8,16 and 32

According to figure 8, the degree of non-periodicity is greater than 0.7. The highest values are obtained for mod 32. According to this, all channels have non-periodicity close to 1, so the fluctuations of the numbers are non-periodic.

6. CONCLUSIONS

In this study, a random number generator using EMG signals was described. In the generation of numbers, a modular arithmetically based normalization algorithm was used. Random numbers were produced for the EMG signals for different values of *n* (8,16 and 32); NIST, Scale Index and autocorrelation tests were used to determine the quality of the numbers produced. According to the results obtained, the statistical properties of numbers generated is not sufficient for mod 8 and mod 16. This result shows a true random number was not generated. However, for the EMG signals obtained from 11 channels, the random numbers obtained by using mod 32 were successful and their statistical qualities were good. It was observed that the EMG signals taken only from the third channel were not successful in NIST test runs and in an Overlapping Template Matching test. In general, it was shown that true random numbers can be generated from EMG signals. These numbers can be used in applications such as cryptology, games theory, secure communication and games of chance.

REFERENCES

[1] Akhshani, A., Akhavan A., Mobaraki, A., Lim, S.C., Hassan, Z. (2014). Pseudo random number generator based on quantum chaotic map. Communications in Nonlinear Science and Numerical Simulation, 19(1): 101-111. http://dx.doi.org/10.1016/j.cnsns.2013.06.017

[2] Lynnyk, V., Sakamoto, N., Čelikovský, S. (2015). Pseudo random number generator based on the generalized Lorenz chaotic system. IFAC-PapersOnLine, 48(18): 257-261. http://dx.doi.org/10.1016/j.ifacol.2015.11.046

[3] Kaya, T. (2019). A true random number generator based on a Chua and RO-PUF: Design, implementation and statistical analysis. Analog Integr. Circ. Sig. Process, 1-12. https://doi.org/10.1007/s10470-019-01474-2

[4] Tuncer, T. (2015). Implementation of duplicate trng on fpga by using two different randomness source. Elektronika ir Elektrotechnika, 21(4): 35-39. http://dx.doi.org/10.5755/j01.eee.21.4.12779

[5] Hu, Y., Liao, X.F., Wong, K., Zhou, Q. (2009). A true random number generator based on mouse movement and chaotic cryptography. Chaos. Solitons & Fractals, 40(3): 2286-2293. http://dx.doi.org/10.1016/j.chaos.2007.10.022

[6] Chen, I., Tsai, J., Tzeng J. (2011). Audio random number generator and its application. 2011 International Conference on Machine Learning and Cybernetics, Guilin, pp. 1678-1683. http://dx.doi.org/10.1109/ICMLC.2011.6017002

[7] Chen, I.T. (2013). Random numbers generated from audio and video sources, s.l. Math. Prob. Eng, 2013: 1-7. http://dx.doi.org/10.1155/2013/285373

[8] Iba, Y., Tanaka-Yamawaki, M. (1996). A statistical analysis of human random number generators. In: Yamakawa, T., Matsumoto, G. (eds) Proceedings of the 4th International Conference on Soft Computing (IIZUKA'96), Iizuka, Fukuoka, Japan, Sep. 30–Oct. 5, 1996, World Scientific, Singapore, New Jersey, London, Hong Kong, pp. 467-472.

[9] Schulz, M.A., Schmalbach, B., Brugger, P., Witt, K. (2012). Analysing humanly generated random number sequences: A pattern-based approach. PLoS One, 7(7): e41531. http://dx.doi.org/10.1371/journal.pone.0041531

[10] Jokar, E., Mikaili, M. (2012). Assessment of human random number generation for biometric verification. Journal of Medical Signals and Sensors, 2(2): 82-87. http://dx.doi.org/10.4103/2228-7477.110403

[11] Szczepanski, J., Wajnryb, E., Amigó, J.M., Sanchez-

Vives, M.V., Slater, M. (2004). Biometric random number generators. Comput. Secur, 23: 77-84. http://dx.doi.org/10.1016/S0167-4048(04)00064-1

[12] Petchlert, B., Hasegawa, H. (2014). Using a Low-Cost electroencephalogram (EEG) directly as random number generator. IIAI 3rd International Conference on Advanced Applied Informatics, Kitakyushu, pp. 470-474. http://dx.doi.org/10.1109/IIAI-AAI.2014.100

[13] Nguyen, D., Tran, D., Ma, W., Nguyen, K. (2017). EEG-Based random number generators. International Conference on Network and System Security, pp. 248-256. http://dx.doi.org/10.1007/978-3-319-64701-2_18

[14] Chen, G. (2014). Are electroencephalogram (EEG) signals pseudo-random number generators. Journal of Computational and Applied Mathematics, 268: 1-4. http://dx.doi.org/10.1016/j.cam.2014.02.028

[15] Arslan Tuncer, S., Kaya, T. (2018). True random number generation from bioelectrical and physical signals. Computational and Mathematical Methods in Medicine, 2018: 1-11. http://dx.doi.org/10.1155/2018/3579275

[16] http://ninaweb.hevs.ch/, accessed on 24 Sep., 2018.

[17] NIST Special Publication 800-22. (2001). https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software, accessed on 28 Sep., 2018.

[18] Chan, J.J., Thulasiraman P., Thomas, G., Thulasiram, R. (2016). Ensuring quality of random numbers from TRNG: design and evaluation of post-processing using genetic algorithm. Journal of Computer and Communications, 4(4): 73-92. http://dx.doi.org/10.4236/jcc.2016.44007

[19] Chen, X.M., Wang, L., Li, B.X., Wang, Y. (2016). Modeling random telegraph noise as a randomness source and its application in true random number generation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(9): 1435-1448. http://dx.doi.org/10.1109/TCAD.2015.2511074

[20] Yang, Y., Zhao, Q. (2016). Novel pseudo-random number generator based on quantum random walks. Scientific Reports, 6(1): 20362. http://dx.doi.org/10.1038/srep20362

[21] Yang ,Y.G., Pan, Q.X., Sun, S.J., Xu, P. (2015). Novel image encryption based on quantum walks. Sci Report 5. http://dx.doi.org/10.1038/srep07784

[22] Behnia, S., Ziaei, J., Ghiassi, M., Yahyavi, M. (2013). Comprehensive chaotic description of heartbeat dynamics using scale index and Lyapunov exponent. Proceedings 6th Chaotic Modeling and Simulation International Conference 11-14 June 2013 Istanbul, Turkey.

[23] Benìtez, R., Bolos, V.J., Ramìrez, M. E. (2010). A wavelet-based tool for studying non-periodicity. Computers & Mathematics with Applications. An International Journal, 60(3): 634-641. http://dx.doi.org/10.1016/j.camwa.2010.05.010