Computer Workstation Vetting by Supply Current Monitoring

Marko Dimitrijević, Miona Andrejević Stošović, Octavio Nieto, Slobodan Bojanić, and Vančo Litovski

Abstract – It is our goal within this project to develop a powerful electronic system capable to claim, with high certainty, that a malicious software is running (or not) along with the workstations' normal activity. The new product will be based on measurement of the supply current taken by a workstation from the grid. Unique technique is proposed within these proceedings that analyses the supply current to produce information about the state of the workstation and to generate information of the presence of malicious software running along with the rightful applications. The testing is based on comparison of the behavior of a fault-free workstation (established in advance) and the behavior of the potentially faulty device.

Keywords – monitoring, malicious software, supply current.

I. INTRODUCTION

These proceedings are based on advanced analysis of power supply current to the Device Under Test (DUT) with the aim of detecting malicious activity. The method stems from the long term research in the fields of electronic design, testing, diagnosis, statistical analysis, and artificial neural networks application within the Laboratory for Electronic Circuit Design Automation at the University of Nis.

TABLE I

MEASURED CONSUMPTION OF A PC

State	V (RMS)	I (RMS)	TPF (%)	THDI (%)	P (W)	QB (W)	U (W)	D (W)
Hibernation	217.5	0.090	5.29	18.4	1.04	-19.3	19.6	3.3
Standby	218.7	0.093	12.4	32.2	2.53	-19.3	20.4	6.0
Idle	217.8	0.339	88.5	16.4	65.4	-31.9	73.9	12.6
High load (Video)	218.0	0.348	89.1	16.2	67.7	-32.3	76.0	12.1
High load (Simulation)	217.6	0.537	95.0	13.0	111.	-32.1	117.	18.3

As part of the research task to characterize the personal computer as an energy consumer [1], Table 1. was reported.

M. Dimitrijević and M. Andrejević-Stošović are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia E-mail: {marko, miona}@elfak.ni.ac.rs.

O. Nieto and S. Bojanić are with Technical University of Madrid, {nieto, sbojanic}@

V. Litovski is with Cluster of Advanced Tecnologies NiCAT Niš, vanco@elfak.ni.ac.rs.

The above measurements were obtained for a desktop computer (DELL Optiplex 980, Intel Core i7 CPU @ 2.8GHz, 4GB RAM, 500GB HDD). Characterization variables above are as follows: the RMS value of the line voltage (V), the RMS value of the line current (I), the total power factor (TPF), the total harmonic distortion of the current (THDI), the active power (P), the reactive power (QB), the apparent power (U), and the distortion power (D). Several states of activity were considered: Hibernation, Standby, Idle, High load (Video), and High load (Simulation).

Table 1 demonstrates that the internal activity of the computer can be deduced from power characterization. Particularly interesting aspect is harmonic distortion here represented by the THDI. Namely, the computer, like most electronic loads to the grid, is nonlinear. That means it distorts the grid current from its original sinusoidal waveform i.e. creates harmonic being measurable from the grid side. The current waveform, as may be partly deduced from Table 1, depends on the activities within the computer and so are the harmonics. That suggests that by measurement of the current waveform one may try to identify malicious activities.

Similar methods are applied in IDDQ testing and diagnosis of CMOS electronics [2]. Here, every event in a CMOS digital system introduces a very short pulse into the DC supply current of the circuit. These pulses aggregate to form the DC current (IDD), imprinting information about all activities in the system to the supply current.

Findings in [1] and [2], suggest a correspondence between the activity within the device and the measured cha racteristics of the supply current. Our proposal is to develop a technique for utilizing this correspondence to perform comparisons between a vetted device, which will be stated in the next as the DUT, and a standard fault free device.

The analysis process can be described as a sequence of following steps:

1. Measurement
2. Creation of time series "strings"
3. Spectrum computation
4. Comparisons – similarity evaluation
5. Proposing a hypothesis

The following sections describe the proposed system in more detail.

II. THE MEASUREMENT SUBSYSTEM

The physical structure of the Vetting by Supply Current Monitoring (VbCM) system we are proposing here is depicted in Fig. 1. It is connected to the power grid via the acquisition module (AC), and transfers power directly to the DUT (load) while sampling the values of the current and voltage waveforms passing through. The sampled values are appropriately conditioned and coded, and then directly delivered to the testing computer (TC) via USB or Wi-Fi connection. The software implemented in the TC performs all computations. TC is used as a visualization device, enabling display of the measured and derived waveforms; as an interactive monitor allowing monitoring and control of the chara-terization process; as a data storage device creating measurement logs and databases; and as a communication means enabling remote control of the measurement and on-line delivery of the results.

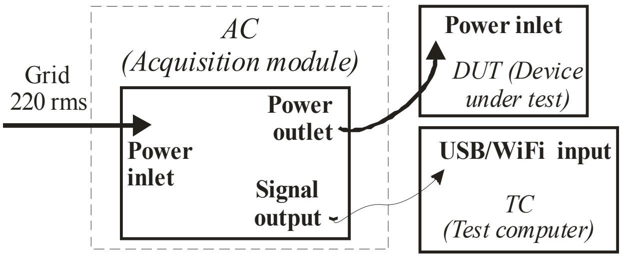

Fig 1. Physical structure of the VbCM system (The 220 Vrms is related to European standards)

The acquisition module is performing acquisition and conditioning of the electrical signals. The module for signal conditioning of the voltage and current waveforms provide attenuation, isolation, and antialiasing.

In the present three-phase application the acquisition is performed by National Instruments cDAQ-9714 expansion chassis, providing hot-plug module connectivity. The chassis is equipped with two data acquisition modules: NI9225 and NI9227. Extension chassis is connected to TC running virtual instrument via USB interface. NI9225 has three channels of simultaneously sampled voltage inputs with 24-bit accuracy, 50 kSa/s per channel sampling rate, and 600 VRMS channel-to-earth isolation, suitable for voltage measurements up to 100th harmonic (5 kHz). The 300 VRMS range enables line-to-neutral measurements of 110V or 240 V power grids. NI9227 is four channels input module with 24-bit accuracy, 50 kSa/s per channel sampling rate, designed to measure 5 ARMS nominal and up to 14 A peak on each channel with 250 VRMS channel-to-channel isolation. The virtual instrument is realized in National Instruments LabVIEW developing package which provides simple creation of virtual instruments. Virtual instruments consist of interface to acquisition module and application with graphic user interface.

III. CREATION OF TIME SERIES

In this stage, the signal obtained after testing is converted to time series. There are three scenarios for obtaining samples:

- Boot sequence
- Idle state
- Application execution

During the measurement phase of the test, time series is obtained for selected device activity as above. For example, in Table 1. five states are established among which three are with no application running while the last two excessively load the processor and the video card. The choice of a number of test runs and application is subject to further analysis.

Further research task is establishing the length of data strings and how many strings will be required for every state of the DUT. It is our experience that for getting the spectrum of the current by the Goertzel algorithm, 200 ms (for the 50 Hz case) are sufficient. The strings measured for the given set of states of the fault-free device are sufficient for its characterization since stationary conditions are established and no change in time may be expected. However, length of testing of the potentially 'faulty' device is also subject to further research.

IV. Comparisons - similarity evaluation

From our experience in time series prediction [3,4], classification [5] and diagnosis [6,7,8], the subject of comparison of the responses of the fault-free and the potentially faulty DUT will be an important research issue within this project.

According to [9], there are several measures of similarity of time series that may be used concurrently. For example, the correlation coefficient may be calculated as:

$$r_{\mathrm{pq}} = \frac{\sum_{k=0}^{N-1}\left(I_k^{\mathrm{p}} - \bar{I}^{\mathrm{p}}\right)\left(I_k^{\mathrm{q}} - \bar{I}^{\mathrm{q}}\right)}{\sqrt{\sum_{k=0}^{N-1}\left(I_k^{\mathrm{p}} - \bar{I}^{\mathrm{p}}\right)^2}\sqrt{\sum_{k=0}^{N-1}\left(I_k^{\mathrm{q}} - \bar{I}^{\mathrm{q}}\right)^2}} \quad (1)$$

where Ip is the first series, Iq is the second series, and , are mean values, and N is the number of samples. To demonstrate we made new measurements, similar to the ones depicted in Table 1, some results of which will be presented here. The DELL Optiplex 980 run under Windows 7 Professional was considered in the following states: 1. Off : meaning the computer was switched off; 2. Idle: Only the operating system is running (65 processes and 975 threads were active during the measurement while 0% of the CPU was utilized); 3. Video: 4 MPEG4 video streams were activated simultaneously while 4%-8% of the processor was loaded; 4. CPU Arithmetic: Synthetic DhrystoneiSSE4.2 and Wetstone iSSE3 Benchmark test were activated with 100% of the CPU loading; 5. Multi-Media CPU: Synthetic Multi-MediaInt x16 iSSE4.4, Float X8 iSSE2 iDouble x4 iSSE2 were activated with 100% of the CPU loading; 6. GPU Rendering: renderbenchmark test were activated to test the Graphic processor: NativeFloatShaders, EmulatedDouble-Shaders; 7. Physical Disks: test for evaluation of the disc performances were activated: Physical disk benchmark WDC5000AAKS-007AA0; 8. File System Benchmark: Performances test for the I/O file system was activated.

TABLE II

CORRELATION COEFFICIENTS FOR THE SIGNALS OBTAINED FOR DIFFERENT STATES OF THE WORKSTATION

r_{pq} p→ q↓	Off	Idle	Video	CPU Arithmetic	GPU Rendering	Multi-Media CPU	Physical Disks	File System Benchmark
Off	1							
Idle	-0,92574	1						
Video	0,287788	-0,01358	1					
CPU Arithmetic	-0,96037	0,857545	-0,49706	1				
GPU Rendering	-0,90384	0,986579	0,079612	0,811307	1			
Multi-Media CPU	0,867432	-0,97951	-0,13936	-0,7647	-0,99222	1		
Physical Disks	-0,60827	0,798615	0,528475	0,422196	0,862064	-0,88421	1	
File System Benchmark	0,785238	-0,59354	0,771507	-0,91172	-0,52713	0,458711	-0,05733	1

Strings of 10,000 samples (200 ms or 10 periods), were taken at the rate of 20 µs. After implementation of (1) the correlation matrix given in Table 2 was obtained. While these results may be used for some further analysis (for example, high correlation coefficient means that the same resources were (equally) active during the run of two particular software packages.) they are introduced here only to demonstrate what is expected to be done with the measurement results when comparing the behavior of the fault-free and the DUT.

In the tables that will be used for testing, while the rows indicating the states will be kept as above, the columns will be related to the strings displaced in time obtained by repetitive measurements of the same state. Accordingly, the table will be a matrix albeit not a symmetric one. Search will be performed for the cases with smallest correlation coefficients, which indicate difference in the behavior of the DUT and fault-free device.

V. COMPARISONS - SPECTRUM COMPUTATION

Additionally, analysis will be performed in spectral domain, for each time series string. The transformation almost entirely preserves the information content of the string while reducing the data size. To do that, because of the instability of the frequency of the grid, the frequency is to be extracted first [10]. Then, an algorithm is to be implemented for Discrete Fourier Transform (DFT) that is resistant to the instability of the period of the signal. The Goertzel algorithm [11] will then be applied.

Approximately 50 harmonics may be observed in a sample (string) of a grid current. We expect approximately 10 strings per device to be created for every state of the device, one of them stored for the fault-free device, to be used as a base for comparisons.

For the measurement of a fault-free device analyzed in Table 2 appropriate DFT was performed. The spectral components obtained are presented in Table 3. Since even harmonics have incomparably smaller values than the odd ones, in Table 3 only the DC, the main, and the odd harmonics are presented. Figure 2 illustrates two columns of Table 3.

While these results may be used for some further analysis we are delivering them here as a kind of proof-of-concept since not the same value of any harmonic may be found while similarities may be extracted.

VI. CREATING THE MOST PROBABLE HYPOTHESIS

The testing will be implemented in the following way.

First, for the fault free device, one string of the supply current will be created after steady state for every test program. We will denote the strings here as Si, i=1,2,..., n. n is the number of software packages developed for testing purposes. At the very moment we are not sure as to what value of n will be necessary.

Then, for every installed malicious software, after equilibrium, m strings of supply current will be taken, se-parated by a fixed time interval, for all n testing soft-ware packages. We will denote these strings by $Q_{i,j}$, i= 1,2,...,n, and j=1,2,..., m.

Accordingly, there will be nHm strings generated from the DUT for every malicious attack.

TABLE III

ODD HARMONICS EXTRACTED FROM ONE STRING MEASUREMENT IN EIGHT DIFFERENT STATES OF THE WORKSTATION

Harm. No.	DC	1	3	5	7	9	11	13	15	17	19	21	23
Off	-0.55	89.7	3.05	8.55	8.94	3.08	8.76	2.77	6.28	4.81	0.69	0.92	0.62
Idle	-0.84	400.26	47.9	23.18	11.41	9.19	6.17	1.4	9.81	3.66	4.16	7.39	5.17
Video	1.3	475.4	54.03	23.52	12.3	7.7	7.24	1.73	12.19	5.1	5.05	6.52	7.15
CPU Arithmetic	-0.52	785.73	34.6	28.7	17.43	10.12	12.27	6.01	5.98	8.91	5.74	4.89	6.06
GPU Rendering	-0.68	747.73	35.84	28.42	16.77	9.26	11.13	5.81	6.84	9.9	5.68	5.12	7.19
Multi-Media CPU	-1.3	394.33	47.79	22.83	9.74	9.17	6.12	1.99	9.32	5.6	3.3	6.65	5.55
Physical Disks	-0.23	381.54	48.05	23.53	6.96	8.63	5.36	2.49	9.94	3.76	5.75	5.55	4.56
File System Benchmark	-0.51	411.72	47.73	24.14	9.61	9.5	5.53	2.96	8.92	3.71	7.31	5.29	4.3

Harm. No.	25	27	29	31	33	35	37	39	41	43	45	47	49
Off	0.53	0.94	0.62	0.54	1.08	0.47	0.45	0.58	0.54	0.24	0.27	0.39	0.21
Idle	4.12	5.18	6.61	4.89	7.58	3.98	2.61	3.9	1.29	1.28	1.91	0.94	0.36
Video	6.2	8.31	6.35	3.64	5.23	2.72	2.09	2.83	0.97	0.46	0.85	0.98	0.53
CPU Arithmetic	5.86	2.29	2.94	2.54	4.48	1.71	0.51	2.94	1.26	1.24	1.44	0.34	1.95
GPU Rendering	4.63	1.28	4.3	3.61	3.67	1.59	0.93	3.55	0.56	0.67	1.79	0.48	1.78
Multi-Media CPU	4.6	4.2	5.85	4.98	7.84	4.27	2.98	3.97	1.54	1.39	2.2	0.55	0.7
Physical Disks	5.2	3.07	4.93	3.96	8.2	4.17	3.19	4.7	0.96	1.24	1.93	0.9	1.34
File System Benchmark	4.76	6.35	6.26	5.16	7.34	2.94	2.2	2.81	1.11	1.82	1.77	1.03	0.95

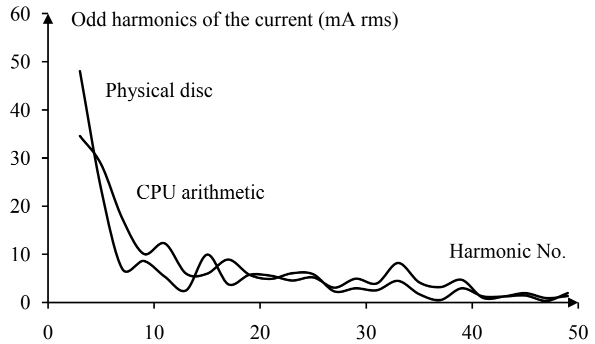

Fig 2. Measured odd harmonics in two cases: Physical disc drive active and CPU loaded by arithmetic computations. The first harmonic is omitted for convenience

Comparisons will be done within the i-th set $Q_{i,j}$, j=1,2,..., m in order to find whether a change happens in time e.g. if there exists $Q_{i,k} \neq Q_{i,l}$. Here $\neq$ means not similar, while k,l **J**={1,2,..,m}. If yes, a probability exist for the malicious attack was activated by the i-th testing software. The chosen $Q_{i,j}$ will be compared with S_i to get final decision. If similar, i will be incremented by one and the procedure will be repeated.

If none of the strings goes below the chosen similarity measure the conclusion will be that no malicious software is running.

In the next, here we will demonstrate that there are some potentially feasible procedures to measure similarity. What we are presenting here is not a solution but a hint of why we will use these mathematical tools.

Suppose the correlation matrix is used as a basis for decision about the similarity between the responses of the fault-free and the faulty device. Since at least ten strings per state will be produced a correlation coefficient will be calculated for every string of the fault-free and the DUT for a given state. That is important since no time synchronization is preserved for the two measurements.

It is obvious that a proper search is to be done to find: (1) the most distant (less correlated) strings within a state and (2) the most distant strings of all. That would become the most probable hypothesis if the correlation coefficient is used as the basis for establishing the non-similarity of the behavior of the faulty and fault-free device. A threshold is to be foreseen in order to enable conclusion as to whether the extracted minimal similarity may be pronounced a proof for the presence of malicious activity within the device. In that way the testing process may be terminated by a go-no-go statement.

TABLE VI

RESPONSES OF THE ANN TO NOISY INPUT DATA

ANN's Output→ Input vector↓	Off	Idle	Video	CPU Arithmetic	GPU Rendering	Multi-Media CPU	Physical Disks	File System Benchmark
-1	0.94189	-0.0082643	-4.98E-05	0.0596502	0.0054563	-2.69E-05	0.0025452	0.0012835
	1	0	0	0	0	0	0	0
-2	-0.100789	0.936809	-6.30E-05	0.107029	-0.0039056	-4.57E-05	0.0353001	0.0301201
	0	1	0	0	0	0	0	0
-3	0.0747284	-0.0347075	1.00742	-0.0946782	0.0368009	6.60E-06	0.0172143	-0.0095049
	0	0	1	0	0	0	0	0
-4	0.0530374	-0.0051336	-3.01E-05	0.94394	0.00599	4.07E-06	-0.0031459	0.0039932
	0	0	0	1	0	0	0	0
-5	-0.0714551	0.141341	0.0002496	0.347383	0.694706	2.93E-05	-0.0165517	-0.0935344
	0	0	0	0	1	0	0	0
-6	-0.0390391	-0.068559	-2.64E-05	0.0464038	-0.0182126	0.994595	0.0357881	0.0513166
	0	0	0	0	0	1	0	0
-7	0.0221675	-0.0245939	-7.76E-06	-0.0287134	0.0235965	-8.00E-07	1.01758	-0.010466
	0	0	0	0	0	0	1	0
-8	0.0524894	-0.0178626	-6.26E-05	-0.0587603	0.0177179	1.40E-06	0.0010393	1.00386
	0	0	0	0	0	0	0	1

Note, the information about the state that produces minimal similarity, in some way, may be used as a diagnostic information since it identifies the state of the device where the malicious activity happens.

Other statistical measures of similarity will be not excluded from analysis

On the other side, if the harmonic spectrum is to be used for extracting the nonsimilarity measure, a proper method of hypothesis generation will be created.

For example, an ANN was trained to create a response recognizing which one of the sets of harmonics of Table 3 is present at its input. Its structure is depicted in Fig. 3. To simplify, for the proper vector of harmonics, the corresponding output of the ANN was forced to unity while the rest of the outputs were kept at zero. In other words, it was trained to recognize which software was running within the computer. Full success was achieved meaning, after training, the ANN was classifying perfectly.

To make the problem harder, we transformed Table 3 so that every entry was recalculated by the formula

$$x_{\text{new}} = x \cdot [1 + (2 \cdot rnd - 1) \cdot 0.025] \quad (3)$$

where *rnd* is a pseudo-random number with uniform distribution within the [0,1] segment. In other words a "noise" of amplitude (peak-to-peak) as large as 5% of the harmonic value was added as "measurement disturbance". Again, as can be seen from Table 4, excellent classification was obtained.

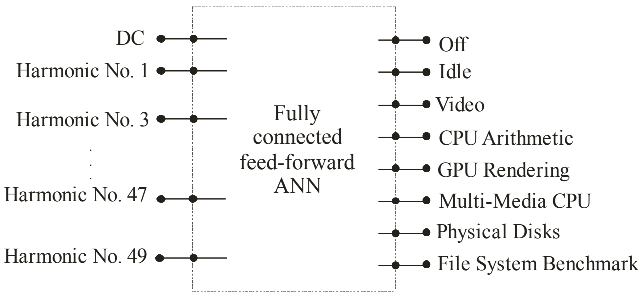

Fig 3. Artificial neural network that eavesdrops the personal computer based on information on harmonics in its mains current

Finally, eight new sets of "harmonics" were created artificially by permutations within the rows in Table 3 and the newly created columns were used as excitation to the ANN. None succeeded to deceive the ANN.

At the very moment we are not aware as to which of the concepts will be the best to be applied for detecting malicious activities in a computer. There is a probability for several of them to be used simultaneously i.e to apply Multiple Criteria Assessment of Discrete Alternatives (MCDA) [12]. That would lead to a creation of an integral measure of nonsimilarity while giving weight to different outcomes from different approaches (correlation of time series, correlation of harmonics, pattern recognition by ANNs, etc).

VII. Description of the system

The analysis system consists of the measurement subsystem and the software subsystem.

Given a fault-free device, measurements are performed to produce strings of supply current. There will be several states analyzed. The resulting data will be stored on the TC.

For testing a DUT, same measurements are repeated several times. Obtained time series strings are used by the software subsystem to reach a decision.The software within TC will enable interactive user-friendly interface with the test engineer. It will also allow for logging, documenting, reporting, storing, and post-processing of the resulting data.

VIII. Unique properties of the solution

There are several aspects of our solution that we believe are novel and unique.

1. It is noninvasive. No change whatsoever happens in the software and hardware of the device.
2. The testing does not interfere with the regular activities of the device and therefore testing could potentially be performed on an active production device (e.g. network router).
3. The measurements, the measurement results, and the processing are taking place outside of the device so no tempering with the test from within the device is possible.
4. No library (or lists) of malware is to be produced and updated.
5. The simplicity of the concept will allow for improvements that are not conceivable at the moment.
6. The solution is entirely independent of device classes, CPU architectures, operating systems etc.

While the proposed solution consists of activities that are known in the testing practice of electronic systems, the uniqueness comes from the fact that these ideas were never utilized and tuned for detection of malicious activities in electronic devices.

References

[1] Nieto, O., et all., “Energy Profile of a Personal Computer”, Proceedings of the LVI Conf. of ETRAN, Zlatibor, Serbia, June 2012, ISBN 978-86-80509-67-9, Paper EL3.3-1-4.

[2] Hurst, S. L., “VLSI testing: digital and mixed analogue/digital techniques”, IEE, London, 1998, ISBN 0852969015.

[3] Milojković, J., and Litovski, V., “Dynamic Short-Term Forecasting of Electricity Load Using Feed-Forward ANNs”, Int. J. of Engineering Intelligent Systems for Electrical Engineering and Communications, 2009, Vol. 17, No. 1, pp. 39-48.

[4] Milojković, J., and Litovski, V., “Short Term Forecasting in Electronics”, Int. J. of Electronics, Vol. 98, No. 2, 2011, pp. 161-172.

[5] Milenković, S., Obradović, Z., and Litovski, V., “Annealing Based Dynamic Learning in Second-Order Neural Networks”, Int. Conf. on Neural Networks, ICNN '96, Washington, D.C., USA, 3.-6. June 1996, pp. 458-463.

[6] Litovski V., Andrejević M., and Zwolinski M., “Analogue Electronic Circuit Diagnosis Based on ANNs”, Microelectronics Reliability, Vol. 46, No. 8, August 2006, pp. 1382-1391.

[7] Andrejević-Stošović, M., Milovanović, D., and Litovski, V., “Hierarchical Approach to Diagnosis of Mixed-mode Circuits Using Artificial Neural Networks”, Neural Network World, Vol. 21, No. 2, 2011, pp. 153-168.

[8] Sokolović, M., Litovski, V., Zwolinski, M., “New Concepts of Worst Case Delay and Yield Estimation in Asynchronous VLSI Circuits”, Microelectronics Reliability, 2009, Vol. 49, No. 2, pp. 186-198.

[9] Lhermitte, S., et all., “A comparison of time series similarity measures for classification and change detection of ecosystem dynamics”, Remote Sensing of Environment, Vol. 115, No. 12, 15 December 2011, pp. 3129–3152.

[10] Terzija, V., Stanojević, V.: “STLS Algorithm for Power-Quality Indices Estimation”, IEEE Transactions on Power Delivery, April 2008, Vol. 24, No. 2, pp. 544-552.

[11] Goertzel, G., “An Algorithm for the Evaluation of Finite Trigonometric Series”, The American Mathematical Monthly, January 1958, No. 1, Vol. 65, pp 34-35.

[12] Makowski, M., and Granat, J., “ Multicriteria Analysis of Large Sets of Alternatives “, 21st CSM Workshop, IIASA, Laxenburg, Austria, August 27–29, 2007