



TIME ANALYSIS OF VIRTUAL SPECTRUM ANALYZER

Wieslaw Winięcki, Piotr Bilski

Institute of Radioelectronics, Warsaw University of Technology,
Nowowiejska 15/19, Warsaw, Poland, w.winięcki@ire.pw.edu.pl, foxtrot@sgk.com.pl

Abstract: *The problem of the Real-Time virtual instrument is presented. The requirements for the Real-Time conditions are explained. The method of the instrument time analysis is proposed. The virtual spectrum analyser is examined in order to reveal properties of the software functions. The optimisation procedure is described and its results are presented. The conclusions for the future developers have been articulated.*

Keywords: *Virtual Instrument, time analysis, Real-Time mode, software optimization*

1. INTRODUCTION

Virtual measurement devices are currently a very numerous and quickly progressing group of instruments. Their applications slowly reach beyond the basic educational projects, heading towards unexploited areas. They become helpful not only to the metrologists, but also to the designers not directly connected to science. Vast possibilities result in complex constructions [5], [6]. Speed of work and accuracy are their main features. Both are determined by the hardware and software at the same time. The former is main infringement of instrument's abilities. The accuracy problems have been considered [1], [2], but the time analysis is not yet introduced. Instrument consisting of a computer, data acquisition (DAQ) card and specialized program for signal processing depends on all those factors. As the computer technologies become more sophisticated [9], the software optimization may be necessary to acquire expected performance of the virtual instrument (VI). It is especially true for the group of virtual devices working under the Real-Time conditions. The novelty of the approach is the idea of designing VI able to work under the Real-Time conditions. As the problem was not considered so far, there are no methods ready to use. The proposed solution links measurement concepts with the statistical methodology of the instrument verification. The paper presents results of the research on the dependencies connecting hardware and software parts.

2. PROBLEM DESCRIPTION

Virtual instrument's work regime can be decomposed into few constituent phases. The first one is the acquisition of the measurement data. The samples are obtained by the DAQ card and transported to the computer's memory. The second stage is the main goal of research. It comprises signal-processing part, according to the designer's assumptions. The data is processed only in digital form. The last phase is connected to the presenting results on the front panel. The dependence is as following:

$$t_c = t_{acq} + t_{proc} + t_{vis} \quad (1)$$

where t_c is a complete time of the instrument's work (from obtaining signal samples to presenting results on screen), t_{acq} is the time of signal acquisition from external source, t_{proc} is a processing time (including mathematical operations) and t_{vis} is the results presenting time. It is worth noticing that according to the assumptions of virtual instrumentation structure the developer can optimize mainly t_{proc} and can be easily modified or changed. The other elements are either hardware based and therefore fixed, or their execution time remains irrelevant. Because t_c depends on both hardware and software features, the simplest, but also expensive method for increasing effectiveness is to deploy it on the faster computer configuration. In the paper the authors focused on the optimization of the software part, which is the easiest to accelerate.

From the Real-Time instrument point of view the features of the software part itself are important. The acquisition time has to be longer than the processing time as both operations are conducted simultaneously. When the condition is met:

$$t_{acq} \geq t_{proc} + t_{vis} \quad (2)$$

there are no delays imposed by the waiting for new samples to come. As the t_{acq} is constant, it imposes main infringement on the VI's speed of work. Even if the software operations are faster than t_{acq} , the instrument will not any faster than t_{acq} .

To define dependencies between software parts, numerous measurements have to be conducted. Empirical approach is the only solution for assumptions and hypotheses verification. Measurements of the operations' time consumption in the virtual instrument are difficult. The greatest problem is the short duration of the operations. It makes direct measurements using standard tools of the programming environments impossible. Such timers measure time with the millisecond resolution, while many functions are conducted in much shorter time. In the presented problem it is convenient to resort to the arithmetical averaging method. Processing operations are then repeated many times in a simple loop. The time measured covers all the iterations and the desired result is obtained by the division of that acquired time by the number of iterations. A simple statistical mechanism is used here [11], with the average described by the formula:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i \quad (3)$$

where t_i is the time of a single execution of the operation, N is the number of iterations and \bar{t} is the single operation execution time.

The formula for a standard deviation is also used:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (t_i - \bar{t})^2} \quad (4)$$

with the same signatures.

The statistical method can be modified (for example, by deleting certain amount of the extreme values), however the presented approach appears to be quite satisfying.

Because of the features and complexity of every operating system for the typical computer, the time of a single operation execution is not constant. It depends not only on an absolute computational cost (as defined in [4]), but also on the other processes

present in the system. They also require attendance during the virtual instrument's work time. Therefore time measurement of the same operation always gives different results. It is presented in Fig. 1.

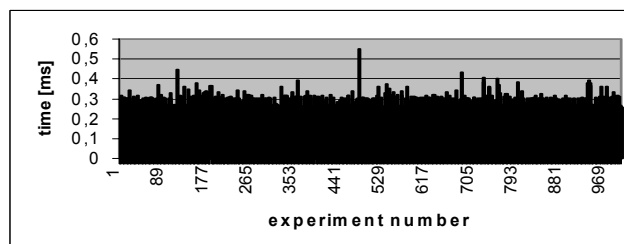


Fig. 1 - Exemplary illustration of the experiments.

A single execution of the experiment is not enough for the research. It has to be repeated many times for the subsequent statistical analysis.

3. PROJECT ASSUMPTIONS

A very important issue is the choice of the hardware and software platform for the experiments. As the base computer there was chosen a PC model equipped with AMD Duron 750 MHz processor, having 256 MB RAM memory, ECS K7S5A mainboard, GeForce 2 MX graphic card (32 MB) and Seagate 17 GB hard drive. The installed operating system was Windows 98 as one of the most popular among professional and home users. As the programming environment was chosen LabVIEW (version 6i) from National Instruments [7]. It is the most efficient and popular tool for the virtual instruments design, setting standards in the art of the specialized programming domain. To prepare the measurements a simple program for the time analysis was written. Its task is the examination of the basic components of the LabVIEW language. The measurement itself is conducted by the standard function 'LabVIEWtimer' with 1 ms resolution [7]. To verify the method a more accurate timer was used, based on the Assembler code. Created especially for the purpose (described with WW acronym) has 1 ns real resolution, using the processor registers. Limited ways of applications make it useful only as an additional tool. The authors focused on the results obtained by standard function, however the comparison between the results of those two timers has been made and will be presented further.

Examination of the basic functions was an introduction to a wider analysis of dependencies between the effectiveness and the hardware platform. Therefore the same experiments were conducted on a few different computer configurations. In conclusion, there was conducted a full-scale analysis of the complex and functional

instrument, i.e. a virtual spectrum analyzer wider presented in [3]. It was examined in detail, working on the base computer. Such analysis was important as the example of the typical and practical work regime. The latter is the most important and these experiments are explained further. The analyzer code consists of six main functional blocks responsible for certain operations. In current research the data acquisition function was not taken into consideration as it is connector rather to the hardware part of the VI. The data processing sections are:

- a) initial processing of the samples vector
- b) signal averaging in time domain
- c) windowing and filtering
- d) calculation of power or amplitude spectrum
- e) signal averaging in frequency domain
- f) displaying results.

Each section was examined separately as the part of the more complex testing program. At the same time there was examined overall analysis time since obtaining measurement data in the digital form until presenting results on the computer screen. Because all the sections are in the sequence one after another, the timers were placed between them. They allowed simultaneous measurement of all partial times during one experiment (the partial time measurements are calculated as the differences between the indications of two adjacent timers). Each program block was put into the FOR loop, which number of iterations is a parameter set by the designer. The same is with the number of conducted experiments, that means repeats of the main program loop containing all the blocks. The method verification was to be conducted according to the methodology presented in [12]. The examinations of the instrument in the practical (“field”) configuration were conducted using another algorithm, also presented in [12].

As the numerous and various experiments were conducted, aroused the need for unequivocal marking of the numbers of the experiment repeats (the letter N) and the section repeats (the letter P). For all of the examinations there was one and unified signal pattern used. It was generated by the standard LabVIEW function. The signal was a sinusoidal pattern of 1024 samples, amplitude of 1V and frequency of 1 kHz.

4. EXAMINATIONS

The analyzer was tested for four sets of work parameters. They have influence on the instrument’s speed of work. Their choice allowed to find out, how significant computational cost is connected to the considered operations. These sets were as follows:

- a) rectangular window without filter and power spectrum,
- b) rectangular window, Butterworth LP filter of the 6th order, power spectrum,
- c) Hamming window, Butterworth LP filter of the 6th order, power spectrum,
- d) rectangular window without filter, amplitude spectrum

For the presented configurations research was conducted for both overall time t_c and partial times t_s of all program blocks. The latter were also summed to obtain time t_{Σ} (result of adding times of all parietal operations, which should be equal to the t_c) and compare it with time t_c . The computations were conducted for three sets of P and N values:

- the experiment was executed once, but every section was repeated (N=1, P=1000),
- both the experiment and all the sections were repeated (N=1000, P=1000),
- the experiment was repeated, but every section was executed only once (P=1, N=1000).

For such parameters there were obtained exemplary results of the overall time measurements. They are presented in the Table 1, where: \bar{t}_c is the average time of the overall experiment, \bar{t}_{Σ} is the averaged sum of all the section partial times, σ_{Σ} is the standard deviation of the averaged partial time measurements and $\bar{t}_{\Sigma N}$ is the average time of the overall experiment (calculated by division by the experiments number). It is worth noticing that despite greater accuracy of the WW timer (which is confirmed by lesser standard deviation σ) both timers give similar values. That confirms assumptions of ability to use the ‘LabVIEWtimer’ function for the time measurement. The results of partial analysis are in the Table 2. The t/t_{Σ} factor is the percentage of the participation in the overall VI execution time.

Table 1. Results of the spectrum analyzer [3] analysis in configuration b

	Exp. conditions		Timer	\bar{t}_c	$\bar{t}_{\Sigma N}$	$\sigma_{\Sigma N}$	\bar{t}_{Σ}
	P	N		[ms]	[ms]	[ms]	[ms]
1	1000	1	LabVIEW	10,44	-	-	10,47
			WW	10,42	-	-	10,38
2	1000	1000	LabVIEW	-	10,55	0,28	10,47
			WW	-	10,29	0,07	10,43
3	1	1000	LabVIEW	14,29	12,95	2,39	11,87
			WW	14,13	12,37	1,38	12,32

Table 2. Results of the partial time analysis in configuration b

Section	WW timer, random fetching, N=1000				
	P=1		P=1000		$t/t_{\Sigma S}$ [%]
	\bar{t} [ms]	σ [ms]	\bar{t} [ms]	σ [ms]	
a.	3,66	0,59	3,34	0,03	31,9
b.	0,11	0,32	0,03	0,002	0,2
c.	0,63	0,48	0,43	0,004	4,1
d.	6,64	0,78	6,15	0,05	58,8
e.	0,10	0,30	0,008	0,005	0,1
f.	0,73	0,47	0,51	0,01	4,8
Sum ($\bar{t}_{\Sigma S}$)	11,87	-	10,47	-	100

As can be easily noticed, the results, which are the average values after summing together, give the values similar to those of the overall analysis. This confirms that the partial measurements are a good method for determining the time of the basic operations and there are no “hidden” operations between the selected program blocks.

The need to examine the dependence between the operation duration time and the number of iterations emerged because of the significant computational effort of the experiment in the configuration P=1000 and N=1000. The processor’s cache memory work regime causes an interesting effect. The averaged result is always lesser than the result of the single operation execution. The feature has been confirmed for both overall and partial times. The illustration of the problem is presented in the Fig. 2 and Fig. 3. It is worth noticing that the times become stable after about 20 repeats and reliable results of the instrument effectiveness can be obtained when P=20 (for both N=1 and N=1000). This helps to significantly shorten research time.

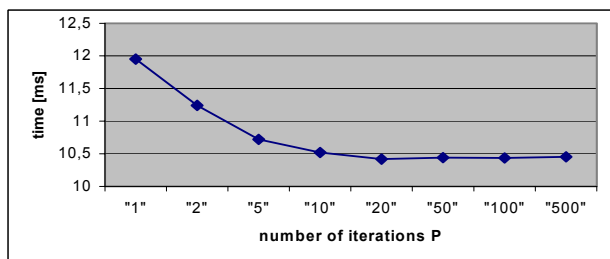


Fig. 2 - Average time of the analyzer’s work with respect to P (configuration b).

Results of the experiments obtained for the P=1000 and N=1000 configuration allow to determine the average time with the greatest accuracy. This includes both partial and overall times. Unfortunately, in the most probable practical case the instrument performs every operation only once. The overall processing algorithm may be

repeated many times. It is justifiable for the stationary signals measurement using linear and exponential averaging [8].

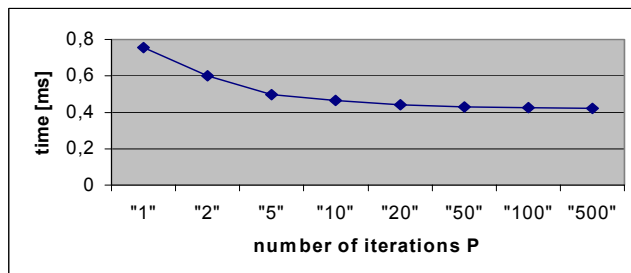


Fig. 3 - Average time of the windowing and filtering function computation with respect to P (configuration b).

The largest amount of information is then delivered by the experiments in the P=1 and N=1000 configuration. This means that during design of the virtual instrument it is highly advisable to test its effectiveness in various situations, as has been presented. Another important fact that has been revealed after inserting into the LabVIEW code the time delay function, just before the first block. Such an experiment allowed to determine that the measurement of the overall time of one analysis sweep may give different results depending on the program delay inserted between the loops in the spectrum analyzer. The results for \bar{t}_c , P=1, N=1000 in Table 1 show values with no delays. Adding “wait” function into the code and setting its argument for 20 ms allowed to obtain results comparable to those from the rest of the table. The effect is connected to the functions responsible for displaying results, which have problems with fast memory allocation and clearing variables of displays and plots. It is also important to know that the function execution time is not equal to the transporting results from the computer memory to the graphic’s card memory. Hardware operations are subsequent to the display function.

5. OPTIMISATION PROCESS

Based on the conducted research the conclusions about the abilities of the LabVIEW functions have emerged. Therefore we decided to use the knowledge about the time efficiency of the operations to redesign some of them and obtain an instrument with significantly better time characteristics. The basis for this approach was the detailed examination of the speed of work of more complex functions, including the average values, standard deviations and histograms. The latter allowed to see how the measured time value changes

during the experiment. The exemplary results of the processing are shown in Fig. 4.

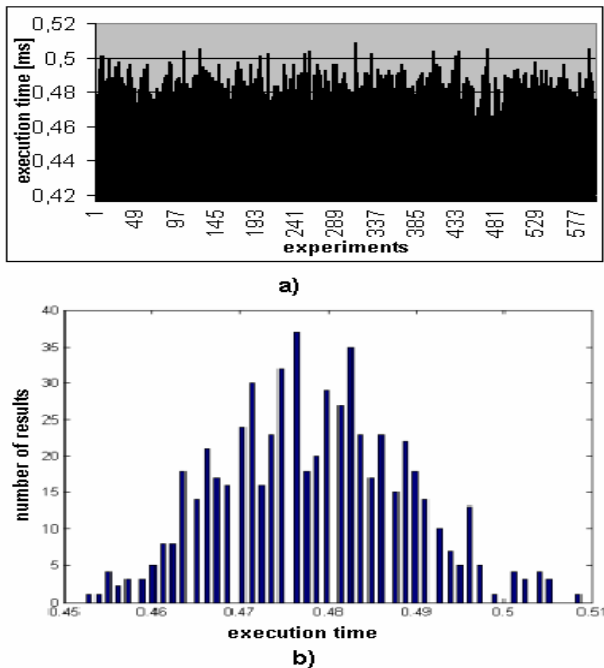


Fig. 4 - The illustration of the measurement process for the displaying graphic block (a) and its histogram (b).

The function’s execution times very often (however, not always) form a gaussian curve with the value close to the average on the top. Therefore the average time was assumed reliable to the purpose of the optimization process. We focused on the blocks which had the longest time of operation and their use in the “Real-Time” instrument was not possible. These were blocks “a” and “d”, as can be seen in the Table 2. The task was easier as these blocks consisted of some user-written functions, which could remain highly ineffective. The assumption for this phase of optimization was that the “ready-made” LabVIEW functions (existing in the programming environment) are designed with respect to all possible requirements and do not cause unnecessary delays. The two mentioned blocks were significantly slow compared to the rest of the operations and we tried to change or omit some of their inherent procedures.

A. The initial processing block

The block consisted of four main operations: sorting the signal vector, choosing the first sample and the last one as the minimum and maximum signal value, comparing them to the constraints forced by the user and passing the vector further or showing an error message. Detailed analysis of every step of the block revealed that sorting function works relatively long and its efficiency reciprocally proportional to the length of the vector. Therefore

we decided to alter this procedure with function searching for the minimum and maximum value iteratively. Moreover we compressed any unnecessary messages to the minimum. As the result we created a very efficient block with the execution time only below 300 μs. The time analysis was conducted with the use of the ‘LabVIEWtimer’. The change in the operations’ speed shows Table 3.

Table 3. Illustration of the optimization process for the initial processing block

Function version	N = 1000			
	P = 1		P = 1000	
	\bar{t} [ms]	σ [ms]	\bar{t} [ms]	σ [ms]
before	3,66	0,59	3,34	0,03
after	0,29	0,19	0,01	0,01

As can be seen, the standard deviation is still low for very large number of repeats within the experiment (P=1000) but for P=1 its value is almost 25 per cent of the average. This is the effect of a wide histogram with significant spread of the measurement results around the average. The standard deviation from the latter case gives more information about the possible spread of the values in the practical situation.

B. Spectrum analysis block

The block consists mainly of amplitude and power spectrum functions, procedures for the vector length tracing and dividing it in the right position. Changing the latter into very basic versions and forcing the dyadic length of the vector (i.e. the length being a power of 2) allowed to accelerate speed of the block in a noticeable way, as only the FFT operation could be performed (instead of the DFT, which is significantly slower [3]). The results of the optimization are shown in the Table 4.

The profits in the time efficiency are even greater in this situation. The deletion of all the unnecessary functions and imposing use of only FFT operation allowed to save over 5 ms in the overall VI work time. The standard deviation is again lesser for P=1000, but more reliable results are still for P=1. Adding \bar{t} and σ in the “worst case” for P=1 we get about 1 ms execution time.

Table 4. Illustration of the optimization process for the spectral operations block

Function version	N = 1000			
	P = 1		P = 1000	
	\bar{t} [ms]	σ [ms]	\bar{t} [ms]	σ [ms]
before	6,64	0,78	6,15	0,05
after	0,81	0,18	0,60	0,006

C. Overall code optimization

The time optimization of the two functions presented proved that the approach to the problem

was correct and that the time factor is a very important matter during the Virtual Instrument design. Without fulfilling very strict time infringements the instrument may work correctly but would not be fast enough to perform operations within the time required for the data acquisition. As the initial processing and the spectrum analysis blocks were the slowest in the whole code, the other blocks were left unchanged. The result of the experiments with the optimized code is shown in Table 5. and Table 6.

Table 5. Results of the optimized spectrum analyzer analysis in configuration b

	Exp. conditions		Timer	\bar{t}_c	$\bar{t}_{\Sigma V}$	$\sigma_{\Sigma V}$	$\bar{t}_{\Sigma S}$
	P	N		[ms]	[ms]	[ms]	[ms]
1	100 0	1	LabVIEW	0,893	-	-	1,06
			WW	0,921	-	-	1,01
2	100 0	100 0	LabVIEW	-	1,09	0,11	1,06
			WW	-	1,04	0,08	1,02
3	1 0	100 0	LabVIEW	2,37	2,37	0,19	2,37
			WW	2,75	3	0,13	2,36
				2,72	8		2,36

Table 6. Results of the optimized partial time analysis in configuration b

Section	WW timer, random fetching, N=1000				
	P=1		P=1000		$t/t_{\Sigma S}$ [%]
	\bar{t} [ms]	σ [ms]	\bar{t} [ms]	σ [ms]	
a.	0,2879	0,1172	0,0980	0,0020	12,13
b.	0,1655	0,0707	0,0122	0,0007	6,98
c.	0,4916	0,1639	0,2213	0,0036	20,71
d.	0,8059	0,1333	0,5654	0,0051	33,96
e.	0,1648	0,1364	0,0073	0,0005	6,94
f.	0,4576	0,0779	0,1604	0,0039	19,28
Sum ($\bar{t}_{\Sigma S}$)	2,3732	-	1,0645	-	100

The whole experimental process applied to the optimized functions inserted into the virtual spectrum analyzer allowed to reveal how the execution times changed as a result of the optimization. The relevant changes can be seen only in the partial times of the “a” and “d” blocks and the overall times, which are dependant. The other blocks’ times of execution remain mainly unchanged. This proves that the functions timely independent on each other. The change in one block does not cause proportional changes in the subsequent functions, their execution times remain constant. It is worth noticing that the percentage participation of the blocks changed, and main proportions are now less distinct, only spectrum calculating function remains the most time consuming. This proves that the signal processing

functions are the most time consuming and their use in the Real-Time instrument should be strictly limited.

The verification of the optimised functions’ properties confirmed some effect detected and described earlier. The dependency between the value of P and function’s execution time still suggests taking measurements for the P=20 (sometimes even P=15), as is shown in the Fig. 5 and 6.

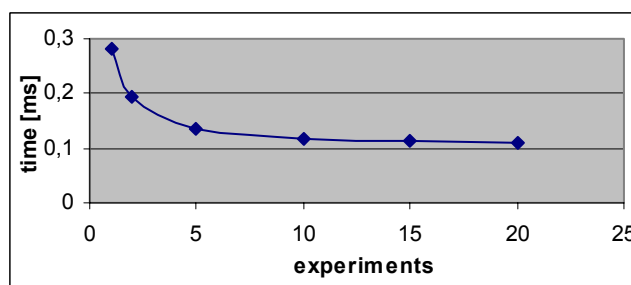


Fig. 5 - Average time of the optimized analyzer’s work with respect to P (block b).

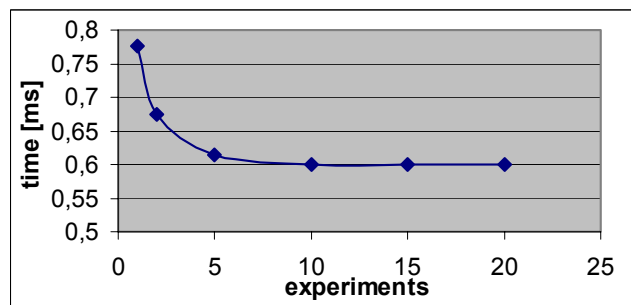


Fig. 6 - Average time of the optimized analyzer’s work with respect to P (block d).

6. CONCLUSIONS

The research allowed to verify assumption about dependencies among software parts. Although the examined analyzer was characterized by the sequential order of the functional blocks, the results obtained can be generalized. The methodology of Real-Time instruments design is affected by the difficulties of the accurate time measurement in the operating system conditions. The accurate results can be obtained for every LabVIEW function, but they shouldn’t be treated as invariantly true. There should be always a margin for disturbances in the operating system (the size of the margin should be calculated on the basis of the standard deviation). Because of the multitasking character of modern operating systems during instrument’s work their working time can sometimes become relatively long. The multithreading processing module is responsible, imposing queues of processes. During design of the instrument one should take such anomalies into consideration. They are very rare, but

possible to appear during normal work regime. The most common are values close to the average, but there may be also small number of values far from them. Enlarging the margin for the time required to execute operations in so-called "hard" Real-Time mode becomes highly advisable. The cost of it is narrowing of the measurement signals' band, as the connections between the acquisition time and the possible measurable band is determined by the equation:

$$t = \frac{1}{f} \quad (5)$$

Another solution is work in the "soft" Real-Time mode. Lessening the band is not required then, but the samples acquired by the DAQ card should be stored in additional registers or cyclical buffers [10]. Moreover, there should be a system of self-control implemented. It could inform the user, when the Real-Time conditions are not met. The simplest solution could be the control diode on the front panel of the VI, but automatic variants are also possible.

The conducted measurements allow to propose consistent and versatile approach to the Real-Time instruments design. During the phase of the instrument design the optimal measurement method appears to be the one with experiment and P-times repeated every functional program block. The method allows measurement of the overall and partial times. It assures conformability of overall time and the sum of partial times $\bar{t}_c \approx \bar{t}_{\Sigma}$. For the assessment of the existing instrument's effectiveness the best is the method of the multiple experiments repeating with single sections (for example N=1000, P=1). The real work conditions are then considered.

Optimization of the designed instrument is possible by means of accelerating functions work or reducing instrument's functionality by removing operations lasting for too long or of lesser importance. The proportions between these two methods should be selected individually according to the needs or possibilities. Parallel computations become then of a great importance in conjunction with synchronization inside the program. The matrix and signal processing operations are the most time-consuming and their use in the VI should be restricted. The problems stated here will become more significant as the abilities of the instruments will expand (widening band, more complex applications etc.). The VI design process must be modified, taking into consideration the time optimization process.

In the soft Real-Time mode the main limitation for the complex devices (such as spectrum analyzer) are software operations. The research of the time

dependencies allow successful optimisation for the price of functionality of the code. For the presented analyzer the bandwidth was widened from 20kHz to 200 kHz on the 750 MHz processor (with use of a faster DAQ card). Accelerating the speed of the software part should cause the acceleration of the whole instrument efficiency. However, the time disturbances of the operating system are main and significant infringement. Its examinations are still under research.

It should be pointed out that as the presented problem is considered in the wide sense for the first time, there is further research needed. The use of the presented method allowed to assess the VI software part time consumption and to optimize it. The Real-Time work mode could become a great advance of the VI technology, expanding its applications. Moreover, the designer of such an instrument might be obliged to optimize the code considering the time analysis. It could require writing the functions' code from scratch (without resorting to the LabVIEW functions but rather using other programming languages, such as Assembler or C). As show the research results, one of the infringements could be the operating system, causing serious disturbances in the VI's work regime. It has been stated that the Windows 9x family is less stable than the NT family and though the execution time on both is comparable, for the Real-Time conditions the latter should be used. The solutions of the problems presented above are yet to come.

7. REFERENCES

- [1] S. Nuccio, C. Spartaro. Approaches to Evaluate the Virtual Instrumentation Measurement Uncertainties. *Proc. IEEE IMTC Conf.*, Budapest, Hungary, May 21-23, 2001, pp.84-89.
- [2] S. Nuccio, C. Spartaro. Assessment of Virtual Instruments measurement Uncertainty. *Computer Standards & Interfaces* 23 (2001), pp.39-46.
- [3] P. Bilski, W. Winiecki. Virtual Spectrum Analyzer Based on Data Acquisition Card. *IEEE Transactions on Instrumentation and Measurement*, Feb. 2002, Vol. 51 No. 1, pp.82-87.
- [4] Ch. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman 1994.
- [5] J. Overstreet, A. Tzes. *Internet Based Client/Server Virtual Instrument Designs For real-Time Remote-Access Control Engineering Laboratory*.
<http://mechanical.poly.edu/research/control/OT98csm.pdf>

- [6] M. Stegawski, R. Schaumann. A New Virtual-Instrumentation-Based Experimenting Environment for Undergraduate Laboratories with Application in Research and Manufacturing. *IEEE Trans. on Instrumentation and Measurement*, Vol. 47, No. 6, December 1998, pp. 1503-1506
- [7] www.ni.com
- [8] R. Lyons. *Understanding Digital Signal Processing*. Adison Wesley Longman 1997.
- [9] B. Wichmann. Measurement System Validation: Validation of Measurement Software. *National Physical Laboratory*, Queens Road, Teddington, Middlesex, United Kingdom, 2000.
- [10] *LabVIEW - User Manual, ver.6.1*. National Instruments Corp., Austin, 2002.
- [11] G. Chiorboli. Uncertainty of Mean Value and Variance Obtained from Quantized Data. *Proc. IEEE IMTC'02 Conf.*, Anchorage, AK, USA, May 21-23, 2002, pp. 1043-1048.
- [12] W. Winięcki. Methodology of Virtual Instruments' Time Analysis. *Proc. of the Workshop IDAACS'2003*, Lviv, Ukraine, Sept. 8-10, 2003).

Piotr Bilski was born in 1977 in Olsztyn, Poland. He received his M.Sc. degree in 2001 from the Faculty of Electronics, Warsaw University of Technology. He is a PhD student in the Institute of Radioelectronics in mentioned university. His science interests are computer-aided measurements, diagnostic of electronic and mechatronic circuits and relative and object databases.



Wieslaw Winięcki received his MSc and PhD degrees from the Faculty of Electronics, Warsaw University of Technology, in 1975 and 1986, respectively. Since 1975 he has been with the Institute of Radioelectronics of that university. Since the beginning of his professional carrier he has been

involved in the activities of the Group on Computer-aided Measurements concerning the hardware and software for measuring systems. Since 1987 he has worked as Assistant Professor on the problems of measurement automation and interface systems. He is the author of two books and over 100 scientific publications. Head of the Computer-Aided Measurement Laboratory; member of the Measuring Systems Section of the Metrology and Instrumentation Committee, Polish Academy of Science; vice-president of the Measurement Committee of the Polish Society for Measurement, Automatic Control and Robotics POLSPAR; and member of IEEE.