Application of Cyclical Buffer Technique in Real-Time Virtual Instrumentation

Piotr Bilski[1], Wiesław Winiecki[1]

[1] *Institute of Radioelectronics, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665, Warsaw, Poland, +48 022 660, pbilski@ire.pw.edu.pl, W.Winiecki@ire.pw.edu.pl*

Abstract- The paper presents an approach to Real-Time virtual instrument design. It uses cyclical buffer technique to obtain an instrument working in Real-Time mode using general purpose operating system and basic data acquisition card. Advantages and drawbacks of the approach in comparison to hardware Real-Time solutions are presented. Software architecture of the cyclical buffer solution is discussed in detail. Dependency between the cyclical buffer fill ratio and convergence function is presented. Experiments conducted in order to determine abilities of the cyclical buffer are described. Three main buffer states, influencing the ability of the instrument to work in RT mode are presented.

I. Introduction

Modern virtual instruments' (VI) progress leads to sophisticated solutions, which compete with the traditional devices. VIs are advancing in two directions: design of the more complex hardware (data acquisition (DAQ) cards, PXI computers or FieldPoint modules), and new versions of integrated software environments. The former allow to create Real-Time (RT) instruments based on the specialized hardware, while the latter help to create a program for RT operating system run on that hardware. Numerous products offered, for example, by National Instruments company [1], enable design of the instruments fulfilling time-critical tasks. These are both hardware solutions (such as specialized DAQ cards [2]) and software modules (such as LabVIEW Real-Time [1]). Their main advantage is the ability to create instrument working in the Hard Real-Time (HRT) mode, when time limits are never violated. Their disadvantage is the high price, increasing the cost of multiple measurement sites. The paper presents the approach for realization Soft Real-Time (SRT) mode instrument, where time limits sometimes may be violated. It answers the question whether creating RT VI without specialized hardware and software solutions is possible, and what are its characteristics. Cyclical buffer technique is used to complete the task. In section II of the paper cyclical buffer architecture is presented. Section III describes the plan of experiments, while in section IV results and their analysis is presented. Conclusions and future prospects are included in section V.

II. Circular buffer technique

A. Theoretical principles

Hardware RT solutions, supported by the integrated software environments are intensely developed. Their application using specialized equipment is easy and efficient. The main disadvantage is the high cost, especially when measurement stand is multiplied (i.e. for educational purpose). Aiming at cost reduction, the designer can develop RT instrument using circular buffer, which is realized using LabVIEW function. The technique, described in [3], requires no specialized hardware and allows to design of the RT instrument with general purpose operating system and basic DAQ card. The idea is to force simultaneous data acquisition and software-based signal processing. While calculations are performed, signal samples are stored in the buffers. RT mode requires simultaneous acquisition of samples and signal processing. If software operations are fast enough to cause no delays in the DAQ operation, HRT mode is possible. If such delays occur rarely, SRT mode is possible. In the latter, the software operations' execution time must not cause buffer overflow. The proposition of such buffer configuration was presented in [4]. Exemplary configuration for two buffers is in Fig. 1.:

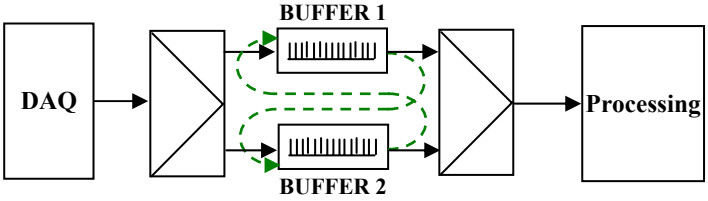


Fig. 1. Scheme of the double-buffer measurement configuration.

The scheme may be easily expanded to the multiple buffer configuration. Additional buffers are useful when no HRT mode is possible and numerous signal vectors must be stored in memory before they are processed. Such solution applies to SRT mode, which is the main interest of the paper. The dotted lines in Fig. 1 indicate the way the samples are stored in the buffers. When one buffer is full, another is filled with samples. One buffer is "primary" (currently filled with samples), another is "secondary" (waiting for its turn). The buffers' roles change repeatedly. Circular buffer realizes multiple buffer configuration, where every buffer stores a single signal vector. LabVIEW functions' implementation reserves an amount of memory for storing all acquired samples, while software operations are performed. Multiple buffer architecture is obtained by configuring length of the memory space as the multiplied length of the signal vector. While one portion of samples is processed by the software part of the instrument, another one is passed to the buffer. For example, double-buffer configuration for the signal of 1024 samples is realized by the circular buffer with 2048 samples of length. When 1024 samples are acquired, they are passed to the software part of the VI, where signal processing is performed. After reaching the end of the buffer, samples are inserted from the beginning only if there is enough empty space. Otherwise, buffer overflow occurs and signal samples are lost. SRT mode requires that no samples are lost in the data acquisition process. It is possible only when software part works fast enough to process subsequent signal vectors, before incoming samples fill the whole buffer. Optimization of the software is then required. Realization of the circular buffer technique is explained in Fig. 2. It is the reserved memory, which stores samples from DAQ card, ensuring parallel processing of the data. Numbers "1/2" and "2/1" determine the way the buffers are filled ("1" is for the buffer currently filled).

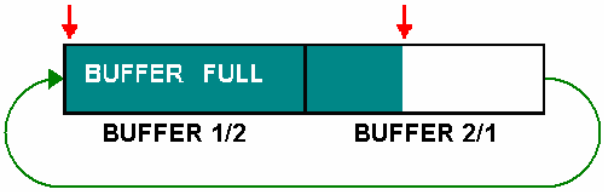


Fig. 2. The idea of the circular buffer.

The technique of software optimization was presented in [7]. It utilizes statistical analysis of the repeated experiments (software functions executions). Time of execution of the program functions is known with finite accuracy. Round-robin scheduling mechanism of the operating system makes execution time measurement difficult. Duration of the same operation performed many times is always different. Due to the operating system architecture (such as Microsoft Windows), every active task (program) gets some of the processor's time. VI application is only one of many processes residing simultaneously in memory. Therefore its execution time is affected by random allocation of the processor's time. Average value of the multiple measured function time executions properly assesses expected duration of the software part. However, it can not be used to determine RT conditions. Existence of the other processes in memory causes time disturbances [7], increasing duration of the VI software part. Experiments [7] have shown that after optimization time disturbances are a significant part of the function execution time. RT conditions require setting safe parameters to measured software operation time, considering disturbances. The predefined time of the data acquisition t_{acq} must not be shorter than the time of the software operations. Their efficiency is determined by the time of signal processing t_{proc}, time of results visualization t_{vis} and duration of disturbances. Therefore for RT conditions:

$$t_{proc} + t_{vis} + t_{m\arg} < t_{acq} \tag{1}$$

where t_{marg} is margin set for the time disturbances. Its length is determined during experiments with software part efficiency [7]. Parameter t_{vis} is negligibly small [7], so:

$$t_{proc} + t_{vis} \approx t_{proc} \tag{2}$$

Hence, to meet RT conditions:

$$t_{proc} + t_{m\arg} < t_{acq} \tag{3}$$

Fulfilling this condition ensures determinism under the general purpose operating system. While Real-Time operating system architecture handles determinism automatically (setting higher priorities to the most important tasks), without it, the VI designer must write a program considering the speed of the software functions.

Condition (3) also limits maximum frequency band of the analyzed signal. HRT mode requires fulfilling this condition all the time, suppressing signal's frequency band to the values useless in practical applications. On the other hand, SRT mode is not so demanding, and for the Windows-based VI it is a compromise between the determinism and practical applications.

B. Circular buffer in practical realization

LabVIEW functions configuring and running the cyclical buffer require three parameters, which will be examined: n_{buf} – buffer size, f_s – sampling frequency referring to n_s (number of samples acquired within a second), n_{proc} – number of samples processed during the software cycle. Buffer mechanism gives information about its fill rate by the average scan backlog s_{bck} value. It is the number of samples left in a buffer after every processing cycle, which will be used as the measure of the operation efficiency. To make it more universal, we scale this parameter (with respect to n_{buf}) to obtain percentage value. Time analysis of LabVIEW operations [5] revealed that the duration of standard software operations is much smaller than one second. This means multiple measurement and processing cycles can be performed during this time period. Therefore we propose the maximum number of cycles k_c run by the VI in SRT mode during one second:

$$k_c = \frac{\max(n_s)}{n_{proc}} \tag{4}$$

The value of k_c depends on the buffer configuration and VI speed. The former was examined, as shown in section IV, according to the value of t_{proc}, which must be determined for every computer configuration. It must fulfil the condition:

$$t_{proc} < \frac{n_{proc}}{f_s} \tag{5}$$

The buffer's size should be large enough to avoid overflow, leading to the loss of the samples. Relation between the circular buffer's size and number of the buffers (as in Fig. 1) is expressed by (6):

$$k_{buf} = \frac{n_{buf}}{n_{proc}} \tag{6}$$

where k_{buf} is the number of the used buffers. Parameter k_c is an efficient measure, determining the instrument's ability to work in the RT mode. However, in [4] the convergence function was also proposed for the parallel processing architecture:

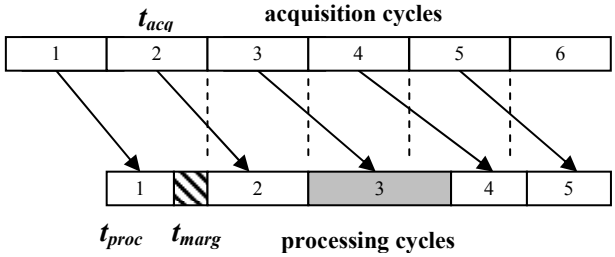


Fig. 3. Illustration of the dependencies between the data acquisition and processing times.

Acquisition cycles are regular and have constant duration, as they are performed by the DAQ card. Processing cycles are performed by PC computer and their length varies, depending on the processor's state. Convergence function expresses delays between acquisition and processing cycles:

$$f_{Nd}(i) = i - j \tag{7}$$

where i is index of the current acquisition cycle and j is index of the executed processing cycle. If there are no delays (signal processing is performed faster or in the same time as acquisition), $i = j$ and $f_{Nd}(i) = 0$. Fulfilling this requirement all the time, HRT mode is possible. If $f_{Nd}(i)>0$, but buffer is not overflowed, SRT mode is possible [4]. Illustration of the convergence function is in Fig. 4.:

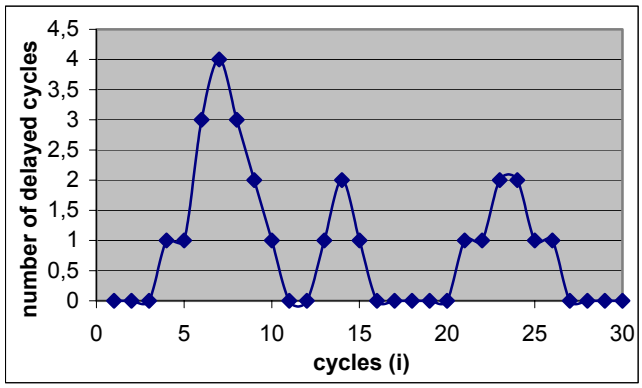


Fig. 4.: Synchronization function $f_{nd}(i)$ of the delayed cycles.

There is a connection between the synchronization function and circular buffer configuration. The parameter "number of delayed cycles" expresses subsequent buffers, storing data waiting to be processed. If $f_{Nd}(i)$ is close to the number of buffers k_{buf}, then SRT mode is in danger. Increasing k_{buf} might be the solution for the circular buffer overflow. Connection between $f_{Nd}(i)$ and another coefficient was determined during the experiments.

III. Plan of the experiments

Research on the circular buffer was conducted in two ways. Firstly, the buffer's mechanism was examined in a simple VI configuration. Secondly, a virtual spectrum analyzer was designed using that technique. Results for the latter were presented in [6]. This paper focuses on the former. The difference between these two experiments is that for the circular buffer only its abilities were measured, while for the virtual spectrum analyzer a complex instrument was tested. The basic SRT instrument for the tests was designed using a PC computer with general purpose operating system (Microsoft Windows) and LabVIEW 7.0 software environment. The PCI-6023E DAQ card was used for samples acquisition. PC computer configuration contains: AMD Duron 1200 processor on Soltek mainboard with 256 MB of RAM. The VI for the tests was designed as a simple program consisting of the buffer configuration functions and the function waiting idle for a defined amount of time. This way the instrument's software work speed would be set. For the preselected value of t_{proc} various values of the f_s and n_{proc} were used to determine relation between the two and confirm or deny equations (4), and (5). The experiments consisted of 10000 processing cycles to observe circular buffer in a long period of time and ensure that there will be no buffer overflow.

IV. Experimental results

The examinations of the cyclical buffer were performed for multiple buffer configurations. Along with the buffer configuration, time of the computation was also selected. Three significant states of the buffer were determined. They will be illustrated the following sections.

A. Acquisition delivers samples faster, than they are processed

This is the most common situation in the real instrument work regime. Modern DAQ cards work with high sampling rates (up to 1Ms per second), which imposes high-speed requirements on the software part of the instrument. Moreover, for such fast computations, time disturbances become more important (their duration closes to the time of the processing) and make holding SRT conditions impossible. Exemplary situation is presented in Tab. 1.

Tab. 1. Results of the cyclical buffer examinations for $n_{buf}=2000$, $n_s=1000$, $n_{proc}=125$

t_{proc} [ms]	50	100	120	123	124	124,6
S_{BCK} [%]	6.90e-4	3.71e-3	1.75e-2	3.48e-2	1.28	5.81

The main factor determining the possibility of working in the SRT mode is here t_{proc}. If (5) is fulfilled and the maximum value of s_{bck} is no more than 50 %, then stable work in SRT mode is possible. For the multiple buffer configuration n_{buf} is important, as it must be large enough to store all the samples, which wait for the processing. Safe work regime requires that

$$n_{buf} \geq 8 \cdot n_{proc}$$

Example of the s_{bck} fill rate in the safe situation is in Fig. 5a, while convergence function is shown in Fig. 5b.:

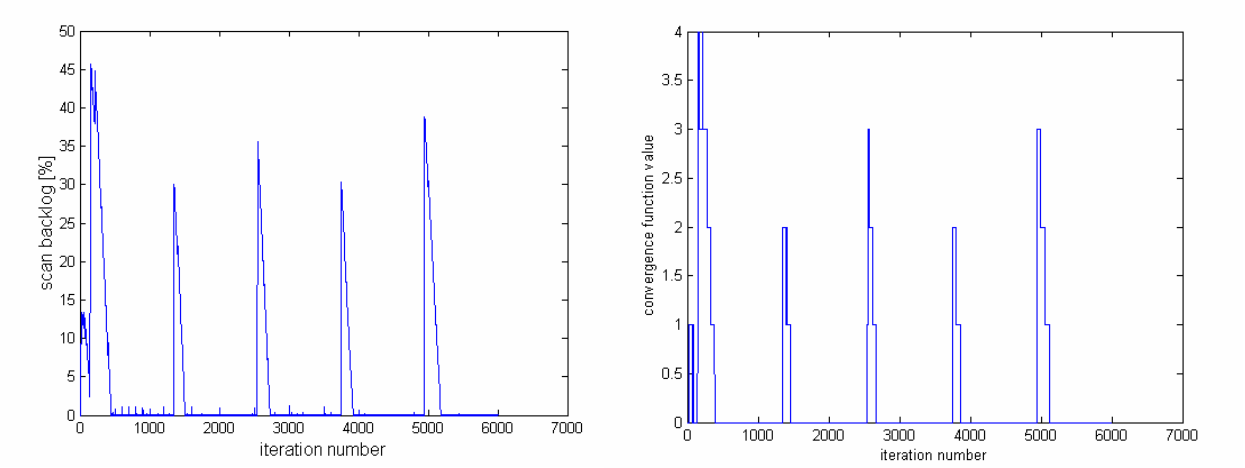


Fig. 5.: Value of s_{bck} (a) and $f_{Nd}(i)$ (b) for $n_b=2000$, $n_s=1000$, $n_{proc}=125$ and $t_{proc} = 120$ *ms*

The convergence function is here greater than zero, but quickly returns to the deterministic state. In Fig. 5b importance of the large n_{buf} is visible – if the buffer was two times smaller, SRT mode would not be obtainable, due to the buffer overflow. For $n_b=2000$ and $n_{proc}=125$ we have $k_{buf} = 8$ (eight buffers) and half of them are filled from time to time. When (5) is not fulfilled, the buffer quickly becomes full and no RT mode is possible. Size of the buffer is then irrelevant, as even large values can only delay inevitable overflow. Such situation is shown in Fig. 6a, while in Fig. 6b convergence function is presented:

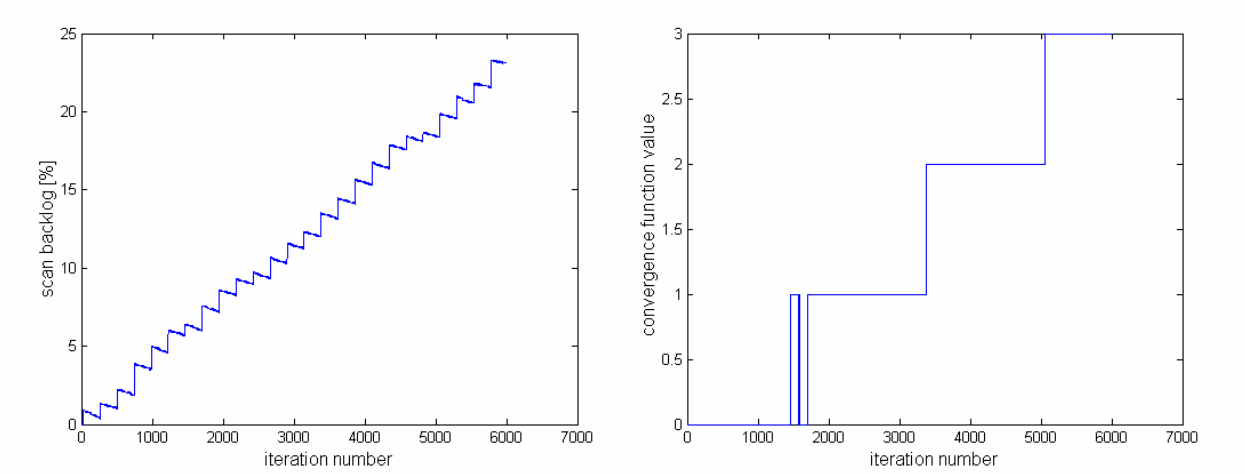


Fig. 6.: Value of s_{bck} (a) and $f_{Nd}(i)$ (b) for $n_b=2000$, $n_s=1000$, $n_{proc}=125$ and $t_{proc} = 124.5$ *ms*

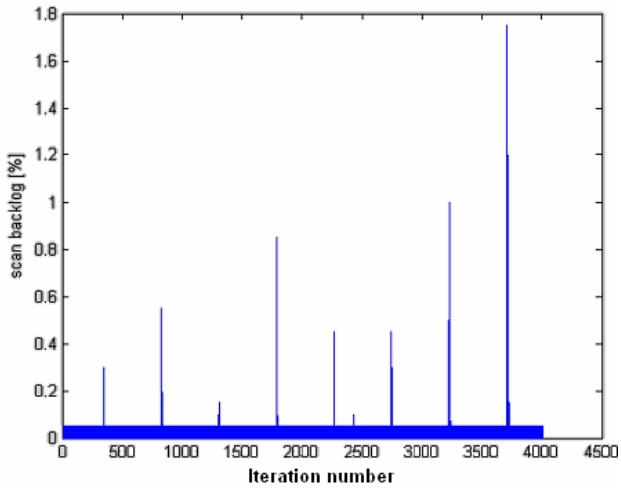
The difference between the experiments from Fig. 5 and Fig. 6 is the value of t_{proc}. Though the circular buffer has constant parameters, increasing time of computations to the border, set by equation (5) caused slow but intermittent filling of the buffer. For the preset buffer configuration maximum t_{proc} is then 124 ms. On the other hand, for $t_{proc} = 124$ *ms*, number of the processed cycles during one second $k_c = 8$. In order to increase the instrument's efficiency (k_c) the designer must suppress t_{proc}. For example, $t_{proc} = 62ms$ allows 16 processing cycles. The parameters t_{proc} and k_c are reversly proportional. Convergence function in Fig. 6b is stable, but increases withot a break, until all of the buffers are full and no further processing is possible. Results presented so far are regular, unlike the results for the virtual spectrum analyzer [6]. The difference is caused by the complexity of the latter – the signal processing operation is performed by the processor in a different way than simple waiting idle. However, main characteristics of the circular buffer are the same for both experiments.

B. Acquisition delivers samples with speed similar to the processing operation

If the number of samples obtained during one second is at most two or three times larger than the number of samples processed during one cycle, both SRT and HRT modes are obtainable, as there is enough time to finish all the required calculations. If the instrument is not too complex, it can perform all the necessary operations within that time. Buffer fill rate is close to zero all the time. Convergence function is also zero and even the first buffer is not filled. The situation is presented in Fig. 7. and in Tab. 2.

Tab. 2. Results of the cyclical buffer examinations for $n_{buf}=2000$, $n_s=1000$, $n_{proc}=500$

t_{proc}[ms]	100	250	400	450	480	490
S_{BCK}[%]	1.4294e-005	2.7473e-003	2.1229e-003	7.2428e-004	1.0989e-003	2,1587e-1



Value of s_{bck} for $n_b=2000$, $n_s=1000$, $n_{proc}=500$ and $t_{proc}=100$ *ms*

C. Acquisition is slower than the processing operation.

When the number of samples acquired during one second is equal or smaller than the number of samples processed, there is no danger of buffer overflow, as the software part waits in the idle state until the DAQ operation is complete. also convergence function is always zero. This configuration is not optimal and should be avoided, as the processor's time is wasted. The situation is presented in Tab. 3.

Tab. 3. Results of the cyclical buffer examinations for $n_b=1000$, $n_s=1000$, $n_{proc}=1000$

t_{proc}[ms]	50	100	200	500	800	1000
S_{BCK}[%]	1.10e-3	1.60e-3	6.08e-2	1.96e-3	8.82e-3	3.45e-8

V. Conclusions

The research presented in the paper allowed to determine, for what buffer parameter values SRT is possible in VI design. These results, along with the similar research outcomes from [6], give detailed knowledge about the usefulness of the circular buffer for RT virtual instrument design. Dependencies between the parameters and time of the software operation were defined. The coefficients proposed in the paper provide a unified tool for VI analysis. They can be used during instrument design. Although detailed results of the analysis consider particular hardware configuration, the proposed coefficients allow assessment of the other configurations. They also facilitate selection of the DAQ card and PC computer for a particular instrument. Results obtained from examinations allow proposing a method of SRT availability assessment, which should be further investigated.

References

[1] www.ni.com/realtime
[2] "Getting Started with the NI 7831R", from www.ni.com
[3] "*LabVIEW Measurements Manual*", National Instruments Company, 2001.
[4] P. Bilski, W. Winiecki: Time Optimisation of Soft Real-Time Instrument Design. *IEEE IMTC'04 Conf. (Como, Italy May, 18-20, 2004)*, pp. 2223-2228.
[5] P. Bilski, W. Winiecki: Time Analysis of Virtual Spectrum Analyzer. *IEEE IDAACS'03 Conf. (Lviv, Ukraine Sept. 8-10, 2003)*, pp. 144-149.
[6] P. Bilski, W. Winiecki: A Low-Cost Real-Time Virtual Spectrum Analyzer. *IEEE IMTC'05 Conf. (Ottawa, Canada May, 17-19, 2005)*, pp. 2216-2221.
[7] W. Winiecki, P. Bilski: Time Aspects of Virtual Instrument Designing. *IEEE IMTC'03 Conf. (Vail, USA May, 20-22, 2003),* pp. 913-918.