STANDARD ENVIRONMENT FOR THE SINE WAVE TEST OF ADC'S

J. Márkus and I. Kollár
Department of Measurement and Information Systems
Budapest University of Technology and Economics
H-1521 Budapest, Hungary

Abstract: ADC data sheets and test methods are not yet standardized. A new attempt to create a common platform for these is the draft standard IEEE -STD-1241. However, the methods described in this standard need an extra effort from the user to exactly understand and implement them. It is therefore very reason able to provide programs which implement these methods, and allow manufacturers and users to use them. This assures that the same characterizing quantities will be used for the same purpose by everyone.

The first attempt for this was a LabView program ann ounced in 1999. In this paper – after general considerations – another implementation is presented, running under MATLAB. This will hopefully extend the number of the users of the methods described in the standard.

Keywords: IEEE-STD-1241, ADC testing, sine wave method, MATLAB, EUPAS.

1 INTRODUCTION

Analog-to-Digital Converters (ADC's) have been in use for more than 20 years. However, there is still no standard for the measurement and testing methods, which could be used by the manufacturers and by the users. So, two ADC's made by two different manufacturers are often not really comparable, because of the different test and measurement methods.

To fill this gap, IEEE[1] began new projects a few years ago. The aim of these projects is to standardize the terminology and test methods for both digitizing waveform recorders [1] and analog to digital converters [2]. The IMEKO TC4 Working Group on "A/D and D/A Converter Metrology", EUPAS[2], undertook the task to manage these sta ndardization projects in Europe [3]. The two bodies work closely together.

There exists another initiative, Dynad [5] which deals with dynamic characterization of ADC's. The programs and principles described in this paper relate also to this project, but the primary intention is to contribute to the IEEE standards. In long run, we should consider the development of a common test program.

In 1999, a LabView program was announced by a workgroup in the IEEE Instrumentation and Measurement Society TC-10 [6,7]. This is maybe the first widely available program to bring part of the methods described in the standards [1,2] into practice, in a standard form. It implements the sine wave test method.

The existence of this program induces further thinking on the applicability of the standard, on ways of distributing the standardized methods, etc. This is the topic of this paper. First we are going to discuss general aspects, then give a short overview of the LabView program, and finally discuss an alternative MATLAB implementation.

2 THE SINE WAVE TEST METHOD

Using only a few general methods, most of the tests presented in the IEEE standard can be performed. Maybe the most prevalent method is the sine wave test. The basic idea of the test is the following: a computer, connected to the ADC under test, measures th e output, by applying a sine wave as excitation signal. From the samples the parameters of the sine wave are determined (using signal processing methods, e. g. DFT or parameter fitting). Finally, by subtracting the calculated sine

[1] The Institute of Electrical and Electronics Engineers
[2] EUropean Project for ADC-based devices Standardisation

wave from the samples, a residual vector is taken, suitable to describe several ADC properties, e.g. SINAD[3] or THD[4] [2].

3 IMPLEMENTATION: GENERAL CONSIDERATIONS

By studying the draft standard, one can quickly conclude that while most methods are straightforward, even the one-step algorithms may have different implementations. This means that the results can be slightly different. If this difference is significantly smaller than the error itself, this is not a problem. However, when the problem to be solved may have local minima, there is a chance for significantly different results. This is the case e.g. with the 4 -parameter fitting. In other words, in such cases, users with different programs can conclude different results, even starting from exactly the same data.

This problem can be overcome either by strictly defining the algorithms, which is very cumbersome, and often cannot be implemented by users of commercially available programs. Another possibility is to provide standard programs which can be used by anybody, for data processing or for comparison. We can state in general that the methods described in the standard should be available in the form of programs, too.

4 REQUIREMENTS FOR THE IMPLEMENTATION

Given the variety of computer platforms (PC, Macintosh, Sun, etc), and operating systems even on the PC's (Win2000, Win9x, Linux), a good solution cannot be just a set of executable programs. There are difficulties in simultaneous upgrading, in the differences between compilers, in necessity of system-dependent code segments, especially for the user interface, etc. Therefore, we should go for a program written for a compiler/interpreter available on most platforms, hiding possibly all differences, or at least a program running under the most common operating systems. Moreover, there is a clear need for easy and direct interfacing to the instrumentation, too. Candidates for these purposes are:

- Matlab [8]
- LabView [9]
- LabWindows [10]
- HP VEE [11]
- VisiDAQ, GeniDAQ [12]
- MATRIXx [13]
- Scilab [14]
- O-Matrix [20]
- Octave (GNU) [21]

The first implementation of the tools was written in LabView (see below). However, when comparing the above programs, MATLAB seems to be the most widely used one, available for many platforms. Therefore, it is reasonable to implement the data processing methods described in [2] also in MATLAB. This has the following advantages:

- MATLAB is available at many labs, for many platforms;
- its numerical software is rather platform-independent;
- its interpreter allows transparent coding (the source code can be examined by the user);
- it is open (it is rather easy to extend the existing code).

Therefore, our goal was to develop a MATLAB program which is

- compatible with the LabView program [6,7];
- freely available [19];
- is carefully documented to assure repeatability of the results by other programs, and assure that it can be used as a reference for comparisons;
- can be easily extended with new code (to check new ideas).

Before discussing the MATLAB program, let us briefly overview the LabView one.

5 THE LABVIEW PROGRAM

The LabView program was originally developed to test waveform recorders, but as the two standards use the same algorithm for sine wave fitting [1, 4.1.3.3] [2, 4.1.4.3], it can be used also to test Analog-to-Digital Converters. It implements the four-parameter sine wave fitting, and contains some additional possibilities for visual evaluation (e.g. PSDF, mod T, see below).

The input data to the program are:

[3] SIgnal to Noise And Distortion ratio
[4] Total Harmonic Distortion

- the measured waveform record (sampled points, or sampling instants and sampled points),
- the sampling frequency (or the sampling instants as above), and
- the full-scale (min to max) value of the ADC.

Additional processing parameters (such as time limits to select the part of the record to be processed) can also be set manually. There is even a possibility to process more than one file, but the user has to make a descriptor for each file and scanning of the files is not automatic. To set all of these properties, there is an input window (VI), which automatically opens when the user starts the program (Fig. 1).

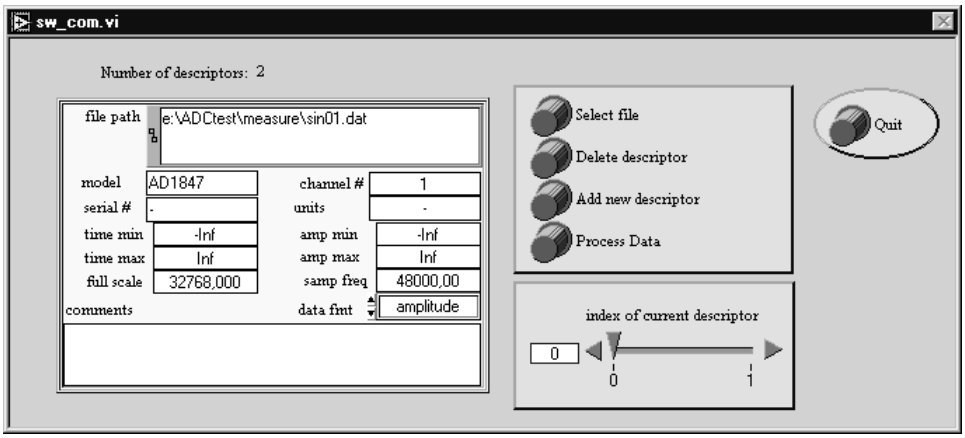

Figure 1: The input window of the LabView program

When pressing the "Process Data" button, the software fits a sine wave to the measured data, using 4-parameter LS fitting, and produces a single-window graphical output, along with two computer files which can be used for later analysis. One example is shown in Fig. 2. The graphical display shows information on the input data vector ("descriptive information"), some single-number outputs ("computed information"), calculated from the fitted sine wave, such as the estimated amplitude and frequency of the signals or the number of effective bits, and displays three different graphs.

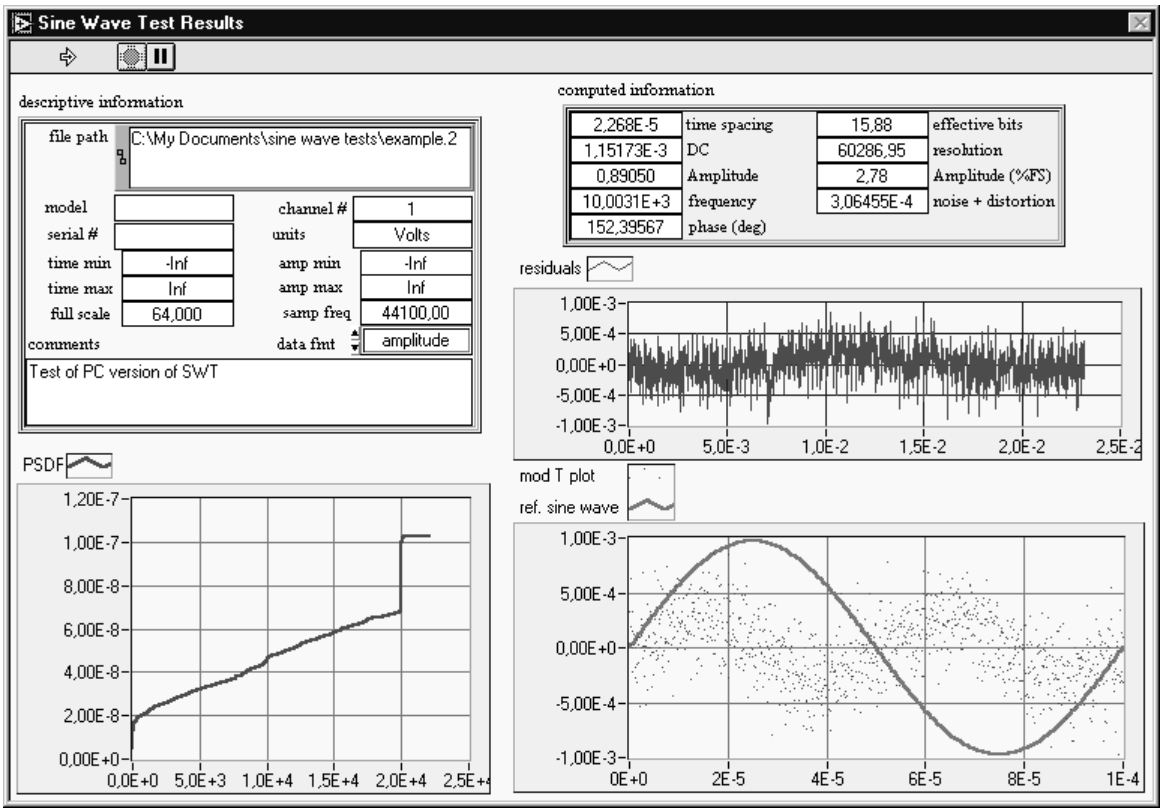

Figure 2: The graphical output of the LabView program

The graph "residuals" shows the difference between the input data and the fitted sine wave. With a usual set of data (more than 1000 samples) in such a small area, it gives rough information only.

The graph "mod T plot" gives more information about the behavior of the residuals; it shows them as a function of the fitted sine wave phase instead of the time (the horizontal axis is wrapped by the period length). The plotted sine wave is a help only to illustrate the value of the phase. Studying this type of plot, it is very easy to determine that the measured device has 2nd harmonic distortion. There are several other properties which can be read from this modulo plot, for more information see [15].

The last graph, labeled "PSDF" shows the power spectral distribution function, the integral of the power spectral density. It gives a lot of useful qualitative and quantitative information about the residuals, such as time jitter and 2nd harmonic distortion [16].

6 THE MATLAB REALIZATION

Our first aim was to develop a MATLAB program which is graphically and algorithmically equivalent to the existing LabView software. The advantage of such a program is that both MATLAB and LabView users can use exactly the same algorithms, obtain the same results, and talk the same language when discussing ADC testing.

A further aim was to define and implement some additional features which can improve usefulness and interactivity of the software package. There are two main concept of the program; first is that while the computing procedures described in the standard are fixed, th e other functionalities (input reading, output writing, logging, etc.) can be customized by the user; second is to ensure a comparative environment for new ideas.

In the following subsections we discuss the main parts of the existing MATLAB software, along with the possibilities for extension.

6.1 Input Window of the Program

By default, the starting window is similar to the previously described LabView one (see Fig. 1 and Fig. 3). However, this is the point where we felt the strongest need for extensions. Depending on the future possibility to extend the LabView program hand-in-hand with the MATLAB one, we suggest the following, either by default, or as an "advanced mode".

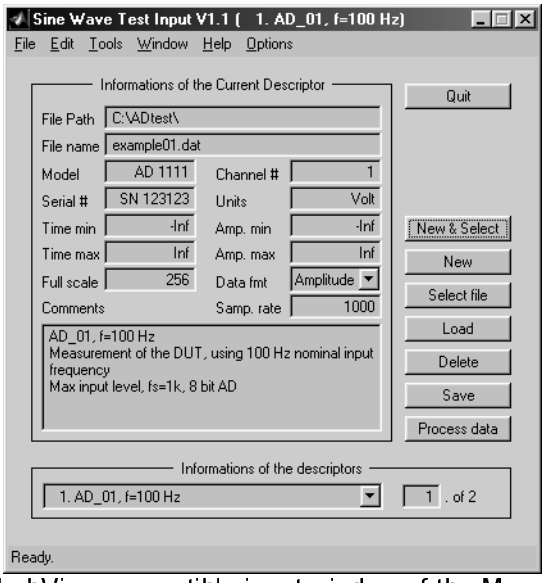

Figure 3: The LabView compatible input window of the MATLAB program

If the user would like to process a lot of measured data from the same type device (e. g. quality control), it is very inconvenient to fill out one descriptor for each file (or just change the file name in each new descriptor). To make the program more efficient, both multiple file opening and wildcards using (e. g. '*' or '?') should be enabled.

Another delicate matter is the input file format. The more the number of the measuring devices/software versions, the more the different file formats will be. Therefore, the "usability" of this program considerably depends of the input file format handling. Therefore we suggest offering three different formats.

The first one is the compatible format, where only the samples or the samples and the time data are in a tab-separated input ASCII file. We allow fixed-point, floating point and hexadecimal format of

the numbers. This format is also suitable to process automatically uploaded and saved memory fields from the DUT without editing the source files manually.

The second choice is to use a precisely defined but flexible input file format. It is very useful when the user would like to process a lot of files with different settings (e. g. to compare a lot of different ADC's or the same ADC with different sampling rates). In this prog ram we have already realized the following experimental structure:

- There is an input data file (which contains the – possibly automatically – collected data) and a description file, which contains the environment parameters of the measurement
- The description file may contain a special comment delimiter signs (default is %). These denote comments: to the end-of-line the text is discarded from the numerical point of view.
- There are fixed keywords which explain the preceding numerical string: device name, fs or sampling frequency, full-scale value, frequency, etc. Most of these (along with the corresponding numbers) are optional and customizable.
- The comment delimiter and the keywords are in a separate ASCII file which can optionally be changed by the user.
- A working example (see the same fields in Fig. 3):

```
% Name          AD_01, f=100 Hz

% Comments
% Measurement of the DUT, using 100 Hz nominal input frequency
% Max input level, fs=1k, 8 bit AD

% PathName      C:\users\markus\matlab
% FileName      example01.dat
% OutPathName   Out

% Model         AD 1111
% Serial        SN 123123

% TimeMin       -Inf
% TimeMax       +Inf
% AmpMin        -Inf
% AmpMax        +Inf

% FullScale     256

% Channel       1
% Units         Volt
% DataFormat    1
% SampleRate    1000
```

Note: first we planned to write the description information into the data-file, but it does not support automatic generation and processing of test-data.

The third solution planned in the program is the most flexible one: user-defined input read function. The user can write a MATLAB-function which opens the file, and returns with the necessary parameters (amplitude vector, time vector and full scale value of the ADC). Using this setting it is very easy to process e.g. standard wave files on a PC.

After obtaining the whole amplitude vector, the user should have the possibility to choose from the samples. Both the time limit and the amplitude limit can be set manually. The program also allows choosing from the samples graphically, using the mouse. In addition, there are also two auto-choose modes: "Auto-choose N periods from the begin/middle/end of the file" and "Auto-choose N samples from the begin/middle/end of the file". Both of them facilitate automatic processing.

Before processing, it is very useful if the user can set the output file parameters. Similarly to the input, three parameters can be set: the output file path and name, the output file format and the output file content. To support full-automatic processing, the output path can be set by the user for every output file, and the output file names can be derived from the input files' names with another extension. The user may choose from three different file formats (similarly to the input file): ASCII file using % signs and comments to explain the preceding string or number, a standard MATLAB mat-file, or file generated by a user-defined function.

The four parameters of the fitted sine wave, the sampling frequency and the vector of the residuals describe all information which is needed to reproduce the result. The rest is optional: saving of most of

the descriptive and computed information (including the residuals and the calculated PSDF) can be enabled or disabled by the user.

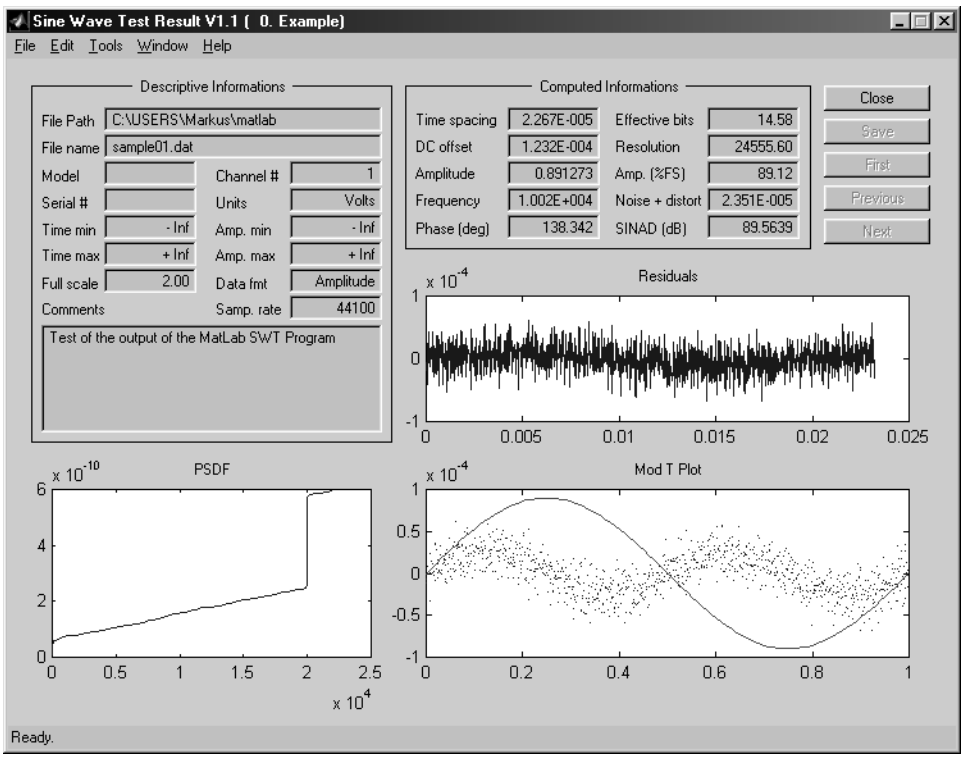

Figure 4: The output window of the MATLAB program

6.2 Computational features

After setting the whole "administrative information", all input files are processed. The program uses the 4-parameter LS sine wave fitting using matrix operations as described in the standard. For greater numerical stability, the calculations use singular value decomposition. To get the same result, the program uses the same initial frequency setting as the LabView one [18].

The calculation of the computed information follows the standard.

A further aim would be to cover the whole standard with just one program. It means that beyond the sine wave test a lot of other testing method should be realized. Some example:

- Measuring transfer curve and computing linearity parameters (incl. static gain and offset, integral nonlinearity, differential nonlinearity, missing codes, etc.)
- Measure and calculate step response parameters
- Measure and calculate frequency response parameters

Another useful development is to extend the program with another advanced mode which could be the international test-bed for new ideas. Using the standard and the new test methods in the same environment causes faster convergence to use better and more exact methods in the ADC -testing. Methods under development are:

- Windowed processing for incoherent sampling [4],
- SYDAC[5], a better model for ADCs, including time-base and nonlinear distortion [22].

6.3 The output window

The main features of the graphical output are the same again as those of the LabView program. An example output window of the MATLAB program can be seen in Figure 4.

One additional feature is that the blocks of the output window can be turned off, if the user would not like to see some of them. Additionally, the buttons on the upright corner help to change from one processed file to another (next, previous files).

For comparison, more than one output window can be opened, and the plots (residuals, Mod T Plot, PSDF) can be zoomed in.

[5] System Identification for Data Acquisition Characterization

7 RESULTS

Implementation aspects of two IEEE standards have been considered. The necessity of a common program for use and reference has been demonstrated. A MATLAB program, compatible to the existing LabView one, has been developed, and suggestions for the extension have been made. The program is free, and is available through the Internet [19].

8 ACKNOWLEDGMENT

The authors are grateful to Jerry Blair for his very useful comments and advice on the LabView program and on the program specifications.

This work has been supported by the Hungarian Fund for Scientific Research, under contract No. OTKA T 033 053.

REFERENCES

[1] IEEE Standard 1057, "*IEEE Standard for Digitizing Waveform Recorders*", Dec. 1994.
[2] IEEE Draft Standard 1241, "*IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*", May 1999.
[3] IMEKO TC4 Working Group on "A/D and D/A Converter Metrology" and EUPAS (European Project for ADC-based devices Standardisation)
URL: *http://elve.le.ttu.ee/MESEL_www_home/R&D/ADC/EUPAS.htm, http://remlab.dis.unina.it/Vpages/ADCWG.htm*
[4] I. Kollár, "Evaluation of Sine Wave Tests of ADC's from Windowed Data" *4th IMEKO TC4 Workshop on ADC Modeling and Testing*, Sep. 9–10, 1999, Bordeaux, France. pp. 64-68. Submitted for publication in *Computer Standards & Interfaces.*
[5] DYNAD, URL: *http://www.fe.up.pt/~hsm/dynad/*
[6] J. J. Blair, "Sine-fitting Software for IEEE Standards 1057 and 1241,", *Proc. IEEE Instrumentation and Measurement Technology Conference, IMTC/1999*, Venice, May 24-26, 1999. pp. 1504-6.
[7] STD 1241 Home Page URL: *http://grouper.ieee.org/groups/1241/,*
STD 1057 Home Page URL: *http://grouper.ieee.org/groups/1057/*
[8] MATLAB, URL: *http://www.mathworks.com/*
[9] LabView, URL: *http://www.ni.com/labview/*
[10] LabWindows, URL: *http://www.ni.com/cvi/*
[11] HP VEE, URL: *http://www.tm.agilent.com/tmo/Products/English/HPVEE.html*
[12] VisiDAQ, GeniDAQ, URL: *http://www.advantech.com/products/automation_software/visidaq.htm, http://www.advantech.com/products/automation_software/GeniDAQ.htm*
[13] MATRIXx, URL: *http://www.windriver.com/products/html/matrixx.html*
[14] Scilab, URL: *http://www-rocq.inria.fr/scilab/scilab.html*
[15] F. H. Irons, D. M. Hummels, "The Modulo Time Plot – A Useful Data Acquisition Diagnostic Tool", *IEEE Trans. on Instrumentation and Measurement*, Vol. 45, No 3, pp. 734-738. June 1996.
[16] J. J. Blair, "A Method for Characterizing Waveform Recorder Errors Using the Power Spectral Distribution" *IEEE Trans. on Instrumentation and Meas.*, Vol. 41, No 5, pp. 604-10. Oct. 1992.
[17] B. Vargha, "3- and 4-parameter fitting of a sine wave as described in the 1241 IEEE Standard" *MATLAB-based program*, BUTE, Dept. of Measurement and Information Syst., Budapest, 1998.
[18] N. Giaquinto and A. Trotta, "Fast and accurate ADC testing via an enhanced sine wave fitting algorithm", *IEEE Trans. on Instrumentation and Measurement*, Vol. 46, pp. 1020-25. Aug. 1997.
[19] "*ADC Test Data Evaluation Program for Matlab*" Home Page, http://www.mit.bme.hu/services/ieee/ADC-test/
[20] O-Matrix: http://www.omatrix.com/
[21] Octave (GNU): http://www.che.wisc.edu/octave/
[22] G. Vandersteen, Y. Rolain, J. Schoukens, "System Identification for Data Acquisition Characterization", Proc. of *IEEE Instrumentation and Measurement Technology Conference,* May 18–21, 1998, St. Paul, USA, pp. 1211–1216.

AUTHORS: Prof. István KOLLÁR, PhD, and János MÁRKUS, PhD student, Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Műegyetem rkp. 9. Hungary, H-1521, Phone: +36 1 463-1774, Fax +36 1 463-4112, E-mail: kollar@mit.bme.hu, markus@mit.bme.hu, URL: http://www.mit.bme.hu/~kollar/, .../~markus/