FREE ACCESSIBLE WEB-BASED PROGRAMS "SIMULATION AND ELIMINATION OF INSTRUMENT DISTORTION" FOR EDUCATIONAL AND SCIENTIFIC APPLICATIONS

Alexei Soloviev[1] and Lidia Luizova[2]

Abstract *— In radioelectronics, image processing or spectroscopy one often cannot ignore the instrument distortions of data. If the instrument pulse response is known the reconstruction of input signal is possible, but the noises are significant obstacles on this way. Computer training programs for educational and scientific applications are described. The programs are designed for simulation of instrument distortions, elimination of those from simulated and real experimental functions, as well as demonstration of noise influence and opportunities of noise effect decreasing. The programs have been successfully used in courses on metrology, spectroscopy, plasma diagnostic, and image processing. One program, performing simulation and elimination of instrument distortion for one-dimensional functions, is executed on local workstation and requires Microsoft Windows 95/98/2000. The second one, destined for 2-D function processing, is implemented as web-service and only requires HTML 4.0 compatible browser. The programs and user's guide is accessible on web site dfe3300.karelia.ru.*

Index Terms — computer program, Fourier transform, image processing, instrument distortions, noise, simulating experiment, training, web-based service.

INTRODUCTION

Before computer engineering physicists solved complex problems approximately (for example, by means of processed function description with a few parameters, that is implicit a priory information about expected results). Calculating capabilities of computers cause the illusion of the absence of such information requirement. It seems a problem of any complexity could be solved if there is a principal solution, even it requires to solve 20-40 combined equations or reverse a matrix of 100x100 dimension. Although received results have no physical sense. Then mathematicians formulated the term "improperly posed problem" [1]. They called so the problems of obtaining information from experimental data when there is a strict unambiguous solution but this solution is highly sensible to experimental errors and even limited precision of calculation. So any variation of source data for a fraction of percent causes the result to vary for hundred times that is out of common sense.

The example of improperly posed problem is the elimination of instrument distortion. We deal with instrument distortion while performing dynamic measurement, i.e. when not a single value should be found but some function should (for example, the voltage as the function of time, the spectral intensity as a function of wave length etc).

Future engineers in optics, metrology, electronics, computer science have to understand the essence of this problem, to know how it may be solved and to see the limited nature of computing methods. The objective of this work is to develop computer programs for educational and scientific applications that allows:

- to simulate and graphically demonstrate the instrument distortions,
- to eliminate these distortions from simulated functions as well as from functions, received in real experiments,
- to demonstrate the influence of noise on the result of exclusion of instrument distortions,
- to show the opportunity of noise effect decrease by using a priory information and to demonstrate the limitations of this method.

These programs have an easy and clear interface and were successfully used in educational laboratories in courses on metrology, spectroscopy, plasma diagnostics, and image processing.

MATHEMATICAL BASIS

We consider only the instruments that could be described as invariant linear filters. Almost every design for dynamic measurement corresponds to this model in definite limits of input signals and usually one avoids to work outside of these limits. So the output function is the result of the convolution of the input signal with the instrument spread function.

$$f_{out}(\tau) = \int_{-\infty}^{+\infty} f_{in}(t) \cdot g(\tau - t) \cdot dt \qquad (1)$$

The instrument spread function *g(x)* (also known as *pulse response*, or *spread function*) characterizes the device

[1] Alexei Soloviev, Petrozavodsk State University, Department of IMS & PhE, 33 Lenin st., 185640 Perozavodsk, Russia, tel: +7(8142)711021, e-mail: avsdov@yahoo.com
[2] Lidia Luizova, Prof., Ph. D, Petrozavodsk State University, Department of IMS & PhE , alim@karelia.ru
The research described in this publication was made possible in part by Award No. PZ-013-02 of the U.S. Civilian Research & Development Foundation for the Independent States of the Former Soviet Union (CRDF).

and corresponds to the output signal when the input signal is a single pulse or delta-like signal.

Of course, in real experiments the input and output coordinates are quite different, for example, wave length at the input and the element number of photodetective matrix at the output of spectrometer. But there is always correspondence between input and output coordinates that have been ascertained at the stage of graduation, so that we can use the same variable *x* for input and output coordinates. For simplicity, we consider one-dimension functions, but computation may be easy generalized to 2-D case of image processing.

The process of the elimination of these distortions and the reconstruction of true function is referred as "reduction to ideal instrument". We can lead this problem to solution of integral convolution equation (1). The well known way of solving this equation is using the convolution theorem [2]: Fourier transformed convolution is equal to product of Fourier transformed convolution components.

$$\Phi\{f_{out}\} = \Phi\{f_{in}\} \cdot \Phi\{g\} \qquad (2)$$

Here Φ is the Fourier transform operator. Obviously, we only have to divide Fourier transform of the output signal to Fourier transform of the spread function and restore the input signal by inverse Fourier transform of the result. In case of noise present we deal with sum of "true" signal and noise.

$$\Phi\{f_{rest}\} = \frac{\Phi\{f_{out}\} + \Phi\{\xi\}}{\Phi\{g\}} = \Phi\{f_{in}\} + \frac{\Phi\{\xi\}}{\Phi\{g\}} \qquad (3)$$

Noise's spectrum $\Phi\{\xi\}$ is much wider than spread function's one as a rule. That's why after division of Fourier transformed output function to Fourier transformed spread function, the result of Fourier inversion is a fast oscillating function, so we receive no information about the input signal. Clearly the simplest way of noise effect diminution is to limit the operative spectral range. The narrower this range the less the noise amplitude in reconstructed signal, but as a result resolution of instrument is worsened too. That's why thin details in input signal would be lost by limitation of spectral range. The constriction of operative spectral range is equivalent to using a priory information about smoothing degree of input function.

This property of the process of reduction to ideal instruments is readily illustrated by our programs. During the simulating experiments one can be assured that noise sets limits on information about the investigated signals.

BRIEF DESCRIPTION OF THE PROGRAMS

Simulation and elimination of instrument distortion for one-dimensional functions can be performed by the "Apparat" sotware [3]. This program is executed on local workstation and requires Microsoft Windows 95/98/2000. The second program was designed for 2-D function processing. It is implemented as web-service and receives its calculation parameters from HTML 4.0 compatible browser on any platfom.

"Apparat" Software

This program was developed by means of Borland C++ 5.5 free command line tools. It uses system modules only, so extra DLLs aren't required. This program was designed to simulate output one-dimensional function when input signal and spread function are given and to restore input one-dimensional function when output signal and spread function are given. The user can simulate noise and can study its influence on the quality of restored signal.

The main program window (Figure1) contains the menu, the toolbar, the graph field and the status line. The program uses both English and Russian labels and messages. The user switches the languages by means of corresponding command from the "Options" menu. The most common commands are placed as buttons on the toolbar. The user can determine the exact value of one of functions if he or she chooses it from the list on the toolbar and clicks on the graph of this function. The function value in this point is placed into the status line. The chosen function can be easily scaled by means of the trackbar on the toolbar.

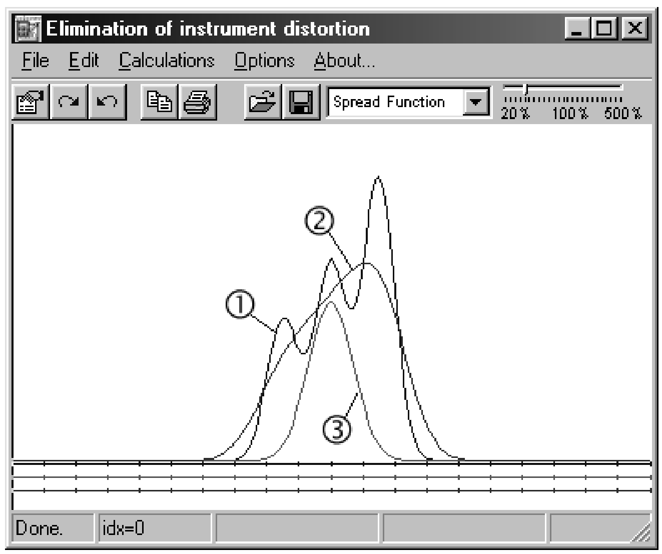


FIGURE. 1
THE MAIN WINDOW OF THE "APPARAT" PROGRAM.
(1 – INPUT SIGNAL, 2 – OUTPUT SIGNAL, 3 – SPREAD FUNCTION)

Various functions charts are drawn with various colors in the graph field. For each chart in the graph field the abscissa axis is drawn with corresponding color. Zero is always located in the center of the graph filed. Each graph presents absolute values of a function. The contents of the graph field can be printed or copied to clipboard.

Calculation starts when the user selects one of the command from the "Calculations" menu. The program places the information about the current step of calculations and its progress into the status line. The user can load input signal, spread function or output signal from file. Fourier-images can't be loaded. They will be calculated. The user can save any function if it is given (or calculated) and visible. The input and output files are texts. The first file line contains the amount of function values, initial value of argument, argument increment and maximal function value. From the second line of the file complex numbers are placed. Real part of a number is separated by space from imaginary part. Except loading from a file, function can be defined as internal one. The "Functions tuning" dialog (Figure 2) allows to configure internal functions, to specify precision and to add noise to a signal. The internal input function is the sum of 3 gaussians. The user should specify for every gaussian the amplitude (A), the abscissa of peak (B) and the C parameter, which determines the width at half-maximum. Spread function can be defined as normalized gaussian (the user specifies the C parameter only and the program recalculates the amplitude to normalize spread function). Fourier transform is performed by means of calculation of integral sum, so there are no restrictions to functions limits.

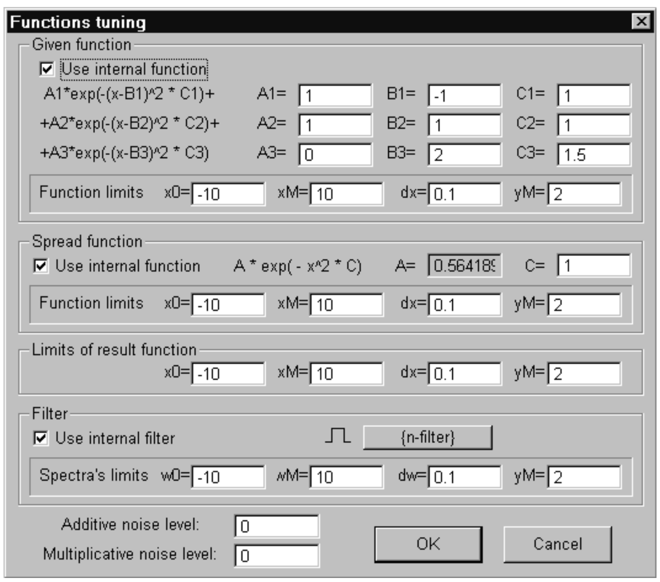


FIGURE. 2
THE FUNCTIONS TUNING DIALOG OF THE "APPARAT" PROGRAM.

The program adds two kinds of noises: additive and multiplicative. The user may specify "noise level" σ in a corresponding field, then the program changes every calculated value f of the output function to $f'=f+\xi f$, when multiplicative noise is chosen, and to $f'=f+\xi Y_{max}$ for the additive one, where ξ is a normally distributed value with zero mean and variance equal to σ^2, Y_{max} – window size, specified in the "yM" field for result function.

For decreasing the noise influence, one can use filtration of Fourier transformed restored function. There are two types of filter selected by the "filter" button:

1) binary filter – in this case the restored function Fourier transform, calculated according to (3), is multiplied by 1 inside of the limits specified in fields "Spectral limits", it is equal to zero outside of this limits;

2) smooth filter – it this case Fourier transform is also multiplied by 1 inside of "Spectral limits", but outside them Fourier transform is multiplied by exponential decreasing value with an index specified in the "C" field.

User can load other specific filter from file as well.

Web-based Service

Web-based service named "Instrument Distortion: Simulation and Elimination" (http://dims.karelia.ru/distort) is a web application, which consists of a set of HTML pages, CGI engine and low-level calculation engine. HTML pages provide a user with graphical interface (Figure 3). CGI engine provides interaction between web-server software (Apache) and low-level calculation engine by means of handling user requests, preparing data for calculations and forming result pictures. It is implemented as a set of platform-independent Perl language scripts. Low-level calculation engine performs required Fourier transforms. It is implemented in C language and compiled for particular platform (Linux).

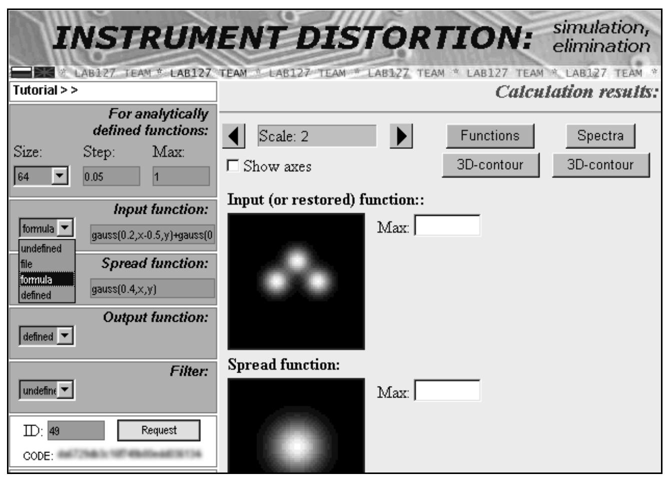


FIGURE. 3
WEB-SERVICE GRAPHICAL USER INTERFACE.

This web-service simulates two-dimensional instrument distortion when the input two-dimensional function and the spread one are defined, or eliminates two-dimensional instrument distortion, when the output two-dimensional function, the spread one and, possibly, the filter are defined. To define a function the user should select the appropriate way of function definition in the drop-down list: "file" or "formula". If user defines a function analytically (by means of formula) he or she should specify the size of the sample, the sample interval and maximal value of the functions. To send data to server when all these fields are filled user presses the "New request" button.

All the requests are registered and numbered. CGI engine stores request information (number, time stamp, client's IP, session code etc.) in MySQL database. Results are stored on the server for 1 hour since the first access. During this time the user can view calculation results: he/she enters its number in the "ID" field and presses "Request" button. User can change some functions of calculated request. Unchanged functions won't be transferred over the network. The low-level calculation engine will use their server copy.

At first CGI engine generates the session code (the "CODE" field). This code expires in one hour. Every time user accesses the server CGI engine prolongs the term of the session code for another hour. After long-time inactivity the other session code is generated. On user's computer the session code is stored in browser cookies. All the registered requests require the session code to be specified. If the specified session code doesn't match the request, the access to data on the server is denied. This feature has a side effect: the user can't access his or her registered requests from the other computer or browser without specifying the correct session code manually. The session code is the "password" for access user's data.

Real function for this service can be specified as formula. The user can compose a formula from arithmetical, logical and bitwise operators of widely used C-like programming languages, from a few internal functions like sine, cosine, gaussian etc. as well:

gauss(σ,x,y) is equivalent to $\exp\left\{-\frac{x^2}{2\sigma^2}-\frac{y^2}{2\sigma^2}\right\}$

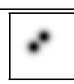
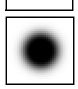
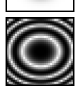
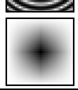
smooth(R,σ,x,y) is equivalent to

$$\begin{cases} gauss(\sigma, x - R\frac{x}{\sqrt{x^2+y^2}}, y - R\frac{y}{\sqrt{x^2+y^2}}), & x^2+y^2 \le R \\ 1, \quad x^2+y^2 > R \end{cases}$$

The latter function is useful for specifying as filter.

Table I contains a few examples of analytically defined functions:

TABLE I
EXAMPLES OF ANALYTICALLY DEFINED FUNCTIONS

gauss(0.2,x+0.5,y)+gauss(0.2,x,y-0.5)	
smooth(0.5,0.3,x,y)	
*cos(2*x*x+3*y*y)*	
(1-abs(x)/1.6)(1-abs(y)/1.6)*	

Data files for this service have the same format as the files for the "Apparat" software except for it is strictly recommended to compress data files when transfering over the network. This web-service can handle archives in ZIP (PkZip, WinZip etc) or GZIP (http://www.gzip.org) format. The calculation results are always compressed by GZIP.

The problem of two-dimensional instrument distortion elimination is usually the interest of image processing. So this web-service supports some image file formats: PNG (Portable Network Graphics) and JPEG (Joint Photography Experts Group). When converting image to real values it is supposed that green color gradation "255" corresponds to 1.0. Red and blue color gradations are ignored. The left bottom corner of the image corresponds to the function value for the least argument values.

SIMULATING EXPERIMENTS IN LABORATORY TRAINING

If student simulates an input function as three clearly separate peaks and a spread function as a gaussian with width comparable with peak separation, the result function has no gap, so by visual observation three lines could not be resolved (Figure1). Restoring input function by the "Apparat" program one can receive almost correct input signal. If a noise is added to the output function an attempt to restore the input signal is failed. Even with low values of noises student may receive senseless results (Figure 4). The noise effect can be decreased by reducing the operative spectral range. (Figure 5.)

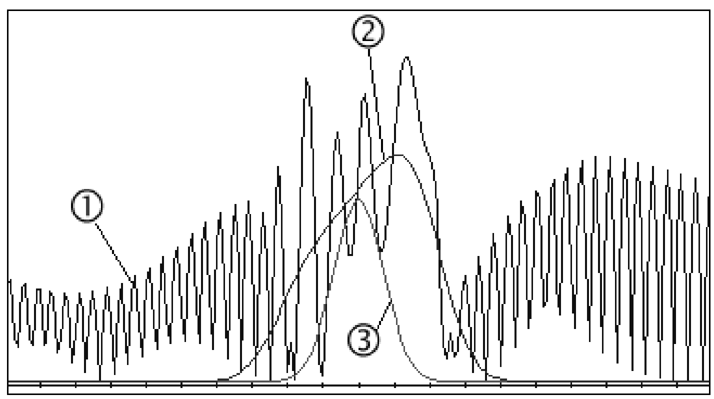


FIGURE. 4
AN ATTEMPT TO ELIMINATE INSTRUMENT DISTORTION.
(1 – RESTORED SIGNAL, 2 – OUTPUT SIGNAL WITH ADDITIVE NOISE σ=0.00001, 3 – SPREAD FUNCTION)

Then student has to investigate the effect of spread function width on the result of reduction to ideal instrument. But if the spread function is wide and the noise level is not low the oscillations disappear with the narrow operative spectral range (Figgure 6), so restored function is now more similar to the output function than to the input one, that is to say the instrument distortion elimination in this case is impossible.

So students have to understand that inevitable noise puts the real limit for optical instruments resolution. The better is the instrument and the narrower is its spread function the lower are the demands to noise level.

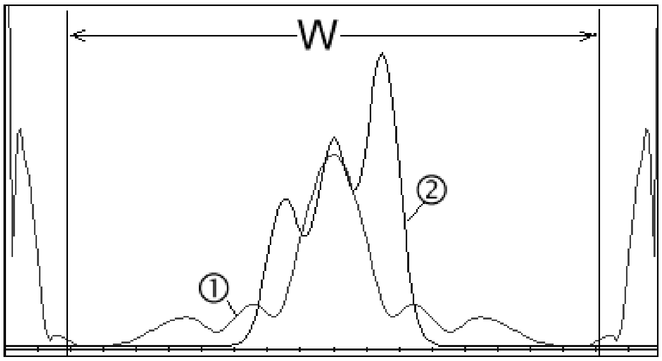


FIGURE. 5
AN ATTEMPT TO ELIMINATE INSTRUMENT DISTORTION
WITH REDUCED SPECTRAL RANGE W.
(1 – RESTORED FUNCTION SPECTRUM, 2 – RESTORED AND "TRUE" INPUT SIGNALS ARE ALMOST IDENTICAL)

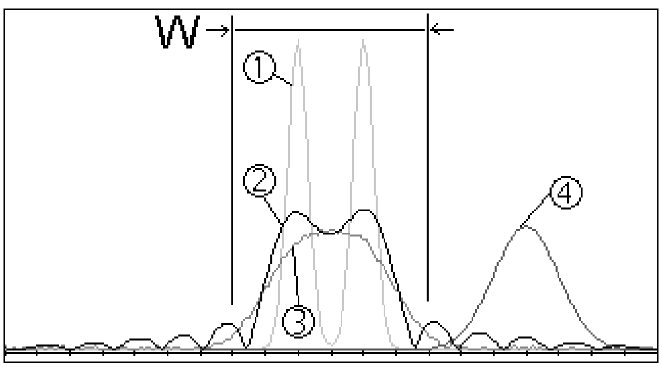


FIGURE. 6
AN ATTEMPT TO ELIMINATE INSTRUMENT DISTORTION
WITH WIDE SPREAD FUNCTION AND REDUCED SPECTRAL RANGE W.
(1 – "TRUE" INPUT SIGNAL, 2 – RESTORED SIGNAL, 3 – OUTPUT SIGNAL WITH ADDITIVE NOISE σ=0.005, 4 – SPREAD FUNCTION IS SHIFTED FROM "NORMAL" PLACE FOR CLEARNESS)

Even if specified noise level is zero, the limited accuracy of computer calculation may have an effect like noises. This effect appears to be the major problem for elimination of 2-D instrument distortion.

If student simulates 2-D instrument distortion by means of the web-service (Figure 7), restoring of input 2-D functions is always a nontrivial problem. Even without special noise simulation, most likely, attempts to restore input signal would failed because of limited accuracy. In most cases reduction of spectral ranges required (by means of filters).

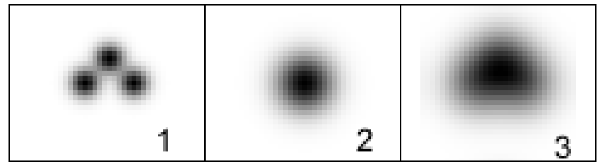


FIGURE. 7
SIMULATION OF 2-D INSTRUMENT DISTORTION.
(1 – INPUT SIGNAL, 2- SPREAD FUNCTION, 3 – OUTPUT SIGNAL)

If student compares various results of instrument distortion elimination when output functions are determined in various ways, he or she will notice that text data files allows to achieve better results than image files (Figure 8 and 9). For text data files elimination of instrument distortion is possible with wider filters. Student should explain this by less accuracy of image data files.

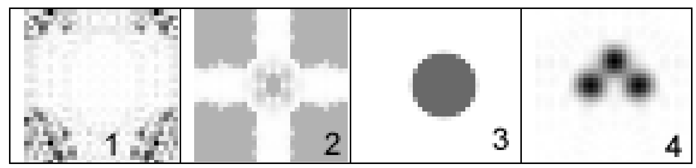


FIGURE. 8
ELIMINATION OF 2-D INSTRUMENT DISTORTION
WHEN OUTPUT FUNCTION FROM FIGURE 7 IS LOADED AS TEXT DATA FILE.
(1 – FIRST ATTEMPT TO RESTORE SIGNAL, 2- RESTORED SIGNAL SPECTRUM, 3 – APPLIED FILTER, 4 – RESTORED SIGNAL AFTER FILTRATION)

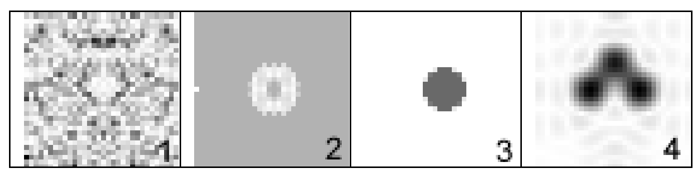


FIGURE. 9
ELIMINATION OF 2-D INSTRUMENT DISTORTION
WHEN OUTPUT FUNCTION FROM FIGURE 7 IS LOADED AS IMAGE FILE.
(1 – FIRST ATTEMPT TO RESTORE SIGNAL, 2- RESTORED SIGNAL SPECTRUM, 3 – APPLIED FILTER, 4 – RESTORED SIGNAL AFTER FILTRATION)

REFERENCES

[1] Tichonov A. N., Arsenin V. Ya, "Methods of solving of improperly posed problems", Moscow, 1979, pp 15-18.

[2] Smirnov V.I. , "Course of High Mathematics" Vol 4, Moscow, 1979, p.158 15-18.

[3] Luizova L.A., Soloviev A. V. "Computer training program for elimination of instrument distortions", *Proc. SPIE.* Vol. 4588, 2002, pp.440-447.