

A Web-based Data Generator for Software-Validation and Algorithm Comparison in Primary Accelerometer Calibration

Th. Bruns¹, C. S.Veldman²

¹Physikalisch-Technische Bundesanstalt, Germany, thomas.bruns@ptb.de

²National Metrology Institute of South Africa, South Africa

Abstract

During the 18th IMEKO World Congress 2006 in Rio de Janeiro the members of the TC22 discussed the options to validate and compare different algorithms used worldwide by NMIs in the field of primary accelerometer calibration. The goal would be to validate calibration results calculated with the help of reference data sets and compare the different algorithms with data sets including well defined disturbances. This would expose the susceptibility or immunity of the algorithms with respect to signal disturbances.

The challenge in such an approach is the multitude of measurement set-ups available in the NMIs and the respective diversity of parameters and data sources. In order to cope with this diversity the approach of a web-based data generator will be introduced. The aim is to provide a browser-based interface for a data synthesis software.

Details about the incorporated calibration set-ups, the considered disturbances, the mode of calculation and the internet technology used for deployment will be discussed in the presentation.

1. Introduction

The performance of key comparisons based on actual measurements of a physical quantity is a standard business for metrology institutes nowadays. A major component in the applied instrumentation is software for the data analysis and correction of known systematic deviations. The implemented software is usually well tested by each individual laboratory under different conditions and with various means. However the result of key comparisons is always a result from the combined effort of an optimized measurement setup and optimized software. In order to further improve the quality of measurements on an international scale, it appears to be a sensible approach to split up this combined exercise and start the comparison work at the software level.

The following sections describe the effort to build up a transparent framework for the comparison of data analysis software in the field of primary vibration calibration according to ISO 16063-11 [1]. This comparison is based on computed datasets based on the virtual motion of a simulated calibration facility. The data sets are ideal in the sense that they represent exactly the signals as they are defined by the simulation. However, they are quite real too, in a sense that the signals include many of the well-known and disrespected disturbing components usually encountered with real calibration facilities.

Besides the pure data simulation exercise for the various measurement systems complying with the written standard, the task comprises also the deployment of the respective data sets to the participants of the comparison

2. Aim, Task and Boundary Conditions

The basic intention of the described work is to provide the means for comparing the data analysis software for primary calibration according to [1] as it is applied in various different implementations throughout the metrology institutes worldwide. This comparison should be accomplished on the basis of comparable data sets. Due to the different implementations of measurement systems and of the respective software it is not feasible to use identical data sets, i.e. the data sets can only be computed from the same definition of motion including the definition of disturbing components [2]. The definition itself is not available to the participant but only to the pilot laboratory of the comparison. This is equivalent to the circulation of an unknown artefact in real key comparisons. The participant will receive a set of data representing the output voltage of a virtual artefact and the signal from a laser interferometer in heterodyne- or homodyne quadrature configuration.

The principle of the comparison is similar to the usual software quality benchmarking performed individually in every laboratory. The only difference is that the data sets are computed with a common, centralized tool.

As the notion of quality benchmarking is so close to the generic purpose of the software, it was decided that it should be built to suit both purposes. The tool can therefore be used as a general generator of data sets for comparison as well as for benchmarking and improving data analysis software in the future.

To make it most usable for other persons and trustworthy for participants of comparisons, it was further decided to develop the code under the auspices of an open source license. Thus, anyone can download, use, distribute and/or modify the code. This even gives rise to an international effort to further improve the code and adapt it to make it suitable for more measurement systems. The different aspects to be considered during the work are summarized as follows:

- data set generation according to ISO 16063-11 (method 3)
- hidden inclusion of sensitivity and disturbing components for the purpose of comparisons
- open arbitrary definition of sensitivity and disturbing components for the purpose of benchmarking
- open availability of the source code for further optimization and adoption to new systems
- deployment of (huge) data sets from a centralized software
- secure transfer of dataset definitions to the pilot laboratory of a comparison for later evaluation of the comparison results.

3. General Concept of Implementation

The general considerations described in the preceding section resulted in a number of consequences for the realization of the work.

In order to make the software easily available to the interested public and

use the community effect of open source projects the whole data generator was designed as a web-browser based application. To achieve the quality of platform independence together with an easily usable graphical user interface (GUI) the language JAVATM was selected for the development of the source code.

However, the general term “web based application” does not distinguish whether the computational burden is carried by the client or the server. Due to the computational burden and large size of the data files produced by the data generator, a client based solution is preferred. Therefore the solution in the form of a “JAVA Applet“ was taken.

Such an applet is a piece of compiled software that is embedded in a web page and can usually be executed in any modern web browser. But there are two properties inherent to applets which are in this case contra productive. Usually applets are executed in a so called sand box, i.e. they are isolated from the local computer environment for security reasons and are therefore not allowed to access the local file system or the local network connections. These restrictions, however, can be circumvented by signing the applet, thereby marking the applet trustworthy to the browser and operation system where they are run.

A second benefit of signing of the applet is that the software cannot be manipulated without invalidating the signature, making it is possible to distribute the source openly on the one hand and simultaneously perform the intended comparison with essential data hidden from the participant on the other hand. This is required because the participants have to use the already compiled and signed applet prepared for the comparison without being able to prepare the source code themselves. Thus the applet for the comparison will be executed in the local browser of the participant's computer. The generated data files for the evaluation of the data analysis software can be stored on the same computer locally without any network transmission and the parameters defining the virtual motion, artefact and included disturbing components will be hidden and send (encrypted) to the pilot laboratory for the analysis of the comparison later.

4. Realization of the Data Generator

4.1 Parameters and Inputs

The metrology institutes volunteering in this pilot study (up to now NMISA, PTB, NIM, INMETRO) of a software comparison have implemented quite different variations of method 3 of [1], including homodyne quadrature and heterodyne setups. In addition to these different laser interferometric methods, most of the other typical measurement parameters such as the sampling rates, digital converter resolution, carrier frequency and many more are individually different. It was therefore necessary to provide inputs in the GUI to define the individual parameters fitting the physical measurement setup in the laboratory. In order to define the necessary inputs as well as the included disturbing components a technical protocol (TP) [2] was compiled and agreed upon by the participating partners.

The input parameters for the comparison were provided as:

- Motion quantities
 - nominal excitation frequency
 - nominal sinusoidal amplitude of acceleration
- Transducer quantities
 - nominal sensitivity
 - sampling rate
 - measurement duration
 - ADC resolution
 - ADC voltage range
- Laser interferometer quantities
 - sampling rate
 - measurement duration
 - ADC voltage range
 - ADC resolution
 - signal voltage amplitude
 - carrier frequency for heterodyne method

In order to complete the definition of the respective simulation, several quantities are defined hidden from the user. These quantities are available via the GUI if the check box for comparison was not marked (c.f. Fig 1). These quantities are within a certain range, calculated by using a random number generator. The quantities are:

- Motion quantities
 - initial phase of motion ($0 \leq \varphi_0 \leq 2\pi$)
 - first and second disturbing motion component (building vibration, hum)
 - frequency of the components
 - amplitude of the components
 - random noise on the acceleration
- Transducer quantities
 - resonant frequency
 - voltage offset
 - total harmonic distortion (THD)
 - random noise
 - ADC jitter
- Laser interferometer quantities
 - initial optical phase
 - offset voltage
 - random noise
 - ADC jitter
 - for homodyne version
 - voltage gain mismatch
 - quadrature mismatch
 - voltage offset mismatch

With the exception of the initial phase, all hidden parameters can be determined by the user if the data generator is used in the general benchmarking mode, i.e. not for comparison.

The GUI of the Java® applet is divided in different tabs representing the different parts of a virtual accelerometer calibration facility. In the first tab called “General” (c.f. Fig 1) some administrative data are requested, for instance the contact data of the participant if in comparison mode and the directory where the simulation data is to be saved on the local computer.

The subsequent tabs are described together with the implementation details in the following sections.

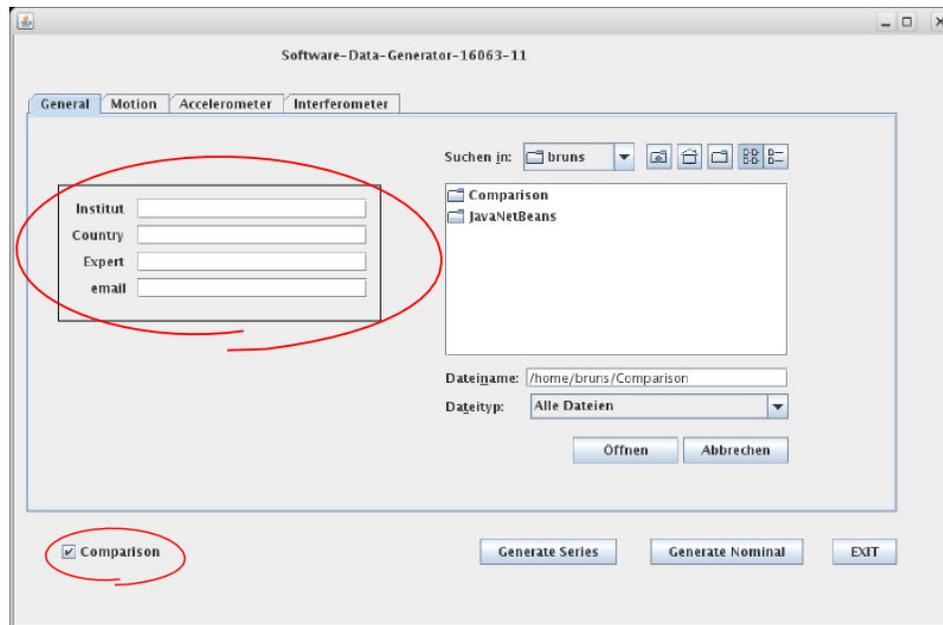


Figure 1: General settings page of the GUI of the data generator. Items for comparison mode are circled.

4.2 Motion Data Generation

The tab “Motion” (c.f. Fig. 2) queries the parameters describing the nominal motion in terms of the frequency and acceleration amplitude. In benchmarking mode two additional motion components can be defined with their corresponding parameters in order to incorporate such disturbances like building vibration and hum. In comparison mode these “1st and 2nd disturbing motion component” fields are hidden. The respective components, however, are still included in the simulated motion.

In addition to the pure periodic definition an auxiliary field for mechanical noise is provided which is again hidden but included in comparison mode

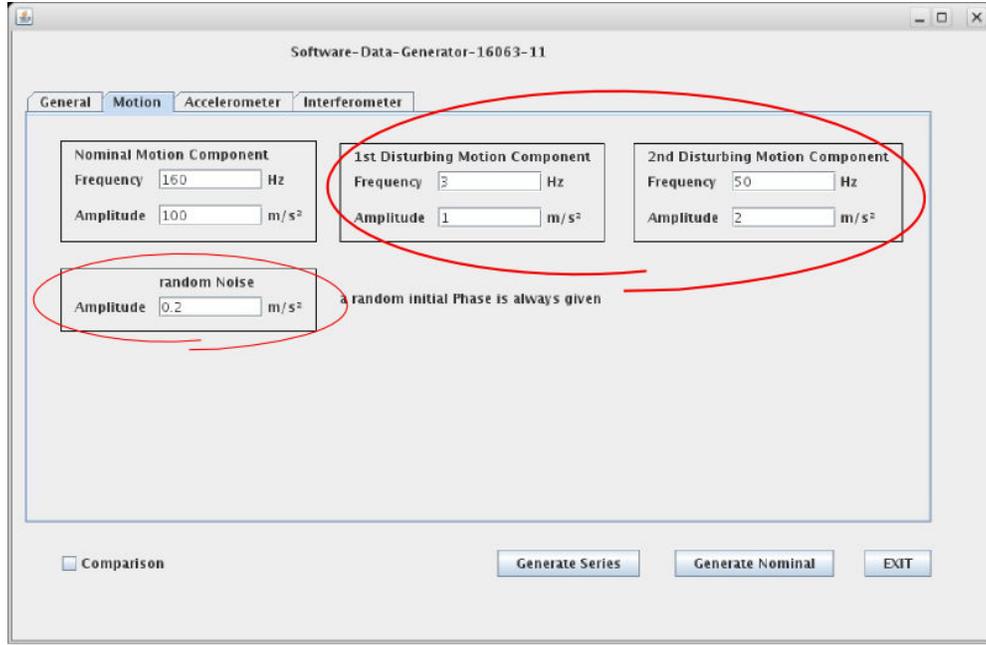


Figure 2: Motion settings page of the GUI of the data generator, defining the motion components of a virtual vibration exciter. Items hidden in comparison mode are circled

From the given parameters the motion is calculated according to the equation

$$\mathbf{a}(t) = \sum_{i=1}^3 \hat{\mathbf{a}}_i \cdot \sin(2\pi f_i t - \varphi_i - \varphi(f)) + \mathbf{a}_{\text{noise}}(t) \quad (1)$$

if acceleration is the necessary quantity or according to

$$\mathbf{v}(t) = \sum_{i=1}^3 \frac{\hat{\mathbf{a}}_i}{\omega_i} \cdot \sin(2\pi f_i t - \varphi_i + \frac{\pi}{2} - \varphi(f)) + \mathbf{v}_{\text{noise}}(t) \quad (2)$$

or

$$\mathbf{x}(t) = \sum_{i=1}^3 \frac{\hat{\mathbf{a}}_i}{\omega_i^2} \cdot \sin(2\pi f_i t - \varphi_i + \pi - \varphi(f)) + \mathbf{x}_{\text{noise}}(t) \quad (3)$$

if velocity or displacement are required, respectively.

The source code is prepared to cope with even more disturbing components and for the case that the input amplitudes might be given in units of velocity or displacement. This particular version of the applet with it's GUI does not provide for these options.

The above given formulas are implemented in methods of the object called "MotionGenerator". This object is used as an argument to the methods that calculate the response of the accelerometer and interferometer. This approach ensures that "Interferometer" and "Accelerometer" work on the same motion simulation which is represented by the "MotionGenerator" object.

4.3 Transducer Output Calculation

The parameters describing the virtual accelerometer are queried in the respective tab named "Accelerometer" (c.f. Fig. 3). Here not only the technical

data of the sensor but also of the respective parameters of the data acquisition channel are represented. In comparison mode, only the acquisition channel can be configured. The transducer's data is hidden

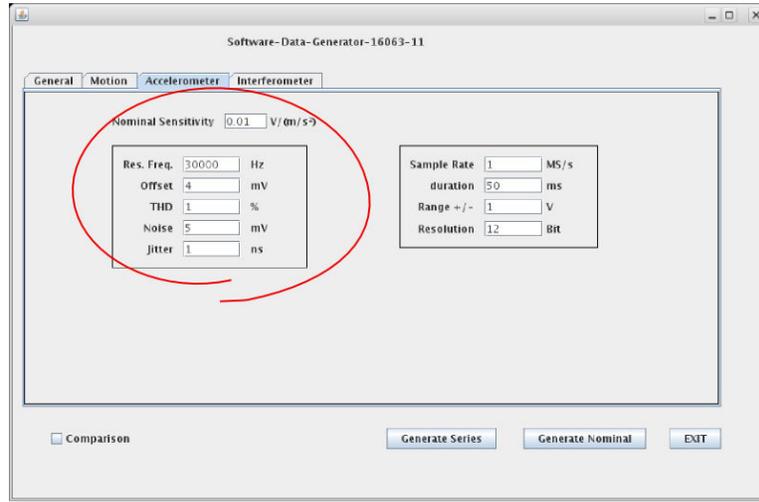


Figure 3: Accelerometer settings page of the GUI of the data generator, defining the virtual artefact for comparison and the respective measurement channel. Items hidden in comparison mode are circled

A core component of the implementation of the accelerometer is a method which calculates the sensitivity of the accelerometer with respect to the actual frequency. This is done according to the formula describing a typical resonant curve

$$S(f) = S_0 \frac{f_0^2}{\sqrt{(f^2 - f_0^2)^2 + \frac{f_0^2}{q^2} f^2}} \quad (4)$$

$$\varphi(f) = \tan^{-1} \left(\frac{f_0 f}{q(f_0^2 - f^2)} \right) \quad (5)$$

where S_0 is the nominal sensitivity from the flat low frequency part of the magnitude response curve and f_0 is the resonant frequency. Both parameters are either given via the GUI input or defined by the random number generator for the comparison case.

In order to cover the superposition of the different motion components (see former section) an average sensitivity is calculated as the weighted mean of the sensitivities for the superimposed frequencies. The weights are the different amplitudes of the motion components. The sensitivity finally applied in the simulation is therefore given as

$$\bar{S} = \frac{\sum_{i=1}^3 S(f_i) \hat{a}_i}{\sum_{i=1}^3 \hat{a}_i} \quad (6)$$

With (6) the output voltage time series is calculated as

$$U(t_i) = \bar{S} \cdot a(t_i) + U_{off} + U_{noise}(t_i) \quad (7)$$

In addition to the different (already included) motion components, the implementation provides for a certain THD of the signal, too. This is achieved by transforming the output voltage with a non-linear characteristic curve of the type:

$$y(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \quad (8)$$

In order to keep the general amplitude characteristics and offset, the following choices for c_i are taken:

$$\begin{aligned} c_0 &= 0 \\ c_1 &= 1 \\ c_2 &= k \cdot c_3 \end{aligned} \quad (9)$$

With a given THD generated by second and third harmonic of an input vibration of magnitude a , the following equations hold:

$$\tau = \sqrt{\frac{P_2 + P_3}{P_1}} = a \cdot c_2 \frac{\sqrt{4 + \left(\frac{a}{k}\right)^2}}{4 + 3a^2 \frac{c_2}{k}} \quad (10)$$

resolving for c_2 leads to

$$c_2 = -\frac{4\tau}{3\tau \frac{a^2}{k} - a \sqrt{4 + \left(\frac{a}{k}\right)^2}} \quad (11)$$

and

$$c_3 = \frac{c_2}{k} \quad (12)$$

Following this, distorted signal can be simulated by applying the non-linear characteristic curve to $U(t_i)$ i.e.

$$U_{Dist}(t_i) = U(t_i) + c_2 U^2(t_i) + c_3 U^3(t_i) \quad (1)$$

4.4 Interferometer Output calculation

For the simulation of the interferometric measurement, two different cases have to be distinguished. I.e. the homodyne quadrature setup with two measurement channels in optical phase quadrature or the heterodyne setup with one measurement channel and a carrier frequency. It is assumed that all

channels have common values for the sample rate, duration, range, resolution and signal amplitude parameters (c.f. Fig. 4).

Disturbances included here are offset voltage, noise and jitter on all channels. In addition for the homodyne case mismatches in gain, offset voltage and quadrature phase are provided for.

The core information used for the calculation in the “Interferometer” is the displacement time series generated by the “MotionGenerator” object according to Eq. (3). The fact that this is the same object which was used for the accelerometer response calculation (c. f. preceding section) ensures that the identical motion is simulated, including the same disturbing components.

Taking the simulated displacement of the shaker armature as the starting point, the interferometer signals and of the homodyne setup are calculated according to:

$$\begin{aligned}
 U^I(t_i) &= U_0^I \cdot \left(1 - \frac{\mu_a}{2}\right) \cdot \sin\left(\frac{4\pi}{\lambda} \cdot x(t_i) - \varphi_0^{optic}\right) + U_{noise}^I(t_i) \\
 U^Q(t_i) &= U_0^Q \cdot \left(1 + \frac{\mu_a}{2}\right) \cdot \cos\left(\frac{4\pi}{\lambda} \cdot x(t_i) - \varphi_0^{optic} + \varphi_0^{mismatch}\right) + U_{noise}^Q(t_i)
 \end{aligned}
 \tag{14}$$

where μ_a is the relative amplitude or gain difference of the two photo detectors (given in % in the GUI), φ_0^{optic} is the optical initial phase of the laser and $\varphi_0^{mismatch}$ is the quadrature phase mismatch i.e. the deviation from 90° phase difference between $U_I(t)$ and $U_Q(t)$.

For a heterodyne system only one channel is calculated with an additional carrier frequency f_c according to:

$$U^{het}(t_i) = U_0^{het} \cdot \sin\left(2\pi f_c t_i + \frac{4\pi}{\lambda} x(t_i) + \varphi_0^{optic}\right) + U_{noise}^{het}(t_i)
 \tag{2}$$

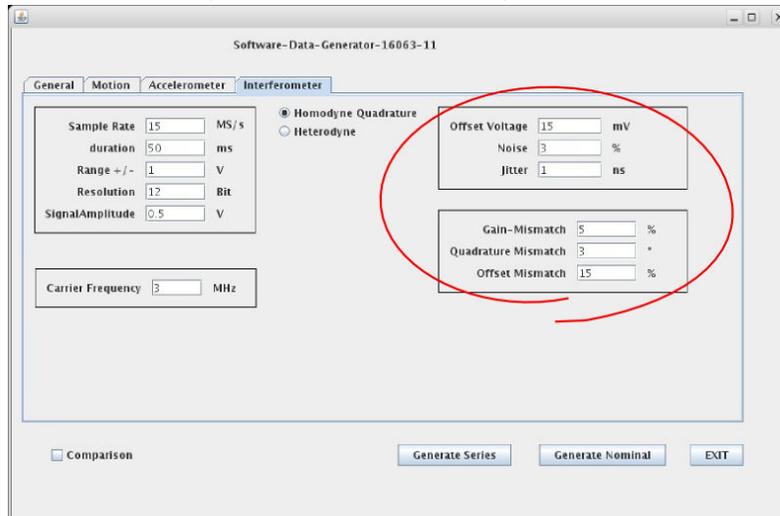


Figure 4: Interferometer settings page of the GUI of the data generator, defining the interferometric method (here homodyne selection), the respective parameters of the measurement channel and typical disturbances according to the selected method. Items hidden in comparison mode are circled.

4.5 Common Properties

All simulated voltage time series have some properties in common which are features encountered in any typical ADC.

For one thing, all voltage measurements suffer from noise which is simulated as an additive term of uniformly distributed random numbers. The upper and lower limits of the distribution are taken as $\pm \frac{n}{2}$ where n is the noise range derived from the GUI input or from hidden data.

The equivalent procedure is used to simulate the occurrence of jitter i.e. some variances in the length of the sampling interval. Also, in this case a uniform distribution of random numbers is used and in this case added to each time-step. i.e. the instance of sampling is taken as $t_i = i \cdot \Delta t + t_{\text{noise}}(i)$ where t_{noise} is the bias-free uniform distribution of random numbers.

The digital resolution of an ADC is taken into account by values of range R (voltages $\pm R$) and resolution b (number of ADC bits) in the GUI. If during calculation of a time series a value \tilde{X}_i is calculated in double precision, it is shifted to the respective ADC resolved value X_i by

$$X_i = \frac{R}{2^{b-1} - 1} \cdot \text{ceil} \left(\tilde{X}_i \frac{2^{b-1} - 1}{R} \right) \quad (16)$$

where $\text{ceil}()$ is rounding to the next greater integer number.

4.6 Data Format

The data calculated for the simulated output of the accelerometer and interferometer are stored into separate files as columns of double precision numbers in text. In the case of two channels for the homodyne interferometer, there are two columns in one file separated by a double space.

It is obvious that this format is not the most efficient for storage and data input/output, however, it is considered to be the most suited because it has no need for further documentation and can be read into all systems without much trouble.

5. Conclusion and Afterthoughts

The described data generator is the first approach in the field of primary vibration calibration to provide a standard tool for benchmarking demodulation and calibration algorithms. The feasibility of this approach will eventually be validated by using it in a pilot study for software comparison on the international stage.

In order to make this tool generally available and maximize it's usefulness, the generator is designed as an applet enabling it to run on almost any personal computer or workstation with an up-to-date browser. In addition, the source

code is available under an open source licence (GPL) which enables interested scientists to read, check, change, further improve or amend the code to their requirement.

However, it shall not be concealed that this open availability is a major problem for the purpose of using the applet for comparison. Since the code is known in principle it is no insurmountable exercise to design a fake application, which uncovers the hidden data to the participant, making the original aim of a comparison, null and void. After careful consideration of many different approaches it seems that faking results can only be avoided by performing all calculations on the web-server. I.e. to use a servlet instead on an applet. This however, would put the whole computational burden on the server and would produce heavy network traffic for the deployment of the data files. As the envisaged comparison will only compare the accuracies of the signal processing implementation of ISO 16063-11, no real advantage is to be gained through the submission of "doctored" results.

6 References

- [1] ISO 16063-11:1999, Ed. 1, Methods for the calibration of vibration and shock transducers -- Part 11: Primary vibration calibration by laser interferometry, Geneva, 1999
- [2] Technical Protocol for the Comparison of Homodyne Demodulation Data Simulation