IMEKO 23rd TC3, 13th TC5 and 4th TC22 International Conference
30 May to 1 June, 2017, Helsinki, Finland

USING OPENSOURCE SOFTWARE TOOLS FOR DATA ANALYSIS IN HIGH INTENSITY SHOCK CALIBRATION OF ACCELEROMETERS

Th. Bruns, H. Volker and S. Eichstädt

Physikalisch-Technische Bundesanstalt (PTB), Germany, thomas.bruns@ptb.de

Abstract: This paper describes the concept and implementation of a model based data analysis of primary shock calibration. The concept follows to a large extent the scheme of ISO 16063-43. In addition, it uses classical statistics to combine a number of measurements. The implementation is programmed in Python, *i.e.* using open source software, in particular a new tool box for the analysis of dynamic measurements, PyDynamic, which features the integrated handling of uncertainties in terms of covariances.

Keywords: Shock calibration, parameter estimation, PyDynamic, Scientific Python

1. INTRODUCTION

The newly published standard ISO 16063-43 [1] describes the model based approach to analyse calibration data of accelerometers and thereby, how to create a quantitative mathematical description of the transfer function of the device. It addresses the use of accelerometers in broadband measurements like shock or other highly dynamic transient motion measurements. The foremost appearance of such signals is in high intensity (primary) shock calibration, where so far only peak ratio results from calibrations were described in documentary standards [2, 3]. Due to recent developments in the general field of the metrology for dynamic measurements [4] there are now open source software tools available [5, 6] that greatly simplify the implementation of the model based parameter identification. In the subsequent text, the application of these tools is described using the high intensity primary shock calibration at PTB as an example.

2. SHOCK DEVICE AND DATA

The working group “realisation of acceleration” of PTB operates two different kinds of shock acceleration devices. The so called low intensity shock acceleration standard device which works according to the hammer anvil principle and the high intensity shock acceleration standard device HISD which facilitates the propagation of an elastic wave in a Hopkinson bar. The focus of this contribution is on the latter, because, in terms of acceleration metrology it is the far more challenging device.

Figure (1) shows a sketch of PTB's HISD in order to illustrate the working principle. With the impact of a steel ball an elastic wave is excited in a thin titanium bar of 4 m length. This shock wave excites the accelerometer under calibration at the far end of this Hopkinson bar. The acceleration is measured by two Laser-Doppler-Interferometers (*c.f.* [2]).

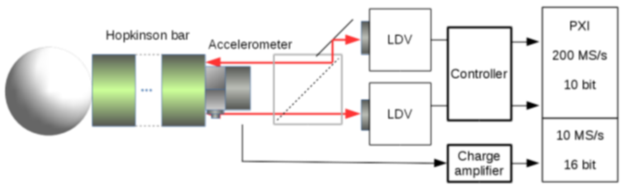

Figure 1: schematic display of PTB’s primary high intensity shock acceleration calibration set-up

Due to set-up and working principle, three signals of two different kinds are generated and sampled during a measurement. These are:

1. The voltage signal of the measurement chain of accelerometer and conditioning amplifier.
2. The two frequency modulated (fm) voltages of the Laser-Doppler-vibrometers (LDV) with a carrier frequency of 40 MHz.

All three signals have to be sampled synchronously in order to preserve the phase information during calibration. Nevertheless, due the different bandwidth requirements of those two kinds of signals, the accelerometer voltage is sampled at 10 MS/s, while the LDV signals are sampled at 200 MS/s for the subsequent demodulation. The synchronicity demand is covered by using a PXI-System with a common 10 MHz clock for the sampling cards and a common trigger line.

Ultimately, each measurement produces three data streams, two of 200 MS/s at 10 bit resolution with the displacement and thus implicit acceleration information and one at 10 MS/s with 16 bit resolution with the accelerometer output information.

The data acquisition is programmed in LabVIEW, as it provides easy access to the facilitated hardware and the necessary drivers of the sampling cards.

3. DATA ANALYSIS PROCEDURE

The data analysis in the study described here was developed in Python.

It makes heavy use of standard packages for scientific computation available in the SciPy stack [7]. In addition, a brand new and evolving package for metrological applications in dynamic measurements called PyDynamic [5] was utilized. PyDynamic is the outcome of a joint European metrology research project [4] and offers algorithms for many kinds of typical operations on dynamic systems. While applying the algorithms to the data, the package takes care of the uncertainty propagation in compliance with the GUM along with the main data processing.

Pre-processing, demodulation

In order to generate a single time series of the input quantity of acceleration for the shock event, the LDV data are demodulated by applying a digital demodulation process based on frequency mixing, applying the arc tangent function with subsequent phase unwrapping and scaling the calculated interferometric phase to displacement. This is a well-documented state of the art procedure.

Subsequently, the two demodulated LDV signals are averaged in order to reduce the effect of tilting or warp [7] of the accelerometer. The averaged signal, which presumably represents the displacement along the centre axis of the accelerometer, is differentiated twice in the time domain to convert it from displacement to acceleration. In order to reduce the inevitable noise amplification during differentiation, a digital low pass filter is applied in this process. This filtering is done in a forward-backward scheme, which cancels the phase delay introduced from each filter run.

Finally, the now available input acceleration signal at 200 MS/s is decimated by a factor of 20 to match the sampling rate of the accelerometer output of 10 MS/s.

Additional information about the reliability of the data is available in terms of the difference of the two LDV-signals. This feature was introduced as warp in [8]. The warp signal is basically calculated the same way as the input acceleration, only the averaging step is replaced by taking the difference between the two LDV-signals.

An example of the two shock time series of input and output is given in figure 2.

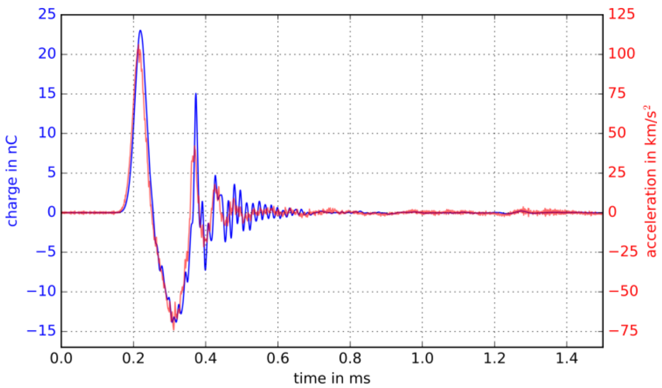

Figure 2: Example of a set of shock measurement time series, input acceleration (red) and output charge (blue).

Transform to the frequency domain

Based on the concept, described in ISO 16063-43:2015 [1], a model based parameter identification should be performed with the available high intensity primary shock calibration data of an Endevco 2270M8 accelerometer.

The presumed model is a linear time invariant single-mass-spring-damper system. The transfer function in terms of the complex sensitivity of such a system is given by:

$$S(\omega) = S_0 \cdot \frac{\omega_0^2}{\omega_0^2 + 2i\omega\delta\omega_0 - \omega^2} \quad (1)$$

For the further analysis in the frequency domain, the available time series with a given measurement uncertainty associated with each sample were Fourier transformed using a DFT. This is one of the procedures taken care of by the PyDynamic package. The subroutine *GUM_DFT(x, Ux, …)* calculates the requested complex Fourier-coefficients F_x of the time series $x(t)$ and returns them. In addition, it returns the complete co-variance matrix UF_x generated by the propagation of the time series of uncertainties $Ux(t)$ of x. Hence, the call of this function returns both a vector and a matrix. Both have block like structure of real parts and imaginary parts.

By supplying an estimate of the time domain uncertainty of the samples (Ux) this routine takes care of GUM-compliant transformation of the uncertainties into the frequency domain. For the given signals, the Fourier coefficients F_a of the acceleration and F_v of the measuring chain voltage are calculated together with the covariances UF_a and UF_v, respectively. While the relative expanded uncertainty of the LDV for the point-like measurement is considered to be better than 10^{-3}, the mean acceleration acting on the accelerometer is depending on the dynamic deformation, the warp, too. Hence the variance for the acceleration in the time domain was taken to be the sum of squares of 10^{-3} times the acting acceleration and 0.05 times the acting acceleration difference (warp). The relative expanded uncertainty of the voltage measurement was estimated to be 10^{-4}.

The accelerometer under investigation for this study is a well-known transfer standard. It has a first resonance frequency at about 55 kHz. For the parameter identification, it is favourable to have a signal bandwidth available that covers this resonance, however, it does not make sense to go far beyond that. Accordingly, prior to the transformation into the frequency domain, the time series were further decimated to a sampling rate of 200 kS/s, resulting in a bandwidth from DC to 100 kHz.

Charge determination

The complex sensitivity given by Eq. (1) is related to the charge output of the accelerometer. The conversion of the sampled output voltage of the measuring chain to a charge output was performed by eliminating the charge amplifier influence in the frequency domain. For the conversion factor S_uq of the charge amplifier, calibration results in magnitude and phase from 0.1 Hz to 100 kHz were available. These results were interpolated in terms of magnitude and phase by use of *scipy.signal.interpolate()* to the frequency points needed for the DFT-transformed voltage. The calculation of the charge sensitivity then followed the equation:

$$S_{\mathrm{qa}}(\omega) = \frac{F_{\mathrm{v}}(\omega)}{S_{\mathrm{uq}}(\omega)} \cdot \frac{1}{F_{\mathrm{a}}(\omega)} = \frac{F_{\mathrm{q}}(\omega)}{F_{\mathrm{a}}(\omega)} \quad (2)$$

This involves two complex valued divisions which are, again, connected with a propagation of uncertainties. However, these cases can be considered a de-convolution in the frequency domain and as such are in the scope of PyDynamic. With the corresponding vectors and their respective covariance matrices the sub-routine call

F_q, UF_q = DFT_deconv(S_uq, F_v, US_uq, UF_v)

calculates the complex charge output vector $F_{\mathrm{q}}(\omega)$ as well as its covariance matrix $UF_{\mathrm{q}}(\omega)$. The covariance $UF_{\mathrm{v}}(\omega)$ needed as input was already calculated via *GUM_DFT* from time domain data, while the covariance $US_{\mathrm{uq}}(\omega)$ of the charge amplifier is simply the diagonal matrix of the variances, i.e. the squared absolute measurement uncertainties of the charge amplifier calibration. The expanded relative uncertainty for the magnitude of the charge amplifier calibration was estimated to be $2\cdot10^{-4}$ (c.f. [11]).

The second division is performed with a similar call

S_qa, US_ qa = DFT_deconv(F_a, F_q, UF_a, UF_q)

Parameter identification

In contrast to the process described in [1] or [9], a two-stage approach was followed for this study. In a first stage the low-frequency limit of the complex sensitivity S_0 was determined as the weighted mean of the respective components of the calculated sensitivities:

$$S_0 = \left(\sum_{\omega_{\mathrm{L}}} \frac{S_{\mathrm{qa}}(\omega)}{u_{\mathrm{qa}}^2(\omega)}\right) \cdot \left(\sum_{\omega_{\mathrm{L}}} \frac{1}{u_{\mathrm{qa}}^2(\omega)}\right)^{-1} \quad (3)$$

with $0 < \omega_{\mathrm{L}} \leq 2\pi \cdot 3.5$ kHz. The weights are the inverse of the variances u_{qa}^2 of the respective S_{qa}. These were in turn the diagonal elements of the covariance matrix US_{qa} calculated previously. Accordingly, the variance of S_0 was

$$\mathrm{var}(S_0) = \left(\sum_{\omega_{\mathrm{L}}} \frac{1}{u_{\mathrm{qa}}^2(\omega)}\right)^{-1} \quad (4)$$

Normalizing the transfer function of eq. (1) and inverting it leads to a polynomial expression depending on powers of ω:

$$\frac{S_0}{S(\omega)} = 1 + 2i\frac{\delta}{\omega_0} \cdot \omega - \frac{1}{{\omega_0}^2} \cdot \omega^2. \quad (5)$$

The parameter identification of the resonant frequency ω_0 and damping δ is then, basically, a complex valued linear regression of that polynomial (5).

Because, the model equations in dynamic metrology are manifold, there is no "one for all" function to solve this kind of problem like given in Eq. (2). Fortunately, the example collection of the PyDynamic package features sample code for a second order system, which could easily be adapted to this individual model equation. The example can be found in PyDynamic/examples/fit_sos.py.[1]

Using Monte Carlo (MC) techniques in compliance with [10] a large number of samples (N=3000) for $\frac{S_0}{S(\omega)}$ was drawn from a multivariate normal distribution. In order to provide the proper covariance of that distribution, the normalisation was again calculated as a deconvolution in the frequency domain. The denominator was a constant, S_0, in this case.

For each sampled frequency series, the regression problem (3) was solved and a set of the parameters ω_0 and δ was identified. From each group of N identified parameters, a best estimate and its variance could be calculated by averaging.

4. FURTHER STATISTICS

A typical calibration run on the HISD involves measurements at several shock acceleration intensities between 10 km/s² and 100 km/s², and repetitions for each of the pre-set intensities. Hence, at the end of a run and the respective identification processes, we obtain several sets of parameters with associated parameter uncertainties for a single artefact. Figures 3 and 4 show the dispersion of the parameters S_0 and $f_0 = \frac{\omega_0}{2\pi}$ for one of PTB's own shock references (Endevco 2270 AEOP9).

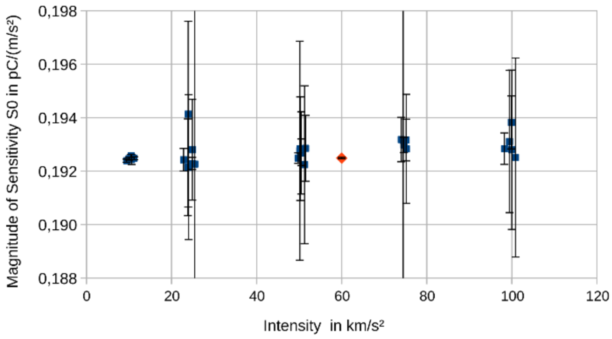

Figure 3: Dispersion of S_0 (blue points) over the applied shock intensity for a typical shock calibration run. The red point indicates the weighted mean.

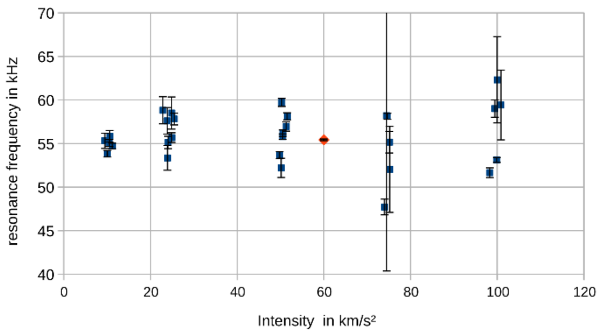

Figure 4: Dispersion of f_0 (blue points) over the applied shock intensity for a typical shock calibration run. The red point indicates the unweighted mean.

[1] At the time of writing this could be found in the development branch of the repository, only.

It should be noted, that in the current state of development the consistency between the calibration runs is not tested. For a fully consistent measurement one would expect the parameter sets of different runs at different intensities to be consistent within the significance limit given by the associated expanded uncertainties. This is still a concern and a matter of ongoing investigation.

Also, the parameter of damping is still of some concern in the data analysis algorithm, because in rare cases it can result in a negative result of the fit. This is not reasonable, however, it cannot be prevented in the described simple linear regression scheme.

The obvious approach to gain a single identification result for a complete run of several shock measurements is averaging and while parameter uncertainties are already at hand, it seems appropriate to use a weighted mean over all measurements in that run.

Note that a different approach to gain a single valid calibration result is to do a joint identification by using all transfer functions in a single identification process. This is possible by setting up a single (large) vector stacking all frequency response data and a single (very large) covariance matrix of block-diagonal shape. However, at the stage where samples from a multivariate normal distribution according to that given covariance are taken, this approach becomes computationally prohibitive. Feasibility on this account may be achieved in the future by implementing the PyDynamic toolbox with support for sparse matrices.

5. MODELL TEST

As there is not much knowledge from experience available yet on the topic of parameter identification, the validity of the gained set of best estimates and uncertainties of the model parameters had to be tested in this study. For this purpose, a kind of bootstrapping process was applied. Again with the support of PyDynamic, a digital IIR-filter was designed using the function *sos_phys2filter(S, d, f0)* in the subpackage *PyDynamic.misc.SecondOrderSystem*. This filter poses as a software implementation of the calibrated accelerometer. As such it should return the same charge output time series as the accelerometer for a given input of an acceleration time series. This was tested with several of the acceleration time series applied during calibration. In order to take account of the parameter's uncertainties, again, a Monte Carlo scheme was applied:

- A number (N=2000) of sets $\{S_0, \delta, \omega_0\}_i$ was drawn from respective normal distributions.
- For each set an IIR-filter was designed:
  - The PyDynamic function *bc,ac = SecondOrderSystem.sos_phys2filter*(S_0, ω_0, δ) returns coefficients of an analogue or continuous filter representing a second order system.
  - *b,a = scipy.signal.bilinear(bc, ac, fs)* transforms and returns the respective coefficients of the discrete or digital filter for a given sampling rate f_S.
  - The digital filter is applied to the time series of the (mean) acceleration of the chosen test case *charge = scipy.signal.lfilter(b, a, acceleration)* resulting in a calculated time series of charge.
- A time series of the cumulative mean and cumulative variance is updated for each of the N time series of charge.

Based on the cumulative mean and variance it can be checked whether the originally measured time series of charge is within the scatter bounds of the set of calculated charge time series.

The results for an example time series are depicted in figure (5). It was measured with an Endevco 2270M8 accelerometer. The identified parameter set (weigted mean) was as follows:

Base sensitivity S_0=	0.19249 (0.00003) pC/(m/s²)
Damping δ=	0.0474 (0.0017)
Resonance frequency ω_0=	2π·55455 (99) s-1

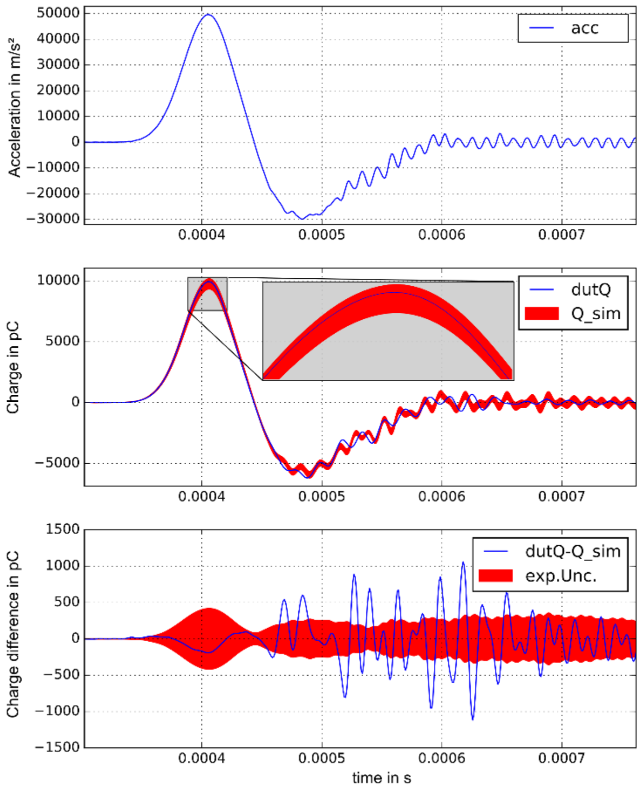

Figure 5: Test case of a 50 km/s² shock calibration measurement. The top chart shows the acceleration time series. The center chart shows the charge time series as measured(blue) and as calculated (red) with the spread of the MC-simulation. The bottom chart shows the difference of measured and simulated charge together with the exp. uncertainty derived from the test case.

This example of a 50 km/s² shock measurement exemplifies that the use of the PyDynamic routines generates a time dependent uncertainty band based on the variance of the identified parameters of the test case (red). In the period of the main shock event, the calculated charge (including its expanded (k=2) uncertainty) covers the measured charge, hence, the identified parameter set is considered to be consistent with this measurement. Note, that the test case was excluded from the weighted mean defining the parameter set.

At the end of the main shock event (t > 450 µs), the difference between measured and calculated charge is no more completely covered by the expanded uncertainty area (red). Although the similarities in the structure of the charge curves is striking, a small delay causes oscillating deviations that exceed the uncertainty spread. This effect is still a point of active investigation.

6. SUMMARY AND OUTLOOK

This contribution demonstrates the concept of model based parameter identification in high intensity primary shock calibration. The concept was implemented in Python with the support of additional open source packages. Beyond the already well established scientific tools of the SciPy-Stack [7], a newly available package called PyDynamic [5, 6] was used for Fourier-transform, de-convolution, identification and filter design. This tool box is the outcome of a joint European Metrology research project and is currently under active development. The major benefit from its use is the integrated GUM-compliant handling of complex uncertainties in terms of covariance matrices and the included examples for second order LTI-systems.

In the demonstrated case, various measurements were combined by classical statistics after identification. The joint simultaneous identification of the combined sets off measurements would be another approach. In that case however, the size of the joint covariance matrix grows with the square of the number of measurements and gets prohibitively large in terms of memory consumption.

The presented use case is not yet perfectly worked out as the consistency of different parameters sets is not assured or even analyzed. In addition, the damping parameter may be identified outside of the possible physical limits in rare cases. The investigation of the reasons for this and a suitable algorithm to avoid it are part of the ongoing research and development.

In a future release of PyDynamic the handling of sparse matrices is supposed to be supported. When this happens, problems of larger scope and joint analysis of multiple measurements will become an additional option.

7. REFERENCES

References:

[1] ISO 16063-43:2015 “Methods for the calibration of vibration and shock transducers -- Part 43: Calibration of accelerometers by model-based parameter identification”, Geneva, Switzerland, 2015

[2] ISO 16063-13:2001 “Methods for the calibration of vibration and shock transducers -- Part 13: Primary shock calibration using laser interferometry”, Geneva, Switzerland, 2015

[3] ISO 16063-22:2005 Methods for the calibration of vibration and shock transducers -- Part 22: Shock calibration by comparison to a reference transducer”, Geneva, Switzerland, 2015

[4] EMPIR 14SIP08, “Standards and software to maximise end user uptake of NMI calibrations of dynamic force, torque and pressure sensors“,http://mathmet.org/projects/14SIP08/

[5] S. Eichstädt et al., “Evaluation of dynamic measurement uncertainty – an open-source software package to bridge theory and practice“, J. Sens. Sens. Syst., 6, 97-105, 2017, doi:10.5194/jsss-6-97-2017

[6] S. Eichstädt et al., “PyDynamic - Analysis of dynamic measurements”, http://pydynamic.readthedocs.io

[7] SciPy developers , “Scientific Computing Tools for Python“ http://scipy.org/about.html

[8] H. Volkers et al., “Investigations of Reference Surface Warp at High Shock Calibrations”, 3rd IMEKO TC 22 International Conference, Capetown, RSA, 2014

[9] A. Link et al., “Calibration of accelerometers: determination of amplitude and phase response upon shock excitation”, Meas. Sc. Techn. 17, 2006, doi:10.1088/0957-0233/17/7/030

[10] JCGM 101:2008, Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in measurement” — Propagation of distributions using a Monte Carlo method”, BIPM, Paris, France 2008

[11] H. Volkers, Th. Bruns, “The Influence of Source Impedance on Charge Amplifier”, XX IMEKO World Congress, Busan, Rep. of Korea, 2012

ACKNOWLEDGEMENT

Part of this work has been carried out as part of the European Metrology Programme for Innovation and Research (EMPIR) project 14SIP08. The EMPIR initiative is co-funded by the European Union's Horizon 2020 research and innovation programme and the EMPIR Participating States.