The ITA-Toolbox: An Open Source MATLAB Toolbox for Acoustic Measurements and Signal Processing

Marco Berzborn, Ramona Bomhardt, Johannes Klein, Jan-Gerrit Richter, Michael Vorländer

Institute of Technical Acoustics, Kopernikusstr. 5, 52074 Aachen, Germany

Website: www.ita-toolbox.org — Email: toolbox-dev@akustik.rwth-aachen.de

Introduction

This paper presents the ITA-Toolbox [12, 14, 11, 10] for acoustic measurements and signal processing tasks in MATLAB. The ITA-Toolbox is being developed at the Institute of Technical Acoustics in order to provide a common and unified code base for researchers and students, and to ensure consistent data formats. This improved the reproducibility and sharing of research results and reduced the fragmentation and redundancy of code written by members of the institute for their daily work. Moreover, students directly benefit from the existing code base as they can purely focus on their thesis projects instead of worrying about implementing fundamental signal processing tasks.

Since 2010, the ITA-Toolbox is available as open source software under the Berkeley Software Distribution (BSD) license. Recently, the authors decided to migrate development to the RWTH Aachen GitLab repository with public access to make research conducted with the ITA-Toolbox more transparent and reproducible and to give researchers the opportunity to contribute.

This paper gives an overview over the core features of the ITA-Toolbox, followed by a summary of the included applications. Finally, the role of the ITA-Toolbox in the light of open science is briefly outlined and informations on where to get the ITA-Toolbox as well as a short introduction on how to contribute to the project are given.

Core Features

The central elements of the ITA-Toolbox are the value classes `itaAudio` for real audio data and `itaResult` for simulated data. An instance of `itaAudio` can contain

- uniformly sampled multi-channel data,
- data in the time and frequency domain,
- audio data details (e.g. sampling rate),
- user meta data (e.g. comments, coordinates),
- automated meta data (e.g. processing history).

In contrast, `itaResult` objects are designed to store non-uniformly sampled data in only one domain of choice without audio specific meta data. The `itaAudio` class provides methods to

- visualize (time/frequency plot (`.pt`,`.ptd`/`.pf`)),
- characterize (length (`.length`), RMS (`.rms`)),
- transform (FFT/IFFT (`.fft`/`.ifft`))

the stored data, whereas the `itaResult` objects consequently cannot be transformed.

At its core, the ITA-Toolbox provides functions to create, manipulate, and save `itaAudio` as well as `itaResult` objects. The class objects are created using the supplied file I/O functions to read recorded or simulated data (function `ita_read`: supported formats include WAV, DAT, SPK, UNV), by generating artificial signals (function `ita_generate`), or can directly be acquired using the ITA-Toolbox measurement routines.

The measurement functionality is provided by the handle classes

- itaMSRecord: recording,
- itaMSPlaybackRecord: additional playback,
- itaMSTF: transfer function measurements,
- itaMSTFinterleaved: interleaved measurements [13].

The measurement class objects contain the

- hardware input/output channel specification,
- excitation and compensation signals,
- measurement chain calibration and latency data.

Among others, class methods are implemented for the

- calibration of hardware,
- measurement execution.

All elements of the measurement chain can be specified and calibrated individually using the command window or a graphical user interface (GUI). Depending on the class, the execution returns either the raw sensor signal at the input or an impulse response.

For the audio hardware communication, the ITA-Toolbox relies on Playrec [4] and PortAudio [5]. ASIO compatible hardware can be selected via command window or GUI and will subsequently be used by the measurement classes.

To facilitate and automate measurements, specific auxiliary instrumentation like step motors, as well as amplifier and interface settings can be directly controlled via

- MIDI (using function `ita_midi`),
- OSC (using class `itaOSC`),
- serial interface (e.g. RS232, as in class `itaEimar`).

Tasks like directivity measurements require the documentation of spatial information, such as measurement points, dimensions, or orientations. The value class `itaCoordinates` provides the suitable container for spatial data. Class instances contain

- Cartesian coordinates in $[x, y, z]$,
- cylindrical coordinates in $[r, \vartheta, z]$,
- spherical coordinates in $[r, \vartheta, \varphi]$.

The transformation between the different coordinate systems is done implicitly within the objects. Class objects can be

- added and subtracted (vector addition, `+/-`),
- plotted (function `scatter`).

For the documentation of measurement positions and orientations, `itaCoordinates` objects can be stored per channel in the `.channelCoordinates` and `.channelOrientation` properties of `itaAudio` and `itaResult` objects and will automatically be saved there in certain measurement procedures.

Acquired or loaded `itaAudio` and `itaResult` objects can be manipulated using the core signal processing features of the ITA-Toolbox. These functions include

- time cropping (function `ita_time_crop`),
- time shifting (function `ita_time_shift`),
- time windowing (function `ita_time_window`)
- filtering (function `ita_mpb_filter`),
- smoothing (function `ita_smooth`),
- resampling (function `ita_resample`),
- re-quantizing (function `ita_quantize`).

Certain operations allow or require the use of several `itaAudio` and `itaResult` objects. Examples for such operations are the

- creation of multi-channel objects (function `merge`),
- time domain summation (operator `+`),
- frequency domain multiplication (operator `*`),
- frequency domain division (operator `/`).

Processed `itaAudio` and `itaResult` objects can be saved using the native MATLAB `save` command or exported into various formats (including WAV, DAT, SPK, UNV) using the function `ita_write`.

Applications

In addition to the core features of the ITA-Toolbox, multiple applications such as beamforming, binaural signal processing, loudspeaker measurement and modeling, numerical acoustics, special functions (e. g. spherical harmonics or Green's function) as well as room acoustics and sound transmission are implemented.

Beamforming

Different types of the beamforming algorithms *Delay and Sum*, *Minimum-Variance Distortionless Response*, *MUSIC*, *Subspace Beamforming*, *Functional Beamforming*, *CLEAN*, and *DAMAS* are available for a specified array geometry and measurement grid [24].

Binaural Signal Processing

For binaural reproduction, the itaHRTF class handles head-related transfer function (HRTF) data sets. This class provide easy access to a specific spatial direction, the determination of the interaural time and level difference (ITD and ILD), spatial interpolation or the import and export of SOFA files [20]. To measure and calculate a headphone transfer function (HpTF), the algorithms of Masiero and Fels [21] are implemented. Additionally, the generation of cross talk cancellation (CTC) filters is available [19].

Loudspeaker Measurement and Modeling

The louspeaker application enables the user to measure the input impedance and determine the Thiele-Small parameters of a loudspeaker [25]. Furthermore, the total harmonic distortion (THD) or the maximum sound pressure level (SPL) for a constant THD can be determined [15].

Numerical Acoustics

An acoustic finite element method (FEM) for solving exterior acoustic problems (numerical partial solution, eigenvalues, and eigenvectors) with different boundary conditions is implemented [27]. Cuboids can be meshed automatically or arbitrary meshs can be imported from UNV files. An image source model for convex geometries is available [22].

Room Acoustics and Sound Transmission

Room impulse responses can be analyzed and modeled using the room acoustics application of the ITA-Toolbox [18, 8]. The reverberation time, clarity, definition (T_{60}, C_{80}, D_{50}) as well as the sound insulation index R can be calculated.

Special Functions, Decomposition and Reconstruction

Spherical harmonic (SH) functions or monopoles can be used to decompose and reconstruct directivities [26]. In addition to this, the decomposition of a directivity in pole-zero models is also possible [17, 9].

Open Science

The source code of the ITA-Toolbox is available as an open source project from the project homepage [3] and the RWTH Aachen GitLab repository [6] and is licensed under the BSD license. In addition to the general availability of the implemented measurement, processing, and analysis methods and techniques, this also allows for a greater transparency of the research

conducted with the ITA-Toolbox. To ensure better reproducibility of research results, the implementation of digital object identifier (DOI) citations for explicit versions of the ITA-Toolbox using a service such as Zenodo [7] is planned for the future.
Datasets generated with the ITA-Toolbox that are currently available are the *ITA Academic HRIRs*, an HRTF dataset of the ITA dummy head [23], and the *Aachen anthropometric HRTF* database by Bomhardt and Fels [16]. Both can be downloaded from the website of the Institute of Technical Acoustics [2].

Getting Involved

The authors would like to invite researchers to contribute to the ITA-Toolbox by reporting bugs, and fixing bugs as well as suggesting features, improving features, and developing new features. Bug reports and source code contributions can be submitted through the bug tracking system and via merge requests to the repository respectively. Detailed contribution guidelines for submitting bug reports and merge requests via GitLab as well as code style guidelines can be found on the GitLab repository pages [6].
To contribute to the project via the repository, users need to login into GitLab either with an existing GitHub [1] account. Users without a login can also submit bug reports and contributions by sending an electronic mail to `toolbox-dev@akustik.rwth-aachen.de`. No login is required to only download and use the ITA-Toolbox.
Requirements for the ITA-Toolbox are

- MATLAB R2014b or higher,
- MATLAB Signal Processing Toolbox,
- MATLAB Curve Fitting Toolbox (partially),
- PortAudio [5] compatible audio interface to run measurements (ASIO driver compatibility recommended).

In order to get started, the ITA-Toolbox includes documented step by step tutorial scripts that cover the core functionality and some applications. An overview over all available tutorial scripts can be displayed by running the command `ita_tutorialOverview` in the MATLAB command window.
An HTML documentation that fully integrates into the MATLAB help browser can be generated by running the command `ita_generate_documentation`. The documentation can then be accessed via the *Supplemental Software* section inside the help browser. For information on an explicit function or class, users can execute the command `doc ita_function_or_class_of_interest` with the corresponding function or class name instead.
For general questions about the ITA-Toolbox, an FAQ section and Wiki pages provided on the project homepage [3] and the RWTH Aachen GitLab repository [6] respectively. The user mailing group (see ITA-Toolbox homepage [3]) can be used to get in contact with other users.

Acknowledgement

The authors would like to thank all former ITA-Toolbox developers and everyone who contributed to the project as well as the Institute of Technical Acoustics for supporting the development.

References

[1] GitHub. `www.github.com`. Online; accessed 24-March-2017.

[2] Institute of Technical Acoustics Homepage. `https://www.akustik.rwth-aachen.de/cms/Technische-Akustik/Forschung/~lxfd/Downloads/`. Online; accessed 24-March-2017.

[3] ITA-Toolbox Homepage. `www.ita-toolbox.org`. Online; accessed 24-March-2017.

[4] Playrec. `www.playrec.co.uk`. Online; accessed 24-March-2017.

[5] PortAudio. `www.portaudio.com`. Online; accessed 24-March-2017.

[6] RWTH Aachen GitLab Repository. Online; accessed 24-March-2017.

[7] Zenodo. `www.zenodo.org`. Online; accessed 24-March-2017.

[8] M. Aretz, P. Dietrich, and M. Vorländer. Application of the mirror source method for low frequency sound prediction in rectangular rooms. *Acta Acustica united with Acustica*, 100(2):306–319, 2014.

[9] D. Deschrijver, B. Haegeman, and T. Dhaene. Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses. *IEEE Transactions on advanced packaging*, 30(2):216, 2007.

[10] P. Dietrich, M. Guski, and J. Klein. Measurements and Room Acoustic Analysis with the ITA-Toolbox for MATLAB. In *Fortschritte der Akust. - DAGA*, 2013.

[11] P. Dietrich, M. Guski, and M. Vorländer. Influence of loudspeaker distortion on room acoustic parameters. In *Fortschritte der Akust. - DAGA*, 2013.

[12] P. Dietrich, B. Masiero, M. Pollow, R. Scharrer, and M. Müller-Trapet. MATLAB Toolbox for the Comprehension of Acoustic Measurement and Signal Processing. In *Fortschritte der Akust. - DAGA*, pages 517–518, 2010.

[13] P. Dietrich, B. Masiero, and M. Vorländer. On the optimization of the multiple exponential sweep method. *AES J. Audio Eng. Soc.*, 61(3):113–124, 2013.

[14] P. Dietrich, B. S. Masiero, R. Scharrer, M. Müller-Trapet, M. Pollow, and M. Vorländer. Application of the MATLAB ITA-Toolbox: Laboratory Course on Cross-talk Cancellation. In *Fortschritte der Akust. - DAGA*, pages 471–472, 2011.

[15] DIN German Institute for Standardization. Sound system equipment - Electroacoustical transducers - Measurement of large signal parameters (IEC 62458:2010); German version EN 62458:2011, 2011.

[16] J. Fels and R. Bomhardt. A high-resolution head-related transfer function dataset and 3d ear model database. *The Journal of the Acoustical Society of America*, 140(4):3276–3276, 2016.

[17] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999.

[18] International Organization for Standardization. Measurement of room acoustic parameters, 2009.

[19] T. Lentz. Dynamic crosstalk cancellation for binaural synthesis in virtual reality environments. *Journal of the Audio Engineering Society*, 54(4):283–294, 2006.

[20] P. Majdak, T. Carpentier, R. Nicol, M. Parmentier, A. Roginska, Y. Suzuki, K. Watanabe, H. Wierstorf, H. Ziegelwanger, and M. Noisternig. Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions. In *Audio Eng. Soc. Conv. 134*, 2013.

[21] B. Masiero and J. Fels. Perceptually robust headphone equalization for binaural reproduction. In Audio Engineering Society, editor, *130th Audio Engineering Society Convention*, 2011.

[22] F. P. Mechel. Improved mirror source method in roomacoustics. *Journal of sound and vibration*, 256(5):873–940, 2002.

[23] A. Schmitz. Ein neues digitales kunstkopfmeßsystem. *Acta Acustica united with Acustica*, 81(4):416–420, 1995.

[24] H. Teutsch. *Modal array signal processing: Principles and applications of acoustic wavefield decomposition*, volume Vol. 348 of *Lecture notes in control and information sciences*. Springer, Berlin and Heidelberg and New York, 2007.

[25] N. Thiele. Loudspeakers in vented boxes: Part 1. *J. Audio Eng. Soc.*, 19(5):382–392, 1971.

[26] E. G. Williams. *Fourier acoustics: sound radiation and nearfield acoustical holography.* Academic press, 1999.

[27] O. C. Zienkiewicz, R. L. Taylor, and R. L. Taylor. *The finite element method.* McGraw-hill London, 1977.