

A MATLAB Toolbox for the needs of acousticians

ITA Toolbox – Getting Started

`toolbox-dev@akustik.rwth-aachen.de`

March 15, 2013

Institute of Technical Acoustics
Prof. Dr. rer. nat. Michael Vorländer
RWTH Aachen University

Contents

1	Introduction	2
2	Installation Instructions	3
3	How to get started?	4
4	FAQ	6
5	Classes – Definitions, Use, Tips	7
5.1	Overview	7
5.1.1	Get and Set Data	7
5.1.2	Constructor	8
5.1.3	Metadata Fields	8
5.1.4	Read-only Metadata Fields	8
5.1.5	Operators	9
5.1.6	Vectors	9
5.1.7	Overloaded Standard Functions	9
6	Plot Routines – Keyboard Shortcuts	10
7	Graphical User Interface (GUI)	11
7.1	How to use it ?	11
7.2	Creating your own GUIs	12
8	Measurements in a Nutshell	13
8.1	Audio Objects – itaAudio	14
8.2	Measurement Chain Concept	14
8.3	ASIO under Windows x64	18
9	Official Stuff	19
9.1	Praise the ITA-TOOLBOX	19
9.2	Literature	19

1 Introduction

In September 2008 we decided to provide the useful MATLAB source code to students and researchers at the Institute of Technical Acoustics at RWTH Aachen independently developed in a unified way. Therefore, a common data structure and similar names for important functions have to be used. This directly led to the idea of building a special MATLAB Toolbox including all these functions and documentation. The starting point was to import and export data acquired by the MS-DOS based measurement system Monkey Forest. The audio data is represented in MATLAB as an audio object containing a data entry, either in frequency or time domain and additional metadata entries, i.e. sampling rate, units, and channel names. All ITA-TOOLBOX functions can directly operate on this data struct and no additional information about the signal is required, since everything is directly stored as metadata inside the audio object. An overview of the ITA-TOOLBOX using an automatically generated HTML documentation is available in the ITA-TOOLBOX folder. In Figure 1 the division in kernel, advanced functions and applications is shown. In the middle of 2011 the decision was made to go public. The kernel of the ITA-TOOLBOX is now available at our website. This kernel is very useful for academic research, students in engineering related to acoustics or also industry. Since spring 2013 the open-source applications *measurement* and *roomacoustics* are available on the website.

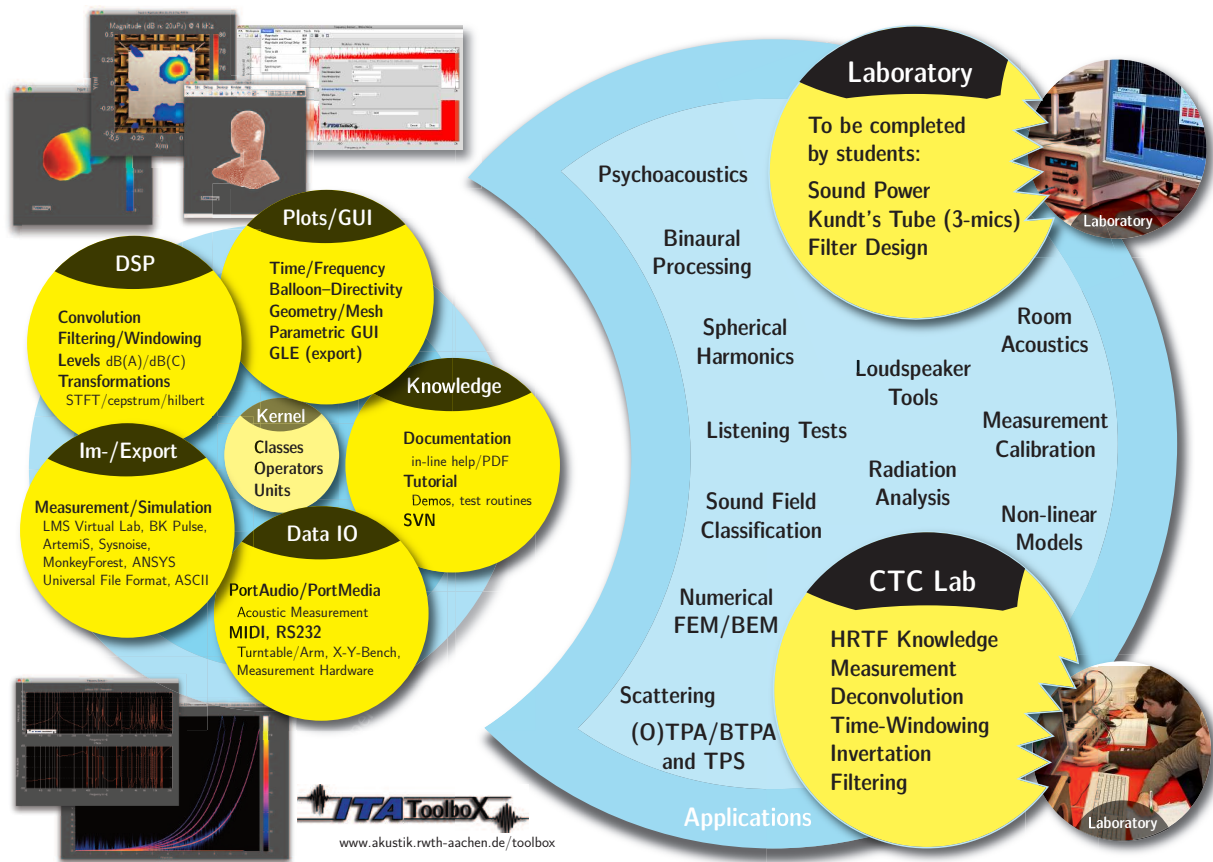


Figure 1: Overview of the ITA-TOOLBOX

2 Installation Instructions

The ITA-TOOLBOX runs on Windows, Linux and Mac OS X. The oldest compatible version of MATLAB used is MATLAB R2008a, all prior versions do not support the object oriented programming approach followed by the ITA-TOOLBOX. The signal processing toolbox is required for some routines, however, most of the code runs without additional toolboxes.

We suggest to create the Folder containing the ITA-Toolbox inside of `..\Eigene Dateien\MATLAB\` or `..\My Documents\MATLAB\`. For Mac OS X or Linux, you can use any directory in your home folder, e.g.: `~/MATLAB/`.

There are known issues due to incompatible syntax between the 2009 versions. This problems will not be solved in particular in the future as the developers work with the up-to-date versions of MATLAB only. A short version of the installation procedure can also be found in the root folder of ITA-TOOLBOX in the `README.TXT`.

Open MATLAB and navigate to `\My Documents\MATLAB\ITA-Toolbox\`. Afterwards type `ita.toolbox_setup`. This function will set all paths to your new ITA-TOOLBOX.

3 How to get started?

We provide a new tutorial script, which can directly be used by typing `edit ita_tutorial.m`. Alternatively you can see the content by opening the file `ita_tutorial.html` (in the folder *Tutorials*) in your favorite web browser. A tutorial dedicated to coordinates and plotting of coordinates can be found at `ita_tutorial_itaCoordinates.m`. For the *measurement* application see `ita_tutorial_measurement.m` and for the *roomacoustic* application see `ita_tutorial_roomacoustics.m`

Please take some time to study this tutorial to get in touch with the object-oriented concept used throughout the ITA-Toolbox. Depending on your MATLAB skills and your experience, this tutorial might take 20–30 minutes for experts and 45–90 minutes for normal MATLAB background knowledge.

There is a graphical user interface available for the ITA-Toolbox. This is a graphical frontend for the console based functions. Please have a look at this GUI by typing `ita_toolbox_GUI`. Depending on your version and your access rights, you also see different entries in the Applications menu.

Type in `ita_preferences` to change your settings for ITA-Toolbox as shown in Figure 2. This includes standard plot settings, IO settings regarding sound card, sampling rate and special settings.

If you experience any graphic issues where your axis labels are written upside down or mirrored, please switch off the transparent toolbox logo under the last tab called *Expert Stuff*!

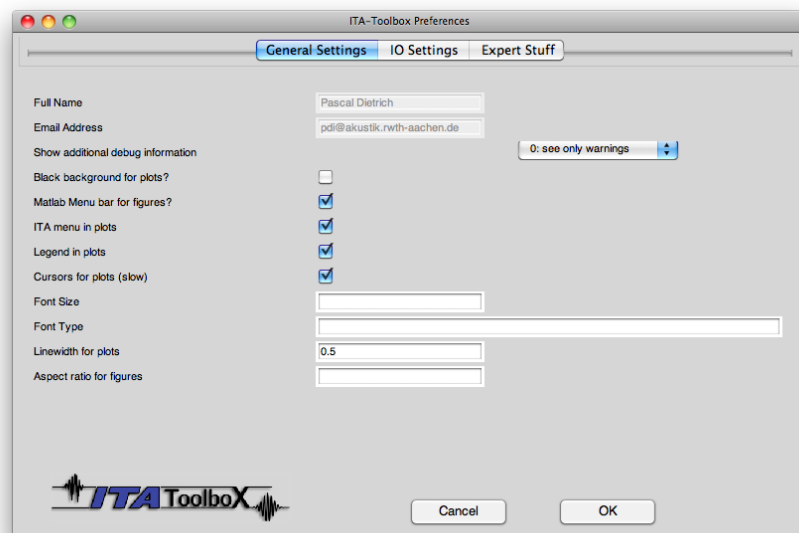


Figure 2: `ita_preferences`: Central point to specify and modify standard settings and behavior of the ITA-TOOLBOX

4 FAQ

Why does the plot export save this plot does not work for me ? There are two possible export routines used to export plots to graphic files on your hard-drive. The simple function requires Ghostscript installed on your computer. The *GLE – Graphics Layout Engine* version requires GLE installed <http://glx.sourceforge.net>.

Why does my plot have mirrored axis ? This is known bug inside the mess of MATLAB/Java. Disable the toolbox logo in `ita_preferences` under *Expert Stuff*.

Why are my plots slow ? You might be plotting to many channels or too long signals. You can significantly speed-up the plots by disabling *legends, toolbox logo, menu bar, ITA menubar, cursors* in the ITA-Toolbox preferences page.

Why are my MATLAB paths not stored correctly ? This is due to a problem in the rights handling inside of MATLAB. If you have more than one user working on a computer this problem might appear. Normally MATLAB stores the paths (`pathdef.m`) in the MATLAB system folder, where most users have no rights to write data to, except for the admins. The workaround is the following: Specify the working directory on Windows computers inside the link to the MATLAB on your desktop. You might have to copy an existing link, since mostly the standard link to MATLAB can only be changed by admins. The working directory should point to your desired directory in which the folder ITA-Toolbox can also be found, e.g. `../Eigene Dateien/MATLAB`. Start MATLAB then and run the Toolbox setup. Navigate inside of MATLAB to the specified working directory and type `savepath([pwd filesep 'pathdef.m'])`. Now you have a file called `pathdef.m` in you working directory which is loaded everytime you start MATLAB by the link/icon you have on your desktop.

5 Classes – Definitions, Use, Tips

Probably you already recognized that every time you work with an `itaAudio` object you see a little abstract of the metadata and some links to trigger common functions such as `transform`, `plot` and `play`. But there is more metadata stored in each `itaAudio` object as can be seen with `doc itaAudio`. To get a general idea of the design, the next part will present the structure of `itaAudio`. You can also get a very good overview of the class structure by typing `doc itaAudio`. The same works for the other classes, e.g. `itaResult`, `itaValue`, `itaCoordinates`. **Please use this way to inform yourself about the insight of classes!**

5.1 Overview

There are four basic classes that can be used to store acoustical data. You might already know them from the tutorial in `ita_tutorial.m`:

- `itaValue`– simple values with physical unit
- `itaAudio`– data that can be transformed from time to frequency domain and vice versa (equidistant sampling in time and frequency domain)
- `itaResult`– for the data that does not fit into the `itaAudio` definition, e.g. simulation data, reverberation times, third-octave band levels, etc.
- `itaCoordinates`– list of points in space, including coordinate transforms and plot functions of geometries and spatial audio data. See `ita_tutorial_itaCoordinates.m` for details.

5.1.1 Get and Set Data

To get and set the data there are following fields which are readable and writeable. For demonstration an `itaAudio`-Object with the name `ao` is used.

```
ao.timeData      % time data stored in a matrix with only two dimensions
ao.freqData      % freq data stored in a matrix with only two dimensions
```

Therefore you are free to choose the `ao.timeData` or `ao.freqData` field no matter in which domain the `ao` is. For internal computations and maybe for other functions the n-dimensional matrix `ao.time` and `ao.freq`, is always squeezed into two dimensions in the fields `ao.timeData` and

`ao.freqData`. For a better understanding: the data is just saved in one domain. So if you are re using the `ao.freq` field since the domain is set to 'time', you always trigger a fft-computation in the background. To switch to the other domain you can just use the overloaded operators `ao'` (FFT) or `ao.'` (IFFT).

5.1.2 Constructor

The constructor of the class can be triggered in some simple ways:

```
ao = itaAudio()           % generate an empty object with samplingRate = NaN
ao = itaAudio(n,m)        % generate a multi-instance, stored in an array
                           % with n-by-m dimensions
ao = itaAudio([x y z])    % generate a multi-instance, stored in an array
                           % with x*y*z dimensions
ao = itaAudio([x y z], size) % generate a multi-instance, stored in an array
                           % with x*y*z dimensions and preinitialized
                           % with datatype NaN of the size i.e.
                           % size=[2^16 2^16]
ao = itaAudio(a, b, c)    % generate a struct with ao.data = a,
                           % ao.samplingRate = b, ao.domain = c
ao = itaAudio(struct)
```

5.1.3 Metadata Fields

Use these fields to store general information on the data.

```
ao.channelNames = cellstr
ao.channelUnits = cellstr
ao.channelCoordinates = itaCoordinates
ao.comment = char
ao.samplingRate = double
ao.signalType = char           % could be 'power' or 'energy'
ao.dataType = class(mData)     % by default 'double' is used,
                               % 'single' is also possible
ao.dataTypeOutput = class(mData) % to separate the output data typ ('double' by
                               % default, 'single' also possible)
```

The metadata fields should store everything important next to the audio data. The `ao.history` gives an overview of functions used to manipulate the object before. The `ao.errorLog` lists all errors i.e. zero-division or clipping. The `ao.signalType` shows if the signal is a power or a energy signal. With the `ao.dataType` and `ao.dataTypeOutput` is the possibility given to get some speed enhancements and anyway stay compatible to existing functions which expect the data from type 'double'. The `ao.userData` cell array gives the possibility to save everything else.

5.1.4 Read-only Metadata Fields

```
ao.dimensions = int
```

```
ao.nChannels = int
ao.nSamples = int
ao.nBins = int
```

5.1.5 Operators

The overloaded operator make working with objects convenient. Please make sure you understand the different possibilities here.

```
ao'           % triggers ctranspose.m: transpose to frequency domain
ao.'          % triggers transpose.m: transpose to time domain
ao + ao       % triggers plus.m
ao * ao       % triggers mtimes.m: multiplication in frequency domain
ao .* ao      % triggers times.m: multiplication in time domain
ao / ao       % triggers mrdivide.m: division in frequency domain
ao ./ ao      % triggers rdivide.m: division in time domain
ao ^ n        % triggers mpower.m: exponentiate in frequency domain
ao .^ n       % triggers power.m: exponentiate in time domain
```

5.1.6 Vectors

These vectors are used as x-axis information for e.g. plots.

```
ao.freqVector % returns the discrete frequencies of the sampled spectrum
ao.timeVector % returns the discrete time values of the sampled time signal
```

5.1.7 Overloaded Standard Functions

```
plot          % directly plots an itaAudio
play          % plays an itaAudio
cast          % sets the field ao.dataType not ao.dataTypeOutput
conv          % convolves
sqrt          % square root
fft           % fast Fourier transformation
ifft          % inverse fast Fourier transformation
```

More functions are e.g. `abs`, `max`, `min`, `mean`, `median`, `conj`. Just try to experiment a bit with them. Functions can be triggered with `ao` as argument or separated by a point, i.e. `play(ao)` or `ao.play`. The function `cast` can only be triggered with two arguments, the `itaAudio` object and the data type, i.e. `cast(ao, 'single')`. Be careful with that, the `ao.dataTypeOutput` has to be set separately.

6 Plot Routines – Keyboard Shortcuts

These files are used for plotting purposes – for example, one can plot a spectrogram (`ita_plot_spectrogram`), a frequency/time domain plot (`ita_plot_spk` or `ita_plot_dat`) or set himself the characteristics of the plot using the options in each function. What is important to note in here is that once the plotting is finished, there are a series of keyboard shortcuts that can help us work on the figure itself. All shortcuts can be seen by typing `h` in the plot windows as shown in Figure 3.

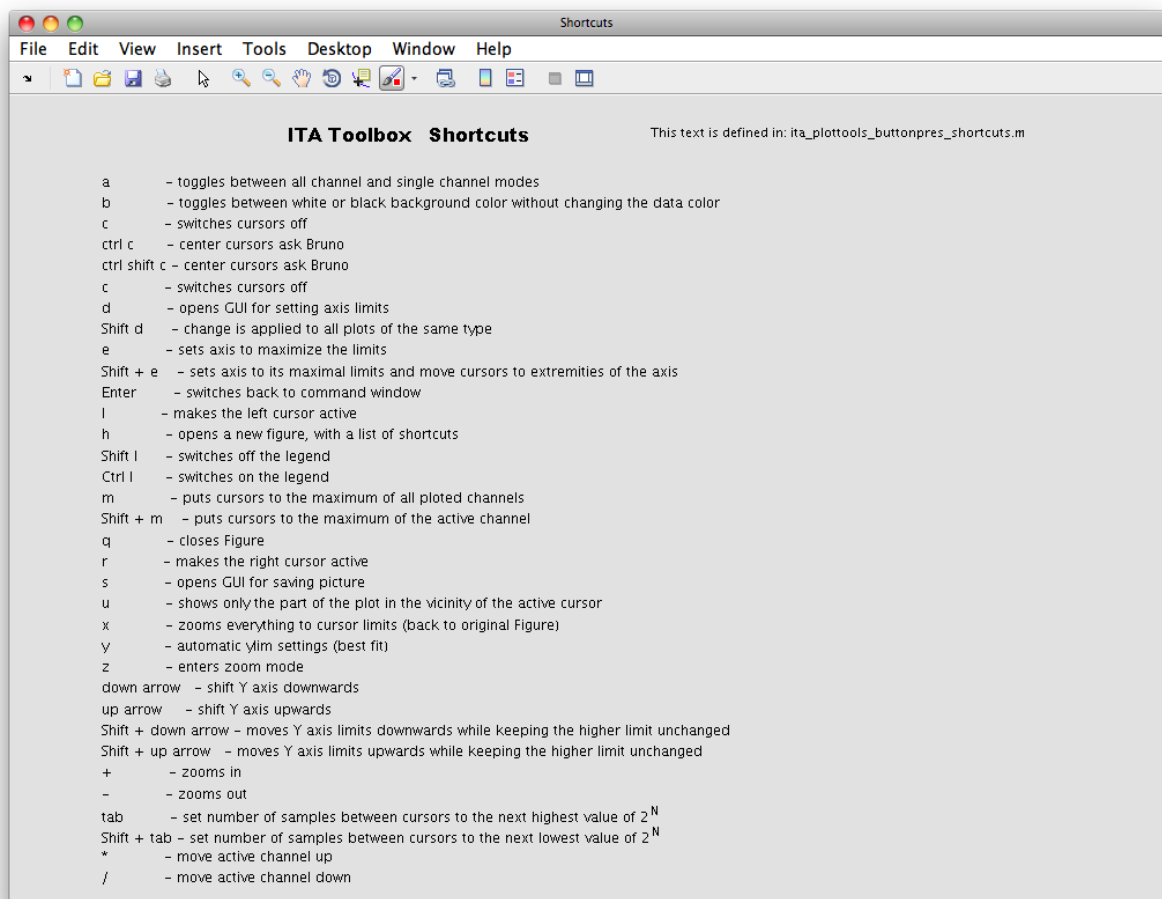


Figure 3: Overview of the keyboard shortcuts by pressing `h` in an ITA-TOOLBOX plot window

7 Graphical User Interface (GUI)

The ITA-TOOLBOX provides a GUI as a graphical frontend to the command line functions which can be called by using the entries in the menu line. You can call the GUI by typing `ita_toolbox_gui` or simply by plotting a variable. In the case of an empty workspace some data to start working on is loaded initially.

7.1 How to use it ?

The most important point is that all variables accessed or modified in the GUI are stored in your MATLAB workspace (base workspace). That means you can by any time switch between command-line functions or the GUI and work on with the same data.

The menu line of the ITA-Toolbox GUI starts after the MATLAB menu entries separated by a long dash (—) and it is structured as follows (see Figure 4):

- **ITA** Read, write, import and export data, store and export plots, change basic settings and preferences
- **Workspace** Here you can find the current variables from your workspace. Choose a variable by clicking on it to e.g. plot it. By hitting *refresh* you can get an updated list of the workspace variables. You can also change the name of the variables here.
- **Domain** Choose the type of plot, e.g. time or frequency domain, include phase or group delay or plot a spectrogram. The data is just plotted, it is not modified by changing the domain.
- **Edit** Modify your audio data, e.g. apply a time window, resample, Hilbert transform. Merge two variables to become one, or split channels of a variable to become separate variables. Change the meta data of your data, e.g. channel information, comment, coordinates.
- **Filtering** Apply band-pass, high-pass, low-pass or peak filters or calculate fractional octave bands.
- **Applications** If you have access or developed applications on your own, they might appear here with a fancy GUI.
- **Tools** Simple play-back of data or frequency generator.

- **Help** Access the online HTML-help of the ITA-Toolbox or open this Getting Started PDF.

7.2 Creating your own GUIs

Applications might add their GUI elements to the main ITA-Toolbox GUI. To start building you own GUI elements there are two ways to distinguish.

- Create a standalone GUI with `ita_parametric_GUI`. See `test_ita_parametric_GUI` for details.
- Add your function or standalone GUI to the main Toolbox GUI by entering information in a file similar to `ita_guiumentries`.

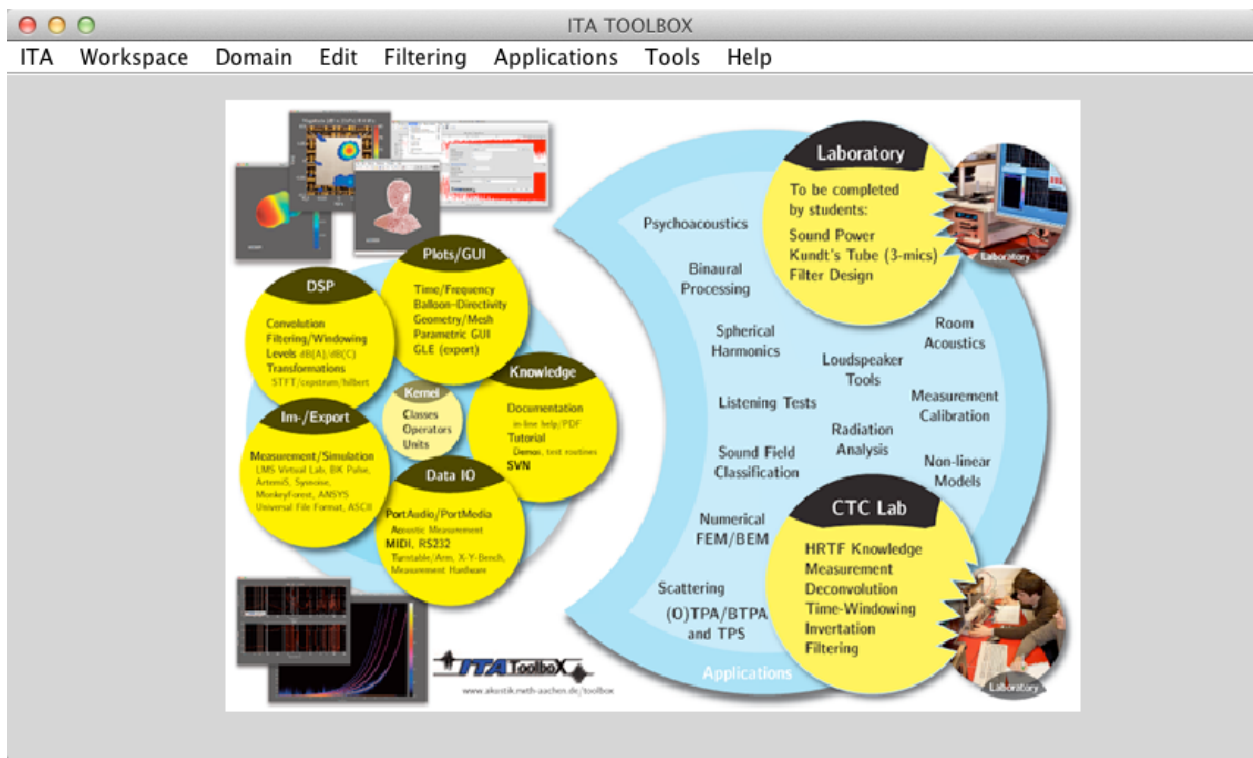


Figure 4: `ita_toolbox_gui`: A graphical user interface for many functions of the ITA-TOOLBOX

8 Measurements in a Nutshell

The ITA-TOOLBOX provides measurement functionality with the application **Measurement**. In order to make measurements as convenient as possible we developed some measurement classes and GUIs to guide you through the process of setting up signals and soundcard channels, calibration, measurement and evaluation.

Start by typing `MS = ita_measurement` to generate a measurement object of your choice which is later stored in the variable `MS` you specified in your workspace. The GUI looks similar to the one shown in Figure 7. The explanation of the different measurement types available is given in the following.

Note: Please be aware that measurement setups are derived from handle classes. If you copy the variable MATLAB only copies the pointer to the object and not the object itself. If you modify one object the copied variable has the same modifications as it is the exact same object.

- Signals – just record no playback
- Signals with Playback – playback and record simultaneously (no deconvolution)
- Transferfunction – Impulse response or Frequency response measurements (with deconvolution)
- Impedance – Loudspeaker electrical impedance measurement

In the following example we choose *transfer function* measurements. The GUI given in Figure 8 appears. Settings are explained with a short help text which appears by moving the mouse over the setting on the left hand side. The next small GUIs appear as shown in Figure 9 and Figure 10 to specify the measurement chain (more in ?? ??), which is used later on for calibration purposes. Now you have a measurement object to measure transfer functions right away. If you want to re-calibrate your equipment type `MS.calibrate` and a GUI guides you through each element as shown in Figure 11.

For measurements there are special classes for the measurement setup. More information is provided in the chapter about measurements.

- `itaMeasurementSetupSignals` – Only record signals
- `itaMeasurementSetupSignalsWithPlayback` – Record signals while playing another signal
- `itaMeasurementSetupTransferFunction` – Obtain impulse responses (deconvolution)
- `itaMeasurementSetupImpedance` – Measure electrical impedances

- **itaMSTF** – The new measurement class, but still under construction (beta).

An overview of these measurement classes along with the rest of the classes used inside the ITA-TOOLBOX is given in Figure 5.

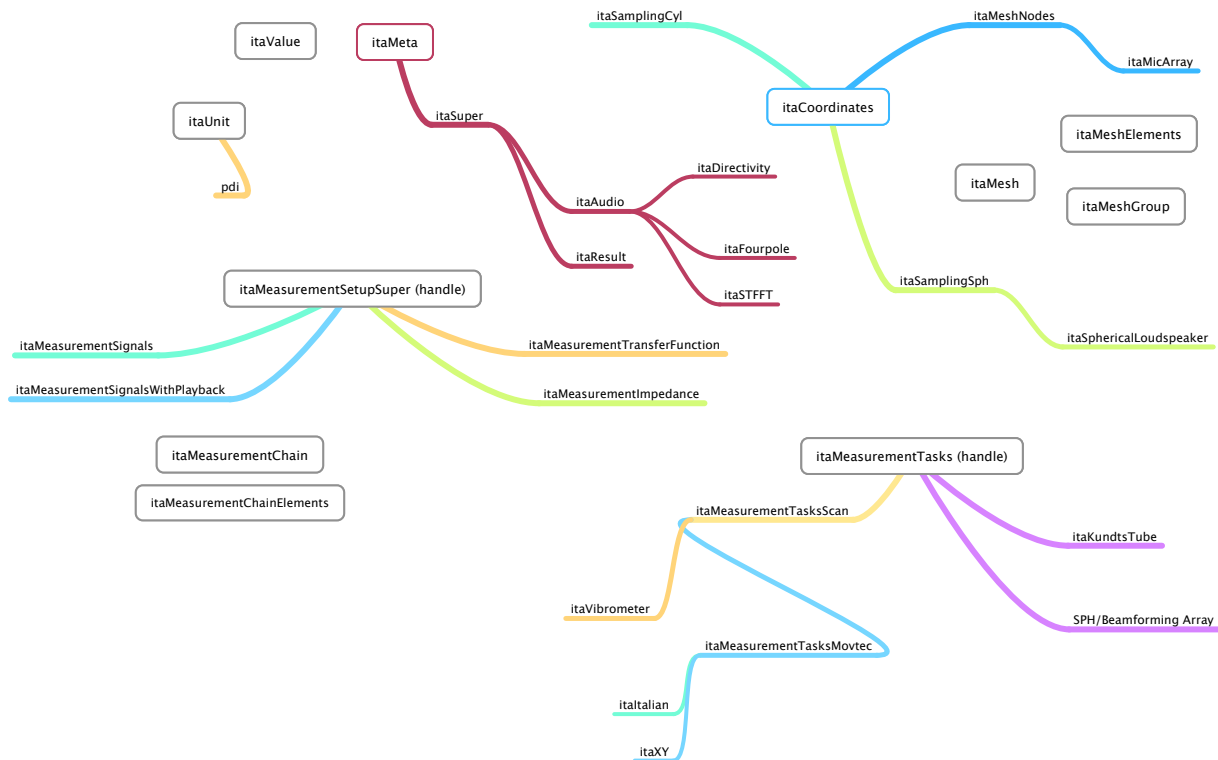


Figure 5: Overview of the classes and their relationship used in the ITA-TOOLBOX

8.1 Audio Objects – itaAudio

This type of data is used for equally sampled audio data in time or frequency domain. Due to this sampling a transformation between time and frequency domain is possible. For not equally sampled data use **itaResult**. Please have a look at doc **itaAudio** for more details!

8.2 Measurement Chain Concept

The conceptual measurement chain as depicted in Figure 6 is implemented by using **itaMeasurementChain** in **itaMeasurementSetup** for the input and if required for output measurement chain. Every block in the diagram can be represented for a single number value (**itaValue**) characterizing the transfer function of this element at a specific frequency. This is mostly used for the input measurement chain, as we calibrate by using a voltage calibrator (1 V RMS, 1 kHz).

The output measurement is only used in case audio is played back during the measurement. The **outputamplification** factor belongs to the Measurement Setup (class property). This value is always compensated after the measurement. The unit of this factor is **dBFS** meaning that for a value of 0 dBFS the excitation signal is directly send to the sound card. The factor can only be smaller than zero, e.g. -20dBFS.

That means the signal is attenuated by 20dB before it is sent to the sound card. After recording, the input is amplified by 20dB to compensate for this factor.

Back to the output measurement chain, where we use also single value numbers to describe the elements, but due to the measurement of this factors at the output we could also use spectra including phase. The current implementation uses only the absolute single number values at 1 kHz and compensates for the audio hardware delay by using an integer number of latency samples stored in the measurement setup. The loudspeaker equalization always takes advantage of using spectra, since loudspeakers have generally a strong dependence on frequency compared to the other elements of the input and output measurement chains.

You can access information inside the `itaMeasurementChain` by using `disp` of your measurement chain or by stepping inside using `MC(channel_idx).elements(1)`. The sensitivity of a channel can be calculated by using `MC(channel_idx).sensitivity`. Have a look at the class documentation inside the toolbox for more details.

A new functionality called `itaMeasuringStation` helps to save time by setting up the measurement chains on and on again for a specific computer with certain audio hardware. The measurement chains can be copied to other measurement setups.

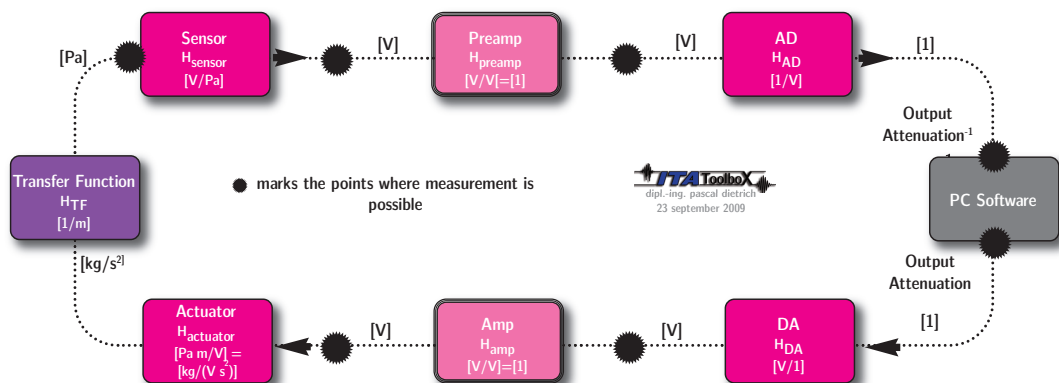


Figure 6: Measurement Chain Diagram for input, output chain and acoustic transducer.

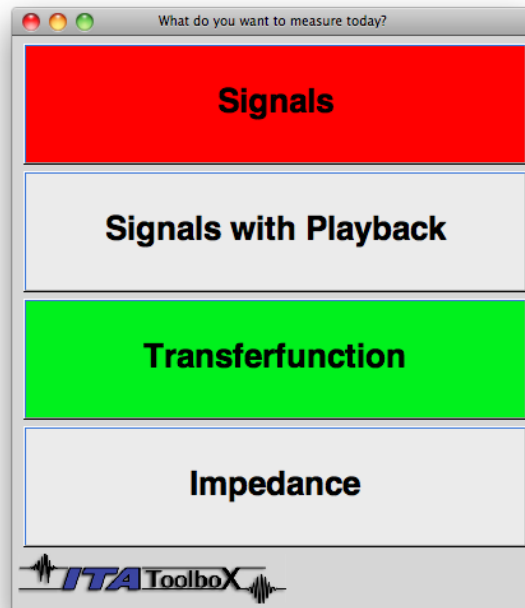


Figure 7: ita_measurement GUI to choose the type of measurement.

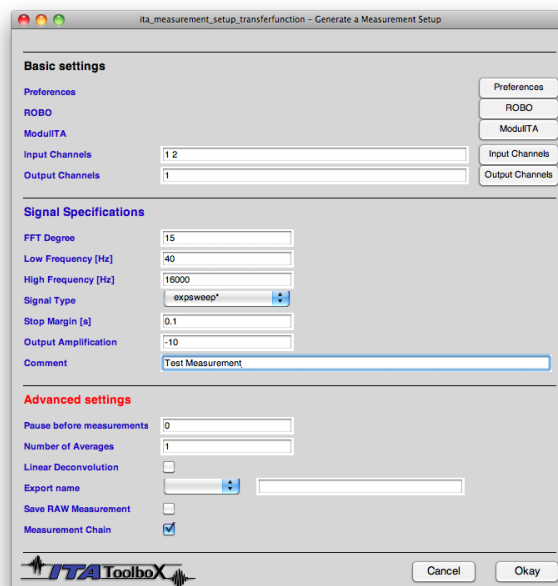


Figure 8: ita_measurement GUI to setup measurement signal, length and channels.

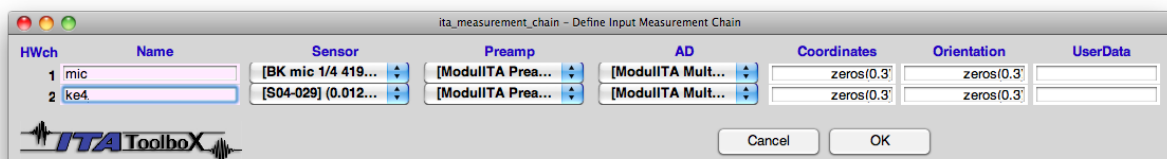


Figure 9: ita_measurement_chain GUI to define the input chain.

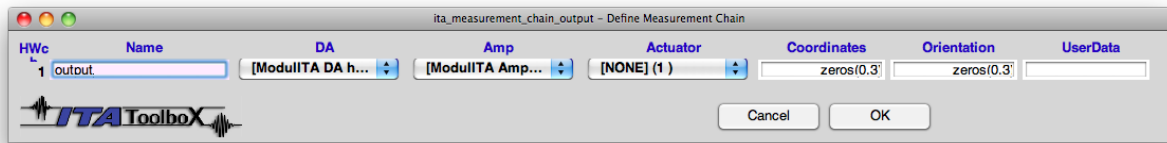


Figure 10: ita_measurement_chain GUI to define the output chain.



Figure 11: Calibration GUI for BK pressure microphone.

8.3 ASIO under Windows x64

On 64-bit Windows systems please make sure the Microsoft Visual C++ Redistributable 2010 is installed. You might see an error message about a module that cannot be found by using your sound card via portAudio or Midi then it is time to finally install this Redistributable.

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=a7b7a05e-6de6-4d3a-a423-37bf0912db84>

<http://www.microsoft.com/downloads/en/details.aspx?familyID=bd512d9e-43c8-4655-81bf-9350143d5867>

9 Official Stuff

9.1 Praise the ITA-Toolbox

If you really like the software and you are working with it to achieve your academic goals we really appreciate mentioning us in your acknowledgments. You can, e.g., copy and paste this text below:

Data analysis and post processing has been realized by using the ITA-Toolbox for MATLAB developed at the Institute of Technical Acoustics at RWTH Aachen University.

9.2 Literature

There are some papers dealing with the concept and applications of the ITA-TOOLBOX, which you can find online at: <http://www.ita-toolbox.org/contact.php>.

Have fun.