AUSTRIAN ACADEMY OF SCIENCES
ACOUSTICS RESEARCH INSTITUTE

Piotr Majdak, Katharina Pollack, David Meijer

The 4th AMT Workshop

Introduction

Graz, 5.9.2026

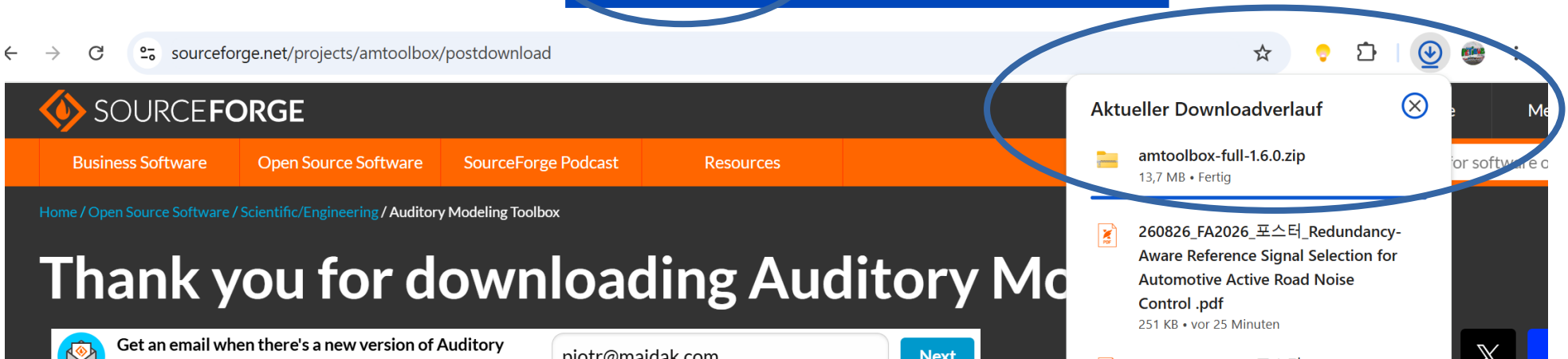
Version: 9/4/26

First step: Download the full package

- http://amtoolbox.org

Download

The easiest and recommended method is to download the '



The AMT Workshop: Structure and Goals

- What is the AMT and how can it be useful for you?
  - General AMT structure
  - Using AMT models
  - Using AMT parameters and functions
- Goals:
  - Do you know what is the AMT and its components?
  - Do you have a basic idea of what you can do with the AMT?
  - Are you able to install the AMT?
  - Are you able to run its models by writing own simulation scripts?

The AMT

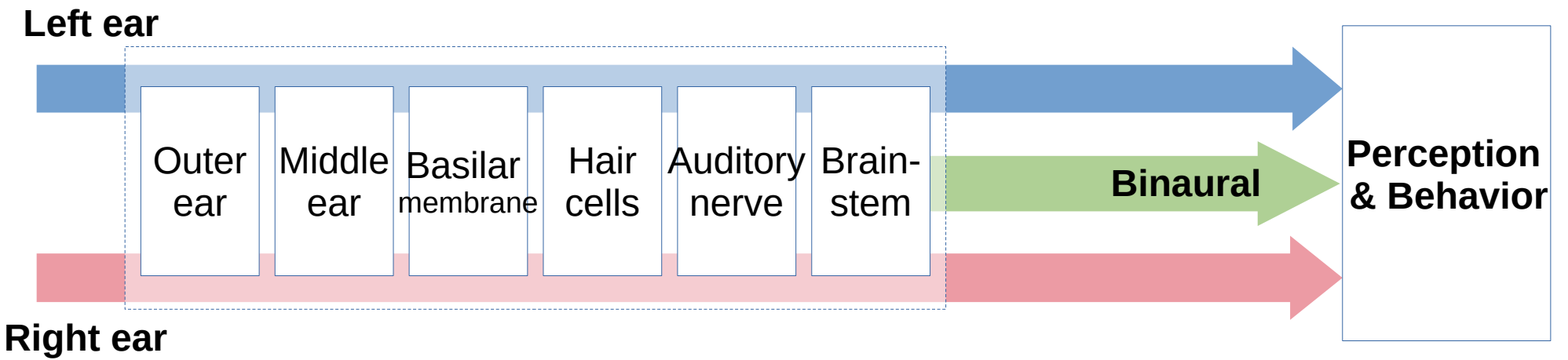
- An open-source and open-access toolbox for auditory modeling
  - A tool for reproducing scientific experiments with auditory models
  - A framework for developing new models
- Environment:
  - Core: Matlab/Octave
  - Models: (nearly) any programming language
- Software license:
  - Core: GPL version 3
  - Models: Multi licensing; (nearly) any GPL3-compatible license
- Link of a large amount of data and complex code
- Integration of other toolboxes useful in auditory research

The AMT

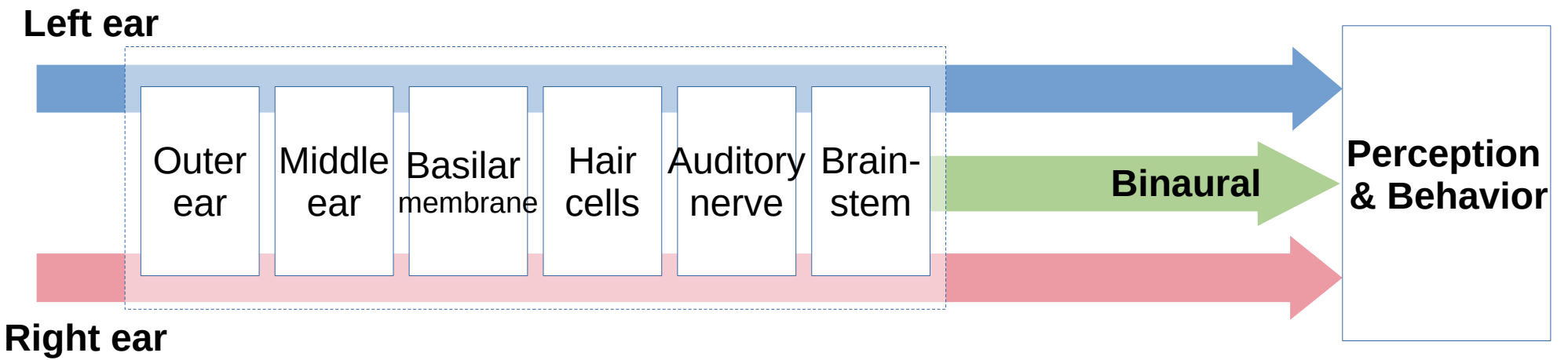
- 2009: Development initiated by Peter Søndergaard
- 2012: Development moved to ÖAW
- 2021: AMT 1.0 released
- Now: Over 60 auditory models, 40 GB of auditory data
- Two releases per year, with a focus on sustainability

- Community work: Most of the models “donated” to the AMT
- Increase the authors’ visibility and add citations

Typical stages of auditory models

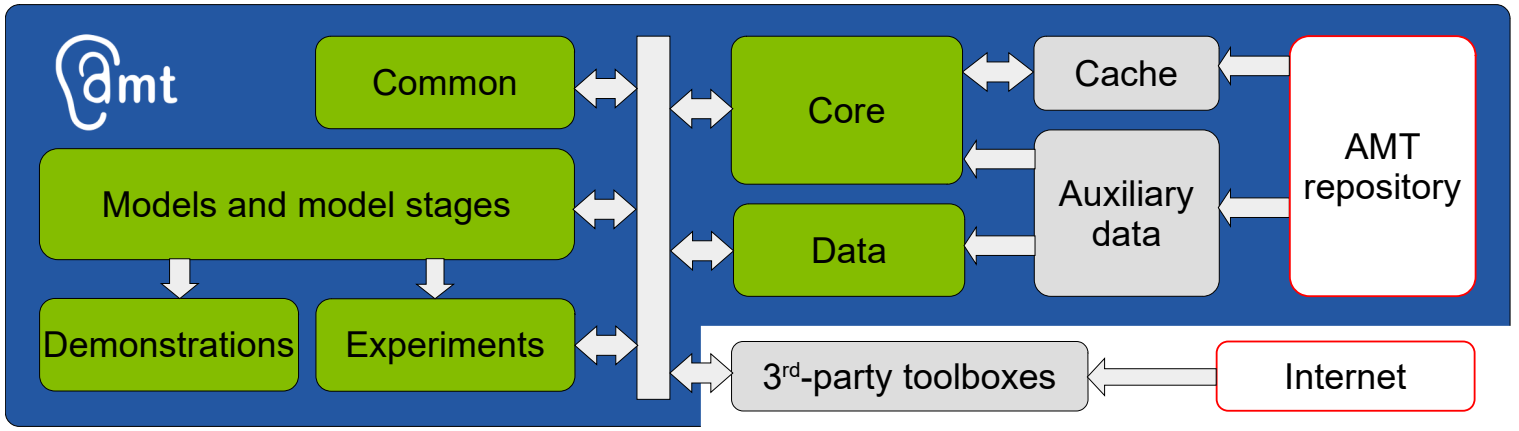


Typical stages of auditory models



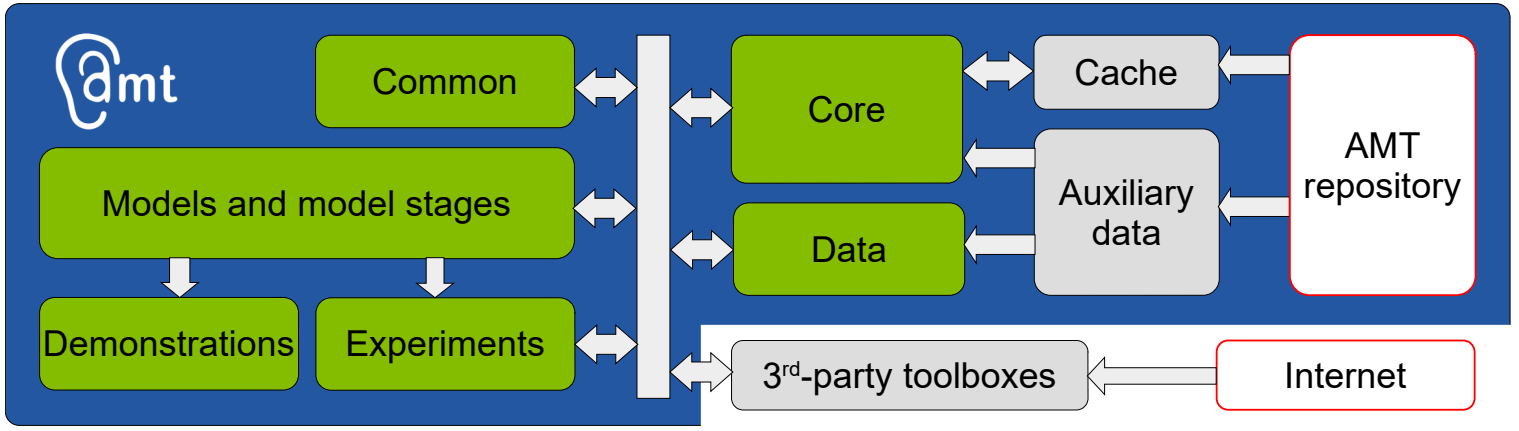
- Model ≠ Model implementation
- Models need data
- Model implementations often rely on toolboxes
- Some models use common functionality, some are stand-alone concepts

The Auditory Modeling Toolbox (AMT)



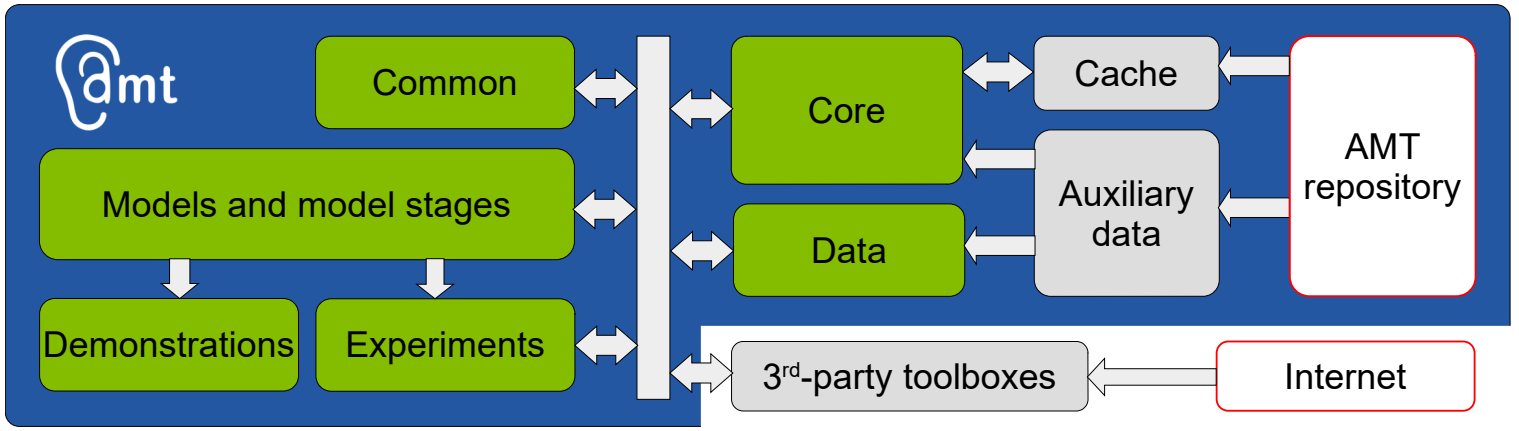
- Model ≠ Model implementation
- Models need data
- Model implementations often rely on toolboxes
- Some models use common functionality, some are stand-alone concepts

The Auditory Modeling Toolbox (AMT)



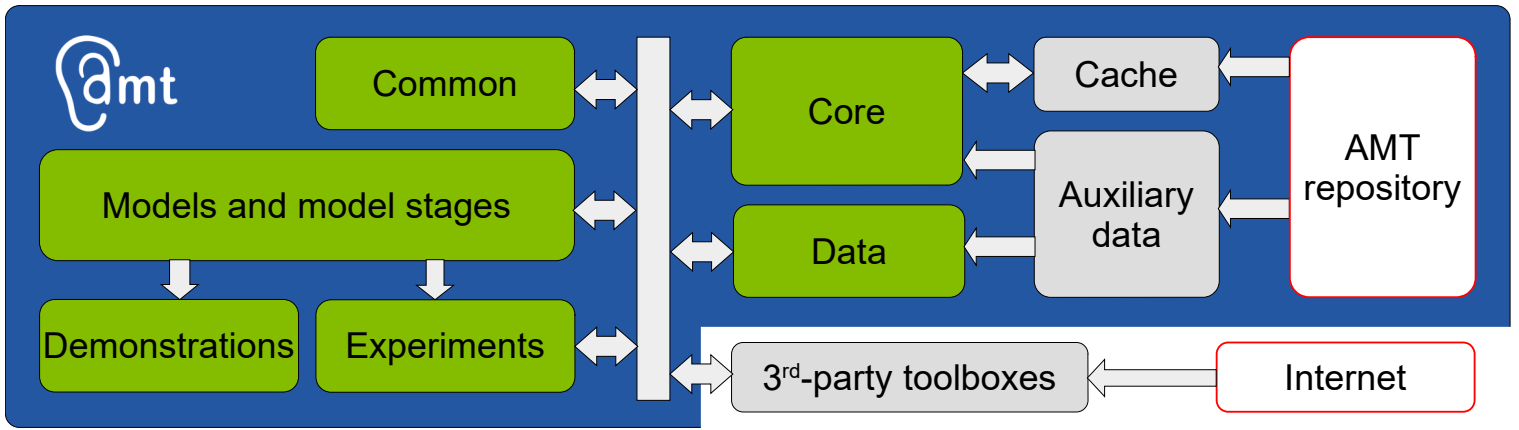
- Model ≠ Model implementation: Baumgartner et al. (2014) → `baumgartner2014`

The Auditory Modeling Toolbox (AMT)



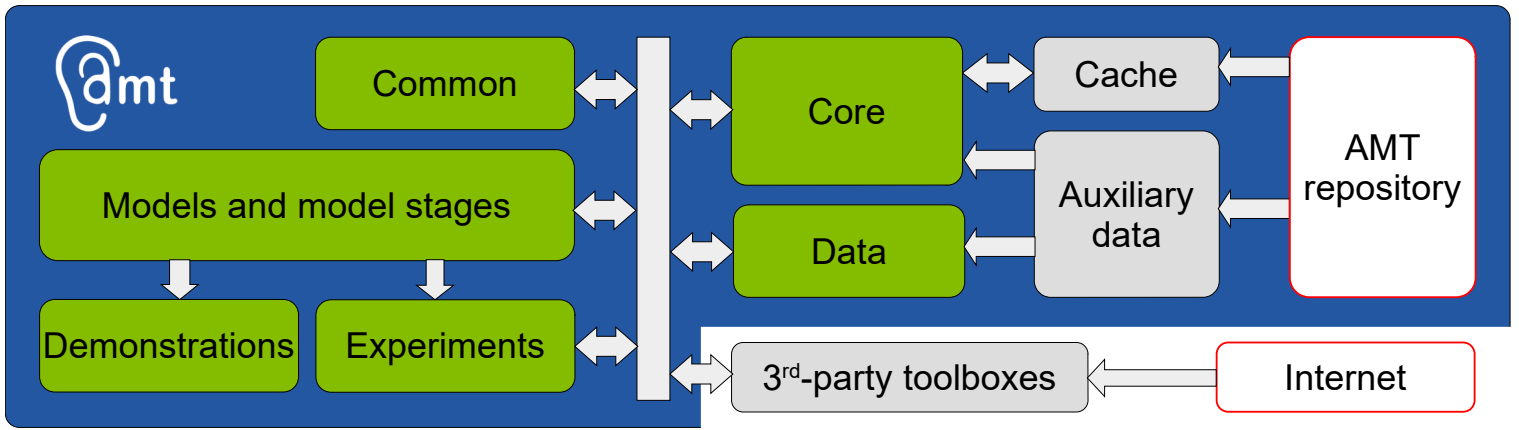
- Model ≠ Model implementation: Baumgartner et al. (2014) → `baumgartner2014`
- Models need data: Auxiliary data
  - Dedicated function: → `data_baumgartner2014`
  - Direct: → `amt_load('baumgartner2014',..)`

The Auditory Modeling Toolbox (AMT)



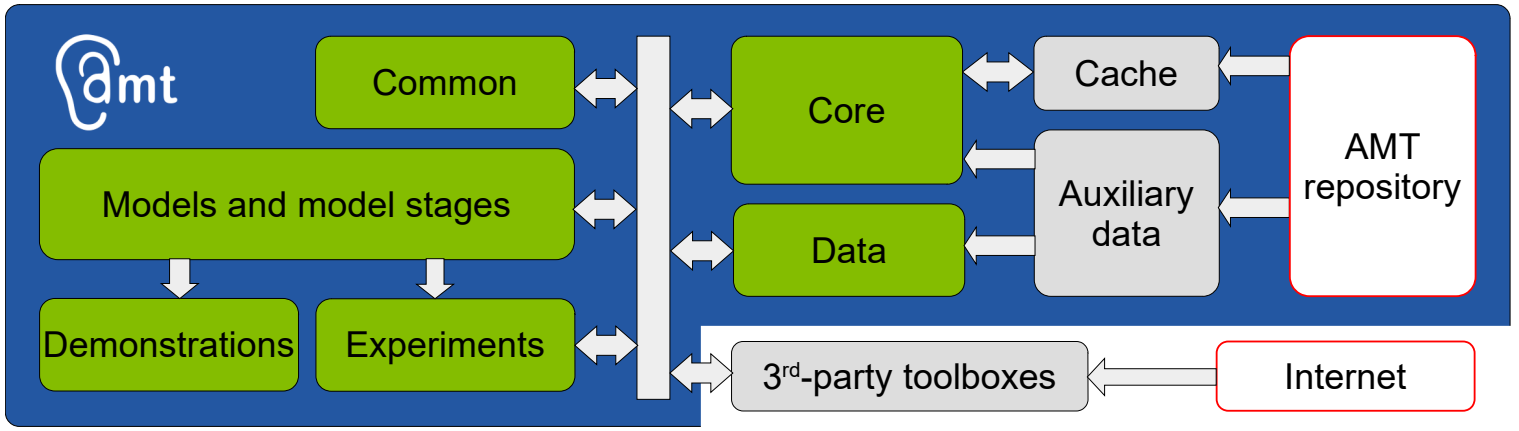
- Model implementations often rely on toolboxes: Third-party toolboxes
  - Large Time Frequency Analysis Toolbox (LTFAT)
  - *Optional*: Spatially Oriented Format for Acoustics (SOFA) Toolbox, Sound Field Synthesis (SFS) Toolbox, Circular Statistics (CircStat) Toolbox, Binaural Spherical Harmonics (BinSH) Toolbox

The Auditory Modeling Toolbox (AMT)



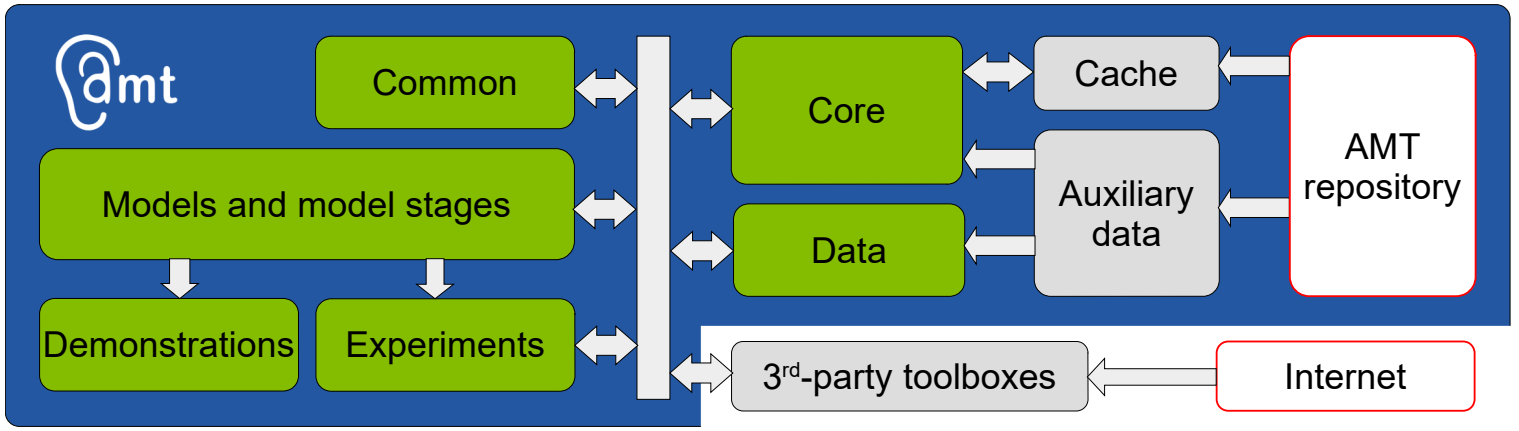
- Core functions: the main power of the AMT
  - Storing pre-calculated data: Cache → `amt_cache`
  - Data handling: Auxiliary data → `amt_load`
  - Simulation of experiments: → `amt_emuexp`

The Auditory Modeling Toolbox (AMT)



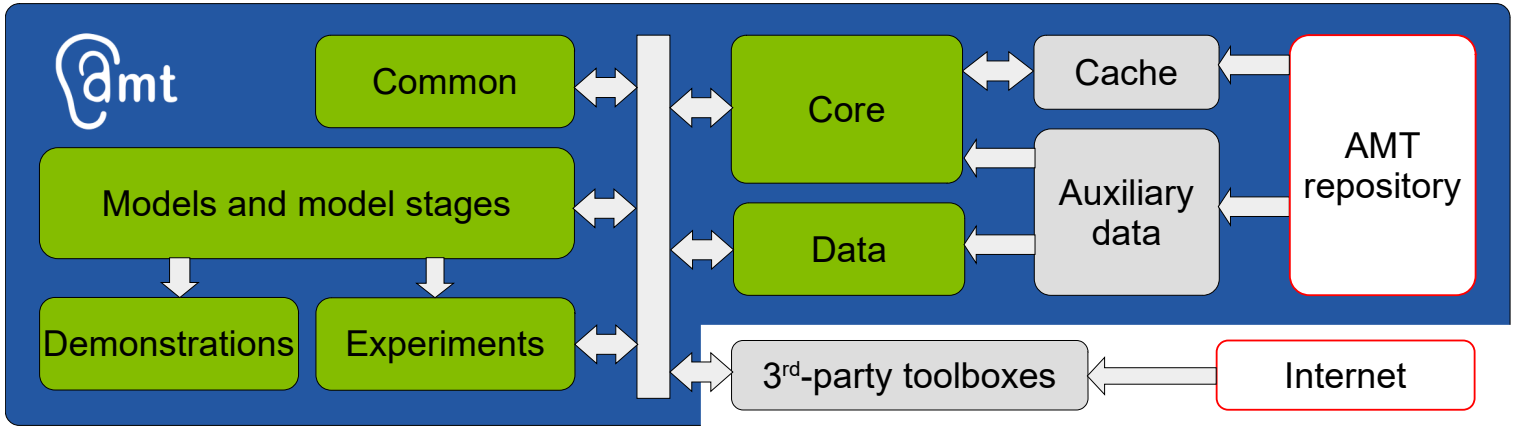
- Models
  - → `breebaart2011`
- Model stages:
  - → `breebaart2011_outmiddlefilter`
  - → `breebaart2011_eicell`
  - → `breebaart2011_centralproc`

The Auditory Modeling Toolbox (AMT)



- Common functions:
  - → `scaletodbspl`
  - → `erb2fc`
  - → `middleearfilter`
  - → `gammatone`
  - → `itdestimator`
  - → `adaptloop`

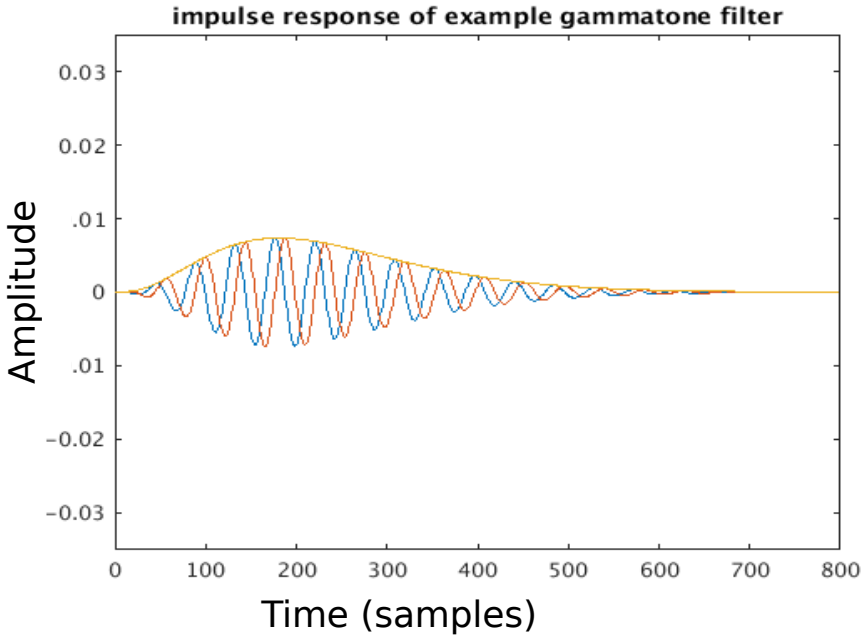
The Auditory Modeling Toolbox (AMT)

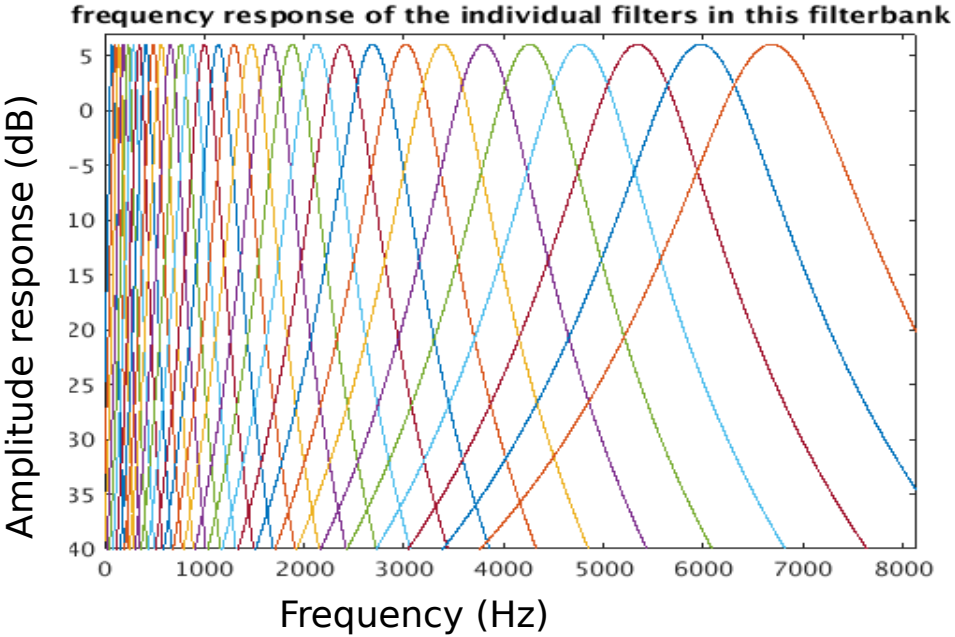


- Demonstrations: Simple scripts demonstrating the functionality
  → `demo_baumgartner2014`

Demonstrations

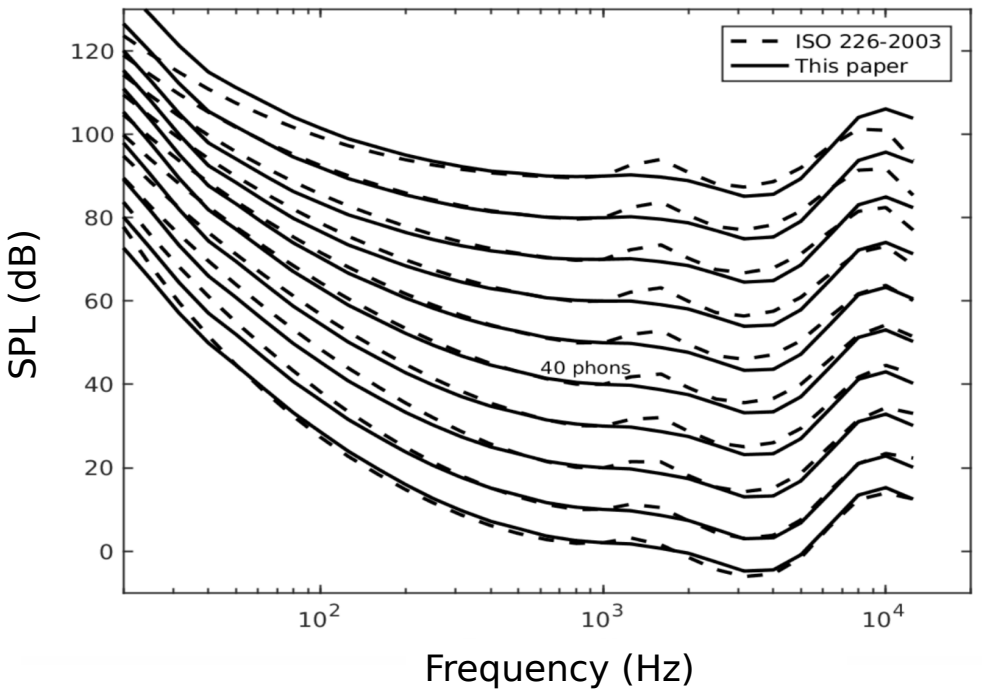
```
demo_hohmann2002;
```





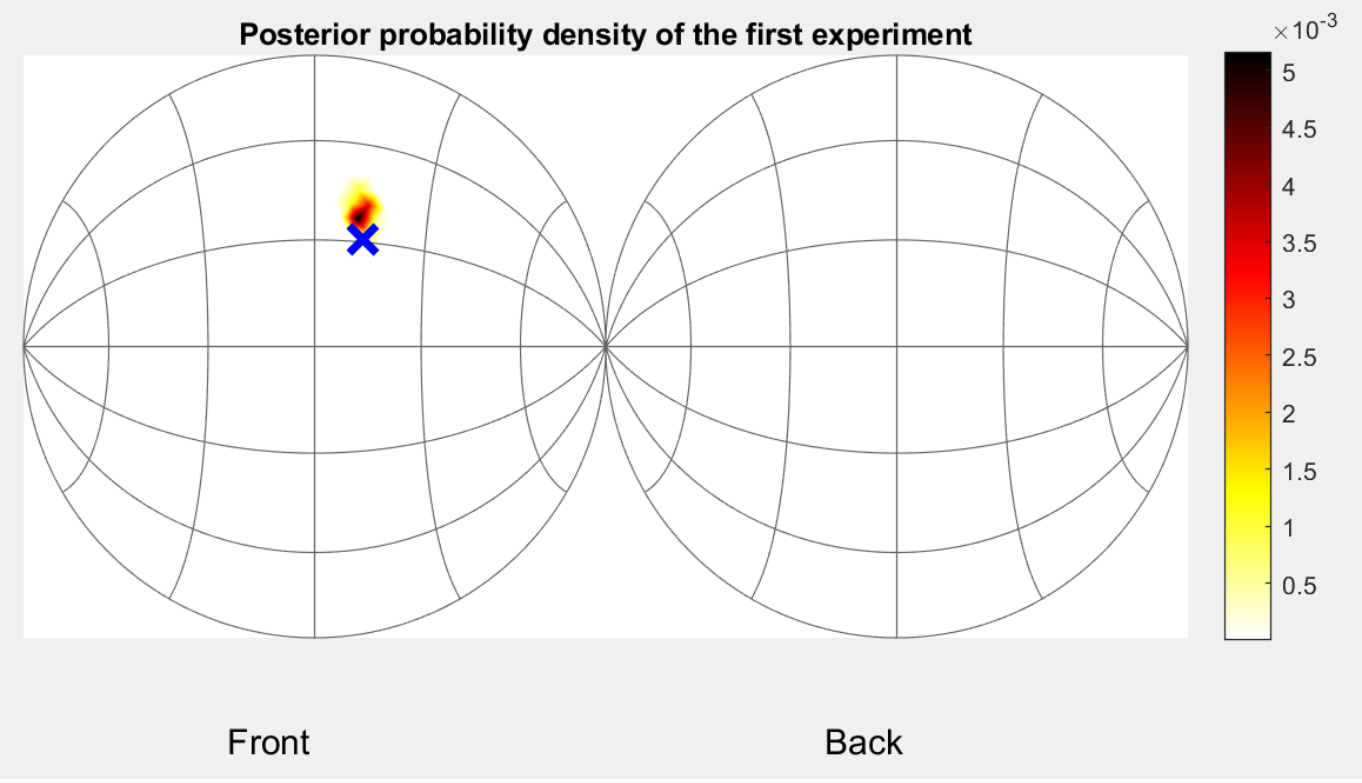
Demonstrations

```
demo_chen2011;
```

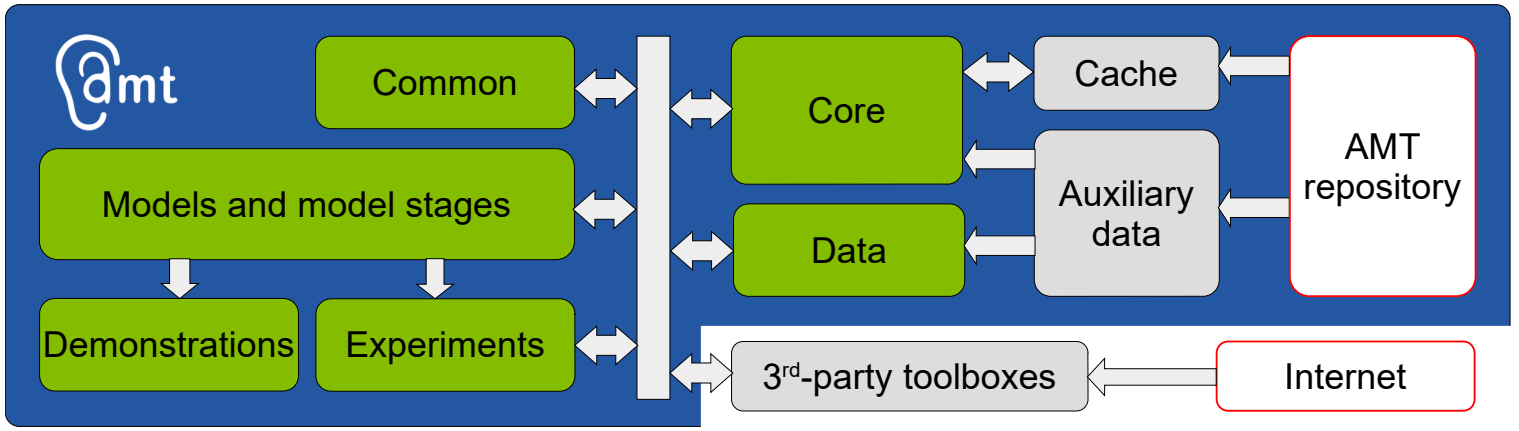


Demonstrations

```
demo_reijniers2014;
```



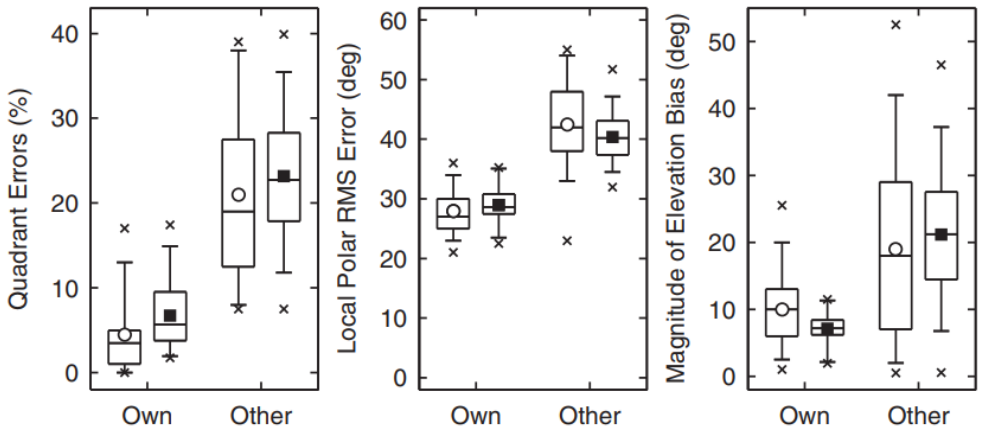
The Auditory Modeling Toolbox (AMT)



- Demonstrations: Simple scripts demonstrating the functionality
  → `demo_baumgartner2014`
- Experiments: Functions reproducing published results
  → `exp_baumgartner2014('fig9')`

Experiments

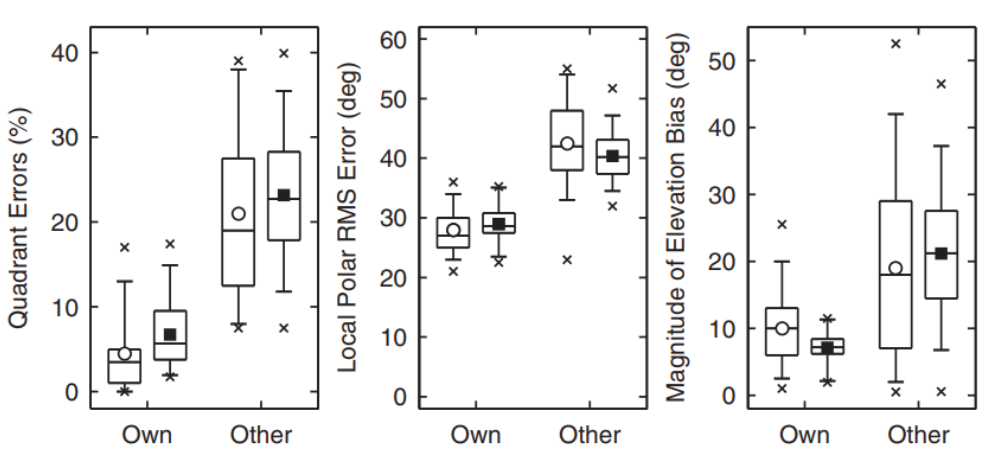
Screenshot from Figure 9
(Baumgartner et al., 2014):



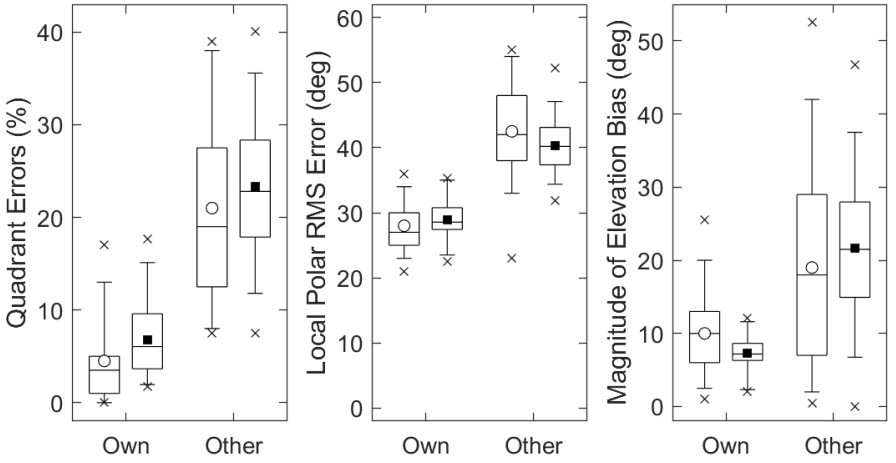
Baumgartner, Majdak, Laback (2014) „Modeling sagittal-plane sound localization“, J Acoust Soc Am 136: 791-802

Experiments

Screenshot from Figure 9 (Baumgartner et al., 2014):



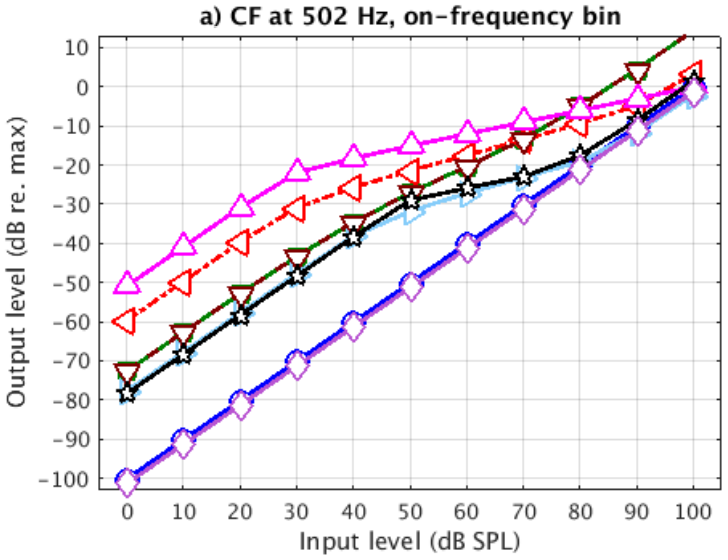
Output of `exp_baumgartner2014('fig9');`



Baumgartner, Majdak, Laback (2014) „Modeling sagittal-plane sound localization“, J Acoust Soc Am 136: 791-802

Experiments

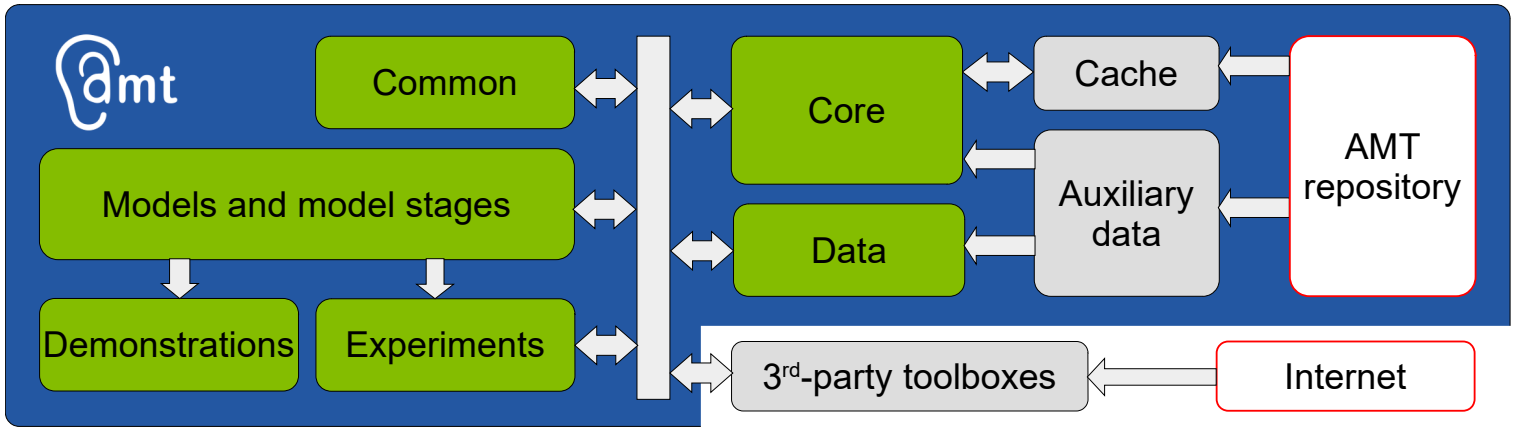
Output of
`exp_osses2022('fig5');`



- Blue: Dau et al. (1997)
- Orchid: Osses and Kohlrausch (2021)
- Green: Zilany et al. (2014)
- Maroon: Bruce et al. (2018)
- Red: Verhulst et al. (2015)
- Skyblue: Verhulst et al. (2018)
- Magenta: King et al. (2019)
- Black: Relano-Iborra et al. (2019)

Osses et al. (2022) „A comparative study of eight human auditory models of monaural processing“, Acta Acustica, 6: 17.

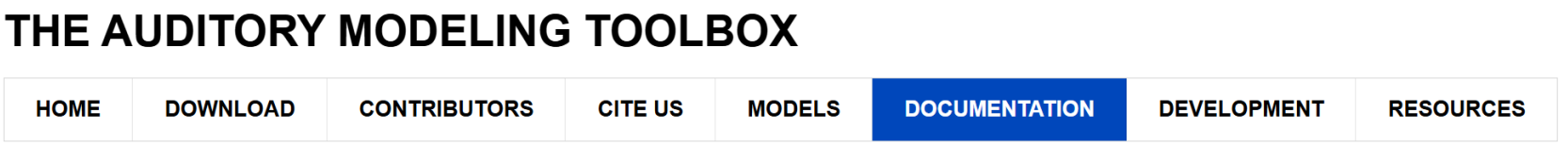
The Auditory Modeling Toolbox (AMT)



- Code: Files structured in directories
- Special directories:
  - auxdata
  - cache
  - defaults
  - mat2doc

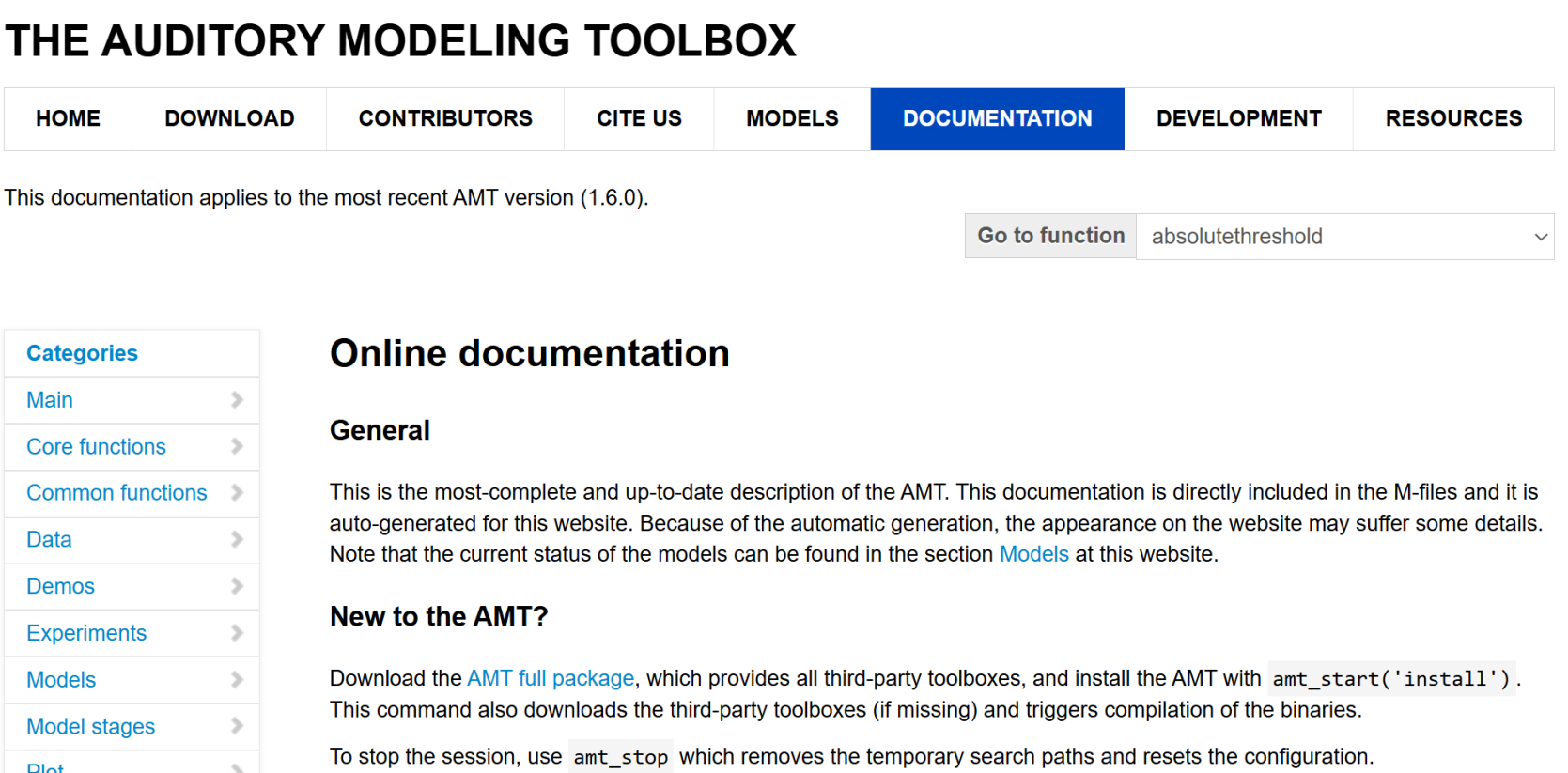
Documentation

- Website: http://amtoolbox.org



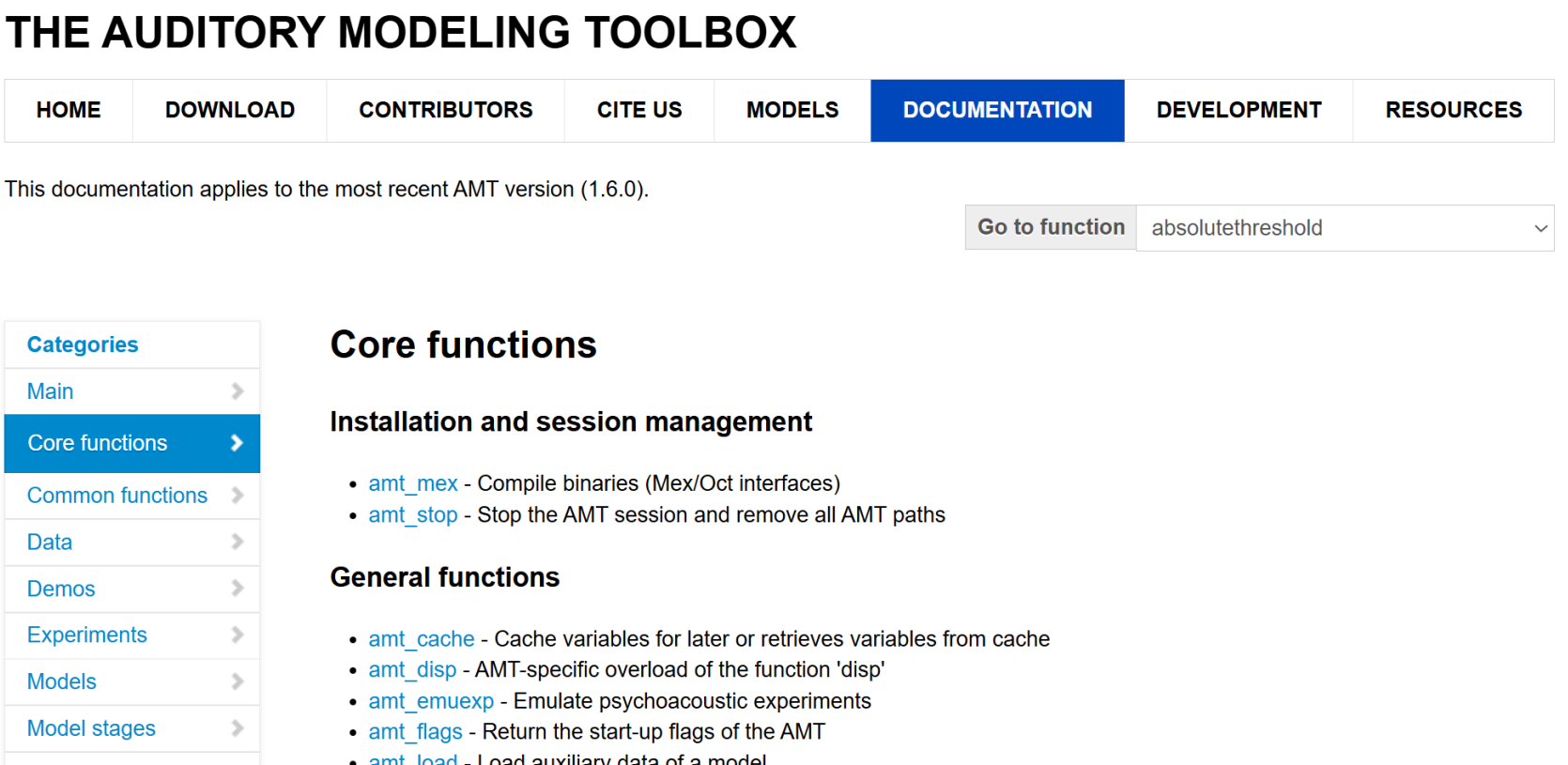
Documentation

- Website: http://amtoolbox.org



Documentation

- Website: http://amtoolbox.org



Documentation

- Website: http://amtoolbox.org

THE AUDITORY MODELING TOOLBOX

HOME | DOWNLOAD | CONTRIBUTORS | CITE US | MODELS | **DOCUMENTATION** | DEVELOPMENT | RESOURCES

This documentation applies to the most recent AMT version (1.6.0).

View the code

Go to function: amt_configuration

Categories

- Main
- Core functions
- Common functions
- Data
- Demos
- Experiments
- Models
- Model stages

AMT_CONFIGURATION

Get and set the configuration of the current AMT session

Usage:

```
[flags, kv] = amt_configuration;
[flags, kv] = amt_configuration('cacheURL', cU);
[flags, kv] = amt_configuration('silent');
[flags, kv] = amt_configuration('silent', 'normal');
```

Description:

Documentation

- Website: http://amtoolbox.org

THE AUDITORY MODELING TOOLBOX

HOME | DOWNLOAD | CONTRIBUTORS | CITE US | MODELS | DOCUMENTATION | DEVELOPMENT | RESOURCES

This documentation applies to the most recent AMT version (1.6.0).

View the help

Go to function: amt_configuration

Categories
- Main
- Core functions
- Common functions
- Data
- Demos
- Experiments
- Models
- Model stages

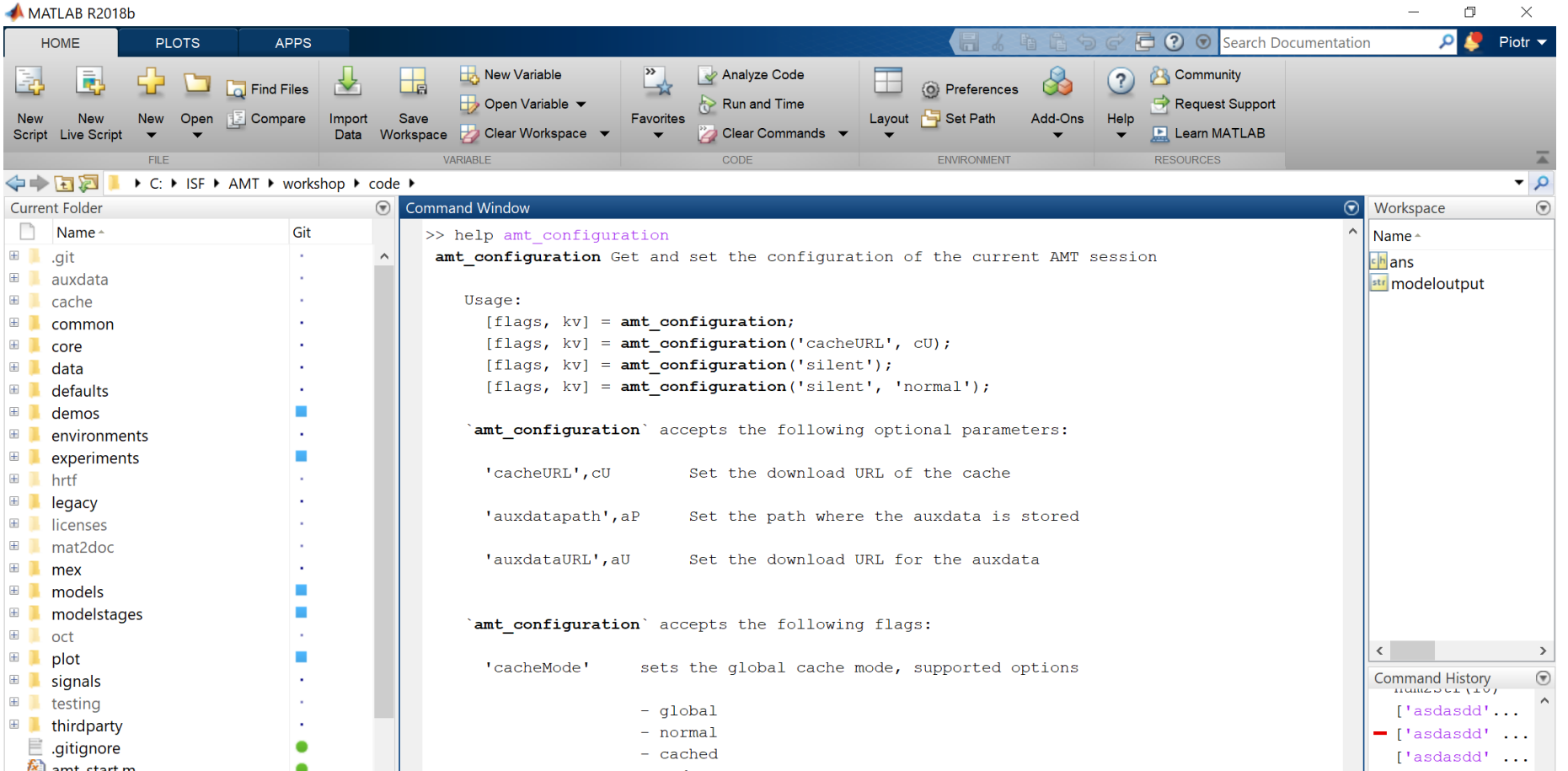
AMT_CONFIGURATION

Get and set the configuration of the current AMT session

Program code:

```
function [flags,kv]=amt_configuration(varargin)
%AMT_CONFIGURATION Get and set the configuration of the current AMT session
%
%   Usage:
%     [flags, kv] = amt_configuration;
%     [flags, kv] = amt_configuration('cacheURL', cU);
```

Documentation



Documentation

- Website: http://amtoolbox.org
  - Compiled version: the website
  - Code version
- In-code documentation
  - Syntax: Similar to reStructuredText
  - Compiler: Based on mat2doc
  - Automatic creation of figures and results
  - Links to the publications

EXP_TABUCHI2016 - Results from Tabuchi et al. (2016)

Usage:

```
data = exp_tabuchi2016(flag)
```

Description:

`exp_tabuchi2016(flag)` reproduces figures of the study from Tabuchi et al. (2016).

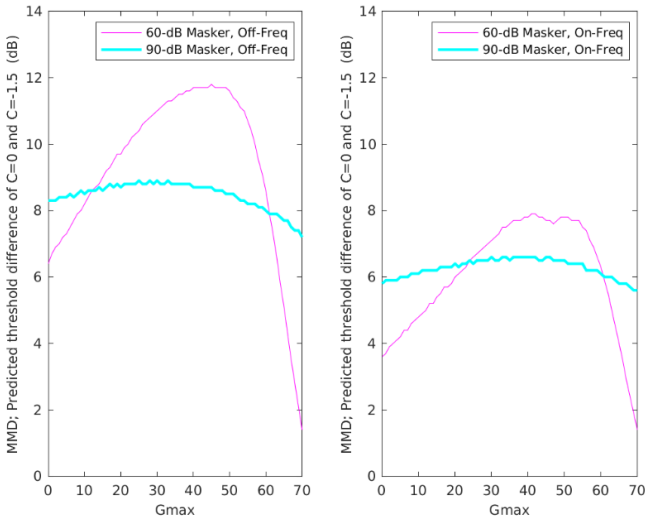
The following flags can be specified

`'fig6'` Reproduces the lower panel of Figure 6

Examples:

To display Fig.6 use :

```
exp_tabuchi2016('fig6');
```



References:

H. Tabuchi, B. Laback, T. Necciari, and P. Majdak. The role of compression in the simultaneous masker phase effect. *The Journal of the Acoustical Society of America*, 140(4), 2016. [DOI]

Categories

- Main
- Core functions
- Models
- Model stages
- Demos
- Experiments
- Common functions
- Plot
- Signals
- Data
- Auxdata
- Cache

See also

- tabuchi2016

EXP_TABUCHI2016 - Results from Tabuchi et al. (2016)

Program code:

```
function varargout = exp_tabuchi2016(varargin)
%EXP_TABUCHI2016 Results from Tabuchi et al. (2016)
%   Usage: data = exp_tabuchi2016(flag)
%
%   EXP_TABUCHI2016(flag) reproduces figures of the study from
%   Tabuchi et al. (2016).
%
%   The following flags can be specified
%
%     'fig6'    Reproduces the lower panel of Figure 6
%
%   Examples:
%   ---------
%
%   To display Fig.6 use :
%
%     exp_tabuchi2016('fig6');
%
%   See also: tabuchi2016
%
%   References:
%     H. Tabuchi, B. Laback, T. Necciari, and P. Majdak. The role of
%     compression in the simultaneous masker phase effect. The Journal of the
%     Acoustical Society of America, 140(4), 2016.
%
%
%   Url: http://amtoolbox.org/amt-1.5.0/doc/experiments/exp_tabuchi2016.php


%   #Author: Hisaaki Tabuchi (2022)
%   #Author: Clara Hollomey (2023): adaptations for AMT
%   #Author: Joonas Guevara (2023): implemented ploting


definput.import = {'amt_cache'};
definput.flags.type = {'missingflag','fig6'};

[flags,~]  = ltfatarghelper({},definput,varargin);
```

Documentation

- Website: http://amtoolbox.org
  - Compiled version: the website
  - Code version
- In-code documentation
  - Based on mat2doc
  - Syntax: Similar to reStructuredText
  - Automatic creation of figures and results
  - Links to the publications
- Model list and ratings

Documentation

THE AUDITORY MODELING TOOLBOX

Models included in the AMT (latest release, 1.5.0)

In order to describe the quality of the models available in the AMT, we rate the implementation of every model by considering its **source code** and **documentation**. We also rate the models in terms of their **verification**, i.e., we rate the results of the implementation versus the results shown in the corresponding publication. The comparison is done within the experiments implemented in the exp_ functions. In the best case, the experiments produce the same results as in the publication - up to some minor layout issues in the graphical representations.

The following table provides an overview of the available models, their documentation, code, and verification status.

Peripheral models	Function	Doc	Code	Verification
Gammatone filterbank	gammatone	✔	✔	✔
Linear filtering for monaural masking (basic)	dau1996	○	✔	?
Linear filtering for monaural masking (improved)	dau1997	○	✔	?
Invertible Gammatone filterbank	hohmann2002	✔	✔	✔
Dual-resonance nonlinear filterbank (DRNL)	lopezpoveda2001	✔	✔	✔
Fast acting compression (CARFAC) model	lyon2011	✔	✔	?

Documentation

- Website: http://amtoolbox.org
  - Compiled version: the website
  - Code version
- In-code documentation
  - Based on mat2doc
  - Syntax: Similar to reStructuredText
  - Automatic creation of figures and results
  - Links to the publications
- Model list and ratings
- Other resources

Getting started

- In you file explorer: Unzip amtoolbox-1.6.0.zip to a directory
- In Matlab/Octave:
  - Go to that directory/amtoolbox-1.6.0
  - Start the AMT: `amt_start;`

```
Starting packages and toolboxes:
* Signal Processing Toolbox: Found.
* Statistics and Machine Learning Toolbox: Found.
* Deep Learning Toolbox: Not found.
* Python Environment: Python 3.13.14.
* LTFAT: LTFAT version 2.6.0 (for AMT). Copyright 2005-
* SOFA: SOFA Toolbox for Matlab/Octave 2.2.0. Copyright
* SFS: Toolbox version 2.5.0 loaded
* CIRCSTAT: Circular statistics toolbox found.
* BINAURALSH: BinauralSH toolbox loaded.
* BADS: Bayesian Adaptive Direct Search toolbox found.
```

- Help: `doc amt_start`
- Stopping the AMT: `amt_stop;`

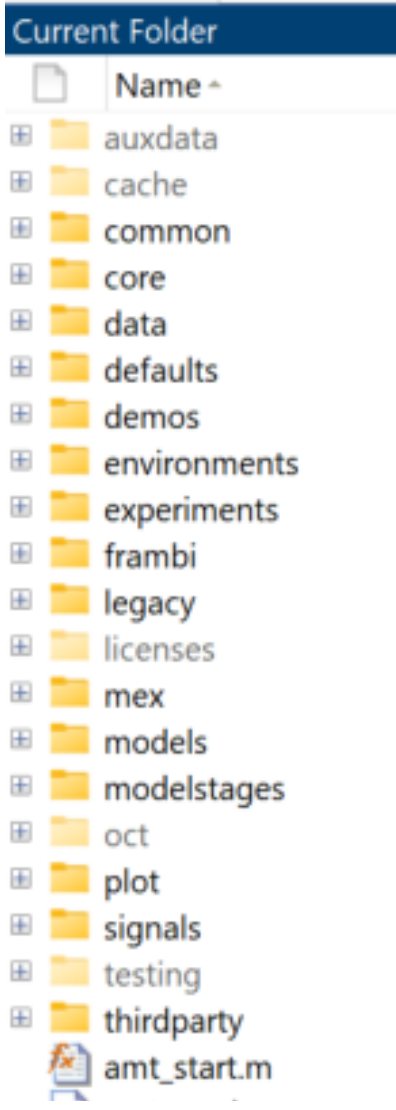
Folder structure

- models: The models! Just one file per model.
- modelstages: Larger models have more files: model stages.

- experiments: Code to compute model's results (e.g., in publications).
- demos: Scripts(!) show-casing the basic functionalities of a model.

- common: Useful functions for hearing related computations.
- data: Code providing published data (used in experiments, demos).
- defaults: Definition of parameters and flags an their default values.
- plot: Code to generate complex visualizations.
- signals: Code to generate generic or specific signals.

- auxdata: Local storage of auxiliary data.
- cache: Local storage of cached data.
- thirdparty: 3rd-party toolboxes like SOFA or LTFAT.



First tests

- Testing the environment:
  - Simple test: `demo_absolutethreshold`
  - SOFA/auxdata/internet test: `demo_baumgartner2014`
  - C-compiler test: `demo_zilany2014`
  - *(Python test: `demo_verhulst2012`)*
- Sneak into the code: `edit demo_baumgartner2014`