Software design of the AM toolbox and the AABBA project

November 27, 2009

1 Goals of the software

- High quality code: Easy to read and maintain
- High quality documentation in the code: This is the best place to keep the documentation, as it is easier to keep up to date when the code changes. The documentation can be extracted in HTML and TEX formats.
- Simplicity: Eliminate unneccesary configuration options. Always choose the simplest possible solution. This makes it much easier to read and modify the code.

2 Directory structure

2.1 toolbox

This directory contains a general toolbox relevant for building auditory models etc. Example contents:

- Filters: Gammatone, roex.
- Conversion between auditory scales: ERB, Mel, Bark

2.2 models

Model stages and full models

2.3 humandata

Function returning recorded (and published) human data

2.4 testing

Test scripts

2.5 reference

There are genrally two categories of files that go here:

1. Very simple implementations of concepts that are easy to read, but perhaps very slow or not general enough etc.

2. Old code that is trusted but not necessarily easy to read or efficient.

2.6 experiments

Descriptions of experiments.

3 Input and output

3.1 Data in general

It is generally assumed that all data is a regular sampling of a continuous phenomena at a certain sampling frequency `fs`. Many routines will therefore ask for both a signal and a sampling frequency. By the same reasoning, frequencies are usually specified in Hz because we assume the sampling frequency to be known (so don't do normalized frequencies between 0 and 0.5 as Matlab sometimes does).

3.2 Definition of the input to the model

If nothing else is noted, this should be the description of sound input to any model:

An acoustical signal is representated by a column vector of numbers. The numbers are obtained by sampling the air pressure of the acoustical signal at a constant sampling rate. The numbers are scaled such that an acoustical signal with a level of 100 dB SPL corresponds to a digital signal with an RMS value of 1.

XXX Is this sufficiently strict for a binaural model?

3.3 General Data structures

It is probably impossible to list the relevant data structures before developing the software, but the few ones listed here should always be used.

- A mono-signal is a column vector.
- A stereo-signal is a 2-column matrix.
- A multi-channel signal stemming from a filterbank has time as first dimension, channel number as second, and original signal channel (left/right) as third dimension.

3.4 Specific data strucures

- The output from the modulation filterbank has time as first dimension, frequency channel number as second, modulation channel number as third and original signal channel (left/right) as fourth.

4 General coding standards

- The same parameter should have the same name across all files. See the section on common variable names.

4.1 Matlab specific coding standards

- A variable should never have the same name as an already existing function is Matlab. This makes the code easier to read and less prone to errors.
- All function names in Matlab should be lowercase. This avoids a lot of confusion because some computer architectures respect upper/lower casing and others do not. Furthermore, function names are traditionally written in uppercase in Matlab documentation.
- All variable names should be lower case.
- It is not allowed to use underscores in variable or function names. They are reserved for structural purposes, i.e. as in `demo_gammatone` or `test_dau96`.
- Models are named after the paper in which they first appeared, as in dau96, or after their commonly accepted name if theres is no doubt as to which model is the name refers.
- *As much as possible*, functions are named after the function they perform, rather than the algorithm they use, or the person who invented it.
- No global variables. Global variables makes it harder to debug, and the code cannot be parallized.
- Global configurations switches must not alter the output of functions. This introduces bugs if people give the code to each other and they forget to tell what options was used. Such bugs can be very hard to find. They should only be used for harmless things like altering the apperaence of plots.

4.2 C specific coding standards

- Variables name are allowed to be both lower and upper case. This convention is called *camel casing*, see http://en.wikipedia.org/wiki/CamelCase. This should be used with extreme care:
  - The benefit is that it makes long variable name easier to read.

– The downside is that the user suddenly has to remember the casing of the variable name, instead of just the name itself.

5 Variable names

In MATLAB: Never use `i` or `j` as a variable name, as they are used for the imaginary unit. This creates a great deal of confusion when reading other peoples code. Please use `ii` and `jj` instead, or something completely different. Using `i` and `j` are allowed in C, which does not have an imaginary unit.

The following is a list of common variables.

`insig` Input signal

`outsig` Output signal

`inoutsig` Some simple functions modify the signal in place, so the signal is both input and output. This may save some memory and processing time. Please write 'insig' and 'outsig' in the documentation, as the user should not care about this detail.

`fs` Sampling frequency.

`siglen` Length of signal

`fc` Center frequency/frequencies of filter/filter-bank.

`flow`,`fhigh` Generic low, high frequencies determining a range of frequencies.

`a`,`b` Filter coefficients to IIR filters.

6 Structure of the AMTOOLBOX Sourceforge repository

The software is kept in a Git repository. The repository has several *branches*. The idea is that you only work on specific things in specific branches.

One of the branches in called *stable*. No development ever takes place on this branch. Instead, the development branches are merged into this one when they are ready.

7 Anchors

An anchor is a capitalized word in the code, that can be extracted by a script.

Each file should contain the following anchors:

AUTHOR The line following this anchor indicates who the authors are.

TESTING The line following this anchor indicates in which routine this particular function is tested, or if testing is not nedeed / not applicable / not done yet.

For debugging it is possible to insert XXX, FIXME, BUG, TODO into the code, followed by a description of the problem.

- **References**