GammaTone Auditory Filter Kit

READ FIRST: All filter coefficient derivation and filter application is the work of Slaney (1993) and can be found in the freely available document "An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank" included in this .zip as "PattersonsEar.pdf". The source files uploaded are just modified versions of those found in the appendix of this document. All other references to related auditory material can be found in the acknowledgements of that document.

Version 2 updates:

- Added better centre frequency bounding for auditory filters. Now minimum and maximum values will be precisely as user specifies.
- Added a DSAM centre frequency emulation mode
- Removed GammaTone.m (all in one function) as it provides no real speed benefits and it contained almost entirely cloned code.

Introduction

This is a small collection of .m files which can be used to implement the Patterson-Holdwsworth auditory filterbank in order to observe corresponding basilar membrane motion (BMM) to an input signal across a number of channels, which is a commonly used pre-processing stage in auditory models. Until now I have had to invoke 3rd party software from MATLAB to accomplish this relatively simple task which is time consuming. The files included to make life easier are described as follows.

GammaToneMake.m:

The user inputs sampling frequency, number of channels required and the range of frequencies to be covered and this function generates filter poles and zeros for each auditory filter accordingly. It also gives useful information such as centre frequency and equivalent rectangular bandwidth (ERB) of each filter. This function has been modified slightly from Slaney's original with the inclusion of being able to switch the spacing method more easily between Moore/Greenwood etc. A highest frequency input variable was added to give more control (rather than just fs/2 to low frequency).

GammaToneApply.m:

Very simple function, nearly identical to Slaney's original which just takes an input array and filters it with the filter weights generated in GammaToneMake.m

BMMplot.m:

This is a simple plotting routine which I put together to mimic the BMM plot types seen frequently in journal articles, as well as software such as AIM and AMS. This allows data from each channel to be viewed as stacked line graphs.

DemoGT.m:

Contains sample script to show how everything works together. Put this in the working directory and run from the command line or editor.

HAVE FUN!!! – email: nick@ihr.mrc.ac.uk