# ASR Suite for Dummies

## Overview

This document is a software manual, describing the tools used to yoke the simulated hearing aid to a model of the patients impaired hearing (the dummy), and how the dummy can be subsequently connected to an automatic speech recogniser. The idea is that if an automatic speech recogniser is trained using feature vectors derived from a model of normal hearing, then the speech recogniser will perform optimally when tested with feature vectors derived from the same normal-hearing dummy. If the recogniser is tested using feature vectors derived from an impaired dummy, then one would expect the recognition score to drop significantly. The representation of the acoustic stimulus can be modified by a hearing aid algorithm before being presented to the impaired dummy. The potential of the hearing aid signal processing to alleviate some of the reduction in speech intelligibility caused by the impairment may then be assessed by comparing the aided and unaided recognition scores. This software framework should eventually enable the tailoring of hearing aid parameters to individuals in their absence.

The software suite exists in a “userPrograms” folder within the main Matlab model of the Auditory Periphery (MAP) folder. This tutorial document assumes that the reader is already familiar and comfortable with using MAP. The full documentation for MAP can be found at:

<http://www.essex.ac.uk/psychology/department/research/hearing_models.html>

## Files included in the suite

The evaluation of the hearing aid is done with the assistance of 3 main classes.

### cEssexAid.m

This is the file that contains the class definition of the Essex Aid wrapper. The Essex Aid is a novel hearing aid algorithm developed at the University of Essex. Once the user is familiar with the software framework, it should be relatively straightforward to swap out the hearing aid algorithm so that any hearing aid algorithm can be tested.

### cJob.m

This is the class definition that is the real workhorse of the operation. It provides many utility functions to assist the user in creating feature sets for use in training and testing the speech recogniser. This class also contains code for scheduling, so that many sounds can be processed in parallel on one or many machines should you have the resources available to do so.

### cHMM.m

This is a wrapper class for the Hidden Markov Tookkit (HTK) <http://htk.eng.cam.ac.uk/>. This class contains helper functions that provide the recogniser with the necessary information for training a HMM and scoring the results produced.

\*\*\*

Other functions within the parent directory include

### worker.m

This is an autonomous little function that takes a path to a folder containing a list of jobs as an input argument. A job is defined here as a list of wav files that need to each be converted into a feature vector. It searches the directory for jobs to do and works until all of the jobs are complete.

### Exp\_DEMO.m

Files prefixed with “Exp\_” are the files that users will edit most frequently. They are at the top of the function call stack and initiate all of the tasks required to run a recognition experiment.

\*\*\*

There are also two folders in the parent directory

### /def

This contains the definition files required by HTK, such as grammar rules, dictionaries and hmm prototypes. Full descriptions of these files are beyond the scope of this tutorial, but detailed information can be found here <http://htk.eng.cam.ac.uk/>

### /ASRfiles

This is a folder containing some additional utility functions used by the main classes.