*MAP1\_14*

Quick Start



|  |  |
| --- | --- |
|  | **Hearing Research Lab**  **University of Essex**  **2011**  **Volume 1, Issue 1** |

Contents

[1. What is MAP1\_14? 4](#_Toc311115050)

[1.1. What can I do with it? 4](#_Toc311115051)

[2. Installation 5](#_Toc311115052)

[2.1. Installation 5](#_Toc311115053)

[3. Demonstration programs 6](#_Toc311115054)

[3.1. demoTwisterProbability 6](#_Toc311115055)

[3.2. demoTwisterSpikes 8](#_Toc311115056)

[3.3. Two tone suppression 9](#_Toc311115057)

[4. User code 11](#_Toc311115058)

[5. MAP1\_14 architecture 13](#_Toc311115059)

[5.1. Model code 14](#_Toc311115060)

[5.2. Segmentation 15](#_Toc311115061)

[5.3. Parameters 15](#_Toc311115062)

[6. MAP1\_14 architecture 17](#_Toc311115063)

[7. Displaying the output 19](#_Toc311115064)

[7.1. showMAPoptions 19](#_Toc311115065)

[8. Physiology check-up 21](#_Toc311115066)

[9. Psychophysics using MAP 23](#_Toc311115067)

[9.1. Auditory profiles using *multiThreshold* 23](#_Toc311115068)

[9.2. Generating a model profile 25](#_Toc311115069)

[9.3. Exploring parameter changes when measuring profiles 27](#_Toc311115070)

**Distribution**

This software is freely shared on the understanding that the recipient accepts full responsibility for how it is used. The author makes no guarantees of any kind by sharing this software. If you know someone who wants a copy, please refer them to the author who will send them the latest bug-free version of the software. Otherwise, old versions circulate to the detriment of our reputation!

This is work in progress. Please send comments, criticisms and suggestions to the author [rmeddis@essex.ac.uk](mailto:rmeddis@essex.ac.uk)

**This guide**

This is a short guide intended to allow users to get started quickly. It invites the reader to try some demonstration programs and gives a general overview of how the MAP software works. A more detailed description of the model is given in the ‘MAP1\_14 technical description’ document.

# What is MAP1\_14?

MAP1\_14.m is the latest version of the Matlab Auditory Periphery (MAP) model from Essex University, UK. It is a computer program that simulates all stages of the auditory periphery up to the auditory nerve (AN) and into the brainstem.

MAP1\_14 now has efferent functionality including both acoustic reflex and MOC efferent function.

|  |
| --- |
|  |

Figure 1: **Schematic of the auditory pathway modelled by MAP1\_14. Each shape represents a separate stage of signal processing in the auditory periphery.**

The model is realised by systems of equations that are described in the accompanying document MAP Model Technical Description.

## What can I do with it?

MAP1\_14 can be used to:

1. Process acoustic waveforms (e.g. tones, speech) to generate multi-channel representations at different levels in the auditory periphery including tympanic membrane, stapes, basilar membrane (BM) displacement, inner hair cell cilia displacement, receptor potential and auditory nerve. Additional representations are possible at brainstem level in terms of neuronal responses including chopper and primary-like responses in the cochlear nucleus (CN). Second level neurons can also be simulated.
2. Illustrate basic physiology experiments in the auditory periphery.
3. Generate auditory features for automatic speech recognisers.
4. Demonstrate a range of psychophysical phenomena such as absolute threshold, threshold as a function of duration, forward masking, threshold masking curves (TMCs), psychophysical tuning curves (PTCs). These demonstrations use the same *multiThreshold software* as used to measure these phenomena in human subjects.

# Installation

The software comes in a zipped folder called MAP1\_14.

Unzip the folder and place it anywhere in the main memory.

# Software architecture

The folder structure has been designed to minimize MATLAB path problems and enhance portability.

The beating heart of the system is a program containing the model equations. It is called ‘MAP1\_14.m’ and can be found in the folder *MAP1\_14\MAP*. The equations are described in the more comprehensive [MAP 1\_14 Model Technical Description](MAP%201_14%20Technical%20Description.docx) document.

The parameters to be used with the MAP model equations are combined in a single parameter file. Different parameter files exist to model both normal and impaired hearing. These files are located in the folder *MAP1\_14\ parameterStore*

The folder *MAP1\_14\demonstrations* contains demonstration and test programs. These allow the new user to explore its potential. Some examples are given immediately below.

Another folder, *MAP1\_14\multithreshold 1.46,* contains psychometric testing software normally used with laboratory subjects. This can be harnessed to the MAP model so that the model can be tested in the same way as clinical patients. An introduction to this software, [***multiThreshold Quick Start***](multiThreshold%20Quick%20Start.docx) is located in the folder *MAP1\_14\Documentation*.

The folder *MAP1\_14\wavFileStore* contains acoustic stimuli stored as *.wav* files*.*

The folder *MAP1\_14\utilities* contains utility functions that are shared across programs and can also be accessed by the user.

Users can write their own programs to control the model. These programs should be stored in the folder *MAP1\_14\userPrograms. The user is strongly encouraged to rename this folder by adding initials (*e.g. *userProgramsRM*.)*.* When sharing software or seeking advice from the author, it is convenient to send this folder for inspection.

When creating new folders, the user is strongly advised to locate them *inside* the MAP1\_14 folder. This allows all paths to be identified as relative to the MAP1\_14 folder. This is very useful when sharing software.

# Demonstration programs

To get started quickly, you can use the demonstration programs in the *demonstrations* folder.

Two programs ‘*demoTwisterProbability’* and ‘*demoTwisterSpikes’* both take the acoustic input from a *.wav* file, *twister\_44kHz.wav,* and process it through the model. This *.wav* file contains the single word ‘twister’ spoken by the author.

## demoTwisterProbability

The first program, *demoTwisterProbability,* plots the model auditory nerve response to this acoustic stimulus. Navigate to the *demonstrations* folder then, in the command window type

demoTwisterProbability

The model processes 21 channels with best frequencies (BFs) between 250 Hz and 8 kHz.

The program plots the model response at four stages (stimulus, stapes, basilar membrane (BM) and auditory nerve (AN) firing rate. This is shown Figure 2 (left panel).

|  |  |
| --- | --- |
|  |  |

Figure 2: **Left panel: Model output for the demonstration file *demoTwisterProbability*. The input utterance is the word ‘twister’ spoken at an rms level of 60 dB SPL by a male speaker. The model has 21 channels with BFs between 250 and 8000 Hz (evenly spaced on a logarithmic scale). The top three panels represent the signal, the stapes displacement and the basilar membrane displacements respectively. The bottom panel shows the AN response firing probability computed for both high and low spontaneous rate fibers in the upper and lower part of the panel respectively. Right panel: Surface plot of the high-spontaneous rate fiber response only. This is based on the same data as the upper half of the bottom pane to the left.**

Two types of auditory nerve fiber are illustrated; high-spontaneous rate (HSR) and low-spontaneous rate (LSR) response. A second image (Figure 2, right panel) is generated showing only the HSR fiber response as a surface plot. This plot represents the probability of action potentials scaled up to represent firing *rates* in each BF channel.

## demoTwisterSpikes

The second program, *demoTwisterSpikes*, illustrates a full spiking model. In the command window type

demoTwisterSpikes

It computes the AN response as spikes and then uses these spikes as the input to arrays of neurons in the cochlear nucleus and these units feed other units higher in the brainstem. In both cases the neurons are configured as simple chopper cells with regular firing patterns. The brainstem circuitry is still fairly primitive and exclusively excitatory.

This model simulates the response of both HSR and LSR AN fibers. These two streams are used to feed two mutually exclusive sets of brainstem units.

Only 11 BF channels are used in this example so that the spiking patterns are easier to see. The spike plots are horizontal rows (rasters) of spikes. Each row is the simulated spiking pattern of a single AN fiber. There are 100 AN fibers of each fiber type in each channel in panel 4 (total 2 x 11 x 100 fibers). There are 10 CN neurons in each channel in each fiber-type stream in panel 5 (2 x 11 x 10). At the second level, there is one neuron in each BF channel in each fiber-type stream in panel 6 (2 x 11 x 1).

|  |
| --- |
|  |

Figure 3: **Model response to the utterance ‘twister’. The first three panels are the same as the previous figure (signal, stapes response, BM response). The auditory nerve response (panel 4) is given as a raster plot of spike activity in 11 channels for two types of fiber (2200 fibers altogether); HSR (top of the matrix) and LSR (bottom of the matrix). The bottom two panels show the spiking activity of simulated chopper units in two successive layers of brainstem activity. In both panels, the spiking activity is separated into two independent streams fed by HSR and LSR AN fibers. The convention is that the HSR-sourced activity is displayed at the top of the images.**

## Two tone suppression

The third demonstration program, *demoTwoTone,* explores the nonlinearity inherent in the model. Navigate to the *demonstrations* folder then, in the command window, type

demoTwoTone

A pure tone is played to the model. Half way through the presentation, a second tone is added to the first. In a linear system, we would expect the second tone to add to the first tone producing an increase in the output. The model shows that this is not always the case; on some occasions, the output is actually reduced. This phenomenon is popularly known as ‘2-tone suppression’.

|  |  |
| --- | --- |
|  |  |

Figure 4: Model output for the demonstration program, *demoTwoTone* (see text)*.*

In this example a 2 kHz probe tone is played continuously for 50 ms at a level of 15 dB SPL. A second tone is added after the first 25 ms. On successive trials, the level of this second tone is varied between 0 and 80 dB SPL and its frequency is varied between 250 and 8000 Hz. The figure shows the signal in the upper panel and the *auditory nerve firing probability* in the second panel.

Fig 4A shows the *final* trial where the suppressor frequency is 8000 Hz at 80 dB SPL. The AN response is reduced when the new tone is added to the probe tone. The third panel shows the response of all 21 channels of the model to this stimulus. The light band in the middle of the display shows that the 2000 Hz tone is generating a maximum response in the 2000 Hz BF channel. The yellow/orange blob at the top right of this panel is the response to the high frequency suppressor. When the second tone is switched on, the strength of the response to the probe is clearly reduced.

The right panel shows an intermediate result with a suppressor tone of 297 Hz at a level of 60 dB. When this becomes active, the response to the 2000-Hz probe tone in the 2000Hz channel is largely extinguished.

When the demonstration is run, the effect of all 189 trials is illustrated in rapid succession. The effect needs to be watched many times to grasp what is happening. Uncomment the pause instruction near line 215 to slow the movie down. At the end of the demonstration, the bottom panel of the figure will be complete. It shows the response to the combined tones expressed relative to the probe alone. A value smaller than one indicates that suppression has taken place. The contour plot shows that tones close in frequency to the probe tone are excitatory (ratios greater than one). Outside this region the effect is largely suppressive.

The small white central dot indicates the level and the frequency of the fixed (probe) tone.

The appearance of the summary panel changes substantially when different probe frequencies and levels are used. When evaluating the two-tone suppressive properties of the model, it is important to choose published animal data and then to replicate the stimulus frequencies and level exactly. Only then is the comparison valid. Most of the parameters of this demonstration can be modified at the head of the program code.

# User code

The two demonstration programs are useful for giving some insight into what the model can do but they are not very useful for regular use with the model. There is, however, a program that can be used for this purpose. *runMAP1\_14* is a general purpose program that is designed so that the user can make changes with a view to exploring more thoroughly the potential of the software. The program is written with comments and accessible code that offers a range of options. Hopefully, these should be self-explanatory as you read through the program. You can find this program in the *MAP1\_14\demonstrations* folder.

Make a copy of this program, put it in your user folder and give it a new name (e.g. myRunMAP). If the program is run in the form supplied, you should get an output that looks like this after rearranging the figures. Bear in mind that small changes in any one of many parameters can substantially change the appearance of this figure, your images may look qualitatively different from this.

|  |
| --- |
|  |

Figure 5: runMAP1\_14; a ‘hands-on’ program for users to explore the MAP model. Left: model output in probability mode in response to the utterance ‘twister’. Middle: summary of efferent activity. Right: 3D image of HSR AN response.

Basically, this code generates an acoustic waveform scaled in Pascals and presents it to the model. The program user has a range of options that can be selected by modifying the program code; the stimulus can be pure tones or the input from a .*wav* file, the level of the signal (rms) can be set, the model can be run in probability mode or spikes mode and the number and frequency range of the channels can be changed. Don’t forget to make a copy of the original in your userPrograms folder before modification!

Read through the program to familiarise yourself with how it works because this is a good starting point for building your own program by modifying this one. It is a useful exercise to run this program after changing some of these values to see the consequences.

***Efferent effects.*** *These are new to MAP1\_14* and remain experimental*. runMAP1\_14* generates a third figure showing how the efferent functions have been activated. In this example (using a 70 dB speech signal) the AR reflex is not activated (because the intensity of the tone is too low). The MOC reflex has a much lower threshold and is moderately activated.

The AR can be demonstrated more effectively by selecting a pure tone signal and presenting it at 90 dB SPL. The AR attenuates the stapes response once it is activated.

|  |
| --- |
|  |

Figure 6: runMAP1\_14; stimulus is a 1–kHz pure tone played at high signal level so as to activate the AR. Left: model output in probability mode. Middle: summary of efferent activity. Right: 3D image of HSR AN response. Note the very strong onset effect in the AN response.

# MAP1\_14 architecture

This section is intended to give greater insight into the functioning of the model itself for those who wish to understand how it works or who may wish to improve it with further insights into how hearing works. More details can be found in *MAP1\_14Model description* also in the *documentation* folder.

The two demonstration programs, *demoTwisterProbability* and *demoTwisterSpike,s* use different model architectures. In both cases MAP is structured as a cascade of stages with two feedback loops.

|  |  |
| --- | --- |
| **Probability model** | **Spiking model** |

Figure 7: **Flow diagrams of the two models implemented in the demonstration programs**

***Probability model.***After the inner hair cell stage, the model bifurcates to give two streams; one for HSR fibers and one for LSR fibers. The probability model ends at the AN fiber stage. The acoustic reflex (AR) and medial olivo cochlear (MOC) efferent feedback loops are fed back from the HSR and LSR paths. This is not physiologically accurate but it provides a useful approximation for users who want rapid computation.

***Spiking* *model*.** This takes the spiking output of the AN fibers and feeds them through two brainstem stages. The first brainstem stage simulates CN neurons and the second stage simulates activity in the medial superior olive (MSO). It is this second stage that is used as the source of the feedback control signals (efferent activity). This is a more accurate representation from a physiological point of view than the probability model but is slower to compute.

***Efferent effects*.** The AR efferent activity derives its input from LSR fibers which have high thresholds. This helps to guarantee that the AR is active only at high signal levels. The MOC efferent activity derives its input from the HSR fibers which have low thresholds. This detail is purely speculative in the absence of definitive anatomical evidence.

## Model code

*MAP1\_14.m* is the core program containing code that simulates the response of the auditory periphery. It is a single program file containing all instructions for executing the model. It is located in the MAP1\_14\MAP subfolder.

The function call is

MAP1\_14 (inputSignal, sampleRate, BFlist, MAPparamsName, ...

AN\_spikesOrProbability, paramChanges)

***inputSignal***  is a single row vector in Pascals. (It is a row vector because the model code always regards the x-axis of a matrix as the horizontal time axis.) It is single row because the model only processes mono signals

***sampleRate***  is the inputSignal sample Rate (Hz)

***BFlist***  is a vector of BFs (in Hz). If this is specified as ‘-1’, the list of BFs is based on the default values specified in the model parameter file (see immediately below).

***MAPparamsName*** e.g. ‘Normal’, is a string used to make up the name of a model parameter file (e.g. *‘*MAPparamsNormal*’).* The parameter file contains all of the parameters of the model. Parameter files are always located in the folder called *parameterStore*

***AN\_spikesOrProbability*** =’spikes’ for full model generating individual spikes in AN as well as simulation of brainstem neuron activity.

=’probability’ for AN model based on spike probabilities. This option computes faster than the alternative ‘spikes’ option

***paramChanges***is a cell array of strings containing commands to make last-minute changes to the model parameters after loading them from the MAPparams<\*> file (see below). This can be omitted.

There are roughly 1300 lines of code representing all of the peripheral auditory processing stages. These lines are divided into two groups. The first 550 lines or so are purely concerned with pre-allocating memory and establishing constants to be used in the computations that come later. These lines serve to speed up the execution of the main body of the code. The second half of the code contains the equations that simulate activity at each stage. These equations are described in the companion document MAP1\_14Model Description.

## Segmentation

The model is evaluated in short segments to allow for efferent feedback (AR and MOC) to be implemented. The segments are typically 10 ms long. This is slightly shorter than the feedback delay of the efferent system.

## Parameters

The model parameters are established by a .m file to be found in the ‘parameterStore’ folder. For example, *MAPparamsNormal.m* is a file containing a recommended set of parameters that aims to simulate normal hearing. MAP1\_14 executes this parameter file before running the model. Examine the *MAPparamsNormal* file in the parameter store to become acquainted with the parameters required by the model. It is a very substantial list.

The parameters in the *MAPparamsNormal* file are simply the values that the author has found to be the most satisfactory in recent months. They are not perfect. If you wish to use them for research purposes, you should ask the author ([rmeddis@essex.ac.uk](mailto:rmeddis@essex.ac.uk)) for the latest parameters and any documentary support. This is ‘cutting-edge’ and they are not yet published. The user is responsible for his own parameters; these are the author’s but you are welcome to them.

The user is free to create new parameter files. The file name must begin with ‘MAPparams’ and be followed by an appropriate identifier, e.g. ‘MAPparamsNormal’. All parameter files should be located only in the *MAP1\_14\parameterStore* folder. One of the aims of this project is to create ‘hearing dummies’ to simulate the impaired hearing of individual patients. Each dummy will have an associated parameter file.

**Changing a small number of parameters**

Small changes to the ‘Normal’ parameter set can be explored using the *paramChanges* argument in the call to MAP1\_14. See *runMAP1\_14* for an example. This facility changes parameters while not disturbing the original parameter file. *paramChanges* is a cell array of strings. Each string is a complete MATLAB statement, exactly as it would occur in the parameter file itself (including the final semi-colon). For example,

paramChanges = {'DRNLParams.a=0;', ‘DRNLParams.g=200’;};

This example makes two changes. It sets the parameter *DRNLParams.a* to zero. (This disables the nonlinear path in the DRNL model of BM response). It also sets a new value for the gain of the linear path (*DRNLParams.g*). These parameter changes are applied after reading the MAPparams file and *overrule* whatever is to be found in the *MAPparams\*.m* function. To set up a string,

1. Copy the relevant command from a *MAPparams\*.m* parameter file. Change it as required. Make sure to include the semi-colon at the end of the command (to avoid printing in the command window every time the change is evaluated).
2. Put single quote marks around the command to indicate that each command is a separate string
3. Use commas to separate multiple strings
4. Wrap the whole lot in curly brackets to indicate that this is a cell array of strings.

Take care when doing this because MAP1\_14 cannot always detect when an inappropriate command is entered. *runMAP1\_14* routinely prints out the parameters at the end of a run. Always check the parameter printout to make sure that your desired change has been implemented.

# MAP1\_14, accessing the output

The *MAP1\_14* function draws no pictures nor does it return any output directly to the calling program. All model results are stored in *global* variables. These are left behind in global memory when MAP1\_14 exits. To access all of these results use the following global statement and all of the data should immediately appear in your workspace.

*global savedParamChanges savedBFlist saveAN\_spikesOrProbability ...*

*saveMAPparamsName savedInputSignal dt dtSpikes ...*

*OMEextEarPressure TMoutput OMEoutput DRNLoutput...*

*IHC\_cilia\_output IHCrestingCiliaCond IHCrestingV...*

*IHCoutput ANprobRateOutput ANoutput savePavailable...*

*saveNavailable ANtauCas CNtauGk CNoutput ICoutput...*

*ICmembraneOutput ICfiberTypeRates MOCattenuation ARattenuation*

As a familiarisation exercise, run either the *demoTwisterSpikes* or *demoTwisterProbability* scripts to generate sample results. Then execute the global statement above in the command window.

Hopefully, the variables have been named so that you can work out what they represent. However, some comments may help the user get started.

All values are international units (m, s, Pa, V, A)

*Recovering some of the input choices:*

***saveMAPparamsName*** (string) holds the string (e.g. ‘Normal’) used to generate the name of the parameter file used (e.g. ‘MAPparamsNormal’).

***savedBFlist***is a list of BFs, one per channel in the model. .

***ANtauCas*** *is* used to define the spontaneous rate of AN fibers. It is a list of calcium clearance time constants controlling activity at the IHC/AB synapse. There is one time constant per fiber type. If the length of this vector is two then both LSR and HSR fibers were computed. If the length is one, then only the HSR fibers were computed.

***saveAN\_spikesOrProbability*** *(string)* indicates whether ‘spikes’ or ‘probability’ computations were requested.

***ANprobRateOutput.*** If MAP1\_14 is run in probability mode, the AN spike rate matrix (BF x time) can be found in. If two fiber types are used (e.g. HSR and LSR), this matrix will be twice as large. The top half of the matrix will be the output from the LSR fibers. NB, when plotting this matrix it is conventional to flip it to put the HSR at the top of the picture and to order the BFs vertically from low to high. The sampling interval for *ANprobRateOutput*  is saved in the variable *dt*.

*Recovering some of the output:*

***ANoutput*** If MAP1\_14 is run in spikes mode, the AN spiking output will be found in the matrix *ANoutput*. This is a logical matrix (0/ 1). The matrix is, therefore, a raster plot. Each BF is associated with many AN fibers (typically 100). All fibers with the same BF and fiber type are in adjacent rows. For example if we use both HSR and LSR fibers, have 21 channels and there are 100 fibers per combination, we should expect to see 2 x 21 x 100 rows in this matrix (i.e.4100 AN fibers). Remember that there is only a total of about 30000 AN fibers in the human auditory periphery.

The sampling interval for the spiking model is *dtSpikes.* The sampling rate (*1/ dtspikes*) is reduced because spikes are relatively sparse. Each row of this matrix is the spiking response of an individual AN fiber.

The output of the early stages of the model is stored in the following variables: (sample rate is *1/ dt*)

1. *OMEextEarPressure* acoustic pressure
2. *TMoutput*  tympanic membrane displacement (m)
3. *OMEoutput* stapes displacement (m)
4. *DRNLoutput*  basilar membrane displacement (m) (BF x time)
5. *IHC\_cilia\_output* IHC cilia displacement (m)
6. *IHCoutput* IHC receptor potential (V)

The amount of efferent activity can be monitored using the variables *ARattenuation* (single row vector) and *MOCattenuation* (matrix BFs x time).

# Displaying the output

*UTIL\_showMAP.m* is a general purpose function for reporting and plotting the output from the most recent MAP1\_14 call.

UTIL\_showMAP (showMapOptions)

UTIL\_showMAP is used by the demo files described above. Consult these scripts for examples of how to use it. It assumes that MAP1\_14 has recently been run, leaving all of its output in global store.

*showMAPoptions* is a structure containing a number of flags (0/ 1) indicating the user’s requirements (see below).

## showMAPoptions

*UTIL\_showMAP* is able to generate a wide range of displays. These must be selected in the showMapOptions structure. The fields in this structure are all logical (true or false).

showMapOptions.printModelParameters=1; % print model parameters

showMapOptions.printFiringRates=1; % mean activity at all stages

showMapOptions.showModelOutput=1; % plot all stages output

showMapOptions.showEfferent=1; % plot of efferent activity

showMapOptions.showACF=1; % SACF (probabilities only)

showMapOptions.surfAN=0; % ANoutput (HSR) surf plot

options modifying surfAN (above):

showMapOptions.fileName=''; % parameter filename for plot title

showMapOptions.PSTHbinwidth=0.001 % binwidth for surface plots

showMapOptions.colorbar=1; % request color bar if appropriate

showMapOptions.view=[0 90]; % angle of the surface plot

***printModelParameters*** prints to the command window all the model parameters used.

***showModelOutput*** plots the output of each stage of processing from signal through to the AN and on to the brainstem response if appropriate

***printFiringRates*** prints in the command window the overall level of activity in the afferent and efferent stages

***showACF*** computes and displays the summary autocorrelation function

***showEfferent*** plots of the AR and MOC activity levels

***surfAN*** generates a 3D plot (surf) of the activity of the HSR AN fibers. *fileName, PSTHbinwidth, colorbar and view* are all options supporting surfAN

# Physiology check-up

MAP is a model based on the known anatomy and physiology of the peripheral auditory system. Its parameters have been chosen to simulate physiological recordings made using small mammals. A number of test programs simulate these physiological experiments as a check that the model is doing what it should be doing. These test programs can either be run individually or together as a series. Separate programs are available for spikes and probability implementations of MAP.

The test programs, *testPhysiology* and *testPhysiologyProb,* can be used to ‘exercise’ MAP to check that it is functioning correctly. To run it you should navigate to the *demonstrations* folder and type on the command line:

testPhysiology(BF, 'Normal') % slow, works in ‘spikes’ mode

**or**

testPhysiologyProb(BF, 'Normal') % faster works in probability mode

This will evaluate MAP1\_14 in a single-channel model with a best frequency of BF using the model parameters to be found in *MAPparamsNormal*.

A range of measurements are made using a sequence of individual test programs,

1. Stapes response at a range of frequencies (Figure 2), *testOME.m*
2. BM I/O function, the MOC elicited by the stimuli and the iso-reponse contour (‘tuning curve) at a number of different response levels (Figure 3), *testBM.m*
3. Various IHC I/O functions (Figure 4) and compares them with published animal data, testRP.m
4. (Figure 15), testAN.m
   1. (*testPhysiologyProb* ) AN firing rate I/O function based on actual spikes (top right)
   2. (*testPhysiology)* AN firing rate I/O function based on actual spikes (top right) also CN and IC I/O functions – one function for each of the fiber types selected.
5. Synapse response (available transmitter) as a function of time and level (Figure 6). Figure 88 is a contour plot of the same data featuring in Figure 6), *testSynapse.m*
6. (*testPhysiology only)* Forward masking simulation of an experiment by Harris and Dallos (Figure 7), *testFM.m*
7. (*testPhysiology only)* AN phase locking assessment based on an experiment by Johnson, *testPL.m*

The graphical output from the function should look like the following.

More information about the individual tests can be found in the ‘*MAP Model Technical Description’* file in the *‘Documentation’* folder or by using the help function in the command window, e.g. *help testOME*

|  |
| --- |
| testPhysiology(1000, 'Normal', paramChanges) |

Figure 8: Simulations of physiological experiments using the spikes implementation of the model. These data were generated using:

testPhysiologyProb(1000, 'Normal')

|  |
| --- |
|  |

Figure 9: Simulations of physiological experiments using the probability implementation of the model. These data were generated using:

# Psychophysics using MAP

## Auditory profiles using *multiThreshold*

The purpose of auditory model is ultimately to model human hearing to such a degree that the model and the individual cannot be distinguished (a kind of Turing test).

A step in this direction is to administer the same tests to the model that might be applied to the human listener. ***multiThreshold*** is a software controller of a range of psychophysical tests that have been designed to be suitable for testing both humans and the MAP model. Three tests are of particular interest; 1) absolute threshold, 2) psychophysical tuning curves (PTCs) and 3) threshold masking curves (TMCs). These are tests of sensitivity, frequency selectivity and compression respectively.

Here is an example of a hearing profile produced using multiThreshold harnessed to the MAP model and using the *MAPparamsNormal* parameter set.

|  |
| --- |
|  |

Figure 10: Auditory profile generated using multiThreshold testing software harnessed to MAP using *MAPparamsNormal* as the parameter file. Top row: temporal masking curves (TMC), an indicator of compression. Middle row: iso-forward masking tuning curves (IFMCs) or psychophysical tuning curves (PTCs), an indicator of frequency-selectivity. Bottom: absolute thresholds for 500-ms tones (thick line), for 16-ms tones, (thin line) and ISO standard (blue dotted line). Grey and dotted lines are data from a human listener with normal hearing.

This profile is similar to a profile for a human listener with normal hearing; original data shown as grey dotted lines).

Here are two profiles designed to simulate *impaired* hearing. The continuous, coloured lines are the impaired models (hearing dummies) and faint dashed lines are comparison data collected from a 20-year old male with good hearing.

|  |  |
| --- | --- |
|  |  |

Figure 11: Two auditory profiles generated using multiThreshold testing software harnessed to MAP, a computer model of the auditory periphery. Two different sets of parameters were used. A: the endocochlear potential was reduced to 70 mV but otherwise the parameters were those in the ‘Normal’ file. B: the gain of the nonlinear path in the DRNL filter was reduced to 0, but otherwise the parameters were those in the ‘Normal’ file. The faint lines are normal human data.

In the left panel, the impairment is induced by reducing the endocochlear potential (supplying the inner hair cell, IHC) from 100 mV to 70 mV (*IHC\_cilia\_RPParams.Et= 0.070;*)

In the right hand panel, the impairment is created by disabling the contribution of all outer hair cells (*DRNLParams.a= 0;*)

The ***multiThreshold*** software can be found in the folder ‘MAP1\_14\multiThreshold 1.46’ and is explained in the associated document: MultiThreshold Manual to be found in the ‘Documentation’ folder.

## Generating a model profile

A complete profile will be generated using the following procedure

1. Navigate to the *multithreshold 1.64* folder in the *MAP 1\_14* folder
2. Run *multiThreshold.m* to display the GUI.

|  |
| --- |
|  |

Figure 12: *multiThreshold* GUI for measuring auditory profiles for both humans and models.

1. From the ear menu (middle panel 2nd from top), select ‘MAPmodelSingleChannel’
2. From the paradigm menu (below the ‘ear’ menu) , select ‘profile’
3. Hit the run button and sit back. A profile should be generated after considerable amount of computation looking like this:

|  |
| --- |
|  |

Figure 13: Hearing profile generated by multiThreshold using the computer model as the subject. The ‘Normal’ parameter set was used for this example.

Notes:

1. Mor information on multiThreshold software can be found in *MultiThreshold Manual* in the *documentation* folder.
2. The model is tested in the same way as the listener except that no cue is used as it is of no help to the model. The stimulus is presented to the model which has to decide ‘yes’ or ‘no’ according to whether or not the probe was ‘heard’ by the model. The measurement software reacts as if the model’s decision was that of a human listener and generates the next stimulus in the adaptive sequence.
3. The thin blue lines in the bottom panel with circles are absolute thresholds for 250 ms and 16-ms tones.
4. Frequency selectivity: the V-shaped curves are iso-forward masking contours (IFMCs). A short (16-ms) probe tone is presented at 10 dB SL and is preceded by a 108-ms pure-tone masker. The listener is asked to say whether or not he hears the probe tone. The level of the masker is adjusted until the probe is heard on 50% of presentations. Measurements are made for different masker frequencies (over an octave range covering the probe tone). The IFMC is a record of these masker levels. IFMCs are measured for probes at 250, 500, 1000, 2000, 4000, 8000 Hz and for masker frequencies at 0.5, 0.7, 0.9, 1, 1.1, 1.3 and 1.6 times probe frequency.
5. Compression: the top panel records 108-ms pure-tone masker levels required to mask a brief, 16-ms probe tone presented at 10 dB SL on 50 % of presentations. The probe and the masker always have the same frequency but the gap between the masker and the probe is systematically varied between 10 and 90 ms. This is called a temporal masking curve (TMC). The masker levels in dB SL are given as the lower of the two lines. The upper line is the same data presented as dB SPL. The steepness of the line is affected by more than one physiological factor. However, if we assume that the recovery from forward masking occurs at a fixed rate, we can infer that the steepness of the line reflects, at least in part, compression of the response to the masker; most likely either at the basilar membrane or the inner hair cell.
6. Diagnosis: in the figure above, the model absolute thresholds are high compared to the human data. The IFMCs are a reasonable fit and the TMCs are a good fit at low frequencies but not at high frequencies. The aim, of course, is to find a set of parameters that offers a good fit to all of the data.

## Exploring parameter changes when measuring profiles

1. The user can seek a better fit by changing the model parameters. The parameters can be found in the parameter file in the folder called ‘parameterStore’. On this occasion the file *MAPparamsNormal* contains the parameters. This file is specified on the GUI as ‘Normal’ in the edit box called name at the top of the GUI. One way to change the parameters is by altering the parameters in this file and repeating the run.
2. If only a small number of parameters need be changed, this can be done on the GUI in the yellow box at the bottom left. This normally contains the empty cell array command *paramChanges= {' '}*; This can be edited by adding a string inside the quotes, e.g. *paramChanges= {' DRNLParams.a=0 '};* This new value is automatically implemented after reading the MAPparams file but before the model is run.
3. At the end of each run, a complete list of parameters and their values is printed in the command window. These can be copied and pasted into the GUI (but remember to add the semi-colon before the final quote.
4. If more than one change is required this can be achieved by putting both changes inside their own quotes but separated by a comma (both inside the curly brackets, e.g. paramChanges= {' *DRNLParams.a=0* ', ‘DRNLParams.g=500’}; . Line breaks at not currently allowed in this process.
5. When *paramChanges= {' DRNLParams.a=0 '};* is requested we get a profile that looks like this. This parameter change effectively disables the OHC contribution to the BM response.

|  |
| --- |
|  |

Figure 14: Hearing profile generated by multiThreshold using the computer model as the subject. The ‘Normal’ parameter set has been changed to simulate outer hair cell (OHC) dysfunction.