Forward masking model

The models described below can be used to replicate (hopefully) the results presented in
the JASA article (2005):

A computer model of the auditory nerve response to forward masking stimuli.
By Ray Meddis and Lowel O’mard.

The model was created using AMS (Auditory Modelling System). If you wish to run the
model you will need to download the appropriate modelling software. The software is
evolving all the time but the version to be found here is the version used to evaluate the
model as published in the article (version 1.1.7). If there are problems when running the
latest version of the software, you might try to go back in time to this version.

If you have problems with these programs or require further clarification, please contact
Ray Meddis ( rmeddis@essex.ac.uk ). If you are having difficulty with program
execution, please zip up the folder you are using and attach it to your email. This can
save 48 hrs of delay! Do bear in mind that some of the programs are stochastic and will
give a different answer each time. hopefully the broad pattern of results will be
maintained but the detail will usually be different.

The computer neural model  is given by the .sim files forwardMaskingSPIKE.sim and
forwardMaskingPROB.sim with associated .par files. You can run these by launching
AMS and loading/running the chosen file. The SPIKE version computes individual spikes
while the PROB version computes only the probability that a spike will be generated.
Single-file versions of the model are present as .spf files.

While the models run in stand-alone mode, the simulations of experiments are controlled
by MATLAB programs that invoke the model on each experimental trial with appropriate
parameters. MATLAB version 6 or higher is required for this.

There are four main demonstrations controlled by MATLAB programs whose procedures
are fully explained in the article. Make the folder the current directory before starting.
There is no need to change the MATLAB path.

rate-level

This evaluates the rate-response of the model using tone pips of increasing level. This
simulates the results shown in Fig. 2 of the article. The program computes the rate-level
function for all three types of fiber and is activated using the command

   rate_level

Two figures are produced:
1. The PSTH of the current stimulus. Note that the program computes a smooth probability function (not spikes which would be slower and add little).

2. The rate-level functions.

![Figure A: PSTH, B: rate-level function for HSR, MSR and LSR fibers.](image)

**Forward masking**

This evaluates the response of a single AN fiber following stimulation by a masker. This simulates the results shown in Fig. 4 of the paper. The response is computed as a spike probability. The programme is invoked by

```python
forward_masking (<fiberType(string)>)
```

where fibertype is ‘HSR’, MSR’ or ‘LSR’. For example:

```python
forward_masking (‘HSR’)
```

Two figures are produced:

![Figure A: PSTH of the single-fiber spike-probability response to the masker-gap-probe stimulus. B: forward masking recovery functions](image)
allThresholdsRelkin
This simulates the experiment by Relkin and Turner featured in Fig. 8 of the paper.

The program is invoked as follows:

```
allThresholdsRelkin(<fiberType (string)>, <list of masker levels>)
```

For example:

```
allThresholdsRelkin('LSR', [0:10:90])
```

The result should be four figures:

A

```
PSTH control upper trace
```

B

```
masker level dB SPL  HSR
```

C

```
thresholds per trial - LSR
```

D

```
mean threshold
```

Figure
A: The PSTH for the current test. The upper trace is the AN fiber response to the masker alone. The lower trace is the response to the masker followed immediately by the probe.
B: The Leavitt up/down path followed while computing the threshold.
C: The threshold obtained for each individual trial.
D: The mean threshold averaged across blocks.
allThresholdsCoincidence

This replicates the results shown in Fig. 9 in the paper. Masked probe thresholds are computed for a range of masker levels and masker-probe gaps. When testing this module, you can obtain the most satisfactory results using a mixture of LSR and HSR fibers. When HSR fibers are used in isolation, a threshold may not be obtained for 80 dB maskers and 5 msec gaps. This is because the fibers are unresponsive and may not produce consistent supra-threshold results even at the highest probe levels.

To invoke the model:

```
[thresholdSummary p allResults] = allThresholdsCoincidence(<fiber List>, <masker level list (s)>, <gap duration list (s)>)
```

where `<fiber List>` is a 3-valued vector specifying the number of HSR, MSR and LSR fibers.

`<masker levels dB>`: use -100 to switch masker off.

Example, (10 HSR and 10 LSR fibers):

```
[thresholdSummary p allResults]=allThresholdsCoincidence([10 0 10], [20 40 60 80 -100], [.005 0.01 0.02 0.04])
```

Most of the parameters of the model are set in the program during initialisation and are stored in a cell array called ‘p’ (for parameters). p is returned with the results at the end of the program. If you add p as a fourth argument in the function call, your cell array p will be used rather than the program defaults.

Example result (this run took about 5 hours on my laptop).

```
thresholdSummary =

30.2679 28.375 26.9821 25.3571
52.8571 43.2143 27.5714 26.1071
65.6607 54.7857 30.6429 24.375
82.5179 62.8571 57.625 30.875
21.9821 18.7857 19.2857 20.0893
```

```
p=

   numFiberList: [10 0 10]
   maskerLevelList: [20 40 60 80 -100]
   gapDurationList: [0.0050 0.0100 0.0200 0.0400]
   coincidenceCriterion: 3
   BF: 4000
   simFile: 'forwardMaskingSPIKE.spf'
   model_AN_latency: 0.0010
   maskerDuration: 0.3000
   probeDuration: 0.0200
   maskerInitialSilence: 0.0500
   rampDuration: 0.0100
   totalDuration: 0.5000
```
\begin{verbatim}

5.8 hours to complete

allResults =
{masker level (repl)  ?  gap(1)  gap(2)  gap(3)  gap(4)}
20.0000  1.0000  0.0400  31.2143  28.1429  23.7143  27.1429
20.0000  2.0000  0.0400  28.2143  31.9286  31.0714  26.2143
20.0000  3.0000  0.0400  30.2143  27.5000  29.4286  22.6429
20.0000  4.0000  0.0400  31.4286  25.9286  23.7143  25.4286
40.0000  1.0000  0.0400  44.7143  46.8571  19.5714  24.5000
40.0000  2.0000  0.0400  54.2143  55.7143  19.5714  24.5000
40.0000  3.0000  0.0400  59.6429  55.7143  27.5143  27.7857
40.0000  4.0000  0.0400  52.8571  40.9286  38.7143  24.0000
60.0000  1.0000  0.0400  71.0714  52.7857  31.9286  25.5714
60.0000  2.0000  0.0400  57.7143  54.5000  20.0714  24.2143
60.0000  3.0000  0.0400  66.3571  53.7143  28.4286  24.6429
60.0000  4.0000  0.0400  67.5000  58.1429  42.1429  23.0714
80.0000  1.0000  0.0400  84.9286  58.5000  52.8571  32.7143
80.0000  2.0000  0.0400  76.0000  60.3571  63.3571  33.8571
80.0000  3.0000  0.0400  88.5000  61.3571  62.5714  28.4286
80.0000  4.0000  0.0400  80.6429  71.2143  51.7143  28.5000
-100.0000  1.0000  0.0400  23.1429  19.5714  20.9286  22.5000
-100.0000  2.0000  0.0400  22.4286  17.6429  18.5000  19.0714
-100.0000  3.0000  0.0400  22.0714  19.7857  19.7857  20.3571
-100.0000  4.0000  0.0400  20.2857  18.1429  17.9286  18.4286

Five figures should result:
\end{verbatim}
Figure
1. The PSTHs for the probe (lower trace) and the no-probe conditions.
2. The progress of the Leavitt threshold function.
3. The thresholds so far for this masker level.
4. The current average across blocks for the current masker level.
5. The mean thresholds (so far) across all masker levels and gaps.

The Excel spreadsheet in the folder contains the following sets of results.
Absolute thresholds are 20 dB SPL. Note changes of scale. B is close to Jesteadt’s results.

Replenish rate
The replenish rate (published as 3) can be changed by resenting the p cell array

\[
p\text{.REPLENISH\_RATE}=3;
\]

This gives a more realistic results in the sense that the recovery functions do not meet at the 40 msec gap. This is at the expense of a severely reduced saturated firing rate for the HSR fiber. More research is needed!