MATLAB Project (Assessments 1 and 3) – Auditory System

This project aims to familiarise with MATLAB by working on a real-world model, specifically an auditory system that simulates the components associated with hearing. Note that this project is an **individual task**.

An auditory system mimics the behaviour of a biological cochlea found in humans and other mammals (Lyon, Katsiamis and Drakakis, 2010). The system converts a 1D discrete-time audio signal to a 2D time-frequency signal called an auditory spectrogram. From this spectrogram, audio information can be extracted, as shown in Table 1. Its applications include hearing aids (Edwards, 2007), speech processing (Tchorz and Kollmeier, 1999) and musical information retrieval (Goto, 2006), multimedia audio systems (Xu *et al.*, 2018), and brain modelling (Heil and Peterson, 2015).

Index	Audio Information	Responsible For
1	Intensity	Sound loudness (Plack and Carlyon, 1995).
2	Direction	Location of a sound (Xu *et al.*, 2019).
3	Pitch	Difference between musical notes and also male and female voices (Meddis and O'Mard, 1997).
4	Timbre	Sound colour and shape indicate the identity of a sound source, e.g. specific person speaking, specific music instrument playing, etc (Shamma, 2003).

Table 1: Information extractable from an auditory spectrogram.

To convert a one-dimensional (1D) sound signal into a two-dimensional (2D) time-frequency representation, a cochlear filterbank is used. A cochlear filterbank comprises multiple gammatone filters either in parallel (Meddis, O'Mard and Lopez-Poveda, 2001) or series (Lyon, 2017). The bandwidth of each gammatone filter increases with increasing frequency so that a high centre frequency filter has a higher bandwidth than a filter with low centre frequency, as shown in Figure 1(a).

A gammatone filter generally behaves like a bandpass filter but has differences associated with the behaviour of the cochlea mechanics (Katsiamis, Drakakis and Lyon, 2007). Each gammatone filter is tuned to a specific centre frequency. It only responds to a specific frequency that corresponds to the mechanics of one specific location on the cochlea. So, when the input signal resonates close to the centre frequency of the filter, the filter will output a resonating signal at its centre frequency. Hence, to model an entire cochlea, a gammatone filterbank is used (Lopez-Poveda and Meddis, 2001). A filterbank will have a number of gammatone filters whose centre frequencies are tuned from low to high for the entire spectrum of a sound signal.

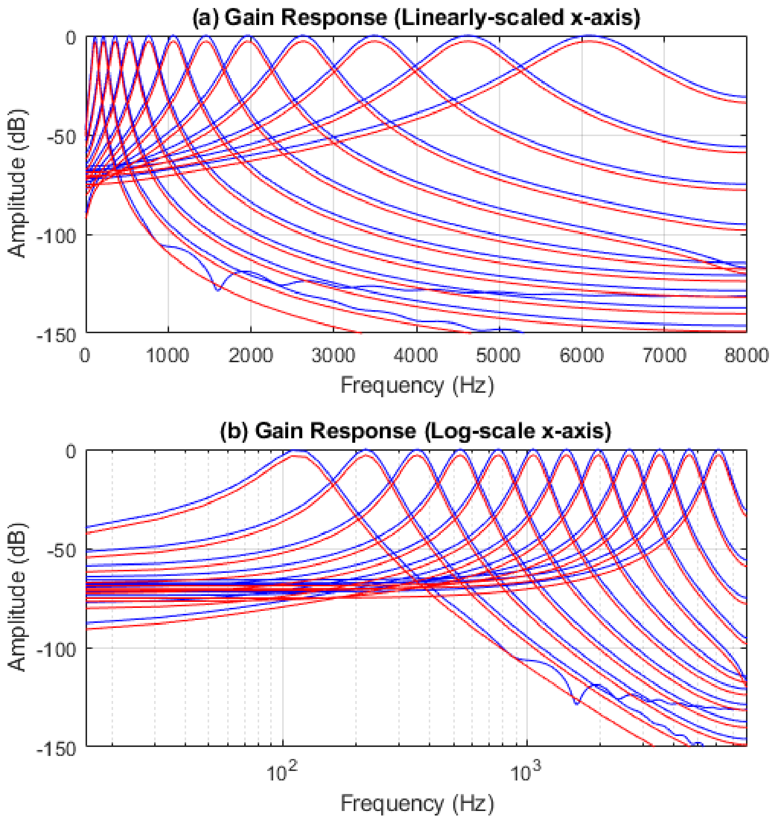

Figure 1: Increasing bandwidth with increasing centre frequency in the gain response of gammatone filters. (a) the x-axis is linearly scaled where intervals between frequencies are the same; (b) the x-axis is logarithmically scaled where intervals between frequencies are nonlinear.

Ideally, the varying filters tuned differently will react to the different frequencies in the input signal and will output multiple signals. These signals are then half-wave rectified, where all negative values are set to 0, and only positive values remain. They can be visualised as a 2D image known as an auditory spectrogram (R. Mergu and K. Dixit, 2011), as shown in Figure 2.

An alternative method of generating a spectrogram is by calculating the short-time Fourier transform (STFT) of a sound signal (Alm and Walker, 2002).

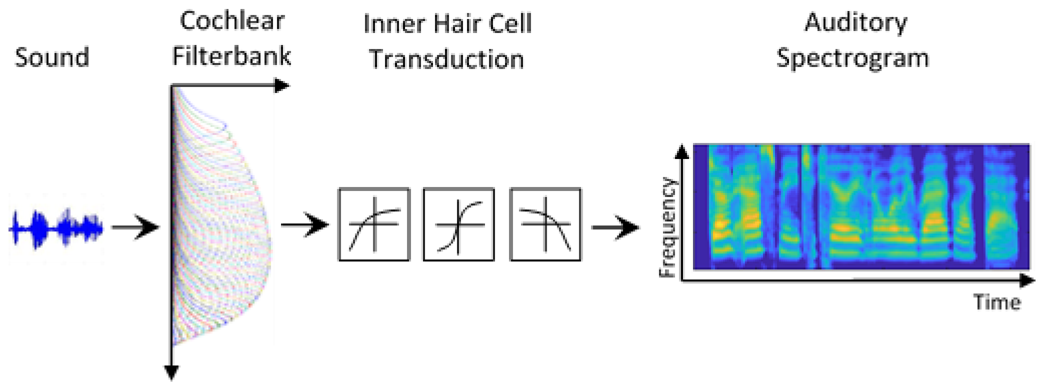

Figure 2: Block diagram of the auditory system

MATLAB Modelling Tasks

Perform the following tasks in MATLAB:

1. The code in the original scripts in gammatonegram.zip contains errors. Correct all errors.

2. Modify the sample code in gammatonegram.zip according to your specifications from Table 2 based on the right-most digit in your student number. After your changes are introduced, **ensure** the following:
   - The heights of the two spectrograms generated by default are the same as the number of channels in your settings.
   - The lowest centre frequency in your gain response display should be within ±8 Hz of your lowest centre frequency setting.

Right-most index of your student number	Number of channels, N (gammatone filters) (Task 2)	Gammatone filter with lowest centre frequency (Task 10)	Gammatone filter order, n (Task 14)
0	90	60 Hz	2
1	92	70 Hz	3
2	94	80 Hz	4
3	96	90 Hz	5
4	98	100 Hz	6
5	100	110 Hz	2
6	102	120 Hz	3
7	104	130 Hz	4
8	106	140 Hz	5
9	108	150 Hz	6

Table 2: Cochlear filterbank specification

3. The original script demo_gammatone.m from gammatonegram.zip generates two spectrograms in MATLAB figure 1 . Change the contents of demo_gammatone.m so that these spectrograms appear in MATLAB figure 2. Display the spectrogram generated by the gammatone filterbank at the top half of the figure and change the graph title to reflect "**Gammatone Spectrogram**". Display the spectrogram generated from the short-time Fourier transform (STFT) at the bottom half of the figure and change the graph title to "**STFT Spectrogram**". Label all axes accordingly.

4. In the same script file, generate a time vector (a vector is also known as an array) $t1$ that contains numbers in the range of [0 to $(T-1)]/sr$. Ensure the division by sr is done after generating the vector 0 to $T-1$. Note that T is the `length` (the number of samples, not time duration) of the sound signal stored in sa2.wav that is found in gammatonegram.zip and sr is the sampling rate of the sound signal. Do not display $t1$ in the command window. Hint: Its size is the same as the sound signal.

5. Generate another time vector $t2$ that contains m number of samples in the range from 0 to the time duration of the sound signal in sa2.wav. Here, m is a fixed number dependent on the **`length`** of the spectrogram (choose either one spectrogram as they are both the same length). Do not display $t2$ in the command window. Hint: The last values in $t1$ and $t2$ must be the same.

6. Generate two temporal profiles – one from the existing gammatone spectrogram and another from the existing STFT spectrogram. A temporal profile can be generated by **summing** all the rows of a spectrogram. **Normalise** (scale) each of the two temporal profiles so that the **maximum** value in both temporal profiles is 1.

7. Generate two spectral profiles – one from the existing gammatone spectrogram and another from the existing STFT spectrogram. A spectral profile can be generated by **summing** all the columns of a spectrogram. **Normalise** (scale) each of the two spectral profiles so that the **maximum** value in both spectral profiles is 1.

8. Use the sound signal found in sa2.wav provided in gammatonegram.zip, as input to your model. In MATLAB, display the waveform of the sound signal with respect to the time vector $t1$ in figure 1 and label the x-axis to reflect time in seconds and the y-axis to reflect amplitude (unitless). Add an appropriate title to the graph. Do not display anything else in figure 1.

9. Calculate and plot the average power signal (in Watts) of the gammatone and STFT spectrograms at the top half and bottom half, respectively, in MATLAB figure 3. Use the $t2$ time vector for the x-axis of both graphs. Label the axes and title the graphs. Hint: See the documentation of the **`bandpower`** command – the average power is to be computed independently for every column in each of the two spectrograms.

10. Figure 1 displayed in this document above, is a screen capture of MATLAB figure 2 generated from the original demo_gammatone.m. In this figure, the gain response of every 5th channel of the gammatone filterbank is displayed. Generate and display the gain response $g1$ (the equation has already been implemented for you in the second argument of the **`plot`** line in demo_gammatone.m) of **all** the channels in the gammatone filterbank on a linearly scaled x-axis and the same response on a logarithmically scaled x-axis in figure 4 in MATLAB. Plot the linearly-scaled gain response at the top half of MATLAB figure 4 and the log-scaled gain response below it. In the graph, your settings from Table 2 can be checked by inspecting the peak of the first filter (left-most curve). This value should be within ±8 Hz of your setting from Table 2. The peak of the last filter (right-most curve) should be close to but less than the Nyquist frequency of 8 kHz. Label all axes.

11. Display the centre frequencies of every 5th channel from the gain response in the command window using **`fprintf`** with the help of a **loop**. The centre frequencies

are the **maximum** values of every channel in the gain response. The following string should be displayed in a new line in the command window for every 5th channel: "Centre frequency of channel n: f_c Hz", where n is the channel number and f_c is the centre frequency.

12. Display the two temporal profiles generated from task 6 and the two spectral profiles generated from task 7 in figure 5. The x-axis of each temporal profile should be displayed with respect to $t2$ (in seconds). The x-axis of each spectral profile should be displayed with respect to F and $F2$ vectors (in Hertz) that correspond to the two spectrograms – these two vectors have been automatically generated for you in demo_gammatone.m. The amplitudes (y-axis) for all four graphs are unitless. Display:
    a. The spectral profile from the gammatone spectrogram on the top-left corner in MATLAB figure 5;
    b. The spectral profile from the STFT spectrogram on the top-right corner in MATLAB figure 5;
    c. The temporal profile from the gammatone spectrogram on the bottom-left corner in MATLAB figure 5.
    d. The temporal profile from the STFT spectrogram on the bottom-right corner in MATLAB figure 5.

13. Use the 2D correlation coefficient (CC) to show the quantitative difference between the following sets of signals (note that only one CC should be generated per comparison). Use `fprintf` to display the comparisons below one line at a time in your command window.
    a. Gammatone spectrogram versus STFT spectrogram.
    b. Gammatone spectrogram bandpower versus STFT spectrogram bandpower.
    c. Gammatone spectrogram temporal profile versus STFT spectrogram temporal profile.
    d. Gammatone spectrogram spectral profile versus STFT spectrogram spectral profile.

14. Use symbolic variables and display the impulse response of an n-order gammatone filter. Then, use MATLAB to substitute the n variable with the value of n found in Table 2 based on the right-most index of your student number. Also, use MATLAB to substitute the numeric centre frequency for channel 10 into the f_c variable. Display the equation in 3 significant figures. The impulse response equation is defined by $g(t)$ in the Auditory Signal Processing.pdf slides.

15. Find and display the transfer function of the impulse response from task 14 in the s-domain.

16. Using the MATLAB instantaneous features algorithms on the gammatone spectrogram, segment each word uttered in sa.wav. Show the start point and end

point of every word in the temporal profile using vertical indicators. Implement your code in the script demo_gammatone.m. Display the accuracy of your spoken word detector.

17. A gammatone filter can be approximated with a bandpass filter (BPF). Design a passive RLC circuit in MATLAB Simscape Electrical that simulates a BPF for channel 10 – use the maximum frequency of the gain response to be the centre frequency of the BPF. Display all input and output signals. Use the following input signals:
    a. 2 Volt peak-to-peak 2-second chirp signal with a ramp frequency from 1 Hz to 2000 Hz.
    b. 1.5 Volt peak-to-peak 2-second complex tone signal comprising $f_c(10)$-50 Hz, $f_c(10)$ Hz, and $f_c(10)$+50 Hz where $f_c(10)$ Hz is the centre frequency of the 10th channel.

Add comments to the code you have modified or introduced in MATLAB. Submit only the MATLAB files that you have modified in vUWS.

Hints

- Implement all new code in the MATLAB script file demo_gammatone.m found in gammatonegram.zip.
- The `blue` words in the previous section indicate the appropriate MATLAB functions that can be used for the associated task.
- You must only modify two MATLAB script files from gammatonegram.zip: demo_gammatone.m and another script file from the zip file that you have changed based on the requirements of the tasks above.
- When submitting your progress report, only submit MATLAB script files you have modified. Do NOT submit files you have not changed.
- The code and circuits in the workshop lectures and tutorials are relevant to the tasks above. Therefore, it is **highly recommended** to use them and/or change them accordingly for the tasks above. If you use any code outside the workshop lecture and tutorials, you must explain their functionalities in your report.
- Do NOT generate spectrograms. Instead, only use the variables that represent spectrograms in demo_gammatone.m – you will need to investigate what these variables are.

Progress Report (25%)

You are expected to complete up to task 9 from the MATLAB model section. Prepare a 2000-word progress report on the tasks – follow the guidelines given in the learning guide where relevant. Include an introduction in the report as per the final report guideline. Describe what you have done to complete the tasks. If you are unable to complete any task, explain what you are experiencing. You **must** include inline citations where necessary (otherwise, it will be interpreted as plagiarism). Use any online English grammar and vocabulary checking application to ensure that your report is coherent and clear, e.g. Grammarly – marks will be given if you can convey your ideas **clearly** and **concisely**.

Submit your **progress report** and only the **MATLAB script files** you have modified from gammatonegram.zip in vUWS under "Assessment 1".

Final Report (50%)

Prepare a 4000-word final report with the structure outlined below. The final report should include the images of all the MATLAB command window results and figures (screen capture – do not use your phone to capture any images) based on the 17 tasks. You **must** include inline text citations where necessary (otherwise, it will be interpreted as plagiarism). Use any online English grammar and vocabulary checking application to ensure that your report is coherent and clear, e.g. Grammarly – marks will be given if you are able to convey your ideas **clearly** and **concisely**. Do **not** include any code in your report.

The sections to be included in the final report are:

1. Introduction. Include a brief background of an auditory model with references. Also, include objectives as outlined in the first paragraph of this document. Additionally, you can include a motivation statement on why this project is important.
2. Components of the auditory model.
3. Modelling the auditory model using MATLAB using the specification from Table 2 clearly described. Also, mention the filter order from Table 2 required to show the gammatone impulse response. Additionally, address the following based on tasks 16 and 17:
    a. Present your design and the selection of the statistical algorithms and their parameters to detect the spoken words in sa.wav.
    b. Discuss your circuit design and the selection of the parameters set in the circuit including the centre frequency and the assumed setting and calculation of the R, L, C parameters.
4. Results and discussion [screen capture of all the; screen capture of MATLAB command window showing correlation coefficients (CC) and the symbolic equations]. Discuss on the CC results to indicate the degree of difference between pairs of vectors and matrices in tasks 12.
    a. Address which CC result is highest and thus, most similar.
    b. Conversely, address which CC result is lowest and thus, least similar.
    c. How is the temporal profile signal related to the original sound signal? Use your observation and any key terminologies to describe their association.

    For task 16, address the success of word detection in percentage based on the number of words detected by your algorithm to the total number of words spoken (ground truth) – discuss either the shortcomings or benefits of your algorithm. For task 17, address whether the centre frequency to be detected can be observed from the fundamental period in your output graphs – discuss either its shortcomings if the fundamental period is not similar to the centre frequency or vice-versa if it is similar.

5. Conclusion (discuss your experience in using MATLAB for modelling the auditory model, its usefulness, and difficulties).
6. References. Use Harvard-style referencing with **inline citations**.

Submit your **final report** and your **MATLAB files** that you have modified from the zip file in vUWS under "Assessment 3". Do not submit other files that you have not modified from the zip file.

Resources

- Signal Processing Toolbox.
- Audio Toolbox.
- Auditory Filterbank Sample Code.
- Auditory Filterbank Documentation.
- Simulink.
- Simscape Electrical.

References

Alm, J. F. and Walker, J. S. (2002) 'Time-Frequency Analysis of Musical Instruments', *SIAM Review*, 44(3), pp. 457–476. doi: 10.1137/S00361445003822.
Edwards, B. (2007) 'The Future of Hearing Aid Technology', *Trends in Amplification*, 11(1), pp. 31–45. doi: 10.1177/1084713806298004.

Goto, M. (2006) 'Music Scene Description', in Klapuri, A. and Davy, M. (eds) *Signal Processing Methods for Music Transcription*. New York, USA: Springer US, pp. 327–359. doi: 10.1007/0-387-32845-9_11.

Heil, P. and Peterson, A. J. (2015) 'Basic response properties of auditory nerve fibers: a review', *Cell and Tissue Research*, 361(1), pp. 129–158. doi: 10.1007/s00441-015-2177-9.

Katsiamis, A. G., Drakakis, E. M. and Lyon, R. F. (2007) 'Practical Gammatone-Like Filters for Auditory Processing', *EURASIP Journal on Audio, Speech and Music Processing*, 2007, pp. 1–15. doi: 10.1155/2007/63685.

Lopez-Poveda, E. A. and Meddis, R. (2001) 'A Human Nonlinear Cochlear Filterbank', *The Journal of the Acoustical Society of America*, 110(6), pp. 3107–3118. doi: 10.1121/1.1416197.

Lyon, R. F. (2017) 'The CARFAC Digital Cochlear Model', in *Human and Machine Hearing: Extracting Meaning from Sound*. Cambridge University Press, pp. 293–298. doi: 10.1017/9781139051699.020.

Lyon, R. F., Katsiamis, A. G. and Drakakis, E. M. (2010) 'History and Future of Auditory

Filter Models', *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, pp. 3809–3812. doi: 10.1109/ISCAS.2010.5537724.

Meddis, R. and O'Mard, L. (1997) 'A unitary model of pitch perception', *The Journal of Acoustical Society of America*, 102(3), pp. 1811–1820. doi: 10.1121/1.420088.

Meddis, R., O'Mard, L. P. and Lopez-Poveda, E. A. (2001) 'A Computational Algorithm for Computing Nonlinear Auditory Frequency Selectivity', *The Journal of the Acoustical Society of America*, 109(6), pp. 2852–2861. doi: 10.1121/1.1370357.

Plack, C. J. and Carlyon, R. P. (1995) 'Loudness Perception and Intensity Coding', in Moore, B. C. J. (ed.) *Hearing - Handbook of Perception and Cognition*. 2nd edn. San Diego, New York, Boston, London, Sydney, Tokyo, Toronto: Academic Press, pp. 123–160. doi: 10.1016/B978-0-12-505626-7.50018-2.

R. Mergu, R. and K. Dixit, S. (2011) 'Multi-Resolution Speech Spectrogram', *International Journal of Computer Applications*, 15(4), pp. 28–32. doi: 10.5120/1937-2587.

Shamma, S. (2003) 'Encoding Sound Timbre in the Auditory System', *IETE Journal of Research*, 49(2), pp. 145–156. doi: 10.1080/03772063.2003.11416333.

Tchorz, J. and Kollmeier, B. (1999) 'A model of auditory perception as front end for automatic speech recognition', *The Journal of Acoustical Society of America*, 106(4), pp. 2040–2050. doi: 10.1121/1.427950.

Xu, Y. *et al.* (2018) 'A FPGA Implementation of the CAR-FAC Cochlear Model', *Frontiers in Neuroscience*, 12(April), pp. 1–14. doi: 10.3389/fnins.2018.00198.

Xu, Y. *et al.* (2019) 'A Binaural Sound Localization System using Deep Convolutional Neural Networks', in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. Sapporo, Japan, Japan: IEEE, pp. 2–6. doi: 10.1109/ISCAS.2019.8702345.