Major Assignment BIOM9660

Due before midnight on 9 October, 2016

Introduction

Fig. 1 (adapted from Feng et al. [1]) illustrates the importance of the algorithms utilised in cochlear implant sound processing strategies – that is, the translation from a captured sound to electrical impulses delivered via metal electrodes placed within the cochlea.

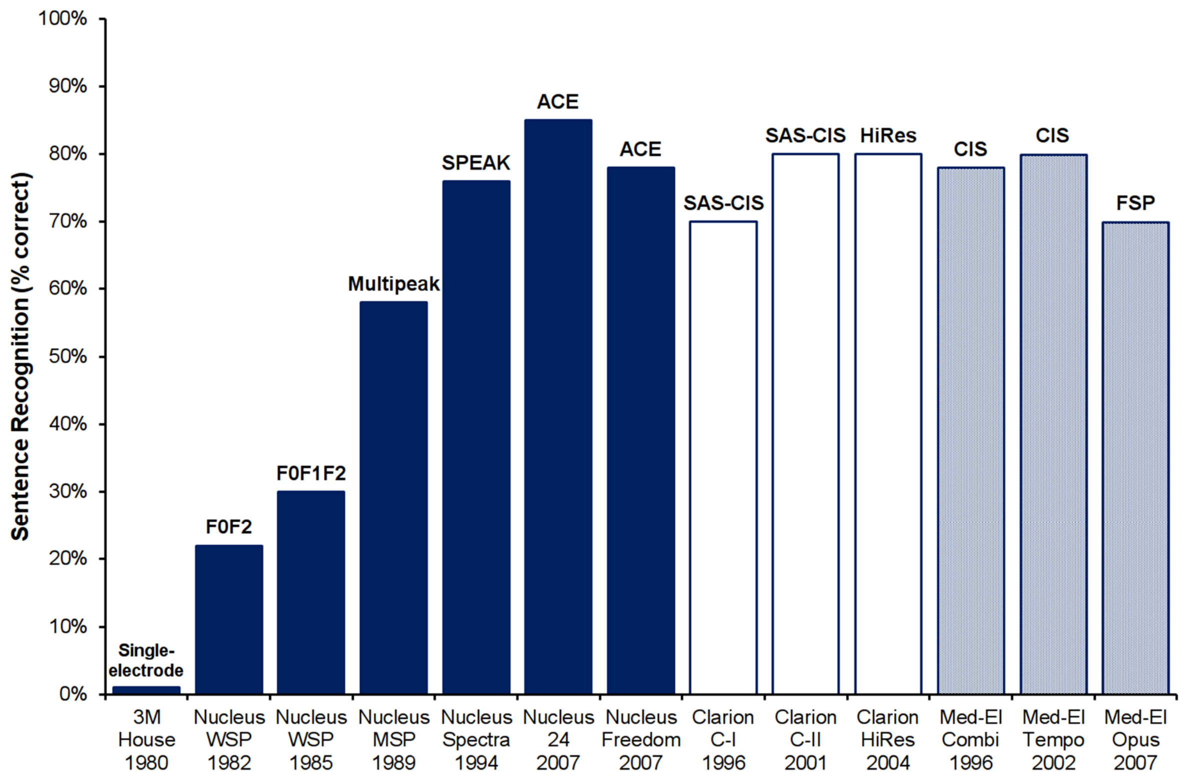

Figure 1 - Performance of various sound processing approaches in cochlear implant recipients

In the BIOM9660 laboratories, formant-based sound processing strategies will be explored. In Fig. 1, these strategies are represented by the F0F2 and F0F1F2 strategies utilised of the 1980s. In those strategies, the electrodes closest to format 1 and 2 were stimulated according to the amplitude of the relevant format. The stimulation rate was at F0[1].

Beyond these, the features of the sound were reassessed with a focus on finding the most important components of the sound as opposed to its dominating frequencies. This latter approach made the difference between cochlear implants being essentially an aide to lip-reading and finding their place in human achievement as a means of truly restoring hearing in their own right.

Practically speaking, the transition from the formant-based processing strategies to the spectral peak strategies allowed cochlear implant recipients to (among other important things) regularly

[1] Note that in this assignment, the stimulation rate will be fixed. Thus, this feature of the F0F2 and F0F1F2 strategies will be replaced with a fixed stimulation rate.

converse on the telephone.

For those of us who can hear well, it may be useful in relating this to our own experience by considering the challenges of learning a new language. In the early stages of learning, one might expect to detect the occasional word in the new language – akin to the 3M/House single electrode implants of four decades ago. With a bit of practice, some – perhaps several - very basic words can be understood or recognised, but the full scope of the sentences using these words are only occasionally understood because of the missing parts of the sentence that have not yet been learned. This is akin to the state of the art in the 1980s with the formant-based strategies.

When the spectral peak approaches came as we entered into the 1990s, it was as if overnight the learning phase of the new language was almost over, and suddenly more than 75% of everything became understandable.

Notice that for the past two decades, the advances have become small and in some cases less successful than their preceding strategies. Perhaps your role in the future will be to deliver that remaining 25% of sound that will make the difference between being able to understand, to being fluent in the language of hearing.

Each student will study and implement into practice three sound processing strategies:

F0F1F2, SPEAK, and CIS. Note that the latter two are similar "N of M" strategies that vary primarily in their presentation rather than processing, but they have a separate history, and are both essential components of modern cochlear implant therapy. Similarities and differences should be noted carefully and described in the various reporting steps described below.

Expectations:

The Major Assignment will be in four parts. Two parts will be assessed on the individual performance of the student (60% of the total mark), and two parts will be assessed on the performance of the group to which students will be assigned (40% of the total mark):

1. Unique and individual MATLAB code written by the student to implement the sound processing strategies listed above. NB: The coding <u>MUST</u> be done using the provided template.
2. An original, individually-written report describing operation and performance of your code.
3. A study prepared by your group that compares the performance of the sound processing strategies. This work will be described in the form of a poster.
4. A group presentation that describes the contents of your poster.

Individual Contributions:

The MATLAB code and report components of the BIOM9660 Major Assignment are to be your own work – no exceptions. There are no restrictions on working together in order to learn – in fact, this is encouraged. However, when it comes to producing work to be submitted for assessment, work by others must be appropriately cited, and work that is presented as being your own must be entirely in your own words. Similarly, any code (aside from the code included within the template) must be entirely your own work and other than built-in MATLAB functions, no code from anyone other than yourself is to be submitted.

To ensure that this happens, students will be required to submit their work in two parts via the TurnItIn facility in Moodle:

1. An original, individual report.
2. Your MATLAB code

Both will be compared with others in the course as well as online resources.

Students are reminded that plagiarism is absolutely, positively unacceptable and a zero tolerance approach is taken in this regard in BIOM9660. TurnItIn is remarkably good at what it does – that is, finding plagiarised material. If you need to ask whether or not you've plagiarised, you probably already have. Don't do it. If you don't know what it is, find out here: https://student.unsw.edu.au/plagiarism

Group Contributions:

This part of the Major Assignment will see each group follow a scientific protocol in order to assess the relative performance of various sound processing strategies by presenting normally hearing individuals processed sounds via a 'vocoder' that translates stimulation commands into sounds.

The outcomes of the study are to be presented in the form of a poster (one poster per group) and a presentation which may be a prepared video, live demonstration, multimedia display, etc. The idea is for the presentation to be fun, light-hearted and entertaining – as well as informative.

MATLAB code (one per student)

Students must strictly adhere to the coding template available on the Moodle site. This template provides a skeleton of the functions that you must write, a standard test method and standardized output files.

Refer to Appendix A for additional information on the code skeleton, installation and submission.

Detailed comments relating to your own code are needed in order to guide the marker through what is taking place. If the marker cannot understand the code, the marker cannot provide marks – help them, help yourself.

1. The code must determine the sampling frequency of the sound file and, where required, resample the sound to ensure that the data contained within the file is managed appropriately within the code. The standard sampling frequency to be used within the code is 16 kHz although the code should be able to manage (and re-sample) any sound from 4 kHz to 64 kHz. HINT: resample()
2. The code is to process the sound into individual, 2 mS epochs in time with overlap of 6 mS (3 mS before and 3ms after) according to a Hann function. HINT: hann() and buffer() https://en.wikipedia.org/wiki/Hann_function
3. Each epoch is to be processed according to the sound processing strategy specified in the command line.
4. The output of the sound processing for each epoch is to be formatted into the following:
    - Electrode number (1-16 where 1 is the apical electrode)
    - Stimulus amplitude range (A = 0-1024). To account for the limited dynamic range available in practice, the useful dynamic range is to be limited to 10 dB - that is, the

difference between the T and C level represents 10 dB of dynamic range. Accordingly, $A_{MAX}/A_{MIN} = 10^{(10/20)}$. So that we can compare outcomes, fix the C level at A_{MAX} (1024).

5. The above output is to be stored in a CSV file (16 rows long, with each column representing 2 mS of the sound). This structure is called a "Frequency/Time Matrix" (FTM)
6. The full sound is to be presented in two formats:
   - an electrodogram with a heat-map display similar to the sound processing lab. The apical electrode should be at the top of the plot.
   - A plot of the original sound in the time domain.
7. All plots must be titled, and all axes on all plots (whether displayed or stored into a file) must be labelled appropriately and a descriptive legend displayed were appropriate.

Report (one per student)

Students will be provided with a set of sample sounds to pass through their code for purposes of comparison and marking. The processed output of one of those sounds (to be specified) is to be presented in a relatively brief - maximum of four pages – report in the IEEE conference format: http://embc.embs.org/2014/wp-content/uploads/2013/08/4-page-template-2-column-format.doc to be submitted as a single ".pdf" file.

The report must contain:

- A description of the operation of the sound processing strategies studied including their history, rationale, technical operation, and performance as determined from a minimum of six refereed journal or conference publications. Citations from websites do not count towards the six.
- Electrodogram plots of the specified sound in each of the stimulation strategies.
- A time domain plot of the specified sound.
- A single flow-chart describing the operation of your code.
- A comprehensive set of references from which you obtained the information required to complete your work.
- Headings within your report are to be: ***Abstract, Introduction, Methods, Results, Discussion, Conclusion.***

Group Poster (one per group)

From the individual assignment, each group will have a source of encoded sounds via each strategy that can be played back via the BIOM9660 vocoder and interpreted by normally hearing individuals.

One group member will serve as the study coordinator and will not be tested. The performance of all other members of the group will be assessed for sentence recognition. Each subject must remain 'naïve' to each of the sounds so it is therefore crucial that the study coordinator isolate subjects from the sounds – both processed and in their original form – until it is time that they are tested.

The coordinator will sit the subject in a quiet place in a comfortable chair and play a series of sample sounds to ensure that volumes are comfortably adjusted, and the clarity of the original sounds is such that the sentence contained within the sound is readily recognised by the subject. These same sample sounds will then be presented to the subject in vocoded form so they know what to expect when the data acquisition begins.

The coordinator will then play from a pre-defined playlist processed sounds that have not been presented to the subject previously in any form. The coordinator will ask the subject to repeat to them the sentence as they understood it to be, and record whether or not the sentence was identified entirely correctly, and if not how many words were correct. This process will be repeated for the entire playlist of 99 sounds, with an equal (+/- 1) representation of the various sound processing strategies tested.

Lastly, the entire playlist will be played in their original (unprocessed) form. Subjects will repeat each sentence back to the study coordinator who will record whether or not the sentence was correctly recounted. In the event that any sentence is not recounted correctly, the group must decide how best to manage this situation in the presentation of their results.

This process will be repeated for all subjects within the group with the exception of the study coordinator.

The group will share responsibility for data compilation, processing and analysis. The mean and standard deviation of the sentence recognitions for each processing strategy are to be displayed on a histogram with error bars representing the standard error of the mean (SEM) of the individual performances. For purposes of comparison, an additional histogram bar should show the unprocessed sound recognition. A test of statistical significance is to be applied to the compiled set of data, and where statistical significance is achieved, this should be noted in its presentation.

A description of your study including your hypotheses, methods, results, interpretation and conclusions is to be presented in poster form. A poster template is provided in Moodle.

Raw data is to be saved in a log-book and submitted upon request by the markers.

Group Presentation (one per group)

A particularly challenging aspect of disseminating data is doing so in a creative, engaging, provocative and informative way. Accordingly, your assignment is to present your data in a creative, engaging, provocative and engaging way. This may include, but is not limited to the following:

1. A video
2. A mock news report
3. A skit

References

[1] Zeng, F-G. Rebscher, S. Harrison, W.V. Sun, X. Feng, H. Cochlear Implants:System Design, Integration and Evaluation. *IEEE Rev Biomed Eng.* 2008 January 1; 1: 115–142.

APPENDIX A – Code template, testing and submission

MatLab file provided

A zip file containing a Matlab "code skeleton" for your cochlear implant simulation will be provided on Moodle. The zip file should be downloaded and expanded into a single folder on your PC.

You will now have the following files: -

1. **cochlearProc.m** This function is used to run your simulation. It requires two parameters: -
   - soundFileName: the name of a .wav file containing sound of any length up to 60 seconds in duration and
   - processingType: a choice of stimulation strategy (1 = F0F1F2, 2 = SPEAK, 3 = CIS).

This function will create the following output files:-

<processing name>.csv	Stimulus file
<processing name>.jpg	Electrodogram
<processing name>_speech.jpg	Speech plot
<processing name>.wav	Vocoded reconstruction of sound from the stimulus file

<processing name> will be replaced by the relevant string. For example 'F0f1f2'

Initially these files will be blank or empty. When you have written the functions below then the files will contain the results of your simulation.

2. **classCochlear.m** – This file contains the definitions of 6 functions that you must write. Each function header contains comments to explain what the function should do. The functions that you will write are:

getWav()	getFTM()	process()
applyDR()	plotSignal()	plotElectrodogram()

3. **classCochlearSupport.m** – This file contains some helper functions and data. For example…

```
fSample       = 16000; % required sampling frequency
fTolerance    = 0.1;   % allowed variation from fSample
tSample       = 0.002; % sample time for cochlear implant processing
frameOverlap = 3/4;    % required overlap when using Hann window
numElectrodes= 16;     % number of electrodes in cochlear implant
numFormants   = 2;     % formants used by the F0F1F2 strategy
dynamicRange = 10;     % dB
maxOutput     = 1024;

function result = procName(obj, procType)
% return string containing the name of the process type <procType>
```

To access the data/function use the syntax obj.<name> eg obj.numElectrodes.

This file also contains functions used by cochlearProc.m such as writeCsv(), writeJpg() and vocoder().

You are welcome to inspect these and any functions provided. You should not change any provided functions or data definitions as any changes outside classCochlear.m are <u>not</u> submitted for testing.

Validating your installation

From MatLab set you current directory to be the one where you expanded the zip file containing the code skeleton.

To validate the code skeleton is correctly installed run a dummy execution of the software by typing

```
cochlearProc('sound.wav', 1)
```

The following output should be generated.

```
Processing sound.wav with strategy F0f1f2
    >>classCochlear.getWav
    >>classCochlear.getFTM
    >>classCochlear.process
    >>classCochlear.applyDR
    >>classCochlear.plotSignal
    >>classCochlear.plotElectrodogram
    >>classCochlearSupport.writeJpg
    >>classCochlearSupport.writeJpg
    >>classCochlearSupport.writeCsv
    >>classCochlearSupport.vocoder
    >>classCochlearSupport.writeWav
Warning: wav data was empty
> In classCochlearSupport>classCochlearSupport.writeWav at 126
  In cochlearProc at 67
```

Additionally, two blank matlab figures will be created and the empty file F0f1f21.csv and two blank files named F0f1f21.jpg and F0f1f21_speech.jpg.

If the output is not as expected, please email a lab demonstrator with your output attached.

Code submission

It is recommended that you make a backup of classCochlear.m before completing the next steps.

Once the backup is complete, copy the file classCochlear.m to class<z number>.m and class<znumber).txt.

For example, classz1234567.m and classz1234567.txt. The letter "z" should be in lower case.

These two files only should be submitted via Moodle.

Important Notes: -

1. **The only file that you may change is classCochlear.m. This is the <u>only</u> code that you will submit for marking.**
2. **The function declarations in classCochlear.m must not be changed.**
3. **The jpg plots that are generated must be the plots that are used in your report.**