

Parte 2

Nesta parte do laboratório foi pedido que se implementasse um sintetizador de formantes de vogais.

1.

Tendo por base a equação da função de transferência de um ressonador em termos da frequência das formantes f_k e da largura de banda b_k :

$$T_k(z) = \frac{A_k}{1 - B_k \times z^{-1} - C_k \times z^{-2}}$$

$$C_k = -e^{-2\pi \times b_k \times T_k}$$

$$B_k = 2 \times e^{-\pi \times b_k \times T_k} \times \cos(2\pi \times f_k \times T_s)$$

$$A_k = 1 - B_k - C_k$$

Utilizamos $T_s = 8000 \text{ Hz}$, mesmo considerando que inicialmente se tratavam de ficheiros de entrada com uma frequência de amostragem de 16KHz.

Para sintetizar uma vogal criou-se a função *FormantSynthesis* (*vowel*, *f0*, *duration*, *intensity*) usando o Matlab. Esta função recebe como parâmetro inicial, uma vogal que pode ser um número de 1 a 9 ou um carácter, por exemplo:

- a = 1
- E = 2
- i = 3
- O = 4
- u = 5
- 6 = 6
- e = 7
- o = 8
- @ = 9

Nesta função usamos como base a seguinte expressão derivada das expressões anteriores:

$$Z_k = e^{(-\pi \times b_k + j \times 2\pi \times f_k)}$$

Em modo de teste correu-se a função com estes parâmetros respetivamente ***FormantSynthesis*** ('u',230,1,100), obtendo assim o ficheiro `formant_synthesis_fixed.wav`. Correndo o comando de Matlab ***help FormantSynthesis*** (ou qualquer outra função criada pelo grupo) é possível obter informações relevantes sobre o seu básico funcionamento.

No nosso código para esta parte temos um conjunto de 5 funções auxiliares que executam apenas verificações de parâmetros de entrada, prevenindo assim o utilizador de inserir dados que prejudiquem o normal funcionamento das funções desenvolvidas. Os seus nomes são:

- `getFormants.m`
 - Tenta ler o ficheiro de `input.mat` que contem as formantes de cada vogal. Ou seja, cada linha representa uma vogal com as 4 formantes (F1, F2, F3 e F4 em Hz) pelo que tem um total de 9 linhas
- `checkInputVowel.m`
 - Faz a verificação do parâmetro de entrada vogal, assegurando que se trata de um valor inteiro entre 1 e 9
- `checkInput.m`
 - Faz a verificação dos parâmetros da frequência fundamental e duração, assegurando que são positivos
- `convertChar.m`
 - Converte o carácter da vogal num número, possibilitando ao utilizador fornecer um carácter ‘a’ ou ‘E’ em vez do seu número correspondente no nosso ficheiro de `input`.
- `headPhonesPrint.m`
 - Alerta o utilizador para que coloque os auscultadores e reproduz o ficheiro de saída ao fim de 3 segundos

2.

Para esta alínea, foi criada a função *FormantSynthVariations* (*vowel*, *f0Min*, *f0Max*, *duration*, *intensityMin*, *intensityMax*)

A diferença entre esta função e a anterior criada reside no facto de existir uma variação da frequência fundamental e na intensidade. Para valores de teste usámos valores que produzissem um ficheiro de output com uma variação de saturação pouco acentuada e uma variação na frequência fundamental de modo a que o som ficasse mais agudo.

Em modo de teste correu-se a função com estes parâmetros respetivamente ***FormantSynthVariations*** (**‘u’**, **160**, **165**, **0.5**, **90**, **100**), obtendo assim o ficheiro `formant_synthesis_var.wav`. Correndo o comando de Matlab ***help FormantSynthVariations*** (ou qualquer outra função criada pelo grupo) é possível obter informações relevantes sobre o seu básico funcionamento.

3.

Na terceira alínea desta parte a foi pedido que em vez de uma sequência de 2 vogais em vez de uma vogal. O grupo escolheu as vogais ‘o’ e ‘i’ e criou a função `SynthesizeSeq(firstVowel, secondVowel, f0, duration, intensity)`.

A função neste caso realiza 2 iterações da função inicialmente criada para a alínea 1. Sintetizando uma vogal de cada vez e reproduzindo ambas em sequência após a sua construção.

Em modo de teste correu-se a função com estes parâmetros respetivamente **SynthesizeSeq** ('o', 'i', 230, 0.9, 100), obtendo assim o ficheiro `formant_synthesis_seq_oi.wav`. Correndo o comando de Matlab **help SynthesizeSeq** (ou qualquer outra função criada pelo grupo) é possível obter informações relevantes sobre o seu básico funcionamento.

4.

Para esta alínea o grupo utilizou a função criada na alínea anterior, ou seja, **FormantSynthesis**.

Utilizando o Wavesurfer obtiveram-se novos valores para as 3 primeiras formantes e criou-se um novo ficheiro `u.mat` com uma linha apenas para teste, que corresponde à alteração das ondas das 3 formantes em mais 200Hz para cada uma relativamente à vogal `u`.

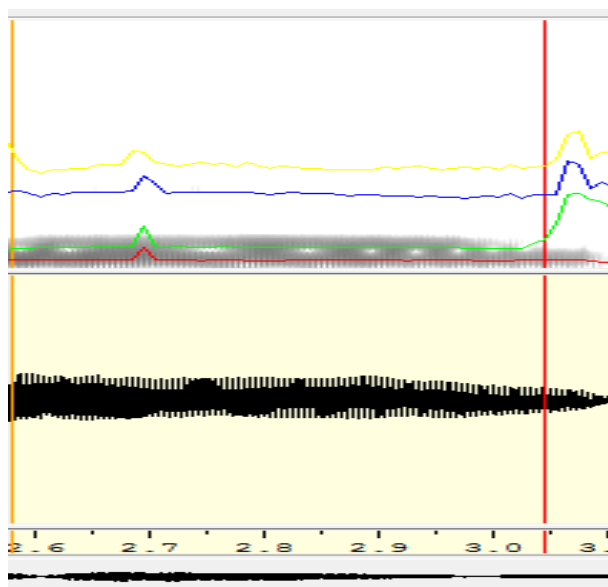


Figura 1 – Modificação dos parâmetros das formantes da vogal `u`

De seguida correu-se a linha de código no Matlab **FormantSynthesis (1,230,1,100)**, o ficheiro de output chama-se `part2_4.wav` e contém um som ligeiramente diferente. A JND (Just-noticeable difference) que se conseguiu observar na modificação na vogal foi uma sensação de ligeiro aumento da intensidade do ficheiro de áudio. O que efetivamente está a acontecer é a tendência de aproximação de outra vogal, que leva o sintetizar a tentar sintetizar outra vogal, daí a nossa sensação de aumento da intensidade relativamente ao ficheiro original.