Práctica 7. Implementación de un sintetizador concatenativo de difonos.

Manuel Moreno Benabat

Contenido

Objetivo

Este trabajo práctico consiste en implementar un sintetizador concatenativo de difonos para el lenguaje **L**.

L es un lenguaje compuesto por seis fonos:

- Vocal: [e].
- Consonantes: [f], [k], [m], [r], [s] y [t].

Estos fonos sólo se pueden agrupar para formar sílabas de los siguientes tipos:

- V: [e].
- CV: [fe], [ke], [me], [re], [se], [te].
- CCV: [fre], [kre], [tre].
- VC: [es].
- CVC: [fes], [kes], [mes], [res], [ses], [tes].
- CCVC: [fres], [kres], [tres].

Además, en L hay dos restricciones fonotácticas:

- El sonido r no puede ser inicial en una frase.
- El sonido r no puede suceder a una [s].

El sistema tendrá en su inventario exactamente una instancia de cada difono. No se realizará selección de unidades al sintetizar una nueva frase.

El sistema debe recibir como entrada un ‘string’ con la secuencia de fonos y generar como salida un archivo de audio conteniendo el habla sintetizada. En la secuencia de fonos pueden marcarse dos aspectos prosódicos:

- Si una vocal debe acentuarse, se introduce en mayúscula (‘E’). En caso contrario, con minúscula (‘e’).
- La secuencia de entrada puede terminar en el carácter ‘?’, en cuyo caso la salida deberá tener la prosodia de una pregunta.

La entrada debe representarse como una cadena de caracteres ASCII.

La entrada no puede contener espacios en blanco ni caracteres distintos de “eEfkmrst?”.

Tareas que realizar

Diseñamos y grabamos el inventario de sonidos (difonos). En mono, 16kHz y 16 bits.

El inventario de todos los difonos posibles que podemos formar con el lenguaje **L** es:

- /ef/, /Ef/
- /ek/, /Ek/
- /em/, /Em/
- /er/, /Er/
- /es/, /Es/
- /et/, /Et/
- /sf/
- /sk/
- /sm/
- /ss/
- /st/
- /-e/, /-E/, /e-/, /E-/
- /fe/, /fE/
- /ke/, /kE/
- /me/, /mE/
- /re/, /rE/
- /se/, /sE/
- /te/, /tE/
- /-f/
- /fr/
- /-k/
- /kr/
- /-t/
- /tr/
- /-s/, /s-/
- /-m/
- /ee/, /EE/, /Ee/, /eE/

Por las restricciones fonotácticas no tenemos en cuenta los difonos:

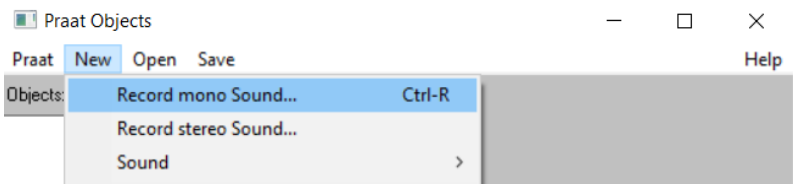
- /-r/: El sonido r no puede ser inicial en una frase.
- /sr/: El sonido r no puede suceder a una [s].

La grabación de los difonos se ha realizado enunciando frases que contienen palabras portadoras del conjunto de difonos.

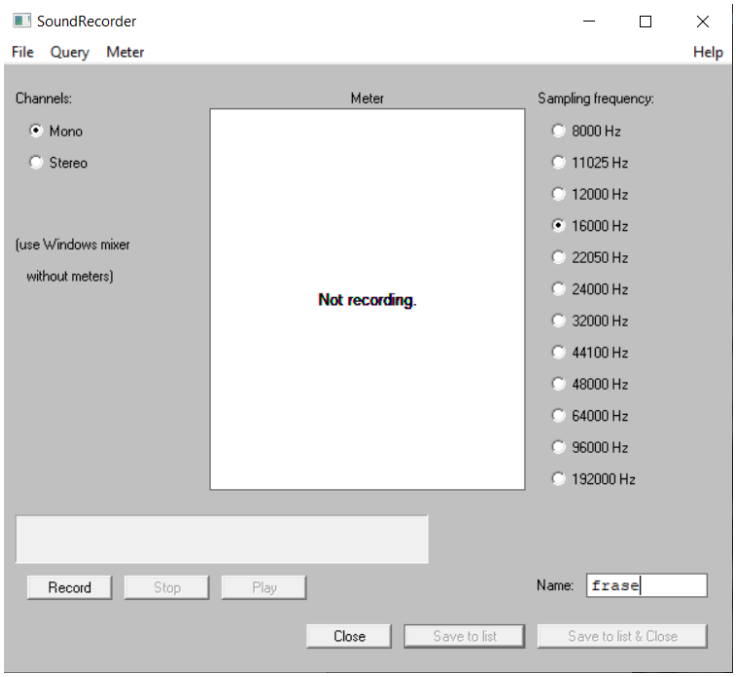
Por ejemplo, en la frase “esfera eskeleto esmeralda” tenemos las palabras portadoras para los siguientes difonos:

- {-e es sf fe er ra a-}: /-e/, /es/, /sf/, /fe/.
- {-e es sk ke el le et to o-}: /-e/, /es/, /sk/, /ke/.
- {-e es sm me er ra al ld da a-}: /-e/, /es/, /sm/, /me/, /er/.

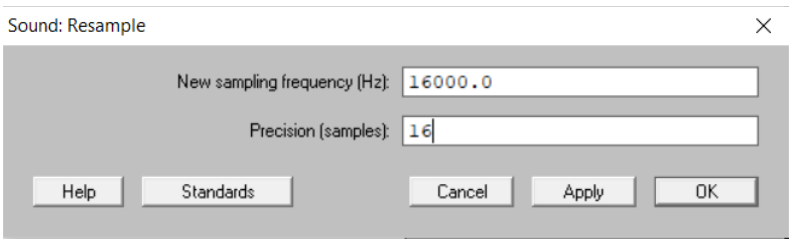
Para grabar la frase, abrimos Praat y creamos un objeto Sound mono.



Le asignamos características al objeto Sound y grabamos la frase con el botón Record.

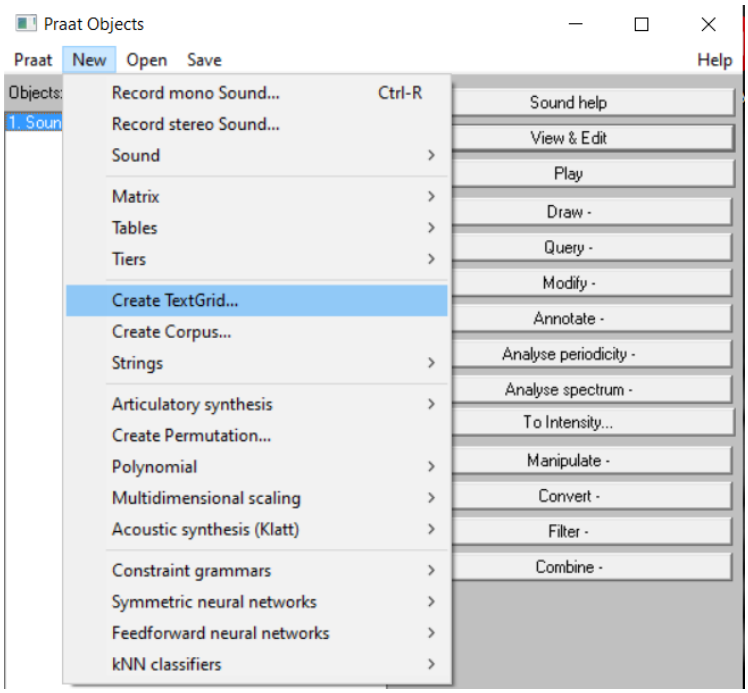


Una vez grabada, con el botón “Convert” de Praat, elegimos la opción “Resample” para poder tener una precisión de 16 bits en el muestreo de la grabación.

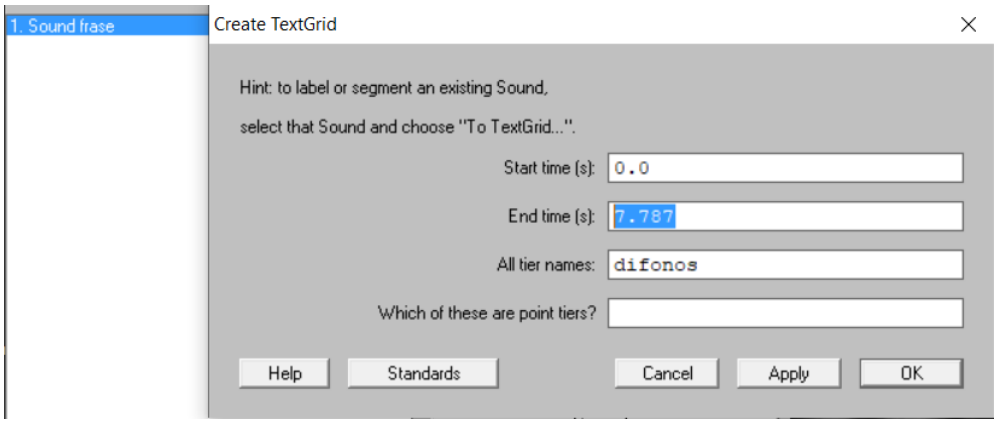


Etiquetamos los difonos con una capa de intervalos (interval tier) en un archivo TextGrid

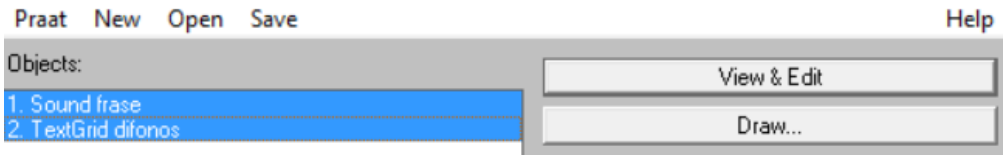
Creamos un TextGrid para etiquetar los difonos que aparezcan en la frase.



"End time" es la duración de la grabación en segundos. El TextGrid posee sólo una capa de intervalos (difonos).

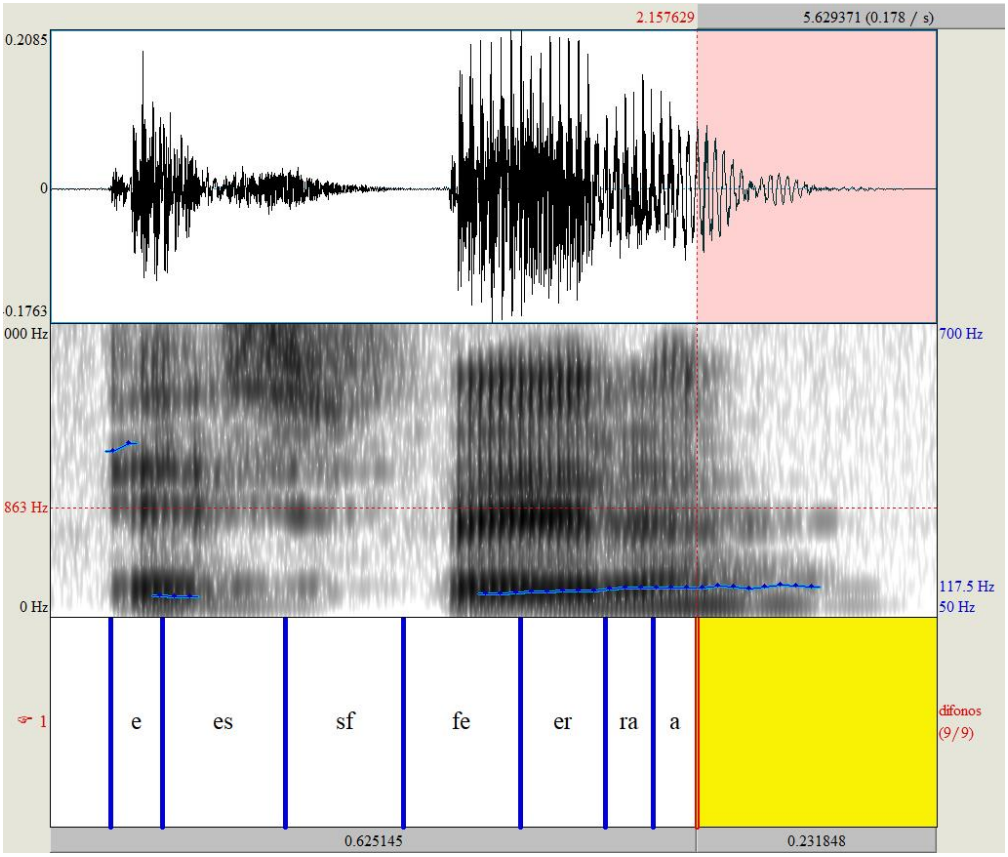


Editamos el TextGrid seleccionando el objeto Sound y TextGrid. Pulsamos el botón "View & Edit".



Seleccionando, una a una, las palabras portadoras en la frase grabada y pulsando “Ctrl+N” realizamos un zoom sobre esta palabra. Ya podemos editar el TextGrid por intervalos de dífonos.

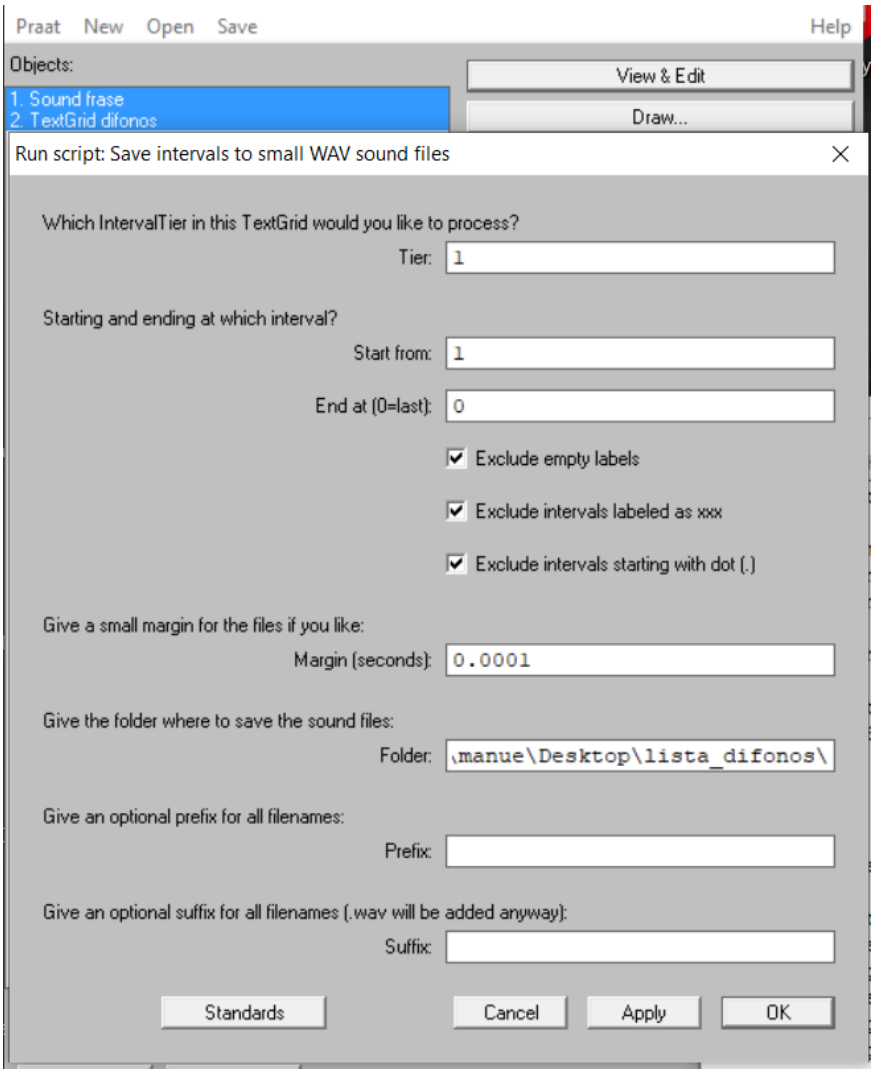
Los espacios entre palabras los dejamos sin etiquetar (en blanco) para que el script de recorte los obvie.



Recortamos los difonos y generamos un archivo '.wav' para cada uno

Recortamos los difonos con el script de Praat "sabe_labeled_intervals_to_wav_sound_files.praat".

Generando así un archivo '.wav' para cada intervalo (difono) marcado en el TextGrid. Usamos 0,0001 segundos en la opción "Margin(seconds)" del script.



Diseñar, grabar, etiquetar y recortar el inventario de difonos es la parte más pesada del sintetizador concatenativo. Hay que ser muy escrupulosos en este trabajo pues de él depende el resultado final de la concatenación.

Creamos un programa que, dada una secuencia de fonos, concatene los archivos de los difonos correspondientes, genere un archivo '.wav' y, de ser necesario, modifique su prosodia.

El programa se ha escrito en Python por la facilidad de uso de este lenguaje.

Existen tres restricciones en su implementación:

- Debe funcionar en modo batch (no interactivo), recibiendo como argumentos la secuencia de fonos a sintetizar y el nombre del archivo '.wav' a crear.
- La salida debe guardarse como un archivo '.wav' (mono, 16kHz, 16 bits).
- El programa no debe reproducir automáticamente el audio.

El programa se completa en cinco pasos:

1. Comprueba que el número de argumentos recibidos sea correcto.
2. Parsea la cadena de entrada y comprueba que es reconocida por el lenguaje **L**.
3. Divide la frase parseada en una lista de difonos.
4. Genera y lanza el script de Praat que concatena los difonos (grabados) de la lista de difonos.
5. Si la cadena de entrada es interrogativa, modifica la prosodia del archivo '.wav' generado en el punto 3.

Paso 1. Comprobamos el número de argumentos recibidos.

```
# comprobamos que el número de argumentos es correcto
if len(sys.argv) != 3:
    raise Exception(len(sys.argv)-1, "Error en el número de argumentos, deben ser dos")
```

Paso 2. Parser (front-end).

2.1. Comprueba que la cadena de entrada no esté vacía.

2.2. Observa si la cadena de entrada es interrogativa (termina en '?'). En caso afirmativo, activa la bandera 'esInterrogativa' y recorta el símbolo '?' de la cadena de entrada.

2.3. Bucle que lee, de la cadena de entrada, las sílabas reconocidas por el lenguaje. Si alguna no es reconocida lanza una excepción.

```
# sílabas que forman el lenguaje
v = re.compile(r'^(e)')
V = re.compile(r'^(E)')
cv = re.compile(r'^(fe|ke|me|re|se|te)')
cV = re.compile(r'^(fE|kE|mE|rE|sE|tE)')
ccv = re.compile(r'^(kre|tre|fre)')
ccV = re.compile(r'^(krE|trE|frE)')
vc = re.compile(r'^(es)')
Vc = re.compile(r'^(Es)')
cvc = re.compile(r'^(fes|kes|mes|res|ses|tes)')
cVc = re.compile(r'^(fEs|kEs|mEs|rEs|sEs|tEs)')
ccvc = re.compile(r'^(kres|tres|fres)')
ccVc = re.compile(r'^(krEs|trEs|frEs)')
# lenguaje completo, de mayor a menor tamaño de sílaba
lenguaje = [ccvc, ccVc, cvc, cVc, vc, Vc, ccv, ccV, cv, cV, v, V]
```

2.3.1. Lee las candidatas a sílabas reconocidas por el lenguaje.

2.3.2. Si no existe ninguna sílaba candidata, lanza una excepción de lenguaje no reconocido.

2.3.3. Selecciona como primera sílaba reconocida de la cadena a la primera de las candidatas. Por el orden que le hemos dado al lenguaje (mayor a menor) la primera será la sílaba más larga de las posibles.

2.3.4. Controla las restricciones del lenguaje **L** por la que una frase no puede empezar por 'r' y una 'r' no puede suceder a una 's'.

2.3.5. Elimina la sílaba reconocida de la cadena de entrada.

2.3.6. Si la cadena de entrada ha quedado vacía el lenguaje se ha reconocido (break).

2.4. Devuelve la frase sin '?', la lista de sílabas reconocidas (que no necesitamos) y la bandera 'esInterrogativa'.

Paso 3. Generamos una lista de difonos a partir de la frase parseada (front-end).

La lista se genera con una sencilla función que toma los 'char', de la frase parseada, dos a dos.

```
# almaceno medio difono inicial (principio de la frase)
listadifonos.append(frase[0])
# por cada char leído del str 'frase' desde el segundo al último
for i in list(range(1, len(frase))):
    # almaceno los difonos entre char de la frase
    listadifonos.append(frase[i-1]+frase[i])
# almaceno el medio difono final (final de la frase)
listadifonos.append(frase[-1])
```

Paso 4. Script de Praat que concatena los difonos de la lista de difonos (back-end).

4.1. Una función escribe en el fichero “Script_concatena” el script de Praat que concatena los difonos de la lista de difonos.

4.1.1. Lee por cada difono de la lista de difonos, su homólogo grabado en el almacén de difonos. Crea un objeto Sound con él y lo renombra para poder seleccionarlo varias veces (si se da el caso) en la concatenación.

```
for difono in listadifono:
    # W10 no diferencia 'e' de 'E' por lo que el acento, que en la cadena se lee 'E', se almacenó como 'E1'
    difono = difono.replace("E", "E1")
    # leo el difono almacenado en la carpeta lista_difonos que corresponde al difono de la cadena de entrada
    f.write('Read from file: "lista_difonos/' + difono + '.wav"' + os.linesep)
    # renombro los ficheros para poder concatenarlos si aparecen repetidos
    f.write('Rename: "d' + str(i) + '"' + os.linesep)
    i = i + 1
```

```
# selecciono los objetos de sonido que vamos a concatenar
for i in list(range(len(listadifono))):
    if i == 0:
        f.write('selectObject: "Sound d' + str(i) + '"' + os.linesep)
    else:
        f.write('plusObject: "Sound d' + str(i) + '"' + os.linesep)
```

4.1.2. Concatena, con superposición en la unión, los objetos Sound seleccionados en un nuevo objeto Sound.

```
# concatenamos los objetos de sonido en un nuevo objeto de sonido llamado 'chain'
f.write('Concatenate with overlap... 0.01' + os.linesep)
```

4.1.3. Convierte la salida a mono, 16kHz y 16 bits.

```
# convertimos la salida a mono, 16kHz, 16 bits.
f.write('Convert to mono' + os.linesep)
f.write('Resample: 16000, 16' + os.linesep)
```

4.1.4. Guarda el objeto Sound, resultado de la concatenación, en un archivo ‘.wav’ con el nombre asignado en el segundo argumento de entrada.

```
# lo almacenamos en un fichero wav
f.write('Save as WAV file: "' + nombrearchivo + '"' + os.linesep)
```

4.2. Lanzamos el script con un subproceso y esperamos a que termine su ejecución.

```
# lanzamos el script de Praat que concatena los difonos y esperamos a que termine su ejecución
proceso_concatena = run('Praat archivos/Script_concatena')
```

Paso 5. Si la cadena de entrada es interrogativa, cambiamos la prosodia del archivo '.wav' resultado de la concatenación de difonos (back-end).

Para simular una frase interrogativa, aumentamos la frecuencia fundamental (el tono) de la última sílaba de la frase (los 0.1s finales). Para aumentar el tono optamos por aumentar el pitch.

5.1. Utilizamos el script de Praat "extraer-pitch-track.praat" para extraer el pitchtrack del archivo '.wav'.
5.2. Modificamos este script con una función de Python.
5.3. Reemplazamos el pitchtrack original por el modificado con el script de Praat "reemplazar-pitch-track.praat".

El primer script genera un fichero '.PitchTier' que almacena un conjunto de puntos (muestras del fichero '.wav'). Tomando de éstas el tiempo de ocurrencia dentro del fichero '.wav' y su valor pitch.

```
File type = "ooTextFile"
Object class = "PitchTier"

xmin = 0
xmax = 1.25625
points: size = 53
points [1]:
    number = 0.035625000000000066
    value = 107.85285819095085
points [2]:
    number = 0.050625000000000066
    value = 107.12441118373346
```

…

```
points [50]:
    number = 1.175625
    value = 109.48665550267616
points [51]:
    number = 1.190625
    value = 111.23033384645886
points [52]:
    number = 1.205625
    value = 114.36928742282629
points [53]:
    number = 1.220625
    value = 114.82019091668259
```

Los últimos diez puntos corresponden aproximadamente a los últimos 0.1s de la frase (la última sílaba de ésta).

Entonces, nos colocamos en esa posición del fichero '.PitchTier' e incrementamos el valor del pitch de los últimos diez puntos linealmente.

```
for i in list(range(1, npuntos+1)):
    for j in list(range(2)):  # salto a points y de points a number
        linea = fr.readline()
        fw.write(linea)
    linea = fr.readline()  # leo value
    ind = linea.index('=')
    value = float(linea[ind+2:])
    value = value + inc * i
    fw.write('\tvalue = ' + str(value) + os.linesep)
```

Ya podemos reemplazar el pitchtrack con nuestro pitchtrack modificado.

Conclusión

La realización de esta práctica me ha hecho comprender la importancia de tomar al difono como unidad de sonido en la síntesis del habla.

El difono representa el sonido comprendido desde la mitad de un fonema hasta la mitad del fonema siguiente.

Después de probar la síntesis (concatenación de unidades) con diferentes unidades de sonido (fonema, sílaba y difono), el difono se reveló como la única unidad capaz de dotar de transición al sonido entre fonemas.