

MAT289 – Laboratorio de Modelación II

Kit demostrativo de efectos de audio

Franco Tapia Rojas
franco.tapia@alumnos.usm.cl

Profesional encargado: Matías Zañartu Salas
Encargado de Difusión del Departamento de Electrónica UTFSM
Profesor de la asignatura: Pablo Aguirre Olea



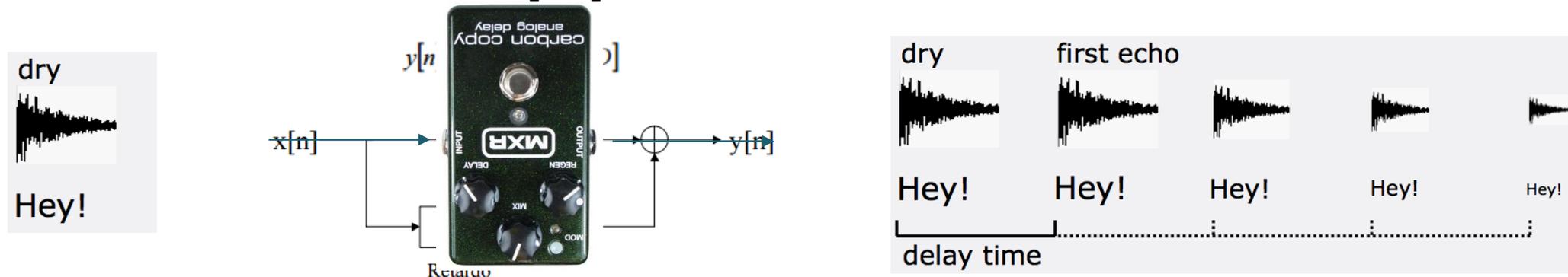
Departamento de Matemática UTFSM

29 de Noviembre de 2016

Descripción del problema y motivación

- Por medio de Procesamiento de Señales es posible modificar una señal de entrada para obtener una salida diferente.
- En música, se les llama *efectos* a estos procesadores que realizan cambios sobre una señal de audio.
- Los *efectos* existentes trabajan como **cajas negras**: No es posible ver las operaciones matemáticas que permiten su funcionamiento.

Producto existente:



Objetivos del proyecto

- Implementar algunos efectos en términos matemáticos en algún programa, como por ejemplo MATLAB.
- Elaborar un Kit demostrativo de Procesamiento de Señales de audio para las actividades de difusión de la UTFSM.
- Mostrar cómo la matemática se vincula con los efectos de sonido usados en música.
- Efectos a implementar: Overdrive.
Distorsión.
Delay.
Reverb.
Phaser.
Flanger.





Etapas del proyecto

- I. Investigación de Procesamiento de Señales (DSP).
- II. Desarrollo, testeo y mejoramiento de los Efectos.
- III. Elaboración de una interfaz visual.
- IV. Creación de gráficos, esquemas, diagramas y/o fórmulas explicativas.
- V. Presentación del Kit de efectos ante el mandante.

```
Coupling01.m  x  Delay01.m  x  Dist01.m  x  Flang01.m  x  Over01.m  x  Phase02.m  x  Rev01.m  x  Vit
26 - G=2.75; %1.25
27
28 - z = zeros(length(rv),1);
29 - for i = 1:length(rv)
30 -     if i<D+1
31 -         z(i,1) = rv(i,1);
32 -     elseif i<2*D+1
33 -         z(i,1) = G*(A*rv(i,1) + L*revlevel*z(i-D,1));
34 -     else
35 -         z(i,1) = G*(A*rv(i,1) + L*revlevel*z(i-D,1) + L*revlevel*z(i-2*D,1));
36 -     end
37 - end
38
39 % Normalizar
40 %maxz = max(abs(z));
41 %z = rv/maxz;
42
43 % (Optional) Crear archivo de audio con grabación con efecto
44 wavwrite(z,fs,16,'Coupling(Despues).wav')
45
46 % Reproducción de la grabación.
47 sound(z,fs)
48
49 % Gráfico.
50 t=linspace(0,stopTime,stopTime*fs);
51 plot(t,z,'g')
52 figure
53 %hold on
54 plot(t,rv,'r')
55 title('Señal original en Azul y señal con Coupling en Verde')
```

Presentación del producto

1. Overdrive:

- Sistema no lineal.
- Modifica la forma de onda del sonido original a partir de un determinado nivel de entrada.
- Similar a los efectos *Distorsión* y *Fuzz*, pero de manera menos pronunciada que ellos.
- Utiliza sobre todo ganancias con soft clipping simétrico.
- Efecto cálido y suave, dado que introduce componentes armónicos en el espectro.



Grabación de guitarra sin efecto.



Señal procesada con Overdrive.

Referencia: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_\(Modulo_5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_5).pdf)

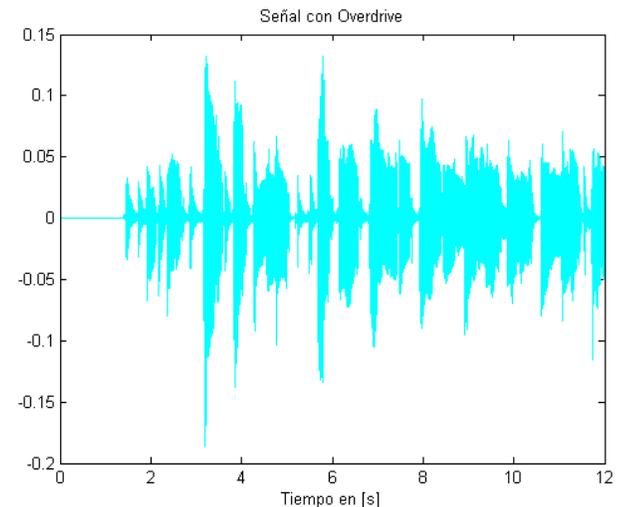
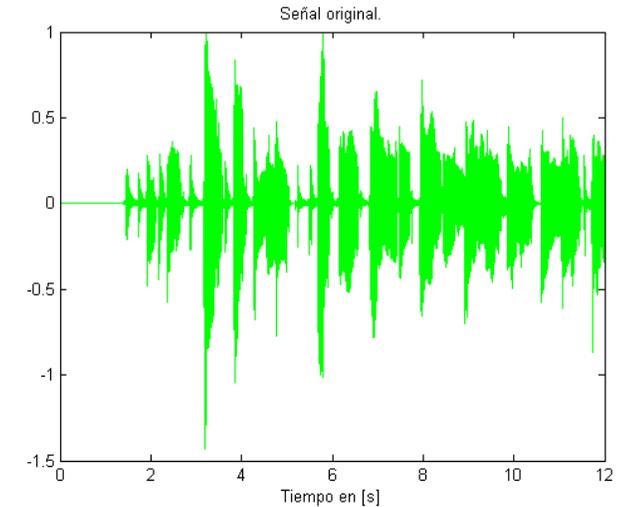
Presentación del producto

1. Overdrive:

- Grabamos por muestras, a una frecuencia de $f_s = 22050$ muestras/segundo.
- Sea $x[n]$ la entrada para la muestra n .
- Sea $y[n]$ la salida para la muestra n .
- Sea $|x[n]|$ la magnitud de la entrada $x[n]$.
- Sea C la variable de compresión.
(C más pequeño \rightarrow Más *se estira* la señal \rightarrow Menos rango dinámico).
- Sea D el *Drive*. D más bajo \rightarrow Overdrive más notorio.
- Sea L el Nivel. Controla *qué tan fuerte sonará* la salida.

$$y[n] = \begin{cases} L \cdot x[n] & , |x[n]| < D \\ L \cdot (C \cdot x[n] + \text{sign}(x[n]) \cdot (1 - C) \cdot D) & , |x[n]| \geq D \end{cases}$$

Referencia: <http://www2.elo.utfsm.cl/~elo314/laboratorios/Lab2p1.pdf>



Presentación del producto

2. Distorsión:

- Sistema no lineal.
- Semejante al Overdrive, pero más notorio.
- La función utilizada para calcular la ganancia determina el efecto producido en el audio.
- Teóricamente, se puede utilizar cualquier función para crear efectos, pero no todos sonarán “deseables” a nuestros oídos.



Grabación de guitarra sin efecto.



Señal procesada con Distorsión.

Referencia: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_\(Modulo_5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_5).pdf)

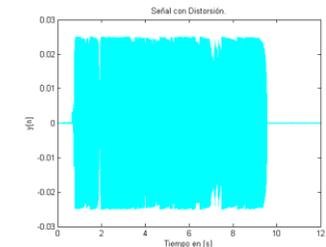
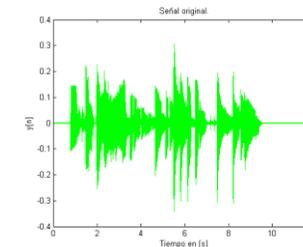
Presentación del producto

2. Distorsión:

- $x[n]$: entrada para la muestra n .
- $y[n]$: salida para la muestra n .
- $|x[n]|$: magnitud de la entrada $x[n]$.
- D : distorsión, $-1 < D < 1$.
- u : umbral de corte de ruidos.
- a : atenuación de ruidos.
- $k := \frac{2,5 \cdot D}{0,997 - D}$: constante de este algoritmo.
- c : *offset*. Agrega armónicos.
- L : nivel de salida.

$$y_1[n] = \begin{cases} a \cdot x[n] & , \quad |x[n]| < u \\ 1,6 \cdot x[n] & , \quad u \leq |x[n]| < 0,85 \\ 1,6 \cdot (x[n] + \text{sign}(x[n])) \cdot D \cdot (1 - D) & , \quad 0,85 \leq |x[n]| < u \end{cases}$$

$$y_j[n] = L \left((1 - k) \cdot \frac{y_{j-1}[n]}{1 + k \cdot |y_{j-1}[n]|} + c \right), \quad j = 2, 3, 4. \quad y_4[n] = y[n]$$



Referencias:

<https://es.mathworks.com/matlabcentral/fileexchange/6639-guitar-distortion-effect/content/gdist.m>
https://www.dsprelated.com/freebooks/pasp/Nonlinear_Distortion.html

Presentación del producto

3. Delay (retraso):

- Efecto muy simple, a partir del cual se pueden elaborar otros efectos.
- Se añade al audio original múltiples repeticiones retardadas y atenuadas.
- Se imita el eco original que se genera en la naturaleza.
- El tiempo del retardo debe ser superior a 50 [ms].
- Las repeticiones se van atenuando hasta que se vuelven imperceptibles.



Grabación de guitarra sin efecto.



Señal procesada con Delay.

Referencia: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_\(Modulo_5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_5).pdf)

Presentación del producto

3. Delay:

- $x[n]$: entrada para la muestra n .
- $y[n]$: salida para la muestra n .

- T : tiempo entre repeticiones (Delay time).
- R : número de repeticiones (Rate).
- L : volumen de repetición (Delay level).

Si $n < T$:

$$y[n] = x[n]$$

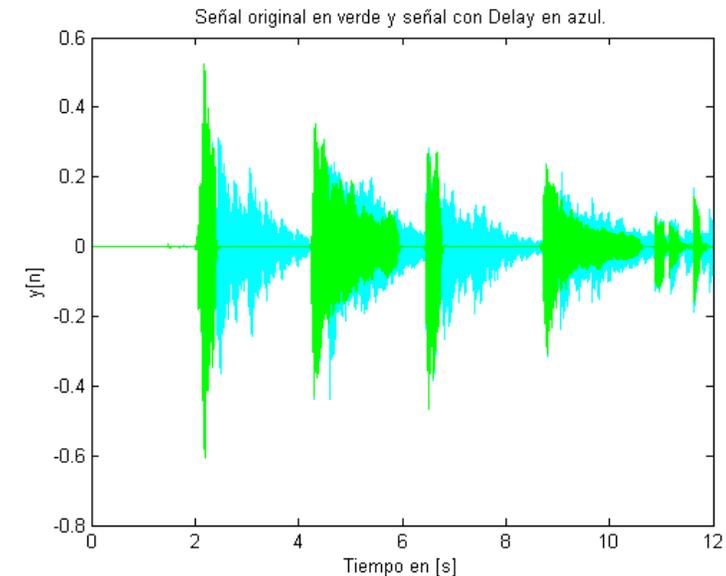
Si $n \geq T$:

$$y_1[n] = x[n]$$

$$y_j[n] = y_{j-1}[n] + L \cdot y_{j-1}[n - T], \quad j = 1, \dots, R$$

$$y_R[n] = y[n]$$

Referencia: <http://www2.elo.utfsm.cl/~elo314/laboratorios/Lab2p1.pdf>



Presentación del producto

4. Reverb:

- Si un número *infinito* de ecos es sumado, se obtiene una reverberación.
- La reverberación es el resultado de múltiples reflexiones de un sonido.
- Se produce naturalmente en cuevas, iglesias o salones.
- Ondas reflejadas llegan más tarde que la onda directa, y con pérdida de energía.
- El tiempo de retraso debe ser menor a 50 milisegundos.



Grabación de guitarra sin efecto.



Señal procesada con Reverb.

Referencias: <http://read.pudn.com/downloads157/sourcecode/math/696448/PRAC3/PRAC3.pdf>
[https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio (Modulo 5).pdf)

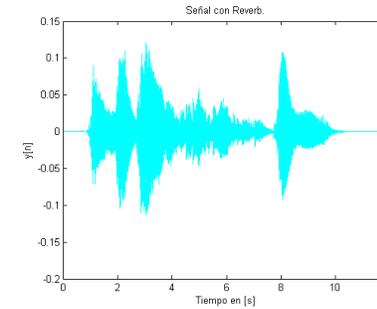
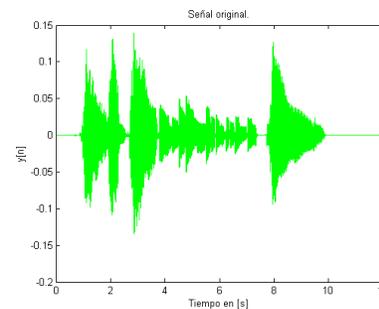
Presentación del producto

4. Reverb:

- $x[n]$: entrada para la muestra n .
- $y[n]$: salida para la muestra n .

- d : tiempo entre ecos (constante).
- $r := \frac{12,45 \cdot d}{f_s}$ constante del algoritmo.
- L : cantidad de reverberación.

$$y[n] = \begin{cases} x[n] & , n < d + 1 \\ 0,6 \cdot x[n] + L \cdot r \cdot y[n - d] & , d + 1 \leq n < 2 \cdot d + 1 \\ 0,75 \cdot x[n] + 1,25 \cdot L \cdot r \cdot y[n - d] + 0,93 \cdot L \cdot r \cdot y[n - 2 \cdot d] & , 2 \cdot d + 1 \leq n \end{cases}$$



Referencias: <http://read.pudn.com/downloads157/sourcecode/math/696448/PRAC3/PRAC3.pdf>
[https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio (Modulo 5).pdf)

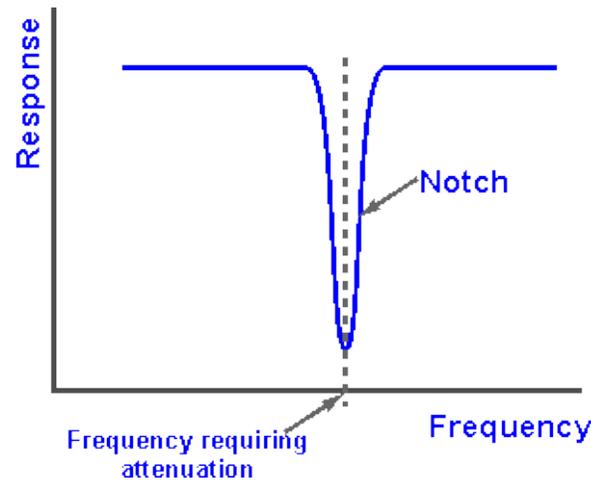
Presentación del producto

5. Phaser:

- Se utiliza una secuencia de filtros de corte, específicamente un número finito de cortes.
- La frecuencia varía con el tiempo por medio de un oscilador de baja frecuencia.
- Se combina la señal filtrada con el audio original.
- Provoca un sonido oscilante, sobre todo en frecuencias medias y altas.



Grabación de guitarra sin efecto.



Señal procesada con Phaser.

Referencias: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento%20de%20audio/Procesamiento%20de%20audio%20(Modulo%205).pdf)
http://users.cs.cf.ac.uk/Dave.Marshall/CM0268/PDF/10_CM0268_Audio_FX.pdf

Presentación del producto

5. Phaser:

- $x[n]$: entrada para la muestra n .
- $y[n]$: salida para la muestra n .
- R : velocidad de oscilación (Rate).

- f_0 : frecuencia mínima.
- f_1 : frecuencia máxima.
(Determinan la profundidad del efecto).

- $\delta := \frac{1300 \cdot R}{f_s}$ cambio en f . central/muestra ([Hz]).
- $d = \frac{Q}{2}$: damping factor (cte. del filtro notch).
- L : nivel de salida.

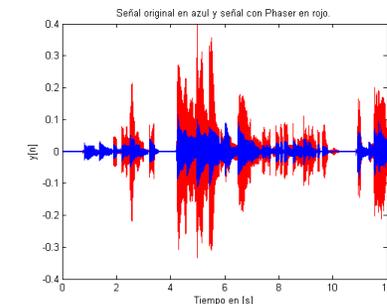
- 1) A partir de f_0 y f_1 se crea el filtro de corte (en nuestro caso con un ciclo *while*) y se ajusta al tamaño de la entrada.
- 2) Se crean vectores vacíos de salida. Se filtra dos veces la entrada para obtener una primera muestra.
- 3) Se usa un ciclo *for* a partir de la segunda muestra:

$$y_h[n] = x[n] - y_l[n - 1] - (\max(\text{abs}(x)) - Q_1) * y_b[n - 1]$$

$$y_b[n] = F_1 * y_h[n] + y_b[n - 1]$$

$$y[n] = F_1 * y_b[n] + y_l[n - 1]$$

- 4) Se normaliza la salida.



Referencias: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento%20de%20audio/Procesamiento%20de%20audio%20(Modulo%205).pdf)
http://users.cs.cf.ac.uk/Dave.Marshall/CM0268/PDF/10_CM0268_Audio_FX.pdf

Presentación del producto

6. Flanger:

- Se mezcla el audio original con una única repetición.
- El tiempo de retardo varía de manera periódica, por ejemplo en forma sinusoidal o triangular.
- En caso de obtener valores sinusoidales no enteros, se usan técnicas de redondeo.
- Provoca un sonido oscilante, sobre todo en frecuencias medias y altas.



Grabación de guitarra sin efecto.



Señal procesada con Flanger.

Referencias: <https://www.dsprelated.com/freebooks/pasp/Flanging.html>

[https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento%20de%20audio/Procesamiento%20de%20audio%20(Modulo%205).pdf)

Presentación del producto

6. Flanger:

- $x[n]$: entrada para la muestra n .
- $y[n]$: salida para la muestra n .
- D : profundidad ($0 < D < 1$).

- R : velocidad de la oscilación (Rate).
- m : longitud promedio de retraso (constante).
- E : excursión (constante).

Para cada valor de n , se calcula:

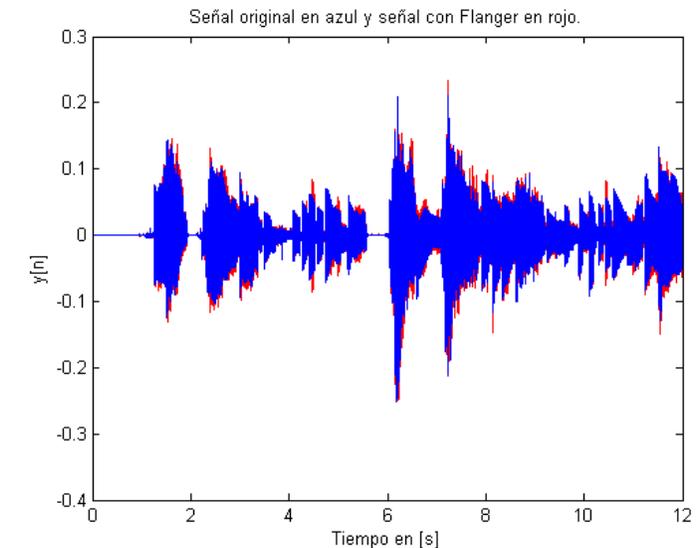
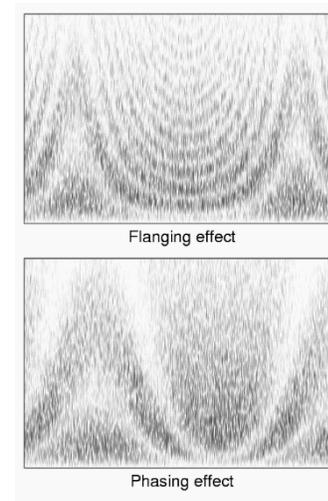
$$M = \text{ceil} \left(m \cdot \left(1 + E \cdot \sin \left(\frac{2 \cdot \pi \cdot R \cdot n}{f_s} \right) \right) \right)$$

Si $n \leq M$:

$$y[n] = x[n]$$

Si $n \geq M$:

$$y[n] = x[n] + D \cdot x[n - M]$$



Referencias: <https://www.dsprelated.com/freebooks/pasp/Flanging.html>

[https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento%20de%20audio/Procesamiento%20de%20audio%20(Modulo%205).pdf)

Presentación del producto

7. Efectos adicionales:

Modificando los efectos existentes, fue posible obtener otros efectos que fueron añadidos a la entrega:

- A partir del Phaser, se elabora un Wahwah:



- A partir del Flanger, se elabora un Vibrato:



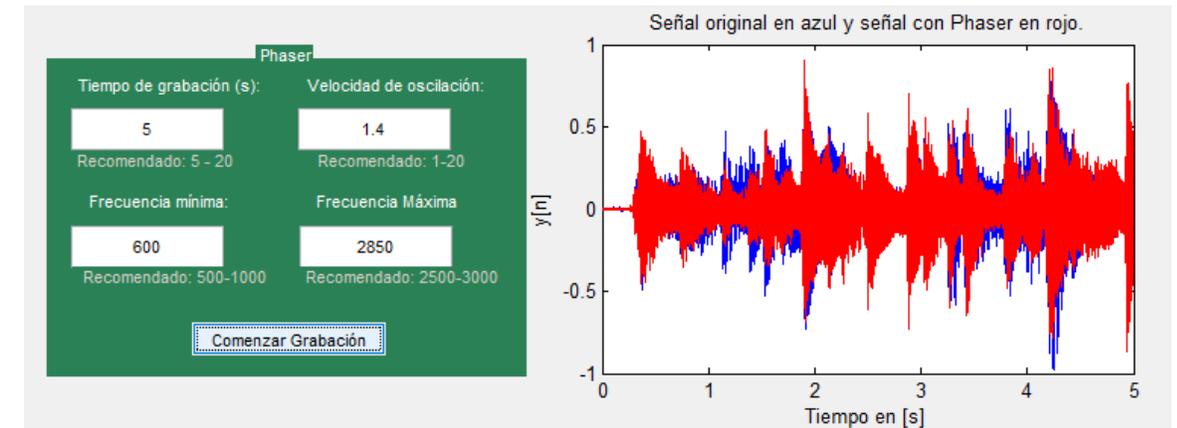
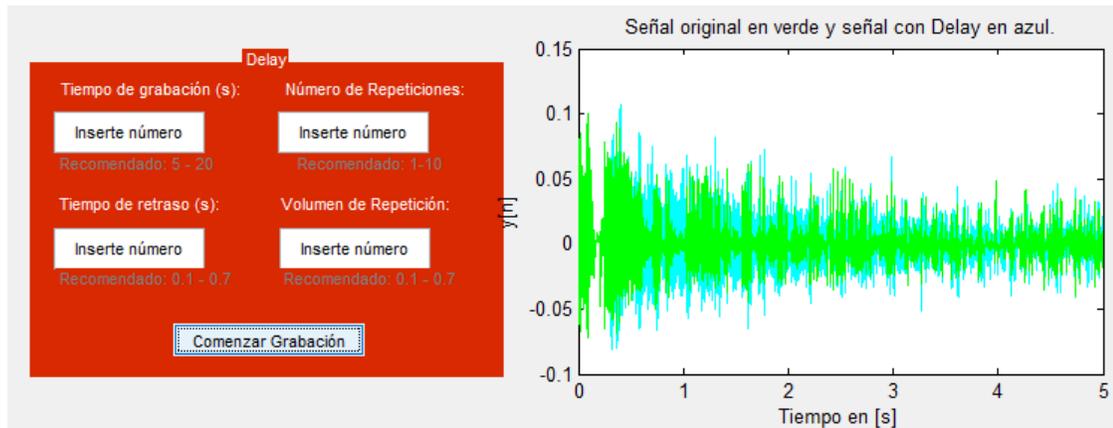
- A partir del Reverb, se elabora un efecto al que llamaremos Coupling:



Referencia: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_5).pdf)

Presentación del producto

Interfaz de efecto y parámetros



Balance final

1. Cantidad de horas de trabajo

- Horas totales de trabajo del proveedor:
57 horas aproximadamente. Presupuestado: 51 horas.
- Horas totales de trabajo del mandante:
4 horas aproximadamente. Presupuestado: 10 horas.
- Horas totales de trabajo del asesor:
2 horas aproximadamente. Presupuestado: 5 horas.
- Horas totales de trabajo de Rodrigo Manríquez (investigador):
1 hora aproximadamente. No presupuestado.

Total Efectivo: 64 horas aproximadamente.

Total Presupuestado: 68 horas.

Balance final

2. Comparación del producto final entregado y el planeado

- Todos los efectos funcionando correctamente + 3 efectos adicionales.
- Interfaz funcionando correctamente.
- Se cumplió con los plazos establecidos.
- Originalmente se pensaba implementar los códigos para procesar en tiempo real, pero esta entrega sólo es capaz de reproducir la señal procesada una vez que se ha terminado de grabar.

Razones:

- Se requiere librerías adicionales de MATLAB y/o compilador (ej: C, C++).
- Limitaciones de tiempo de proveedor y mandante.
- Esto último no evita que se cumpla el objetivo pedagógico fundamental del proyecto.

Conclusiones generales

1. Aporte e impacto

- Se cumple el objetivo pedagógico esperado.
- Todos los efectos cumplen correctamente su función y son modificables a gusto del usuario.

2. Posibles mejoras al producto entregado

- Mejorar la calidad del efecto Reverb.
- Hallar una forma de evitar el ruido presente en los efectos Flanger y Vibrato.

3. Potencial trabajo a futuro

- Implementación de todos los efectos para que procesen en tiempo real.
- Elaboración de una única interfaz donde sea posible activar o desactivar uno o más efectos a la vez y modificar sus parámetros.



Referencias bibliográficas

- <https://www.mathworks.com/help/audio/gs/real-time-audio-in-matlab-1.html?requestedDomain=www.mathworks.com>
- <https://www.mathworks.com/products/audio-system/>
- <https://es.mathworks.com/help/matlab/ref/audiorecorder.html>
- [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio \(Modulo 5\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento de audio/Procesamiento de audio (Modulo 5).pdf)
- <http://www2.elo.utfsm.cl/~elo314/laboratorios/Lab2p1.pdf>
- <https://es.mathworks.com/matlabcentral/fileexchange/6639-guitar-distortion-effect/content/gdist.m>
- https://www.dsprelated.com/freebooks/pasp/Nonlinear_Distortion.html
- <http://read.pudn.com/downloads157/sourcecode/math/696448/PRAC3/PRAC3.pdf>
- http://users.cs.cf.ac.uk/Dave.Marshall/CM0268/PDF/10_CM0268_Audio_FX.pdf
- <https://www.dsprelated.com/freebooks/pasp/Flanging.html>

MAT289 – Laboratorio de Modelación II

Kit demostrativo de efectos de audio

Franco Tapia Rojas
franco.tapia@alumnos.usm.cl

Gracias por su atención.



Departamento de Matemática UTFSM

29 de Noviembre de 2016