

Laboratorio #6 – Procesamiento de Audio

M.C. Fernando Hermosillo Reynoso

fhermosillo@up.edu.mx

Universidad Panamericana

Sesión #6

28 de Enero del 2020



● Repositorio GitHub del curso: [UP_DSP24](#)

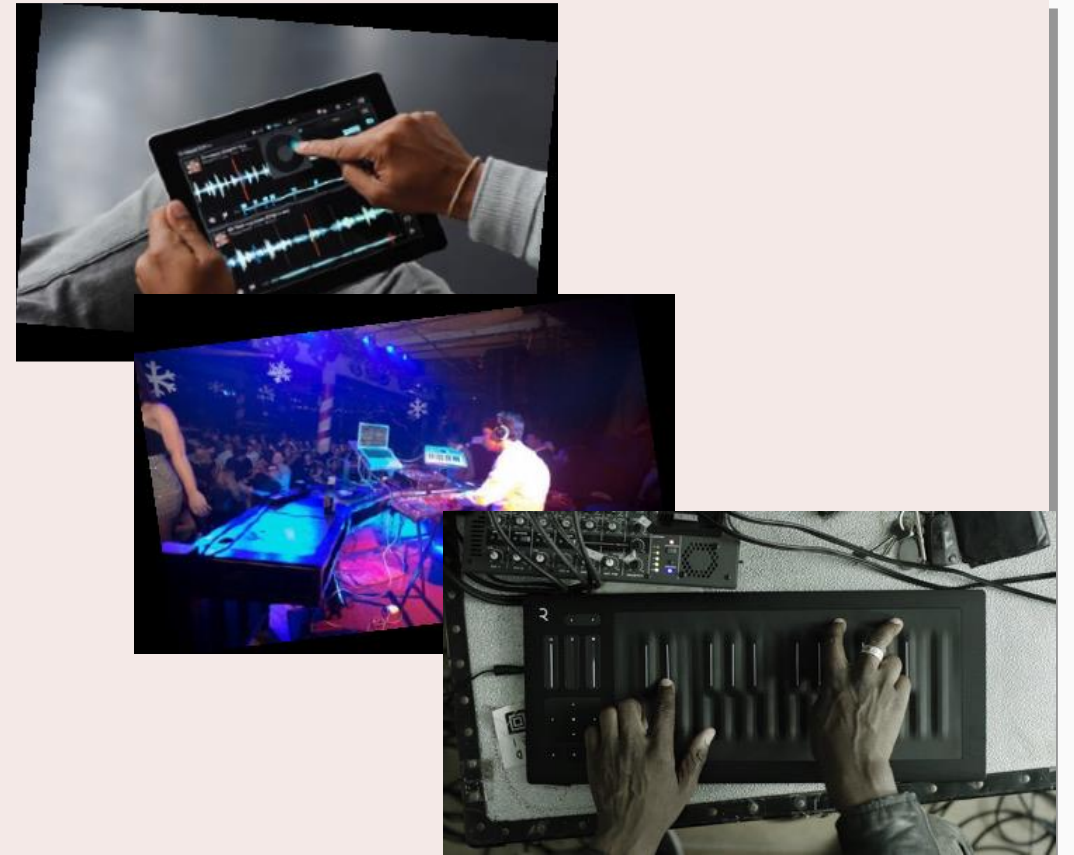
- Documentos
- Ejemplos
- Notas rápidas
- Laboratorios

Prelab

Procesamiento Digital de Audio

● Aplicaciones

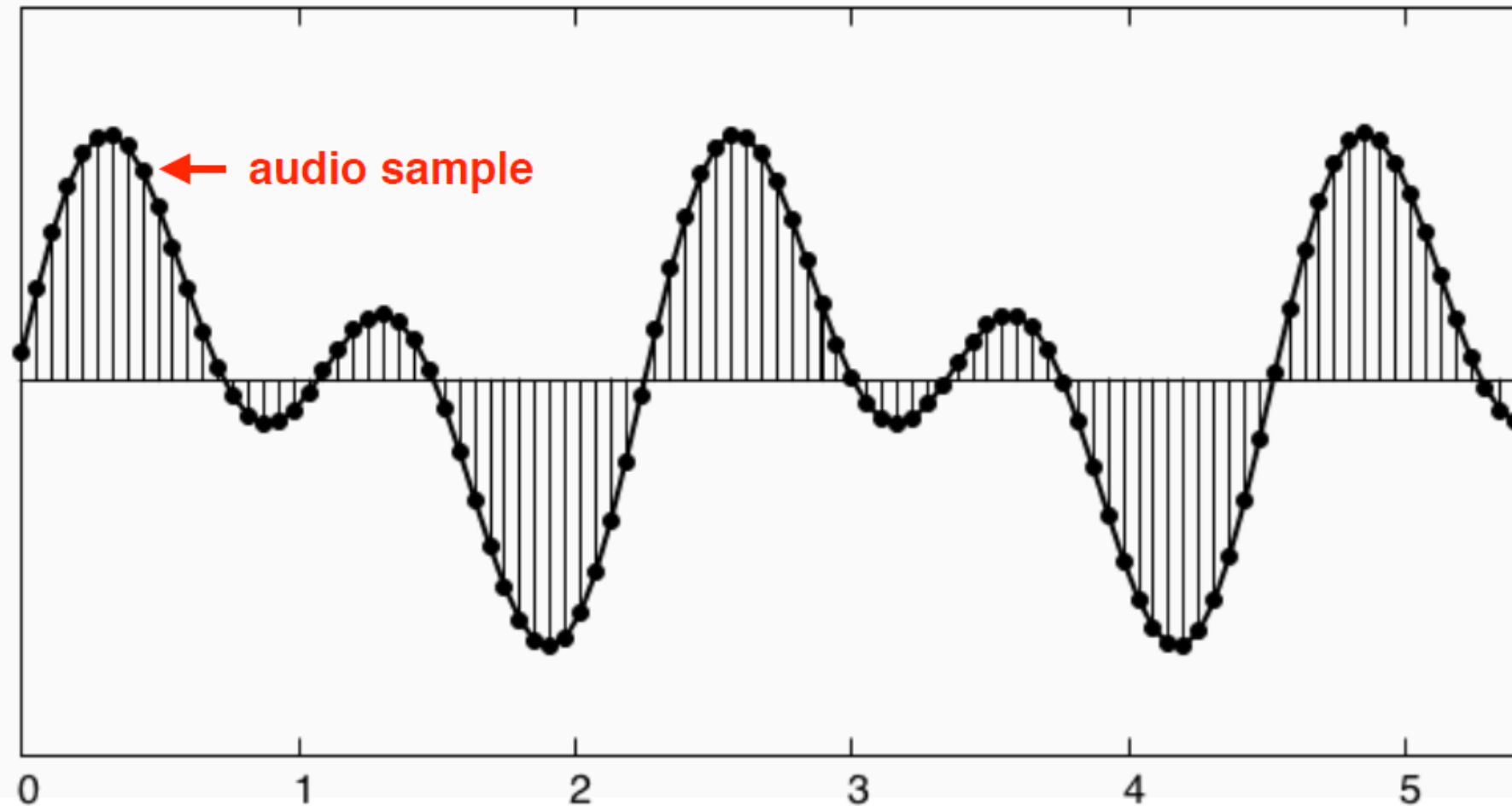
- Síntesis de Voz
- Compresión de datos
- Recuperación de información musical
- Cancelación de ruido
- Filtros
- Reconocimiento de voz
- ⋮



Representación de una Señal de Audio

sample rate

44.1 kHz, 48 kHz, 96 kHz, ...



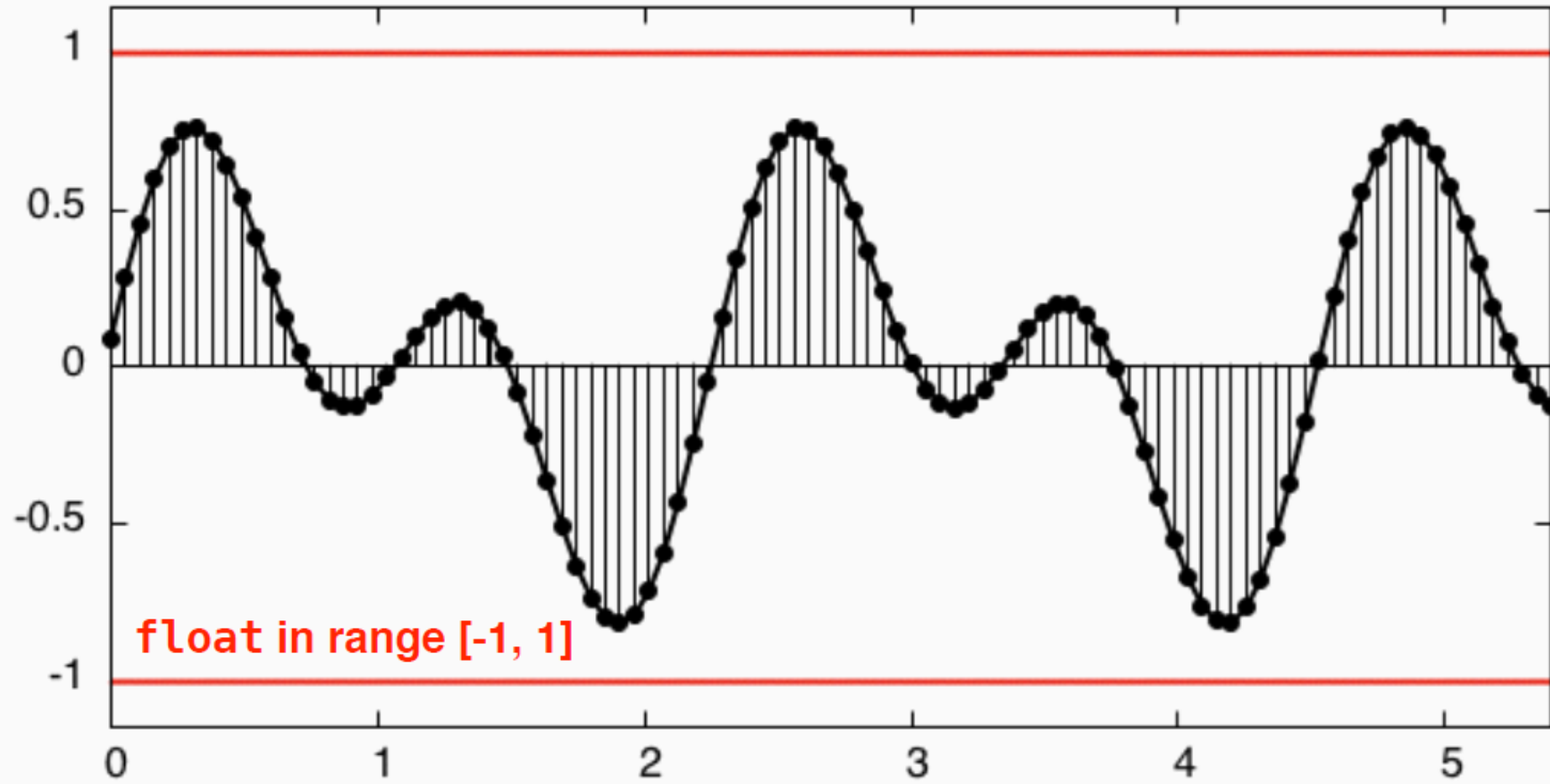
Representación de una Señal de Audio

El rango de audio es de
-1V a 1V

Posterior a este rango se
satura el ADC o el DAC!

➤ Distorsión

Hay que garantizar dicho
rango en ambos
convertidores!

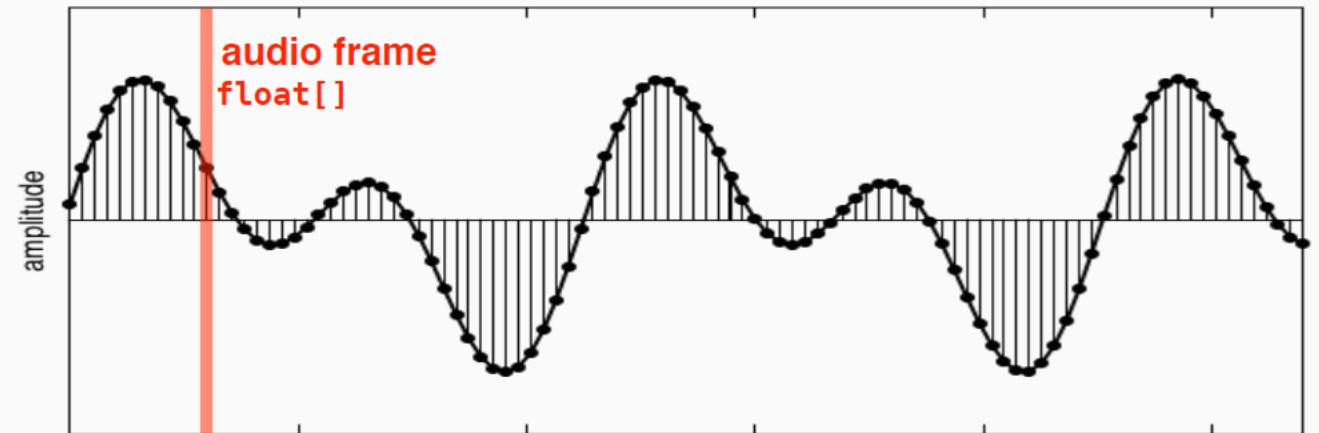


Representación de una Señal de Audio

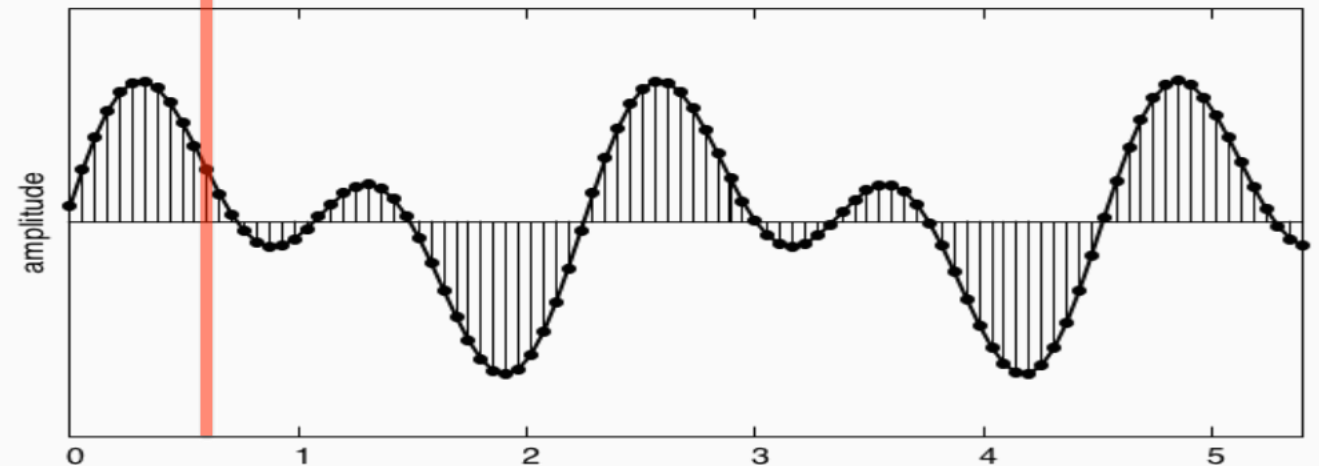
Usualmente la señal de audio es estéreo, es decir, contiene “dos muestras” por cada muestra:

- Canal izquierdo
- Canal derecho

channel 0



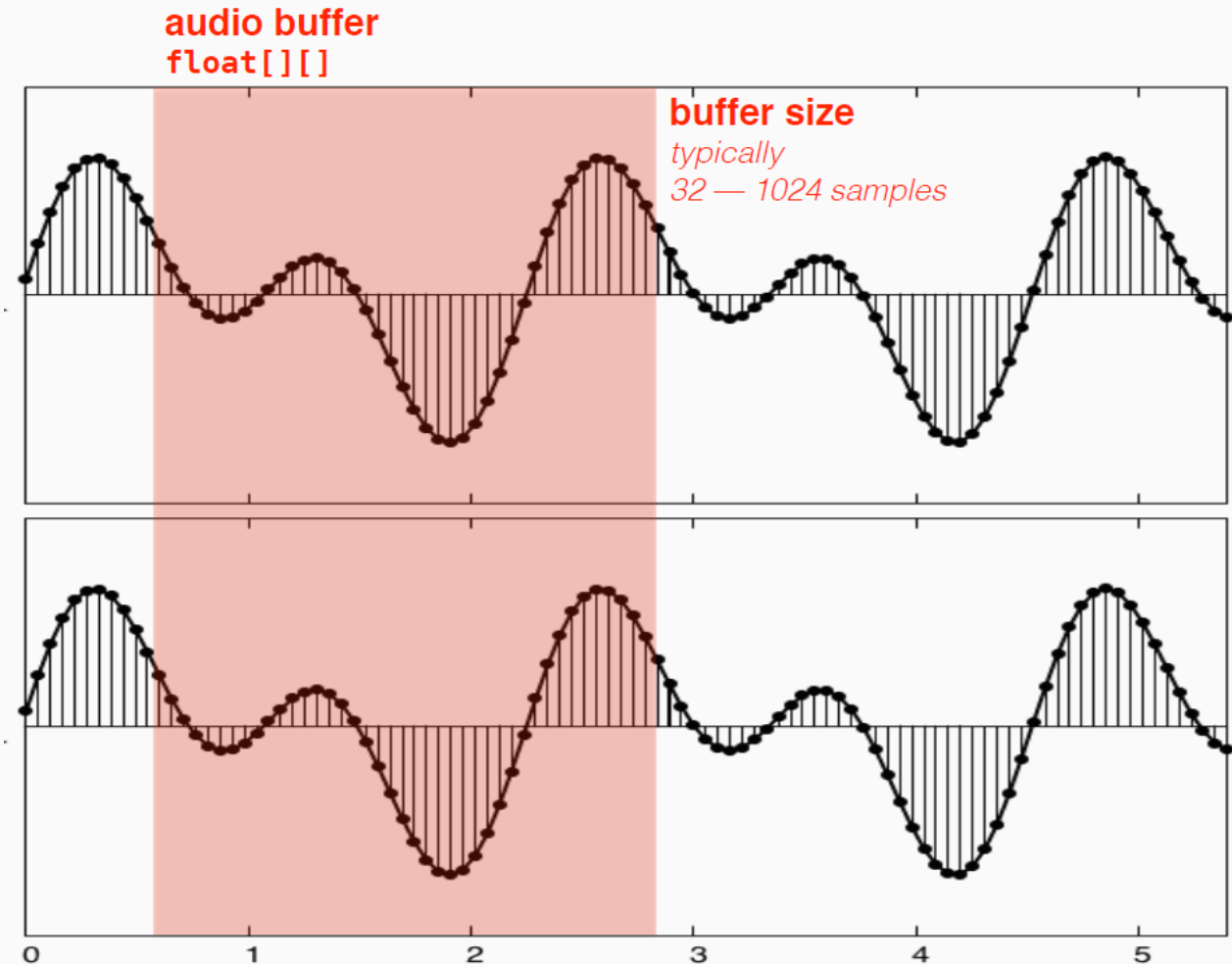
channel 1



Representación de una Señal de Audio

Las señales de audio se pueden procesar por bloques, es decir, tomando cierta cantidad de muestras denominado bloque, y posteriormente se procesa dicho bloque para enviarlo al DAC

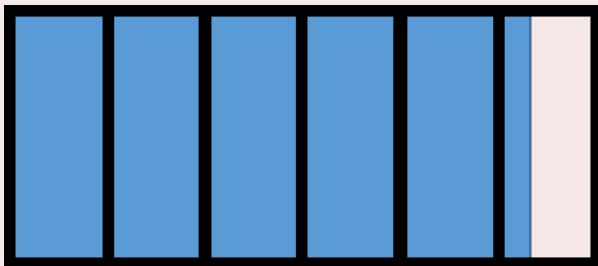
La programación de audio es
“hard real-time”



Conceptos Básicos

● Tipo de Procesamiento.

- **Basados en muestras.** Cada muestra de entrada se procesa en el tiempo “n”
- **Basados en bloques.** Datos se transfieren a un búfer de memoria, después se procesa el búfer.



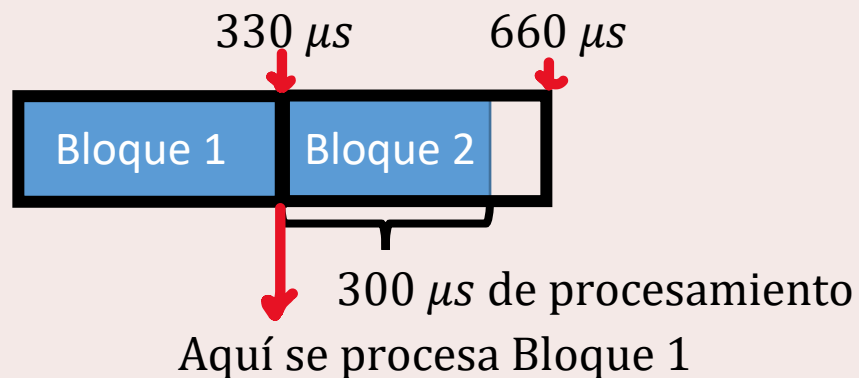
Buffer *de* 7
muestras

Conceptos Básicos

● Latencia τ .

- Tiempo total transcurrido que contempla almacenamiento y procesamiento

1. $\tau < 10 \text{ ms}$: Bueno para “en vivos”
2. $10 < \tau < 30 \text{ ms}$: Se percibe el retardo
3. $\tau > 30 \text{ ms}$: No apto para tiempo real



● Ejemplo de latencia (almacenamiento)

Buffer size	Buffer length	Buffer length
	@ sample rate 44.1 kHz	@ sample rate 96 kHz
32 samples	0.73 ms	0.33 ms
64 samples	1.45 ms	0.66 ms
128 samples	2.90 ms	1.33 ms
...
1024 samples	23.2 ms	10.7 ms

$T_s = 22.67 \mu\text{s}$
por muestra

Si se adquieren 32 muestras

$$\tau_{\text{storage}} = 32 \cdot T_s \approx 0.73 \text{ ms}$$

Conceptos Básicos

● Latencia τ .

- Tiempo total transcurrido que contempla almacenamiento y procesamiento
- $\tau < 10 \text{ ms}$: Bueno para “en vivos”
- $10 < \tau < 30 \text{ ms}$: Se nota el retardo
- $\tau > 30 \text{ ms}$: No apto para tiempo real

● Se tiene que garantizar que

1. Procesamiento termina antes del tiempo del próximo búfer
2. Terminará de procesar el búfer (no se estanca)
3. El búfer de salida contendrá datos de audio válidos
4. No habrá error y/o excepción

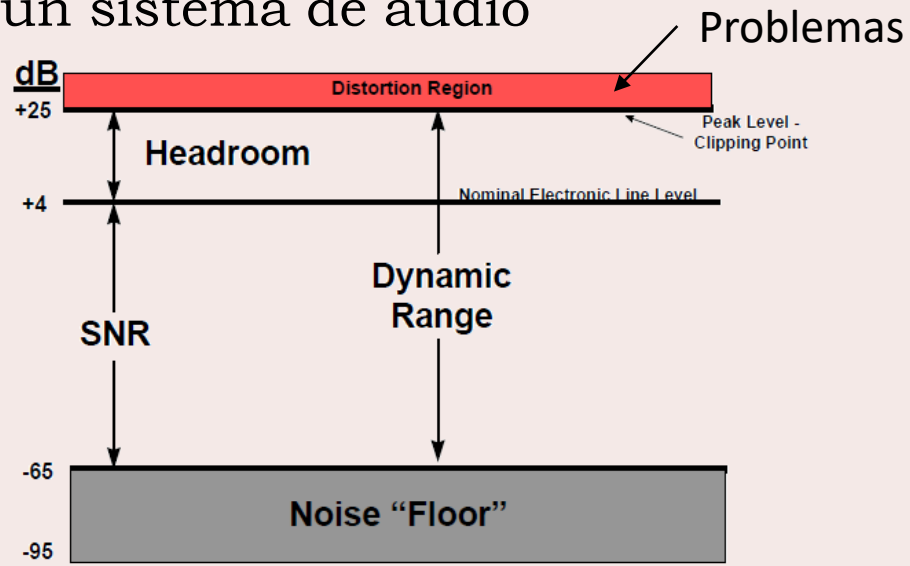
● De lo contrario

Audio Dropout

Conceptos Básicos

● Calidad del servicio.

- Calidad de señal aceptable dependiendo de la aplicación
- Principal consideración al diseñar un sistema de audio



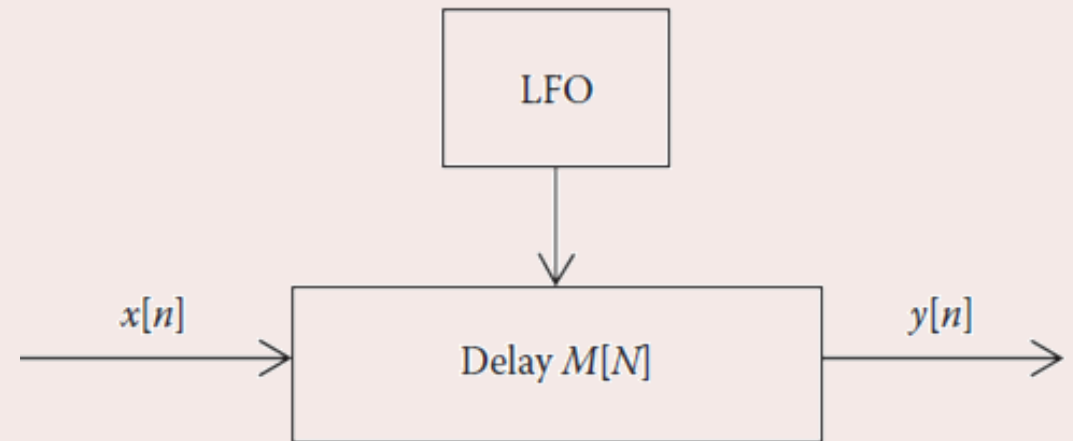
● Ejemplos.

<u>Audio Device/Application</u>	<u>Typical Signal Quality</u>
AM Radio	48 dB
Analog Broadcast TV	60 dB
FM Radio	70 dB
Analog Cassette Player	73 dB
Video Camcorder	75 dB
ADI SoundPort Codecs	80 dB
16 Bit Audio Converters	90 to 95 dB
Digital Broadcast TV	85 dB
Mini-Disk Player	90 dB
CD Player	92 to 96 dB
18-bit Audio Converters	104 db
Digital Audio Tape (DAT)	110 dB
20-bit Audio Converters	110 dB
24-bit Audio Converters	110 to 120 dB
Analog Microphone	120 dB

Efectos de Retardos de Línea

● Vibrato.

- Pequeña variación cuasi-periódica en el **pitch** de un tono
- Retardo variable de línea bajo el control de oscilador de baja frecuencia (LFO) $M(n)$
 - Función periódica $F < 20 \text{ Hz}$
 - $M_{avg} \geq W$: Causal, $M(n) \geq 0$
 - W : Ancho del LFO en muestras

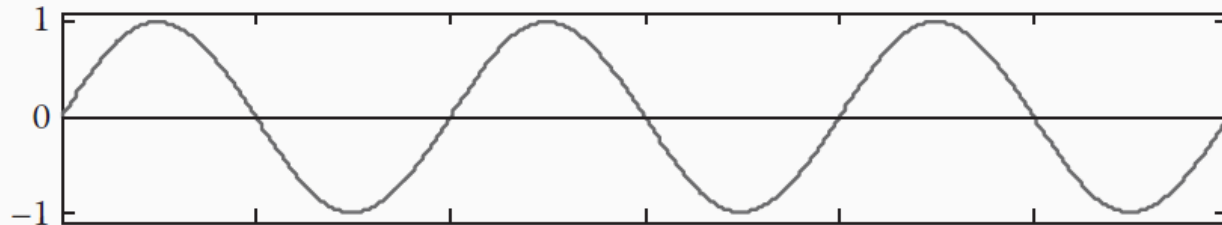


$$y(n) = x(n - M(n)) \quad (5)$$

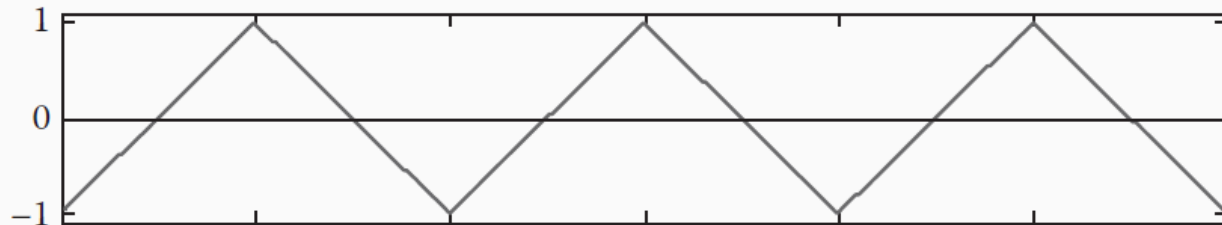
$$M(n) = \frac{W}{2} \left[\sin \left(2\pi \frac{F}{F_s} n \right) + 1 \right] \quad (6)$$

Osciladores de Baja Frecuencia

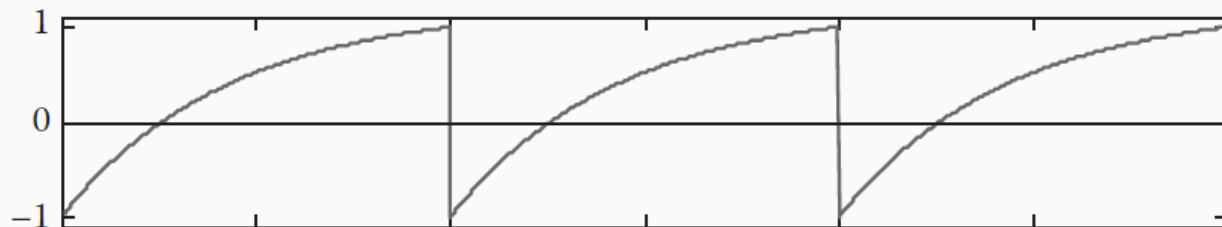
Sine LFO Waveform



Triangle LFO Waveform



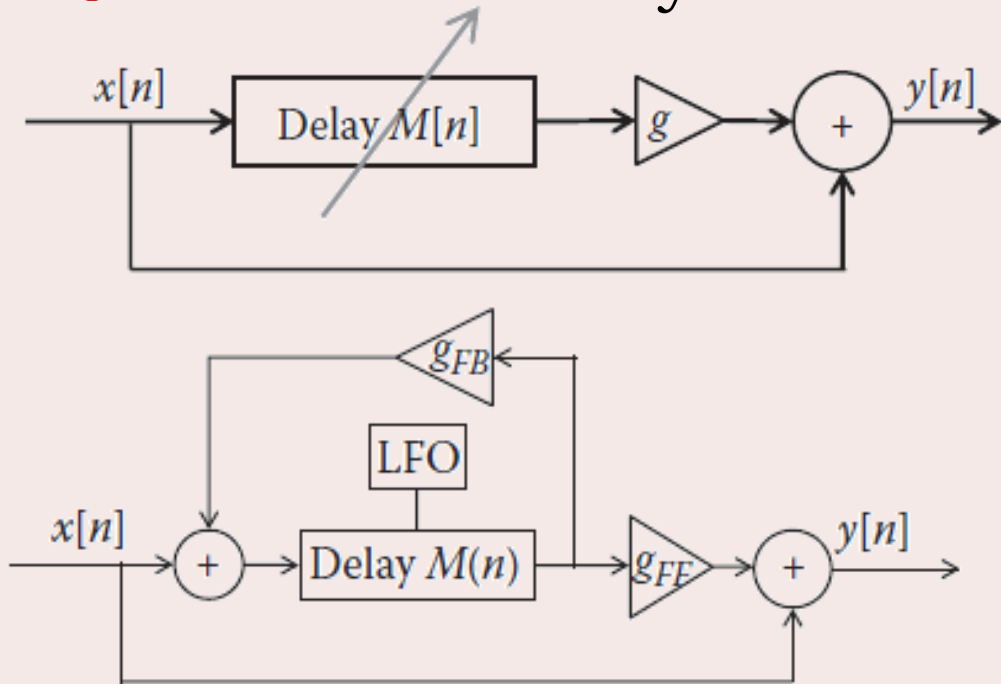
Exponential LFO Waveform



Efectos de Retardos de Línea

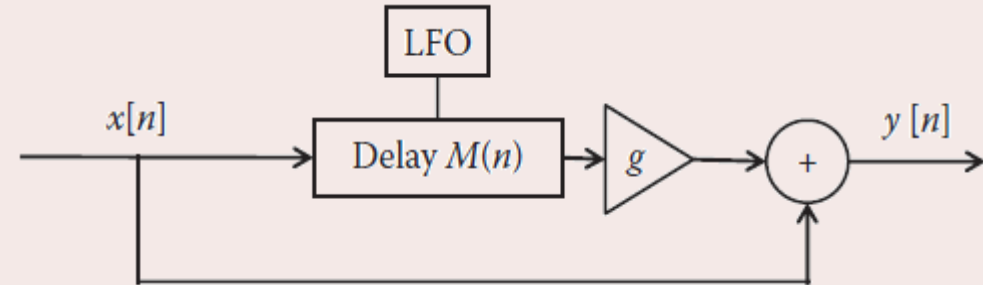
● Flanger.

- Retardos entre 1 y 10 ms



● Chorus.

- Retardos entre 20 y 30 ms
- $M(n): F \leq 3 \text{ Hz}, W \geq 5 \text{ ms}$



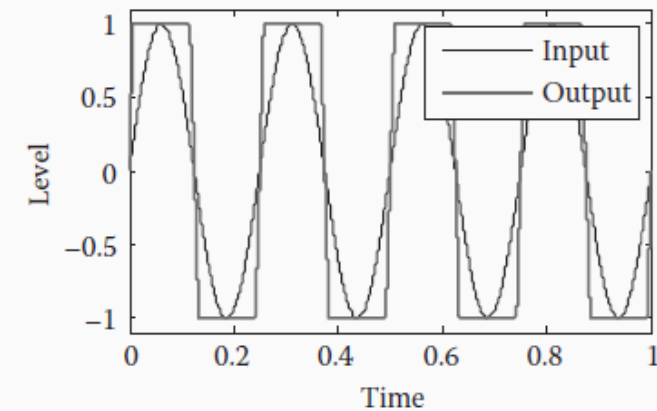
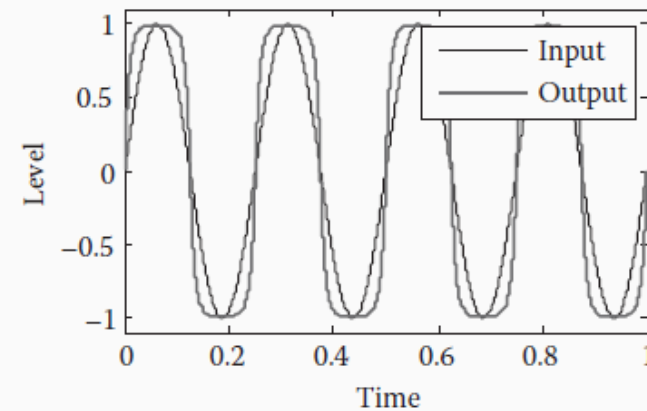
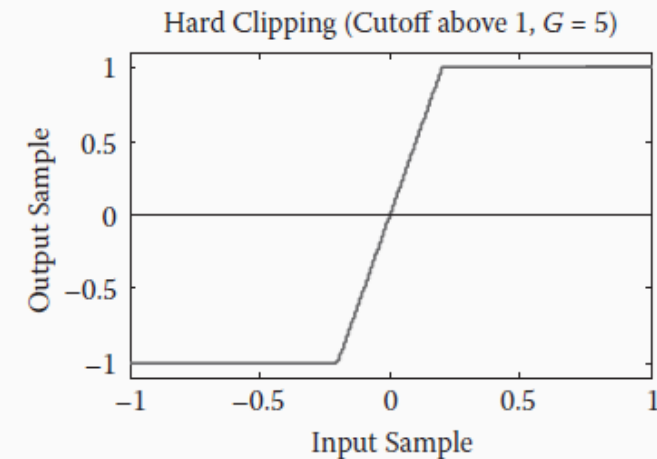
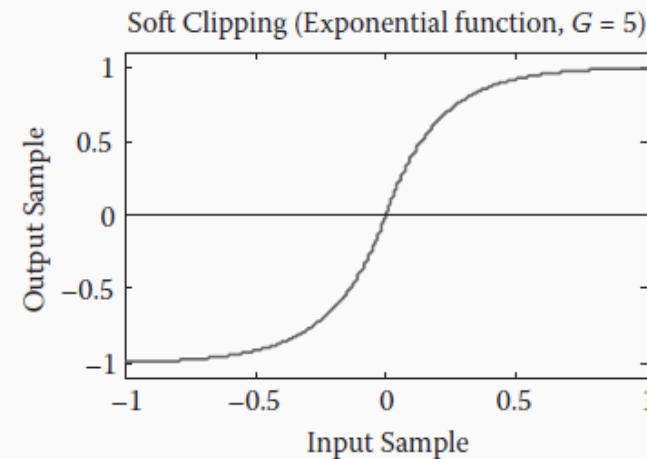
Distorsión

Hard-clipping se caracteriza por una transición abrupta entre las regiones recortadas y no recortadas de la forma de onda, lo que produce esquinas cuadradas en la salida.

$$f(x) = \begin{cases} -1 & Gx \leq -1 \\ G \cdot x & -1 < Gx < 1 \\ 1 & Gx \geq 1 \end{cases}$$

Soft-clipping se caracteriza por un enfoque suave del nivel de recorte, creando esquinas redondeadas en los picos de la forma de onda.

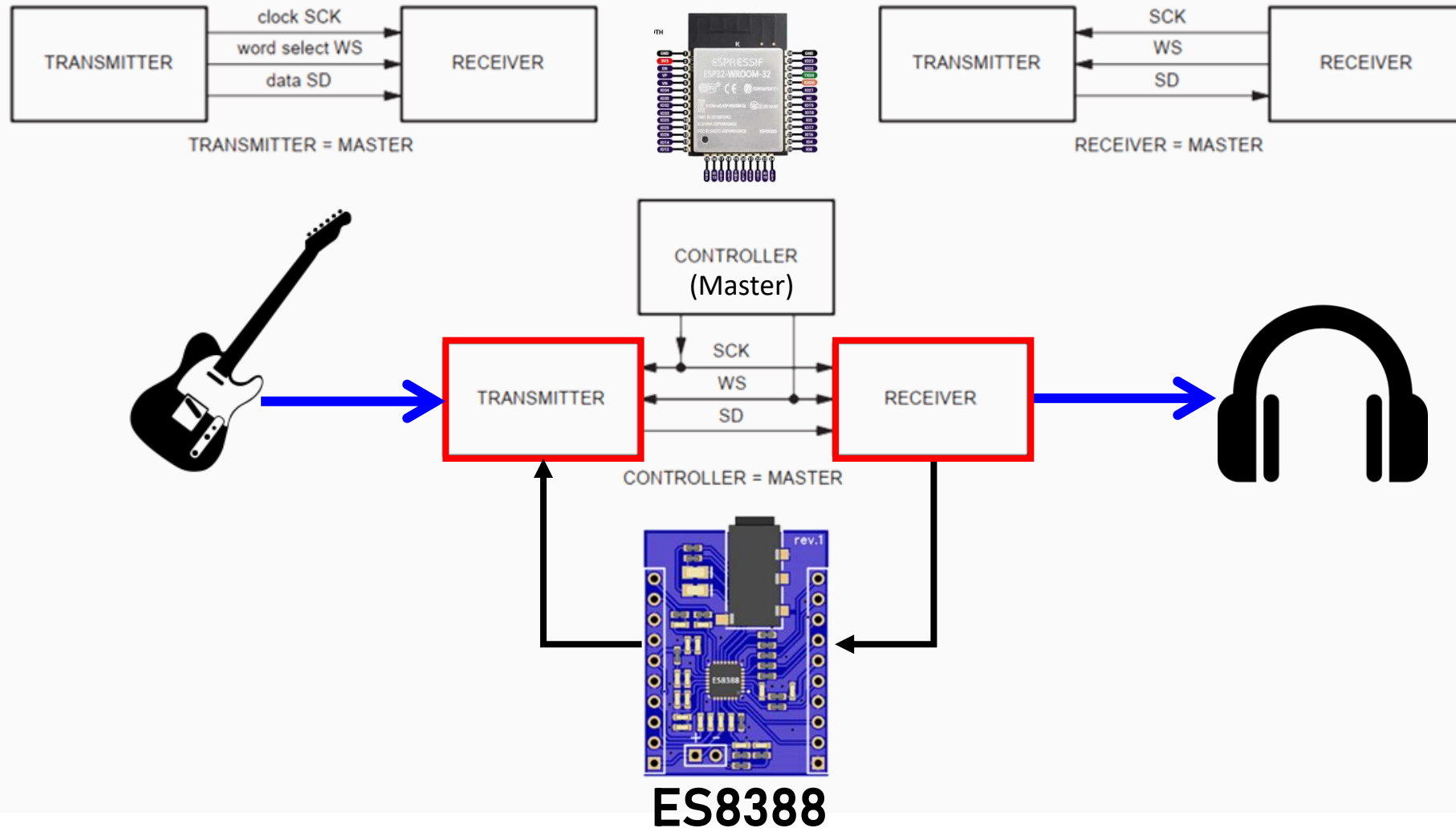
$$f(x) = \text{sign}(x) \cdot [1 - e^{-|Gx|}]$$



Laboratorio 6.

Procesamiento de Audio

Esquema General



AI-Thiker AudioKit V2.2: Descripción de I/O

GPIO	Señal	GPIO	Señal	GPIO	Señal
IO0	I2S-MCLK	IO36	KEY1	IO19	LED5
IO25	I2S-WS	IO13	KEY2	IO14	SCK
IO26	I2S-DOUT	IO19	KEY3	IO15	MOSI
IO27	I2S-BCK	IO23	KEY4	IO2	MISO
IO35	I2S-DIN	IO18	KEY5	IO13	CS
IO32	I2C0-SCL	IO5	KEY6	IO34	SD_INTR
IO33	I2C0-SDA	IO22	LED4		

Actividad #0: Configuración del Proyecto

- Crear nuevo proyecto en PlatformIO “pds24_lab_1”
- Añadir “AudioToolkit HAL”
- Configurar “platform.io”

- Reemplazar
PATH_TO_AUDIOKIT_LIB
Por ubicación de la biblioteca

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
lib_ldf_mode = deep+
lib_extra_dirs = PATH_TO_AUDIOKIT_LIB
build_flags = -DAUDIOKIT_BOARD=5
monitor_speed = 115200
```

Actividad #0: Configuración del Proyecto

1. Configure códec ES8388

1. Entrada y salida
2. ADC Línea 2
3. DAC Línea 1
4. Frec. Muestreo: 48KHz
5. Bits por muestra: 16
6. Tamaño buffer DMA: ?
7. Número buffers DMA: 4

- Establecer el tamaño del Buffer DMA de tal forma que se tenga una latencia de:

1. $\tau = 1ms$
2. $\tau = 30ms$

Actividad #1: Implementación de Algoritmos de Procesamiento de Audio

1. Diseñar e implementar tres algoritmos de procesamiento de audio en la tarjeta AudioKit

1. Ver los algoritmos en la siguiente diapositiva
2. Para algoritmos de retardo de línea, utilice el búfer circular visto en el laboratorio 3
3. Para algoritmos que empleen osciladores de baja frecuencia LFO senoidales, implementar los LFO con ecuaciones en diferencia

1. Puede escribir un programa por cada efecto o combinar los tres en uno solo, seleccionando el efecto a aplicar por medio de los botones

1. Si KEY1 esta presionado, aplicar efecto 1
2. Si KEY2 esta presionado, aplicar efecto 2
3. Si KEY3 esta presionado, aplicar efecto 3
4. Si ningún botón se presiona, no aplicar ningún efecto

Efectos Típicos

ECHO FILTER

$$y(n) = x(n) + \alpha x(n - D)$$

$$D \rightarrow t > 100ms$$

FEEDBACK ECHO FILTER

$$y(n) = x(n) + \alpha y(n - D)$$

$$D \rightarrow t > 100ms$$

$$|\alpha| < 1: \text{Stable}$$

$$|\alpha| = 1: \text{Critical-Stable}$$

$$|\alpha| > 1: \text{Unstable}$$

CHORUS FILTER

$$y(n) = x(n) + \sum_{k=1}^M \alpha_k x(n - d_k(n))$$

$$d_k(n): \text{Variable Delay } (10ms < t < 25ms)$$

$$d_k(n) = \frac{1}{2} (1 - \cos(\omega_0 n)) (D_2 - D_1) + D_1$$

$$F \leq 3Hz$$

FLANGER

$$y(n) = x(n) + \alpha x(n - d_k(n))$$

$$d_k(n): \text{Variable Delay } (1ms < t < 5ms)$$

$$d_k(n) = \frac{1}{2} (1 - \cos(\omega_0 n)) (D_2 - D_1) + D_1$$

$$1 \leq F \leq 5Hz$$

Efectos Típicos

TREMOLO FILTER

$$y(n) = (1 - \alpha)x(n) + \alpha x(n)m(n)$$

$m(n)$: Modulation signal

$$m(n) = \cos(\omega_0 n) \text{ o } \sin(\omega_0 n)$$

$$1 \leq F \leq 25\text{Hz}$$

DISORTION FILTER

$$y(n) = \text{atan}(G \cdot x(n))$$

$$G > 1$$

DISORTION FUZZ FILTER

$$y(n) = \text{sign}(x(n)) \cdot (1 - e^{-G|x(n)|})$$

$$G > 1$$

OVERDRIVEN FILTER

$$y(n) = \begin{cases} 2x(n), & |x(n)| < \frac{1}{\tau} \\ \frac{\text{sign}(x(n))(3 - (2 - 3|x(n)|)^2)}{3}, & |x(n)| < \frac{2}{\tau} \\ \text{sign}(x(n)), & \text{otherwise} \end{cases}$$

τ : Umbral de distorsión

Cuestionario

● Responda a las siguientes preguntas

1. De los tres algoritmos implementados
 1. Cuales son lineales?
 2. Cuales son no lineales?
 3. Cuales son invariantes en tiempo?
 4. Cuales son variantes en tiempo?

1. Si implemento un sistema LTI, determine su función de transferencia y calcule su respuesta al impulso