

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



## **PROYECTO FIN DE CARRERA**

**Laboratorio de Tecnologías de Audio**

**Sergio Carrero Jordán**

**Julio de 2014**



# **Laboratorio de Tecnologías de Audio**

**AUTOR: Sergio Carrero Jordán**

**TUTOR: Daniel Ramos Castro**



**Área de Tratamiento de Voz y Señales**

**Dpto. de Ingeniería Informática**

**Escuela Politécnica Superior**

**Universidad Autónoma de Madrid**

**Julio de 2014**



## ***Agradecimientos***

En primer lugar, me gustaría agradecer a mi tutor, Daniel Ramos por haberme concedido este proyecto. Nunca creí que podría unificar los conocimientos adquiridos a lo largo de la carrera con mi verdadera pasión, el mundo del audio. Además, no solo ha ejercido la labor de tutor de proyecto de fin de carrera, sino que para mí ha sido mi referente en todo momento, muchas gracias Daniel.

A mis padres, que siempre han estado y estarán a mi lado. Me han apoyado durante toda la carrera y me han dado todo sin pedir nada a cambio. Ni todas las palabras del mundo podrían expresar lo agradecido que estoy.

A mi hermano, he querido imitarle en todo porque para mí, es el mejor. Ya somos dos ingenieros en casa.

A mi novia, gracias por todo, has comprendido lo dura que ha sido la carrera y me has ayudado a ser más feliz cada día que paso a tu lado.

A mi mejor amigo Juan, hemos crecido juntos y compartido el difícil camino que conlleva sacarse una ingeniería.

Y como no, a mis amigos y compañeros de carrera Javier, Pablo, Jesús, Guille, Luis, David, Marcos y Agustín. Gracias por cruzaros en mi camino, hemos compartido una de las mejores etapas de nuestras vidas juntos y seguiremos compartiendo más.



## RESUMEN

En el presente proyecto, se ha desarrollado software específico para poder analizar y desarrollar un estudio en profundidad de los diferentes ámbitos de las Tecnologías del Audio.

Los principales bloques que se desarrollan a lo largo del proyecto son:

- Síntesis de audio con Matlab™: en el cual, se ha desarrollado un sintetizador sustractivo y posteriormente realizado un exhaustivo estudio y análisis del mismo.
- Síntesis granular: En este apartado se desarrollan los conceptos de dicha teoría para implementar en un software de producción musical o DAW (Digital Audio Workstation) un sintetizador programable. Utilizando el entorno de programación Earsketch™ (<http://earsketch.gatech.edu/>).
- Efectos: Se han desarrollado una serie de efectos en Matlab™ para analizar mediante pruebas analíticas y reales su comportamiento en profundidad y entender su funcionamiento. Se han implementado efectos con líneas de retardo (Eco FIR, Eco IIR, Vibrato y Chorus), efectos con filtros (*Wah-Wah*), efectos de procesado de dinámica (Compresor) y efectos de distorsión (Emulador del pedal *TubeScreamer* de la empresa *Ibanez*™).

La finalidad del proyecto es proporcionar material docente para la asignatura Tecnologías del Audio, de 4º de Grado ITST de la EPS-UAM.

## PALABRAS CLAVE

Audio, efectos, síntesis, sintetizador, modulador de envolvente, oscilador, LFO, VCA, VCF.

## **ABSTRACT**

In this project, we have developed specific software to analyze and develop an in-depth study of the various fields of Audio Technologies.

The main blocks that develop during the project are:

- Synthesis audio MATLAB™: in which we have developed a subtractive synthesizer and then completed a comprehensive study and analysis of it.
- Granular Synthesis: In this section the concepts of the theory are developed to implement a music production software, or DAW (Digital Audio Workstation) a programmable synthesizer. Using the programming Earsketch™ (<http://earsketch.gatech.edu/>).
- Effects: We have developed a number of effects on MATLAB™ to analyze by analytical testing and actual behavior in depth and understand its operation. Have been implemented effects delay lines (Eco FIR, Eco IIR, Vibrato and Chorus), effects filters (Wah-Wah), effects processing dynamics (compressor) and distorting (Emulator Pedal Tubescreamer company Ibanez™).

The purpose of the project is to provide educational materials to the subject Audio Technologies, 4th grade ITST EPS-UAM.

## **KEY WORDS**

Audio, effects, synthesis, synth, envelope modulator, oscillator, LFO, VCA, VCF.



# GLOSARIO

**ADSR:** Attack, Decay, Sustain, Release.

**DAW:** Digital Audio Workstation.

**Delay:** Efecto de sonido basado en multiplicación y retraso de replicas de la señal de entrada.

**FIR:** Finite Impulse Response.

**IIR:** Infinite Impulse Response.

**LFO:** Low Frequency Oscilator.

**PCM:** Pulse Code Modulation.

**Sample:** Muestra obtenida a partir de un archivo de audio pregrabado.

**VCA:** Voltage Controlled Amplifier.

**VCF:** Voltaje Controlled Filter.

**VCO:** Voltage Controlled Oscillator.



# ÍNDICE DE CONTENIDOS

---

1. INTRODUCCIÓN .....	19
1.1. Motivación .....	19
1.2. Objetivos.....	19
1.3. Metodología y planificación del proyecto .....	20
2. ESTADO DEL ARTE.....	21
2.1. Introducción .....	21
2.1.2. Síntesis de audio .....	21
2.1.3. DAWs y automatización.....	27
2.1.4. Efectos de audio .....	28
3. SÍNTESIS GRANULAR DE AUDIO EN DAW.....	34
3.1. Introducción .....	34
3.2. Síntesis granular .....	34
3.3. Earsketch™.....	35
3.3.1. Introducción a la música programable .....	35
3.3.3. Resultados.....	35
3. 4. Conclusiones.....	41
4. SÍNTESIS SUSTRACTIVA AUDIO EN MATLAB™.....	42
4.1. Introducción .....	42
4.2. VCO (Voltage Controlled Oscillator) Oscilador principal .....	43
4.3. VCF (Voltage Controlled Filter) Filtro controlado por tensión .....	46
4.4. VCA. (Voltage Controlled Amplifier) Amplificador controlado por tensión, o modulador de envolvente.....	51
4.5. LFO (Low-Frequency Oscillator) Oscilador de baja frecuencia .....	53
4.5.1. LFO 1 actúa sobre el VCO .....	54
4.5.2. LFO 2 actúa sobre el VCF .....	56
4.5.3. LFO 3 actúa sobre el VCA.....	60
4.6. Conclusiones.....	63
5. EFECTOS DE AUDIO EN MATLAB™.....	64
5.1. Efectos con líneas de retardo .....	64
5.1.1. ECO FIR.....	64
5.1.2. ECO IIR.....	70
5.1.3. VIBRATO .....	76
5.1.4. CHORUS.....	81
5.2. Efectos con filtros .....	87

5.2.1. Wah-Wah.....	87
5.3. Efectos de procesado de dinámica.....	97
5.3.1 Compresor.....	97
5.4. Distorsión.....	104
5.4.1. Tube screamer.....	104
6. CONCLUSIONES Y TRABAJOS FUTUROS.....	117
6.1. Conclusiones.....	117
6.3. Trabajos futuros.....	118
ANEXO A: CÓDIGO.....	119
ANEXO B: REFERENCIAS.....	120
PRESUPUESTO.....	121
PLIEGO DE CONDICIONES.....	122

## ÍNDICE DE FIGURAS

---

<i>Figura 1. Onda cuadrada a partir de suma de armónicos. (Rossing08).....</i>	<i>21</i>
<i>Figura 2. Esquema síntesis aditiva (Rossing08).....</i>	<i>22</i>
<i>Figura 3. Kawai K5000.....</i>	<i>22</i>
<i>Figura 4. Esquema sintetizador PCM (Rossing08).....</i>	<i>23</i>
<i>Figura 5. Sonido de trompeta sintetizado (Rossing08).....</i>	<i>23</i>
<i>Figura 6. Diagrama de bloques síntesis sustractiva.....</i>	<i>24</i>
<i>Figura 7. MiniMoog™.....</i>	<i>24</i>
<i>Figura 8. Modulación FM: portadora, moduladora y modulada.....</i>	<i>25</i>
<i>Figura 9. Estructura Sintetizador FM.....</i>	<i>26</i>
<i>Figura 10. Espectro sintetizador por modulación.....</i>	<i>26</i>
<i>Figura 11. Logo Reaper™.....</i>	<i>27</i>
<i>Figura 12. Ejemplo entorno de producción en Reaper™.....</i>	<i>27</i>
<i>Figura 13. Tipos de filtro [Zölder11].....</i>	<i>28</i>
<i>Figura 14. Ejemplo de filtro de segundo orden [Zölder11].....</i>	<i>28</i>
<i>Figura 15. Boss OverDrive.....</i>	<i>29</i>
<i>Figura 16. Relación entrada/salida Overdrive [Zölder11].....</i>	<i>30</i>
<i>Figura 17. Ejemplo clipping overdrive. [Zölder11].....</i>	<i>30</i>

Figura 18. Fuzz Face.....	31
Figura 19. Esquemático Fuzz Face .....	31
Figura 20. Clipping y armónicos a la salida del Fuzz [Zölder11] .....	31
Figura 21. Diagrama de bloques de procesamiento de dinámica [Zölder11] .....	32
Figura 22. Ejemplo de curva estática [Zölder11] .....	32
Figura 23. Ejemplo compresión de señal de entrada.....	33
Figura 24. Creación de la base de datos.....	36
Figura 25. Creación de la base de datos.....	36
Figura 26. Función granular_synth.....	37
Figura 27. Resultado en Reaper™ de ejecución granular_synth.....	38
Figura 28. Definición efectos estáticos.....	39
Figura 29. Resultado en Reaper™ de ejecución de efectos estáticos.....	39
Figura 30. Definición efectos dinámicos.....	40
Figura 31. Resultado en Reaper™ de ejecución de efectos dinámicos.....	40
Figura 32. Definición efectos rítmicos.....	41
Figura 33. Resultado en Reaper™ de ejecución de efectos rítmicos .....	41
Figura 34. Diagrama de bloques de síntesis .....	42
Figura 35. Entorno gráfico creado con GUIDE.....	43
Figura 36. Entorno gráfico de la aplicación, control VCO .....	43
Figura 37. Entorno gráfico de la aplicación, control VCF.....	46
Figura 38. Diagrama de polos y ceros en diferentes instantes de tiempo.....	49
Figura 39. Análisis filtro VCF, señal $X(j\omega)$ , $K_1(t)$ , $K_1(j\omega)$ e $Y(j\omega)$ .....	50
Figura 40. Filtro VFC.....	51
Figura 41. Entorno gráfico de la aplicación, control VCA .....	52
Figura 42. Gráfica estándar del modulador de envolvente en Matlab™ .....	52
Figura 43. Entorno gráfico de la aplicación, control LFOs.....	54
Figura 44. Señal procedente del LFO en tiempo y frecuencia (gráficas de arriba), señal a la salida del VCF en tiempo y frecuencia (gráficas de abajo). .....	55
Figura 45. Señal ampliada en frecuencia a la salida del VCF. ....	56
Figura 46. Parámetros introducidos. ....	57
Figura 47. Señal diente de sierra generada por el VCO .....	58
Figura 48. Señal a la salida del módulo VCF.....	58

<i>Figura 49. Señal ampliada en tiempo a la salida del módulo VCF.....</i>	<i>59</i>
<i>Figura 50. Señal a la salida del módulo VCF sin el LFO 2.....</i>	<i>59</i>
<i>Figura 51. Parámetros introducidos. ....</i>	<i>61</i>
<i>VCO -&gt; Type : Sawtooth, Amplitude: 1, Duration: 5 seg., Frequency: 440 Hz. ....</i>	<i>61</i>
<i>VCF -&gt; Fl: 100 Hz, Fh: 11000 Hz, Bandwidth: 1000 Hz. ....</i>	<i>61</i>
<i>LFO 3 -&gt; ON, Type: Sine, Frequency: 10 Hz, Amplitude: 0.1. ....</i>	<i>61</i>
<i>Figura 52. Modulador de envolvente utilizado.....</i>	<i>61</i>
<i>Figura 53. Señal a la salida del sistema.....</i>	<i>62</i>
<i>Figura 54. Señal a la salida del sistema sin LFO 3. ....</i>	<i>62</i>
<i>Figura 55. FIR filtro de peine.....</i>	<i>65</i>
<i>Figura 56. Módulo y fase de <math>H(j\omega)</math> para <math>M = 1</math>.....</i>	<i>66</i>
<i>Figura 57. Módulo y fase de <math>H(j\omega)</math> para <math>M = 2</math>.....</i>	<i>66</i>
<i>Figura 58. Módulo y fase de <math>H(j\omega)</math> para <math>M = 10</math>.....</i>	<i>67</i>
<i>Figura 59. Módulo y fase de <math>H(j\omega)</math> para <math>M = 50</math>.....</i>	<i>67</i>
<i>Figura 60. Módulo y fase de <math>X(j\omega)</math>.....</i>	<i>68</i>
<i>Figura 61. Módulo y fase de <math>Y(j\omega)</math>.....</i>	<i>69</i>
<i>Figura 62. Respuesta al impulso <math>H(j\omega)</math> ECO FIR.....</i>	<i>69</i>
<i>Figura 63. Módulo ampliado de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math>.....</i>	<i>70</i>
<i>Figura 64. IIR filtro de peine.....</i>	<i>70</i>
<i>Figura 65. Módulo y fase de <math>H(j\omega)</math> para <math>M = 1</math>.....</i>	<i>71</i>
<i>Figura 66. Módulo y fase de <math>H(j\omega)</math> para <math>M = 2</math>.....</i>	<i>72</i>
<i>Figura 67. Módulo y fase de <math>H(j\omega)</math> para <math>M = 5</math>.....</i>	<i>72</i>
<i>Figura 68. Módulo y fase de <math>H(j\omega)</math> para <math>M = 10</math>.....</i>	<i>73</i>
<i>Figura 69. Módulo y fase de <math>X(j\omega)</math>.....</i>	<i>74</i>
<i>Figura 70. Módulo y fase de <math>Y(j\omega)</math>.....</i>	<i>74</i>
<i>Figura 71. Módulo de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math>.....</i>	<i>75</i>
<i>Figura 72. Módulo ampliado de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math>.....</i>	<i>75</i>
<i>Figura 73. Vibrato.....</i>	<i>76</i>
<i>Figura 74. Vibrato Muestras (Ref: Digital Audio Effects, Zölzer11).....</i>	<i>77</i>
<i>Figura 75. Módulo y fase de <math>H(j\omega)</math>.....</i>	<i>77</i>
<i>Figura 76. Primeras muestras de las señales <math>x[n]</math> e <math>y[n]</math>.....</i>	<i>78</i>

<i>Figura 77. Módulo y fase de <math>X(j\omega)</math></i> .....	79
<i>Figura 78. Módulo y fase de <math>Y(j\omega)</math></i> .....	79
<i>Figura 79. Módulo de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	80
<i>Figura 80. Módulo ampliado de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	80
<i>Figura 81. Diagrama de bloques Chorus</i> .....	81
<i>Figura 82. Respuesta en frecuencia de la función Low Frequency Noise</i> .....	82
<i>Figura 83. Low Frequency Noise representado en tiempo</i> .....	83
<i>Figura 84. Respuesta al impulso Chorus, ejecución 1</i> .....	83
<i>Figura 85. Respuesta al impulso Chorus, ejecución 2</i> .....	84
<i>Figura 86. Primeras muestras de las señales <math>x[n]</math> e <math>y[n]</math></i> .....	85
<i>Figura 87. Módulo y fase de <math>X(j\omega)</math></i> .....	85
<i>Figura 88. Módulo de la salida <math>Y(j\omega)</math></i> .....	86
<i>Figura 89. Módulo de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	86
<i>Figura 90. Módulo ampliado de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	87
<i>Figura 91. Diagrama Wah-Wah</i> .....	88
<i>Figura 92. Filtro variable</i> .....	88
<i>Figura 93. Módulo y fase de <math>H(j\omega)</math></i> .....	89
<i>Figura 94. Rango de frecuencias recorrido</i> .....	90
<i>Figura 95. Módulo y fase de <math>H(j\omega)</math> para <math>n = 300</math></i> .....	90
<i>Figura 96. Módulo y fase de <math>H(j\omega)</math> para <math>n = 126</math></i> .....	91
<i>Figura 97. Módulo y fase de <math>H(j\omega)</math> para <math>n = 101</math></i> .....	91
<i>Figura 98. Módulo y fase de <math>H(j\omega)</math> para <math>n = 76</math></i> .....	92
<i>Figura 99. Módulo y fase de <math>X(j\omega)</math></i> .....	93
<i>Figura 100. Módulo y fase de <math>Y(j\omega)</math></i> .....	93
<i>Figura 101. Módulo de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	94
<i>Figura 102. Módulo ampliado de <math>X(j\omega)</math> solapado con <math>Y(j\omega)</math></i> .....	94
<i>Figura 103. Señales <math>x[n]</math> e <math>y[n]</math></i> .....	95
<i>Figura 104. Señales <math>x[n]</math> e <math>y[n]</math> ampliadas</i> .....	96
<i>Figura 105. Ejemplo de curva estática [Zölder11]</i> .....	98
<i>Figura 106. Diagrama de bloques Compresor</i> .....	98
<i>Figura 107. Señales <math>x[n]</math> e <math>y[n]</math> <math>CT = -20</math> dB <math>CS = 0.5</math></i> .....	99

Figura 108. Señales $x[n]$ e $y[n]$ $CT = -20$ dB $CS = 0.3$ .....	100
Figura 109. Señales $x[n]$ e $y[n]$ $CT = -20$ dB $CS = 0.7$ .....	101
Figura 110. Señales $x[n]$ e $y[n]$ $CT = -30$ dB $CS = 0.5$ .....	102
Figura 112. Señales $x[n]$ e $y[n]$ .....	103
Figura 113. Señales $x[n]$ e $y[n]$ ampliadas .....	104
Figura 114. Pedal de efectos Tube Screamer .....	105
Figura 115. Circuito Distorsión TubeScreamer.....	105
Figura 116. Señal de salida $y[n]$ respecto a $x[n]$ .....	106
Figura 117. Señales $y[n]$ $x[n]$ en tiempo.....	107
Figura 118. Señal de salida $y[n]$ respecto a $x[n]$ para $R_{in} = 100$ ohm.....	108
Figura 119. Señales $y[n]$ $x[n]$ en tiempo para $R_{in} = 100$ ohm.....	108
Figura 120. Señal de salida $y[n]$ respecto a $x[n]$ para $R_{in} = 25$ ohm .....	109
Figura 121. Señales $y[n]$ $x[n]$ en tiempo para $R_{in} = 25$ ohm .....	109
Figura 122. Señal de salida $y[n]$ respecto a $x[n]$ para $R_f = 1000$ ohm.....	110
Figura 123. Señales $y[n]$ $x[n]$ en tiempo para $R_f = 1000$ ohm.....	111
Figura 124. Señal de salida $y[n]$ respecto a $x[n]$ para $R_f = 10$ ohm.....	111
Figura 125. Señales $y[n]$ $x[n]$ en tiempo para $R_f = 10$ ohm.....	112
Figura 126. Señal de salida $y[n]$ respecto a $x[n]$ para $I_o = 0.05$ A.....	113
Figura 127. Señales $y[n]$ $x[n]$ en tiempo para $I_o = 0.05$ A.....	113
Figura 128. Señal de salida $y[n]$ respecto a $x[n]$ para $I_o = 0.001$ A.....	114
Figura 129. Señales $y[n]$ $x[n]$ en tiempo para $I_o = 0.001$ A.....	114
Figura 130. Señales $x[n]$ e $y[n]$ .....	115
Figura 131. Señales $x[n]$ e $y[n]$ ampliadas .....	116
Figura 132. Señal de salida $y[n]$ respecto a $x[n]$ .....	116

## ÍNDICE DE FÓRMULAS

---

Fórmula 1.....	21
Fórmula 2.....	21
Fórmula 3.....	25
Fórmula 4.....	29



<i>Fórmula 5</i> .....	29
<i>Fórmula 6</i> .....	29
<i>Fórmula 7</i> .....	29
<i>Fórmula 8</i> .....	29
<i>Fórmula 9</i> .....	47
<i>Fórmula 10</i> .....	47
<i>Fórmula 11</i> .....	47
<i>Fórmula 12</i> .....	47
<i>Fórmula 13</i> .....	47
<i>Fórmula 14</i> .....	54
<i>Fórmula 15</i> .....	54
<i>Fórmula 16</i> .....	54
<i>Fórmula 17</i> .....	65
<i>Fórmula 18</i> .....	65
<i>Fórmula 19</i> .....	71
<i>Fórmula 20</i> .....	71
<i>Fórmula 21</i> .....	76
<i>Fórmula 22</i> .....	76
<i>Fórmula 23</i> .....	76
<i>Fórmula 24</i> .....	81
<i>Fórmula 25</i> .....	82
<i>Fórmula 26</i> .....	82
<i>Fórmula 27</i> .....	88
<i>Fórmula 28</i> .....	88
<i>Fórmula 29</i> .....	88
<i>Fórmula 30</i> .....	88
<i>Fórmula 31</i> .....	88
<i>Fórmula 32</i> .....	97
<i>Fórmula 33</i> .....	98
<i>Fórmula 34</i> .....	105
<i>Fórmula 35</i> .....	106



# 1. INTRODUCCIÓN

---

## 1.1. Motivación

En este proyecto se pretende desarrollar prácticas de laboratorio de la asignatura, Tecnologías del Audio, de cuarto curso del Grado en Ingeniería de Tecnologías y Servicios (ITST) de la Universidad Autónoma de Madrid.

Estas prácticas tratan de introducir al alumno en distintos ámbitos y campos de la ingeniería de sonido, mediante el estudio de la síntesis de audio, producción de sonido y post-procesado de audio.

Todo material docente ha de ser estudiado exhaustivamente y desarrollado, para posteriormente, transmitir esos conocimientos a los alumnos.

## 1.2. Objetivos

El objetivo general del proyecto es hacer un estudio en profundidad y desarrollar plataformas software para las prácticas de la asignatura Tecnologías del Audio, de 4º de Grado ITST de la EPS-UAM.

El objetivo principal es aprender las técnicas básicas de ingeniería de sonido, a nivel de procesado (síntesis de audio), de programación de interfaces de Digital Audio Workstations (DAW) y de post-procesado de audio. Asimismo, el estudiante aprenderá los procesos necesarios para generar los materiales necesarios, para poner en funcionamiento prácticas de laboratorio de calidad, a partir de los subproyectos desarrollados.

Como objetivos particulares se citan los siguientes:

- Diseño e implementación de un sintetizador sustractivo en Matlab™.
- Diseño e implementación de un sintetizador granular utilizando la plataforma EarSketch™ y la Estación de Trabajo de Audio Digital (Digital Audio Workstation, DAW) Cockos Reaper™, que se utilizará en las prácticas de la asignatura.
- Diseño y realización de un set de efectos para el post-procesado de audio utilizando la herramienta Matlab™.

### 1.3. Metodología y planificación del proyecto

En este apartado se presenta la metodología seguida para la realización del Proyecto de Fin de Carrera y una estimación del tiempo empleado para su desarrollo.

- **Documentación previa:** Se lleva a cabo durante los primeros meses profundizando en los campos de estudio de Síntesis de audio, desarrollo de software para DAWs y efectos de audio digitales.
- **Desarrollo:** Implementación del código para generar las múltiples aplicaciones y funciones de cada apartado, que han facilitado la creación de un sintetizador granular, un sintetizador sustractivo y un set de efectos de audio.
- **Resultados y conclusiones:** Se han realizado pruebas y análisis para verificar el funcionamiento de todo el software desarrollado con el fin de poder utilizarlo como material docente en la asignatura Tecnologías de Audio , de 4º de grado ITST de la EPS-UAM.
- **Memoria:** La memoria se ha ido completando según se han obtenido los resultados requeridos en cada apartado y en los últimos meses se ha unificado y finalizado por completo.

ACTIVIDAD	HORAS DEDICADAS	2013			2014						
		OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL
<b>Síntesis granular de audio en DAW</b>											
Estudios previos y familiarización con DAWs	60	█	█								
Estudio y desarrollo de software para DAWs	25	█	█								
Estudio y profundización en sintetizadores	20	█	█								
Desarrollo del Sintetizador Granular	60		█	█	█						
Pruebas, análisis y documentación	35			█	█						
<b>Síntesis de audio en Matlab™</b>											
Realización del curso Introduction to Digital Sound Desing	21			█	█						
Estudios previos de síntesis sustractiva	20				█	█					
Desarrollo software del Sintetizador Sustractivo en Matlab™	60				█	█	█				
Pruebas, análisis y documentación	50					█	█	█			
<b>Efectos de audio en Matlab™</b>											
Estudios previos de efectos de audio digitales	30				█	█					
Desarrollo del set de efectos en Matlab™	130					█	█	█	█	█	█
Pruebas, análisis y documentación	60						█	█	█	█	█
<b>Memoria</b>	30			█	█	█	█	█	█	█	█

## 2. ESTADO DEL ARTE

---

### 2.1. Introducción

Este proyecto abarca un estudio específico de tres grandes campos en las técnicas de análisis y procesamiento de audio, por lo que se desarrollará en esta sección cada apartado individualmente.

#### 2.1.2. Síntesis de audio

La síntesis de audio consiste en generar sonido a partir de elementos no acústicos o con software específico. Hay diferentes técnicas de síntesis de audio, las más importantes son las siguientes:

- **Síntesis Aditiva:**

La síntesis aditiva consiste en la generación de sonido a partir de ondas complejas, utilizando la superposición de diferentes ondas simples. Este método de síntesis está basado en el Teorema de Fourier, el cual propuso que cualquier señal se podía descomponer en una suma infinita de funciones continuas.

$$F(t) = a_0 + \sum_m [b_m \cos(2\pi f_0 m t) + c_m \sin 2\pi f_0 m t] \quad \text{(Fórmula 1)}$$

$$A_m = \sqrt{b_m^2 + c_m^2} \quad \theta_m = \arctan(c_m/b_m) \quad \text{(Fórmula 2)}$$

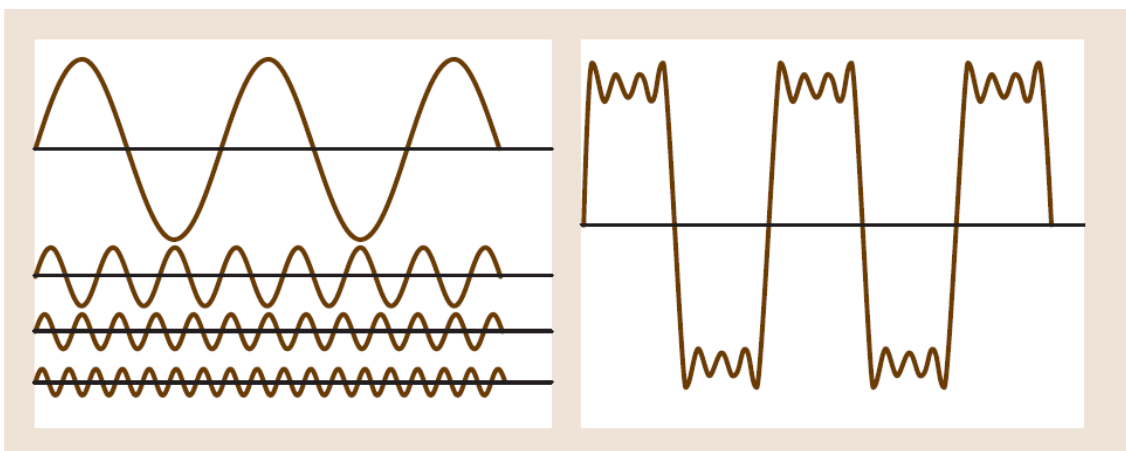


Figura 1. Onda cuadrada a partir de suma de armónicos. (Rossing08)

El esquema básico para generar sonido a partir de ondas simples es el siguiente:

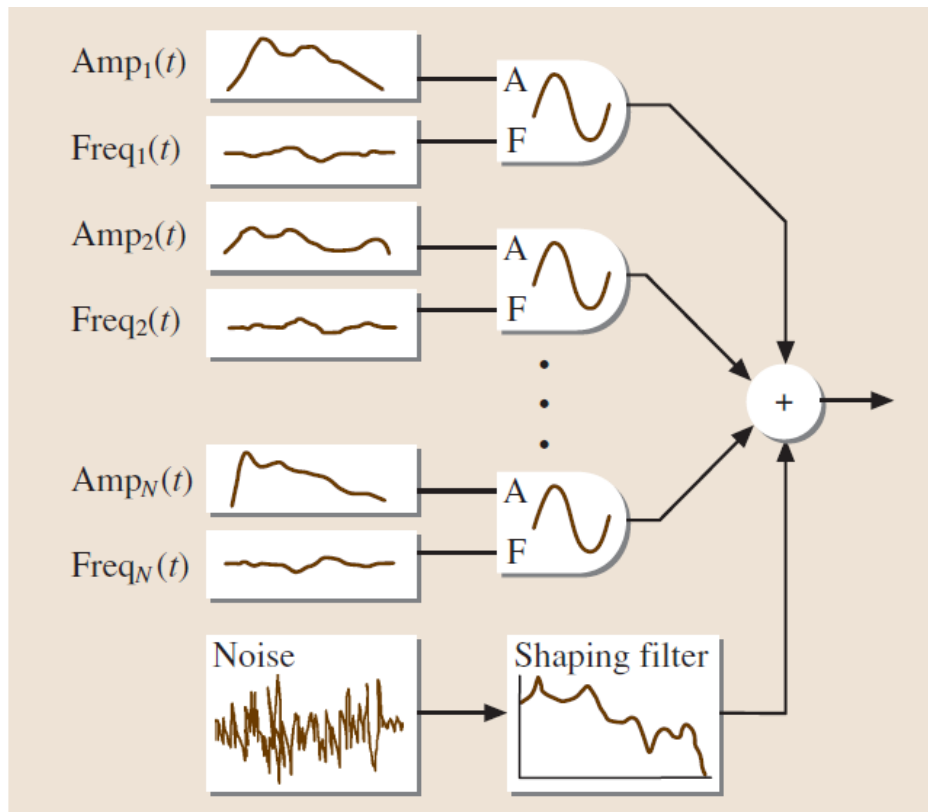


Figura 2. Esquema síntesis aditiva (Rossing08)

Se generan armónicos a partir de amplitudes y frecuencias variables en el tiempo. Todas las señales generadas se suman con ruido filtrado para obtener el espectro deseado.

Un claro ejemplo de síntesis aditiva es el funcionamiento de los órganos clásicos o la serie k5000 de los sintetizadores de la marca Kawai.



Figura 3. Kawai K5000

### ○ Síntesis PCM (Pulse Code Modulation):

Es un tipo de síntesis basada en muestras obtenidas de audio que se manipulan y concatenan para generar diferentes sonidos. El esquema básico de un sintetizador PCM se muestra a continuación:

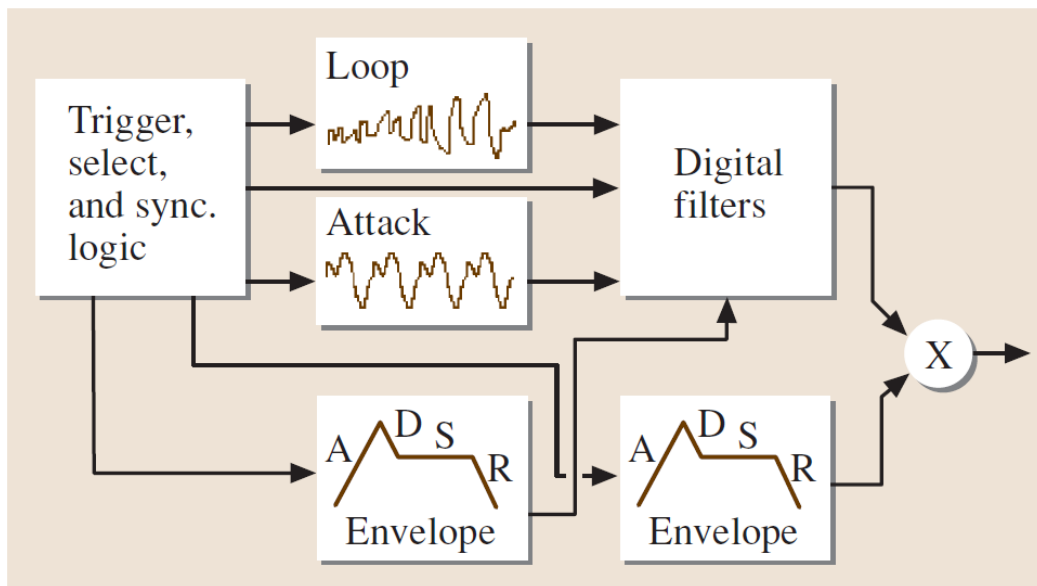


Figura 4. Esquema sintetizador PCM (Rossing08)

Este sintetizador genera las muestras de audio a partir de unos registros digitales de sonido existente. Mantiene las muestras reproduciendo continuamente en un bucle hasta que se desea y para obtener una característica más real del sonido se usan moduladores de envolvente y filtros digitales.

A continuación se muestra un ejemplo de un sonido de trompeta generado por un sintetizador PCM:

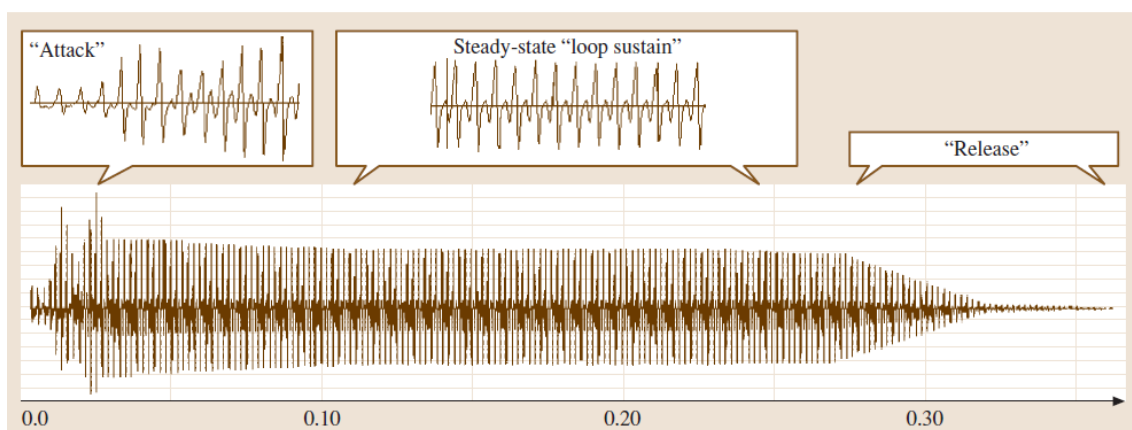
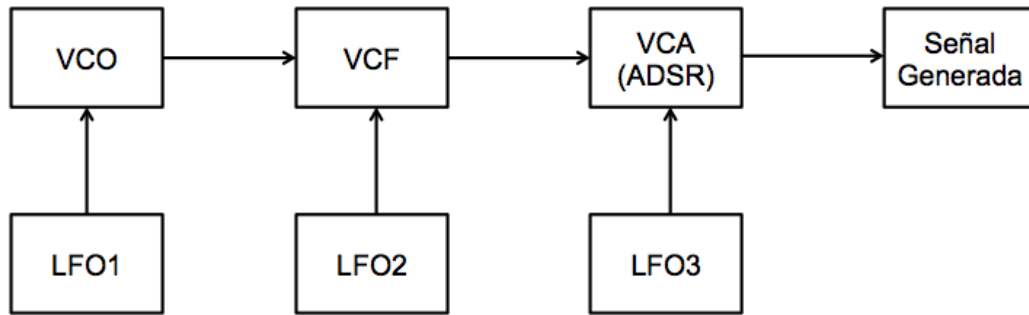


Figura 5. Sonido de trompeta sintetizado (Rossing08)

#### ○ Síntesis Sustractiva:

La síntesis sustractiva es el método por el cual se produce sonido a partir de osciladores y filtros analógicos. El modelo se basa en el siguiente diagrama de bloques:



**Figura 6. Diagrama de bloques síntesis sustractiva.**

Su funcionamiento se basa en generar una señal con el VCO (Voltage Control Oscillator), posteriormente filtrarla con un el VCF (Voltage Control Filter) y se hace pasar por un modulador de envolvente VCA (Voltage Control Amplifier). Todos estos elementos del sistema pueden ser controlados por osciladores de baja frecuencia LFO (Low Frequency Oscillator).

Al aplicar individualmente cada LFO a cada elemento, se genera un efecto característico:

- LFO1 : Aplica una modulación FM a la señal procedente del VCO y genera el efecto vibrato.
- LFO2 : Aplicado al VCF genera un filtro con resonancia variable y se produce el efecto Wah-Wah.
- LFO3 : Aplicado al VCA genera un cambio de amplitudes y produce el efecto tremolo.

Un ejemplo claro de síntesis sustractiva, es el famoso MiniMoog™.



**Figura 7. MiniMoog™**

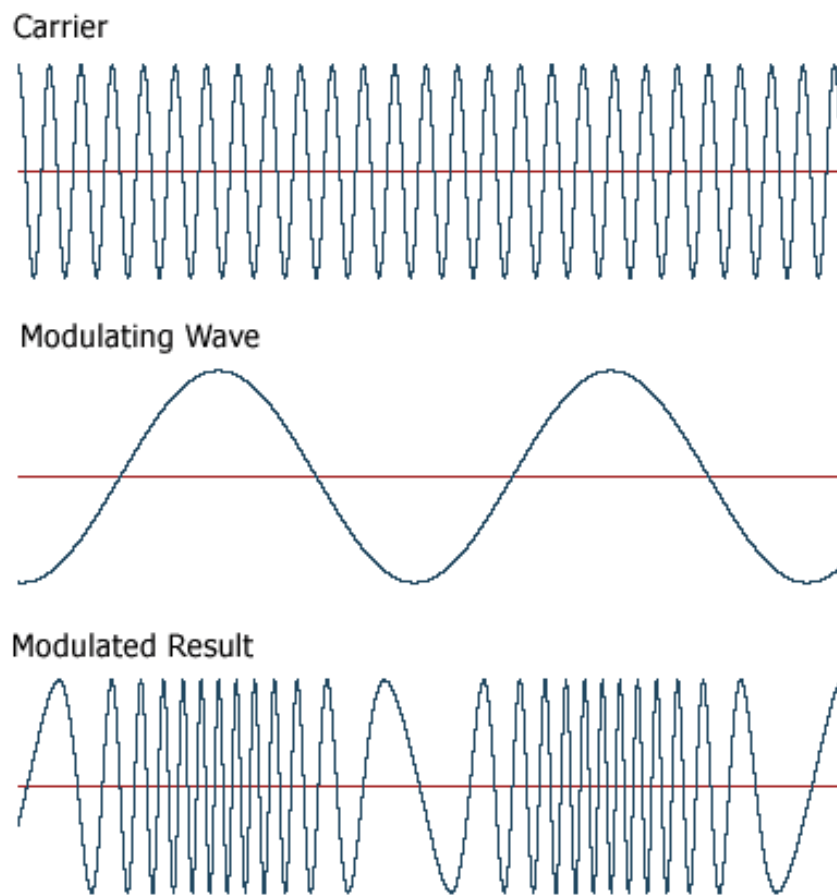


○ **Síntesis por modulación:**

Como su propio nombre indica, se realiza mediante la modulación de una señal portadora y otra moduladora, los sintetizadores más famosos que utilizan esta técnica son los FM, evidentemente, utilizan moduladores FM para generar sonido.

$$y(t) = \sin[2\pi t f_c + \Delta f_c \cos(2\pi t f_m)] \quad \text{(Fórmula 3)}$$

Dónde  $f_c$  es la frecuencia portadora y  $f_m$  la frecuencia moduladora y el índice de modulación viene definido por  $i = (\Delta f_c / f_c)$ . Este índice de modulación indica el número de bandas laterales que se pueden generar.



**Figura 8. Modulación FM: portadora, moduladora y modulada**

Este tipo de sintetizador utiliza mayor número de osciladores, por lo que es más eficiente a la hora de generar señales complejas con múltiples frecuencias. Su diagrama de bloque básico viene dado por la siguiente estructura:

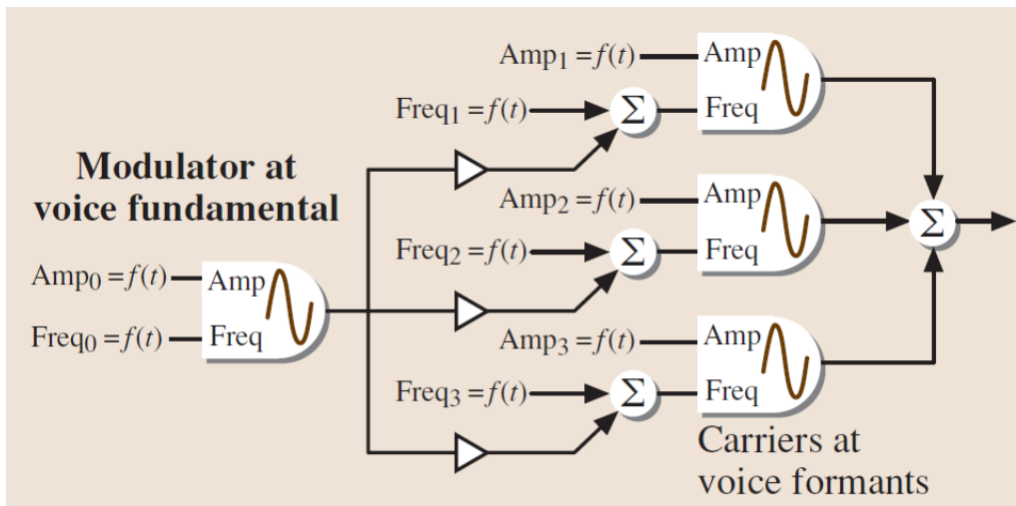


Figura 9. Estructura Sintetizador FM.

Ejemplo del espectro resultante a la salida del sintetizador por modulación:

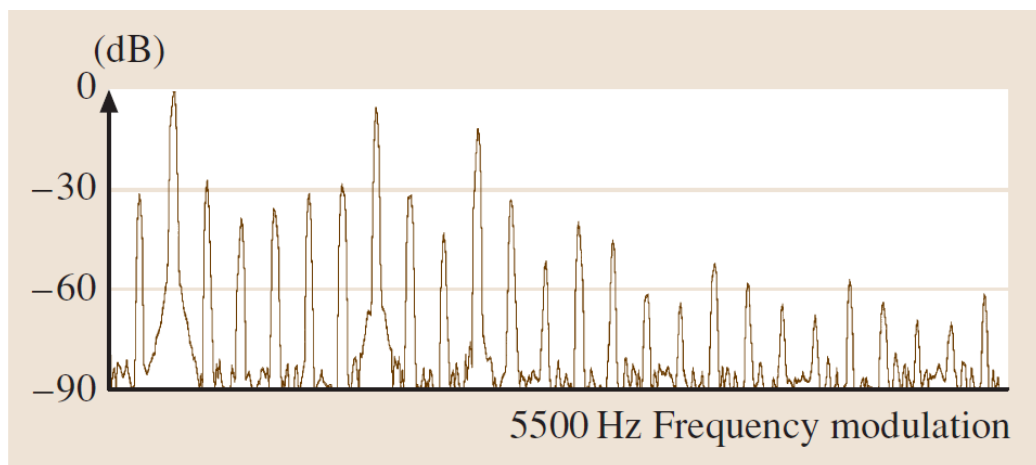


Figura 10. Espectro sintetizador por modulación.

○ **Síntesis granular:**

La síntesis granular está basada en la teoría de la fragmentación del sonido en eventos de duración mínima (entre 5 y 100 milisegundos) llamados granos y la posterior utilización de dichos granos para la síntesis de audio.

La primera teoría desarrollada de la síntesis granular fue de Iannis Xenakis en los años 60, en la cual estipuló una teoría compositiva:

<http://www.iannis-xenakis.org/Articles/Georgaki.pdf>

Más adelante, en los años 70 Curtis Roads implementó digitalmente la técnica de síntesis granular. [8]

### 2.1.3. DAWs y automatización

Las DAWs o Digital Audio Workstation son sistemas electrónicos utilizados para grabar, editar y producir audio digital. Estos sistemas se componen hardware para realizar la conversión analógico-digital del audio y software específico para editar el audio y procesarlo.

Cockos Reaper™ (Rapidd Environment for Audio Production, Engineering and Recording) es una DAW desarrollada por Cockos que permite realizar cualquier tarea de grabación, edición y producción.



Figura 11. Logo Reaper™.

La herramienta EarSketch™ desarrollada por Georgia Tech para ser utilizada por Reaper™ posee un compilador específico que permite realizar composiciones musicales a través de la programación musical, ejecutando código en lenguaje Python. EarSketch™ está diseñado para automatizar tareas tediosas en el ámbito de la producción musical.



Figura 12. Ejemplo entorno de producción en Reaper™.

### 2.1.4. Efectos de audio

Los efectos de audio son utilizados para matizar, realzar o transformar alguna cualidad del audio original. Son utilizados para el post-procesado y producción musical. La principal característica de los efectos en audio es la posibilidad de controlar o manipular alguna serie de parámetros, para modificar las características de la señal de entrada. Los tres principales géneros de efectos que más se utilizan son los siguientes:

- **Filtros y delays:**

Los filtros modifican la amplitud y la fase de la señal de entrada, realizan una selección de ciertos parámetros de acuerdo con las frecuencias que se quieren rechazar, retener o enfatizar. Se pueden ver como un sistema lineal que realiza un cambio en el espectro de la señal de entrada. Lo tipos de filtros más comunes son: paso bajo (LP), paso banda (BP), paso alto (HP), paso todo (Allpass) o de banda rechazada (BR).

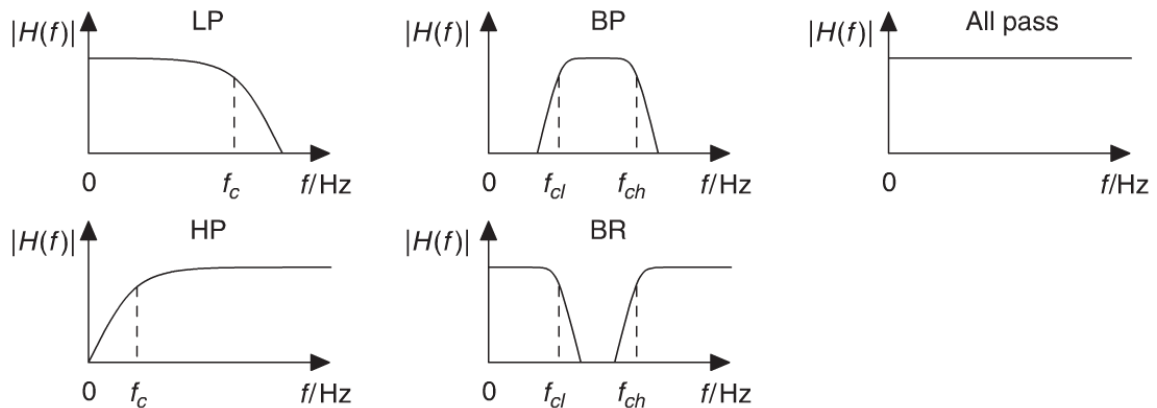


Figura 13. Tipos de filtro [Zölder11]

Existen varios métodos para implementar filtros digitales, con la forma canónica se pueden obtener filtros de cualquier orden. En la siguiente figura se muestra la forma canónica de un filtro de segundo orden:

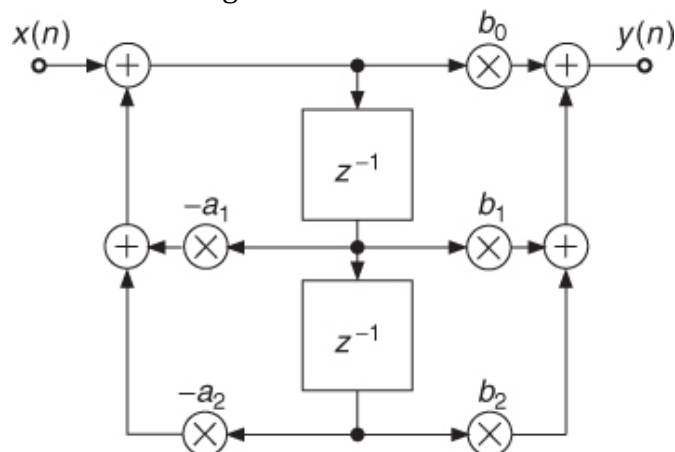


Figura 14. Ejemplo de filtro de segundo orden [Zölder11]

Su función de transferencia se define como:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad \text{(Fórmula 4)}$$

Eliminando la segunda línea de retardo  $a_2 = 0$  y  $b_2 = 0$  se obtendría un filtro de primer orden. Con este filtro se pueden implementar los filtros típicos variando los parámetros restantes:

- Paso bajo (LP):  $b_0 = K/K + 1$        $b_1 = K/K + 1$       (Fórmula 5)  
 $a_1 = (K - 1)/(K + 1)$

- Paso alto (HP):  $b_0 = 1/(K + 1)$        $b_1 = -1/(K + 1)$       (Fórmula 6)  
 $a_1 = (K - 1)/(K + 1)$

- Paso todo (Allpass):  $b_1 = (K - 1)/(K + 1)$        $b_2 = 1$       (Fórmula 7)  
 $a_1 = (K - 1)/(K + 1)$

Dónde la variable K depende de la frecuencia de corte del filtro:

$$K = \tan(\pi f_c / f_s) \quad \text{(Fórmula 8)}$$

Lo efectos digitales basados en retardos (Delay, ECO...) también se basan en la implementación con filtros de este tipo.

○ **Distorsión no lineal:**

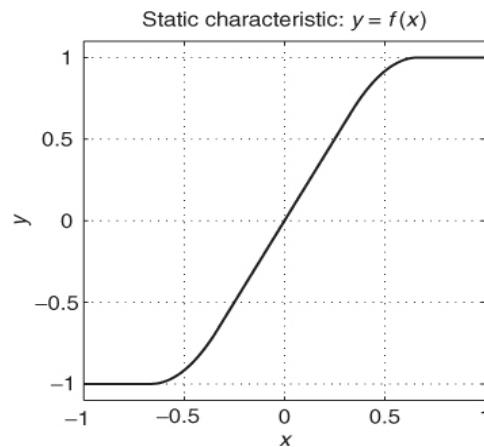
La distorsión se basa en la deformación de la señal de entrada con respecto a la señal de salida. Hay múltiples implementaciones y diseños de distorsionadores, pero los más característicos son:

- Overdrive:



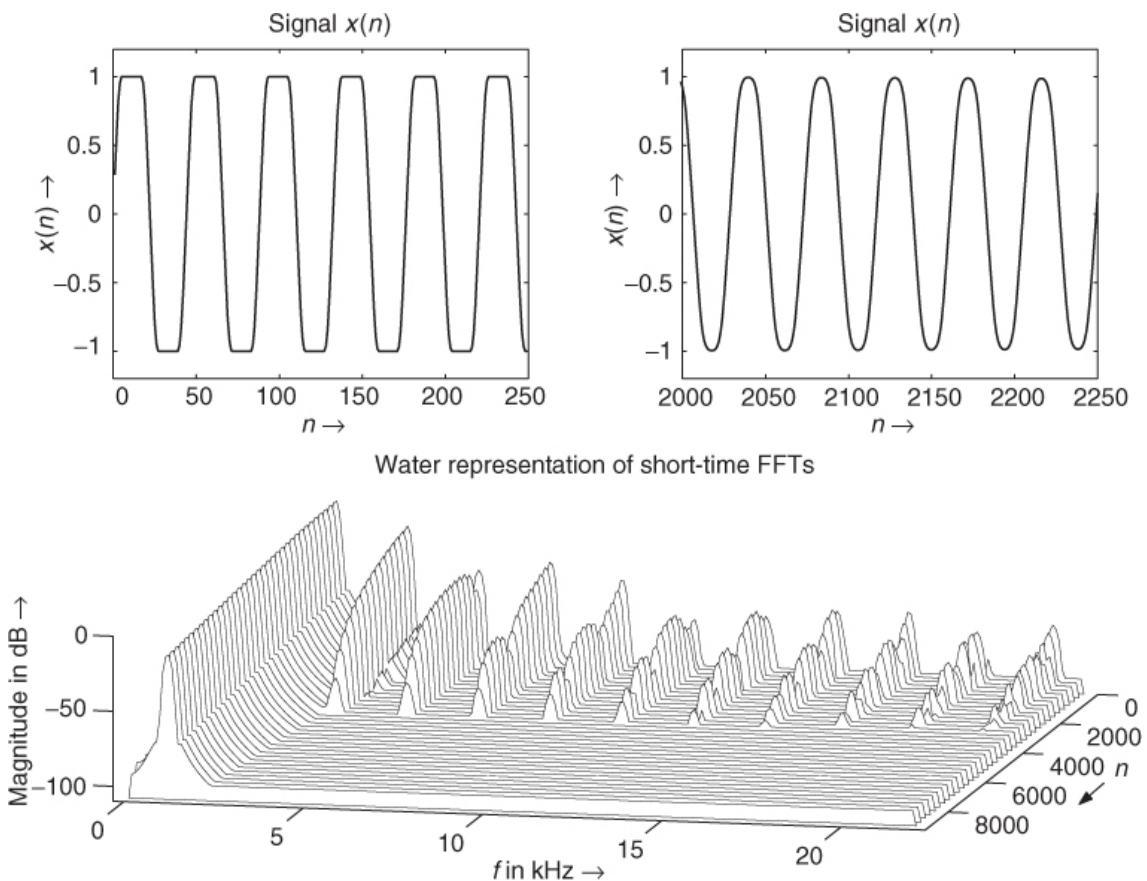
Figura 15. Boss OverDrive

Este efecto viene definido por la curva estática que relaciona la señal de entrada con la de salida, en la cual, se puede observar que la señal de salida quedará distorsionada, cuando la señal de entrada alcance un cierto valor. Mientras que para los valores bajos de la señal de entrada, su comportamiento es lineal.



**Figura 16. Relación entrada/salida Overdrive [Zölder11]**

Introduciendo una señal sinusoidal a la entrada del overdrive, se observa a la salida que la señal queda recortada tanto en los niveles superiores como en los inferiores (figura de la izquierda), este fenómeno se denomina clipping y es muy característico de la distorsión, produce armónicos a diferentes frecuencias, como se puede observar en la figura de abajo.



**Figura 17. Ejemplo clipping overdrive. [Zölder11]**

➤ Fuzz:

El efecto fuzz, se encarga de saturar asimétricamente la señal de entrada produciendo otro tipo de distorsión no lineal, también genera un recorte de la señal de salida o clipping.



Figura 18. Fuzz Face

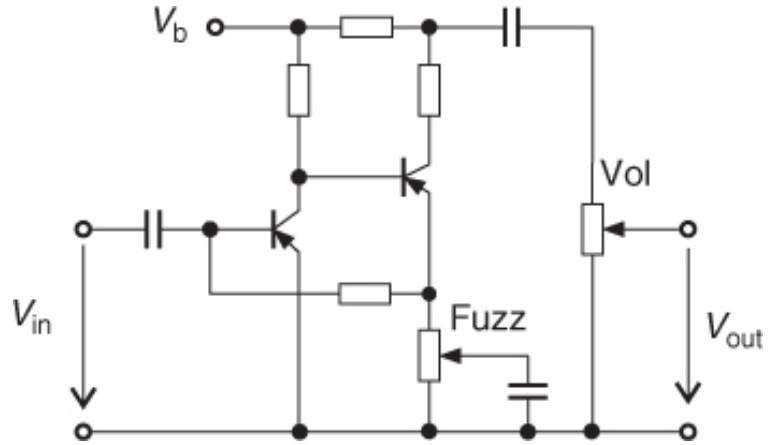


Figura 19. Esquemático Fuzz Face

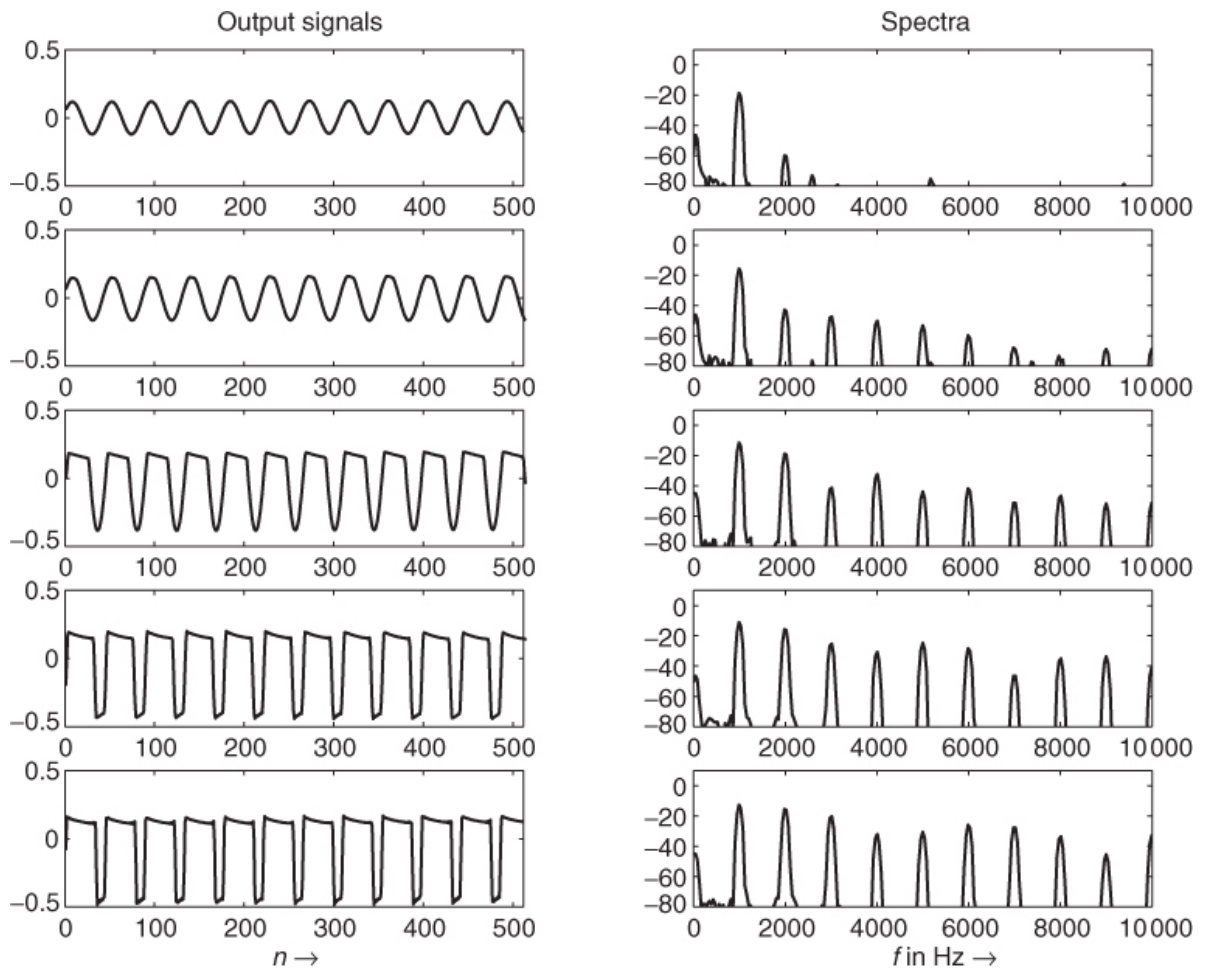
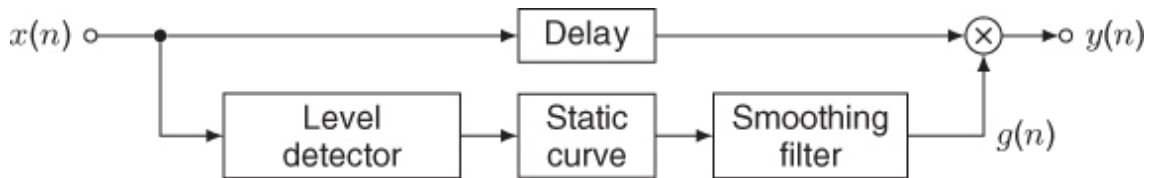


Figura 20. Clipping y armónicos a la salida del Fuzz [Zölder11]

○ **Procesado de dinámica:**

El procesado de dinámica se basa en el control de la intensidad percibida del sonido (Loudness). Este efecto digital actúa de manera automática controlando la ganancia de la señal de salida mediante el nivel de la señal de entrada.

El esquema básico de este tipo de efecto es el siguiente:

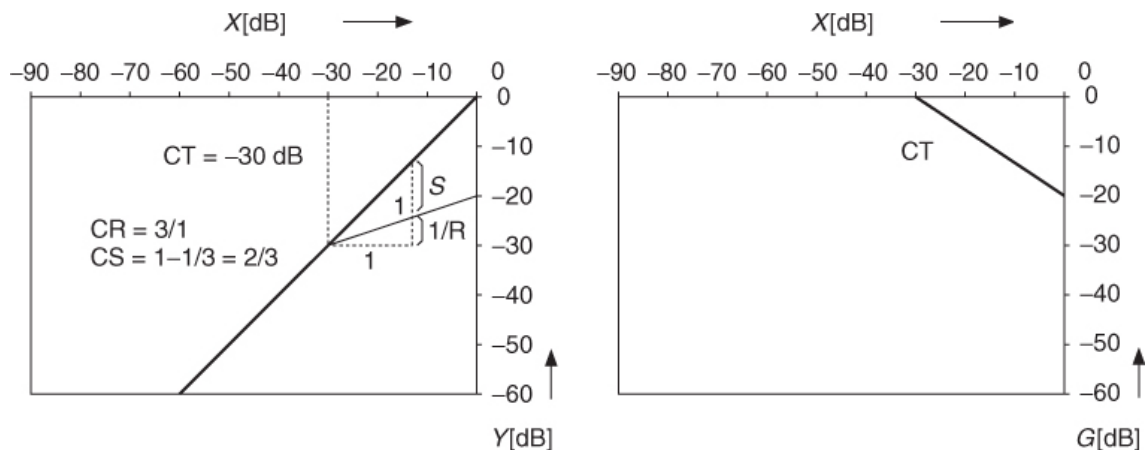


**Figura 21. Diagrama de bloques de procesamiento de dinámica [Zölder11]**

- Level detector: Detector de nivel, determina el nivel de la señal de entrada.
- Static curve: Curva estática, relación entre la amplitud de entrada y la de salida.
- Smoothing filter: Filtro de suavizado, previene transiciones bruscas.

Se puede observar en la estructura, que se está detectando continuamente el nivel de entrada de la señal y se actúa sobre la misma modificando su nivel a la salida.

La curva estática determina el funcionamiento del procesador de dinámica, en la siguiente gráfica de ejemplo se puede observar los parámetros que la controlan:



**Figura 22. Ejemplo de curva estática [Zölder11]**

Threshold (CT) : Nivel de entrada a partir del cual la ganancia de la curva es diferente a 0 dB.

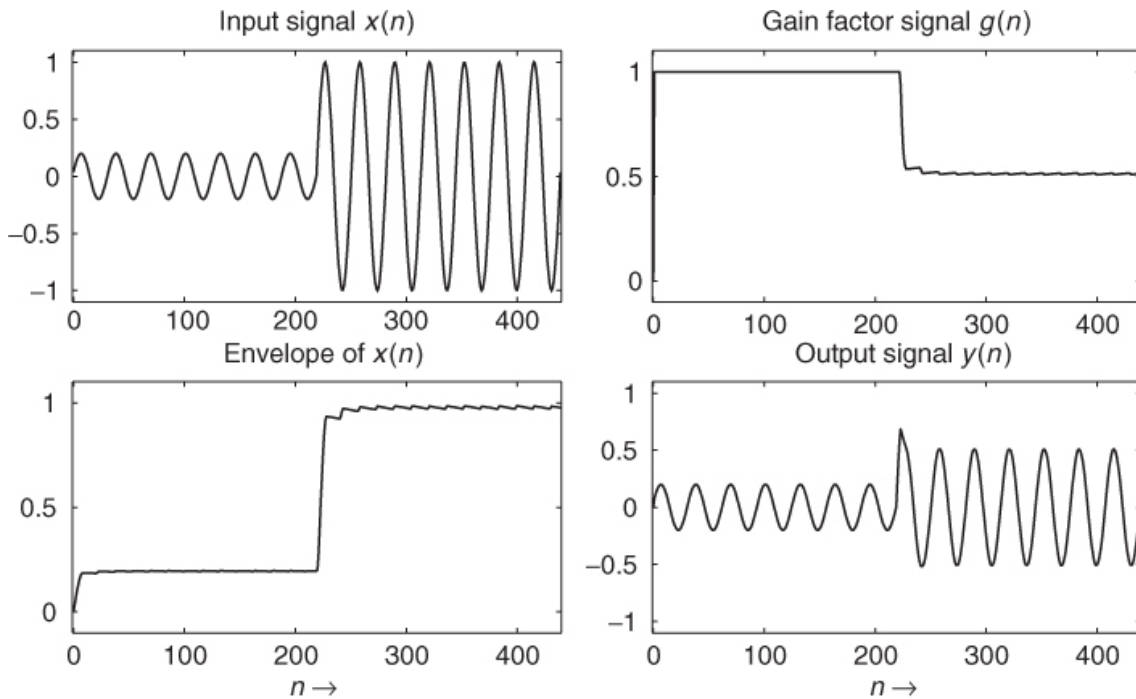
Ratio (CR) : Razón entre el incremento de la entrada frente al de la salida.

Slope (CS) : Indica el valor de reducción de la ganancia.



Dependiendo de la elección de los parámetros y del umbral actuación de los mismos se pueden diseñar los dos efectos más típicos en procesamiento de dinámica: el compresor o el expansor.

A continuación se muestra un ejemplo gráfico del funcionamiento de un compresor, reduciendo el valor de entrada de la señal si supera un cierto umbral:



**Figura 23. Ejemplo compresión de señal de entrada**

## 3. SÍNTESIS GRANULAR DE AUDIO EN DAW

---

Este capítulo constituye la base para el material docente de la práctica de síntesis granular, tanto para el profesor como para el estudiante.

### 3.1. Introducción

Una de las herramientas primordiales para grabar, producir, masterizar y mezclar cualquier muestra de sonido son las DAWs (Digital Audio Workstation). Hay una gran cantidad de productos dedicados a realizar dichas tareas y para este apartado, se va a utilizar la herramienta Reaper™ con Earsketch™ instalado. Este software permite realizar composiciones musicales a través de la programación musical.

El objetivo de este apartado es realizar un sintetizador granular utilizando una base de datos de muestras pregrabadas de audio, o *samples*, y posteriormente producir y masterizar la composición. Todo ello realizado con Earsketch™, que utiliza el lenguaje de programación Python y trabaja sobre el motor de Reaper™.

### 3.2. Síntesis granular

Uno de los nuevos campos de la producción de sonido es la síntesis granular, que se basa en obtener granos aleatorios de muestras de audio y agruparlos en pequeños instantes temporales de milisegundos de separación entre ellos.

Históricamente se han creado múltiples teorías sobre la síntesis granular, la primera fue la de Iannis Xenakis en los años 60', la cual se basaba en la composición de granos sonoros. La teoría se basa en la producción de una alta densidad de pequeños eventos acústicos llamados granos, de menos de 50 ms de duración y típicamente en el intervalo de 10 a 30 ms. La densidades típicas de grano van desde cientos a varios miles de granos por segundo [8].

Los granos pueden venir de muestras de ondas puras (sinusoidal, cuadrada, triangular...), o de fragmentos de muestras de audio (samples).

Mediante la variación de la forma de onda, envolvente, duración, posición espacial, y la densidad de los granos, se pueden producir infinidad de sonidos.

## 3.3. Earsketch™

### 3.3.1. Introducción a la música programable

Earsketch™ es una herramienta de Reaper™ que permite realizar cualquier funcionalidad del programa, gracias a un potente compilador que ejecuta script escritos en código Python.

¿Por qué programar música? Todavía hay cosas que no se pueden hacer con facilidad, o en absoluto, con un programa DAW : la programación puede ofrecer a los músicos un control más flexible sobre la música, lo que les permite crear algo verdaderamente único y personal. Además pueden automatizarse tareas tediosas, como la síntesis granular, ya que el posicionamiento de las muestra de audio o la elección de los instantes temporales donde deben ir es completamente aleatoria y requiere un gran volumen de datos a tratar.

Earsketch™ además de permitir realizar música programable, está creado como un proyecto social de enseñanza abierta, permite a los usuarios aprender y compartir sus conocimiento en comunidad [7].

En el siguiente apartado, se detallarán los pasos de la creación de un sintetizador granular utilizando Earsketch™.

### 3.3.3. Resultados

- **Creación de la base de datos:**

En primer lugar, se ha decidido tener libertad a la hora de seleccionar los archivos de audio con los que se diseñará nuestro sintetizador granular, para tener una gran versatilidad a la hora de crear composiciones. En este caso, se han utilizado varias muestras de audio de un sintetizador Roland System 100.

Para añadir los archivos que se deseen a la base de datos de Earsketch™, se debe abrir la aplicación Reaper™ e importar los archivos. Seleccionamos las pistas y pinchamos en **Actions -> Export Audio Selection**.

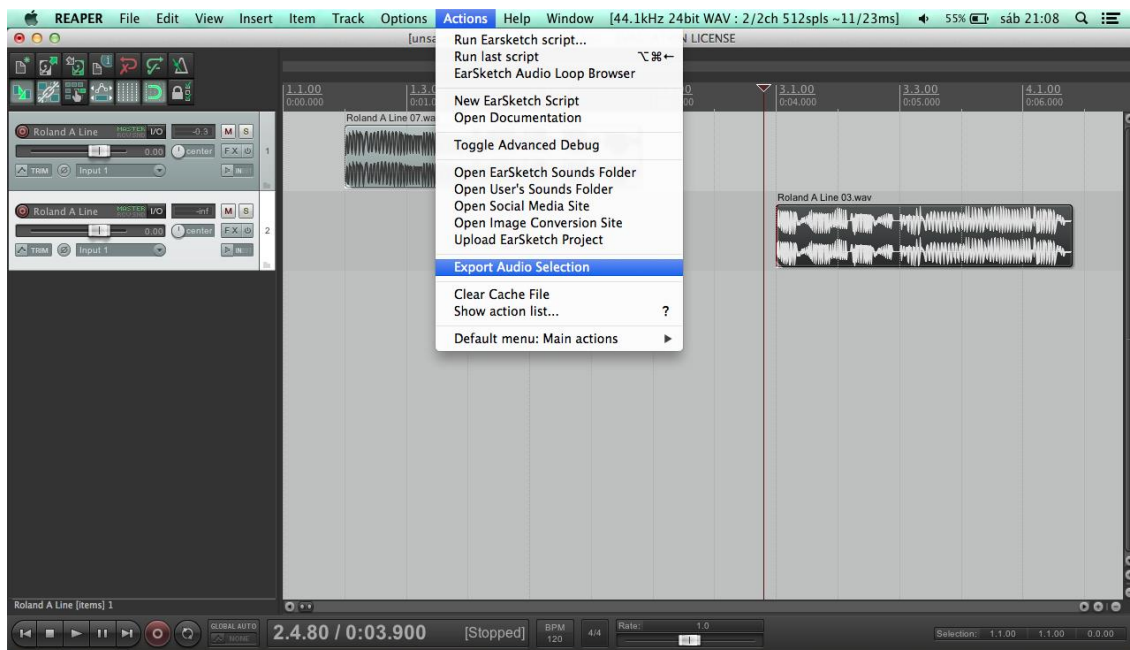


Figura 24. Creación de la base de datos

Guardamos los archivos en la carpeta Documentos/ReaperMedia/EarSketch Sounds. Para comprobar que se han añadido con éxito los archivos a la base de datos pinchamos en **Actions -> EarSketch Audio Loop Browser**, nos saldrá una ventana del explorador y pincharemos en la pestaña USER SOUNDS, tendrá que salir una lista con todos los archivos añadidos.

- **Estructura del sintetizador:**

Una vez creada nuestra base de datos, el siguiente paso será desarrollar el código para que genere nuestro sintetizador, basándose en la teoría expuesta en la introducción.

En primer lugar, se genera una variable que contenga los archivos de audio de la base de datos. Nuestra variable se llama synth y contiene 35 samples del sintetizador Roland System 100.

```

10 #
11 from reaper_python import *
12 from earsketch import *
13
14
15 init()
16 setTempo(120)
17
18 #list of audio
19 synth = [ROLAND_A_A1, ROLAND_A_A2, ROLAND_A_A3, ROLAND_A_A4,
20          ROLAND_A_A5, ROLAND_A_B1, ROLAND_A_B2, ROLAND_A_B3,
21          ROLAND_A_B4, ROLAND_A_B5, ROLAND_A_C1, ROLAND_A_C2,
22          ROLAND_A_C3, ROLAND_A_C4, ROLAND_A_C5, ROLAND_A_C6,
23          ROLAND_A_D1, ROLAND_A_D2, ROLAND_A_D3, ROLAND_A_D4,
24          ROLAND_A_D5, ROLAND_A_E1, ROLAND_A_E2, ROLAND_A_E3,
25          ROLAND_A_E4, ROLAND_A_E5, ROLAND_A_F1, ROLAND_A_F2,
26          ROLAND_A_F3, ROLAND_A_F4, ROLAND_A_F5, ROLAND_A_G1,
27          ROLAND_A_G2, ROLAND_A_G3, ROLAND_A_G4, ROLAND_A_G5]
28

```

Figura 25. Creación de la base de datos

A continuación definimos una función, llamada **granular\_synth**, que se encarga de generar el sintetizador y es llamada por referencia con una serie de parámetros. Dicha función realiza los siguientes pasos:

- Genera un proyecto en Reaper™ que empieza y termina según las variables pasadas por referencia **startMeasure** y **endMeasure**.
- Crea tantas pistas de audio en el proyecto como se le pasen por referencia en la variable pasada por referencia **Ntracks**.
- Crea una variable aleatoria llamada **Sample** que se utiliza para extraer un archivo aleatorio de nuestra base de datos. Consulta la variable **synth** que contiene la referencia a todos los archivos de audio de nuestra base de datos.
- Crea dos variables aleatorias llamadas **G1** y **G2**, estas variables obtienen valores entre **G1start - G1end** y **G2start - G2end** que son pasados por referencia en la función. Dichas variables se utilizan para indicar la duración de la muestra de audio obtenida de **synth**. En concreto, este paso se encarga de construir los pequeños eventos acústicos o granos.

Para que el código no sea redundante, se ha creado un bucle *for* que se ejecuta tantas veces como pistas se le hayan pasado por argumento (**Ntracks**). En cada iteración, se ejecuta otro bucle *for* que rellena la pista actual en la que se encuentre de forma aleatoria desde **startMeasure** a **endMeasure** con muestras aleatorias (**sample**) de nuestra base de datos **synth**.

A continuación, se muestra el código descrito. Se puede observar que las variables aleatorias se obtienen con la función **random.uni** de Python, la cual genera un número decimal entre los dos valores pasados por referencia (**G1start - G1end** y **G2start - G2end**).

```
29 def granular_synth(startMeasure, endMeasure, Ntracks, Nsamples, G1start, G1end, G2start, G2end):
30
31     for track in range(1, Ntracks):
32
33         for duration in range(startMeasure, endMeasure):
34
35             sample = random.randint(0, Nsamples) # number of synth sound
36             G1 = random.uniform(G1start, G1end) # variable duration
37             G2 = random.uniform(G2start, G2end) # variable duration
38
39             fitMedia(synth[sample], track, (duration+G1), (duration + G2))
40
41     return sample
42
43
44
45 samples = granular_synth(1, 20, 11, 35, 0, 0.8, 0.3, 1)
46
```

Figura 26. Función **granular\_synth**.

En la siguiente captura de pantalla, se puede visualizar la ejecución de nuestro programa con los siguientes parámetros de entrada:

- **startMeasure: 1**
- **endMeasure: 20**
- **Ntracks: 11**
- **Nsamples,: 35**
- **G1start: 0**
- **G1end: 0.8**
- **G2start: 0.3**
- **G2end: 1**

Cabe destacar que cada vez que se ejecuta el código, los granos, su posición y su duración varían. Es decir, cada ejecución del código genera una composición distinta.

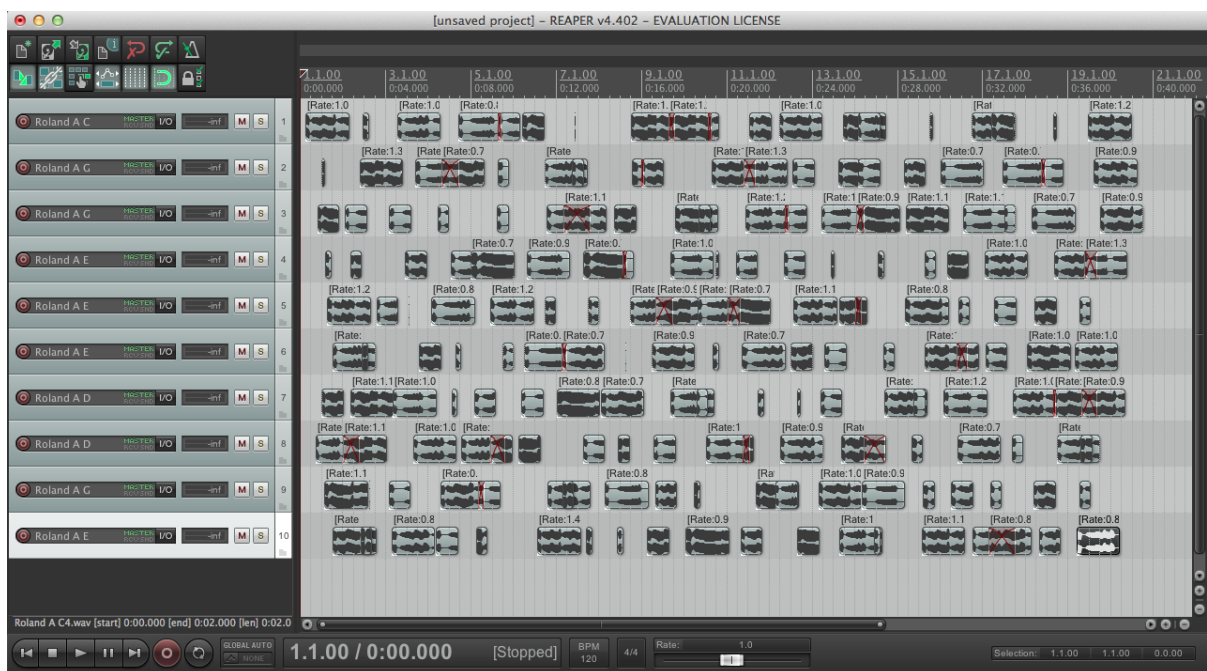


Figura 27. Resultado en Reaper™ de ejecución granular\_synth

#### ○ **Efectos:**

En esta sección se detallan los efectos que se han implementado en este proyecto, se ha implementado porque la mayoría de los sintetizadores granulares comerciales, tienen la posibilidad de introducir efectos.

En Earsketch™ hay varias formas de introducir efectos a las pistas:

### ➤ Efectos estáticos:

Se aplican a la pista que se desee y permanecen inalterables en toda ella, para las pistas 1, 2 y 3, se ha implantado el efecto DELAY del tipo DELAY\_FEEDBACK y con una intensidad de feedback de 5.

```
test_time.py (~/Desktop)
test_time.py x
47 #####
48 ### EFFECTS ###
49 #####
50
51 setEffect(1, DELAY, DELAY_FEEDBACK, 5)
52 setEffect(2, DELAY, DELAY_FEEDBACK, 5)
53 setEffect(3, DELAY, DELAY_FEEDBACK, 5)
54
55
```

Figura 28. Definición efectos estáticos

En la siguiente captura, se observan las pistas 1, 2 y 3 con el DELAY añadido en las pistas de efectos inferiores. Una de las utilidades de Reaper™ es que al añadir cualquiera de sus efectos a alguna pista nos permite modificar su intensidad simplemente modificando la envolvente de la pista de efectos.

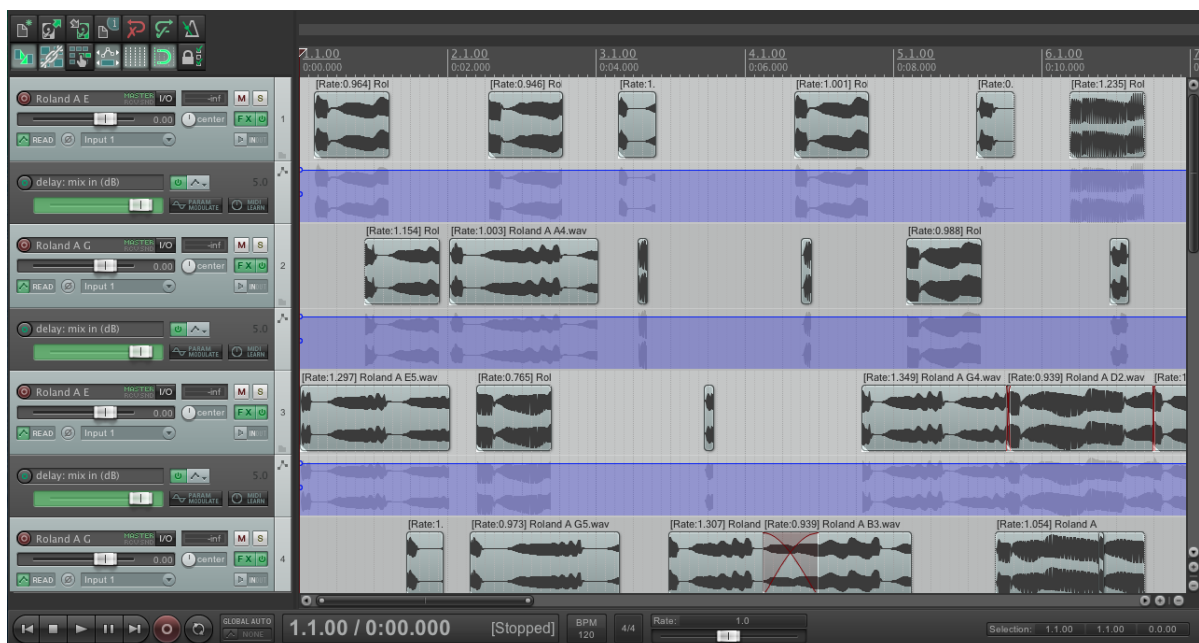


Figura 29. Resultado en Reaper™ de ejecución de efectos estáticos

### ➤ Efectos dinámicos o envolventes:

Este tipo de efectos varían con el tiempo, utilizan la misma función que los anteriores, pero se les añade tres variables más que permiten localizar la zona de actuación del mismo y definir la envolvente. A continuación se muestra un ejemplo genérico de la llamada a la función y los parámetros.

```
setEffect(trackNumber, FXName, parameter, effectStartValue, effectStartLocation, effectEndValue, effectEndLocation)
```

Para la pista 4, 5 y 6 se ha implementado un efecto de envolvente de ganancia, donde dicha ganancia varía en los instantes definidos effectStartLocation y effectEndLocation, y sus valores de ganancia van desde effectStartValue y effectEndValue.

Para la pista 7, se ha añadido un efecto de envolvente de delay que varía su retardo en los instantes effectStartLocation y effectEndLocation y sus valores de retardo van desde effectStartValue y effectEndValue.

Como se puede observar en la siguiente captura, la envolvente de dichos efectos varía de forma lineal y se puede construir a tramos. Esta función permite una gran flexibilidad a la hora de producir las pistas en zonas localizadas e individuales.

A continuación, se muestra la implementación de dichos efectos con los siguientes parámetros:

```
test_time.py *
50
51 setEffect(1, DELAY, DELAY_FEEDBACK, 5)
52 setEffect(2, DELAY, DELAY_FEEDBACK, 5)
53 setEffect(3, DELAY, DELAY_FEEDBACK, 5)
54
55
56 # Envelopes Effects
57 # envelope delay
58 setEffect(7, DELAY, DELAY_TIME, 1200, 1, 1400, 2)
59 setEffect(7, DELAY, DELAY_TIME, 1400, 4, 1200, 6.5)
60 setEffect(7, DELAY, DELAY_TIME, 2000, 7, 2200, 8.5)
61 setEffect(7, DELAY, DELAY_TIME, 0, 8.5, 0, 9)
62 setEffect(7, DELAY, DELAY_TIME, 1230, 10, 1230, 19)
63 # envelope gain for track 4
64 setEffect(4, VOLUME, GAIN, -20, 1, 0, 2)
65 setEffect(4, VOLUME, GAIN, 0, 4, -60, 6)
66 setEffect(4, VOLUME, GAIN, -50, 7, -10, 8)
67 setEffect(4, VOLUME, GAIN, -8, 9, 0, 15)
68 setEffect(4, VOLUME, GAIN, -5, 16, 0, 19)
69 # envelope gain for track 5
70 setEffect(5, VOLUME, GAIN, -20, 1, 0, 2)
71 setEffect(5, VOLUME, GAIN, 0, 4, -60, 6)
72 setEffect(5, VOLUME, GAIN, -50, 7, -10, 8)
73 setEffect(5, VOLUME, GAIN, -8, 9, 0, 15)
74 setEffect(5, VOLUME, GAIN, -5, 16, 0, 19)
75 # envelope gain for track 6
76 setEffect(6, VOLUME, GAIN, -20, 1, 0, 2)
77 setEffect(6, VOLUME, GAIN, 0, 4, -60, 6)
78 setEffect(6, VOLUME, GAIN, -50, 7, -10, 8)
79 setEffect(6, VOLUME, GAIN, -8, 9, 0, 15)
80 setEffect(6, VOLUME, GAIN, -5, 16, 0, 19)
81
```

Figura 30. Definición efectos dinámicos

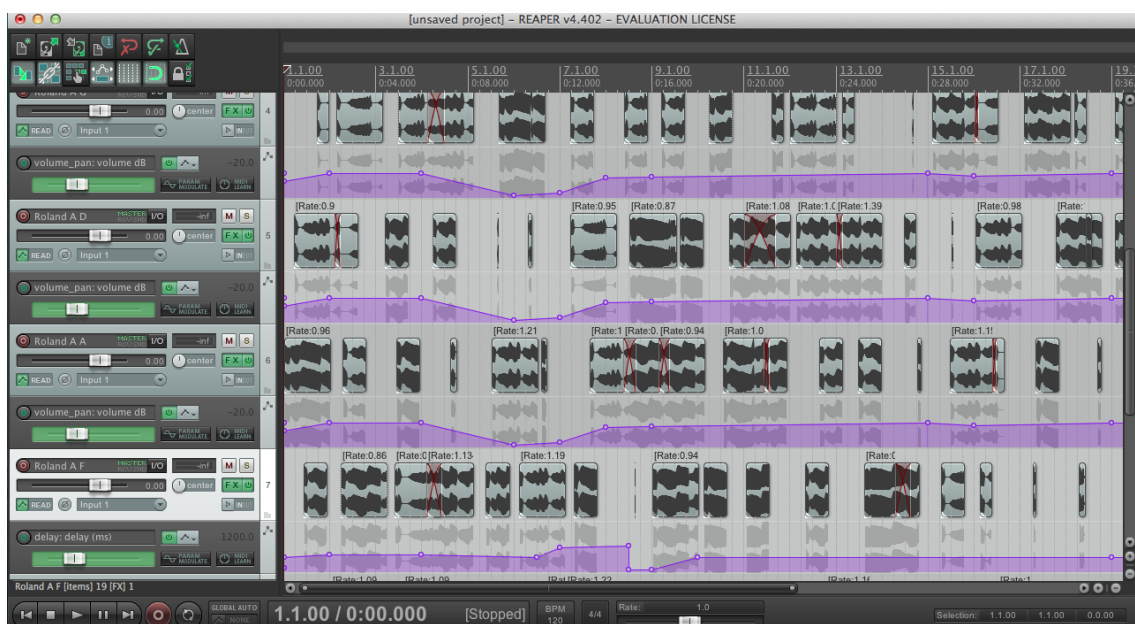


Figura 31. Resultado en Reaper™ de ejecución de efectos dinámicos



## ➤ Efectos rítmicos

Es otro tipo de implementación de efecto, generado por la función **rhythmeffects**:

```
rhythmEffect(trackNumber, FXFamily, FXName, timeRange, rhythmString)
```

La cual recibe como parámetros, la pista a la que se le aplica el efecto, la familia del efecto, el tipo de efecto de dicha familia, un patrón de variación, el rango donde se aplica y un patrón de ritmo.

```
83 # Rimth effect
84 pan_List = [0, -100, 100]
85 pan_String = '0+++0+++1+++2+0+'
86 for time in range(1,20):
87     rhythmEffects(10, PAN, LEFT_RIGHT, pan_List, time, pan_String)
88
89
```

Figura 32. Definición efectos rítmicos

Como se puede observar, para que se aplique en toda una región hemos de incluirlo en un bucle *for* que recorra toda esa región, la función **rhythmeffects** según el patrón de ritmo y el patrón de variación genera la forma del efecto.

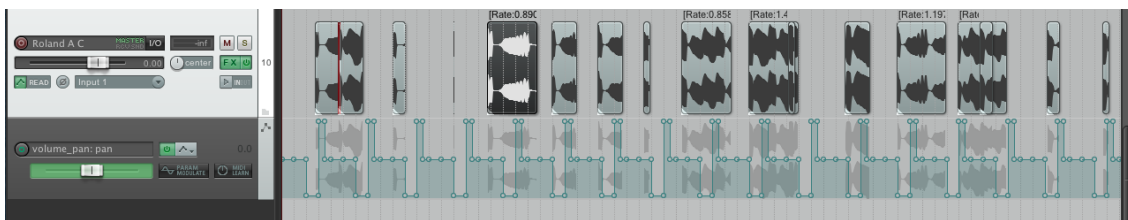


Figura 33. Resultado en Reaper™ de ejecución de efectos rítmicos

## 3. 4. Conclusiones

Como se puede apreciar, al ejecutar el programa del sintetizador, se obtienen los resultados esperados. Se genera una composición granular que con la función que se ha desarrollado para crear el sintetizador, nos da gran versatilidad y dinamismo para generar cualquier tipo de proyecto con los archivos que queramos y los parámetros requeridos. Dichos parámetros que se pueden variar en cada ejecución son:

- Ntrack :Número de pistas.
- Ng1: Número de elementos de la base de datos.
- duration\_end: Variable que pondera la duración de la composición.
- t\_start: Variable que indica la duración mínima de las muestras.
- t\_end: Variable que indica la duración máxima de las muestras.
- g2\_start : Variable que modifica la duración de las muestras.
- g2\_end: Variables que modifica la duración de las muestras.

## 4. SÍNTESIS SUSTRACTIVA AUDIO EN MATLAB™

---

### 4.1. Introducción

En los años 60 Robert Moog creó uno de los primeros sintetizadores analógicos de música utilizando moduladores de tensión, filtros paso bajo y modulación de envolvente. La creación y comercialización de este instrumento abrió las puertas al estudio e investigación de la síntesis musical [6].

En este apartado se abordará el estudio del modelo de síntesis sustractiva, y se proporcionará una aplicación que simule dicho modelo.

La síntesis sustractiva se basa en generar una señal de entrada mediante un oscilador y hacerla pasar por un filtro y un modulador de envolvente, de forma que se pueda alterar su timbre y dinámica. En el esquema de la figura 1 se muestran los diferentes componentes que se utilizan para realizar síntesis de sonido:

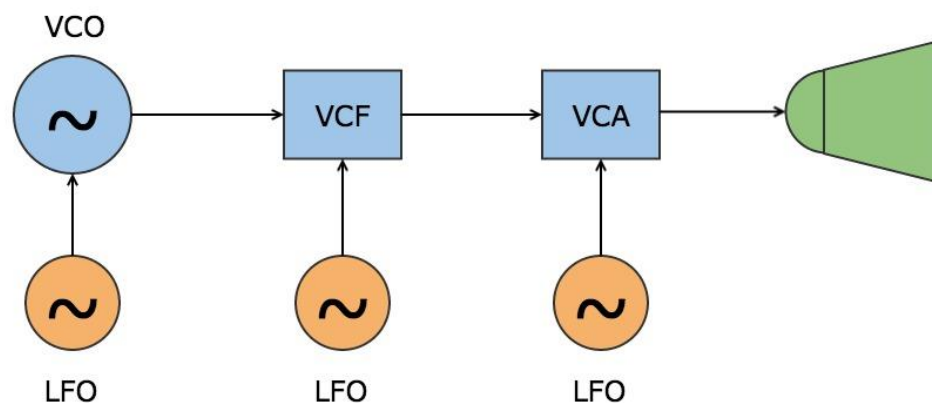


Figura 34. Diagrama de bloques de síntesis

- **VCO** (voltage controlled oscillator) Oscilador principal.
- **VCF** (voltage controlled filter) Filtro controlado por tensión.
- **VCA** (voltage controlled amplifier) Amplificador controlado por tensión o modulador de envolvente
- **LFO** (low-frequency oscillator) Oscilador de baja frecuencia.

Basándose en el modelo anterior de síntesis de sonido, se ha realizado una aplicación con la herramienta Matlab™ que sintetiza el sonido a partir de diferentes parámetros. Todo ello controlado por una interfaz gráfica implementada con la Guía de Matlab™. A continuación se muestra una figura con el entorno gráfico diseñado y utilizado:

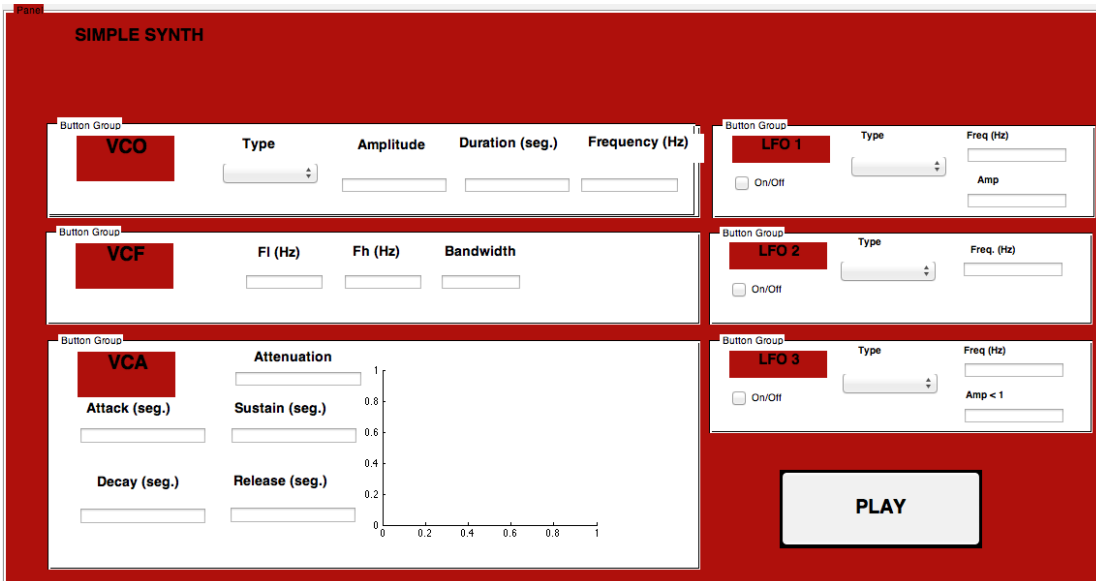


Figura 35. Entorno gráfico creado con GUIDE

En los siguientes apartados, se van a realizar análisis exhaustivos de los diferentes componentes que forman el sintetizador desarrollado y explicar su funcionamiento e implementación.

## 4.2. VCO (Voltage Controlled Oscillator) Oscilador principal

Es el primer componente que se encuentra a la entrada del sistema, genera una señal portadora que será modificada a lo largo de las diferentes etapas. Los parámetros que se pueden manipular de esta señal son la amplitud, su frecuencia y el tipo, como bien se muestran en la siguiente figura:

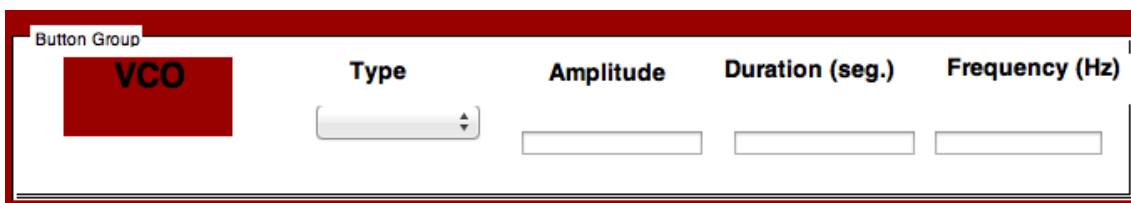
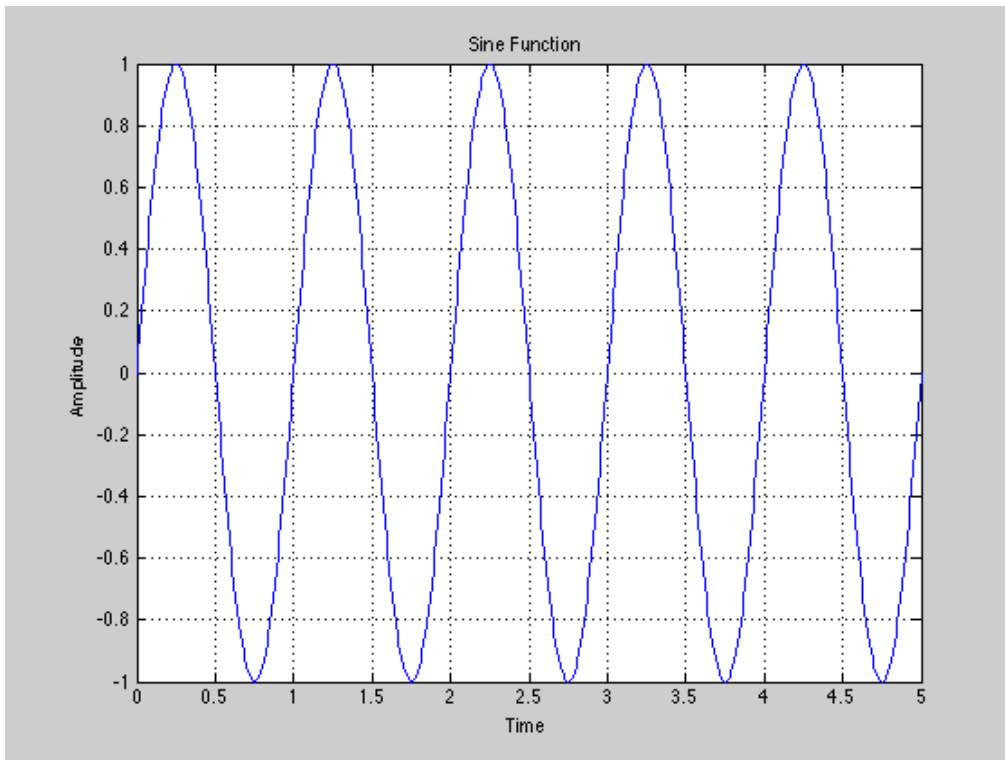


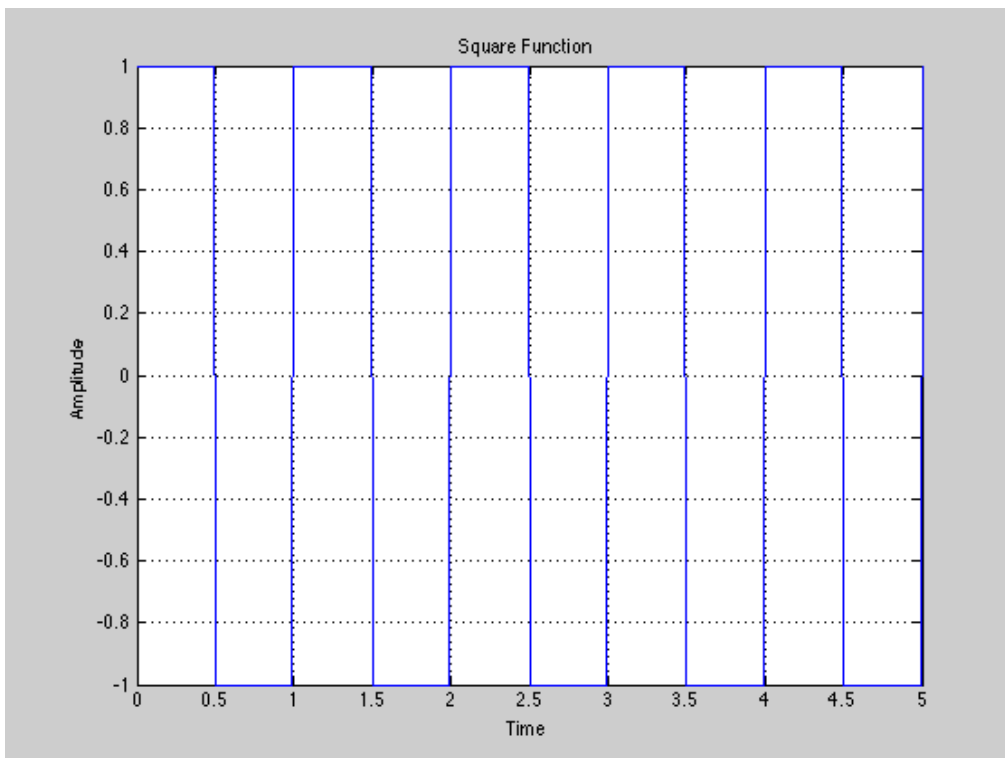
Figura 36. Entorno gráfico de la aplicación, control VCO

A continuación se muestran los diferentes tipos de señales que se pueden introducir en el VCO:

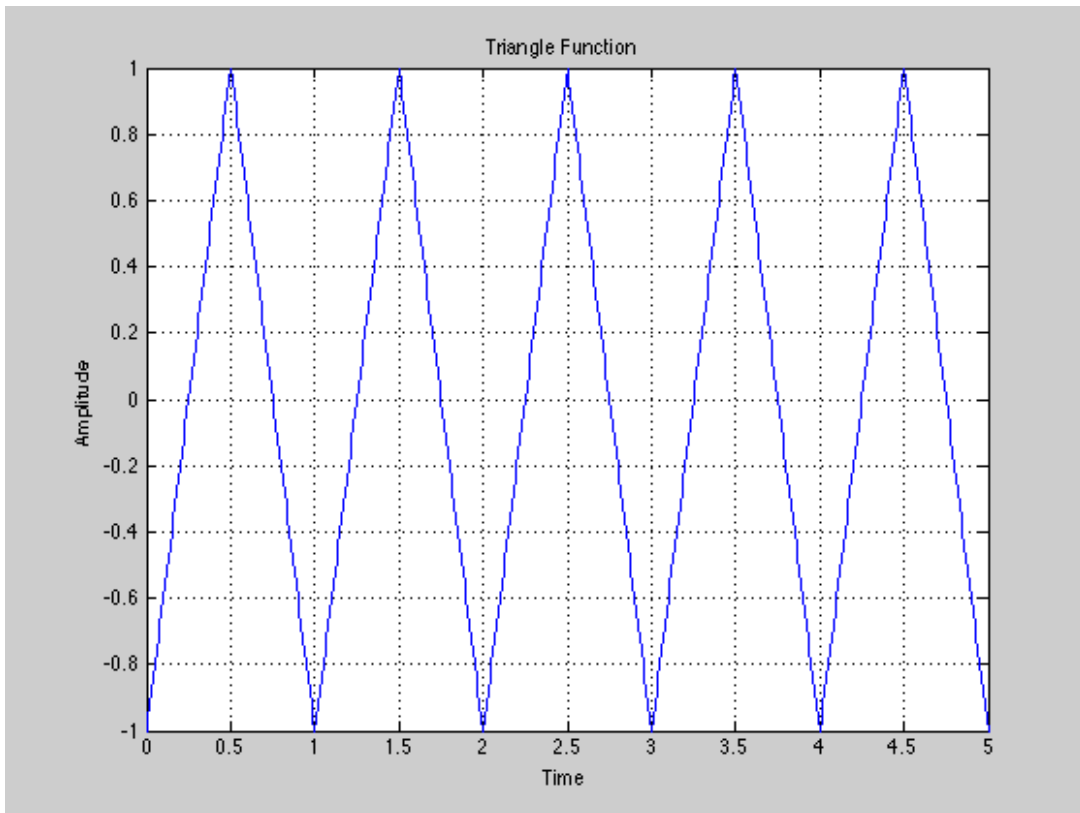
- Seno:



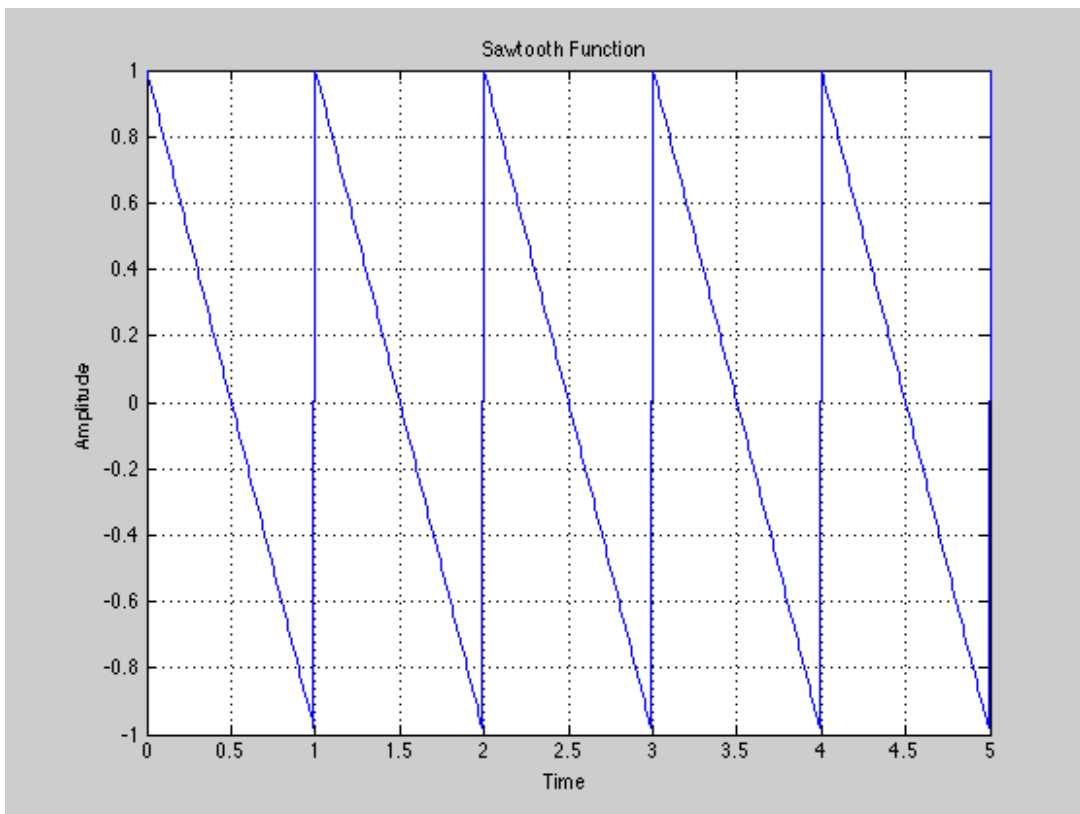
- Cuadrada:



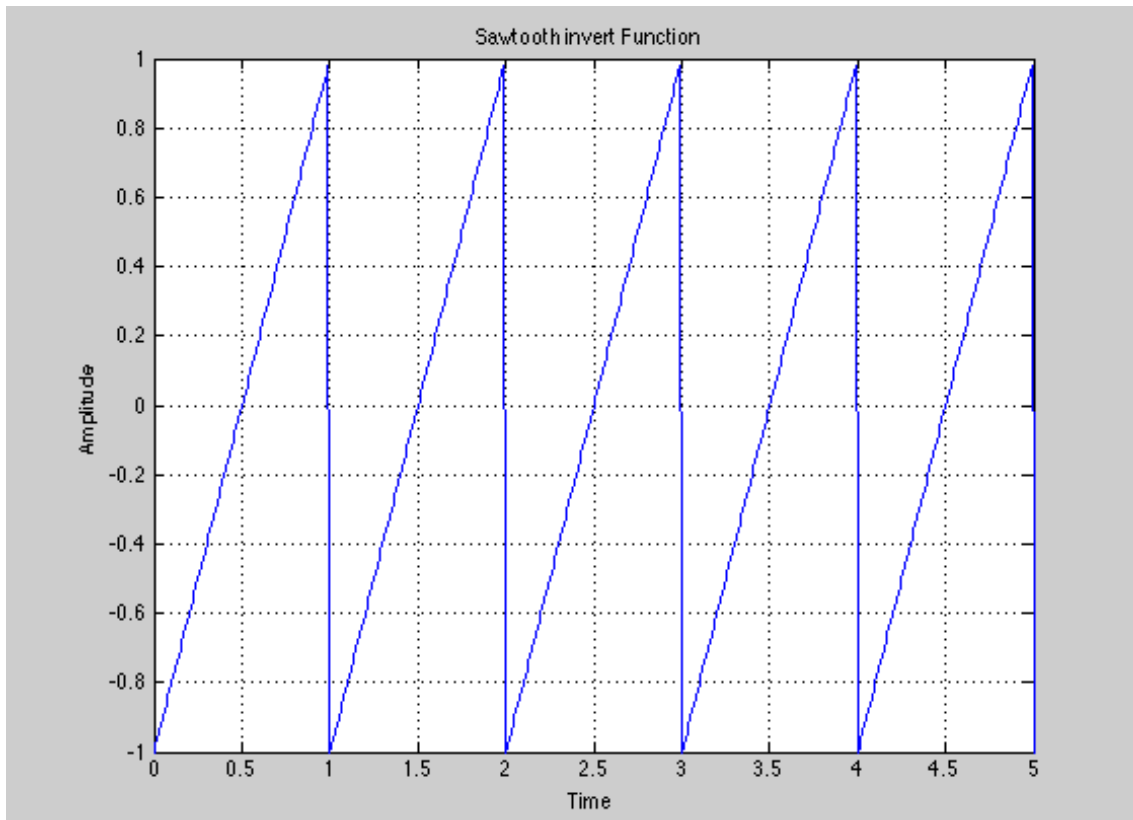
- Triangular:



- Diente de sierra:



- Diente de sierra invertido:



Hay que tener en cuenta que el contenido espectral de las diferentes señales varía, con lo cual, para apreciar los efectos generados por los diferentes componentes del sintetizador, es mejor utilizar señales 'ricas' en el dominio espectral para tener mayor número de armónicos. Como es el caso de las señales diente de sierra o triangular.

### 4.3. VCF (Voltaje Controlled Filter) Filtro controlado por tensión

Tras generarse la señal del VCO, esta es filtrada en el VCF (Voltaje Controlled Filter). Los parámetros que se pueden escoger de este filtro son:

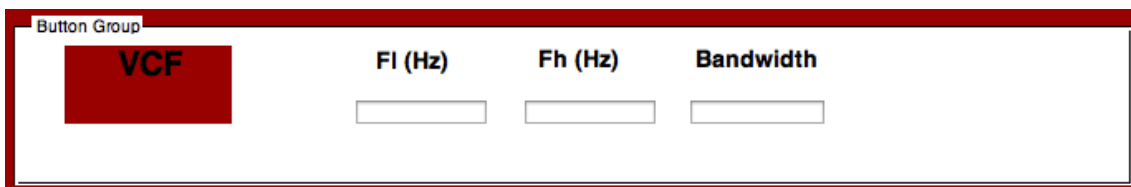


Figura 37. Entorno gráfico de la aplicación, control VCF

- Frecuencia baja del filtro ( $f_l$ )
- Frecuencia alta del filtro ( $f_h$ )
- Ancho de banda del filtro ( $B$ )

Se trata de un filtro de segundo orden con un pico de resonancia variante situado entre la frecuencia baja y la frecuencia alta. Su respuesta al impulso viene definida por la siguiente expresión canónica:

$$y[n] = x[n] + K_1 y[n - 1] - K_2 y[n - 2] \quad \text{(Fórmula 9)}$$

Las constantes tienen las siguientes expresiones:

$$K_1 = 2e^{\left(-\frac{2\pi B}{f_s}\right)} \cos\left(\frac{2\pi F(t)}{f_s}\right) \quad K_2 = e^{\left(-\frac{4\pi B}{f_s}\right)} \quad \text{(Fórmula 10)}$$

Donde F es:

$$F(t) = f_l + LFO2(t) \cdot (f_h - f_l) \quad \text{(Fórmula 11)}$$

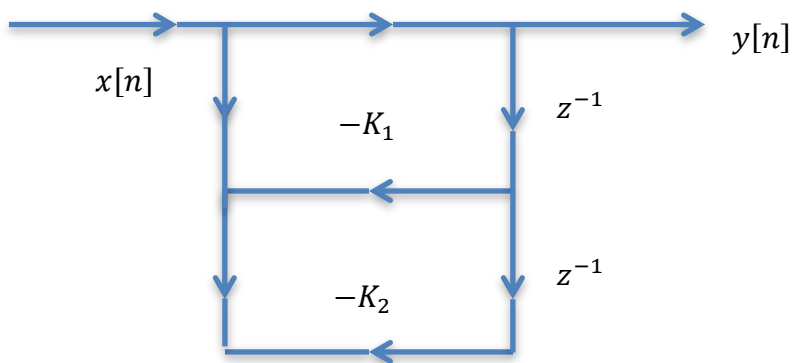
LFO(t) es la señal que introduce el oscilador de baja frecuencia. Por lo tanto, se observa un comportamiento dependiente con el tiempo, de tal forma que esta señal modifica el ancho de banda del filtro para instantes de tiempo determinados, alterando la posición del pico de resonancia del filtro, que más adelante se ilustrará.

Haciendo la transformada Z de la función anterior para obtener su respuesta y aplicando las propiedades adecuadas, queda de la siguiente forma:

$$Y(z) = X(z) + K_1 z^{-1} Y(z) - K_2 z^{-2} Y(z) \quad \text{(Fórmula 12)}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - K_1 z^{-1} + K_2 z^{-2}} = \frac{z^2}{z^2 - K_1 z + K_2} \quad \text{(Fórmula 13)}$$

Implementando la estructura del filtro se puede observar que es un filtro IIR con el parámetro  $K_1$  dependiente del tiempo.



Realizando un análisis en el dominio Z más profundo para observar el comportamiento del filtro de manera más visual, y ver como varía con respecto al tiempo, se han introducido los siguientes parámetros a la función que genera el VCF:

- Frecuencia baja ( $f_l$ ) : 8.000 Hz
- Frecuencia alta ( $f_h$ ) : 11.000 Hz
- Ancho de banda ( $B$ ) : 1.000 Hz
- $f_{lfo} = 2$  Hz
- $LFO(t) = \sin(2\pi \cdot t \cdot f_{lfo})$

A continuación se muestra el diagrama de polos y ceros para los instantes de tiempo de 0 a 1 segundo en pasos de 0.2:



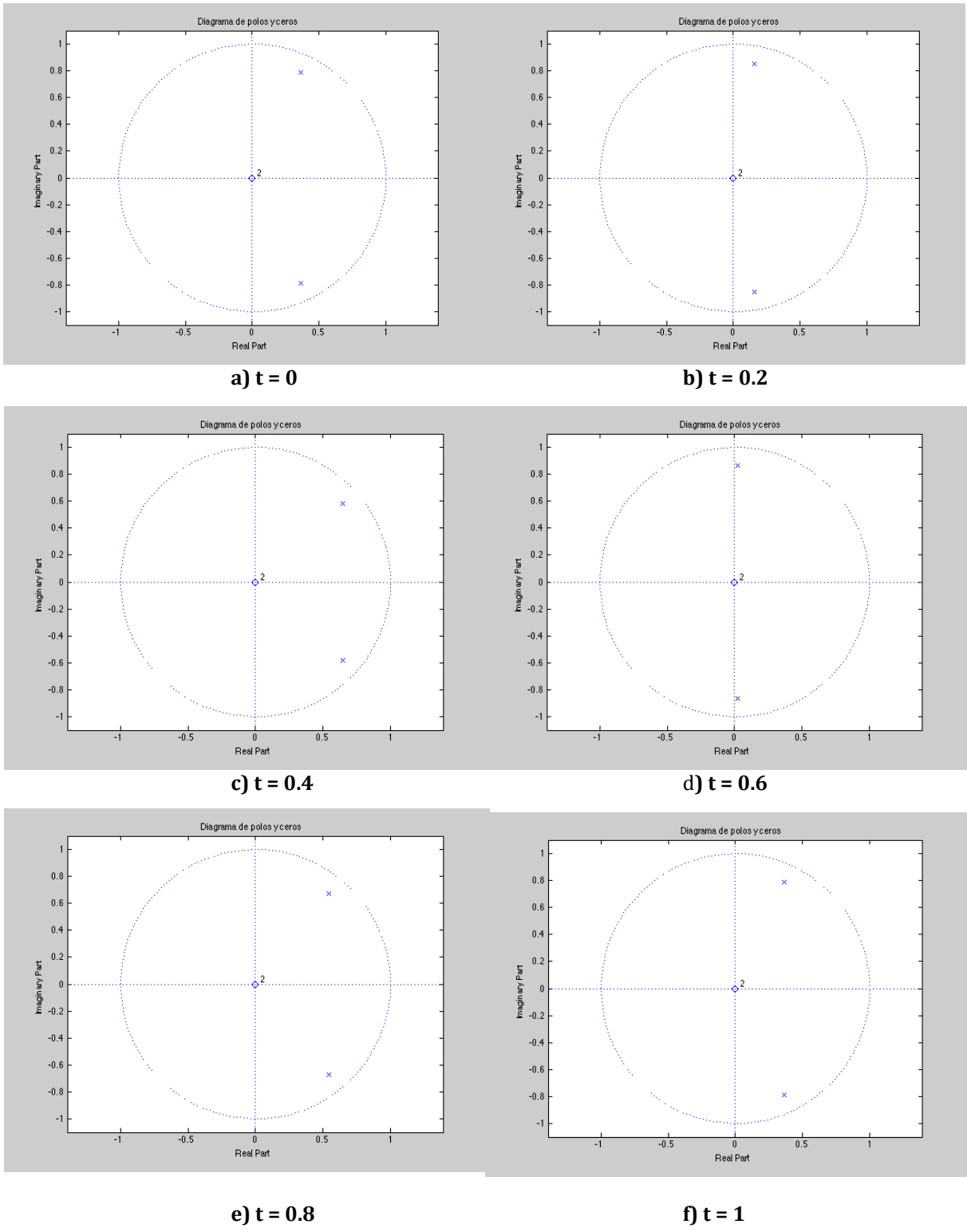


Figura 38. Diagrama de polos y ceros en diferentes instantes de tiempo

Como es de esperar, los polos de la función son simétricos con respecto al eje real y hay un cero doble en el origen.

Conforme avanza el tiempo, en instantes determinados los polos están más cerca del eje imaginario o del real, esto se debe a la constante  $K_1$  que depende del tiempo y varía la parte imaginaria del VCF.

Realizando un análisis en el dominio de la frecuencia para ver la forma del filtro, se han seguido los siguientes pasos:

- Señal a la entrada: respuesta al impulso  $x(t) = \delta(t)$
- Frecuencia baja ( $f_l$ ) : 8.000 Hz
- Frecuencia alta ( $f_h$ ) : 11.000 Hz
- Ancho de banda ( $B$ ) : 1.000 Hz
- $f_{lfo} = 1$  Hz
- $LFO(t) = \sin(2\pi \cdot t \cdot f_{lfo})$

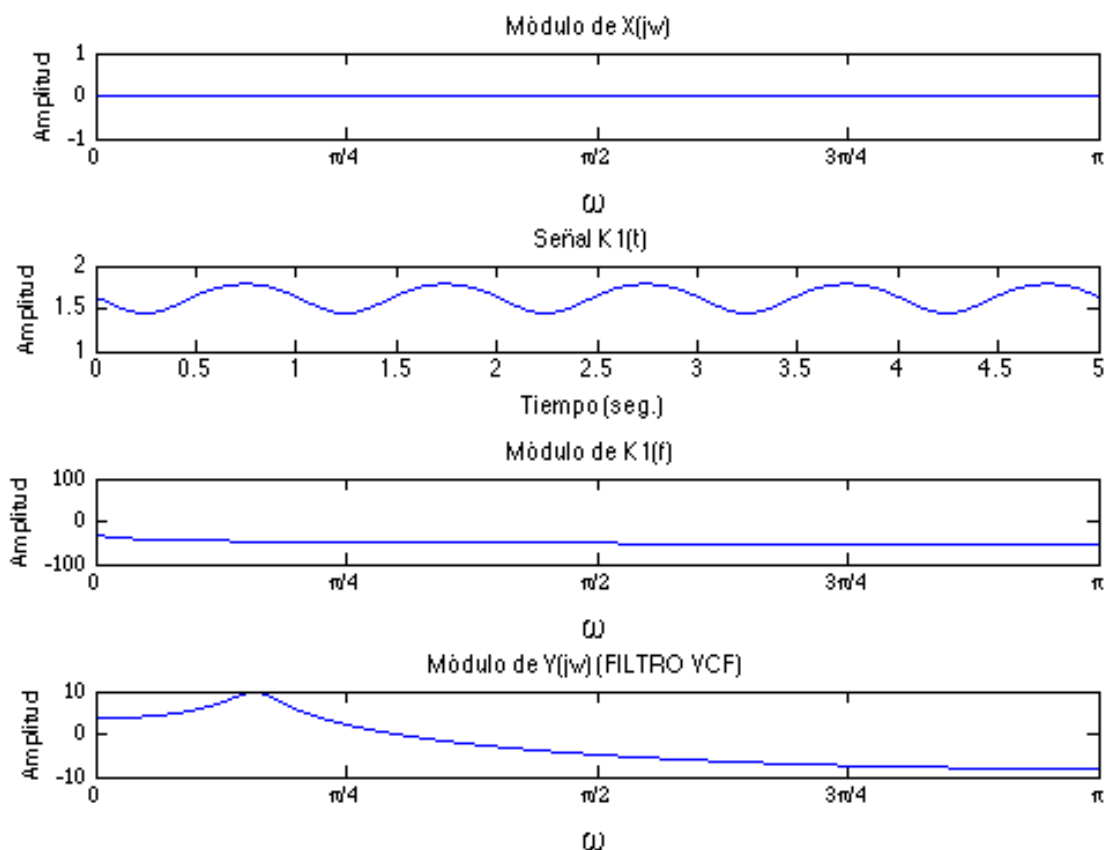


Figura 39. Análisis filtro VCF, señal  $X(j\omega)$ ,  $K_1(t)$ ,  $K_1(j\omega)$  e  $Y(j\omega)$

A continuación se muestra la transformada de Fourier de la señal  $y(t)$ . Esta señal coincide con la respuesta al impulso del sistema ya que a la entrada se ha introducido la delta.

Se puede observar con mayor claridad el pico de resonancia del filtro centrado a 8000 Hz.

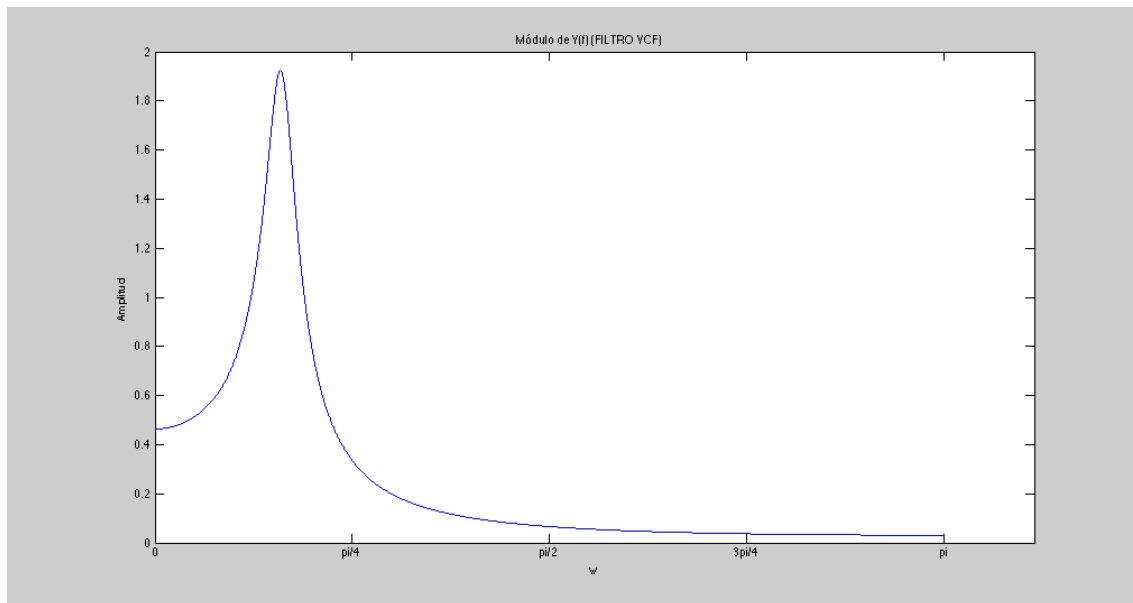


Figura 40. Filtro VFC

#### 4.4. VCA. (Voltage Controlled Amplifier) Amplificador controlado por tensión, o modulador de envolvente

Al obtener la señal de salida del filtro VCF, esta se pasa por un modulador de envolvente del tipo ADSR (Attack, Decay, Sustain y Release).

El modulador de envolvente ADSR utilizado imita la envolvente de amplitud de los instrumentos musicales acústicos. El volumen y el contenido espectral del sonido cambia con el tiempo, este efecto se puede modelar con los siguientes parámetros característicos:

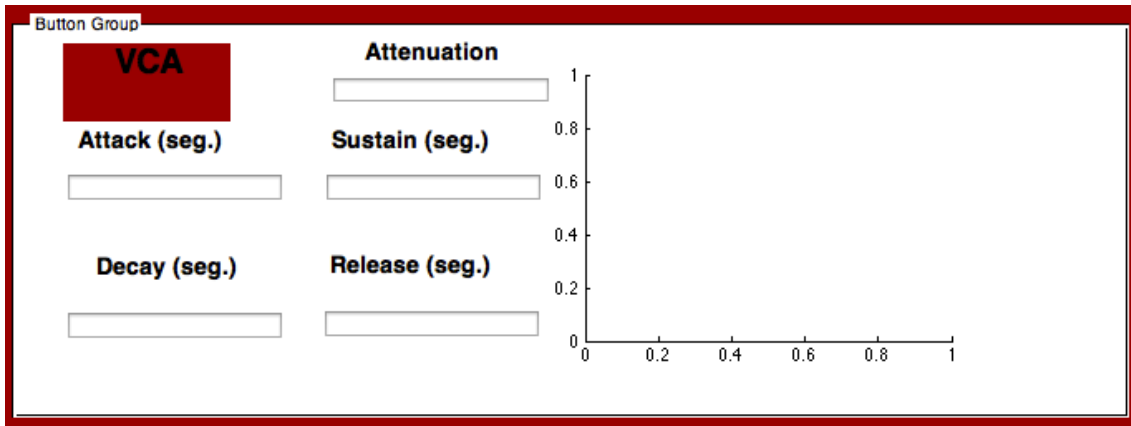


Figura 41. Entorno gráfico de la aplicación, control VCA

- **Tiempo de ataque (Attack):** es el tiempo que tarda el sonido en ir desde el nivel mínimo al máximo.
- **Tiempo de decaimiento (Decay):** es el tiempo necesario de estabilización para el posterior funcionamiento o nivel de sostenimiento.
- **Tiempo de sostenimiento (Sustain):** es el nivel de secuencia principal de la duración del sonido.
- **Tiempo de salida (Release):** es el tiempo necesario para que el nivel de sostenimiento decrezca hasta llegar al nivel mínimo.

A continuación se muestra una gráfica con la representación mas significativa de un modulador de envolvente:

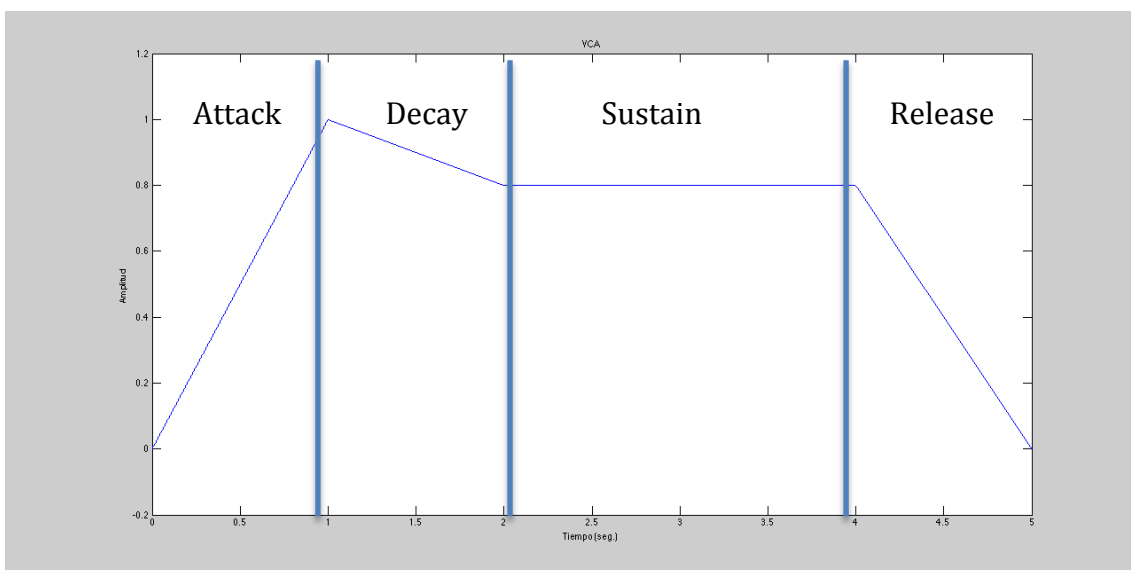


Figura 42. Gráfica estándar del modulador de envolvente en Matlab™

En el entorno gráfico diseñado, se pueden escoger todos los parámetros anteriormente descritos del modulador de envolvente, siempre y cuando el tiempo del modulador sea de la misma duración que la señal generada en el VCO. En caso contrario, el programa no utilizara ningún modulador de envolvente.

Otro parámetro que se debe especificar en el modulador de envolvente, para facilitar su diseño, es la amplitud de atenuación del *sustain*, llamado en el entorno gráfico **Attenuation**. Su valor está definido entre 0 y 1. Esta amplitud se aplica a los instantes determinados de tiempo en los cuales se aplica el *sustain*.

Este sería el último bloque del sistema, la señal de entrada generada por el VCO, sin ningún tipo de oscilador de baja frecuencia (LFO) activo, quedaría filtrada por el VCF y modelada por el VCA.

A continuación, se analizará el comportamiento de los osciladores en el sintetizador y las posibles funciones dentro del sistema.

## 4.5. LFO (Low-Frequency Oscillator) Oscilador de baja frecuencia

El LFO u oscilador de baja frecuencia que se ha diseñado en la aplicación ha de trabajar en frecuencias por debajo de 20 Hz. Las señales que pueden generarse son de tipo:

- Seno.
- Rectangular.
- Triangulo.
- Diente sierra.
- Diente sierra inverso.

Este componente se utiliza para estimular los tres bloques principales del sintetizador (VCO, VCF y VCA) y generar diferentes efectos. Se ha introducido la posibilidad de que el usuario pueda escoger la actuación de los diferentes LFOs en el sistema.

- LFO1 actúa sobre el VCO.
- LFO2 actúa sobre el VCF.
- LFO3 actúa sobre el VCA.

Los parámetros que se pueden modificar en los LFOs son: activación o desactivación del oscilador, tipo de señal, amplitud y frecuencia.

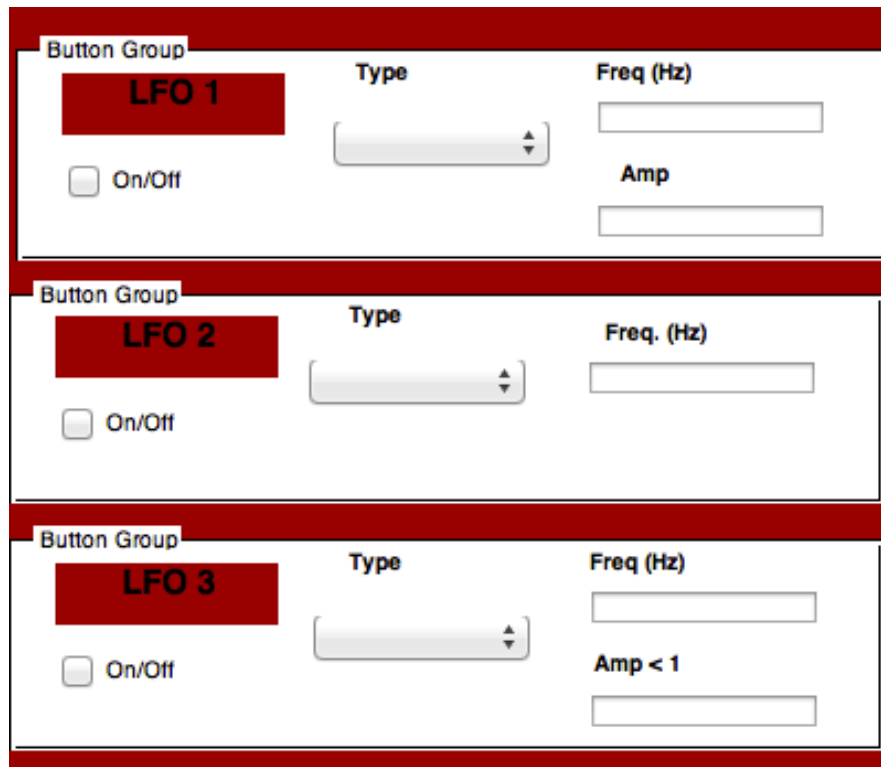


Figura 43. Entorno gráfico de la aplicación, control LFOs

#### 4.5.1. LFO 1 actúa sobre el VCO

Cuando se aplica el oscilador de baja frecuencia al VCO se genera una modulación FM, siendo la señal del LFO1 la moduladora y la señal del VCO la señal portadora. A continuación se muestra el desarrollo matemático:

$$VCO(t) = A_{vco} \cdot \sin(2\pi \cdot t \cdot f_{vco}) \quad LFO(t) = A_{lfo} \cdot \sin(2\pi \cdot t \cdot f_{lfo}) \quad (\text{Fórmula 14})$$

$$y(t) = A_{vco} \cdot \sin\left(2\pi \cdot t \cdot f_{vco} + 2\pi \cdot f_D \int_{-\infty}^t S_N(t) dt\right) \quad (\text{Fórmula 15})$$

$f_D$  es la constante de máxima desviación de frecuencia

La desviación de frecuencia viene dada por:

$$f_d(t) = \frac{1}{2\pi} \frac{d\rho(t)}{dt} \quad \rho(t) = 2\pi \int_{-\infty}^t f_d(t) dt = 2\pi \int_{-\infty}^t f_D S_N(t) dt \quad (\text{Fórmula 16})$$

Dónde la señal  $S_N(t)$  es la señal LFO(t) normalizada.

Para realizar la modulación FM de la señal generada por el VCO se ha utilizado el siguiente código que simplifica los cálculos anteriores, sustituyendo la integral infinita por un sumatorio finito que aproxima el resultado, ya que no se puede analizar la señal desde menos infinito:

```
%% MODULACION FM
Fs=44100; %frecuencia de muestreo
inct=1/Fs;
Am=1.1; %Amplitud de la seÑal modulada
Ac=1.2; %Amplitud de la seÑal portadora
fm=2; %Frecuencia de modulación
fc=200; %Frecuencia de portadora
fd=50; %Frecuencia de desviación
T=5; %Duración de la seÑal
N=T/inct;
t = [0:1/Fs:T-1/Fs];
f = [-Fs/2:Fs/N:Fs/2 - Fs/N];

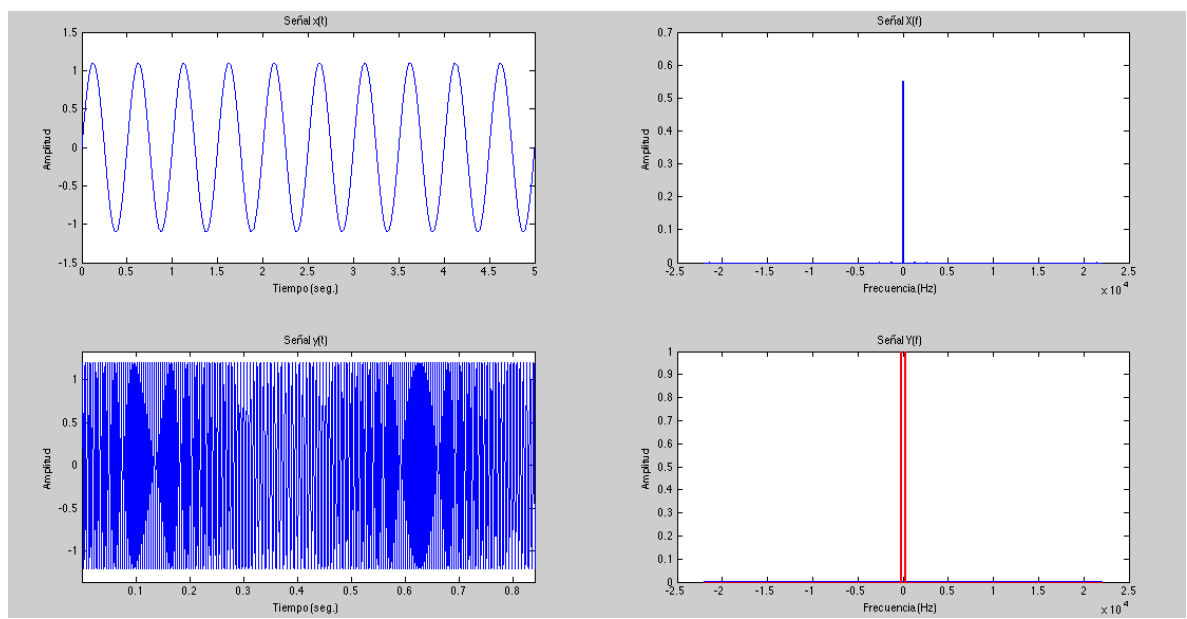
x=Am*sin(2*pi*fm*t);

vc=Ac*sin(2*pi*fc*t);

xn=x./(max(abs(x)));

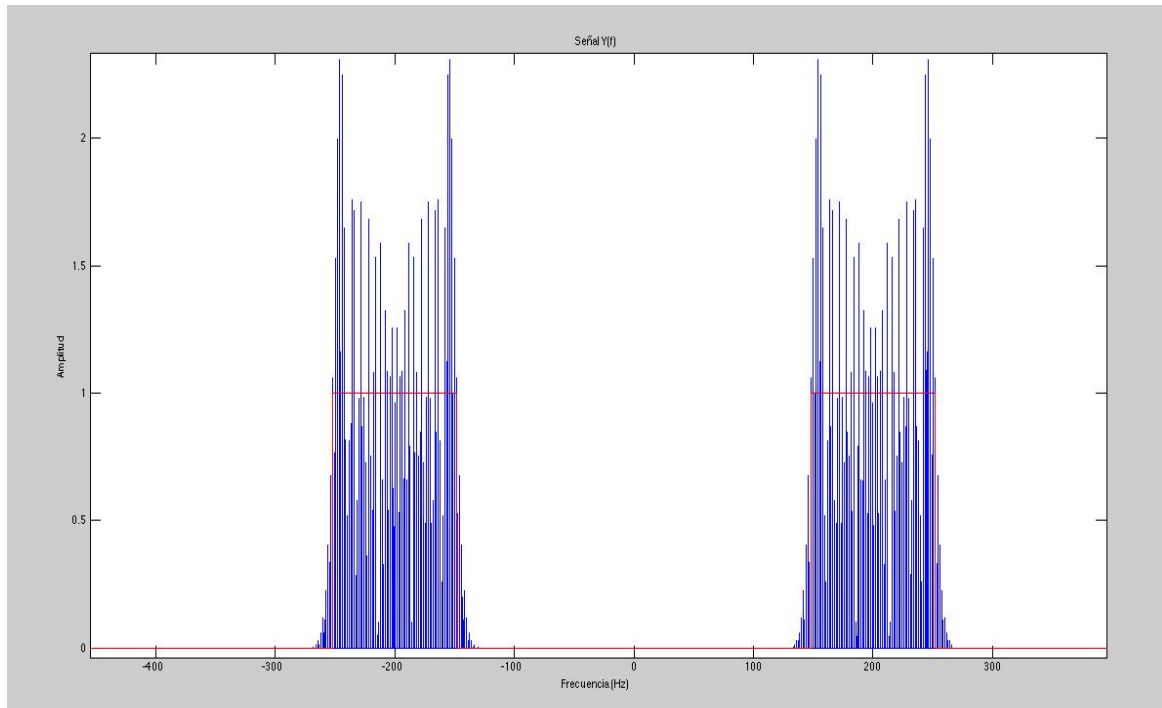
y=Ac*sin((2*pi*fc*t)+(2*pi*fd*(cumsum(xn*inct))));

%FFT de la seÑal original y la seÑal modulada
Xf=fftshift(inct*fft(x));
Yf=fftshift(inct*fft(y));
```



**Figura 44. Señal procedente del LFO en tiempo y frecuencia (gráficas de arriba), seÑale a la salida del VCF en tiempo y frecuencia (gráficas de abajo).**

Si se utilizan señales para el VCO en torno a 20Hz-20000Hz y en el LFO señales con muy baja frecuencia. Se aprecia el efecto **vibrato**. Este efecto se basa en la variación periódica de la frecuencia de la señal producida por el VCO, en definitiva, es una modulación FM de baja frecuencia de la señal producida por el VCO. En la figura 51, señal ampliada a la salida del VCF en frecuencia, se puede observar este efecto.



**Figura 45. Señal ampliada en frecuencia a la salida del VCF.**

#### 4.5.2. LFO 2 actúa sobre el VCF

Al combinar el LFO con el filtro controlado por tensión, se genera un cambio en el timbre de la señal, esto sucede porque se está modificando el ancho de banda y la frecuencia de resonancia del filtro de segundo orden utilizado. Como se ha apreciado anteriormente el filtro es paso-bajo con una resonancia de pico (ver figura 45), si se introduce una señal moduladora mediante el LFO2 , el ancho de banda del pico de resonancia varía y produce el efecto llamado **wah-wah**. Este efecto se define como el cambio de un tono grave a otro agudo pasando por los intermedios.

A continuación se muestra el código de Matlab™ que implementa el desarrollo matemático del apartado VCF (fórmulas 9, 10 y 11). Se ha utilizado una función, llamada `vcf_gen`, de la librería **Audio Toolkit** [9] que realiza todo el desarrollo matemático del filtro IIR:

```
function y = vcf_gen( x , ctrl , fl , fh , d , sr )
% Applies a control signal to a resonant filter to continuously update its
% center frequency.
%
```



```

% y = vcf_gen( x , ctrl , fl , fh , d , sr )
%
% Inputs:
% x = input signal
% ctrl = control signal (row vector, [ 0 , 1 ])
% fl = lowest center frequency of the filter in Hz.
% fh = highest center frequency of the filter in Hz.
% d = half-bandwidth of the filter in Hz. (100 Hz. typical)
% sr = sample rate in samples per second
%
% Output:
% y = vector of samples of the input signal filtered as specified
n = length( x ) ;

F = fl + ctrl.*( fh - fl ) ;
I2 = 2 * exp(-2*pi*d/sr) * cos(2*pi*F/sr) ;
I3 = exp(-4*pi*d/sr) ;
y = zeros(1,n) ;
prev1 = 0 ;
prev2 = 0 ;
for i = 1:n
    y(i) = x(i) + I2(i) * prev1 - I3 * prev2 ;
    prev2 = prev1 ;
    prev1 = y(i) ;
end
y = y / max(abs(y)) ;

```

end

Introduciendo los siguientes valores en el entorno gráfico (Figura 46), obtenemos las siguientes graficas que muestran de manera más visual el efecto **wah-wah**. En concreto en la gráfica de la figura 48, en la cual se ve en el dominio del tiempo que la señal original, de tipo diente de sierra (Figura 47), ha sido modificada por la acción del VCF con el LFO:

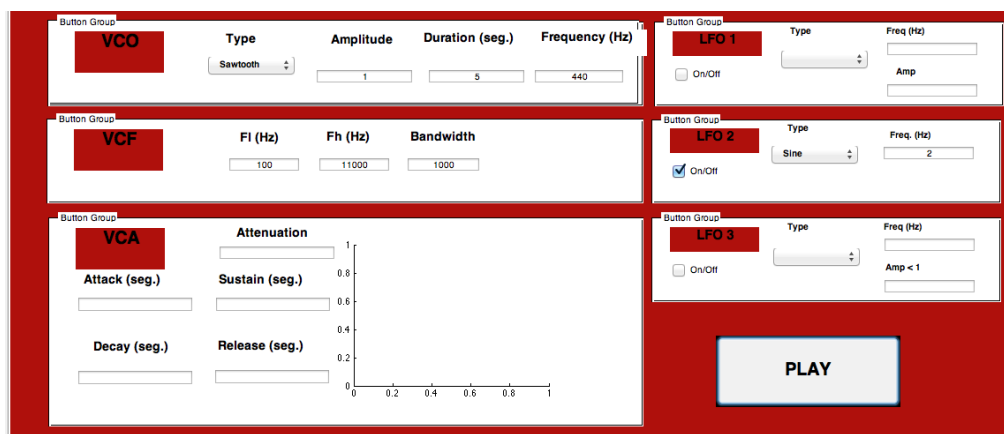


Figura 46. Parámetros introducidos.

VCO -> Type : Sawtooth, Amplitude: 1, Duration: 5 seg., Frequency: 440 Hz.

VCF -> Fl: 100 Hz, Fh: 11000 Hz, Bandwidth: 1000 Hz.

LFO 2 -> ON, Type: Sine, Frequency: 2 Hz.

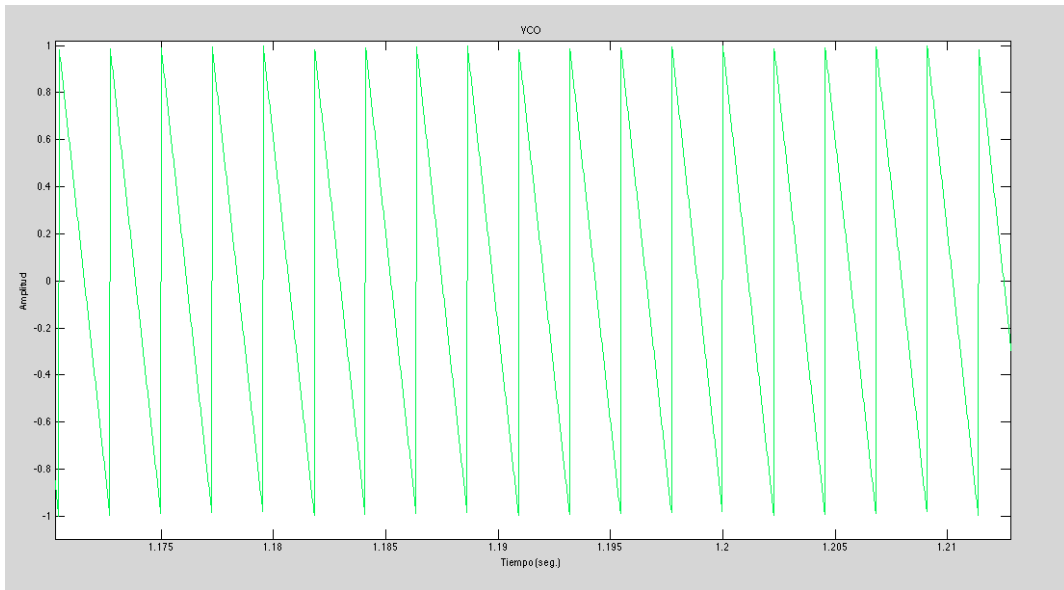


Figura 47. Señal diente de sierra generada por el VCO

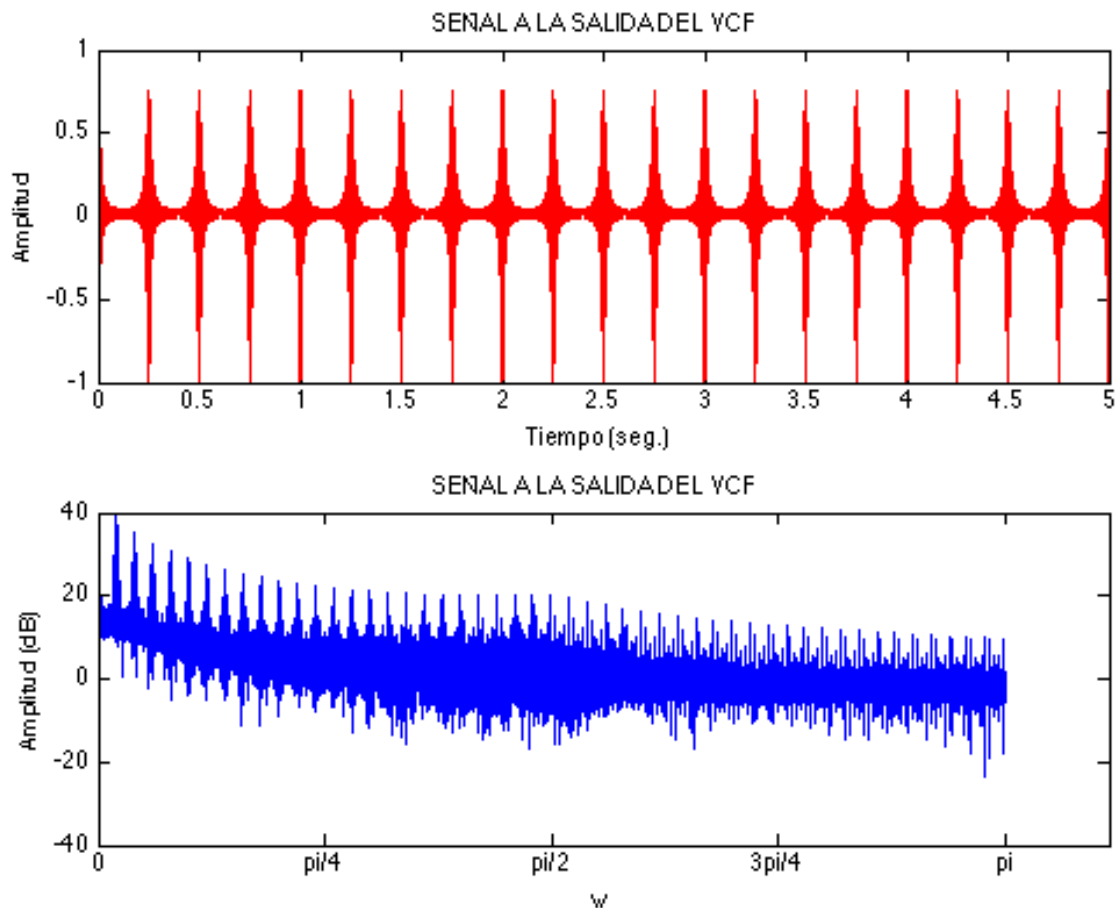


Figura 48. Señal a la salida del módulo VCF

A continuación, se muestra la señal a la salida del VCF en tiempo y ampliada, para observar con detalle el efecto producido por el oscilador LFO 2. Se produce un filtrado de la señal original por el VCF, variando su pico de resonancia.

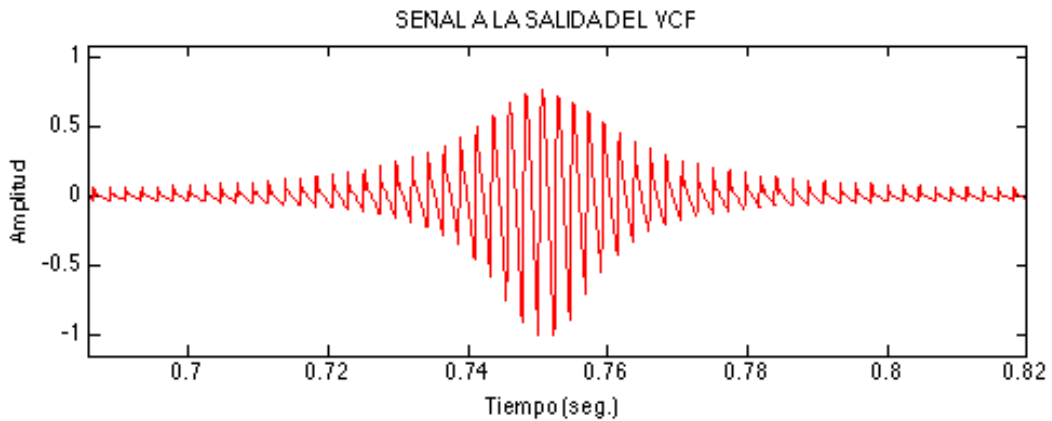


Figura 49. Señal ampliada en tiempo a la salida del módulo VCF

Si comparamos la señal sin el LFO 2 activo, observamos que la señal en el tiempo ha sido filtrada, pero no ha sufrido tantas modificaciones que con el LFO 2 activo. Ya que al estar el filtro alterando su pico de resonancia con el tiempo, en algunos instantes está resonando sobre zonas espectrales de mucha energía, y en otros instantes en zonas del espectro del diente de sierra con baja energía.

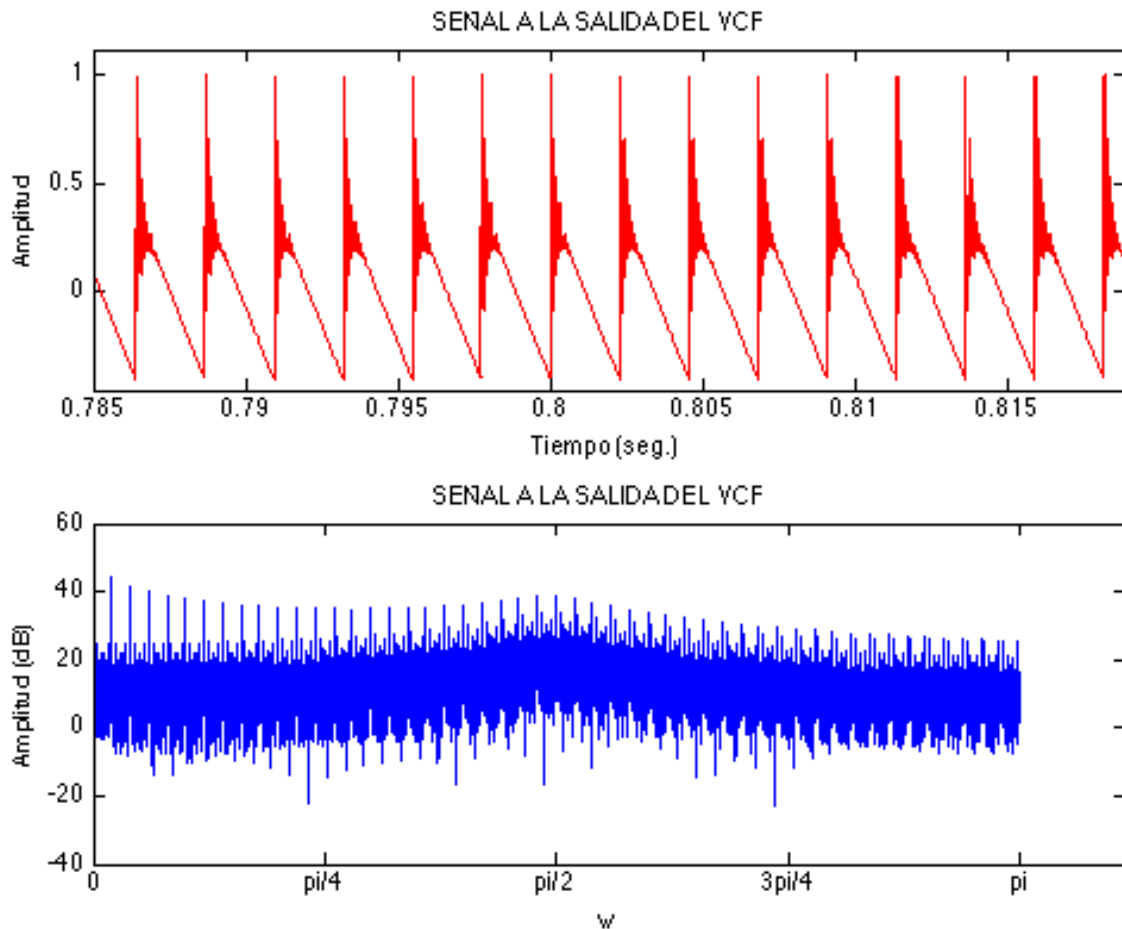


Figura 50. Señal a la salida del módulo VCF sin el LFO 2

### 4.5.3. LFO 3 actúa sobre el VCA

Si se multiplica la señal del LFO con la respuesta de la envolvente, se produce en denominado efecto **tremolo**. Es un término musical que describe la fluctuación o variación periódica en la intensidad (amplitud o volumen) de un sonido, mientras se siguen manteniendo el resto de propiedades del sonido. Si se aumenta la amplitud de la moduladora, el efecto tremolo es más aparente.

El código en Matlab™ utilizado para la combinación del VCA y el LFO3, es simplemente una multiplicación de la señal procedente del LFO3 con el modulador de envolvente, generado por el VCA.

```
%% LFO 3 (VCA)

if lfo3_on == 1

    if lfo3_type == 1    %seno
        lfo3 = (1-lfo3_amp)*sin(2*pi*t*lfo3_fr);
    elseif lfo3_type == 2    %triangular
        lfo3 = (1-lfo3_amp)*sawtooth(2*pi*t*lfo3_fr, 0.5);
    elseif lfo3_type == 3    %cuadrada
        lfo3 = (1-lfo3_amp)*square(2*pi*t*lfo3_fr);
    elseif lfo3_type == 4    %diente sierra
        lfo3 = (1-lfo3_amp)*sawtooth(2*pi*t*lfo3_fr, 0);
    elseif lfo3_type == 5    %diente sierra invertido
        lfo3 = (1-lfo3_amp)*sawtooth(2*pi*t*lfo3_fr, 1);
    end;

    y = vca.*lfo3.*vcf_y;

else
    lfo3 = ones(1,length(t));
    y = vca.*vcf_y;
end
```

En las siguientes gráficas se muestran los comportamientos del VCA cuando se modula con una señal de tipo seno, con una frecuencia de 1 Hz y 0.9 de amplitud. Se han utilizado los siguientes parámetros para el sistema:

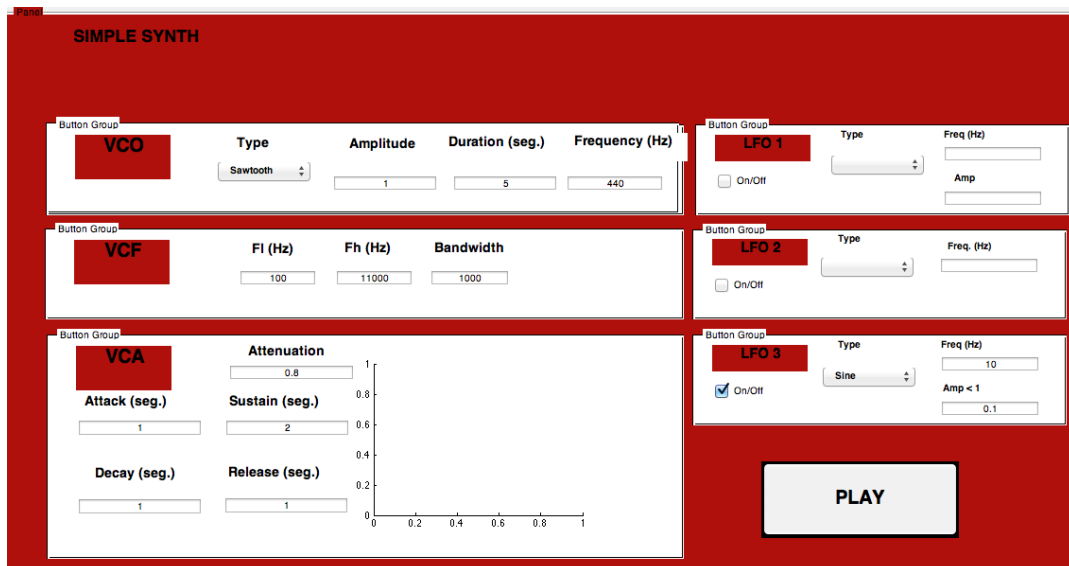


Figura 51. Parámetros introducidos.

VCO -> Type : Sawtooth, Amplitude: 1, Duration: 5 seg., Frequency: 440 Hz.

VCF -> F1: 100 Hz, Fh: 11000 Hz, Bandwidth: 1000 Hz.

VCA -> Attenuation: 0.8, Attack: 1 seg., Sustain: 2seg., Decay: 1 seg., Release: 1 seg.

LFO 3 -> ON, Type: Sine, Frequency: 10 Hz, Amplitude: 0.1.

Se ha utilizado una señal de tipo sawtooth en el VCO, porque son más ricas en componente frecuencial y así observar con más precisión los efectos del LFO3.

A continuación, se muestra la respuesta del modulador de envolvente con respecto al tiempo, y la señal a la salida del sistema en el dominio del tiempo y de la frecuencia:

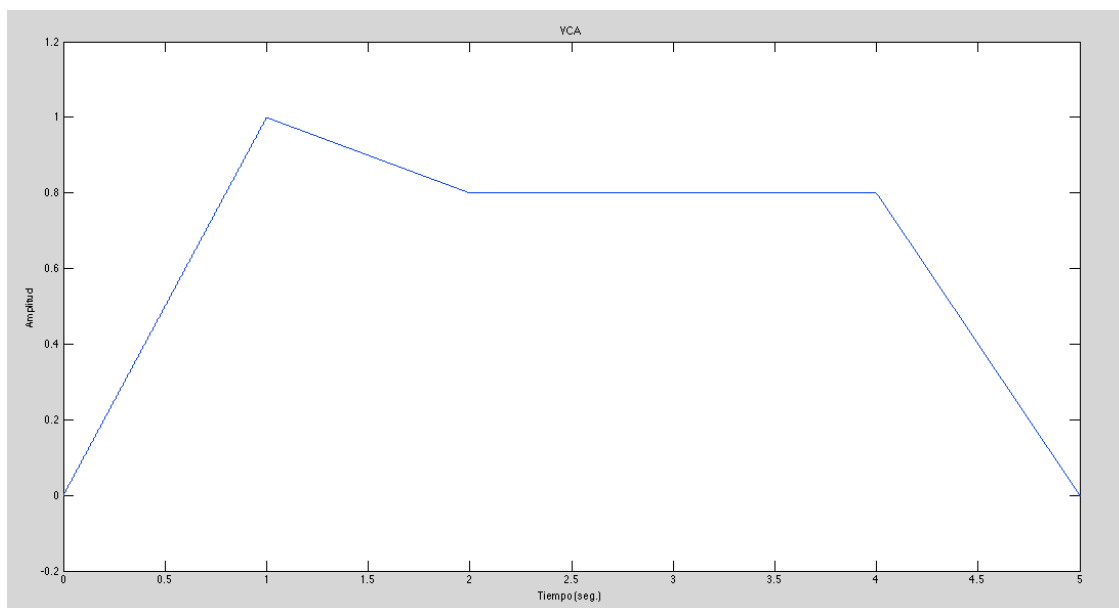
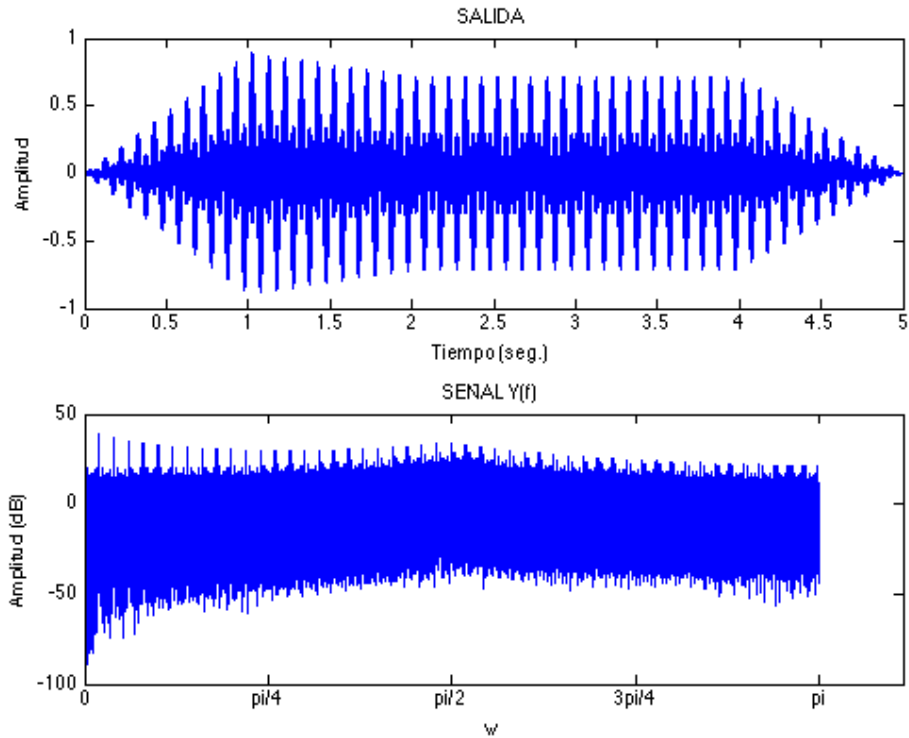
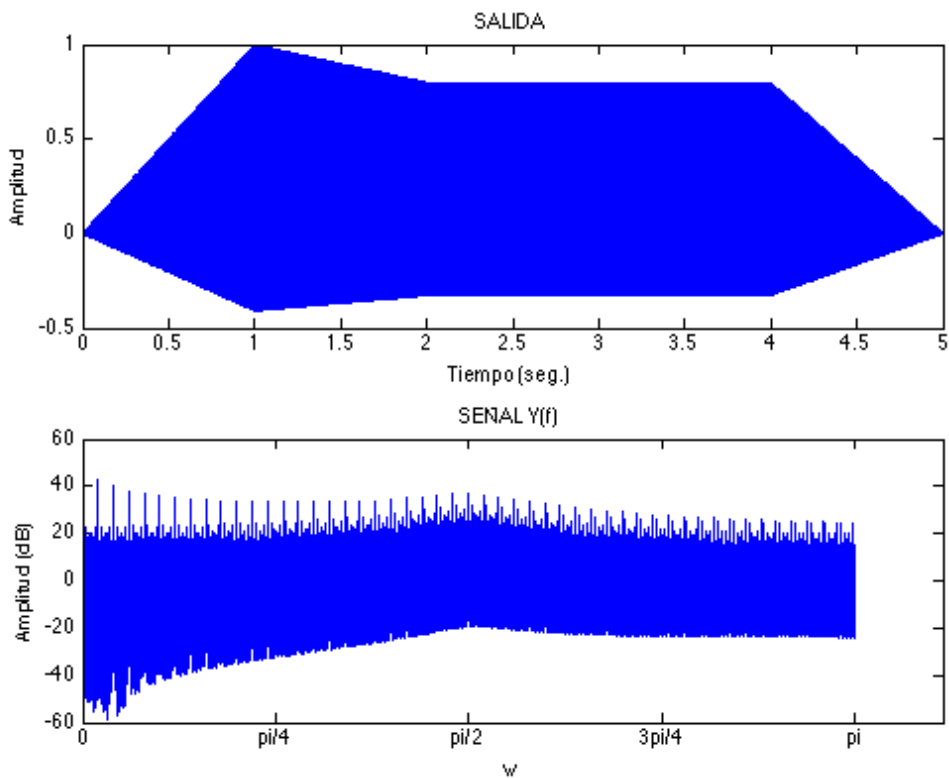


Figura 52. Modulador de envolvente utilizado



**Figura 53. Señal a la salida del sistema**

En esta última gráfica, se puede observar con más precisión el efecto trémolo generado. Se puede apreciar en el dominio temporal el crecimiento y decrecimiento de la amplitud de la señal resultante. La señal a la salida del sistema sin el efecto del LFO 3 quedaría como en la siguiente gráfica:



**Figura 54. Señal a la salida del sistema sin LFO 3**

Como se puede observar, el VCA modela la envolvente de la salida y sin embargo, al aplicar el LFO 3, se cambia por completo la forma de la señal, obteniéndose el efecto tremolo.

## **4.6. Conclusiones**

El sintetizador diseñado en Matlab™ para este apartado cumple con una serie de requisitos y características muy similares a sintetizadores analógicos célebres, como el original MiniMoog™ diseñado por Robert Moog. Aunque dicho sintetizador es completamente analógico, en esta sección se ha generado un sintetizador digital con una interfaz gráfica dinámica y sencilla.

Gracias a Matlab™, se ha podido diseñar en primer lugar un entorno gráfico de control del sintetizador y posteriormente un desarrollo del funcionamiento de cada bloque. Posteriormente se ha podido realizar un análisis y ver lo que ocurre con las señales en cada etapa del sintetizador, además de entender con más claridad su funcionamiento paso a paso.

## 5. EFECTOS DE AUDIO EN MATLAB™

---

A lo largo de los siguientes apartados, se hará un estudio en profundidad de las diferentes técnicas de procesado de audio con fines de producción musical, técnicas que se denominan genéricamente “efectos de audio” [2], en el cuál se englobaran los siguientes tipos de efectos:

- Efectos con líneas de retardo:
  - Eco FIR.
  - Eco IIR.
  - Vibrato.
  - Chorus.
- Efectos con filtros.
  - *Wah-Wah*.
- Efectos de procesado de dinámica.
  - Compresor.
- Distorsión
  - Emulador del pedal *Tube Screamer* de la empresa Ibanez™.

En cada subapartado se hará un análisis exhaustivo de cada uno de los efectos anteriormente presentados utilizando como herramienta la aplicación MatLab™.

### 5.1. Efectos con líneas de retardo

#### 5.1.1. ECO FIR

- **Introducción:**

En audio, el eco se define como la reflexión del sonido producida por una fuente. O más bien, la onda reflejada que recibe un receptor con un retardo mínimo de 100 ms con respecto a la señal directa recibida. En procesado de audio se conoce el efecto ECO como Delay y a continuación se va a estudiar una de sus implementaciones.

- **Análisis:**

El sistema se puede modelar como un filtro FIR (Finite Impulse Response) combinando cada muestra de la señal de entrada más la propia señal retardada, en el instante anterior  $M$ , y escalada. A este tipo de filtro se le denomina filtro de peine ya que su respuesta al impulso, para  $M > 1$ , tiene una serie de lóbulos muy característicos,



parecidos a las púas de un peine, repartidos a lo largo de todo el espectro. En el análisis se verá esto con mayor detenimiento. A continuación se presenta el diagrama de bloques:

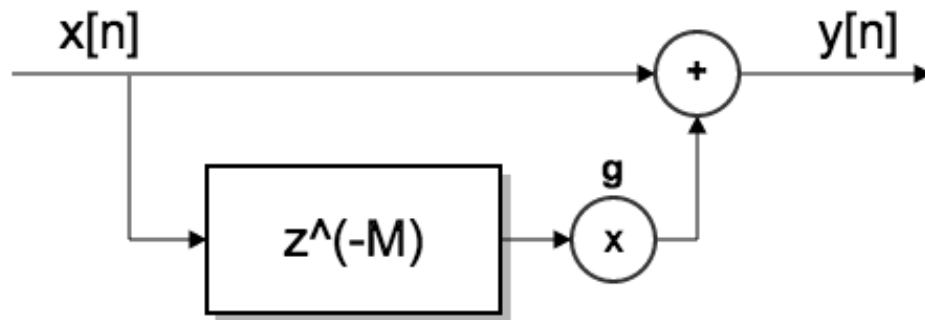


Figura 55. FIR filtro de peine

La expresión matemática de este sistema en tiempo discreto viene definida como:

$$y[n] = x[n] + gx[n - M] \quad \text{(Fórmula 17)}$$

Donde  $M = \tau f_s$ ,  $f_s$  : frecuencia de muestreo,  $\tau$ : tiempo de retarde entre onda transmitida y reflejada.

Si realizamos la transformada de Laplace de la anterior expresión para analizarla y obtener la función de transferencia del sistema, se puede observar que tiene un zero de orden M:

$$H(z) = 1 + gz^{-M} \quad \text{(Fórmula 18)}$$

El parámetro g toma valores ente 0 y 1 y controla la amplitud de los lóbulos del filtro.

Al representar la respuesta al impulso del sistema para  $g = 1$  y diferentes valores de M, se observa que el número de lóbulos es proporcional a  $M = \tau f_s$ , lo que le da al filtro la característica forma de peine.

M = 1

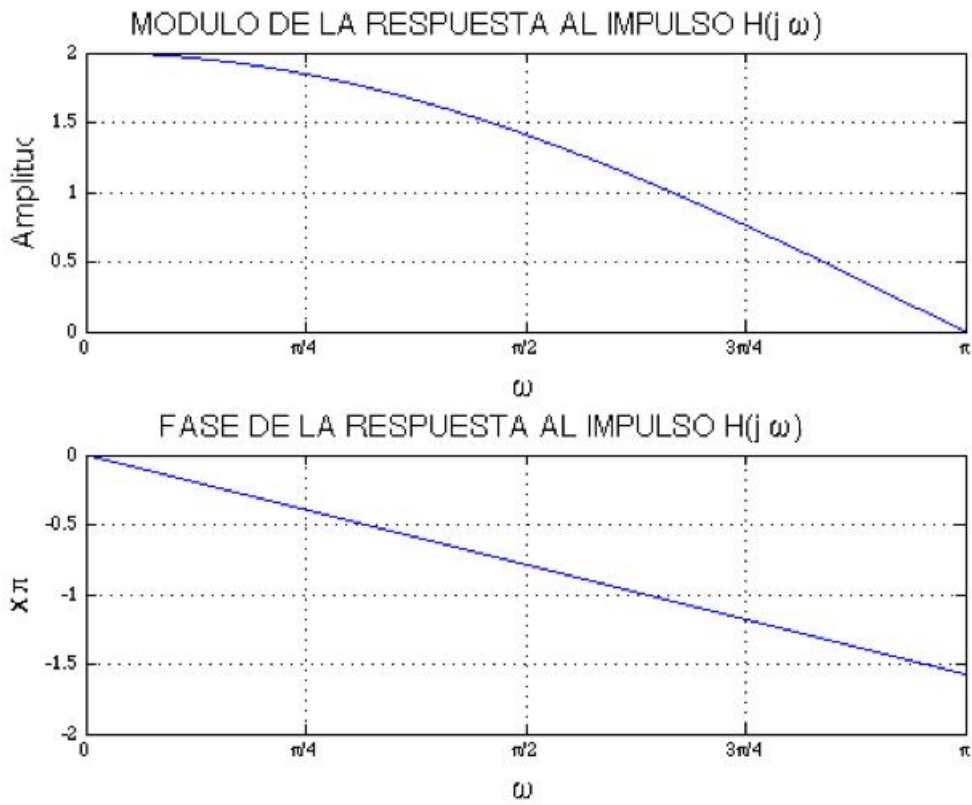


Figura 56. Módulo y fase de H(jω) para M = 1

M = 2

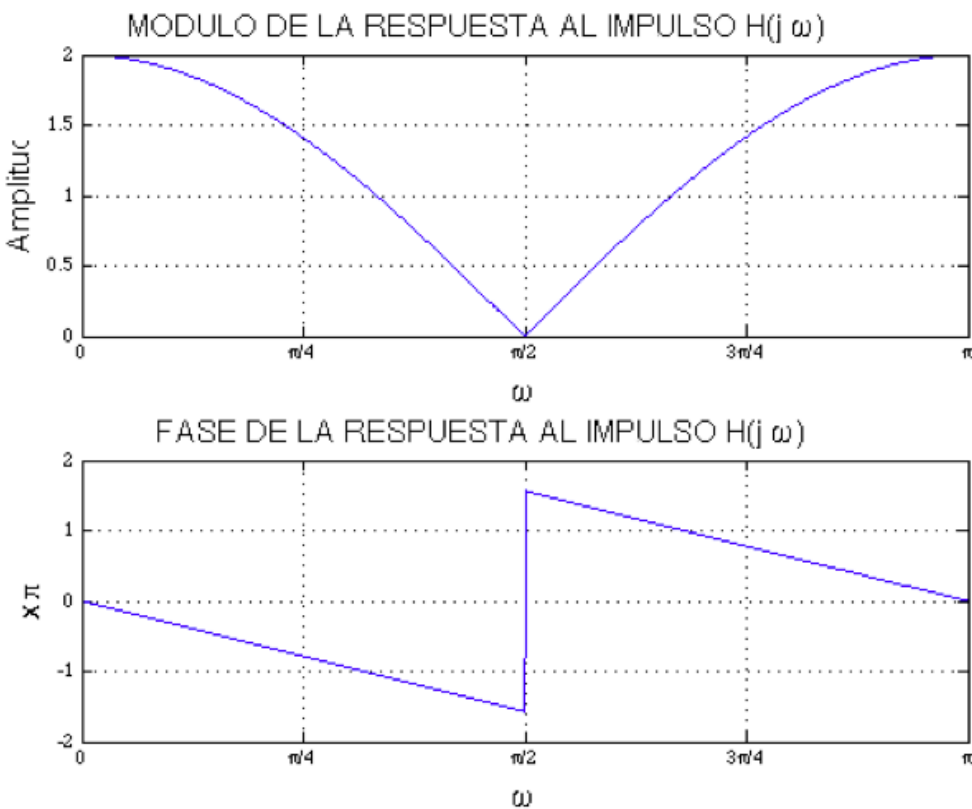


Figura 57. Módulo y fase de H(jω) para M = 2

M = 10

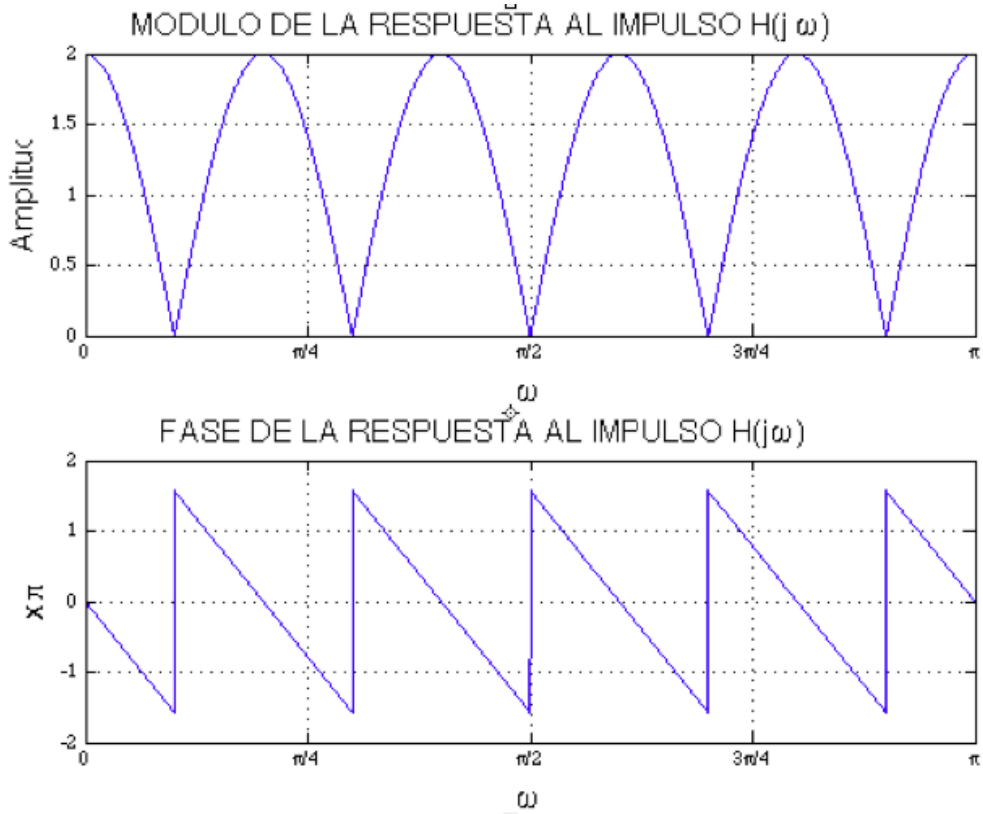


Figura 58. Módulo y fase de H(jω) para M = 10

M = 50

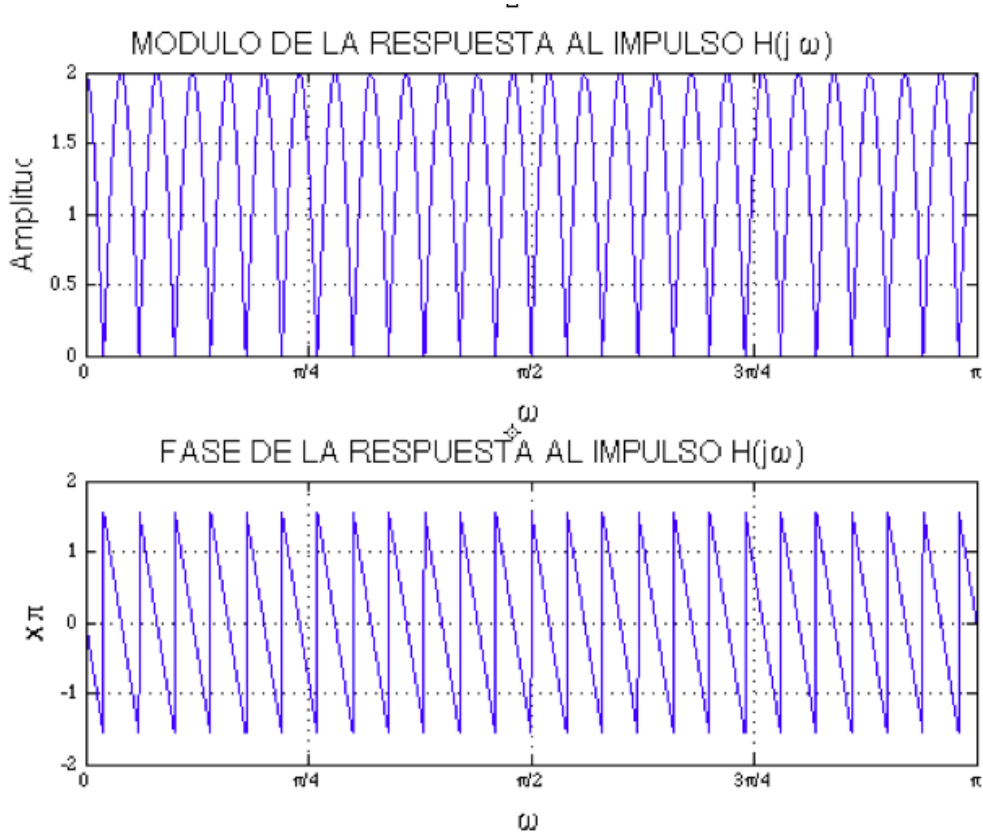


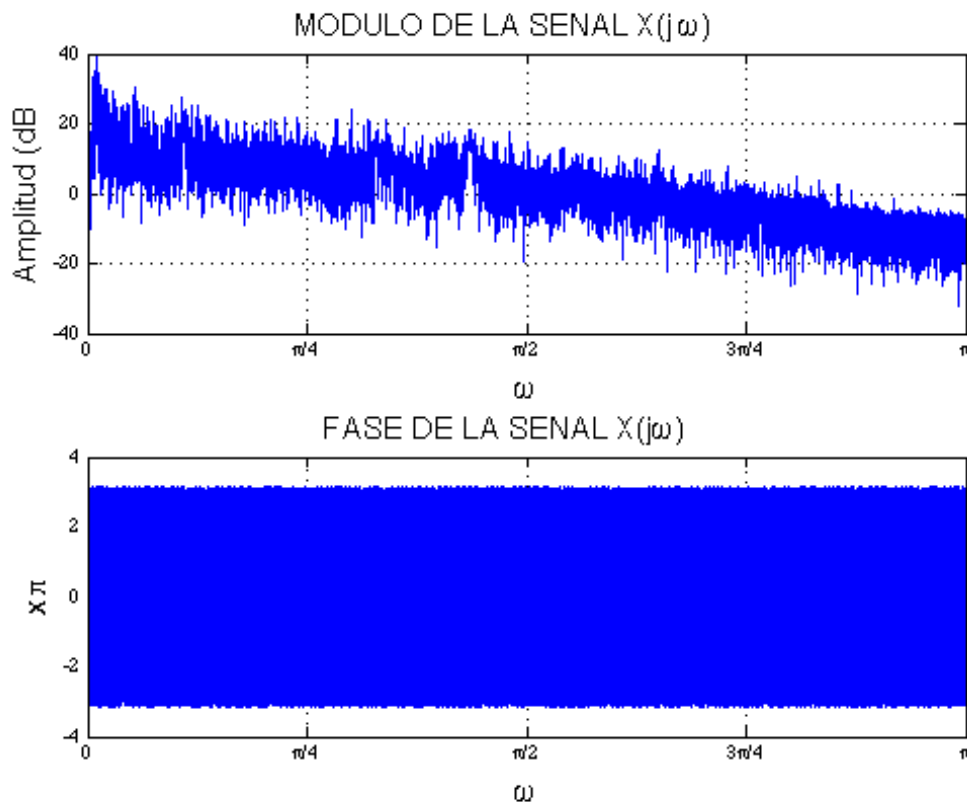
Figura 59. Módulo y fase de H(jω) para M = 50

○ **Resultados caso práctico:**

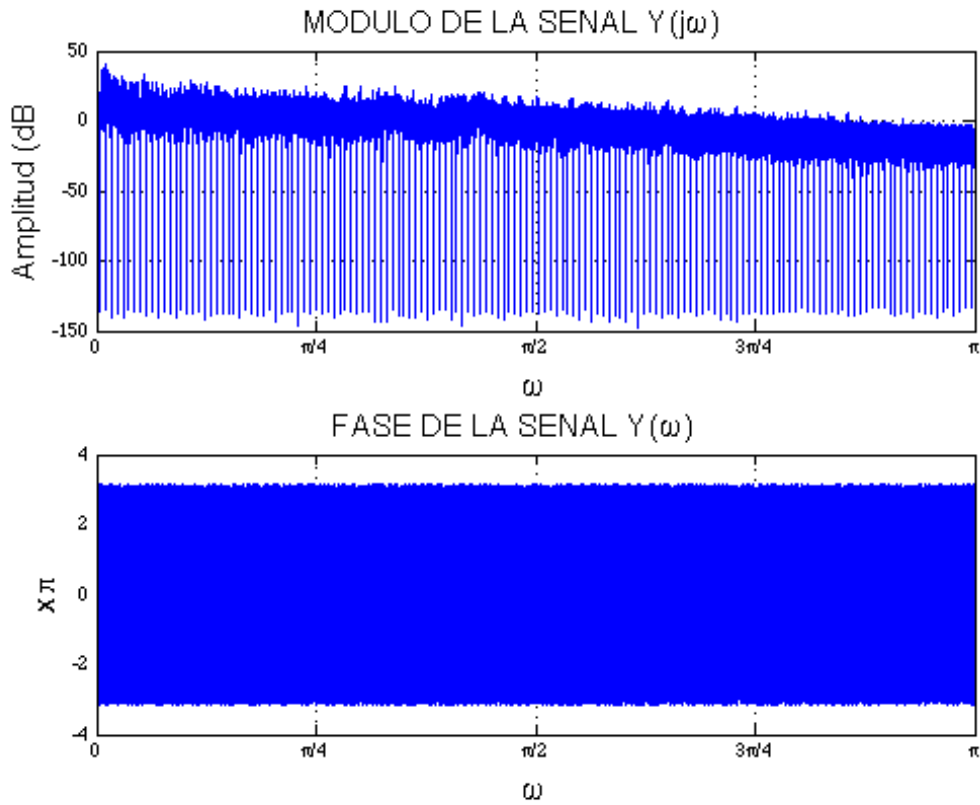
Para percibir este efecto de forma sensorial, se dispone de un archivo de audio de muestra, el cual se filtra y se ejecuta con los siguientes parámetros de entrada  $M = 7500$  y  $g = 1$ .

Como se puede apreciar, al ejecutar el código, se produce un eco del archivo original, que se puede aumentar con la variable  $M$ , que es la que introduce el tamaño del retardo en la muestras. Se han tabulado los valores de  $M$  entre 5000 y 10000 ya que si se escoge un valor de  $M$  menor, no se apreciaría el efecto del filtro. Si se escoge un valor superior, lógicamente el tamaño del filtro será mayor al igual que el retardo y aumentará el tiempo de ejecución considerablemente.

A continuación se representa el espectro de la señal de entrada y de salida del filtro FIR.

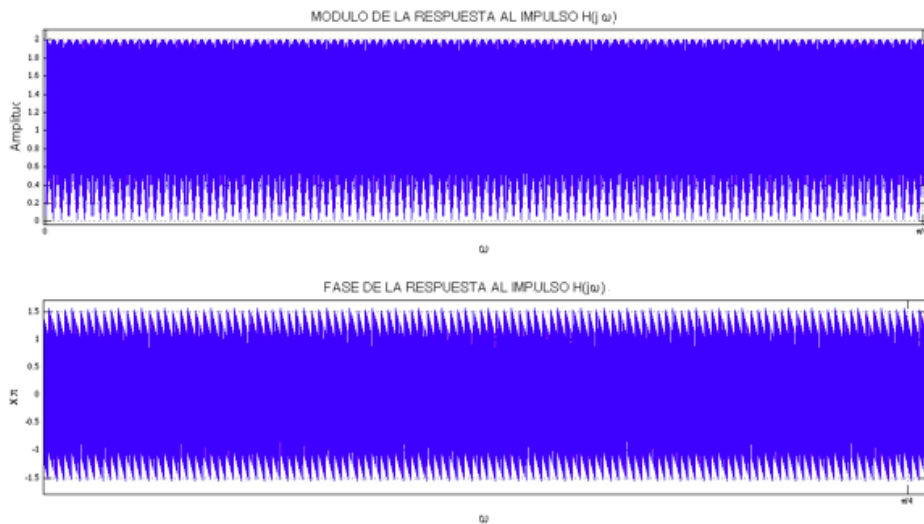


**Figura 60. Módulo y fase de X(jw)**



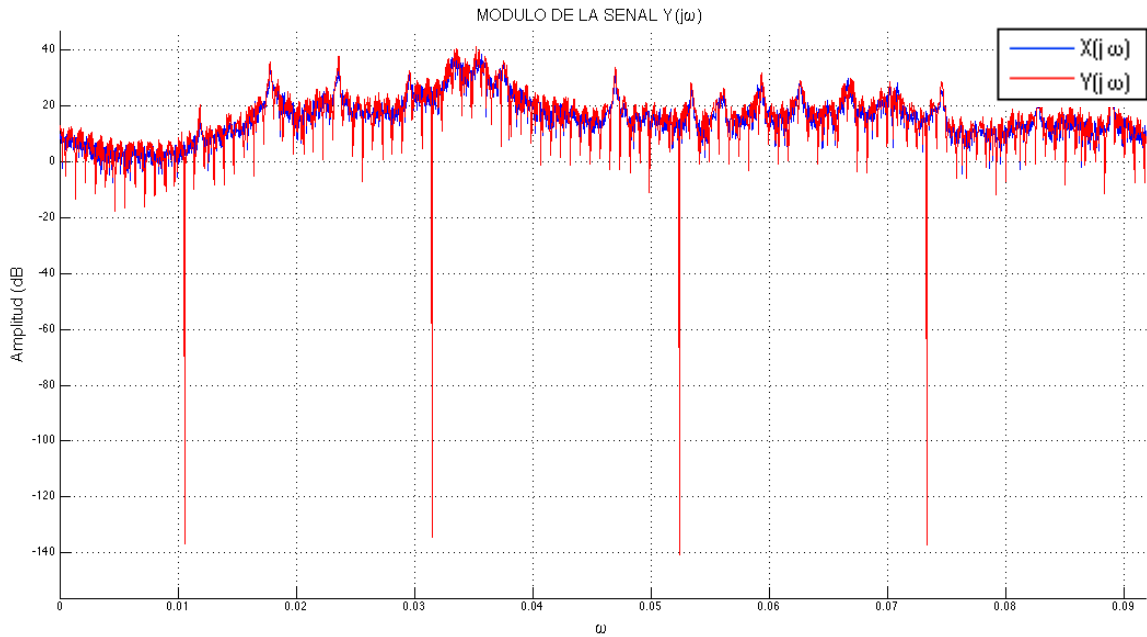
**Figura 61. Módulo y fase de  $Y(j\omega)$**

El filtro utilizado es el siguiente:



**Figura 62. Respuesta al impulso  $H(j\omega)$  ECO FIR**

Si ampliamos el espectro y comparamos la señal de salida con la de la entrada:



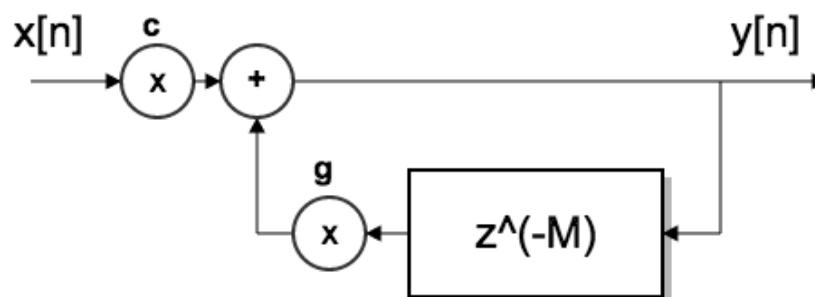
**Figura 63. Módulo ampliado de X(jω) solapado con Y(jω)**

Como se puede apreciar en la anterior gráfica, la señal a la salida del ECO queda filtrada y se replican los picos del filtro a las frecuencias en las cuales decaen.

### 5.1.2. ECO IIR

- **Introducción:**

Otra forma de simular el efecto de retardo en audio es aplicando un filtro de peine IIR (Infinite Impulse Response) combinando cada muestra de la señal de entrada con la salida de la señal escalada en el instante anterior  $M$ . A continuación se presenta el diagrama de bloques:



**Figura 64. IIR filtro de peine**

○ **Análisis:**

La expresión matemática de este sistema en tiempo discreto viene definida como:

$$y[n] = cx[n] + gy[n - M] \quad \text{(Fórmula 19)}$$

Donde  $M = \tau fs$ .

Si realizamos la transformada de Laplace de la anterior expresión para analizarla y obtener la función de transferencia del sistema, se puede observar que tiene un polo de orden  $M$ :

$$H(z) = c/(1 - gz^{-M}) \quad \text{(Fórmula 20)}$$

Los parámetros  $c$  y  $g$  controla la amplitud de los picos del filtro, si  $c$  tiene signo negativo, habrá un desfase de  $180^\circ$ .

Al representar la respuesta al impulso del sistema para  $c = 1$   $g = 0.5$  y diferentes valores de  $M$ , se observa que el número de lóbulos principales es proporcional a  $M = \tau fs$ .

$M = 1$

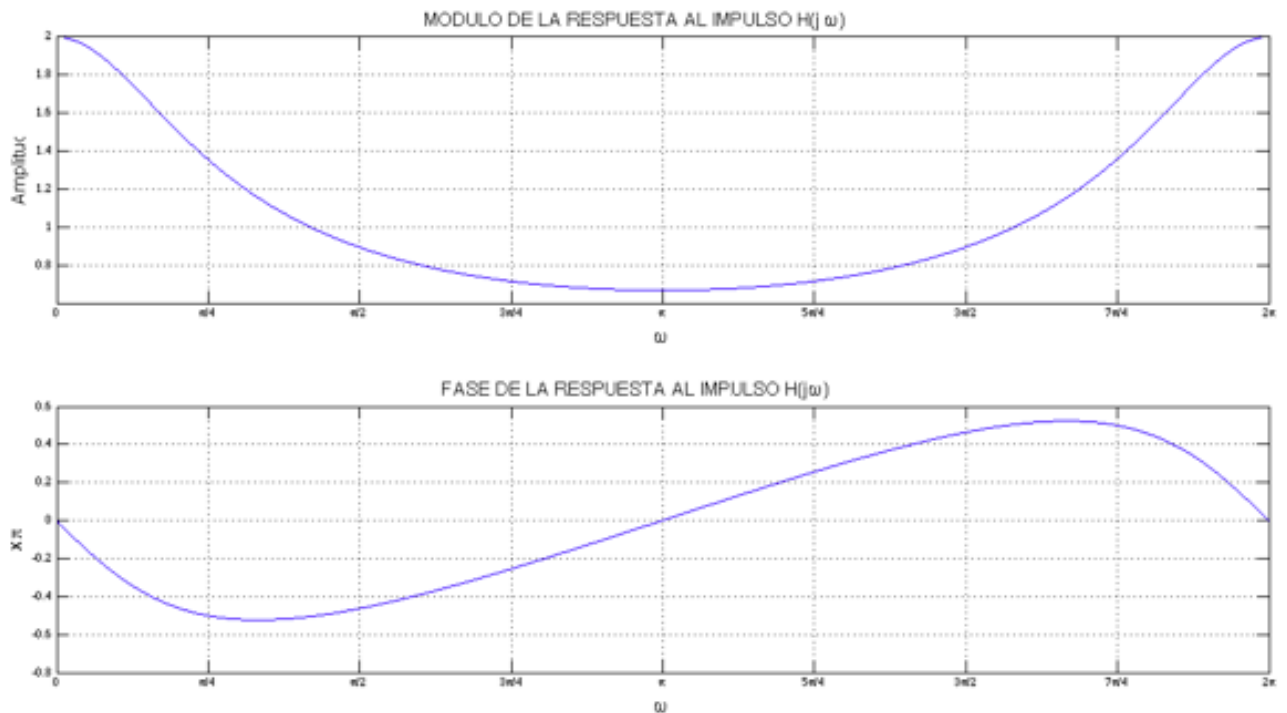


Figura 65. Módulo y fase de  $H(jw)$  para  $M = 1$

$M = 2$

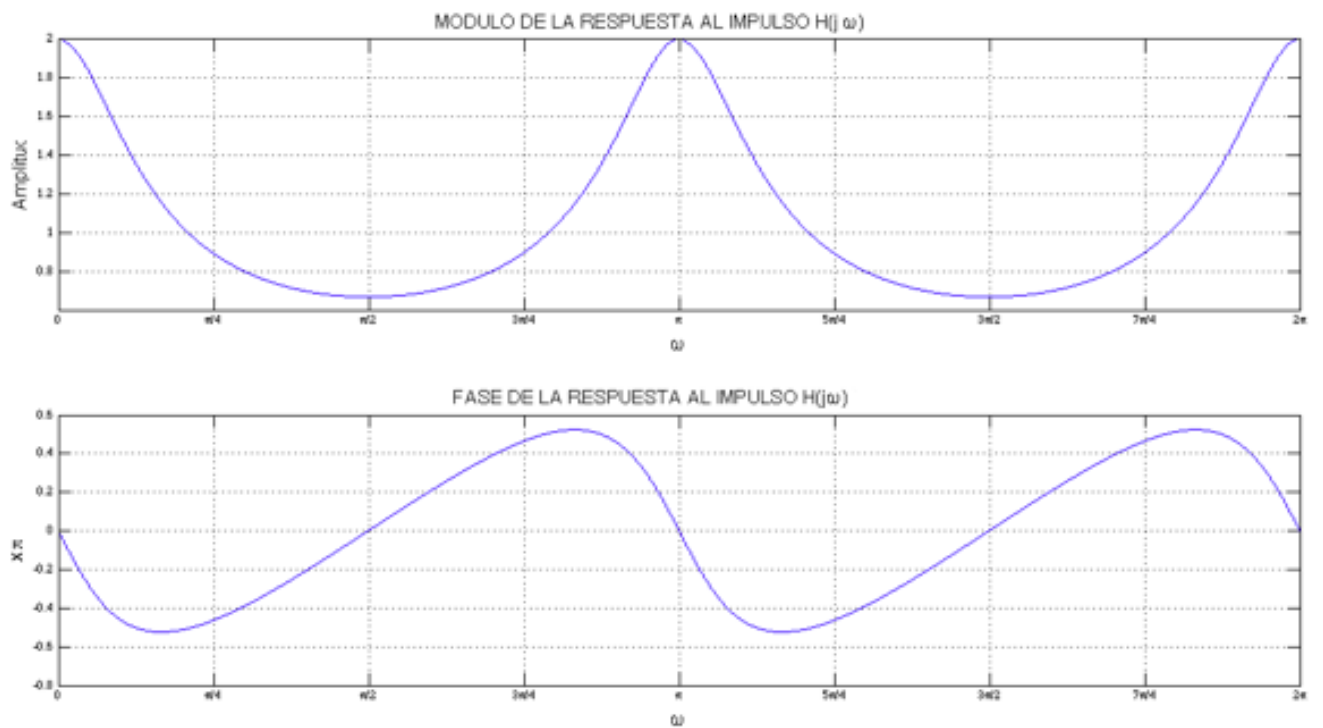


Figura 66. Módulo y fase de  $H(j\omega)$  para  $M = 2$

$M = 5$

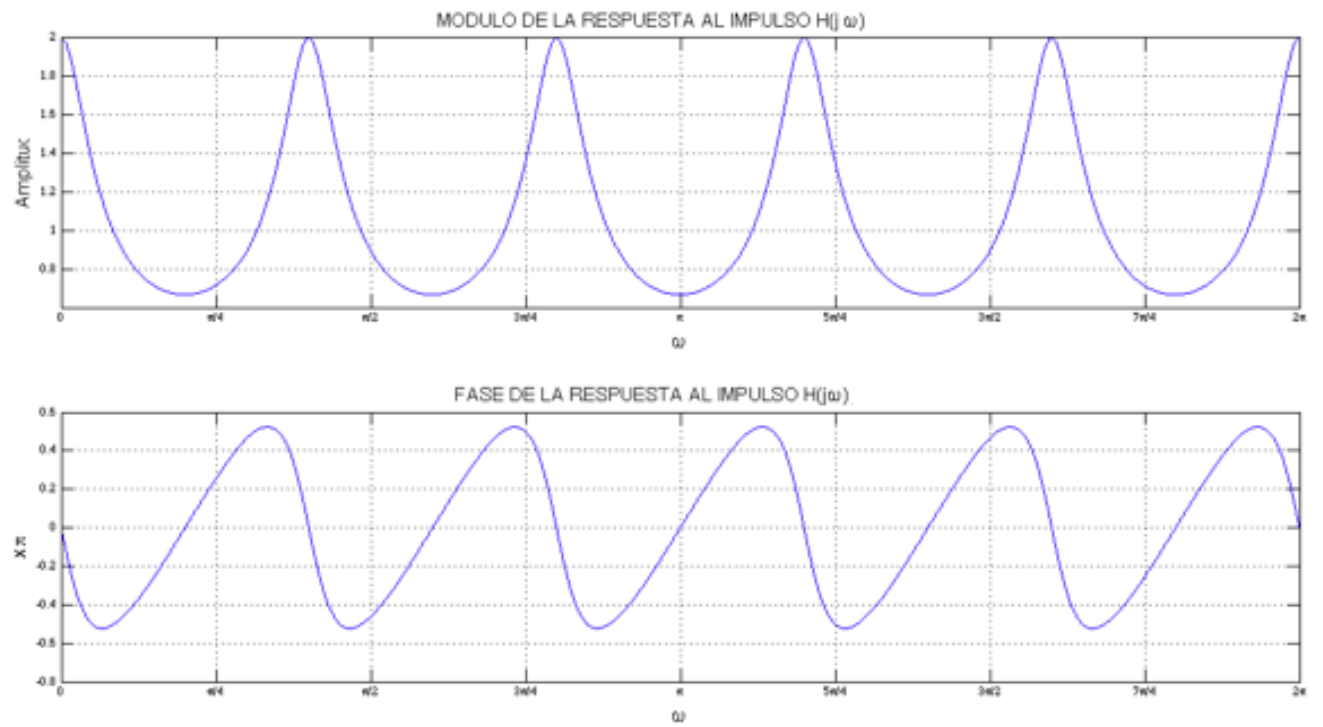


Figura 67. Módulo y fase de  $H(j\omega)$  para  $M = 5$



$M = 10$

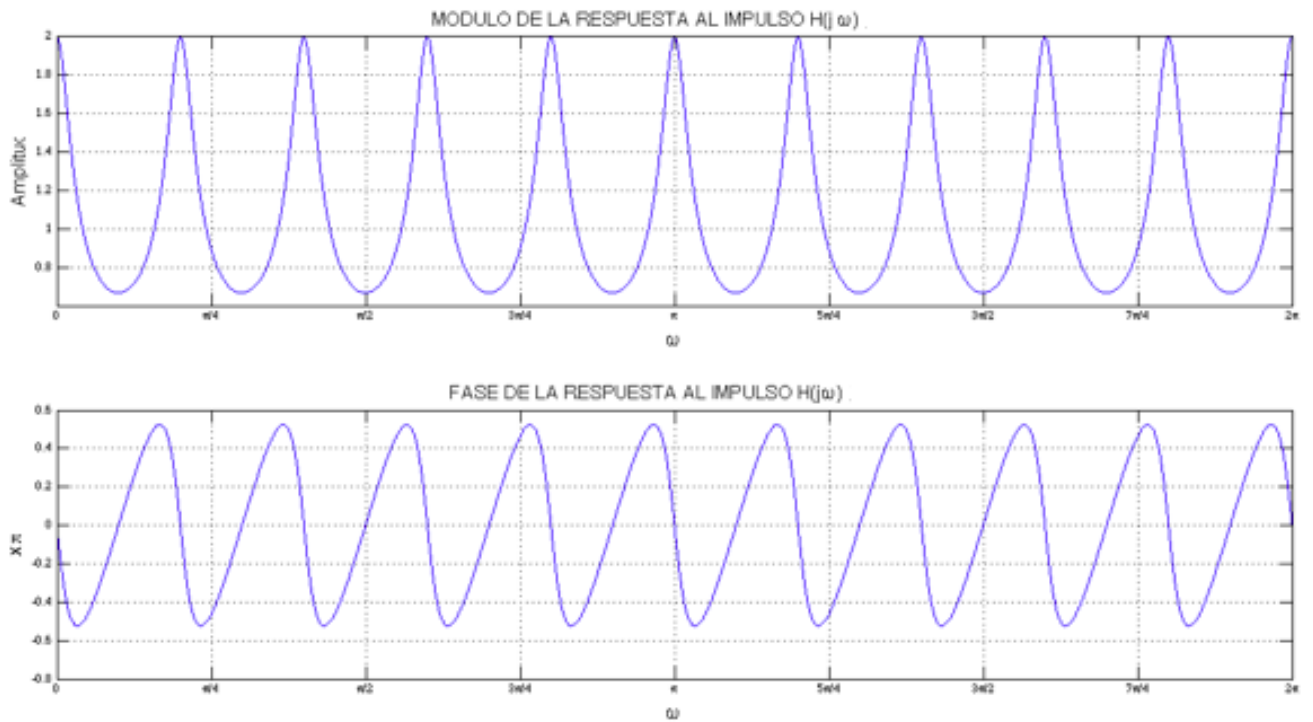


Figura 68. Módulo y fase de  $H(j\omega)$  para  $M = 10$

○ **Resultados caso práctico:**

Para percibir este efecto de forma sensorial, se dispone de un archivo de audio de muestra, el cual se filtra y se ejecuta con los siguientes parámetros de entrada  $M = 7500$ ,  $c = 1$  y  $g = 0.5$ .

A continuación se representan la transformada de la señal de entrada y de salida del filtro IIR:

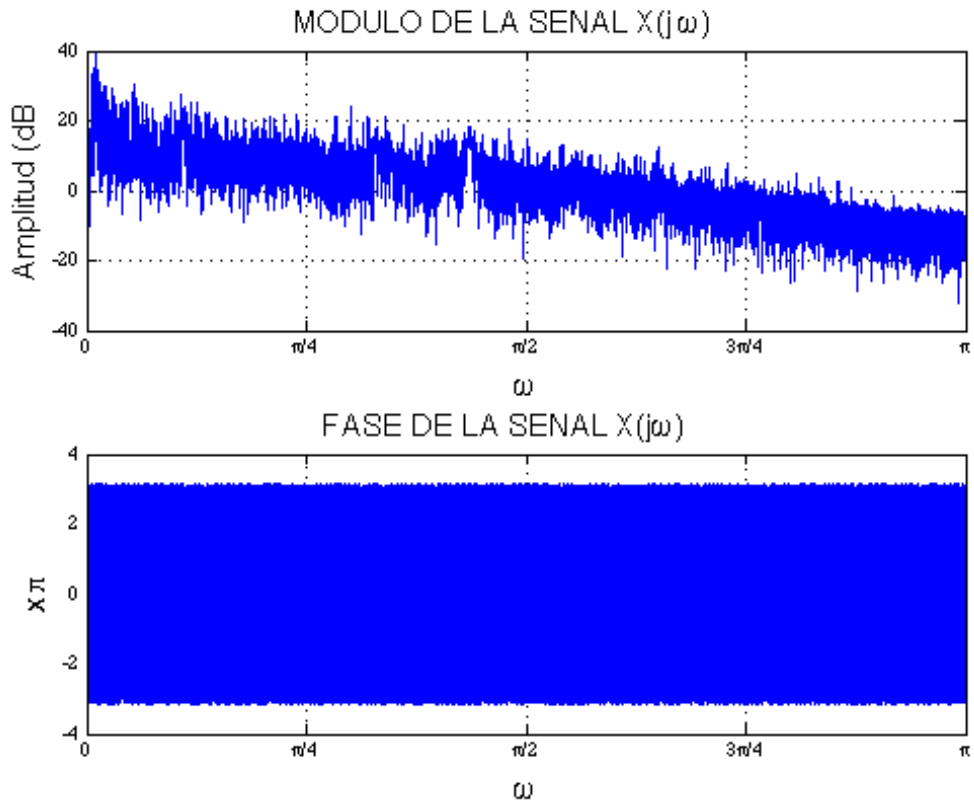


Figura 69. Módulo y fase de  $X(j\omega)$

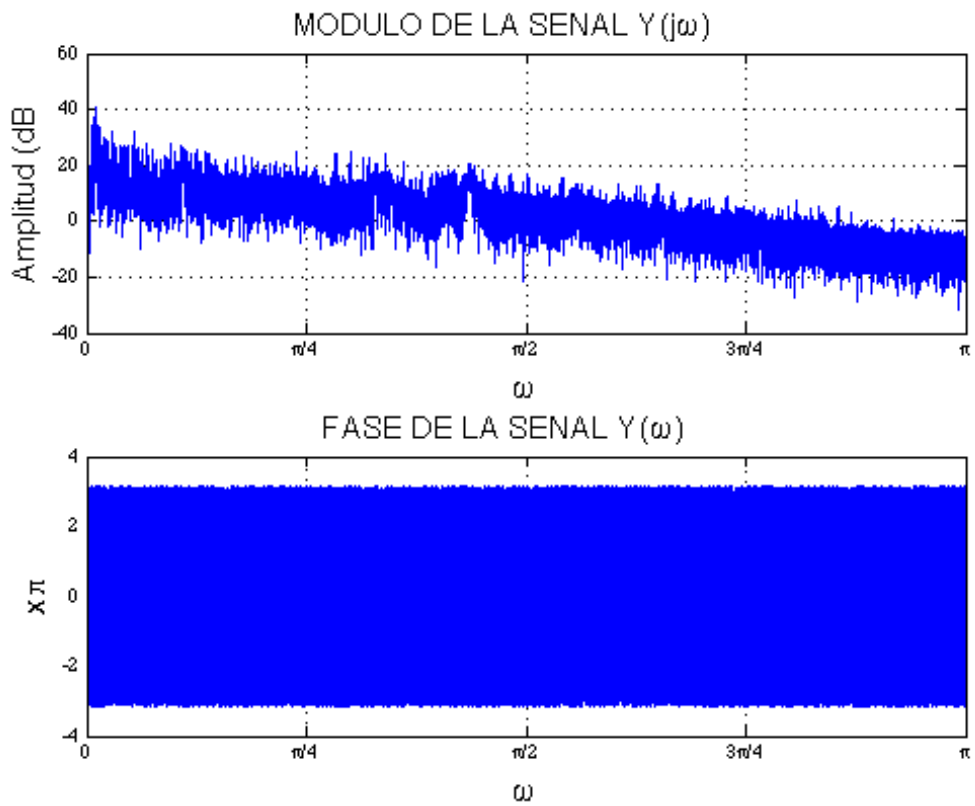


Figura 70. Módulo y fase de  $Y(j\omega)$

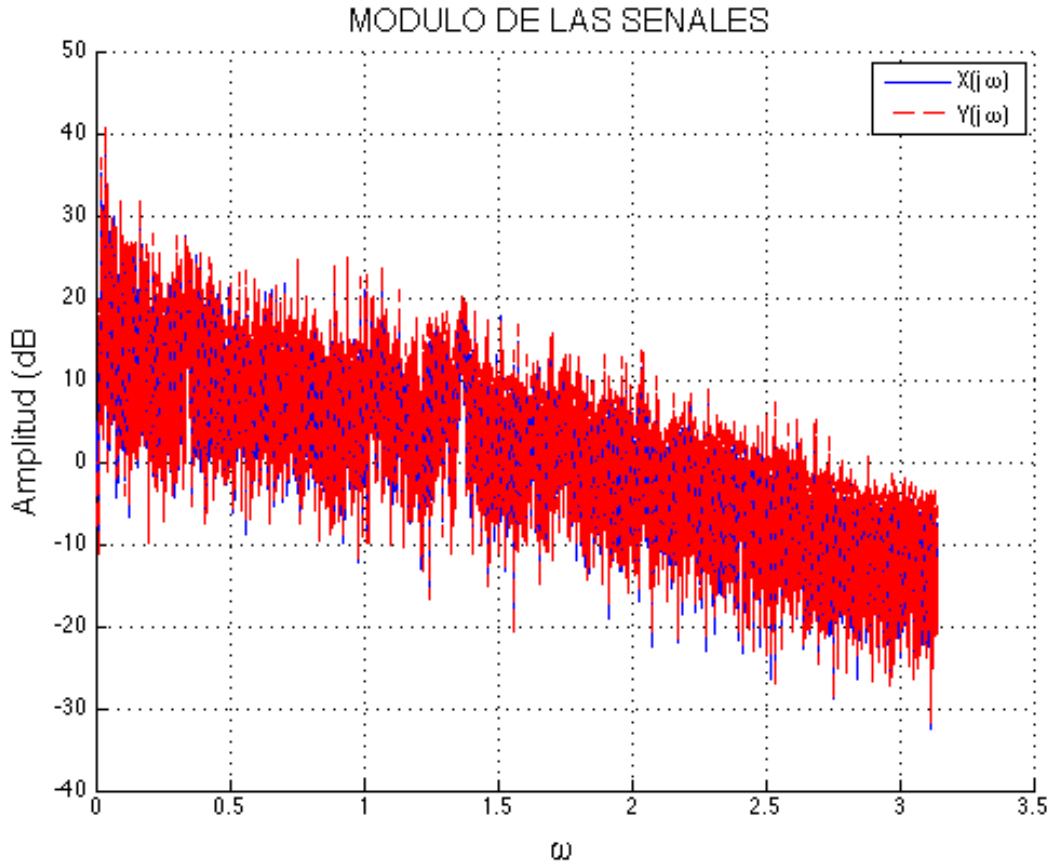


Figura 71. Módulo de  $X(j\omega)$  solapado con  $Y(j\omega)$

En la siguiente gráfica, se ha ampliado el espectro y se puede observar que la señal filtrada  $Y(j\omega)$  representada en color rojo, tiene casi las mismas características que la señal original, excepto por los picos del filtro.

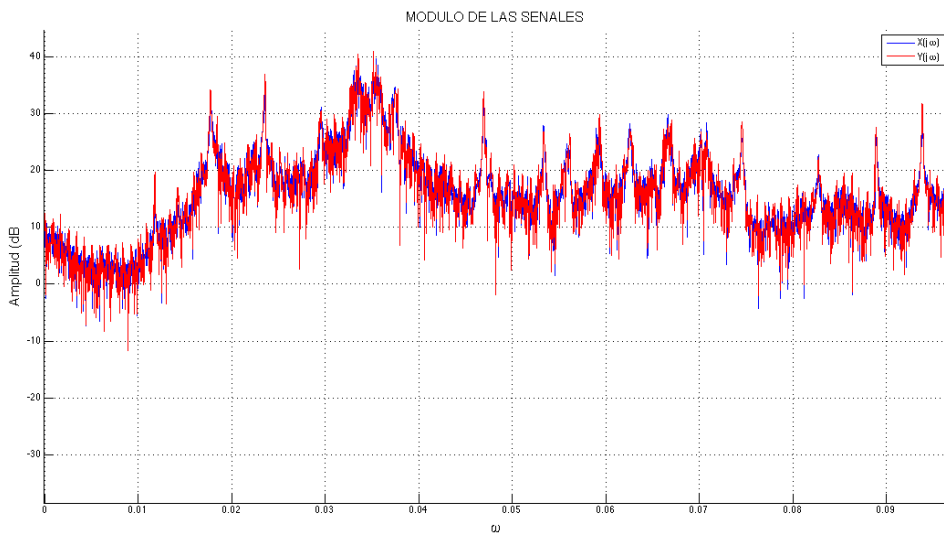


Figura 72. Módulo ampliado de  $X(j\omega)$  solapado con  $Y(j\omega)$

### 5.1.3. VIBRATO

- **Introducción:**

En audio, el Vibrato se define como la variación periódica de la frecuencia de un sonido. Las dos características que permiten definir el Vibrato son: el rango de variación de la frecuencia y la velocidad de variación de la frecuencia.

- **Análisis:**

El sistema se puede modelar como un filtro cuyo retardo en el tiempo es variable y producido por un oscilador de baja frecuencia LFO. El diagrama siguiente muestra el sistema:

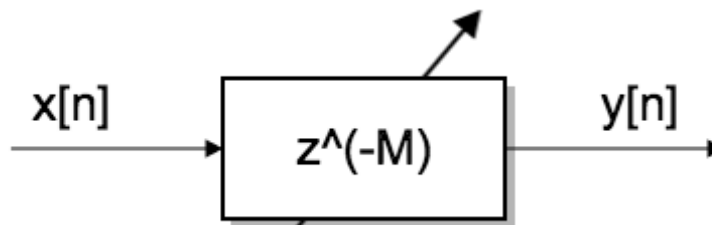


Figura 73. Vibrato

Para poder implementar este sistema se debe tener en cuenta que la salida  $y[n]$  obtiene muestras interpoladas de  $x[n]$ , dado que la variación del LFO puede dar lugar a un valor de retardo no entero medido en muestras. Dichas muestras se obtienen en torno a un rango acotado que oscila, analíticamente su expresión matemática es la siguiente:

$$y[n] = x[n - (M + 1)]frac + x[n - M](1 - frac) \quad \text{(Fórmula21)}$$

Donde  $M$  y  $frac$  se calcula a partir de los siguientes parámetros de entrada:

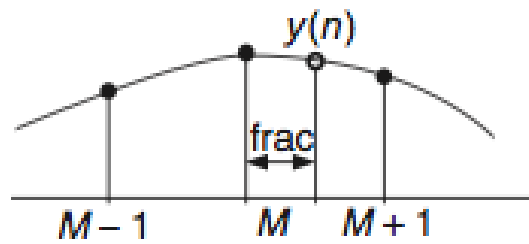
- Delay : Tamaño del buffer de retardo.
- $f_s$  : Frecuencia de muestreo.
- $f_{MOD}$  : Frecuencia de variación de las muestras.

$$M = \left\lceil 1 + Delay \cdot f_s + Delay \cdot f_s \cdot \sin\left(\frac{f_{MOD}}{f_s} 2\pi n\right) \right\rceil \quad \text{(Fórmula22)}$$

$$frac = \left( 1 + Delay \cdot f_s + Delay \cdot f_s \cdot \sin\left(\frac{f_{MOD}}{f_s} 2\pi n\right) \right) - M \quad \text{(Fórmula23)}$$

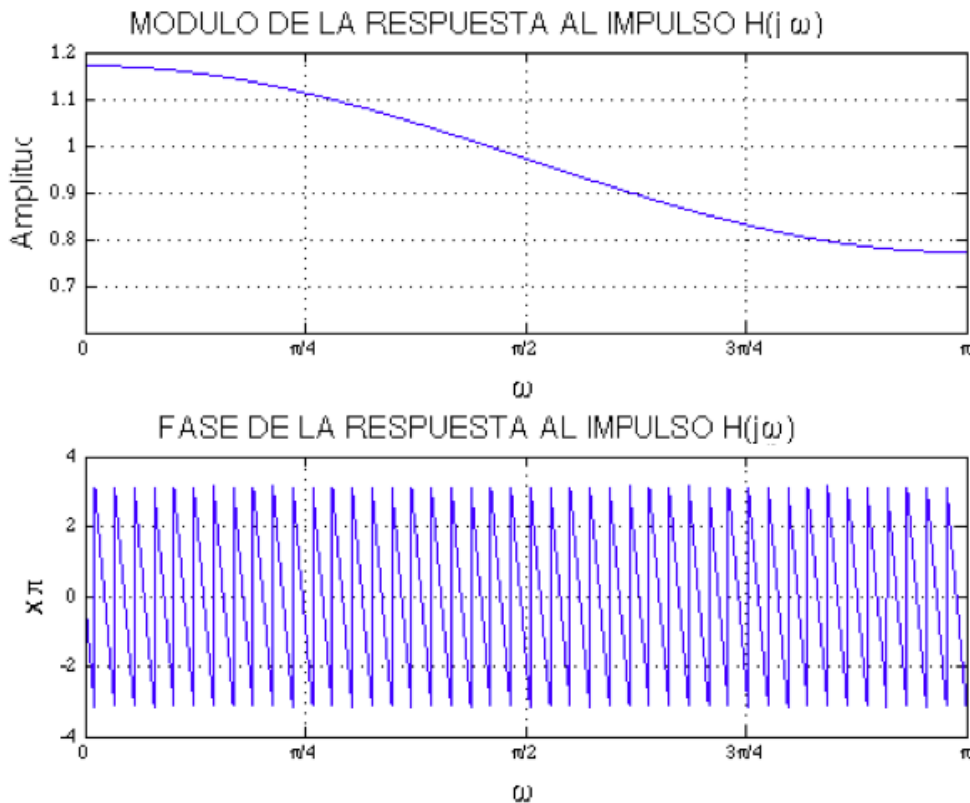
Si analizamos la señal de salida,  $M$  dará unos valores variables y enteros que oscilan dependiendo de los valores  $Delay$ ,  $f_s$ ,  $f_{MOD}$ . Sin embargo,  $frac$  multiplica las dos muestras interpoladas por un numero decimal que sería el error resultante de la aproximación de  $M$ .

Básicamente, el vibrato se encarga de obtener de entre dos muestras interpoladas  $M-1$  y  $M+1$  de la entrada, un valor intermedio dependiente de  $frac$ :



**Figura 74. Vibrato Muestras (Ref: Digital Audio Effects, Zölzer11)**

Al representar la respuesta al impulso del sistema para un retardo  $Delay$  de 8 ms y una frecuencias  $f_s = 10000$  Hz  $f_{MOD} = 2$  Hz se puede observar que el sistema se comporta como un filtro paso bajo de muy baja atenuación para las altas frecuencias. Si se decrementa la frecuencia de muestreo se pierde resolución y la respuesta al impulso sería plana.



**Figura 75. Módulo y fase de  $H(j\omega)$**

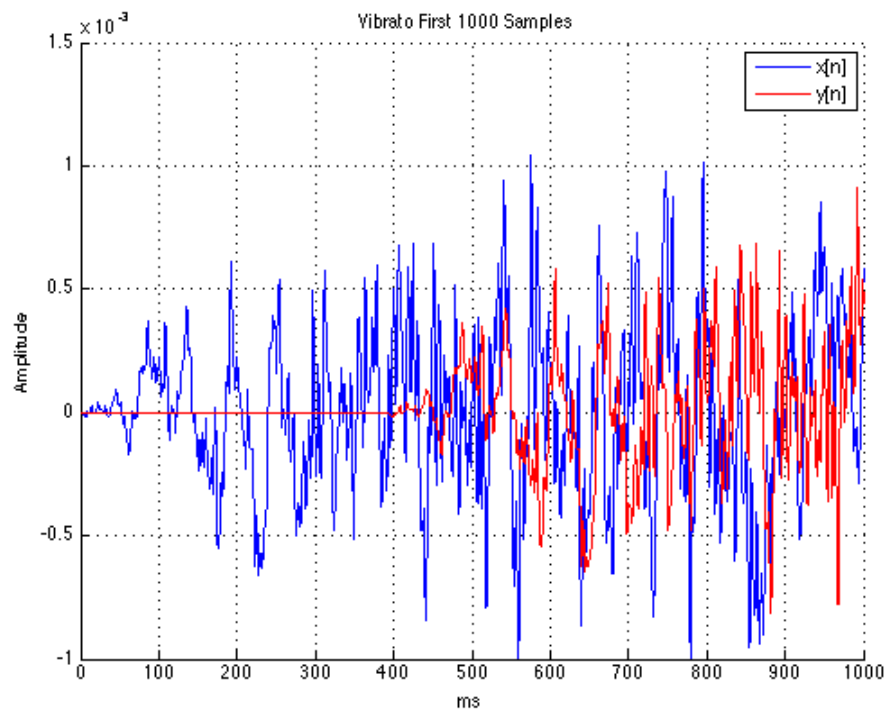
○ **Resultados caso práctico:**

Como se puede apreciar, al ejecutar el código con un archivo de audio, se produce el efecto vibrato alterándolo completamente. Este efecto se puede modificar variando los parámetros:

- $f_{MOD}$  : Controla la velocidad de variación del rango de frecuencias en el que se produce el efecto. Es interesante estudiar los posibles valores que puede tomar ya que si se escogen frecuencias muy bajas, en torno a 1 Hz se aprecia notablemente el rango de frecuencias recorrido. Sin embargo si se escogen frecuencias muy altas, en torno a 100 Hz la variación será tan alta que se generaran productos de intermodulación.

- $Delay$  : Controla el tamaño de buffer de retardo, que se traduce al rango de frecuencias en el que oscila el vibrato, según las fórmulas (22) y (23). Al introducir un retardo mayor, el rango de variación será superior. Los valores recomendados están entorno a ms. Si se utilizan valores inferiores, no se aprecia el efecto vibrato, sin embargo al utilizar retardos en torno al segundo, se produce una saturación ya que el rango de frecuencias que se recorre es superior al audible.

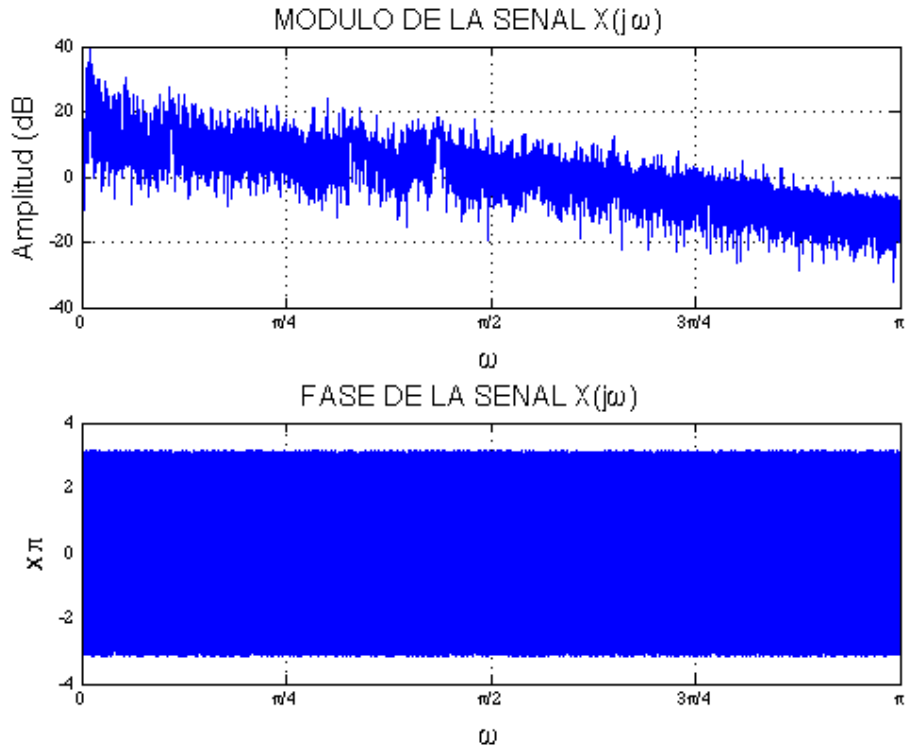
A continuación se representan las señales de entrada y salida del sistema para los valores  $f_{MOD} = 2$  Hz y  $Delay = 8$ ms:



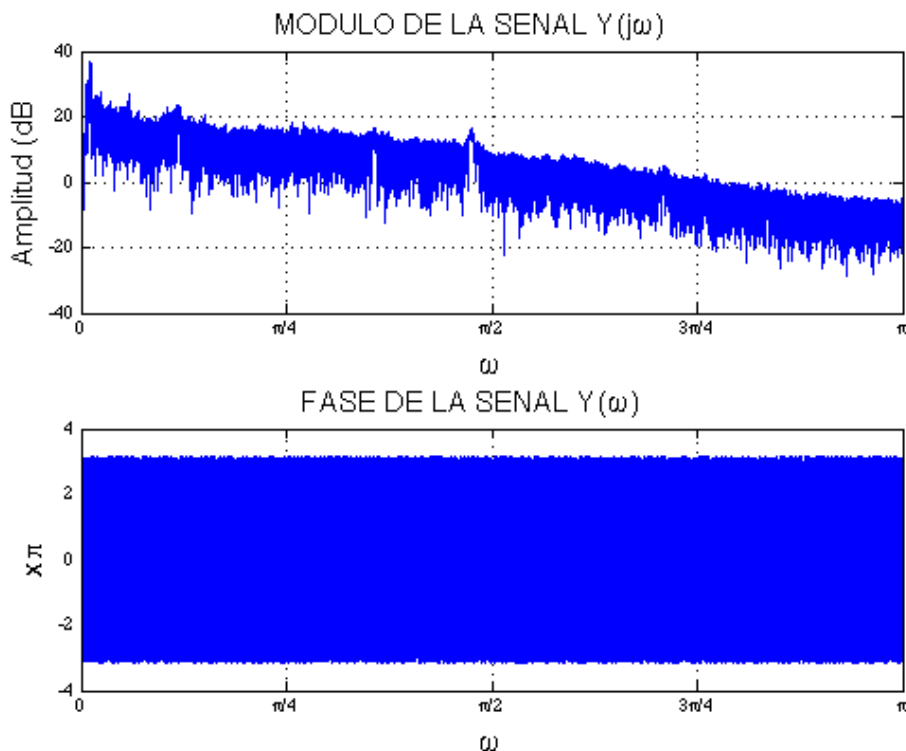
**Figura 76. Primeras muestras de las señales  $x[n]$  e  $y[n]$**

Como se puede observar en la anterior gráfica, la señal  $y[n]$  está ligeramente retardada con respecto a  $x[n]$  ya que el efecto vibrato introduce un pequeño retardo variable, como se ha comentado anteriormente.

A continuación, se representa el espectro de la señal de entrada y de salida para los valores anteriormente citados:



**Figura 77. Módulo y fase de  $X(j\omega)$**



**Figura 78. Módulo y fase de  $Y(j\omega)$**

Si ampliamos el espectro y representamos la señal de entrada junto a la de salida:

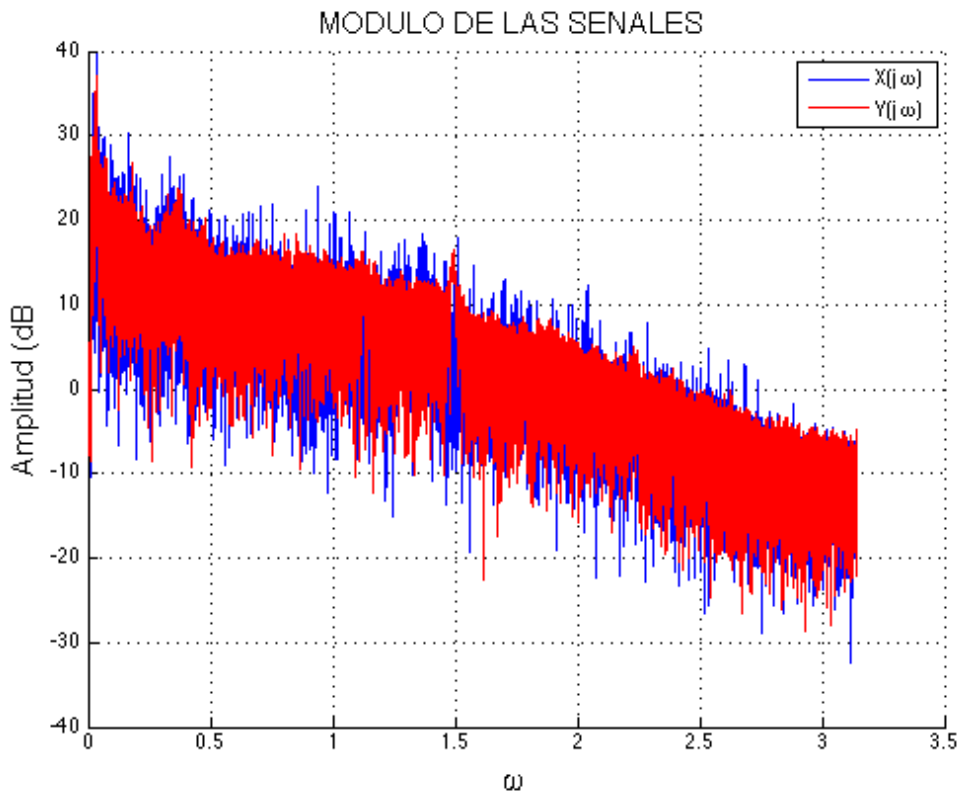


Figura 79. Módulo de  $X(j\omega)$  solapado con  $Y(j\omega)$

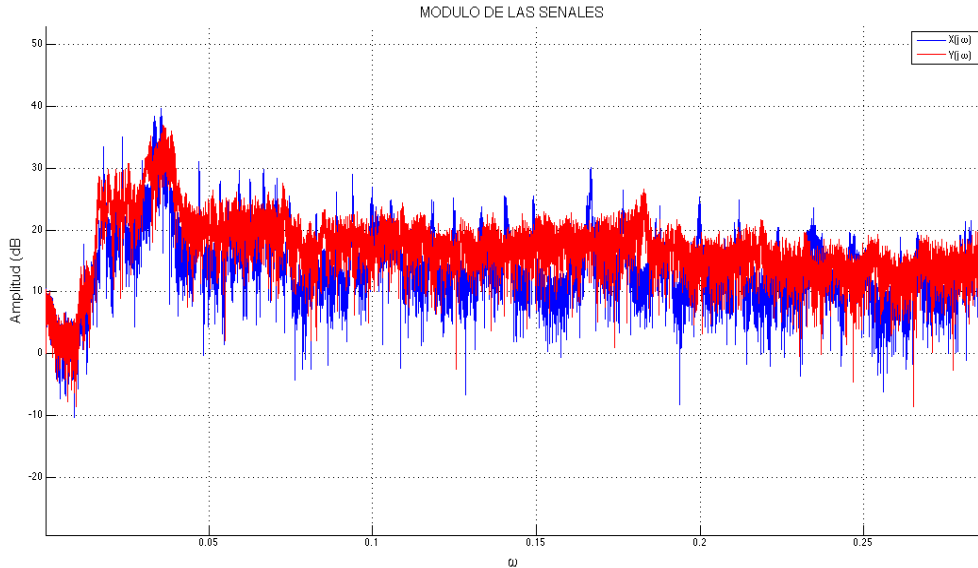


Figura 80. Módulo ampliado de  $X(j\omega)$  solapado con  $Y(j\omega)$

Se puede observar que los espectros no son parecidos como en los casos del ECO ya que el filtro utilizado para simular el vibrato es dinámico e introduce retardos variables.



### 5.1.4. CHORUS

- **Introducción:**

El efecto chorus se puede definir como la suma de contribuciones ligeramente retardadas (entre 10 – 25 ms) y con pequeñas variaciones de tono.

Este efecto se puede sintetizar con varios filtros de peine IIR dispuestos en paralelo cuyos retardos en el tiempo son variables y generados aleatoriamente.

- **Análisis:**

A continuación se muestra el diagrama del sistema:

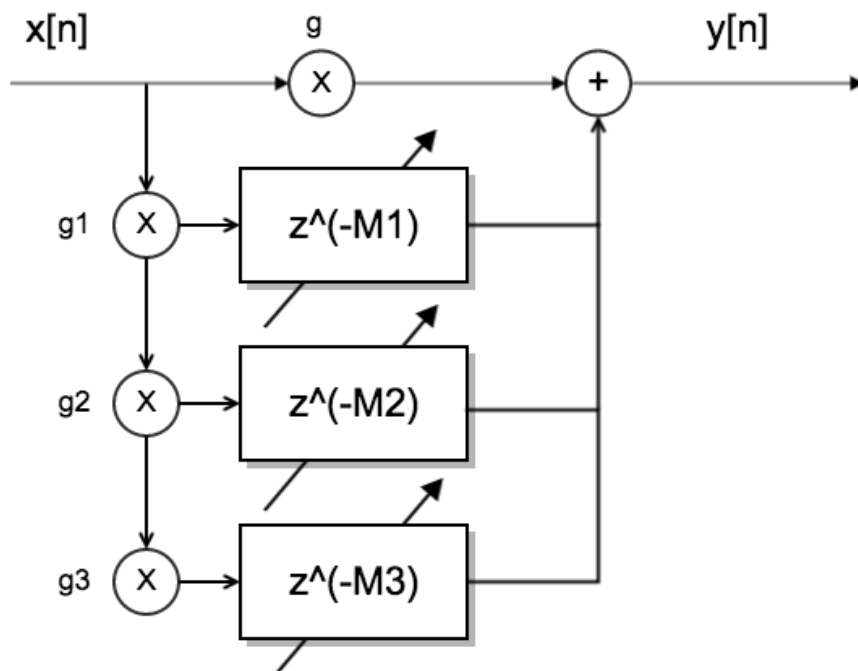


Figura 81. Diagrama de bloques Chorus

La implementación de los filtros es similar a la del vibrato, pero los retardos se introducen con una función de ruido aleatoria de baja frecuencia en vez de utilizar un seno. La expresión matemática del sistema es la siguiente:

$$y[n] = gx[n] + \sum_{i=1}^3 g_i \{x[n - (M_i + 1)]frac_i + x[n - M_i](1 - frac_i)\} \quad \text{(Fórmula24)}$$

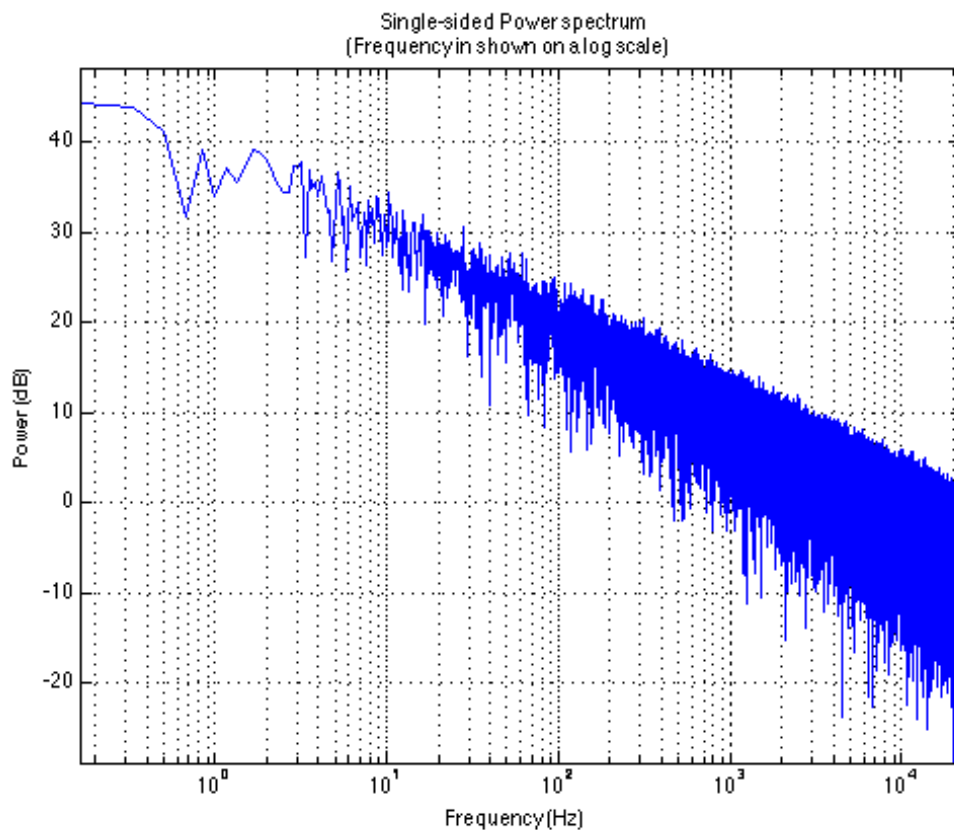
Donde  $M$  y  $frac$  se calcula a partir de los siguientes parámetros de entrada:

- Delay : Tamaño del buffer de retardo.
- $f_s$  : Frecuencia de muestreo.

$$M_i = \lfloor 1 + Delay \cdot f_s + Delay \cdot f_s \cdot noise_i \rfloor \quad \text{(Fórmula 25)}$$

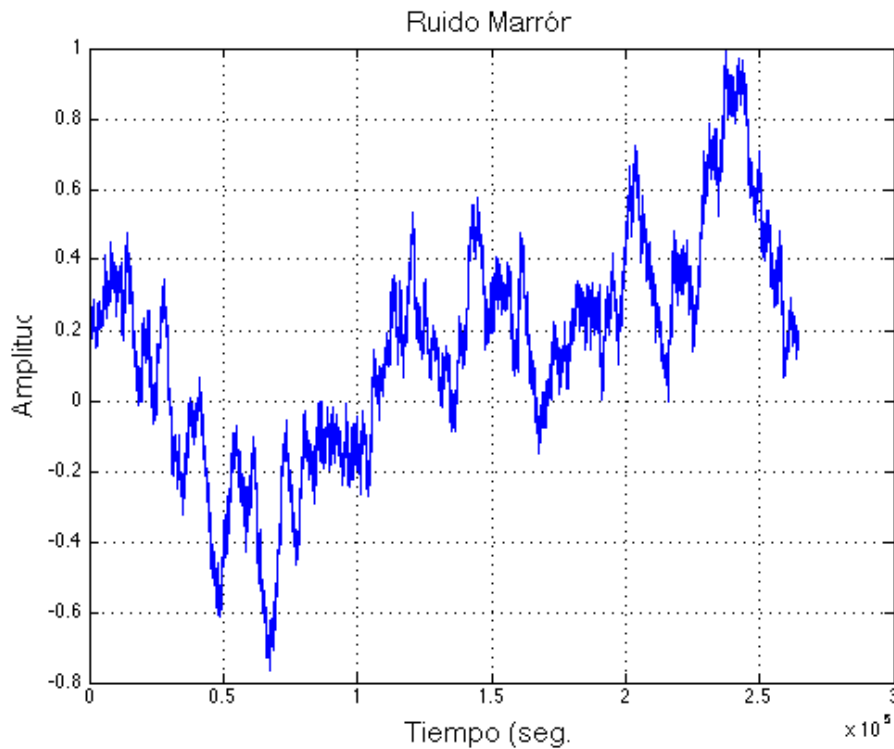
$$frac_i = (1 + Delay \cdot f_s + Delay \cdot f_s \cdot noise_i) - M_i \quad \text{(Fórmula 26)}$$

La función LowFrequencyNoise da los valores de noise y genera una secuencia de muestras de ruido de baja frecuencia muy parecida al ruido marrón. La peculiaridad de este tipo de ruido es que su densidad espectral de potencia decae con la frecuencia. A continuación se muestra la densidad espectral de potencia de esta función:



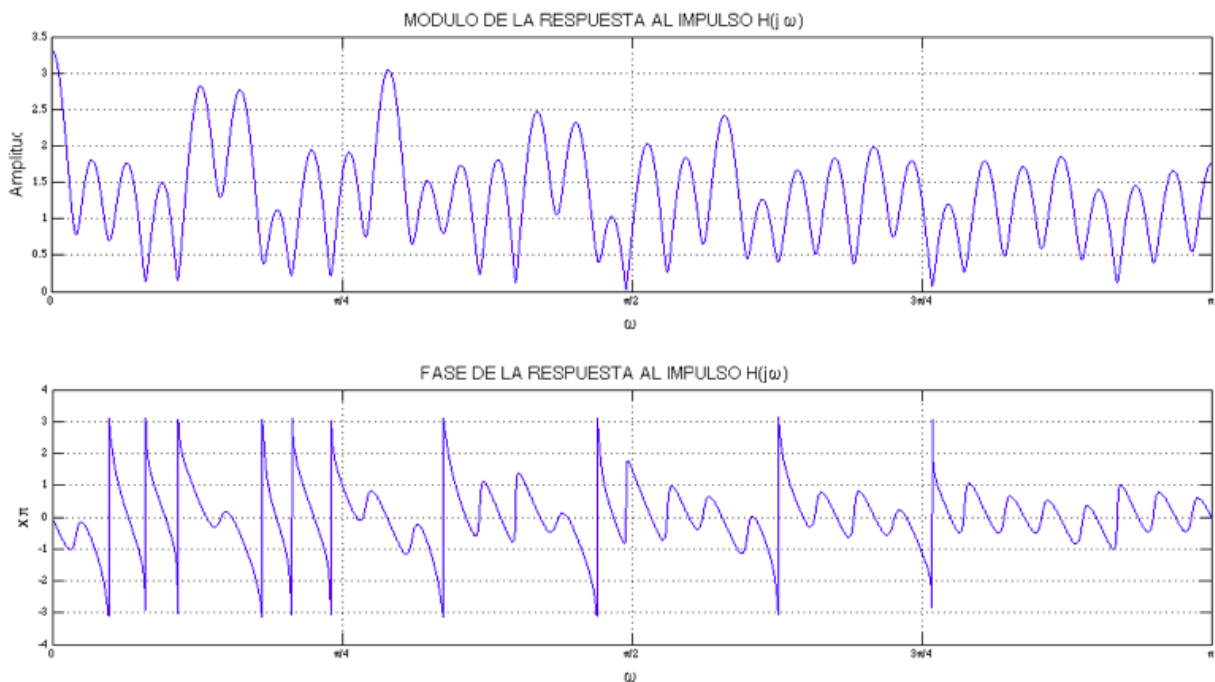
**Figura 82. Respuesta en frecuencia de la función Low Frequency Noise**

En el tiempo, observamos que la señal es completamente aleatoria y con variaciones poco bruscas:



**Figura 83. Low Frequency Noise representado en tiempo**

La respuesta al impulso del sistema es variable y dependiente de la función que genera el ruido, en cada ejecución, el sistema produce una respuesta distinta pero con características similares a las de los filtros de peine. A continuación, se muestra la respuesta al impulso en dos ejecuciones independientes, pero con los mismos parámetros.



**Figura 84. Respuesta al impulso Chorus, ejecución 1**

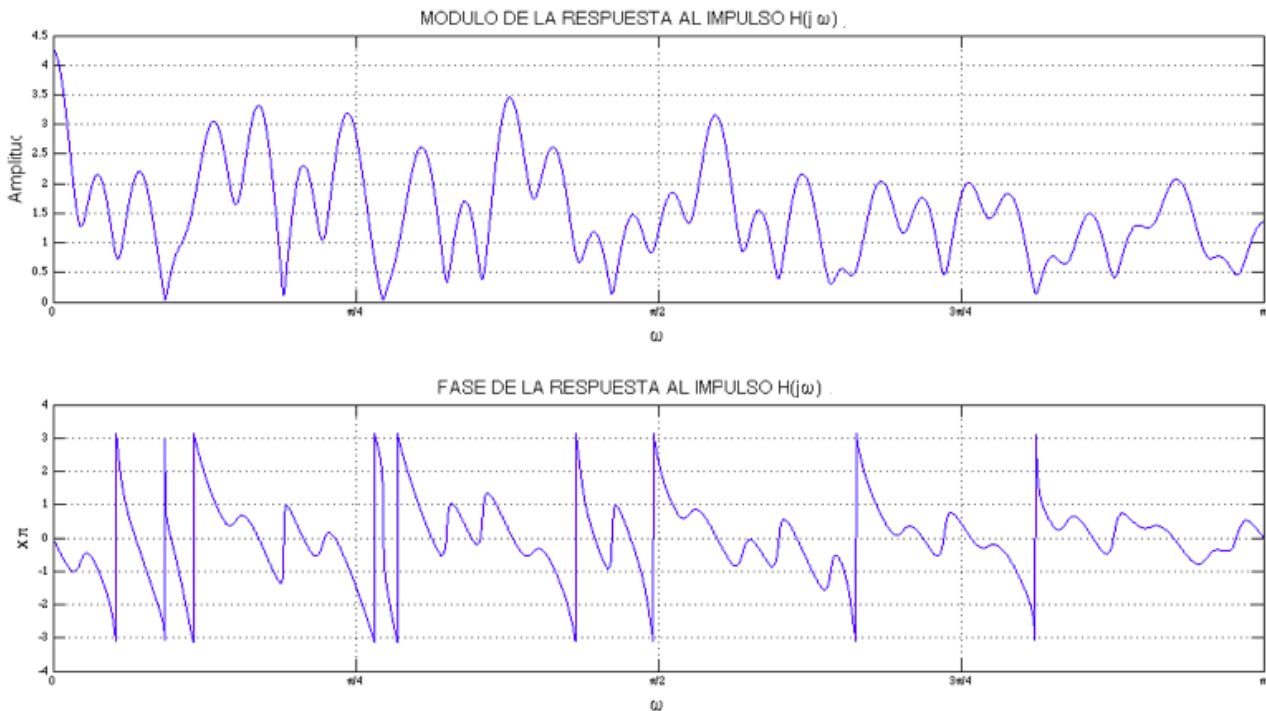


Figura 85. Respuesta al impulso Chorus, ejecución 2

○ **Resultados caso práctico:**

Al ejecutar el código con un archivo de audio de muestra, se produce el efecto chorus alterándolo completamente. Este efecto se puede modificar variando los parámetros:

- **g1, g2 y g3:** Controlan la ganancia del ruido y a su vez la intensidad del efecto. Sus valores máximos no deben superar la unidad, ya que se satura la señal de salida.
- **I:** Controla la ganancia de la señal de entrada, como en el caso anterior, su valor máximo no debe superar la unidad ya que saturaría la salida.
- **Width o Delay :** Controla el tamaño de buffer de retardo, que se traduce al rango de frecuencias en el varía el chorus, según las fórmulas (25) y (26). Al introducir un retardo mayor, el rango de variación será superior y la señal se convierte en ruido. Los valores recomendados están en torno a 0,5 ms. Si se utilizan valores inferiores, no se aprecia el efecto.

A continuación se representan las señales de entrada y salida del sistema para  $g_1 = g_2 = g_3 = I = 1$  y  $Width = 0.8$  ms:

Como se puede observar en la anterior gráfica, la señal  $y[n]$  empieza a retardarse a partir de los 100 ms respecto a  $x[n]$ . Esto sucede ya que el efecto chorus introduce retardos aleatorios, como se ha explicado anteriormente.

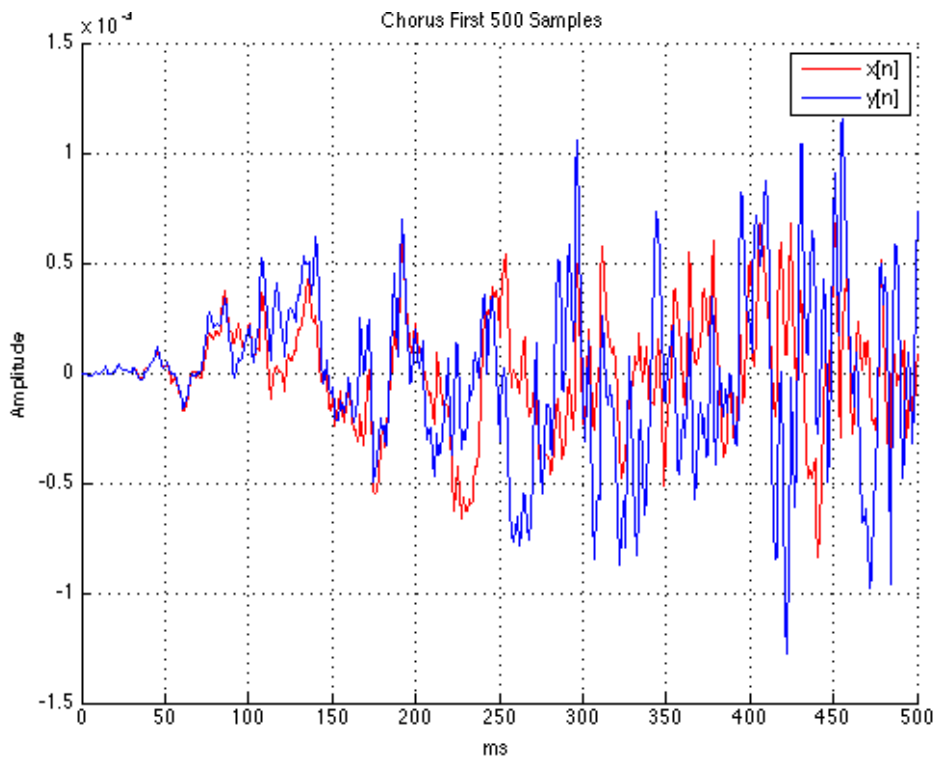


Figura 86. Primeras muestras de las señales  $x[n]$  e  $y[n]$ .

A continuación, se representa el espectro de la señal de entrada y de salida para los valores anteriormente citados:

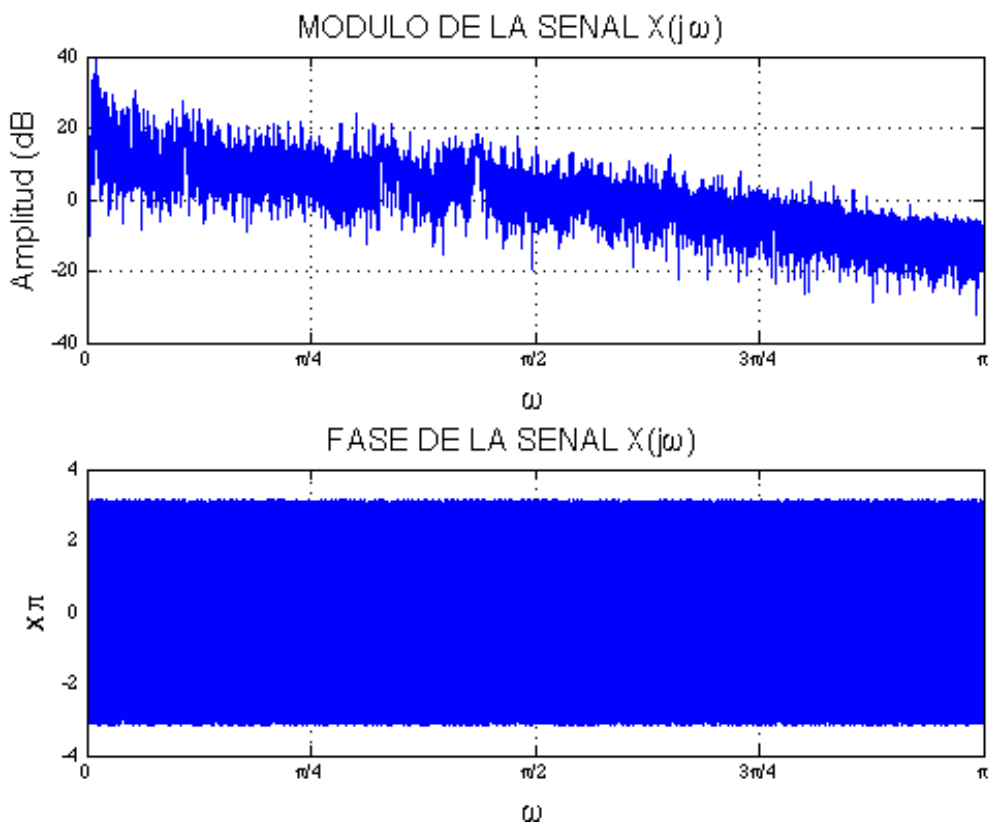


Figura 87. Modulo y fase de  $X(j\omega)$

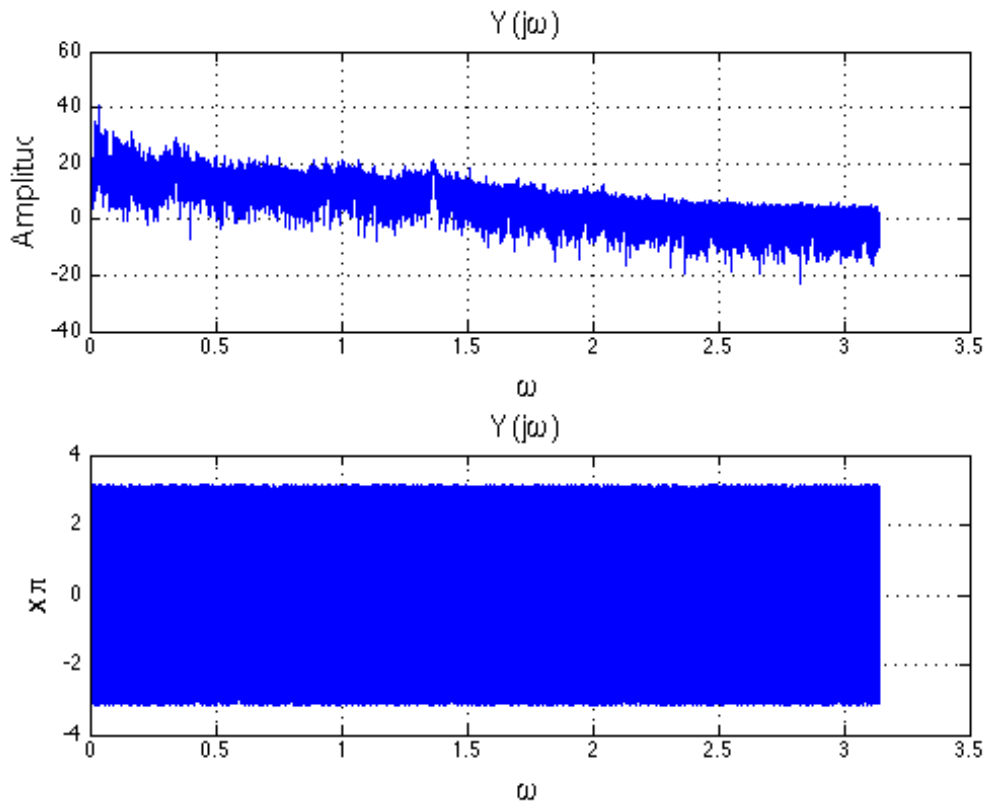


Figura 88. Módulo de la salida  $Y(j\omega)$

Si ampliamos el espectro y representamos la señal de entrada junto a la de salida:

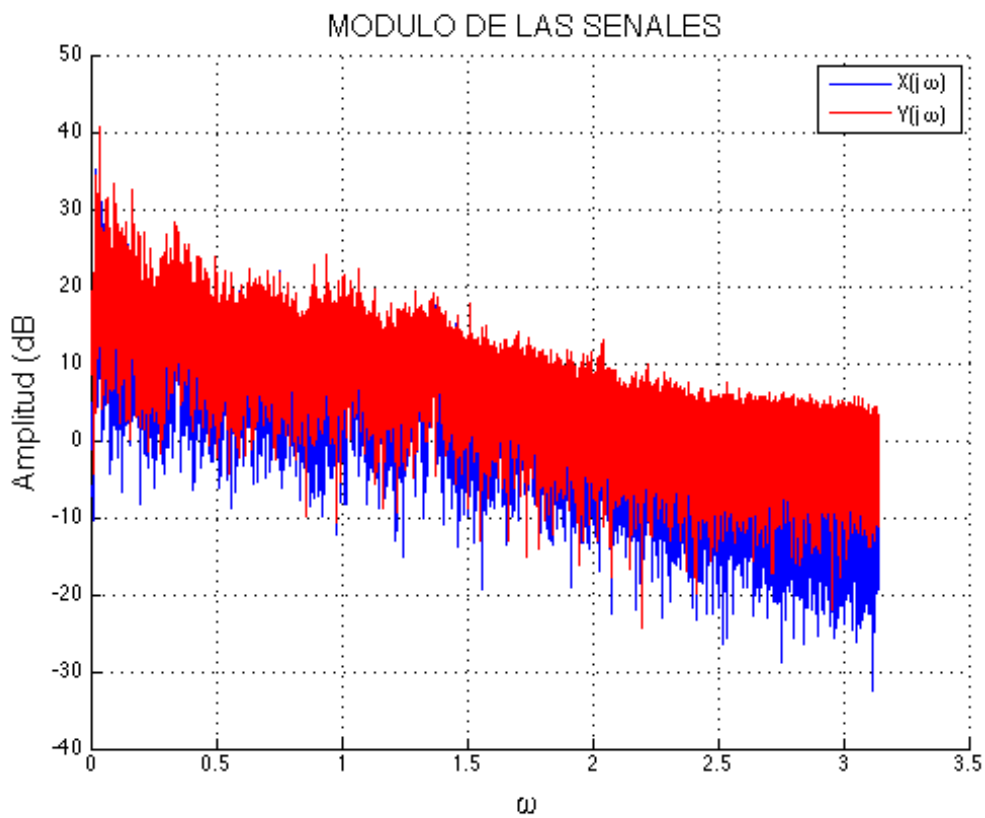
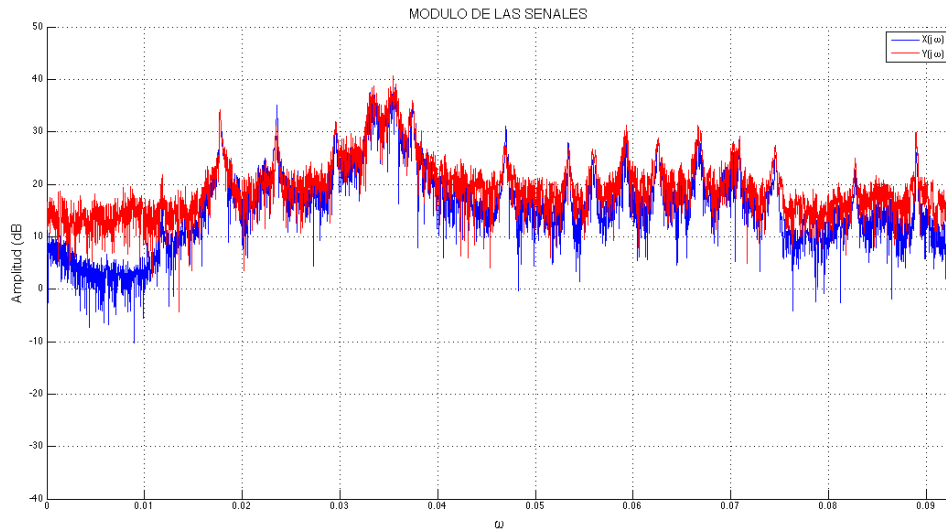


Figura 89. Módulo de  $X(j\omega)$  solapado con  $Y(j\omega)$



**Figura 90. Módulo ampliado de  $X(j\omega)$  solapado con  $Y(j\omega)$**

Como se ha mostrado anteriormente, la respuesta al impulso del sistema  $H(j\omega)$  es variable con cada ejecución, el número de lóbulos para las anteriores ejecuciones es muy alto y no altera de forma muy abrupta la forma de la señal de entrada  $X(j\omega)$ .

## 5.2. Efectos con filtros

### 5.2.1. Wah-Wah

#### ○ Introducción:

El efecto wah-wah consiste en un filtro paso banda dinámico, cuya frecuencia de corte varía con respecto al tiempo. El ancho de banda de estos filtros, generalmente suele ser estrecho y la variación de la frecuencia de corte se puede realizar manualmente o de forma automática, para el siguiente análisis y diseño se ha utilizado la segunda.

#### ○ Análisis:

El sistema se puede implementar como un mezclador que combina la señal de entrada más la señal de salida del filtro dinámico. Su diagrama de bloques es el siguiente:

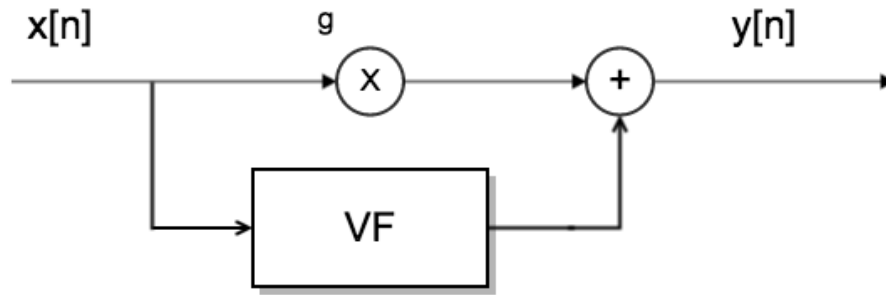


Figura 91. Diagrama Wah-Wah

El filtro variable (VF) tiene el siguiente diseño:

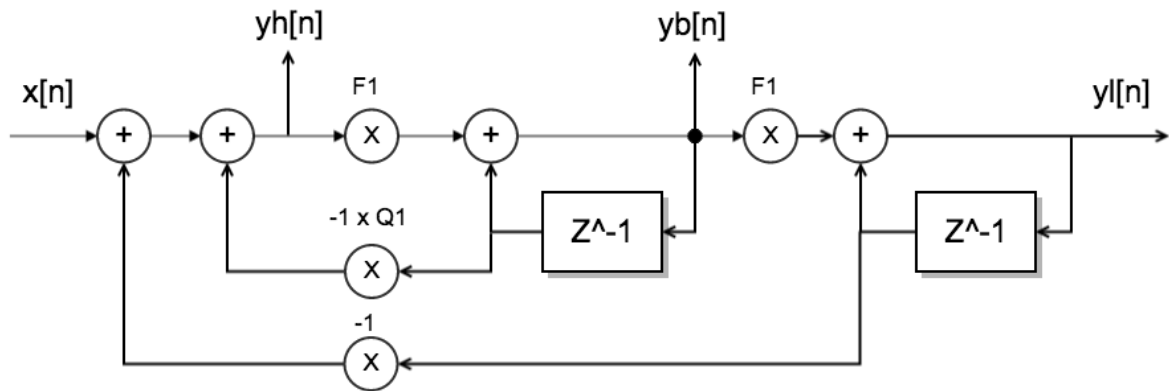


Figura 92. Filtro variable

Donde la salida de dicho filtro es  $y_b[n]$  que es combinación de las tres señales  $x[n]$ ,  $y_h[n]$  y  $y_b[n]$ . A continuación se muestra el análisis matemático del wah-wah.

$$y_l[n] = F_1 y_b[n] + y_l[n - 1] \quad \text{(Fórmula 27)}$$

$$y_b[n] = F_1 y_h[n] + y_b[n - 1] \quad \text{(Fórmula 28)}$$

$$y_h[n] = x[n] - y_l[n - 1] - Q_1 y_b[n - 1] \quad \text{(Fórmula 29)}$$

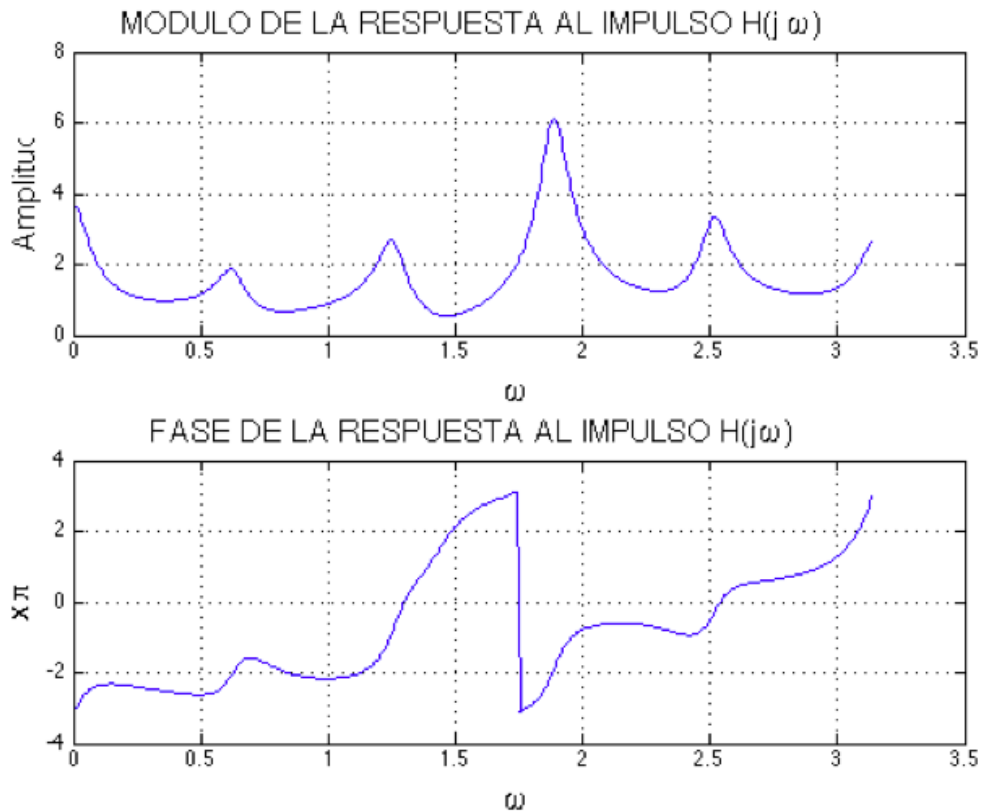
Donde las constantes  $F_1$  y  $Q_1$  permiten modificar la frecuencia de corte y el factor de amortiguamiento del efecto, respectivamente:

$$F_1 = 2 \sin\left(\frac{\pi f_c}{f_s}\right) \quad \text{(Fórmula 30)}$$

$$Q_1 = 2d \quad \text{(Fórmula 31)}$$



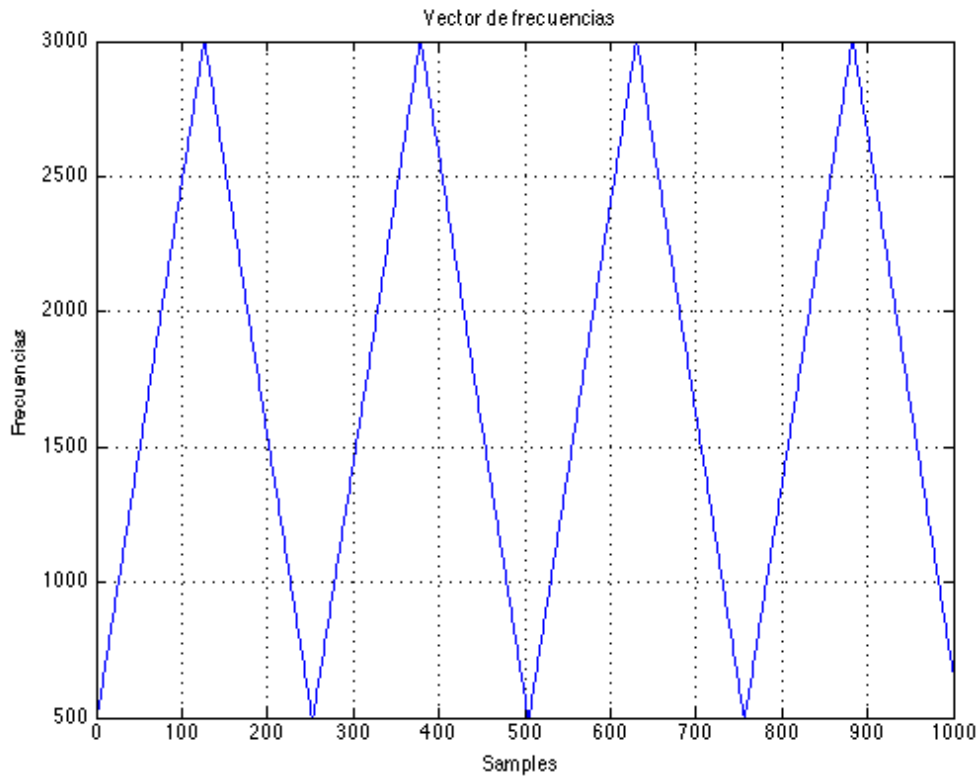
Al representar la respuesta al impulso  $H(j\omega)$  del sistema para  $g = 1$  y un rango de frecuencias de  $f_l = 500 \text{ Hz}$  a  $f_h = 3000 \text{ Hz}$  se obtiene la siguiente representación:



**Figura 93. Módulo y fase de  $H(j\omega)$**

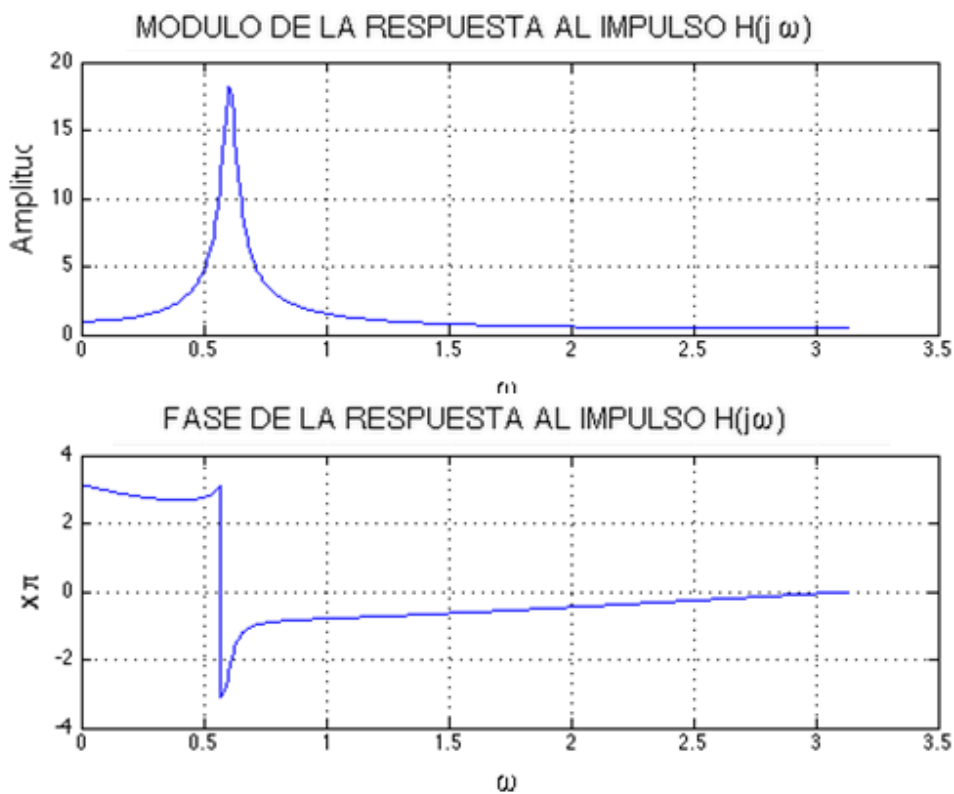
Como se puede observar, el sistema se comporta como un filtro con varios picos, pero en realidad, en cada instante de tiempo, se comporta como un filtro paso banda con un pequeño ancho de banda que varía.

En la siguiente gráfica, se aprecia que a medida que avanza el tiempo, la frecuencia de corte obtiene distintos valores. Este cambio de frecuencias se genera de forma lineal, primero empieza con un aumento de la frecuencia de corte desde  $f_l$  a  $f_h$ . Al finalizar, se recorre en sentido contrario el rango de frecuencias. Esta es la singularidad del efecto wah-wah.



**Figura 94. Rango de frecuencias recorrido**

A continuación, se muestran las respuestas al impulso del sistema en determinados instantes de tiempo, para observar que el filtro en dichos instantes es de pico.



**Figura 95. Módulo y fase de  $H(j\omega)$  para  $n = 300$**

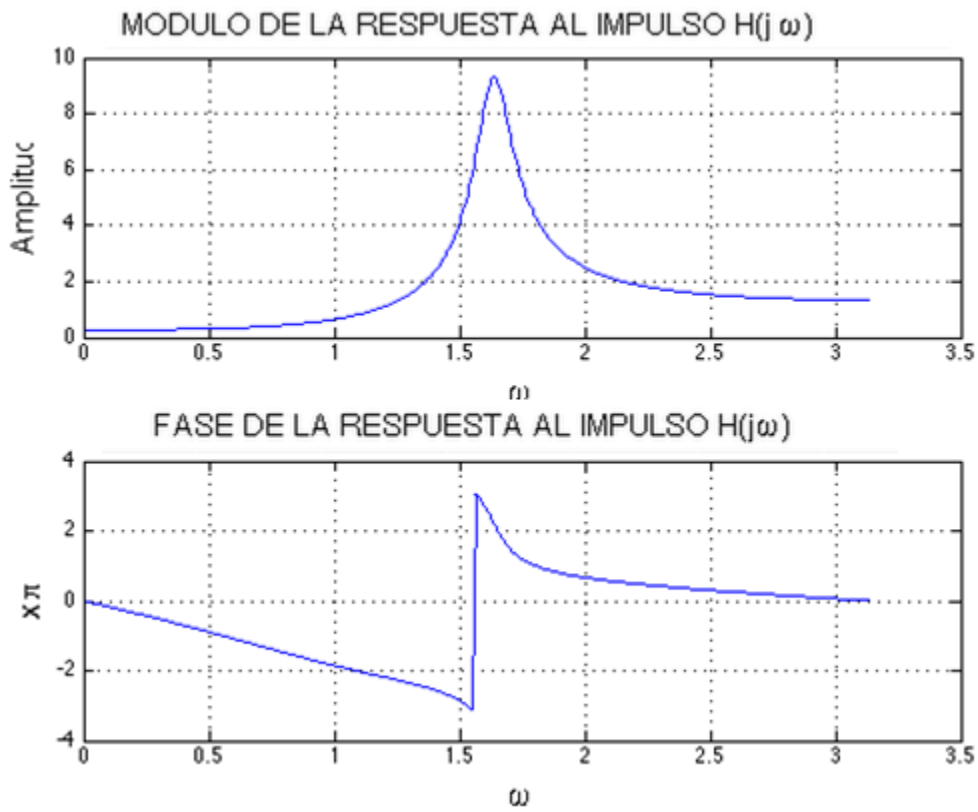


Figura 96. Módulo y fase de  $H(j\omega)$  para  $n = 126$

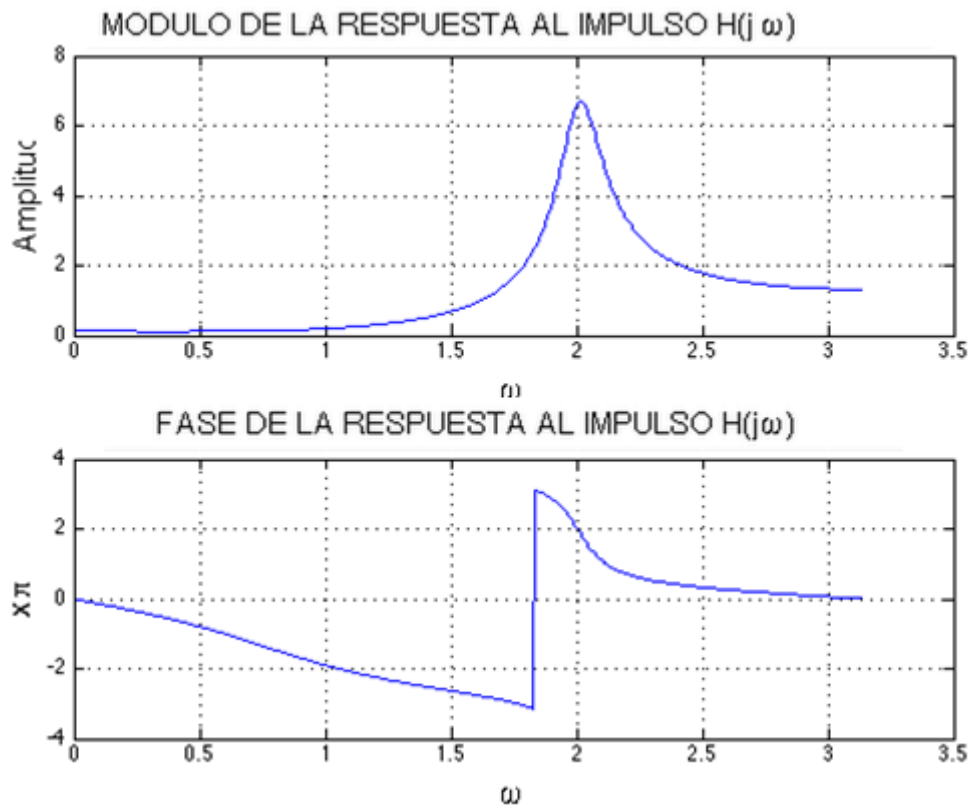


Figura 97. Módulo y fase de  $H(j\omega)$  para  $n = 101$

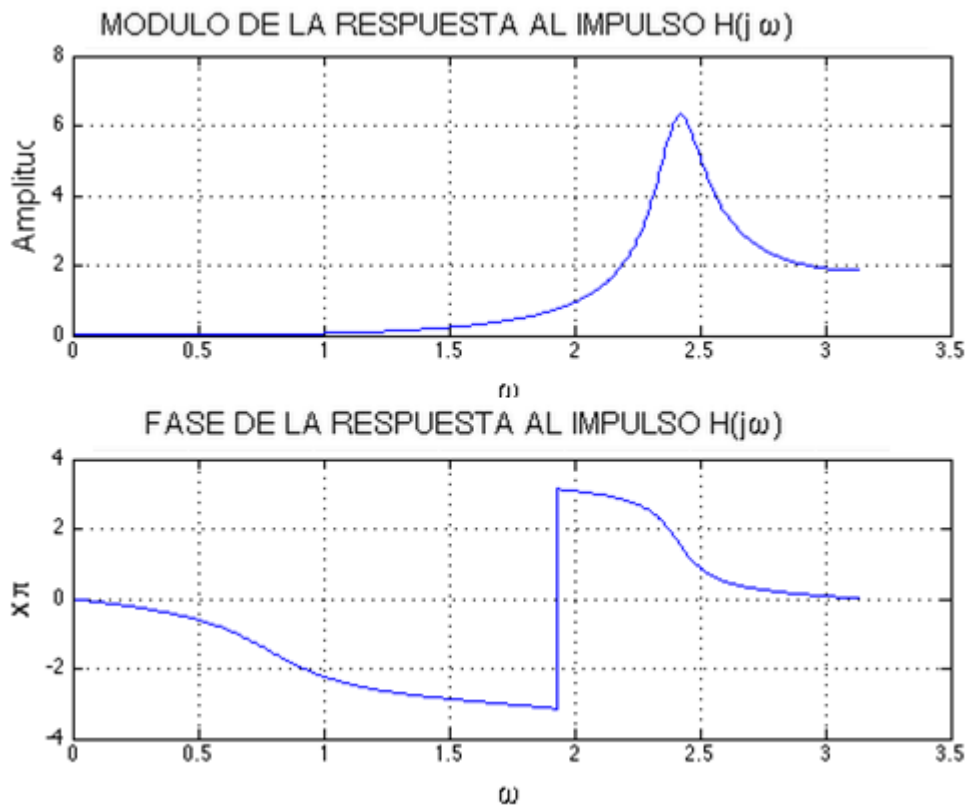


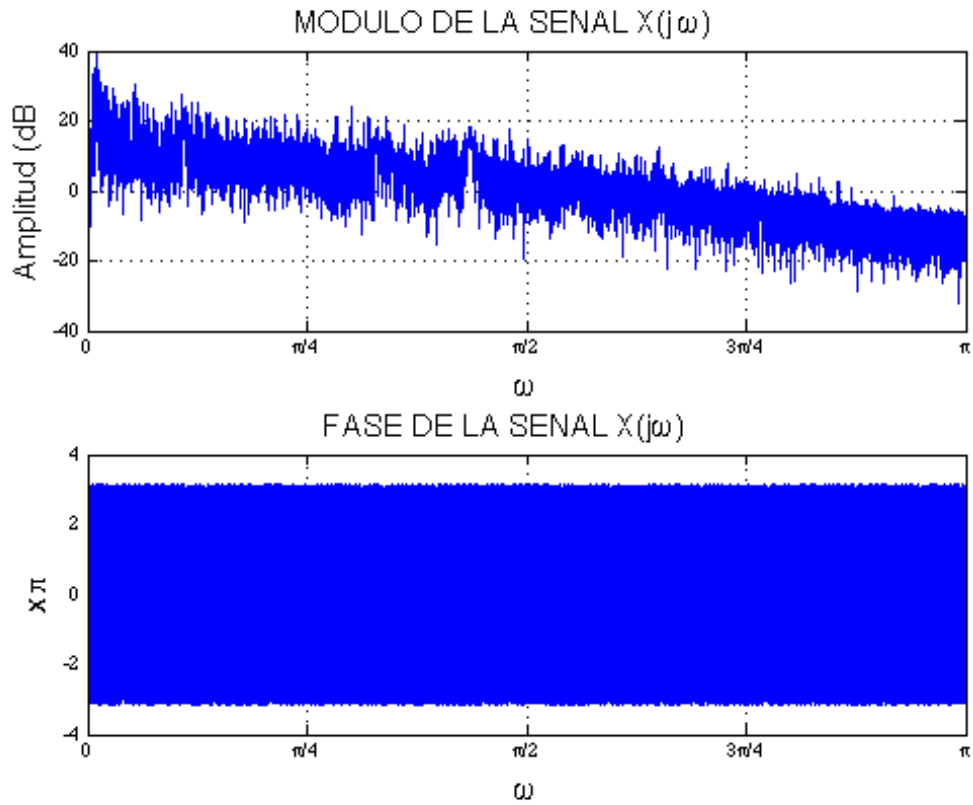
Figura 98. Módulo y fase de  $H(j\omega)$  para  $n = 76$

○ **Resultados caso prácticos:**

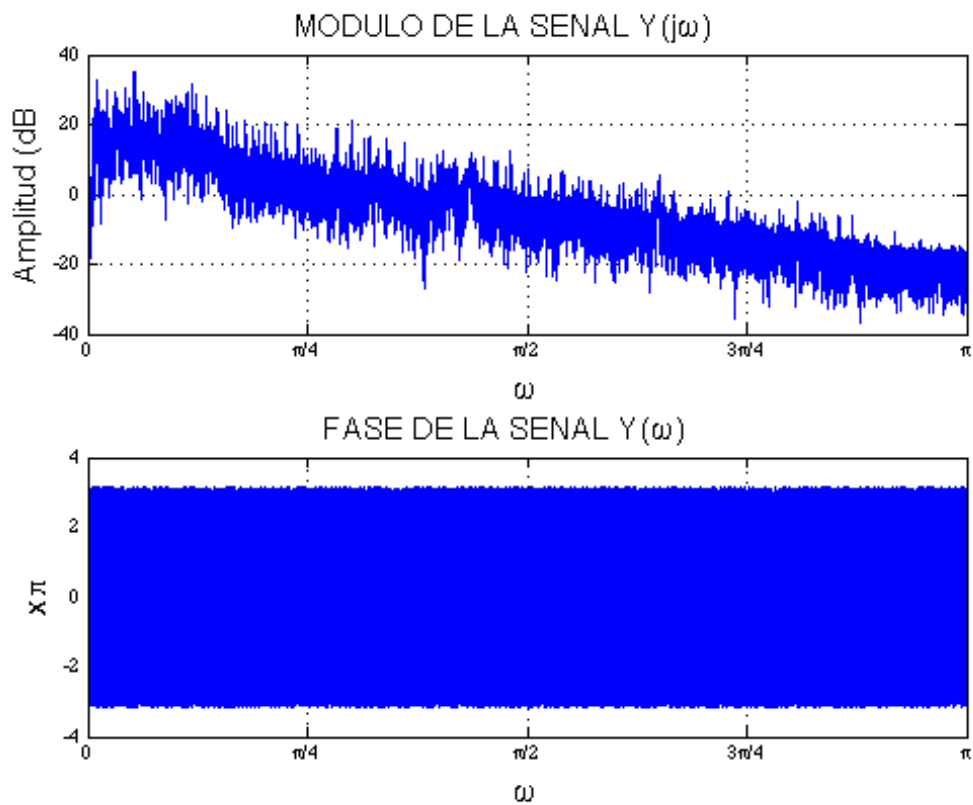
Para percibir este efecto, se dispone de un archivo de audio de muestra, el cual se introduce en el sistema y genera un vector de salida. Para realizar la ejecución se han dado los siguientes valores:

- Mínima frecuencia de corte del filtro:  $f_l = 500 \text{ Hz}$ .
- Máxima frecuencia de corte del filtro:  $f_h = 3000 \text{ Hz}$ .
- Factor para obtener el paso de frecuencias:  $f_w = 2000 \text{ Hz}$ .
- Ganancia de la señal de entrada:  $g = 1$ .
- Factor de amortiguamiento:  $d = 0.05$ .

A continuación se representa el espectro de la señal de entrada y de salida del wah-wah.



**Figura 99. Módulo y fase de  $X(j\omega)$**



**Figura 100. Módulo y fase de  $Y(j\omega)$**

Si ampliamos el espectro y representamos la señal de entrada junto a la de salida:

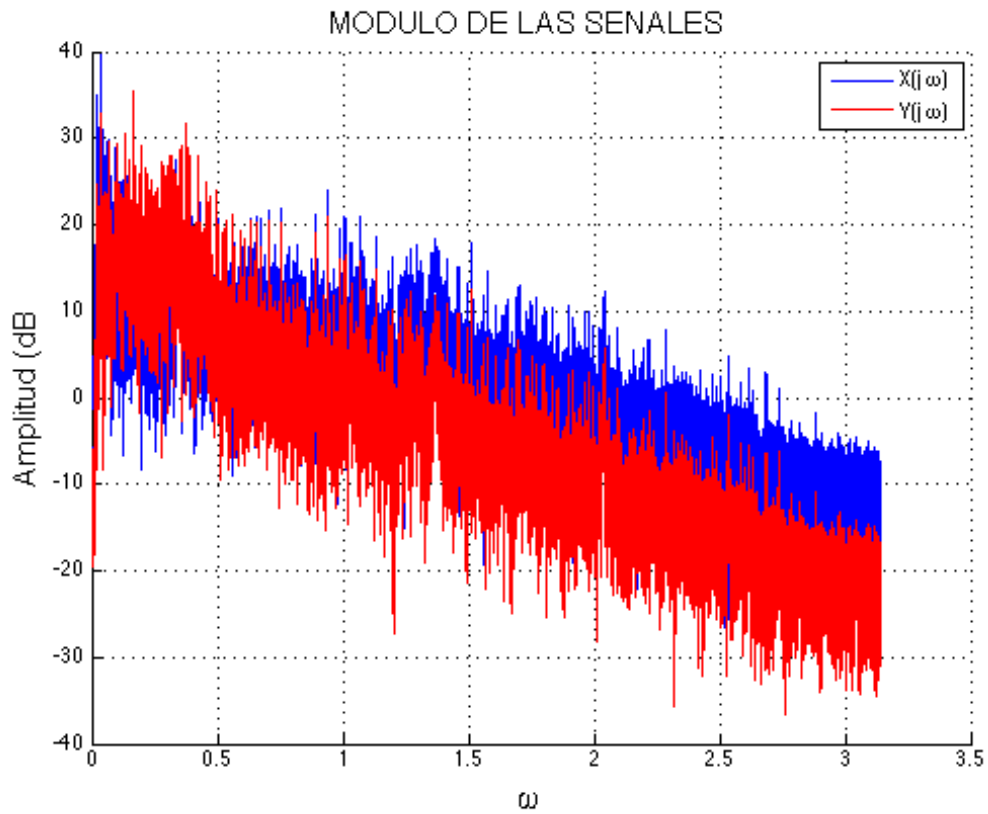


Figura 101. Módulo de  $X(j\omega)$  solapado con  $Y(j\omega)$

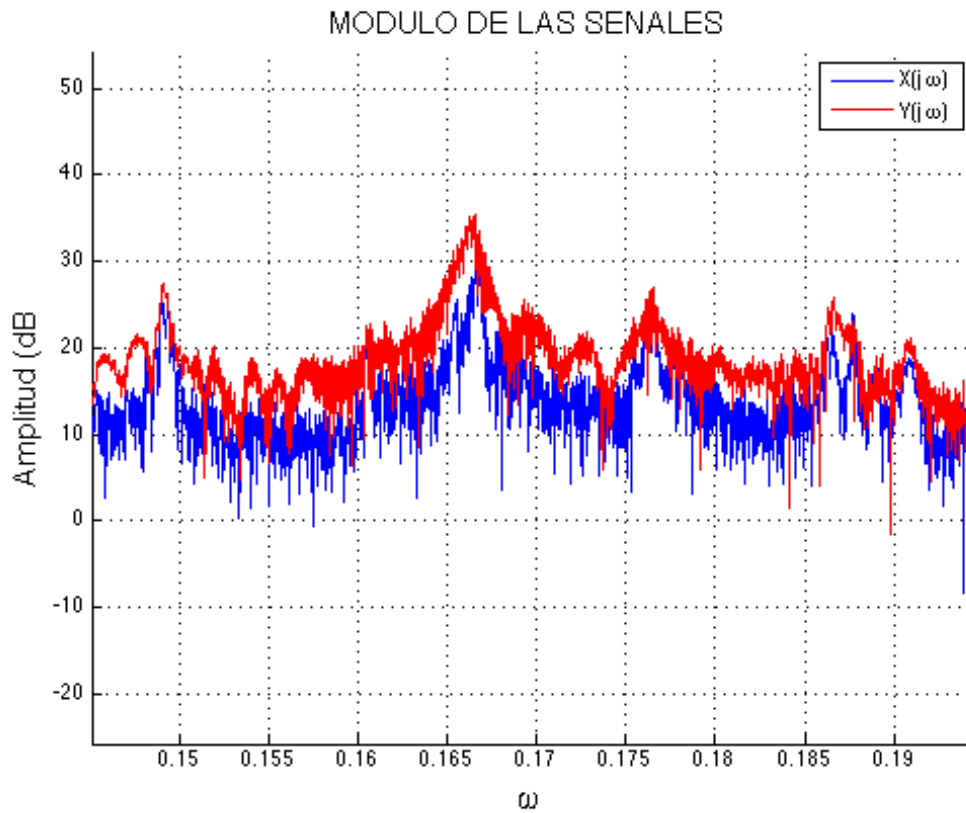
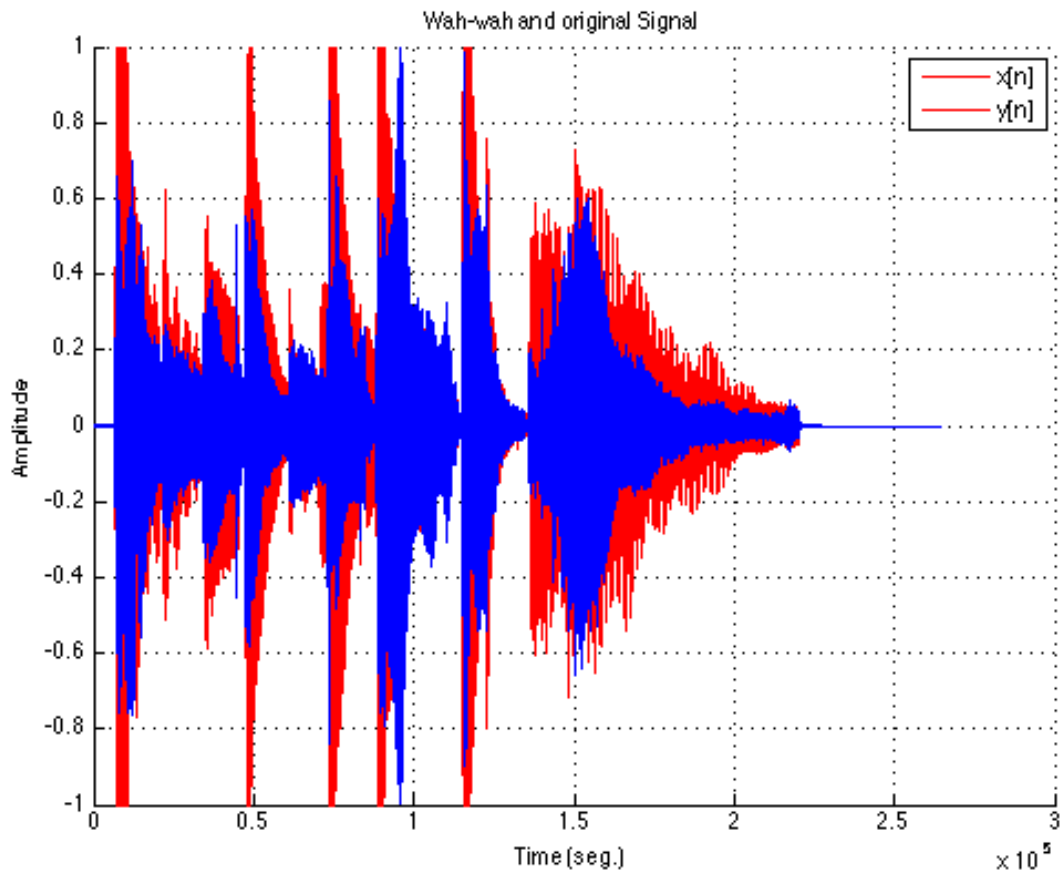


Figura 102. Módulo ampliado de  $X(j\omega)$  solapado con  $Y(j\omega)$

Se puede observar que los espectros son parecidos ya que el sistema solo filtra la señal original por un filtro dinámico que modifica muchas frecuencias del espectro y globalmente realza la ganancia. Sin embargo, al ejecutar el código, gracias a la función `audioplayer` de Matlab™, se puede apreciar que la señal original adquiere el efecto wah-wah y se filtra dinámicamente.

Si se observan las señales en el tiempo, como se puede observar en la siguiente gráfica, la señal  $y[n]$  también queda modificada con respecto  $x[n]$ . En las siguientes gráficas se muestran las señales de entrada y salida solapadas.



**Figura 103. Señales  $x[n]$  e  $y[n]$**

A continuación se muestran ambas señales ampliadas para ver el efecto del filtro dinámico:

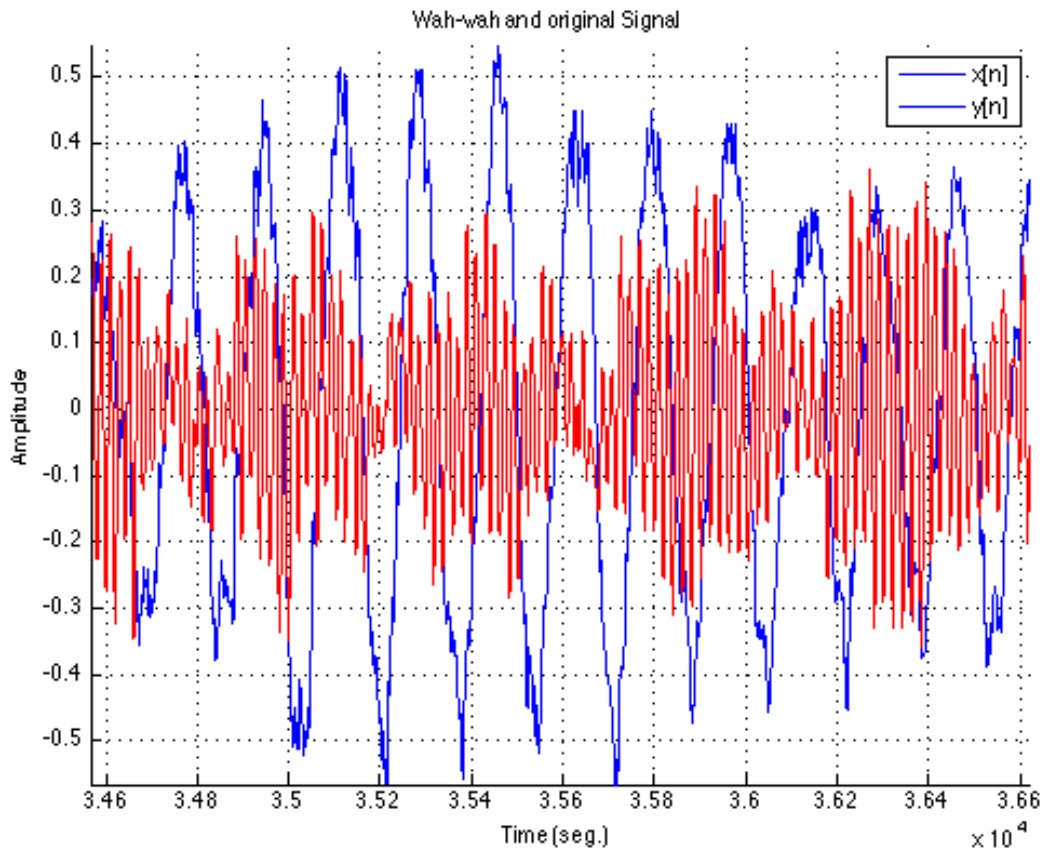


Figura 104. Señales  $x[n]$  e  $y[n]$  ampliadas.



## 5.3. Efectos de procesamiento de dinámica

### 5.3.1 Compresor.

#### ○ Introducción:

Los efectos de procesamiento de dinámica se utilizan para realzar, controlar o corregir el rango dinámico de la señal de entrada. En este apartado se van a abordar el más típico, el compresor. Además, se ha desarrollado la función de Matlab™ `compress` que nos permite aplicar el efectos a una señal de entrada cualquiera.

El compresor se encarga de reducir el rango dinámico de una señal, se utiliza para aumentar el nivel global de la señal de entrada sin exceder el rango de amplitudes permitidos a la salida, o comúnmente llamado, la sonoridad.

#### ○ Análisis:

A continuación, se muestra la ecuación de la ganancia de salida que define este efecto:

$$G = \begin{cases} CS(CT - X) & \text{si } X > CT \\ 0 \text{ dB} & \text{si } X \leq CT \end{cases} \quad \text{(Fórmula 32)}$$
$$= \min(0, CS(CT - X))$$

El sistema se puede implementar como un sistema dinámico que recalcula la ganancia de salida en función del nivel  $X(\text{dB})$  de la señal de entrada, todo ello en función de los parámetros  $CT$ ,  $CS$  de la ecuación. Los cuales nos permiten variar:

- $CT$ : Threshold del compresor: Este parámetro indica el umbral de compresión de la señal de entrada y siempre viene dada en dB, sus valores típicos han de ser menores que 0 dB.
- $CS$ : Slope del compresor: Indica el valor de reducción de la ganancia, de ahí sus valores típicos que deben ser entre 0 y 1.

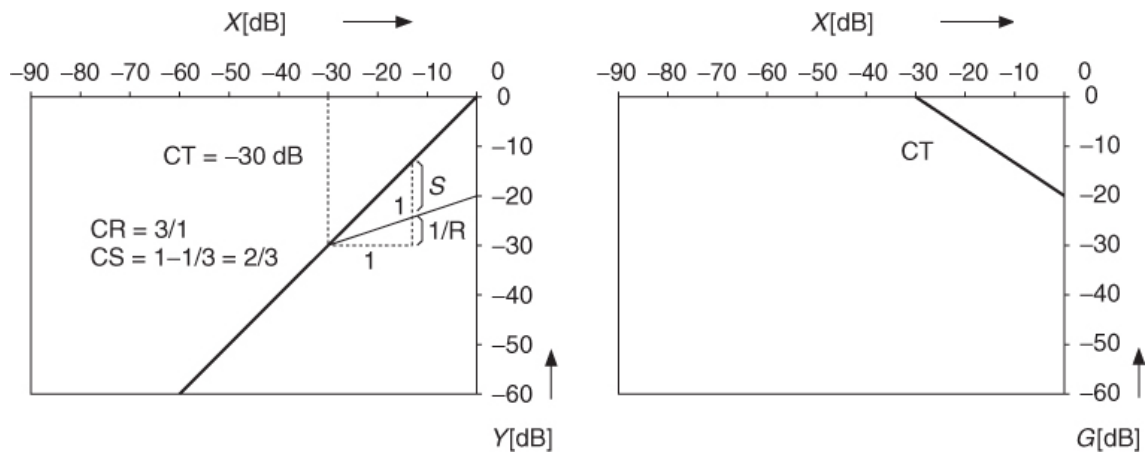


Figura 105. Ejemplo de curva estática [Zölder11]

En unidades naturales, la ganancia que se aplica a la señal de entrada se calcula con la siguiente expresión:

$$g[n] = \min\left(1, \left(\frac{x_{RMS}(n)}{ct^2}\right)^{-CS/2}\right) \quad \text{(Fórmula 33)}$$

El diagrama de bloques del sistema sería el siguiente:

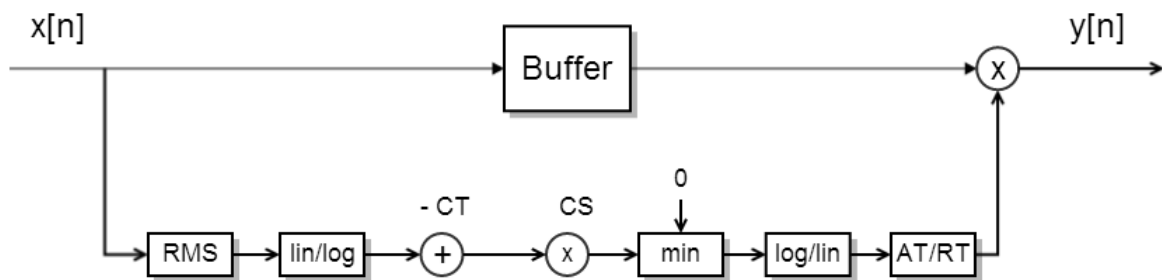


Figura 106. Diagrama de bloques Compresor

El último componente que se muestra en el diagrama de bloques es el Attack y Release del compresor. Este bloque controla la velocidad de acción del efecto, dependiendo de los valores que se den a estas dos constantes, actuará unas milésimas de segundo antes o después, pero en las muestras de audio utilizadas no es apreciable, por lo que no se estudiará su efecto.

A continuación, se muestra de forma visual el funcionamiento del compresor introduciendo una sinusoidal a la entrada, para analizar el comportamiento y observar que como actúa dependiendo de los parámetros que se le pase.

En la siguiente gráfica se puede observar la señal de entrada  $x[n]$  en rojo y la señal de salida  $y[n]$  en azul, comprimida. Para este caso se han escogido los siguientes valores:

- $CT = -20 \text{ dB}$
- $CS = 0.5$

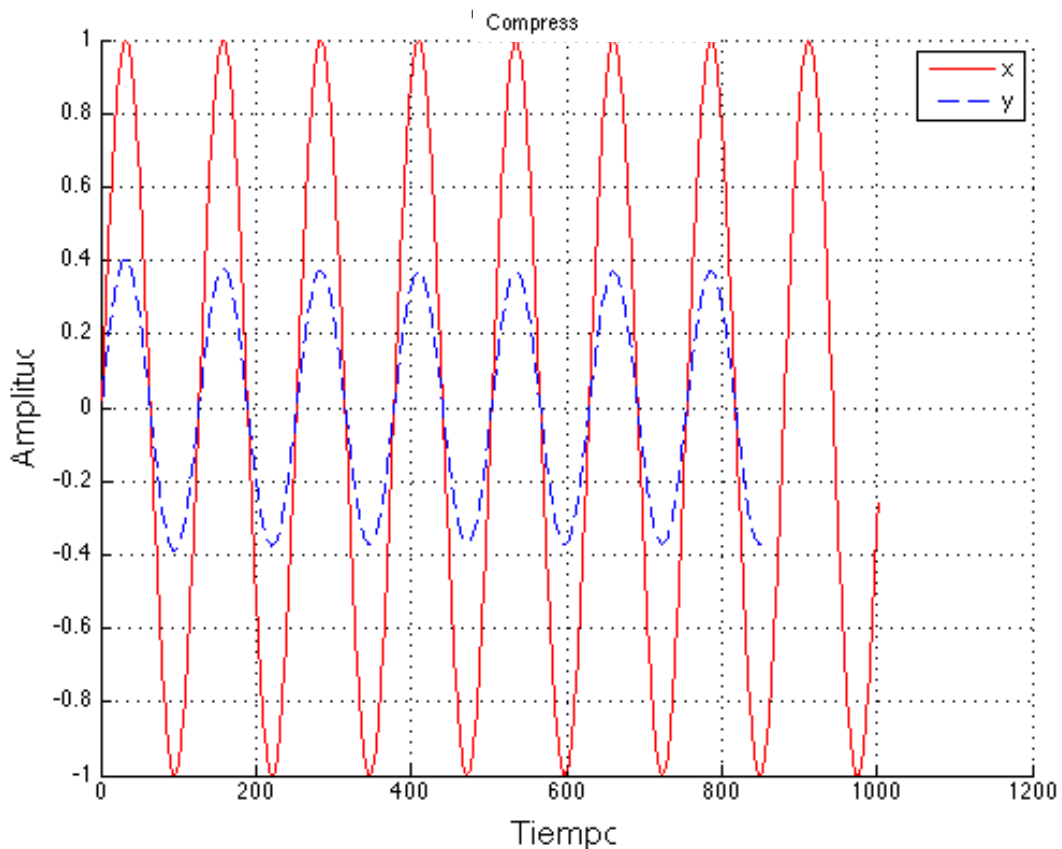
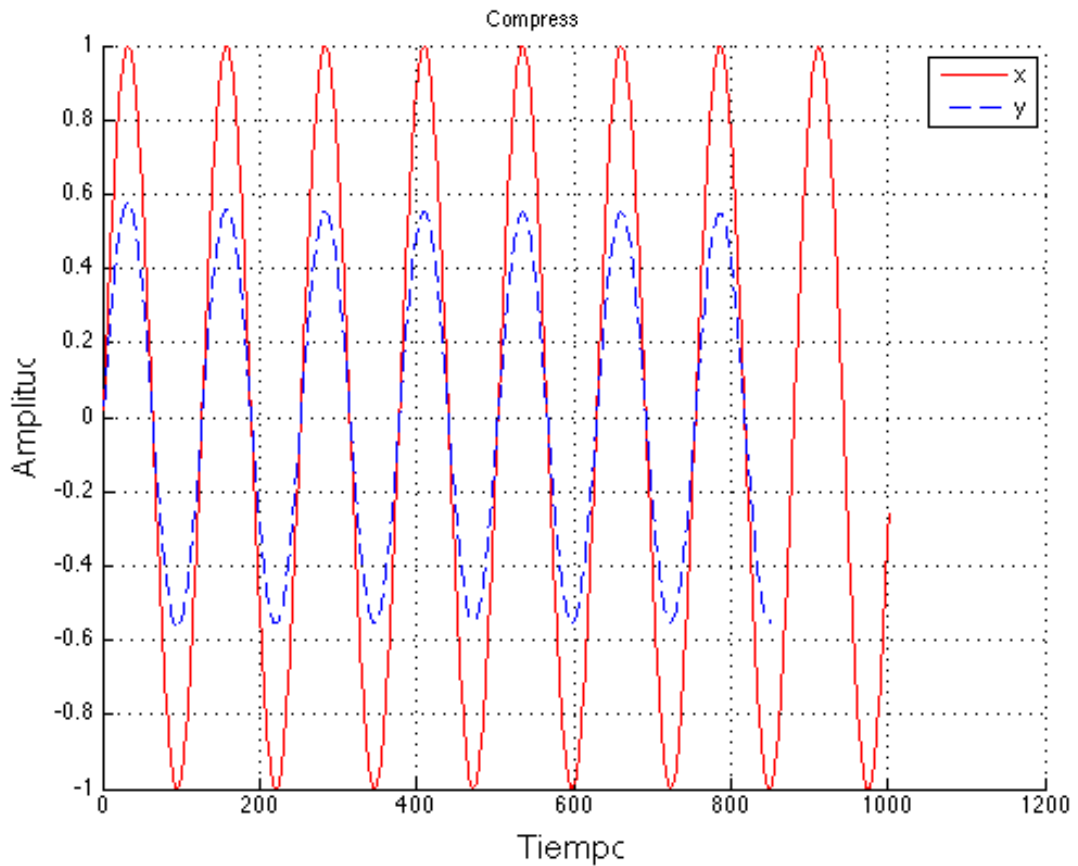


Figura 107. Señales  $x[n]$  e  $y[n]$   $CT = -20 \text{ dB}$   $CS = 0.5$ .

Como se puede observar, la señal de entrada queda comprimida. En el caso siguiente, se reduce el Slope del compresor:

- $CT = -20 \text{ dB}$
- $CS = 0.3$

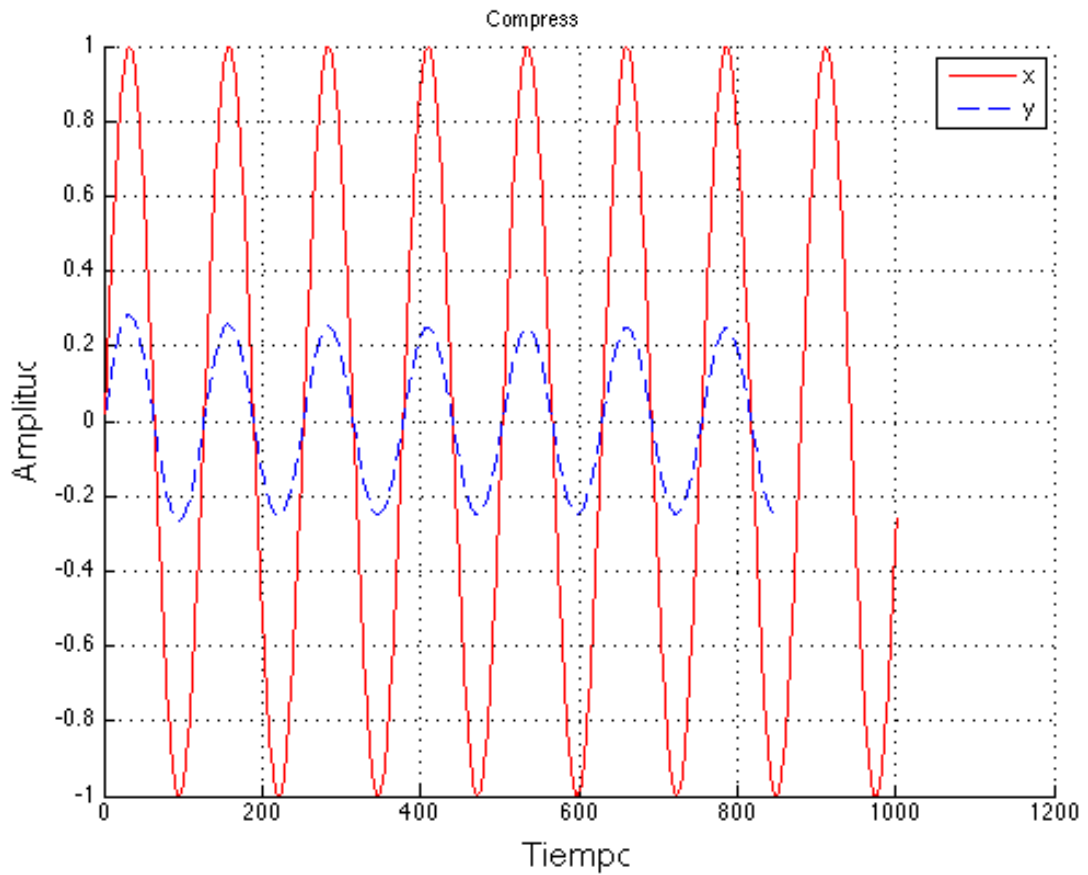


**Figura 108. Señales  $x[n]$  e  $y[n]$   $CT = -20$  dB  $CS = 0.3$ .**

Como se puede observar, la señal sigue comprimida, pero su amplitud ha aumentado ya que el slope del compresor controla el valor de reducción de la ganancia.

Si aumentamos el slope CS del compresor, la señal debería quedar más comprimida, como en la siguiente gráfica se muestra:

- $CT = -20$  dB
- $CS = 0.7$



**Figura 109. Señales  $x[n]$  e  $y[n]$   $CT = -20$  dB  $CS = 0.7$ .**

En los siguientes casos se muestra el efecto en la señal de entrada al aumentar o reducir el Threshold, evidentemente al aumentar este valor el umbral de compresión se incrementará (Figura 108) y al reducirlo, disminuirá (Figura 109).

- $CT = -30 \text{ dB}$
- $CS = 0.5$

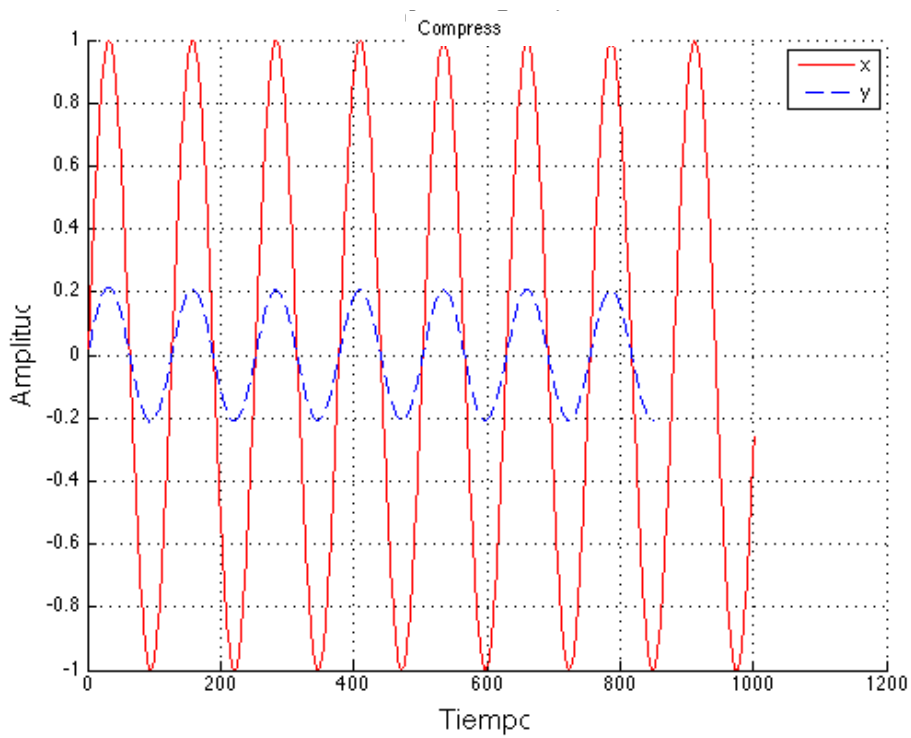


Figura 110. Señales  $x[n]$  e  $y[n]$   $CT = -30 \text{ dB}$   $CS = 0.5$

- $CT = -10 \text{ dB}$
- $CS = 0.5$

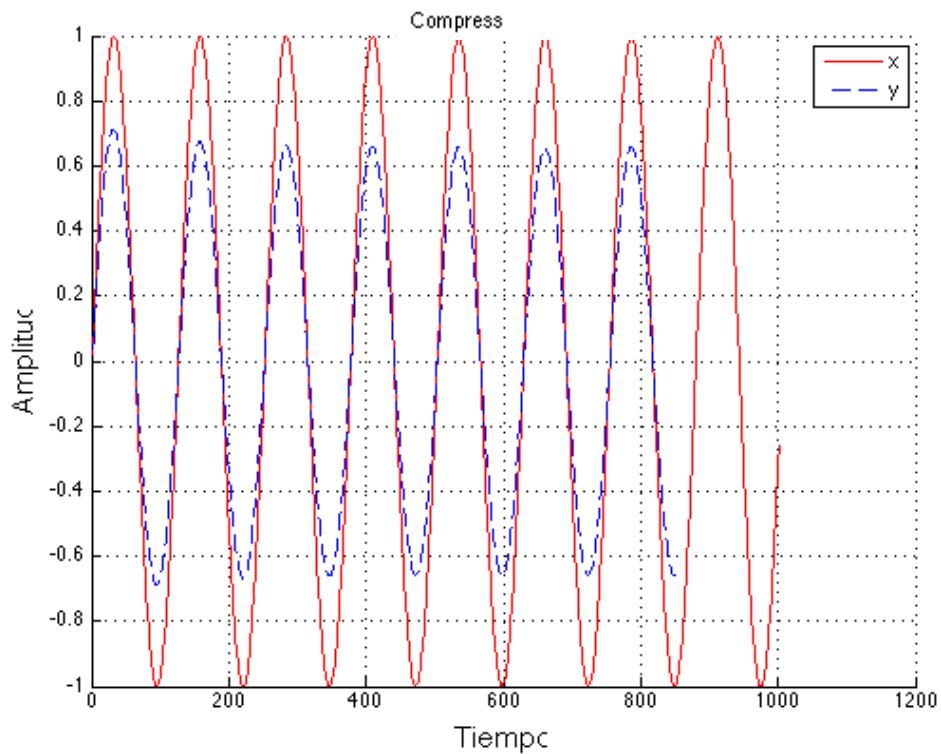


Figura 111. Señales  $x[n]$  e  $y[n]$   $CT = -10 \text{ dB}$   $CS = 0.5$

### Resultados caso práctico:

Para percibir este efecto, se dispone de un archivo de audio de muestra, el cual se introduce en el sistema y genera un vector de salida. Para realizar la ejecución se han dado los siguientes valores:

- $CT = -20 \text{ dB}$  .
- $CS = 0.5$

A continuación se representa la señal de entrada  $x[n]$  en azul y de salida del compresor  $y[n]$  en rojo:

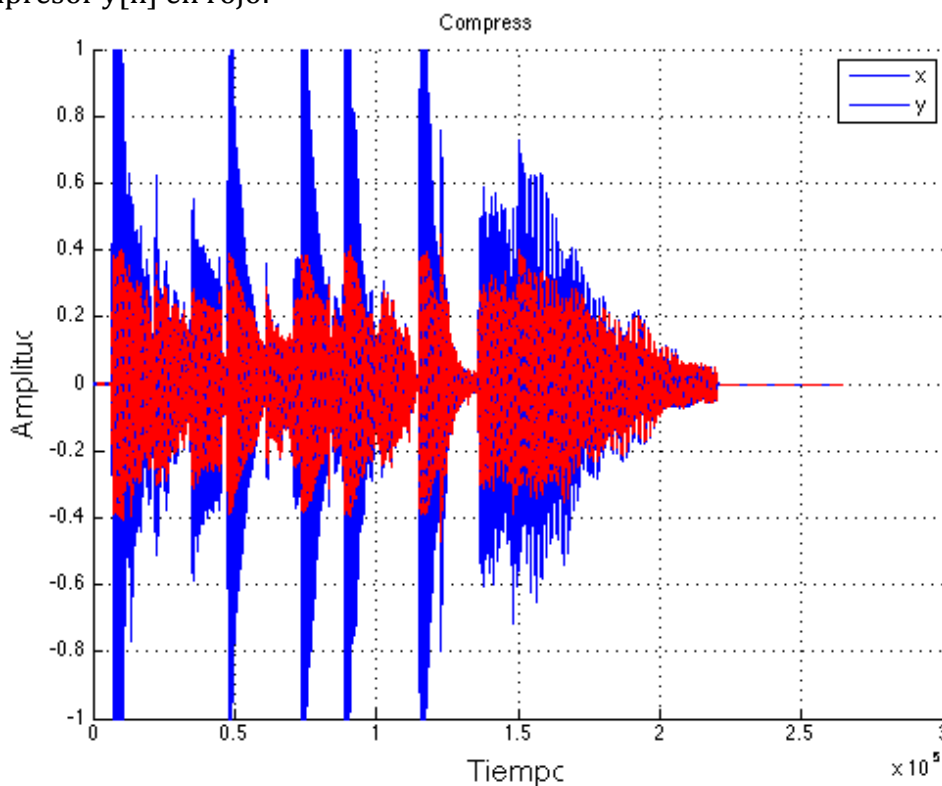
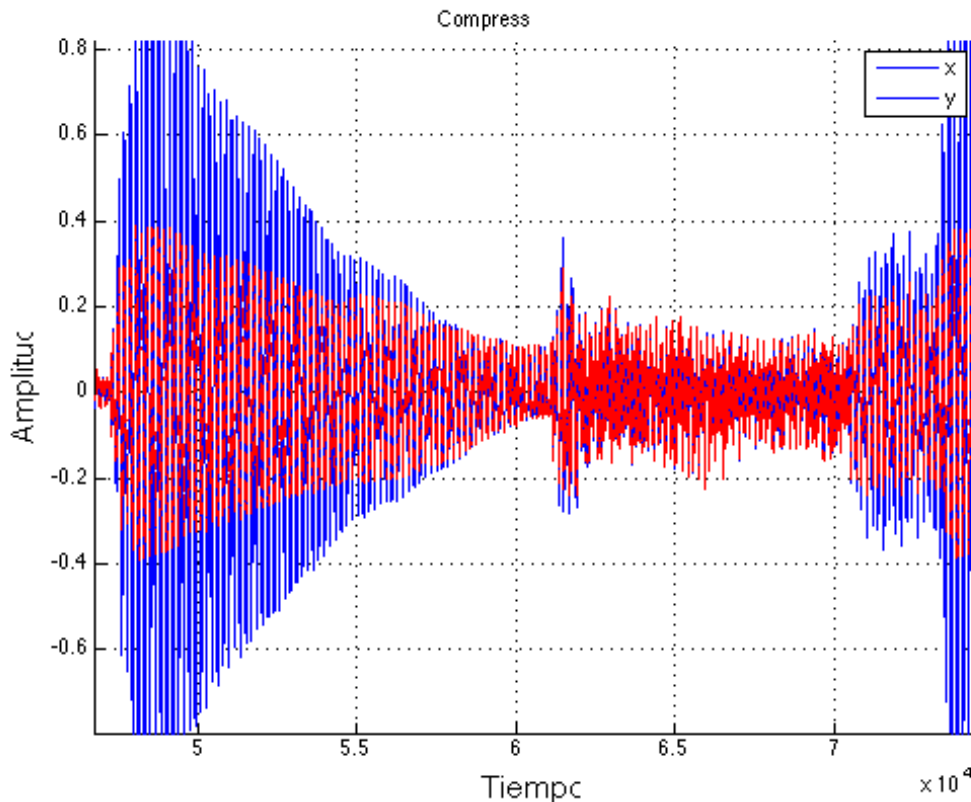


Figura 112. Señales  $x[n]$  e  $y[n]$

Si ampliamos las señales justo en una transición de un máximo, se puede observar que la señal de salida no tiene las mismas transiciones abruptas como la entrada y los máximos suelen ser más homogéneos en amplitud.



**Figura 113. Señales  $x[n]$  e  $y[n]$  ampliadas**

En las anteriores gráficas, se puede observar que la señal de entrada queda reducida en amplitud, pero sobre todo, sus máximos y mínimos no tienen transiciones tan abruptas, como se ha comentado anteriormente.

Este efecto es muy utilizado en post-procesado de audio, ya que evita tener una pista de audio poco homogénea en amplitud y que tenga picos tan muy altos que tapen las demás pistas de audio. También se usa en entrada de micrófonos para evitar picos en la adquisición de la voz.

## 5.4. Distorsión

### 5.4.1. Tube screamer

- **Introducción:**

Una de las familias de efectos más utilizadas en guitarra eléctrica es la distorsión u overdrive. Dicho efecto proporciona un sonido muy característico y peculiar que comenzó a ser popular en los años 60'. En esta sección se analizará y desarrollará en Matlab™ una réplica del efecto analógico de overdrive Tube Screamer, efecto



comercializado por la marca Ibanez™ y popularizado por muchos guitarristas como Stevie Ray Vaughan.



Figura 114. Pedal de efectos Tube Screamer

El Tube Screamer es un circuito analógico que distorsiona la señal de entrada. El núcleo central del circuito del Tube Screamer que realiza el overdrive de la señal de entrada, se compone de un amplificador operacional con realimentación negativa, en cuya rama de realimentación se encuentra un par de diodos. En el siguiente apartado analizaremos este comportamiento y las características del overdrive.

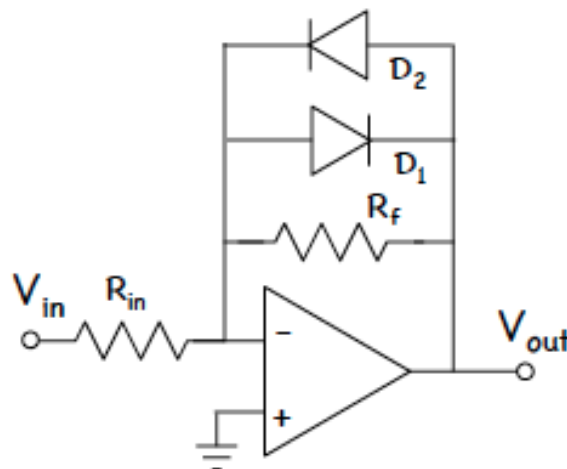


Figura 115. Circuito Distorsión TubeScreamer

○ **Análisis:**

La función matemática que define el comportamiento del circuito anterior es la siguiente:

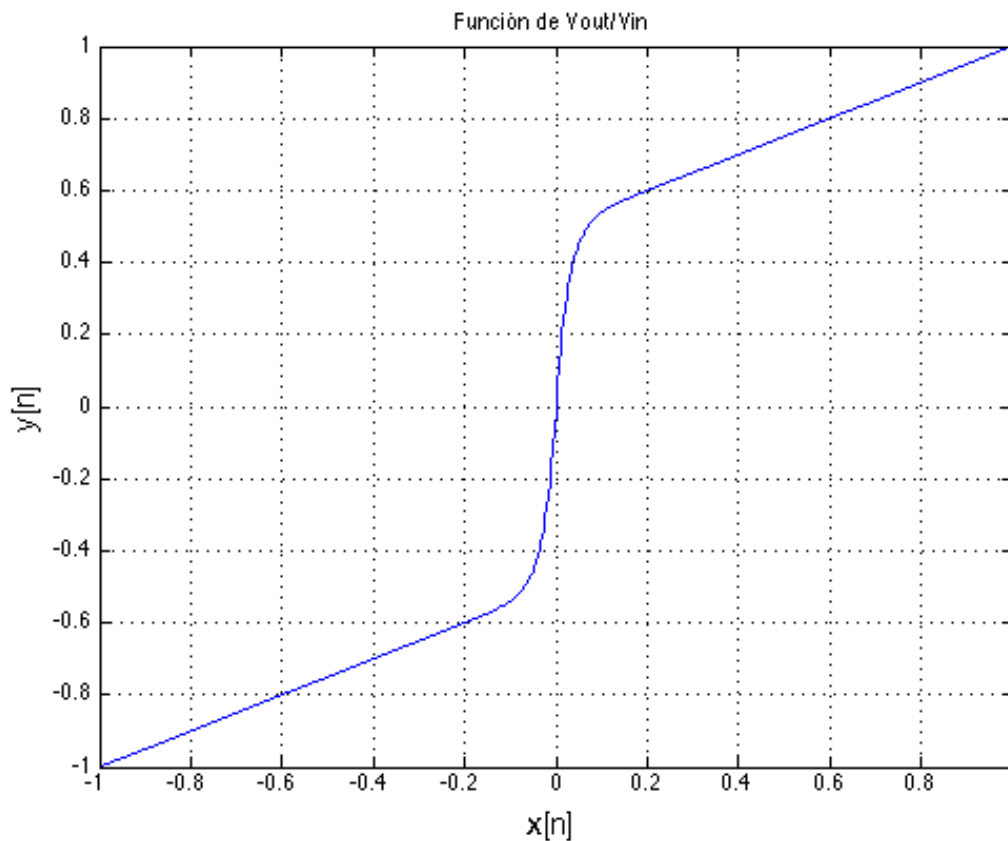
- Si D1 conduce:

$$x[n] = -R_{in} \left[ I_0 \left( e^{\frac{-y[n]}{V_T}} - 1 \right) - \frac{y[n]}{R_f} \right] \quad \text{(Fórmula 34)}$$

- Si D2 conduce:

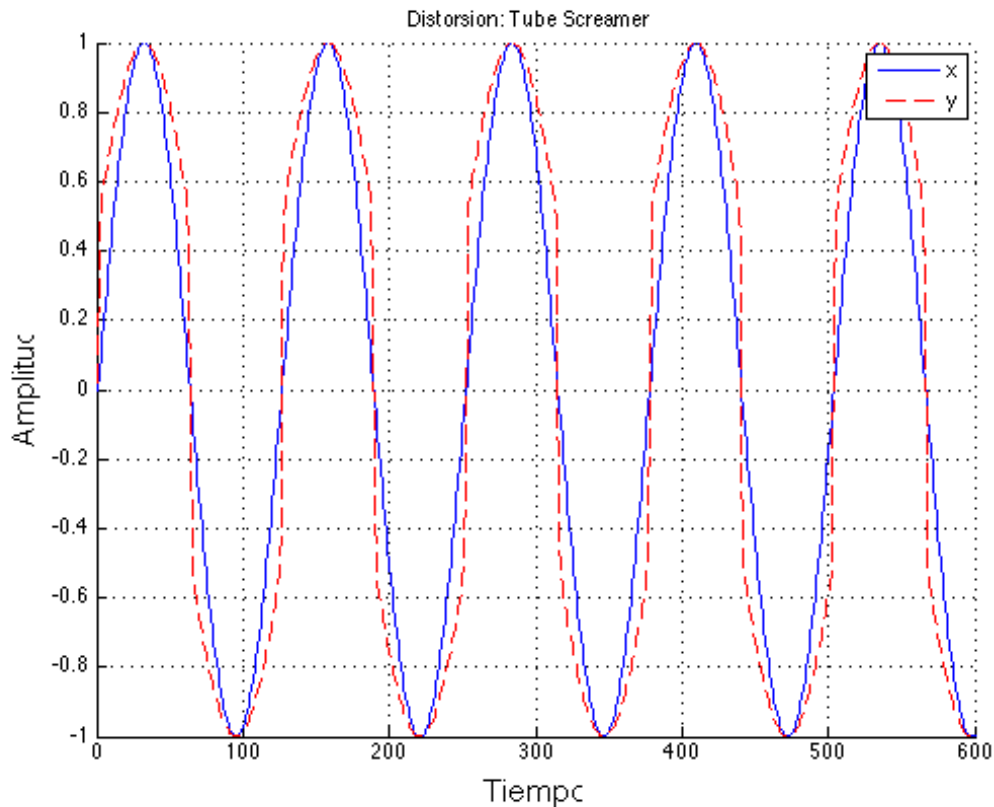
$$x[n] = R_{in} \left[ I_0 \left( e^{\frac{y[n]}{V_T}} - 1 \right) + \frac{y[n]}{R_f} \right] \quad \text{(Fórmula 35)}$$

A la hora de resolver esta ecuación se puede observar que no se puede dejar en función de la salida  $y[n]$ . La solución propuesta, es utilizar un rango fijo de valores de entrada e interpolar dicha función. En Matlab™ se ha utilizado la función *interp1*, y la curva característica de la señal de salida frente a la entrada para los siguiente valores  $R_{in} = 50\Omega$   $R_f = 100\Omega$   $I_0 = 0.01A$   $V_T = 0.026v$  se muestra a continuación.



**Figura 116. Señal de salida  $y[n]$  respecto a  $x[n]$**

Para poder observar el efecto que produce a la señal de entrada el Tube Screamer se utiliza como señal de entrada una sinusoidal, se representan las señales de salida respecto a todos los posibles valores de entrada, (recordemos que al ser señales de audio el valor máximo debe de ser 1 y el mínimo -1)  $R_{in} = 50\Omega$   $R_f = 100\Omega$   $I_0 = 0.01A$   $V_T = 0.026v$



**Figura 117. Señales  $y[n]$   $x[n]$  en tiempo**

Como se puede observar, la señal de salida para los anteriores valores queda ligeramente modificada en la pendiente debido a la transición que hay en torno a los valores de  $x[0]$ .

A continuación, se va a estudiar el efecto que tiene sobre la señal de entrada modificando solo uno de los parámetros.

- **Variación de  $R_{in}$ :**

Si se aumenta el valor de la resistencia de entrada  $R_{in} = 100 \Omega$  la curva característica del circuito y las señales son las siguientes:

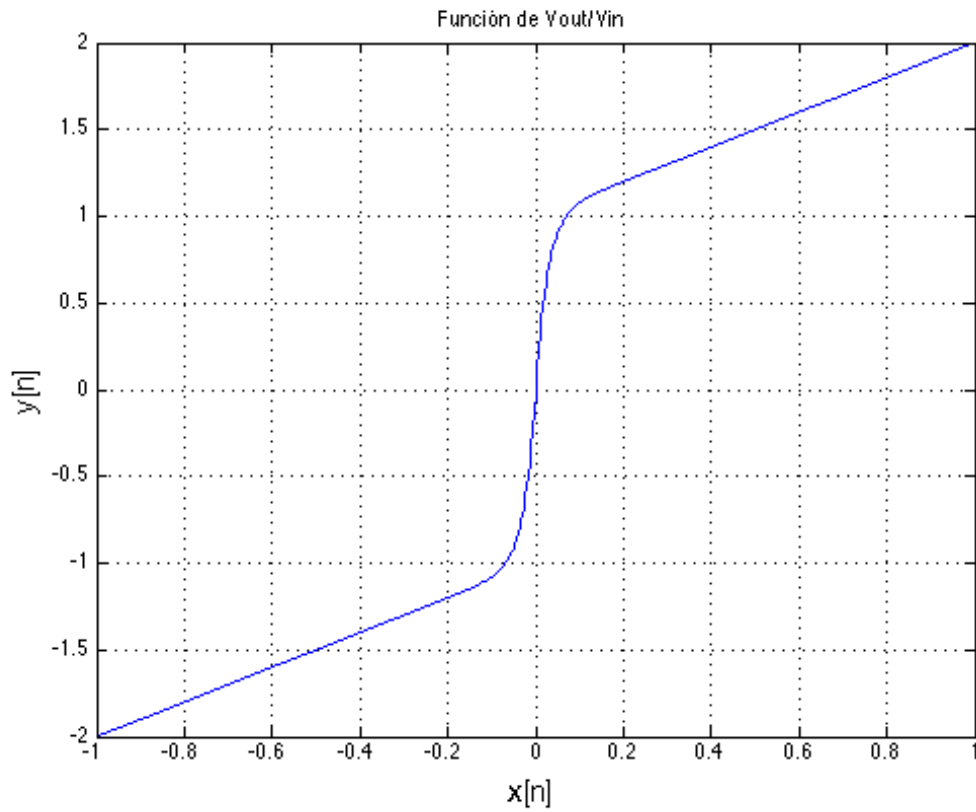


Figura 118. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $R_{in} = 100 \text{ ohm}$

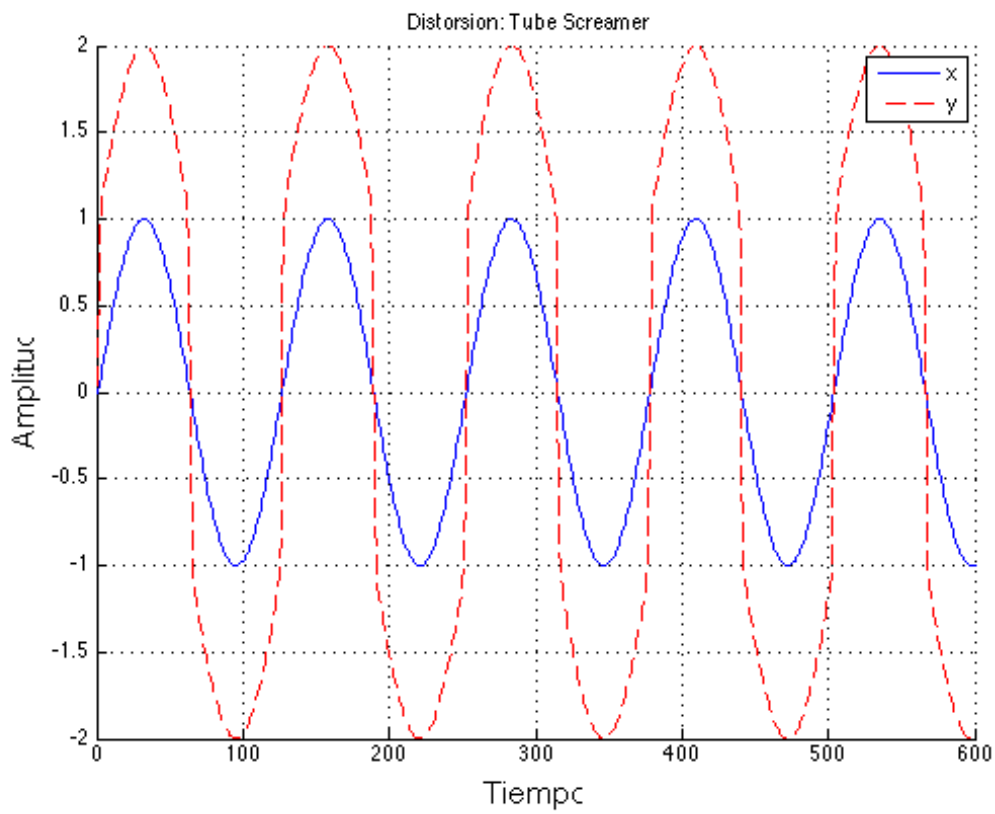


Figura 119. Señales  $y[n]$   $x[n]$  en tiempo para  $R_{in} = 100 \text{ ohm}$

Si se disminuye el valor de la resistencia de entrada  $R_{in} = 25 \Omega$  la curva característica del circuito y las señales son las siguientes:

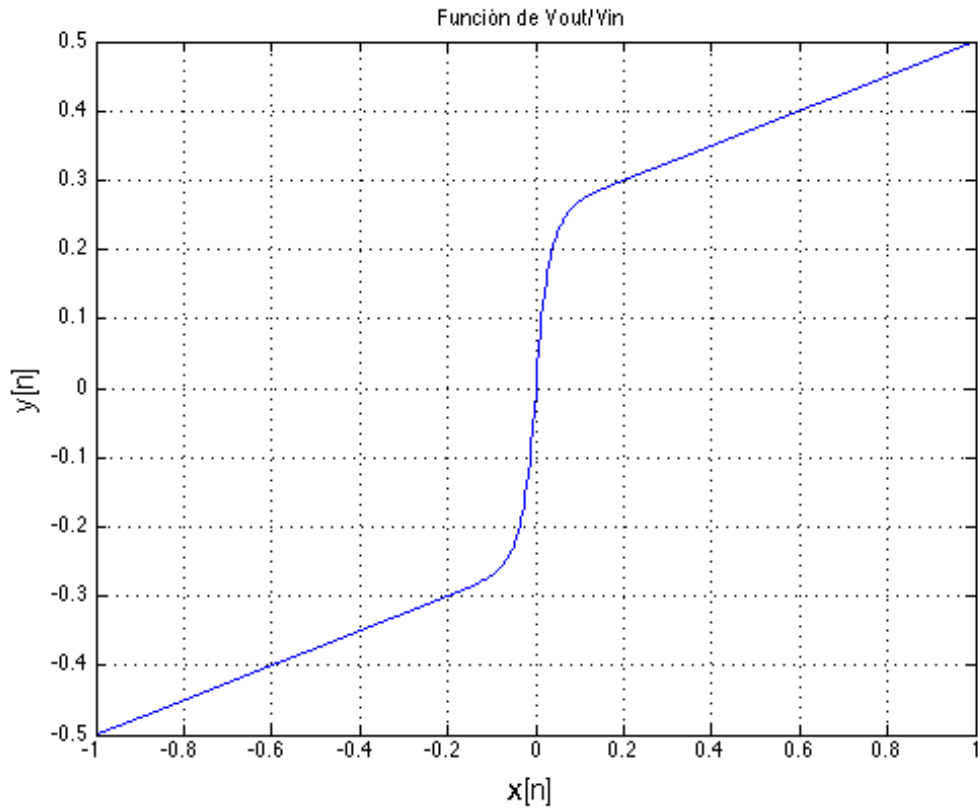


Figura 120. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $R_{in} = 25 \text{ ohm}$

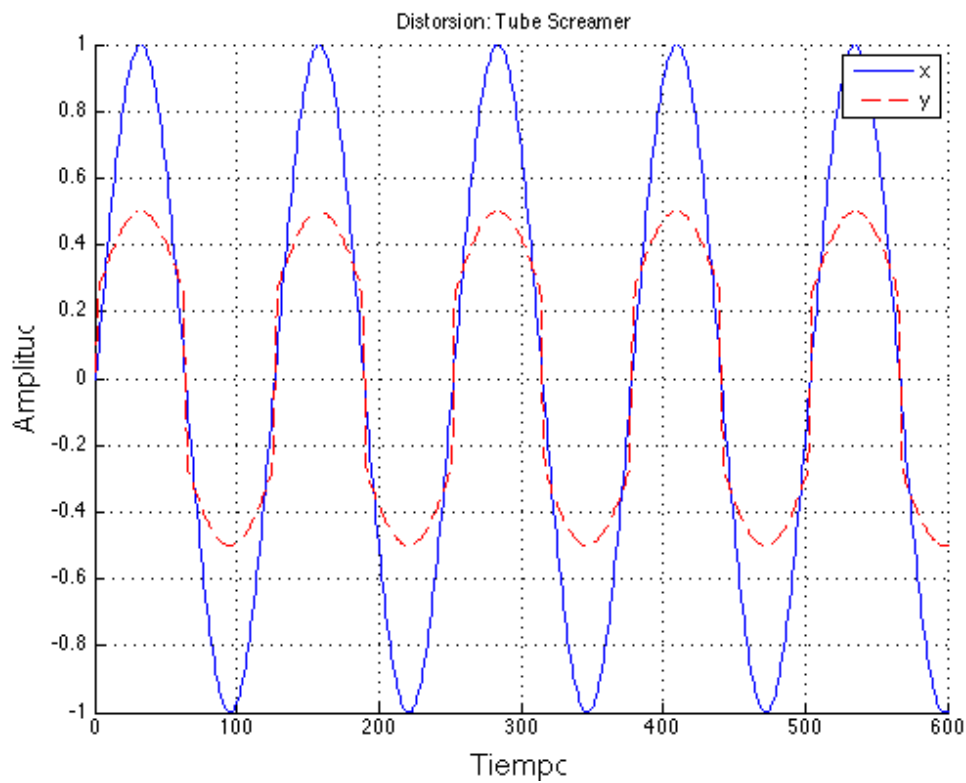
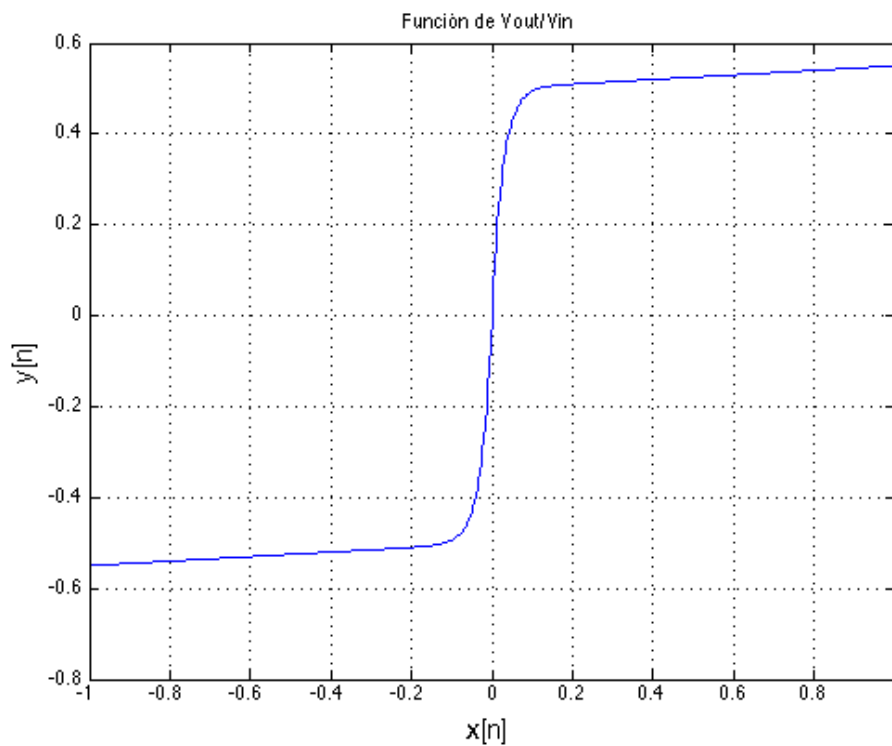


Figura 121. Señales  $y[n]$   $x[n]$  en tiempo para  $R_{in} = 25 \text{ ohm}$

Como se puede observar, la resistencia de entrada del circuito, controla la amplitud de la señal de salida, si se aumenta la resistencia de entrada, crece la señal de salida. Para la utilización de señales de audio, hay que tener en cuenta que si se sobrepasa la unidad al distorsionarla, la señal de salida se satura.

- **Variación de  $R_f$ :**

Si se aumenta el valor de la resistencia del lazo de realimentación  $R_f = 1000 \Omega$  la curva característica del circuito y las señales son las siguientes:



**Figura 122. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $R_f = 1000 \text{ ohm}$**

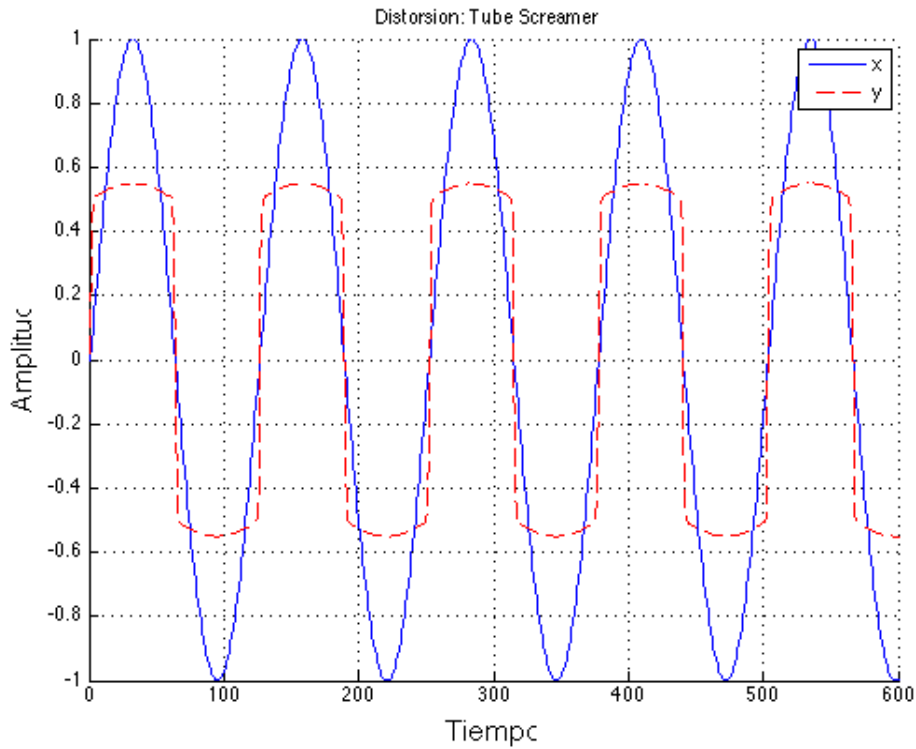


Figura 123. Señales  $y[n]$   $x[n]$  en tiempo para  $R_f = 1000 \text{ ohm}$

Si se disminuye el valor de la resistencia del lazo de realimentación  $R_f = 10 \Omega$  la curva característica del circuito y las señales son las siguientes:

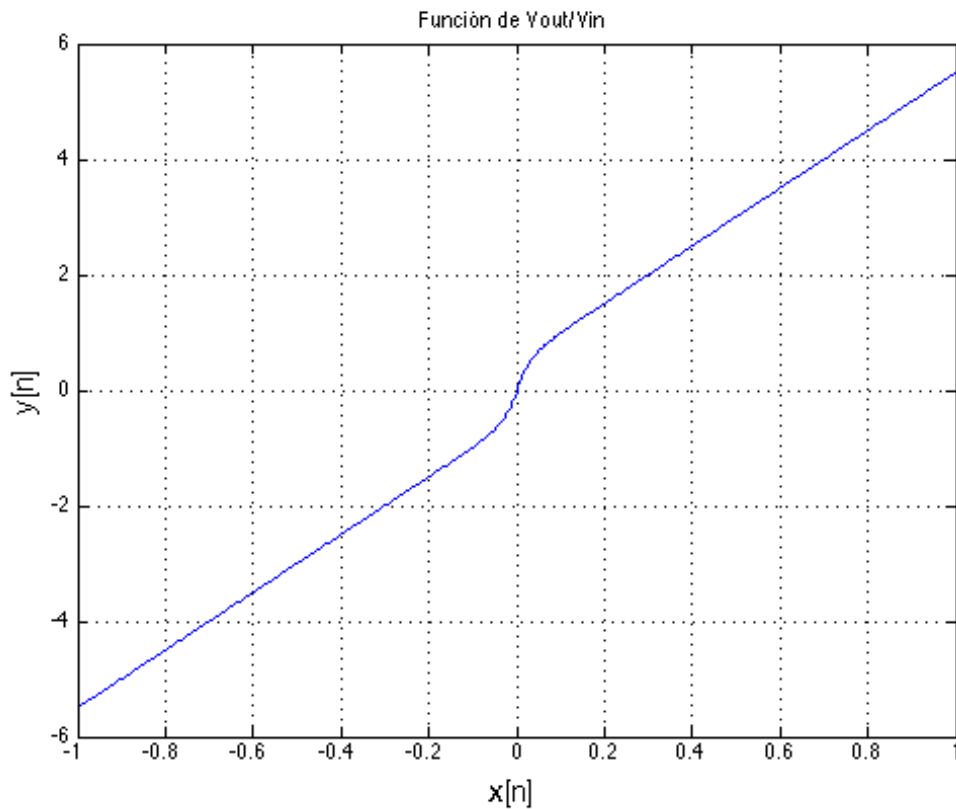
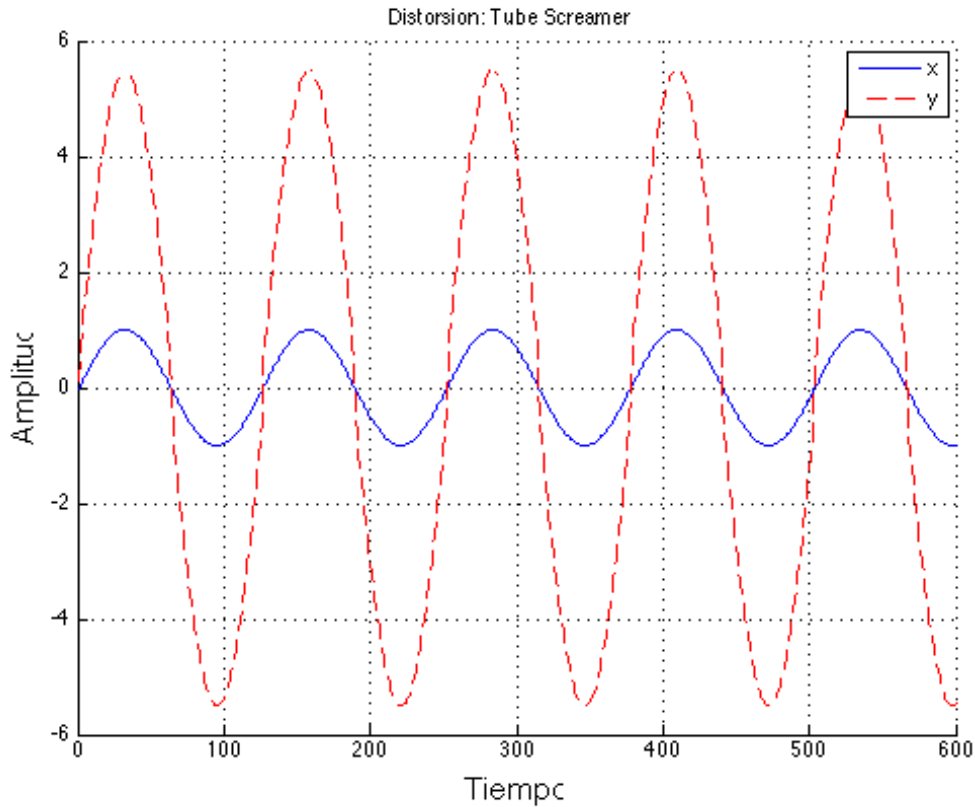


Figura 124. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $R_f = 10 \text{ ohm}$



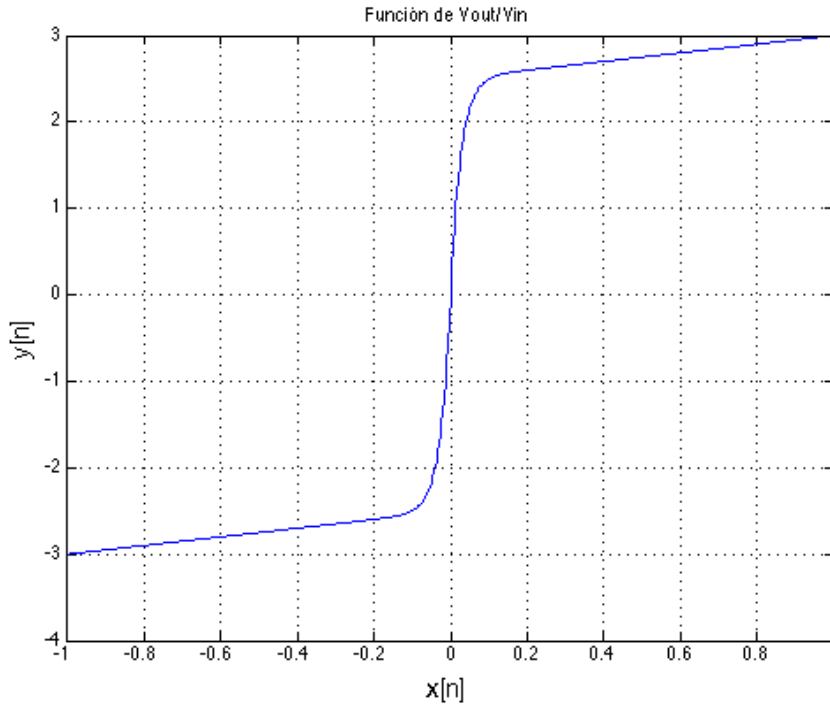
**Figura 125. Señales  $y[n]$   $x[n]$  en tiempo para  $R_f = 10 \text{ ohm}$**

Variando el valor de la resistencia  $R_f$  se puede apreciar que se modifican los máximos y mínimos de la señal. Cuando se aumenta el valor de dicha resistencia, los máximos y mínimos se recortan (clipping), como se puede apreciar en la figura 98. Sin embargo, al disminuir dicha resistencia, se realzan los máximos y mínimos. Esto es debido a que se varía la corriente que circula por el lazo de realimentación y al aumentar la resistencia  $R_f$ , circula más por los diodos, que tienden a saturar cuando la señal de entrada es superior a su valor de tensión de corte.

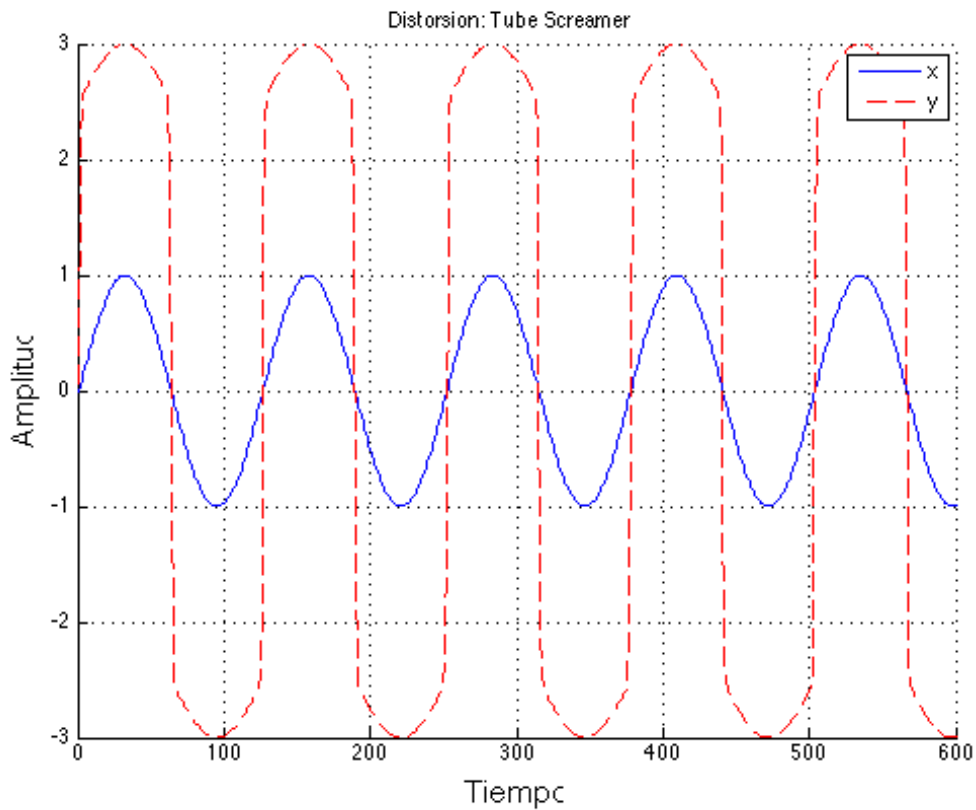


- **Variación de  $I_0$ :**

Si se aumenta el valor de la corriente  $I_0 = 0.05 \text{ A}$  la curva característica del circuito y las señales son las siguientes:

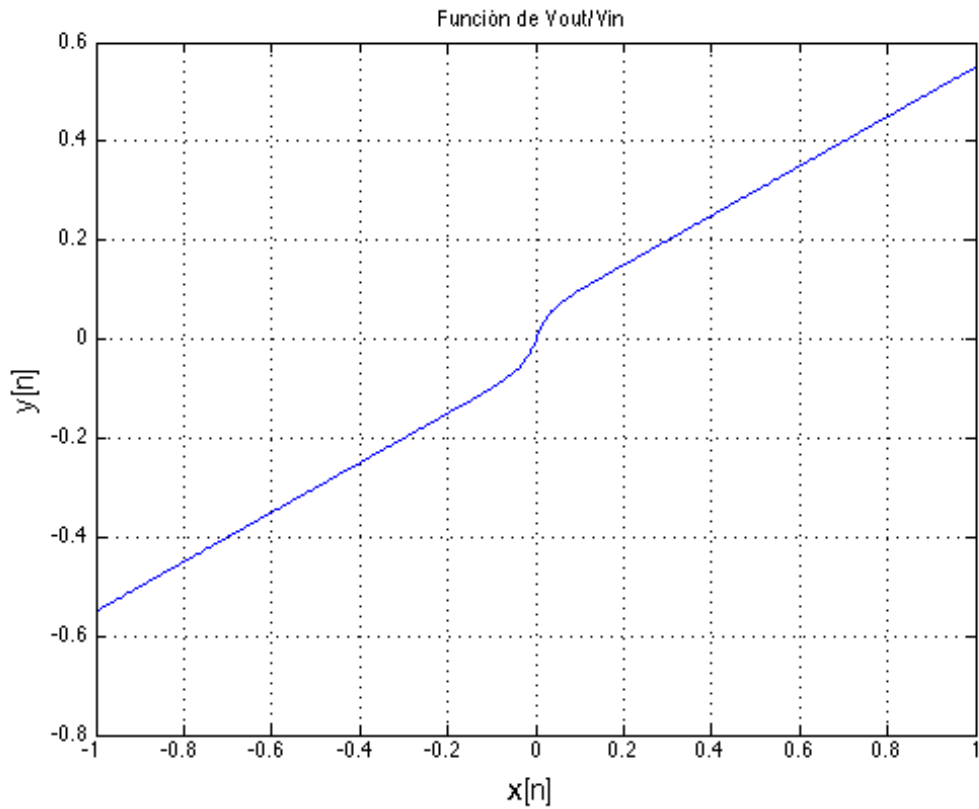


**Figura 126. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $I_0 = 0.05 \text{ A}$**

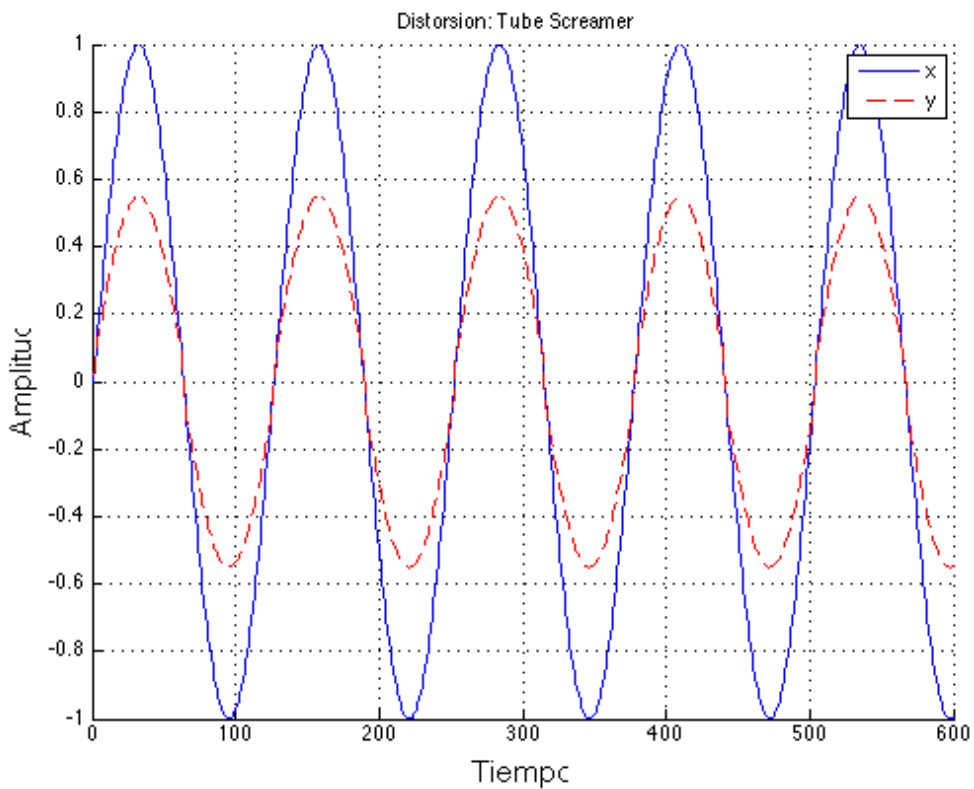


**Figura 127. Señales  $y[n]$   $x[n]$  en tiempo para  $I_0 = 0.05 \text{ A}$**

Si se disminuye el valor de la corriente  $I_0 = 0.001$  A la curva característica del circuito y las señales son las siguientes:



**Figura 128. Señal de salida  $y[n]$  respecto a  $x[n]$  para  $I_0 = 0.001$  A**



**Figura 129. Señales  $y[n]$   $x[n]$  en tiempo para  $I_0 = 0.001$  A**

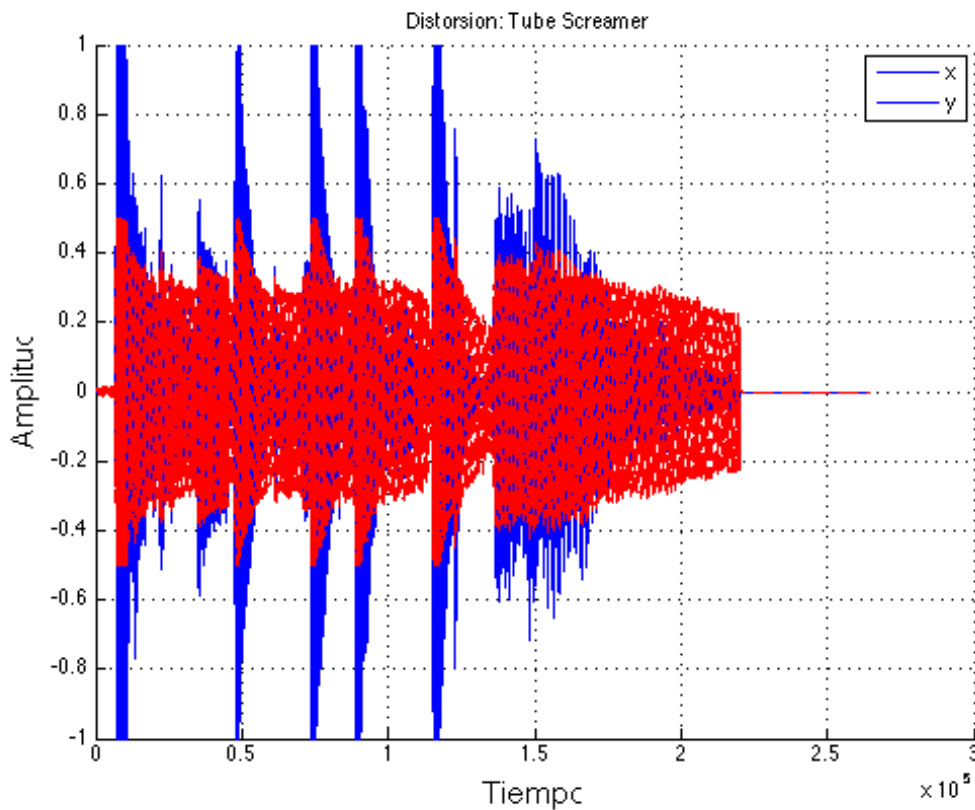
Como se puede observar, al variar el valor de la corriente de los diodos  $I_0$ , se modifica la linealidad de la señal de salida debido al comportamiento de los diodos. Si se aumenta dicha corriente, las transiciones entre los estados de los diodos serán más abruptas. Además, se aprecia en la figura 104 dichos cambios de estados.

○ **Resultados caso práctico:**

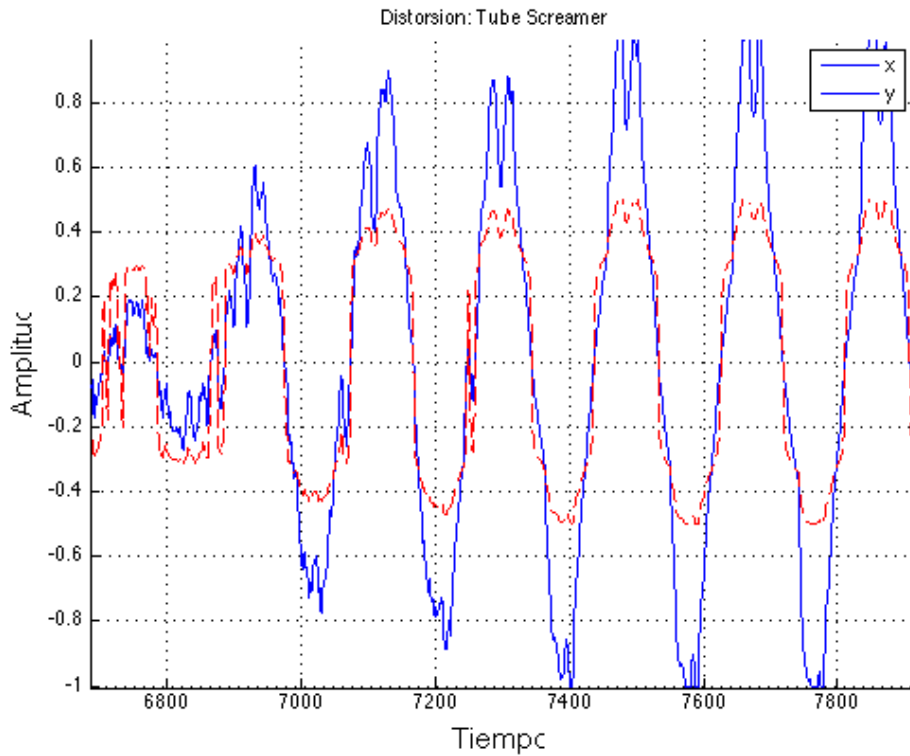
Para percibir este efecto, se dispone de un archivo de audio de muestra, el cual se introduce en el sistema y genera un vector de salida. Para realizar la ejecución se han dado los siguientes valores:

- $R_{in} = 50 \Omega$ .
- $R_f = 200 \Omega$ .
- $I_0 = 0.005 \text{ A}$ .
- $V_T = 0.026 \text{ v}$ .

A continuación se representa la señal de entrada y de salida del Tube Screamer.

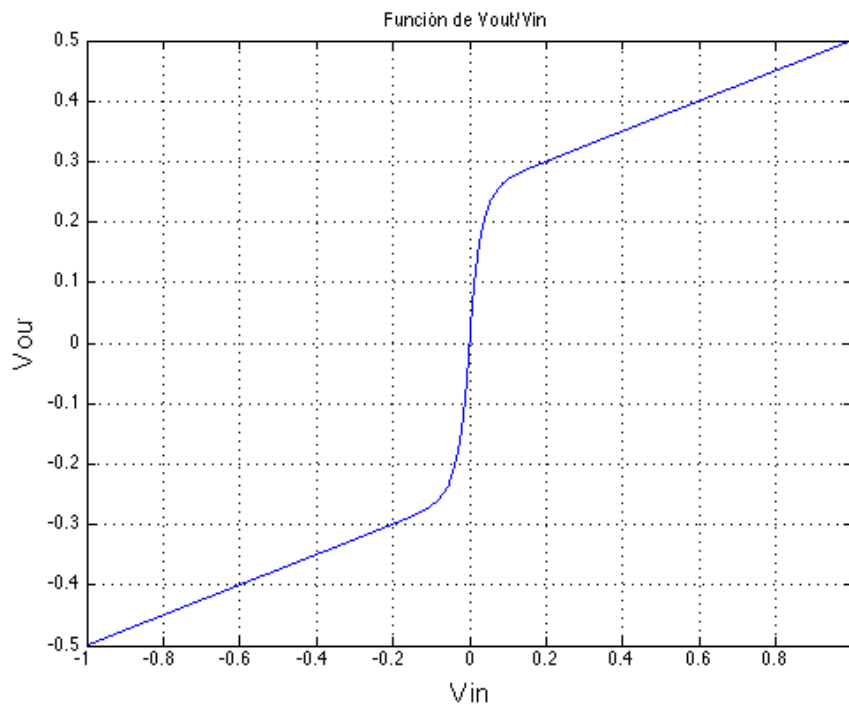


**Figura 130. Señales  $x[n]$  e  $y[n]$**



**Figura 131. Señales  $x[n]$  e  $y[n]$  ampliadas**

En las anteriores gráficas, se aprecia que la señal no sobrepasa el valor de amplitud de 0.5 y quedan recortadas sus variaciones en los máximos y mínimos locales. Todo este comportamiento de la señal de salida viene reflejado en la curva característica del circuito para este caso (Figura 107).



**Figura 132. Señal de salida  $y[n]$  respecto a  $x[n]$**

## 6. CONCLUSIONES Y TRABAJOS FUTUROS

---

### 6.1. Conclusiones

A lo largo de este proyecto de fin de carrera, se han visto en profundidad los diferentes temas propuestos para desarrollar e implementar diferentes aplicaciones y material docente para la asignatura Tecnologías de Audio.

En primer lugar, para desarrollar el sintetizador granular se obtuvieron los conocimientos necesarios sobre el tratamiento y edición de audio en las DAWs. Seguidamente se estudió la composición musical en EarSketch™ y finalmente se desarrolló el sintetizador granular, así como una serie de programas en Python para utilizar Earsketch en la DAW Cochkos Reaper™.

Para desarrollar el sintetizador sustractivo, la base fundamental ha sido el desarrollo de la aplicación en Matlab™ del sintetizador. A priori se creó la interfaz gráfica para seguir la lógica de funcionamiento y las posibles interacciones del usuario con dicha interfaz. Seguidamente se desarrolló todo el código con las posibles casuísticas de funcionamiento. Tras finalizar su desarrollo, se han podido realizar los distintos estudios de su funcionamiento y aplicaciones prácticas para el análisis y comportamiento del sintetizador.

Por último, en el apartado de efectos, se comenzó con el estudio teórico de los diferentes efectos propuestos:

- Efectos con líneas de retardo:
  - Eco FIR.
  - Eco IIR.
  - Vibrato.
  - Chorus.
- Efectos con filtros:
  - Wah-Wah.
- Efectos de procesado de dinámica:
  - Compresor.
- Distorsión.
  - TubeScreamer.

Seguidamente se realizó un set de funciones en Matlab™ de dichos efectos, con pruebas analíticas y reales, para ello se hizo uso de archivos de audio reales. Después de obtener los resultados con éxito, se generaron los archivos de audio resultantes de la aplicación de los efectos.

En resumen, este Proyecto Fin de Carrera abarca diferentes ámbitos y aspectos a tratar en la asignatura Tecnologías del Audio. Se ha desarrollado una gran cantidad de software y un desarrollo amplio de la temática, para aplicar los conocimientos y obtener una visión intuitiva de los temas a tratar en la asignatura.

### **6.3. Trabajos futuros**

A partir del material desarrollado a lo largo del proyecto, se podría continuar con lo siguiente:

- Síntesis granular de audio en DAW. Desarrollo de una interfaz gráfica que permita un manejo fluido y consistente del sintetizador. Ampliación de la base de datos y posibilidad de implementar diferentes algoritmos de composición.
- Síntesis de audio en Matlab™. Ejecución en tiempo real de la aplicación, definiendo previamente el control de un interfaz MIDI para poder seleccionar y definir la señal de entrada.
- Efectos de audio en Matlab™. Desarrollo de una interfaz gráfica para controlar cada efecto individualmente. Ampliar el set de funciones en cada apartado con nuevos efectos.

## ANEXO A: CÓDIGO

---

El siguiente link permite acceder al código implementado para el desarrollo del Proyecto de Fin de Carrera. El contenido está dividido en tres carpetas que engloban los temas tratados en los siguientes apartados:

- Síntesis granular de audio en Matlab™.
- Síntesis de audio en Matlab™.
- Efectos de audio en Matlab™.

<https://www.dropbox.com/sh/t9bsvwf28rclyf2/AAA4XPo0ONLYjRBsZneJCETBa>

Ha sido consultado por última vez el 11 de julio de 2014.

## ANEXO B: REFERENCIAS

---

- [1] Material docente de la asignatura Tecnologías de Audio. Prof. Daniel Ramos Castro. Consultado a través de <https://moodle.uam.es/>
- [2] Udo Zolder. (2011). "DAFX: Digital Audio Effects, 2nd Edition". Wiley.
- [3] Ballou. G. (2008). "Handbook for Sound Engineers", 4th ed." Focal Press, 2008.
- [4] Lerch. A. (2012). "An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics". Wiley.
- [5] Everett. S. Emory University. "Introduction to Digital Sound Design". Curso online impartido en el portal Coursera. <https://www.coursera.org> (Diciembre 2013-Enero 2014).
- [6] Stilson. T, Smith. J (2006). Music Department, Stanford University. "Analyzing the Moog VCF with Considerations for Digital Implementation".
- [7] EarSketch™. <http://earsketch.gatech.edu/category/learning>
- [8] Roads. C. (2004). "Microsound, First MIT Press paperback edition". 3. Granular Synthesis.
- [9] AudioToolkit, Matlab™. <http://www.cs.uml.edu/~stu/cs211.html>



# PRESUPUESTO

---

## 1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2.000 €
- Alquiler de impresora láser durante 6 meses ..... 50 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 2.200 €

## 2) Gastos generales

- 16 % sobre Ejecución Material ..... 352 €

## 3) Beneficio Industrial

- 6 % sobre Ejecución Material ..... 132 €

## 4) Honorarios Proyecto

- 711 horas a 15 € / hora ..... 10.665 €

## 5) Material fungible

- Gastos de impresión ..... 200 €
- Encuadernación ..... 20 €

## 6) Subtotal del presupuesto

- Subtotal Presupuesto ..... 15.769 €

## 7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 3.311,49 €

## 8) Total presupuesto

- Total Presupuesto ..... 19.080,49 €

Madrid, Julio de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Sergio Carrero Jordán

Ingeniero Superior de Telecomunicación

# PLIEGO DE CONDICIONES

---

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un Laboratorio de Tecnologías de Audio. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

## Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.