

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Síntesis de audio basada en sistemas oscilatorios y caóticos

Trabajo de graduación presentado por Guillermo Javier de León Archila
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2018

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Síntesis de audio basada en sistemas oscilatorios y caóticos

Trabajo de graduación presentado por Guillermo Javier de León Archila
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,


2018


Vo.Bo.:

(f) 
MSc. Pablo Oliva

Tribunal Examinador:

(f) 
MSc. Pablo Oliva

(f) 
MSc. Carlos Esquit

(f) 
Ing. Luis Pedro Montenegro

Fecha de aprobación: Guatemala, 5 de diciembre de 2018.

Este trabajo es la culminación de cinco años de estudio en la Universidad del Valle de Guatemala. Un camino que ha presentado retos, pero también ha resultado agradable y sin mayores contratiempos. Esta trayectoria limpia y constructiva en mi formación superior ha sido posible por el constante apoyo de familiares, amigos y catedráticos; personas a quienes dedico este trabajo. Quiero agradecer principalmente a mi familia nuclear conformada por mi madre y hermanos, quienes que han sido un soporte fundamental tanto en el ámbito personal como académico. También quiero agradecer a mi abuela Elba, quien estoy seguro espera con ansias la graduación de todos sus nietos. Agradezco a tíos y tías que han prestado consejo y apoyo a lo largo de esta trayectoria. Las amistades han resultado indispensables durante este proceso y han hecho que esta sea una experiencia que va más allá de lo académico. Agradezco a todas las personas con las que he trabajado, estudiado y compartido tanto fuera como dentro de la universidad. Finalmente quiero agradecer a Miguel Zea, quien propuso el desarrollo de este proyecto, y a mi asesor de tesis Pablo Oliva. Personas que guiaron este trabajo a la obtención de resultados significativos y prometedores.

En el año 2017 surge la propuesta de realizar un proyecto de graduación que aplique la teoría de sistemas dinámicos no lineales a sistemas de audio. La segunda mitad de ese año fue aprovechada para estudiar temas relacionados al proyecto que no son tratados dentro del pènsum de ingeniería electrónica o mecatrónica. Luego a inicios del año 2018 se plantea tres grandes temas a tratar dentro de la temática principal; la síntesis de audio, implementación de efectos digitales de audio y la emulación de efectos analógicos. En este trabajo se trata el primer tema mencionado.

El lector encontrará la teoría necesaria para comprender el desarrollo del proyecto más referencias a los textos de los cuales se extrajo la información mostrada. Se recomienda tener conocimiento previo en procesamiento digital de señales. En la metodología encontrará los criterios que se consideraron para seleccionar los distintos sistemas dinámicos implementados y los métodos considerados para realizar la síntesis de audio. Siendo esta una primera etapa del proyecto, se considera que se han obtenido resultados significativos que abren puertas a nuevas posibilidades de experimentación y generación de conocimiento. Se espera que el lector encuentre en este trabajo la posibilidad de desarrollar su creatividad en conjunto con su desempeño académico.

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiii
Resumen	xv
Abstract	xvii
1. Introducción	1
2. Antecedentes	3
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	9
6. Marco teórico	11
6.1. Síntesis de audio	11
6.1.1. Representación digital del sonido	11
6.1.2. Sintetizadores de audio	12
6.2. MIDI	16
6.2.1. Mensajes de voz del canal MIDI	16
6.2.2. Mensajes de modo del canal MIDI	17
6.3. Desarrollo de <i>plug-ins</i> en RackAFX	17
6.3.1. Características generales de un <i>plug-in</i>	17
6.3.2. Interfaz de programación de aplicación	18
6.4. Sistemas dinámicos	19
6.4.1. Estabilidad de sistemas dinámicos autónomos	19

6.4.2.	Soluciones numéricas de sistemas autónomos	21
6.4.3.	Ciclos límite	27
6.4.4.	Caos y atractores extraños	27
6.5.	Ejemplos de sistemas dinámicos no lineales	29
6.5.1.	El oscilador de Van der Pol	29
6.5.2.	El oscilador de Chua	30
6.5.3.	El atractor de Lorenz	33
7.	Metodología	35
7.1.	Selección de los sistemas dinámicos a implementar	35
7.2.	Simulaciones experimentales de los sistemas seleccionados	36
7.3.	Creación de <i>plug-in</i> VST	36
8.	Análisis de los sistemas dinámicos a implementar	37
8.1.	Oscilador de Van der Pol	37
8.2.	Oscilador de Chua	39
8.3.	Atractor de Lorenz	42
9.	Desplazamiento del contenido espectral de los osciladores	47
9.1.	Intuición a partir del oscilador de Van der Pol	47
9.2.	Desplazamiento del contenido espectral en osciladores no lineales	48
10.	Implementación de algoritmos de síntesis de audio	53
10.1.	Implementación y análisis de métodos de integración numérica	53
10.2.	Acondicionamiento de las señales	54
10.3.	Síntesis de audio en tiempo real	55
11.	Desarrollo de primer prototipo en RackAFX	57
11.1.	Resultados obtenidos de primer <i>plug-in</i>	58
12.	Desarrollo de prototipo en RackAFX con controlador MIDI	63
12.1.	Implementación de señales envolventes	63
12.2.	Implementación de instrumento polifónico	65
12.3.	Recepción e interpretación de mensajes MIDI	65
12.3.1.	Inicialización de nota	65
12.3.2.	Finalización de nota	66
12.3.3.	Modificación de la frecuencia por controlador MIDI	66
12.4.	Resultado final	66
13.	Uso de <i>plug-in</i> en Software DAW	71
14.	Conclusiones	73
15.	Recomendaciones	75
16.	Bibliografía	77
17.	Anexos	79
17.1.	Implementación de métodos numéricos de paso variable	79

Lista de figuras

1.	Conversión de datos digitales a una señal analógica de voltaje	12
2.	Topología básica de un sintetizador de audio	12
3.	Efectos de Aliasing en ondas con Discontinuidades	14
4.	Señal Envolvente en un Sintetizador Digital	15
5.	Distintos tipos de señales envolventes en sintetizadores comerciales	15
6.	Comunicación entre cliente y <i>plug-in</i>	18
7.	Ejemplos de solución, curva integral y órbita	21
8.	Representación gráfica del método de Euler	22
9.	Error en el método de Euler	23
10.	Origen del ciclo límite en el oscilador de Van der Pol	30
11.	Circuito del oscilador de Chua	31
12.	El diodo de Chua como una función por partes	31
13.	Ciclo límite en el oscilador de Van der Pol	38
14.	Ejemplos de atractores en el oscilador de Chua	38
15.	Ejemplos de atractores en el oscilador de Chua	39
16.	Respuesta en el espacio de fase del sistema 44	40
17.	Respuesta en el tiempo de las variables de estado del sistema44	40
18.	Respuesta en frecuencia de una variable de estado del sistema 44	41
19.	Divergencia de trayectorias con condiciones iniciales cercanas en el oscilador de Chua	41
20.	Espectrograma del oscilador de Chua	42
21.	Atractor de Lorenz	43
22.	Coefficientes de Lyapunov en el atractor de Lorenz	43
23.	Divergencia de trayectorias con condiciones iniciales cercanas en el atractor de Lorenz	44
24.	Atractor de Lorenz en el tiempo	44
25.	Variable de estado z de atractor de Lorenz en frecuencia	45
26.	Espectrograma para variable de estado z en atractor de Lorenz	45
27.	Desplazamiento del Contenido Espectral en el Oscilador de Chua	49
28.	Tendencia en el desplazamiento del contenido espectral en el oscilador de Van der Pol	49

29.	Tendencia en el desplazamiento del contenido espectral en el oscilador de Chua	49
30.	Tendencia en el desplazamiento del contenido espectral en el oscilador Lorenz	50
31.	Coefficientes de Lyapunov en el atractor de Lorenz tras escalar el campo vectorial	51
32.	Mapa de Poincaré del Oscilador de Chua Para Distintos Factores de Escalamiento	51
33.	Mapa de Poincaré del oscilador de Chua Para distintos factores de escalamiento	52
34.	Ciclo límite del oscilador de Van der Pol con distintos factores de escalamiento	52
35.	Magnitud de filtro pasa banda	54
36.	Variación de frecuencia en oscilador de Chua al aumentar factor de escalamiento	59
37.	Espectro del oscilador de Chua al variar el parámetro α	60
38.	Espectro del atractor de Lorenz al variar el parámetro r	60
39.	Frecuencia dominante en atractor de Lorenz contra variación de parámetro r .	60
40.	Frecuencia fundamental en oscilador de Van der Pol contra variación de parámetro ε	61
41.	Cambio de estado en una señal envolvente	64
42.	Señal envolvente aplicada al oscilador de Van der Pol	64
43.	Interfaz gráfica de usuario de <i>plug-in</i> VST implementado	67
44.	Variación de la pendiente de la recta contra parámetro ε	68
45.	Transformada rápida de Fourier para escala musical	69
46.	Comparación del contenido espectral entre solución con ODE45 y RK4	79

Lista de cuadros

1.	Parámetros modulables en primer prototipo de <i>plug-in</i>	58
2.	Rango para controles de plug-in VST	67
3.	Frecuencias de potencia máxima experimental para una escala musical	69

Este trabajo explora un método para sintetizar sonidos musicales basado en la solución numérica de sistema dinámicos no lineales y caóticos. Esto con el propósito de crear algoritmos de síntesis de sonido que puedan ser utilizados en la industria de audio. Se propuso el uso de tres sistemas dinámicos ampliamente estudiados en la literatura. Estos sistemas son el oscilador de Van der Pol, un oscilador no lineal, y dos sistemas caóticos; el atractor de Lorenz y el circuito de Chua. Se ha encontrado que el contenido espectral de los tres osciladores puede ser desplazado linealmente al escalar el campo vectorial que describe la evolución de los sistemas. Este resultado ha permitido la síntesis de sonidos con tonos específicos dentro de un margen de error reducido. Finalmente, el método propuesto fue implementado en un plug-in VST con el fin de acoplar los algoritmos de síntesis de audio con un controlador MIDI comercial.

This work explores a method to synthesize musical sounds based on the numerical solution of nonlinear and chaotic dynamical systems. This with the purpose of creating sound synthesis algorithms that can be used in the audio industry. The use of three dynamic systems widely studied in the literature was proposed. These systems are the Van der Pol oscillator and two chaotic oscillators; the Lorenz attractor and Chua's circuit. It has been found that the spectral content of these three oscillators can be shifted in a linear fashion by scaling the vector field which describes the evolution of the systems. This result has permitted the synthesis of sounds with specific pitches within a narrow margin of error. Finally, the proposed method was implemented in a VST plug-in in order couple the audio synthesis algorithms with a commercial MIDI controller.

La síntesis digital de audio desde sus inicios ha buscado generar sonidos que emulen fielmente su contraparte analógica. Este paradigma es consistente con la necesidad de sintetizar sonidos aplicables en la industria de audio. Dentro de las ventajas de la síntesis digital de audio está que un mismo equipo de hardware puede ejecutar una infinidad de algoritmos de síntesis de audio distintos. Esto siempre dentro de las capacidades computacionales de este. La implementación digital conlleva un bajo costo de producción en comparación con los sintetizadores y efectos analógicos que requieren de una pieza de hardware distinta para cada uno. Tomando en cuenta las evidentes ventajas presentes en la síntesis digital de audio, se plantea crear algoritmos novedosos basados en sistemas dinámicos no lineales y caóticos. En este caso la calidad del sonido generado no es regida por la calidad de emulación de un sistema analógico. Esto explota otra ventaja de la síntesis digital; la capacidad de implementar sistemas puramente teóricos y modular parámetros más allá de sus limitantes físicas.

En este proyecto se implementan tres sistemas dinámicos distintos para la generación de sonido: el oscilador de Van der Pol, el oscilador de Chua y el atractor de Lorenz. A partir de diversas condiciones iniciales para cada sistema, se encuentran las trayectorias correspondientes que describen la evolución de los sistemas. Las trayectorias son encontradas de forma dinámica aplicando el método de integración numérica Runge-Kutta 4. Para la generación de tonos específicos se desarrolló un método que permite desplazar la frecuencia dominante de un sistema dinámico por medio del escalamiento del campo vectorial en el espacio de estados. Habiendo obtenido la señal requerida, esta es acondicionada por medio de filtros digitales y normalizada a un rango de oscilación acotado entre ± 1 . La funcionalidad de los algoritmos de síntesis de audio se maximiza al acoplarlos a un complemento de software utilizando la interfaz de aplicación VST. Esto permite generar notas musicales utilizando un controlador MIDI y compatibilidad del complemento de software con diversas estaciones de trabajo virtual de audio, conocidas como softwares DAW.

El uso de sistemas dinámicos en el área de síntesis de audio no es nuevo. En la literatura se pueden encontrar diversas aplicaciones. En el artículo por D. Medine [8] se describen diferentes ejemplos del uso de sistemas dinámicos para realizar síntesis de audio. El primer uso que se plantea es utilizar la solución numérica de un oscilador armónico simple para la generación dinámica de un senoide. Posteriormente se plantea un sistema que introduce no linealidades al oscilador armónico simple. Este sistema involucra acoplamiento de dos osciladores armónicos simples, donde ambos osciladores funcionan como esclavo y maestro, forzando el reinicio a cero de la fase del oscilador esclavo. Se plantea que el uso de sistemas dinámicos tiene la ventaja de evitar el uso de tablas de datos extensas, las cuales pueden llegar a consumir demasiada memoria y se evita la introducción de retrasos en la síntesis. Si se plantean los dos osciladores acoplados como un único sistema dinámico es posible encontrar el comportamiento de ambos con el mismo computo. En el mismo documento se muestran dos aplicaciones adicionales del uso de sistemas dinámicos en la síntesis de audio. La primera es la modulación recíproca de frecuencia en aplicaciones de síntesis de voz. Nuevamente se utiliza el modelo del oscilador armónico simple, sin embargo este es modulado en fase por otro o varios osciladores, introduciendo nuevamente no linealidades. Finalmente se plantea la emulación de instrumentos analógicos o acústicos por medio del modelado de sistemas dinámicos. Se expone el caso del oscilador forzado por fricción, el cual describe la dinámica de un instrumento de cuerda tocado con arco. En este caso la no linealidad se presenta en el modelo de la fricción entre el arco y la cuerda. A partir del modelo físico del fenómeno es posible emular el sonido resolviendo el sistema dinámico de forma numérica. Además, se presenta la ventaja de poder experimentar con modelos irreales de la fricción, lo cual brinda una mayor diversidad de opciones por explorar.

La síntesis de audio con sistemas caóticos ha sido explorada e implementada previamente, aunque se ha hecho de forma limitada. Existe el software de síntesis de audio y composición algorítmica *SuperCollider*, desarrollado por James McCartney en 1996. Este actualmente es un software gratuito bajo licencia de dominio público. El ambiente de desarrollo de *SuperCollider* utiliza unidades denominadas *Ugens* las cuales son bloques para desarrollo modular de síntesis y procesamiento de audio. Dentro del repositorio de *plug-ins* para *SuperCollider*

se encuentra la clase abstracta denominada *ChaosGen* de la cual se derivan *Ugens* de síntesis de audio que emplean sistemas caóticos. Algunos de estos sintetizan el audio a partir de soluciones numéricas a sistemas dinámicos. Dentro de los ejemplos se puede encontrar el atractor de Lorenz, el oscilador de dos pozos (encontrado en la literatura como *double well oscillator*) y las ecuaciones de Lotka-Volterra. La manipulación de los *Ugens* por el usuario consiste en variar los parámetros que rigen la dinámica del sistema y así obtener diversos sonidos. Estos ejemplos de síntesis de audio con sistemas dinámicos no están orientados al uso con controladores MIDI, posiblemente porque el rango de frecuencias que pueden generar es limitado o el sistema no mantiene una frecuencia fundamental a lo largo del tiempo.

El comportamiento de sistemas caóticos se ha empleado más ampliamente en el campo de composición musical algorítmica. En el artículo por R. Bidlack [1] se explica que la dinámica del sistema puede ser impresa a diferentes aspectos de una composición musical. El comportamiento caótico se puede plasmar en volumen, tono, duración de la nota, etc. Para ver un ejemplo de como se puede realizar composición algorítmica con un sistema caótico puede referirse al artículo *Reading complexity in chua's oscillator through music* [2].

Como se menciona en el texto *Designing Software Synthesizer Plug-Ins in C++* [15], la síntesis de audio cuenta con elementos subjetivos respecto a la calidad del sonido generado. Dicho aspecto ha sido un limitante en la investigación académica del ámbito. Eduardo Miranda [10] expone que la mayor parte de métodos de síntesis utilizan osciladores lineales sometidos a métodos substractivos y aditivos debido a que estos cumplen con la generación de sonidos con timbres predecibles y fácilmente manipulables. Se plantea el siguiente trabajo como respuesta a la carencia de investigación en el ámbito, buscando expandir las aplicaciones de la ingeniería electrónica y así explotar el carácter interdisciplinario del proyecto. Este combina áreas como el procesamiento digital de señales, teoría de sistemas dinámicos y teoría musical. Se plantea explorar un nuevo paradigma en la síntesis de audio, empleando osciladores no lineales y caóticos para la implementación de sintetizadores de sonido. Este nuevo paradigma aprovecharía al máximo la versatilidad propia de la síntesis digital de audio, que en comparación a sistemas analógicos no esta sujeta a restricciones físicas de los osciladores implementados.

4.1. Objetivo general

Crear e implementar algoritmos de síntesis de audio basados en la dinámica de osciladores no lineales y caóticos que sean útiles para aplicaciones prácticas en la industria de audio.

4.2. Objetivos específicos

- Definir según literatura existente las condiciones necesarias para que un sistema no lineal oscile y sea caótico.
- Experimentar con la simulación de modelos de osciladores no lineales, buscando la generación de sonido aplicable a la industria de audio.
- Implementar en tecnología de estudio virtual (VST) un instrumento digital basado en un oscilador no lineal.

Dentro de este trabajo se busca demostrar la viabilidad del uso de sistemas dinámicos para la implementación de instrumentos virtuales. Para ello, se necesita que los algoritmos generados sean aplicables a la industria de audio. El uso de sistemas caóticos se plantea con el fin de generar ondas de sonido con contenido espectral rico y variable en el tiempo. Comportamiento que no se puede obtener con métodos tradicionales de síntesis de audio. No se busca emular instrumentos analógicos sino explotar las ventajas de la implementación digital, con la capacidad de simular modelos abstractos o puramente teóricos. Para cumplir con lo planteado previamente se implementará un instrumento virtual, en forma de un *plug-in* VST. Las características mínimas que se busca tener en el instrumento virtual son las siguientes:

- El instrumento virtual contará con tres voces distintas, cada una basada en un sistema dinámico específico. Se plantea utilizar un sistema dinámico no lineal con respuesta periódica y dos sistemas dinámicos caóticos.
- La generación de sonido será controlada por un controlador MIDI. Los mensajes MIDI que podrán ser recibidos e interpretados por el *plug-in* serán *note on*, *note off* y *pitch bend*.
- El *plug-in* contará con una interfaz gráfica de usuario amigable, por medio de la cual se podrá modular la funcionalidad de este.
- Las voces del instrumento contarán con un generador de señales envolventes de cuatro segmentos (*attack*, *decay*, *sustain* y *release*). Las características de cada segmento podrán ser moduladas desde la Interfaz gráfica de Usuario.
- El instrumento implementado será de carácter polifónico, esto implica que puede reproducir más de una nota a la vez. Se busca la capacidad de reproducir un mínimo de tres notas simultáneas y agregar más según la cantidad de recursos computacionales que requieran los algoritmos implementados.

- El *plug-in* será compatible con cualquier estación de audio virtual que soporte el formato de VST en el sistema operativo Windows.

Una vez implementada la funcionalidad listada, se habrá demostrado que los modelos de sistemas dinámicos no lineales son aplicables en la implementación de instrumentos virtuales.

6.1. Síntesis de audio

6.1.1. Representación digital del sonido

En esta sección se describe brevemente cómo se representa el sonido dentro de un ordenador y como este puede ser generado a partir del flujo de datos digitales. La información que se muestra a continuación fue recuperada de *Computer sound design: synthesis techniques and programming* [10].

El sonido es producto de vibraciones mecánicas que se propagan en un medio. En el aire este se propaga en forma de ondas esféricas que se manifiestan como variaciones de presión. Estas variaciones pueden ser impresas en una señal de voltaje por medio de un micrófono, produciendo una señal analógica. Esta señal, al ser muestreada, puede ser almacenada y procesada en un ordenador. Las muestras se almacenan en forma de datos digitales.

El teorema de Nyquist-Shannon propuesto por Harry Nyquist [11] y Demostrado formalmente por Claude Shannon [18], establece que una señal analógica banda-limitada puede ser completamente recuperada si es muestreada a un frecuencia dos veces mayor que la frecuencia límite de la banda. La frecuencia máxima que presente en una señal muestreada se denomina frecuencia de Nyquist y es igual a la mitad de la frecuencia de muestreo. Si se muestrea una señal que no está banda-limitada, las frecuencias superiores a la frecuencia de Nyquist introducen contenido espectral adicional que distorsiona la señal original y no permite que esta pueda ser recuperada completamente. A este fenómeno se le conoce como aliasing.

El sonido puede ser sintetizado de forma analógica al implementar circuitos que generan señales oscilatorias de voltaje. Estas señales se utilizan para hacer vibrar una membrana, transformando la onda de voltaje en una onda acústica. La representación del sonido en forma de una señal digital brinda una ventaja evidente al momento de hacer síntesis de audio. Las señales digitales pueden ser generadas por software en una gran diversidad de

formas, mientras que la forma de las señales analógicas son regidas de forma estricta por los componentes del circuito que las genera. Para transformar una señal digital a una onda acústica se requiere de un convertidor de digital a analógico (DAC por sus siglas en inglés). Este es un dispositivo que recibe como señal de entrada muestras digitales que se transforman a un voltaje analógico. El voltaje correspondiente al dato se mantiene estable durante el periodo de muestreo y cambia hasta que llega la siguiente muestra. Posteriormente la señal es transferida a un filtro reconstructor cuya dinámica suaviza la onda para finalmente poder ser transferida a una bocina y generar la onda acústica. El filtro reconstructor es un filtro pasa bajas con frecuencia de corte igual a la frecuencia de Nyquist.

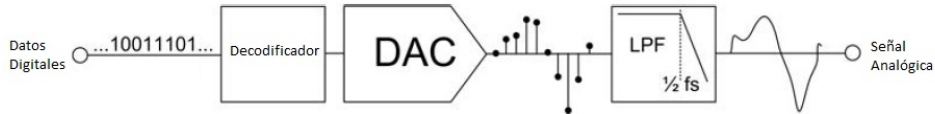


Figura 1: conversión de datos digitales a una señal analógica de voltaje [14]

6.1.2. Sintetizadores de audio

La síntesis de audio consiste en la generación de sonido por medios no acústicos. Un sintetizador de audio es un dispositivo que cumple con la función descrita anteriormente. Will Pirkle [15] describe tres componentes básicos que conforman un sintetizador de audio.

- Fuentes: Objetos que generan señales de audio, como osciladores y generadores de ruido.
- Modificadores: Objetos que alteran la señal de audio generada por las fuentes, tales como filtros y ondas envolventes.
- Controladores: Objetos que controlan parámetros tanto de las fuentes como de los modificadores. Un ejemplo de controlador utilizado en la industria de audio es el controlador MIDI.

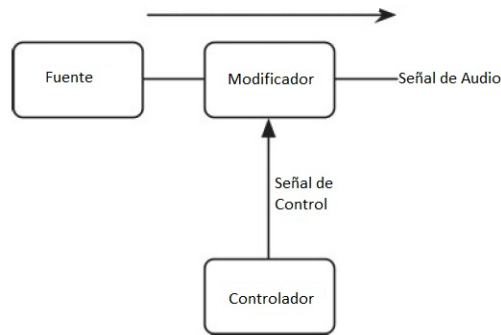


Figura 2: Topología básica de un sintetizador de audio [15]

Osciladores

Un oscilador en la industria de audio es un objeto que genera una señal sinusoidal o periódica sin necesidad de tener una señal de entrada. Originalmente se utilizaron osciladores analógicos implementados como circuitos para generar sonido. Luego, con la introducción de los sistemas de computo programables, los osciladores también fueron implementados en software. Will Pirkle [15] describe algunas de las formas más comunes para implementar osciladores digitales, las cuales se enlistan a continuación:

- Representación matemática de la señal.
- Tablas de datos.
- Extracción de muestras de audio contenidas den archivos WAV.
- Modulación en frecuencia y fase.

En este trabajo se utilizará exclusivamente con el primer métodos enlistado.

Síntesis aditiva

Un paradigma sumamente popular dentro de la síntesis de sonido es la síntesis aditiva. Esta consiste en la generación de una onda periódica por medio de su análisis espectral. Conociendo la amplitud que tiene la frecuencia fundamental y sus respectivos armónicos en el espectro de una onda, es posible sintetizarla por medio de la adición de sinusoides con amplitudes y frecuencias específicas. Este método tiene la ventaja de eliminar por completo, o casi en su totalidad, los efectos de aliasing, ya que la señal está banda-limitada a la frecuencia del armónico más alto que se decida agregar a la adición. Este método puede presentar un alto costo computacional para señales con frecuencia fundamental en la parte baja del espectro audible. Esto se debe a la cantidad de armónicos significativos necesarios para representar la señal. Este fenómeno es ejemplificado en *Designing software synthesizer plug-ins for C++* [15] de la siguiente manera: Considérese la nota de frecuencia más baja en un piano. Esta nota tiene su frecuencia fundamental a $27.5Hz$, lo cual incluye 800 armónicos antes de alcanzar la frecuencia de Nyquist ($22,050Hz$ con una frecuencia de muestreo estándar de $44,100Hz$).

Síntesis sustractiva

La síntesis sustractiva consiste en el uso de señales básicas como una onda triangular, cuadrada o de diente de sierra que posteriormente es procesada por filtros y efectos digitales. El efecto del aliasing, se vuelve evidente al momento de implementar osciladores con su representación matemática en el dominio del tiempo. Esto se debe a que la suma de sinusoides en una señal periódica no está banda-limitada. La cantidad de armónicos significativos aumenta con la ausencia de suavidad en las curvas. En la Figura 3 se evidencia este fenómeno. Al efectuar una transformada rápida de Fourier a una señal periódica se espera encontrar contenido espectral únicamente en la frecuencia fundamental y armónicos, sin embargo se

puede observar que frecuencias superiores a Nyquist, se han introducido en partes más bajas del espectro.

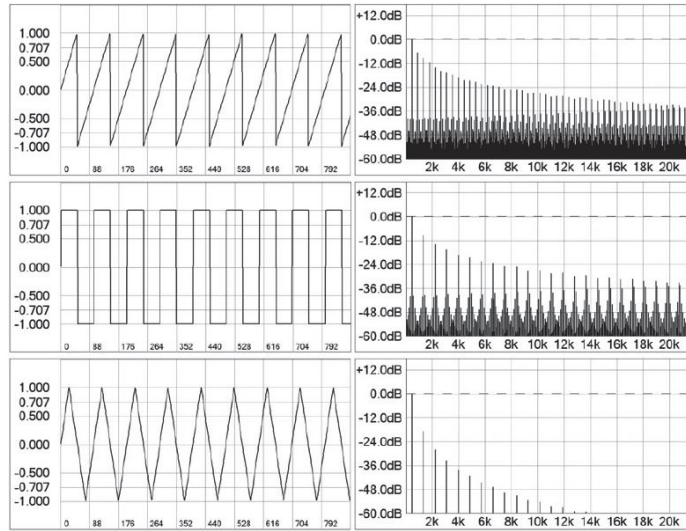


Figura 3: Efectos de aliasing en ondas con discontinuidades [15]

Señales moduladoras

La señal producida por un oscilador es modulada de diversas formas en un sintetizador de audio. Esta modulación se puede dar tanto en frecuencia como en amplitud. Los osciladores también pueden ser utilizados para generar señales de modulación, a este tipo de osciladores se les conoce como osciladores de baja frecuencia, ya que suelen oscilar fuera del espectro audible. Las señales de modulación son utilizadas para variar parámetros de bloques modificadores como lo sería la frecuencia de corte de un filtro digital.

En este trabajo se implementa únicamente señales moduladoras para la amplitud de las ondas generadas. Estas señales se conocen como envolventes (encontradas en la literatura como *Envelope signals*) ya que únicamente modifican la amplitud del sonido sin afectar significativamente el contenido espectral. El objetivo de las señales envolventes es emular la dinámica presente en la intensidad del sonido en instrumentos acústicos. Esto con el fin de lograr un sonido más natural al oído humano. En los primeros sintetizadores analógicos se tenía amplificadores controlados por voltaje para modular la amplitud de las señales provenientes de los osciladores. El voltaje modulador es producido por la carga y descarga de circuitos RC, los cuales se comportan de forma exponencial. Esta forma resulta natural para el oído humano ya que este percibe el sonido de forma logarítmica [14]. Este mismo tipo de modulación ha sido adoptado en los instrumentos digitales.

Una señal envolvente se plantea como una función por partes. Esta señal tiene una amplitud menor o igual a la unidad y es multiplicada punto a punto con la señal del oscilador, la cual, también está acotada en valores entre ± 1 . De esta forma se asegura que los datos modulados se mantendrán dentro del rango aceptado para datos de audio digital. Una señal envolvente tradicional cuenta con cuatro partes: ataque, decaimiento inicial, nivel sostenido y decaimiento final [15] (Controles usualmente encontrados en la industria como *attack*, *decay*,

sustain y *release*). Este es el modelo que se implementa en este trabajo. El ataque se modela como la carga del capacitor en un circuito RC con tiempo de carga modulable. Este segmento se inicia al momento de presionar una tecla en el controlador. Tras haber transcurrido el tiempo de carga establecido se procede al decaimiento inicial. Este es un decaimiento parcial que es modelado como la descarga de un circuito RC. El tiempo de decaimiento y el nivel hasta el cual decae la señal es modulado por el controlador. La siguiente etapa consiste en sostener el nivel previo siempre que la tecla se mantenga presionada. Se procede al decaimiento total de la señal al liberar la tecla. Este segmento también se modela como la descarga de un circuito RC. En la Figura 4 se puede observar como se comporta la señal envolvente a lo largo del tiempo.

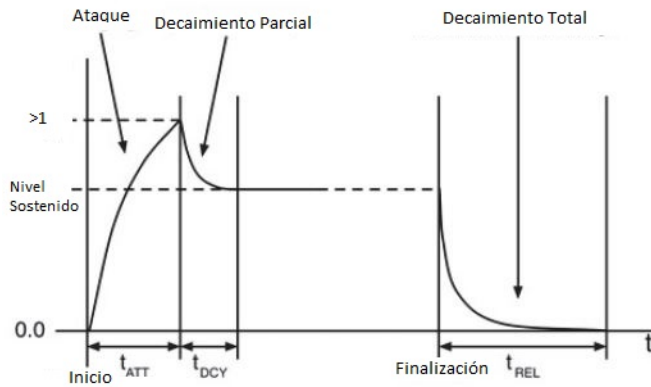


Figura 4: Señal Envolvente en un Sintetizador Digital [15]

En la implementación digital de sintetizadores de audio, la generación de señales envolventes no está restringida por los modelos de circuitos utilizados tradicionalmente e la industria de audio. Varios sintetizadores comerciales utilizan señales envolventes conformadas por más de cuatro segmentos y con formas distintas a las curvas exponenciales.

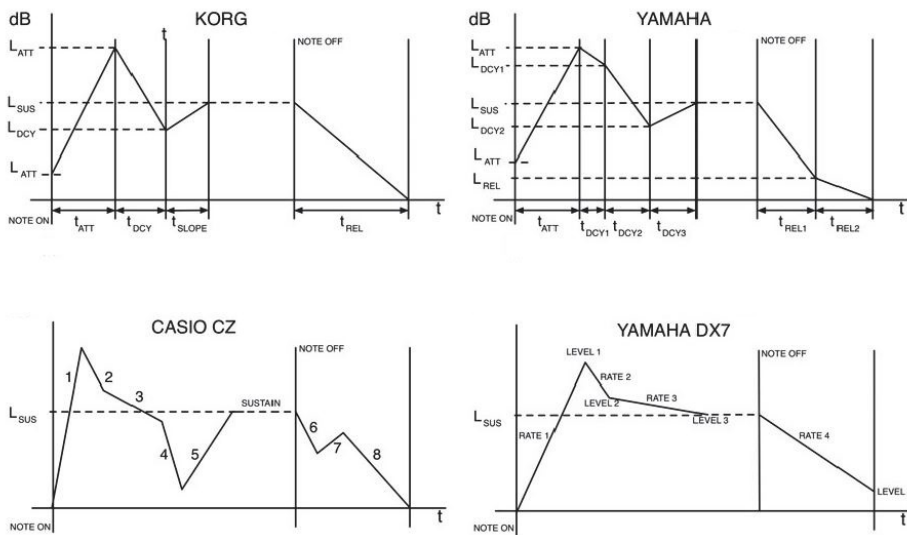


Figura 5: distintos tipos de señales envolventes en sintetizadores comerciales[15]

6.2. MIDI

En *Maximum MIDI: Music Applications in C++* [9] se detallan los conceptos básicos del estándar MIDI así como los orígenes y aplicaciones del mismo. En 1983 se propuso la interfaz digital de instrumentos musicales (MIDI por sus siglas en inglés) como una forma de interconectar y estandarizar el manejo de instrumentos digitales. Esto implica el intercambio de datos entre diversos dispositivos conectados en una red regida por el estándar MIDI. Es importante hacer aclaraciones respecto al estándar, ya que usualmente este es asociado con la síntesis de audio como tal, pero, este no es el caso. MIDI cumple los siguientes propósitos:

- Es un estándar de interconexión para instrumentos musicales, computadoras y otros dispositivos.
- Es un estándar para reproducir sonidos sintetizados en una tarjeta de audio.
- Es un protocolo de comunicación serial estandarizado.
- Brinda la capacidad de controlar dispositivos no musicales, como luces de teatro, grabadoras de sonido, etc.

El estándar MIDI no incluye métodos para sintetizar sonido.

Un instrumento digital se controla por el usuario utilizando un controlador MIDI. Este es el dispositivo encargado de enviar los mensajes que inicializan y modulan la síntesis de sonido. A. Pejrolo y R. DeRosa [12] se describen los tipos de mensajes que se manejan entre un controlador MIDI y el dispositivo encargado de la síntesis de sonido. El controlador MIDI puede tener incrustada una tarjeta de audio la cual le permitirá producir el sonido de forma directa. Los mensajes manejados por el estándar MIDI se dividen en dos categorías principales: mensajes de canal y mensajes de sistema. Los mensajes de canal por su parte se subdividen en mensajes de voz del canal y mensajes de modo del canal.

6.2.1. Mensajes de voz del canal MIDI

Los mensajes de voz portan información referente a la interpretación musical.

Note On: Este mensaje se envía cuando una tecla es presionada en el controlador MIDI. El mensaje incluye cual nota fue presionada, información codificada en un rango de enteros entre 0 y 127. Se incluye el canal en el cual se envió la nota (1-16) y la velocidad con la que se presionó la tecla, valor que varía de 0 a 127. Este último dato provee información de la intensidad con la que se debe reproducir la nota correspondiente a la tecla presionada.

Note Off: Este mensaje se envía en el momento en el que se libera una tecla previamente presionada. Este se envía con el propósito de finalizar la reproducción de la nota activada con el mensaje *Note On*. El mensaje también puede incluir información de la velocidad con la que se liberó la tecla, aunque la mayoría de instrumentos MIDI no utilizan este dato.

Aftertouch Este es un mensaje que se envía cuando el usuario ejerce mayor presión en una tecla que mantiene presionada. Usualmente se asigna al control de efecto de vibrato del

sonido, sin embargo, también se utiliza en otros parámetros como el volumen. Este mensaje puede ser monofónico o polifónico. En el primer caso basta con que una tecla envíe el mensaje para afectar todas las teclas presionadas en el momento. En el caso polifónico cada tecla es afectada individualmente por el mensaje.

Pitch Bend: Este mensaje es controlado por la rueda o palanca de portamento en el controlador MIDI. Este se utiliza para subir o bajar el tono generado por una nota mientras esta se reproduce. Con la finalidad de brindar un control más exacto, puede adquirir un valor de 0 a 16,383, en comparación con el rango usual de 8 bits.

6.2.2. Mensajes de modo del canal MIDI

Mensaje de cambio del programa: Este mensaje se utiliza para cambiar el programa del canal. El programa está asociado al instrumento digital que cuenta con un banco de voces. Este mensaje tiene un rango de 0 a 127. En la actualidad un controlador MIDI puede tener mucho más que 128 opciones de instrumentos y sonidos a sintetizar, por lo que los programas se organizan en bancos. Debido a ello es necesario incluir un mensaje de cambio de banco junto con el cambio de programa.

Mensaje de cambio de control: Este mensaje permite controlar varios parámetros del canal MIDI. existen 128 distintos controles a modular. Algunos controles están estandarizados para todos los controladores MIDI. Para ver la función asociada a cada control refiérase a la Cuadro 1.2 en *Acoustic and MIDI Orchestration for the Contemporary Composer* [12].

6.3. Desarrollo de *plug-ins* en RackAFX

RackAFX es un software desarrollado por Will Pirkle, autor de *Designing Software Synthesizer Plug-Ins in C++* [15] y *Designing Audio Effect Plug-Ins in C++* [14], que permite el diseño de *plug-ins* en C++ trabajando en conjunto con *Visual Studio*. Dentro de las capacidades que ofrece el software está el prototipado rápido de algoritmos de efectos de audio y síntesis de sonido, diseño de *plug-ins* compatibles con cualquier cliente de Windows que utilice el estándar VST2 o VST3, soporte completo del protocolo MIDI y un espacio de diseño de interfaces gráficas. Más información sobre las capacidades e historia del software *RackAFX* se puede encontrar en el sitio web oficial de RackAfx [17].

6.3.1. Características generales de un *plug-in*

Un *plug-in* es un componente de software que se comunica con un ejecutable denominado el cliente. El objetivo del *plug-in* es agregar funcionalidad al cliente. En la industria de audio un *plug-in* puede tener una gran variedad de funciones. Dentro de las más comunes están efectos de sonido, analizadores de frecuencia y sintetizadores de audio [14].

En el sistema operativo Windows los *plug-ins* son empaquetados en forma de una biblioteca de enlace dinámico (Dynamic link library, archivos con extensión .dll). El archivo DLL contiene funciones pre-compiladas, las cuales el cliente puede llamar para que el *plug-in* las

ejecute. La ventaja de manejar funciones en forma de bibliotecas de enlace dinámico es que al ser independientes del cliente pueden ser compatibles con software de distintos fabricantes. Además, no se requiere de recompilar el cliente para la corrección de errores, sino solo se requiere recompilar el *plug-in*.

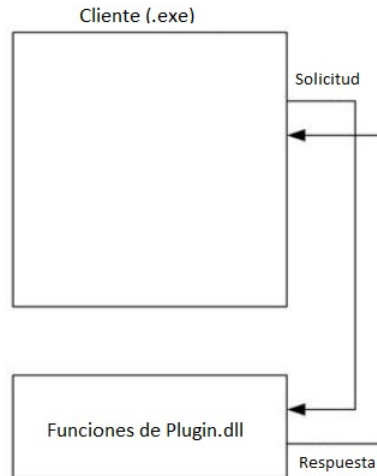


Figura 6: Comunicación entre cliente y *plug-in*[14]

Un *plug-in* usualmente está asociado a una interfaz gráfica de usuario que hace accesible la modulación de parámetros por medio de controles gráficos. Cuando se crea una nueva instancia del *plug-in*, se crea una nueva ventana con la interfaz gráfica incrustada. Cada vez que un control de la interfaz cambia de estado, se llama a una función del *plug-in* para ejecutar el cambio. Existen diversos paradigmas para manejar la comunicación entre el cliente y la biblioteca de enlace dinámico al momento que el usuario realiza cambios en la interfaz gráfica. En el caso de los *plug-ins* desarrollados en *RackAFX* el cliente crea, mantiene y destruye la interfaz gráfica de usuario pero esta se comunica directamente con el *plug-in*, sin utilizar al cliente como intermediario.

6.3.2. Interfaz de programación de aplicación

Para que un cliente pueda comunicarse con un *plug-in* es necesario que este conozca las funciones accesibles del último. Para garantizar un estándar que permita la comunicación entre el cliente y la biblioteca se utilizan interfaces de programación de aplicación o APIs por sus siglas en inglés (applications programming interface). Esta es una clase abstracta que contiene todas las funciones a implementar y atributos de un *plug-in*. Si un cliente conoce las funciones de una API, conoce las funciones de todos los *plug-ins* derivados de esta. Algunos ejemplos de APIs para el desarrollo de *plug-ins* en la industria de audio son Direct-X[®], VST[®], AU[®], y AXX[®] [14]. El software RackAFX cuenta con la implementación de la API de VST3 en la plantilla de cada proyecto, permitiendo al usuario sobrescribir las funciones necesarias para dar la funcionalidad requerida al *plug-in*.

En la API utilizada en la plantilla de proyectos en RackAFX se puede evidenciar las funciones que el cliente requiere "llamar" para la obtención de las muestras de sonido. La función a partir de la cual el cliente obtiene dichas muestras es "processAudioFrame", la cual debe ser sobrescrita por el usuario ya que los datos de salida para los canales de audio izquierdo y derecho están inicialmente fijados a cero.

6.4. Sistemas dinámicos

Un sistema dinámico en tiempo continuo (Es necesario hacer la aclaración ya que en la literatura se puede encontrar referencias a mapas iterativos como sistemas dinámicos), es definido por S. Wiggins [22] en su forma más general por la siguiente ecuación:

$$\dot{x} = f(x(t), t; \mu) \quad (1)$$

Donde $x \in U \subset \mathbb{R}^n$, $t \in \mathbb{R}^1$ y $\mu \in V \subset \mathbb{R}^p$. Además U y V son sets abiertos en \mathbb{R}^n y \mathbb{R}^p , respectivamente. La notación punto representa la derivada respecto al tiempo y μ son los parámetros de los cuales depende el sistema.

En este trabajo se estará tratando con sistemas dinámicos autónomos. Esto implica que no hay una dependencia explícita de la variable independiente, t .

$$\dot{x} = f(x(t); \mu) \quad (2)$$

Las variables dependientes describen el estado del sistema en un tiempo dado, por lo cual se denominan variables de estado. El espacio representado por las variables de estado se denomina espacio de fase [22]. Las soluciones al sistema dinámico representadas en el espacio de fase se denominan trayectorias. Cada trayectoria se encuentra a partir de un punto particular en el espacio de fase en un tiempo inicial. Este punto representa las condiciones iniciales del sistema.

$$x(t_0) = x_0 \quad (3)$$

Al no haber dependencia explícita del tiempo en un sistema autónomo se considera $t_0 = 0$.

6.4.1. Estabilidad de sistemas dinámicos autónomos

Un concepto importante para definir la estabilidad local de un sistema dinámico es el concepto de Punto fijo o punto de equilibrio. considerando la ecuación 2, se considera que un Punto fijo es:

$$f(x^*) = 0 \quad (4)$$

Esto quiere decir que el campo vectorial en el espacio de fase es igual a cero en el punto fijo. Se dice que un sistema es estable en la vecindad del punto x^* si una trayectoria con condiciones iniciales cercanas al punto fijo, se mantiene cercana al punto para todo tiempo futuro. El punto x^* es considerado asintóticamente estable si una trayectoria que comienza cercana al punto converge a este cuando el tiempo tiende a infinito. Los conceptos descritos anteriormente son definidos de forma formal en *Introduction to Applied Nonlinear Dynamical Systems and Chaos* [22] de la siguiente manera:

Estabilidad en el sentido de Lyapunov. x^* es estable si, dado $\gamma > 0$, entonces existe un $\delta = \delta(\gamma) > 0$ tal que, para toda solución, $y(t)$ de la ecuación 2, satisface que $|x^* - y(t_0)| < \delta$, entonces $|x^* - y(t)| < \varepsilon$ para todo $t > t_0$.

Estabilidad asintótica. Se dice que x^* es asintóticamente estable si es estable en el sentido de Lyapunov y además para cada solución $y(t)$, de la ecuación 2, existe una constante $b > 0$ tal que, si $|x^* - y(t_0)| < b$, entonces $\lim_{t \rightarrow \infty} |x^* - y(t)| = 0$.

El concepto de estabilidad puede ser extendido a una órbita, la cual está compuesta por un set de puntos. Para ejemplificar en que consiste una órbita se considera el ejemplo 0.0.1 de [22], que considera el siguiente sistema dinámico:

$$\begin{aligned} \dot{u} &= v \\ \dot{v} &= -u \end{aligned} \tag{5}$$

El cual tiene una solución cerrada para las condiciones iniciales $t_0 = v_0 = 0$ y $u_0 = 1$

$$(u(t), v(t)) = (\cos(t), -\sin(t)) \tag{6}$$

Al interpretar geoméricamente la solución notamos que las trayectorias del sistema dinámico se desenvuelven en la órbita dada por el círculo $u^2 + v^2 = 1$. En la Figura 7 se puede observar la interpretación geométrica de una trayectoria en el espacio de fase, una curva integral y una órbita.

Teniendo una idea geométrica de lo que representa una órbita es posible definir la estabilidad alrededor de una órbita, sin embargo es necesario hacer algunas definiciones adicionales previo a la definición formal de la estabilidad orbital. Las siguientes definiciones fueron tomadas del texto de S. Wiggins [22]. La órbita positiva que pasa por el punto x_0 , para $t \geq t_0$ se define como:

$$O^+(x_0, t_0) = \{x \in \mathbb{R}^n | x = x^*(t), t \geq t_0, x^*(t_0) = x_0\} \tag{7}$$

luego considerando un set de puntos arbitrario $S \in \mathbb{R}^n$ y un punto arbitrario $p \in \mathbb{R}^n$ se define la distancia entre punto p y el set S como:

$$d(p, S) = \inf_{x \in S} |p - x| \tag{8}$$

con las dos definiciones anteriores es posible definir la estabilidad orbital.

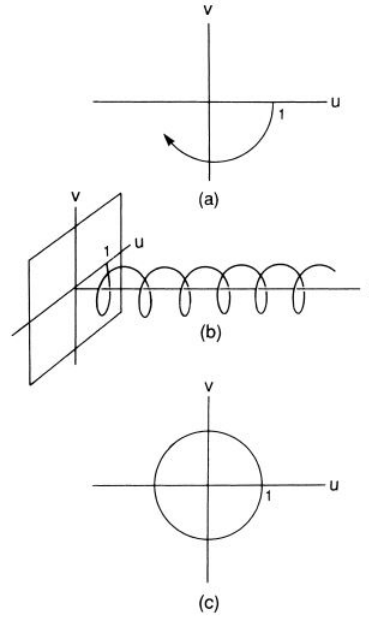


Figura 7: a) Solución a partir de $(1,0)$. b) Curva integral. c) Órbita que define la solución [22]

Estabilidad orbital. $x^*(t)$ se dice ser orbitalmente estable si, dado $\gamma > 0$, existe un $\delta = \delta(\gamma) > 0$ tal que, para cualquier otra solución $y(t)$, de 2 satisfaciendo $|x^*(t_0) - y(t_0)| < \delta$, entonces $d(y(t), O^+(x_0, t_0)) < \gamma$ para $t > t_0$.

Estabilidad orbital asintótica. Se dice que $x^*(t)$ es asintóticamente orbitalmente estable si es orbitalmente estable y para cualquier otra solución, $y(t)$, de 2, existe una constante $b > 0$ tal que, si $|x^*(t_0) - y(t_0)| < b$, entonces $\lim_{t \rightarrow \infty} d(y(t), O^+(x_0, t_0)) = 0$.

Es importante recalcar que la gran mayoría de los sistemas dinámicos se presentan como un sistema no lineal. Estos son todos los sistemas que no se pueden representar de la siguiente forma:

$$\dot{x} = Ax \tag{9}$$

donde A es una matriz de coeficientes constantes $\in \mathbb{R}^{n \times n}$. Los sistemas dinámicos no lineales no suelen tener soluciones cerradas, por lo tanto se opta por la utilización de métodos numéricos para la aproximación de trayectorias en el espacio de fase.

6.4.2. Soluciones numéricas de sistemas autónomos

Considérese el sistema dinámico autónomo no lineal con n variables de estado mostrado en la ecuación 10.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix} \quad (10)$$

Es posible encontrar soluciones numéricas de dicho sistema haciendo uso de métodos iterativos de integración numérica, como el método de Euler y sus versiones modificadas como los métodos de Runge-Kutta. Los métodos descritos a continuación serán desarrollados en su versión unidimensional para posteriormente ser extendidos a \mathbb{R}^n . Las ecuaciones diferenciales a tratar serán de la siguiente forma.

$$\dot{x} = f(t, x) \quad (11)$$

Nótese que a pesar de limitarse a sistemas autónomos (ecuación 10) en el presente trabajo, es posible resolver sistemas variantes en el tiempo con los métodos descritos.

Método de Euler

El método de Euler, también conocido como método de Euler-Cauchy o de punto-pendiente, está dado por la siguiente expresión iterativa [3].

$$x_{i+1} = x_i + f(t_i, x_i)h \quad (12)$$

donde x_{i+1} es el siguiente punto de la solución a encontrar y x_i es el dato actual de la solución. La pendiente actual está dada por $f(t_i, x_i)$ que corresponde a la ecuación 11 evaluada en las coordenadas x y y actuales. El valor h representa el paso de tiempo con el cual se resuelve para cada punto.

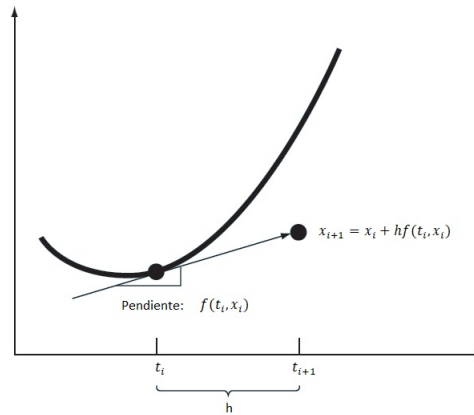


Figura 8: Representación gráfica del método de Euler [3]

De forma cualitativa lo que se hace al aplicar el método de Euler es tomar un punto inicial a partir del cual se encontrará la solución de la ecuación diferencial. Se encuentra la pendiente en este punto, la cual está dada por la expresión para la derivada. Luego se extrapola al siguiente punto multiplicando la pendiente por un paso fijo y adicionando el resultado a la solución del punto anterior, el cual es la condición inicial en la primera iteración.

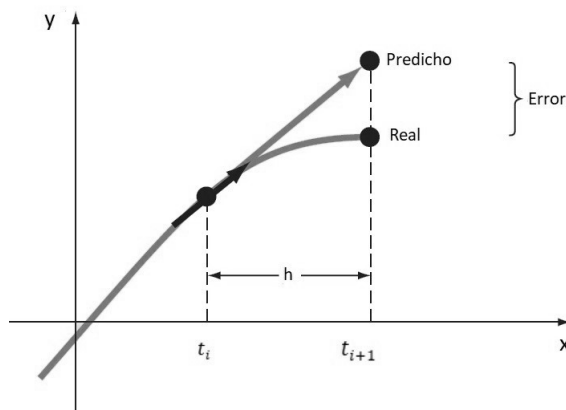


Figura 9: Error en el método de Euler [3]

Como se puede evidenciar en la Figura 9 el resolver una ecuación diferencial de forma numérica conlleva el acarreo de error, el cual se deriva de dos errores distintos. Uno es el error de truncamiento el cual se da localmente en cada iteración y luego es acarreado a lo largo de la solución. El segundo error es por redondeo el cual es resultado de la precisión limitada de las computadoras. Usualmente el error por redondeo es mucho menor que el error por truncamiento.

Es posible evidenciar el por qué del error de truncamiento haciendo uso de la expansión por series de Taylor de la ecuación 12. Tal expansión corresponde a la ecuación 13.

$$x_{i+1} = x_i + f(t_i, x_i)(x_{i+1} - x_i) + \frac{f'(t_i, x_i)}{2!}(x_{i+1} - x_i)^2 + \dots + \frac{f^{(n-1)}(x_i, y_i)}{n!}(x_{i+1} - x_i)^n + \dots \quad (13)$$

Donde $x_{i+1} - x_i$ es igual a h , el paso con el que se da cada iteración. Nótese que el método de Euler utiliza únicamente los dos primeros términos de la serie de Taylor. Por lo tanto el error de truncamiento local está dado por la suma todos los términos de orden $O(h^2)$ y mayor. Como el término de orden cuadrático es el dominante sobre el resto, se dice que el error de truncamiento local para el método de Euler es de orden $O(h^2)$. De igual forma es evidente que el error decae de forma cuadrática al disminuir el valor de h . No es posible calcular el valor exacto del error, sin embargo, una cota superior de este se puede dar de la siguiente forma [25].

$$E_{max} = Mh^2/2 \text{ donde } M = \max_{x_n < x < x_{n+1}} |f''(t, x)| \quad (14)$$

Existe una forma mejorada del método de Euler, para la cual se utiliza el calculo de dos pendientes por iteración.

$$\begin{aligned} x_{i+1} &= y_i + h \frac{f(t_i, x_i) + f^*(t_{i+1}, x_{i+1})}{2} \\ f_{n+1}^* &= x_i + hf(t_i, x_i) \end{aligned} \tag{15}$$

Para el método mejorado se calcula la pendiente en el punto actual y el siguiente y luego se encuentra el siguiente punto de la solución con el promedio de las pendientes encontradas. Para este método el error es de orden $O(h^3)$. La contraparte de la reducción del error de truncamiento local en la implementación de este método es que se requiere de más recursos computacionales por iteración.

Método de Runge-Kutta

La fórmula para el método de Euler mejorado 15 puede ser generalizada para una pendiente calculada con el promedio ponderado de n pendientes en el intervalo $t_n \leq t \leq t_{n+1}$.

$$x_{n+1} = x_n + h(w_1k_1 + w_2k_2 + \dots + w_nk_n) \tag{16}$$

donde $w_1 + w_2 + \dots + w_n = 1$ y k_i es la función evaluada en un punto entre el intervalo definido por h . A este conjunto general de métodos iterativos para la solución de ecuaciones diferenciales se les conoce como métodos de Runge-Kutta y el orden de los mismos está dado por el subíndice n . Usualmente se abrevia el método utilizado de la forma RKn , donde n es el orden del método de Runge-Kutta implementado. Bajo esta definición general descrita el método de Euler sería un método Runge-Kutta de primer orden. Los parámetros del método se eligen de modo que la ecuación 16 cumpla con el polinomio de Taylor de grado n [25].

Uno de los métodos de Runge-Kutta más utilizados debido a su precisión y fácil implementación es el método de cuarto orden o RK4.

$$\begin{aligned} y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(t_i, x_i) \\ k_2 &= f(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_1) \\ k_3 &= f(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_2) \\ k_4 &= f(t_i + h, x_i + hk_3) \end{aligned} \tag{17}$$

Como este método utiliza los primeros cuatro términos del polinomio de Taylor para hacer la aproximación a la solución, el error resultante es de orden $O(h^5)$. mientras que la complejidad en tiempo de recursos computacionales es de orden $O(n)$ ya que se requieren $4n$ cálculos de la pendiente para hacer n iteraciones del método.

Nótese que la aplicación de los métodos de Runge-Kutta puede ser extendida fácilmente para encontrar la solución de sistemas de ecuaciones diferenciales de primer orden. Por simpleza se muestra el ejemplo de RK4 para un sistema de dos ecuaciones, sin embargo, a partir del mismo se puede deducir la forma para sistemas de un mayor orden. Suponga que se tiene un sistema de la siguiente forma.

$$\begin{aligned}\dot{x} &= f_1(t, x, y) \\ \dot{y} &= f_2(t, x, y)\end{aligned}\tag{18}$$

Donde el método iterativo está dado por:

$$\begin{aligned}x_{i+1} &= x_i + \frac{h}{6}(k_{x1} + 2k_{x2} + 2k_{x3} + k_{x4}) \\ y_{i+1} &= y_i + \frac{h}{6}(k_{y1} + 2k_{y2} + 2k_{y3} + k_{y4})\end{aligned}\tag{19}$$

Y las pendientes son calculadas de la siguiente forma.

$$\begin{aligned}k_{x1} &= f_1(t_i, x_i, y_i) \\ k_{y1} &= f_2(t_i, x_i, y_i) \\ k_{x2} &= f_1\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_{x1}, y_i + \frac{1}{2}hk_{y1}\right) \\ k_{y2} &= f_2\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_{x1}, y_i + \frac{1}{2}hk_{y1}\right) \\ k_{x3} &= f_1\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_{x2}, y_i + \frac{1}{2}hk_{y2}\right) \\ k_{y3} &= f_2\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}hk_{x2}, y_i + \frac{1}{2}hk_{y2}\right) \\ k_{x4} &= f_1(t_i + h, x_i + k_{x3}h, y_i + k_{y3}h) \\ k_{y4} &= f_2(t_i + h, x_i + k_{x3}h, y_i + k_{y3}h)\end{aligned}\tag{20}$$

Es importante recalcar que los métodos descritos pueden ser utilizados para encontrar las trayectorias de sistemas dinámicos ya que los sistemas planteados hasta el momento se pueden expresar en la forma general de un sistema dinámico autónomo.

Método de Runge-Kutta adaptativo

Los métodos adaptativos de Runge-Kutta son aquellos que tienen un paso variable, el cual cambia según la estimación del error de truncamiento actual. Estos tienen la finalidad de minimizar la cantidad de iteraciones a realizar para la estimación de soluciones a una ecuación diferencial. Son particularmente útiles para soluciones con cambios abruptos en el tiempo. El control adaptativo se puede incorporar de dos forma: Estimando el error entre dos predicciones con diferente paso utilizando un RK del mismo orden o estimando el error de truncamiento local como la diferencia entre dos métodos RK de distinto Orden. En este caso se tratará la segunda opción, conocidos como métodos de Runge-Kutta Fehlberg.

Un inconveniente de los métodos RK Fehlberg es que requieren de cálculos para cada método RK de distinto orden implementado. Este problema se ha solucionado con la implementación de métodos encapsulados. En ellos el métodos RK de orden superior utiliza las evaluaciones del método de orden inferior. Uno de estos métodos es el método RK Cash-Karp, el cual utiliza métodos RK de cuarto y quinto orden para hacer la estimación del error de truncamiento local. Se procede a mostrar la implementación del método como se realizó en el artículo original por Jeff Cash y Alan Karp [16].

$$\begin{aligned}
y_{i+1} &= y_i + \left(\frac{37}{378}k_1 + \frac{250}{621}k_3 + \frac{125}{594}k_4 + \frac{512}{1771}k_6 \right) h \\
y_{i+1} &= y_i + \left(\frac{2825}{27648}k_1 + \frac{18575}{43384}k_3 + \frac{13525}{55296}k_4 + \frac{277}{14336}k_5 + \frac{1}{4}k_6 \right) h
\end{aligned} \tag{21}$$

donde las pendientes k_n están dadas por:

$$\begin{aligned}
k_1 &= f(t_i, x_i) \\
k_2 &= f\left(t_i + \frac{1}{5}h, x_i + \frac{1}{5}k_1h\right) \\
k_3 &= f\left(t_i + \frac{3}{10}h, x_i + \frac{3}{40}k_1h + \frac{9}{40}k_2h\right) \\
k_4 &= f\left(t_i + \frac{3}{5}h, x_i + \frac{3}{10}k_1h + \frac{9}{10}k_2h + \frac{6}{5}k_3h\right) \\
k_5 &= f\left(t_i + h, x_i - \frac{11}{54}k_1h + \frac{5}{2}k_2h - \frac{70}{27}k_3h + \frac{35}{27}k_4h\right) \\
k_6 &= f\left(t_i + \frac{7}{8}h, x_i + \frac{1631}{55296}k_1h + \frac{175}{512}k_2h + \frac{575}{13824}k_3h + \frac{44275}{110592}k_4h + \frac{253}{4096}k_5h\right)
\end{aligned} \tag{22}$$

con el calculo de los dos valores para y_{i+1} es posible estimar el error como el valor absoluto de la resta de ambos valores.

El control del paso se hace de la siguiente forma:

$$h_{nuevo} = h_{actual} \left| \frac{\Delta_{nuevo}}{\Delta_{actual}} \right|^\alpha \tag{23}$$

donde Δ_{nuevo} es la exactitud deseada y Δ_{actual} es la exactitud actual. El exponente α corresponde a 0.2 si $\Delta_{actual} \leq \Delta_{nuevo}$ lo cual implica un incremento en el paso, y a 0.25 si $\Delta_{actual} \geq \Delta_{nuevo}$ lo cual es una disminución del paso. Una forma sencilla de calcular Δ_{nuevo} es

$$\Delta_{nuevo} = \psi y_{escala} \tag{24}$$

donde ψ representa el nivel de tolerancia global y y_{escala} determina el escalamiento del error [3]. Una selección habitual es $y_{escala} = y$. Para obtener errores relativos constantes se recomienda utilizar:

$$y_{escala} = |y| + \left| h \frac{dy}{dx} \right| \quad (25)$$

6.4.3. Ciclos límite

Con la aproximación de trayectorias por métodos numéricos es posible identificar un tipo especial de órbita. Un ciclo límite es una órbita en el espacio de fase hacia la cual convergen las trayectorias de un sistema independientemente de sus condiciones iniciales. En el texto *Pattern Formations and Oscillatory Phenomena* [5] se expresa que un ciclo límite generalmente se da a partir de una condición denominada el teorema de Liénard. Este teorema es aplicable para ecuaciones de la forma:

$$\ddot{x} + f(x)\dot{x} + g(x) = 0 \quad (26)$$

Las condiciones para generar un ciclo límite en la ecuación 26 son las siguientes:

1. Las derivadas de $f(x)$ y $g(x)$ son continuas.
2. $g(x)$ es una función impar de x , mientras $f(x)$ es una función par.
3. Para $x > 0$ se tiene $g(x) > 0$.
4. Una constante positiva α satisface las condiciones $F(x) < 0$ para $0 < x < \alpha$, $F(\alpha) = 0$ y $F(x)$ es positiva e incrementa monótonamente con x para $\alpha < x$ donde $F(x) = \int_0^x f(\gamma)d\gamma$.

La presencia de un ciclo límite en un sistema dinámico corresponde a un comportamiento periódico en las variables de estado.

6.4.4. Caos y atractores extraños

Características de un sistema caótico

Un fenómeno frecuentemente encontrado en los sistemas dinámicos no lineales es el caos. Aunque hay discrepancias que no han permitido una definición global del caos, existen características fundamentales con las que cumple un sistema caótico. Steven Strogatz [19] describe el caos de la siguiente forma: Es un comportamiento aperiódico en el largo plazo de un sistema determinista que presenta alta dependencia a las condiciones iniciales. Posteriormente desglosa las características importantes de la siguiente forma.

- Comportamiento aperiódico a lo largo del tiempo: Las trayectorias no convergen a órbitas periódicas o cuasi-periódicas mientras $t \rightarrow \infty$.

- Es un comportamiento determinista: Esto quiere decir que el comportamiento del sistema no se rige por parámetros estocásticos. En su lugar el comportamiento caótico surge de las no-linealidades del sistema.
- Sensibilidad a condiciones iniciales: Las trayectorias con distintas condiciones iniciales divergen unas de otras con una rapidez exponencial. Esto implica que el sistema tiene un exponente de Lyapunov positivo.

Exponentes de Lyapunov

S. Lynch [7] define cualitativamente los exponentes de Lyapunov como cantidades que caracterizan el rango de separación para trayectorias infinitesimalmente cercanas. Por tal característica estos se han utilizado para definir si un sistema es caótico ya que son una forma de cuantificar la sensibilidad a condiciones iniciales de un sistema dinámico. Una definición más formal se da en el artículo *Determining Lyapunov exponents from a time series* [23]. Se plantea un espacio de fase n -dimensional, donde se estudia la evolución a largo plazo de una esfera infinitesimal de n dimensiones. La esfera es deformada a un elipsoide debido a la deformación producida por el flujo del campo vectorial. El i -ésimo exponente de Lyapunov esta dado en términos del eje principal del elipsoide, $p_i(t)$.

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)} \quad (27)$$

Un sistema n -dimensional tendrá n exponentes de Lyapunov. Para sistemas tridimensionales con coeficientes de Lyapunov λ_1, λ_2 y λ_3 , se tiene el siguiente comportamiento. Para atractores caóticos $\lambda_1 > 0, \lambda_2 = 0$ y $\lambda_3 < 0$. Para ciclos límite $\lambda_1 = 0, \lambda_2 < 0$ y $\lambda_3 < 0$. Un toroide corresponde a $\lambda_1 = \lambda_2 = 0$ y $\lambda_3 < 0$.

En el texto *A Gallery of Chua Attractors* [13] se presenta de forma más general las condiciones de los exponentes de Lyapunov para que un sistema presente un atractor caótico.

$$\lambda_1 > 0 = \lambda_2 > \lambda_3 \text{ y } \sum_{i=1}^3 \lambda_i \quad (28)$$

Si las condiciones anteriores se cumplen se dice que las trayectorias del sistema estarán confinadas en una región finita de espacio. Esta región está definida por un set de puntos denominado atractor, que presenta estabilidad en su localidad. Cuando un atractor presenta caos, se denomina atractor extraño o atractor caótico.

6.5. Ejemplos de sistemas dinámicos no lineales

6.5.1. El oscilador de Van der Pol

El oscilador de van der Pol fue descubierto por Balthasar van der Pol en 1926. Este es un sistema no lineal, no conservativo y estable. Está descrito por la siguiente ecuación diferencial de segundo orden:

$$\ddot{x} - \varepsilon(1 - x^2)\dot{x} + x = 0 \quad (29)$$

Esta ecuación puede ser reducida a un sistema de dos ecuaciones de primer orden.

$$\begin{cases} \dot{x} = y \\ \dot{y} = \varepsilon(1 - x^2)y - x \end{cases} \quad (30)$$

S. Kinoshita [5] realiza un breve análisis de las características fundamentales del oscilador de van der Pol. Primero cabe notar que el sistema depende únicamente de un parámetro. Cuando $\varepsilon \rightarrow 0$ notamos que el sistema adquiere la forma de un oscilador armónico simple. Cuando ε aumenta la no linealidad se hace más dominante sobre el sistema. La respuesta del oscilador de van der Pol se da en forma de un ciclo límite ya que este cumple con el teorema de Liénard, siempre que $\varepsilon > 0$. Para entender de mejor forma como el parámetro ε afecta el comportamiento del sistema se aplica la transformada de Liénard. Tras la aplicación de la transformada el sistema adquiere la siguiente forma:

$$\begin{cases} \dot{x} = \varepsilon(x - \frac{x^3}{3} - y) \\ \dot{y} = \frac{x}{\varepsilon} \end{cases} \quad (31)$$

Se observa que mientras ε aumenta la forma del ciclo límite difiere más de un círculo, que se presenta en el caso $\varepsilon = 0$. Este fenómeno es consistente con el siguiente análisis. Fijando $\dot{x} = 0$ se obtiene la curva $y = x - \frac{x^3}{3}$ la cual es denominada una isocline de crecimiento nulo (*nullcline* en inglés). Se puede encontrar otra isocline de crecimiento nulo al hacer $\dot{y} = 0$. Al analizar el campo vectorial producto de las isoclinas de crecimiento nulo se puede observar como el ciclo límite se rige de forma pronunciada por la curva $y = x - \frac{x^3}{3}$.

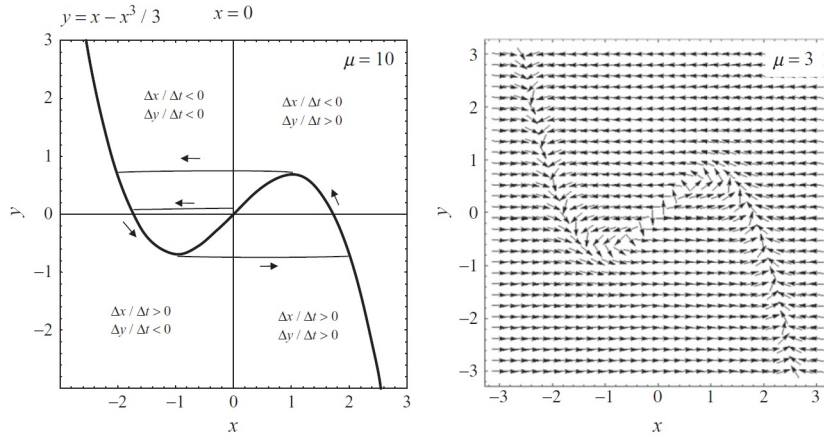


Figura 10: Origen del ciclo límite en el oscilador de Van der Pol [13]

Es posible encontrar más evidencia del ciclo límite al hacer una linealización del sistema entorno al punto fijo en (x^*, y^*) tal que \dot{x} y \dot{y} son iguales a cero. Este punto está en la coordenada $(0, 0)$ del espacio de fase.

teniendo la linealización del sistema

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \varepsilon & -\varepsilon \\ \frac{1}{\varepsilon} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (32)$$

es posible encontrar los siguientes eigen-valores

$$\lambda = \frac{\varepsilon \pm \sqrt{\varepsilon^2 - 4}}{2} \quad (33)$$

cuando $\varepsilon > 0 \Rightarrow Re\{\lambda_{1,2}\} > 0$, lo cual implica que el punto $(x_0, y_0) = (0, 0)$ es absolutamente inestable. Este comportamiento explica por qué las trayectorias nunca convergen al origen, caso que si ocurre cuando $\varepsilon < 0$.

6.5.2. El oscilador de Chua

El oscilador de Chua surge de un circuito no lineal que se comporta de forma caótica para ciertos parámetros. El circuito general se muestra a continuación.

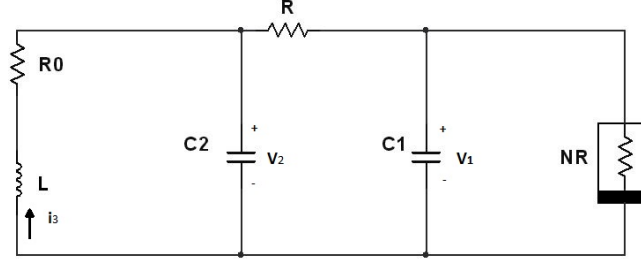


Figura 11: Circuito del oscilador de Chua [5]

El último componente de izquierda a derecha es conocido como el diodo de Chua. Este es modelado usualmente como una resistencia no lineal. En tal modelo la corriente está dada por la siguiente expresión.

$$I = f(V) = G_b V + \frac{1}{2}(G_a - G_b)(|V + E| - |V - E|) \quad (34)$$

Donde los parámetros G_a y G_b son las pendientes en distintos tramos de la función del diodo de Chua y E el valor que define los límites de los mismos. Véase la Figura 34.

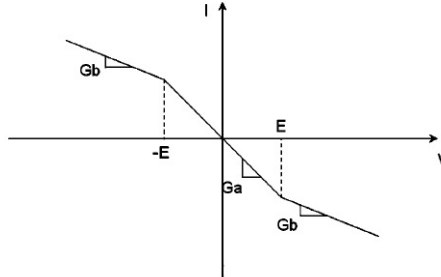


Figura 12: El diodo de Chua como una función por partes [13]

El comportamiento del circuito puede ser modelado por el siguiente sistema dinámico.

$$\begin{cases} \frac{dV_1}{dt} = \frac{1}{C_1}[(V_2 - V_1)G - f(V_1)] \\ \frac{dV_2}{dt} = \frac{1}{C_2}[(V - 1 - V_2)G + i_3] \\ \frac{di_3}{dt} = -\frac{1}{L}(V_2 + R_0 i_3) \end{cases} \quad (35)$$

donde el parámetro G es igual a $\frac{1}{R}$. Haciendo las siguientes sustituciones es posible obtener la conocida forma adimensional del oscilador de chua.

$$\begin{aligned}
x &= \frac{V_1}{E} & y &= \frac{V_2}{E} & z &= \frac{Ri_3}{E} \\
\alpha &= \frac{C_2}{C_1} & \beta &= \frac{R^2C_2}{L} & \gamma &= \frac{R_0RC_2}{L} \\
a &= RG_a & b &= RG_b & \tau &= \frac{t}{|RC_2|} \\
\kappa &= \text{sgn}(RC_2)
\end{aligned}$$

Lo cual lleva al siguiente sistema

$$\begin{cases} \frac{dx}{d\tau} = \kappa\alpha(y - x - f(x)) \\ \frac{dy}{d\tau} = \kappa(x - y + z) \\ \frac{dz}{d\tau} = -\kappa(\beta y + \gamma z) \end{cases} \quad (36)$$

donde $f(x)$ es el modelo adimensional del diodo de Chua y está dado por:

$$f(x) = bx + \frac{1}{2}(a - b)(|x + 1| - |x - 1|) \quad (37)$$

Obsérvese la no linealidad del sistema es introducida por la función $f(x)$. Este hecho a llevado al desarrollo de distintos modelos para caracterizar de mejor forma el comportamiento del diodo de Chua. En el artículo *Implementation of Chua's circuit with a cubic nonlinearity* [24] se manifiesta que la función por partes, presentada en la ecuación 37, presenta dificultades al momento de capturar la dinámica del circuito. Por lo que se propone una función no lineal suave en forma de un polinomio cúbico. Para observar comparaciones gráficas entre la respuesta obtenida con la ecuación 37 y la ecuación 38 refiérase al artículo *A gallery of attractor from smooth Chua's equation* [21].

$$f(x) = h_1x + h_3x^3 \quad (38)$$

Incluso se ha llegado a desarrollar un modelo del oscilador de chua para la generación de atractores de n volutas (encontrado en la literatura como *n-scroll attractors*). Esto surge de generalizaciones del modelo que han permitido cambiar las pendientes y puntos de quiebre de la no linealidad. El modelo para dicho oscilador está dado en [20] como se muestra a continuación

$$\begin{cases} \frac{dx}{dt} = \alpha(y - h(x)) \\ \frac{dy}{dt} = x - y + z \\ \frac{dz}{dt} = -\beta y - \gamma z \end{cases} \quad (39)$$

donde la no-linealidad con distintas pendientes y puntos de quiebre se da de forma general por la siguiente ecuación.

$$h(x) = m_{2q-1}x + \frac{1}{2} \sum_{i=1}^{2q-1} (m_{i-1} - m_i) \times (|x + c_i| - |x - c_i|) \quad (40)$$

6.5.3. El atractor de Lorenz

Edward Lorenz, un meteorólogo del instituto tecnológico de Massachusetts, propuso las siguientes ecuaciones como un modelo sobre-simplificado del comportamiento de corrientes de convección en la atmósfera.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = rx - y - xz \\ \dot{z} = xy - bz \end{cases} \quad (41)$$

Para ver de forma detallada el desarrollo del modelo refiérase al artículo original de Edward Lorenz [6]. El modelo propuesto por Lorenz fue uno de los primeros ejemplos donde se observó el fenómeno del caos. Esto llevó a Lorenz a ser reconocido como el padre del caos. De sus conclusiones se deriva el popularmente conocido efecto mariposa. Este es un ejemplo llamativo de la alta sensibilidad a condiciones iniciales de los sistemas caóticos. El batir de las alas una mariposa en Londres puede provocar o evitar el desarrollo de un huracán en las Bahamas en un futuro no tan lejano.

En el texto *Dynamical Systems with Applications using MATLAB* [7] se indican varias características fundamentales de las ecuaciones de Lorenz.

- El sistema 41 presenta simetría $(x, y, z) \Rightarrow (-x, -y, z)$
- El eje z presenta simetría.
- El flujo de volumen en el espacio se contrae ya que $\nabla X = -(\sigma + b + 1) < 0$. La divergencia del campo vectorial correspondiente al sistema es negativa.
- Si $0 < r < 1$, el origen es el único punto de equilibrio y es un atractor global.
- Cuando $r = 1$ ocurre una bifurcación y surgen dos puntos de equilibrio adicionales, C_1 y C_2 .
- Cuando $r \approx 24.06$, se crea la existencia de un atractor extraño.
- Si $1 < r < r_0$, donde $r_0 \approx 24.74$, el origen es un punto fijo inestable y C_1 y C_2 son puntos fijos estables.
- Cuando $r > r_0$, C_1 y C_2 pierden su estabilidad.

Los puntos fijos C_1 y C_2 están dados por la siguiente expresión.

$$\begin{aligned} C_1 &= (\sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1) \\ C_2 &= (-\sqrt{b(r-1)}, -\sqrt{b(r-1)}, r-1) \end{aligned} \quad (42)$$

Para el desarrollo y cumplimiento de los objetivos se estableció un procedimiento secuencial a seguir que puede ser englobado en tres grandes secciones.

7.1. Selección de los sistemas dinámicos a implementar

La primera parte y quizá la más importante para el desarrollo del proyecto es la selección de los sistemas dinámicos que se utilizarán para realizar la síntesis de audio. Para seleccionar estos sistemas se busca que estos cumplan con varias características:

- La respuesta del sistema debe estar acotada en todo momento (esto implica que el sistema dinámico es estable).
- Debe existir una frecuencia dominante diferente a cero en el contenido espectral de al menos una de las variables de estado.
- La frecuencia dominante debe permanecer inmutable independientemente de las condiciones iniciales del sistema.
- La frecuencia dominante debe ser constante a lo largo del tiempo.

Si el sistema presenta comportamiento periódico el término frecuencia dominante debe ser sustituido por frecuencia fundamental. Al cumplirse todas las características listadas, se considera que el sistema será un candidato válido. Se busca realizar esta selección según la literatura existente, sin necesidad de realizar análisis formales adicionales.

7.2. Simulaciones experimentales de los sistemas seleccionados

Con los modelos necesarios se procederá a simular la respuesta del oscilador en software utilizando la aplicación Matlab de MathWorks[®]. Este software se ha seleccionado debido a la diversidad de herramientas y librerías que brinda. Algunas de las facilidades proporcionadas por Matlab son: gráficos en tres dimensiones, transformada rápida de Fourier, herramienta para el diseño de filtros digitales, espectrogramas y métodos de integración numérica.

La respuesta de los sistemas se obtendrán solucionandolos de forma numérica. Estableciendo una frecuencia de muestreo estandarizada en la industria de audio (44.1 KHz, 48 KHz, 192 KHz, etc.) y una resolución de 16 bits o 24 bits por dato, se transformará la respuesta del sistema a una señal de audio. Según la caracterización matemática del sistema se tratará de modular los parámetros que afectan la respuesta del mismo para obtener un sonido aplicable a la industria de audio. Para obtener dicho sonido es posible que se requiera de etapas adicionales como el filtrado de la señal generada, aplicación de efectos de audio o manipular el modelo para limitar su comportamiento. El resultado óptimo resultaría de poder modular con precisión la frecuencia dominante o fundamental del sistema dinámico, según sea el caso.

7.3. Creación de *plug-in* VST

Tras la etapa de simulación se procederá a la implementación de los algoritmos de síntesis acorde a los resultados obtenidos. Para asegurar la aplicabilidad de los algoritmos en la industria de audio se hará uso del software RackAFX, el cual facilita la implementación de *plug-ins* con la interfaz de programación VST. El objetivo de la implementación en forma de un *plug-in* VST es que el instrumento virtual pueda ser utilizado en cualquier estación de trabajo de audio virtual (Software DAW). Se espera que el resultado final sea un instrumento virtual que reciba comandos provenientes de un controlador MIDI.

Análisis de los sistemas dinámicos a implementar

La presencia de un ciclo límite en las trayectorias de un sistema asegura el cumplimiento de todas las condiciones listadas en la metodología. Debido a ello uno de los sistemas seleccionados es el oscilador de Van der Pol, ya que este es un sistema no lineal cuyas trayectorias son estables y periódicas.

Los otros dos sistemas seleccionados son el oscilador de Chua y el atractor de Lorenz. La dinámica de dichos sistemas se rige por la presencia de un atractor extraño. Un atractor se caracteriza por ser estable, lo cual implica que mientras las trayectorias estén sujetas este, se tendrá una respuesta acotada de las variables de estado. En tanto a la presencia de una frecuencia dominante, se busca encontrar la configuración de parámetros adecuada para que esta exista.

8.1. Oscilador de Van der Pol

El oscilador de Van der Pol se modela con una ecuación diferencial no lineal de segundo orden. Este puede ser expresado como un sistema con dos variables de estado de la siguiente forma:

$$\begin{cases} \dot{x} = y \\ \dot{y} = \varepsilon(1 - x^2)y - z \end{cases} \quad (43)$$

Este es un sistema cuyas soluciones, cuando $\varepsilon > 0$, presentan un ciclo límite. Esto implica que la respuesta del sistema es periódica y eventualmente todas las trayectorias convergen a la misma órbita independientemente de las condiciones iniciales. Este es un comportamiento deseado para realizar síntesis de audio ya que garantiza la estabilidad del sistema y la

presencia de una frecuencia fundamental que no varía con las condiciones iniciales del sistema. Se ha verificado experimentalmente la presencia del ciclo límite encontrando soluciones numéricas al sistema con distintas condiciones iniciales. Tanto trayectorias que comienzan fuera como dentro de la órbita, convergen a esta.

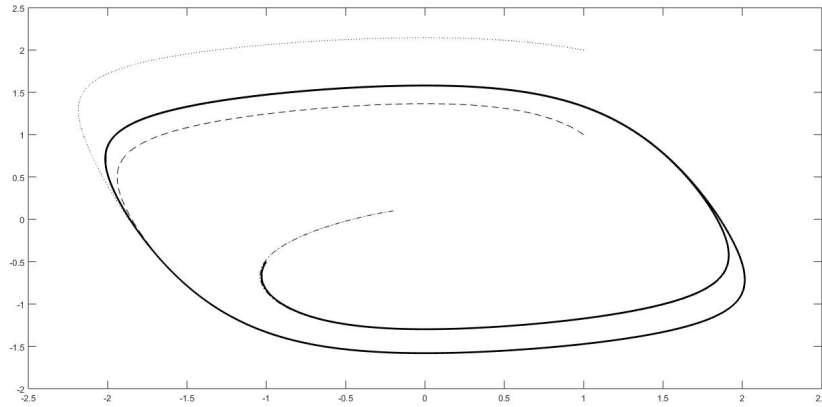


Figura 13: Ciclo límite en el oscilador de Van der Pol

Una característica importante del oscilador de Van der Pol es que este cuenta únicamente con un parámetro. Cuando $\varepsilon \rightarrow 0$, el sistema se reduce a un oscilador armónico simple, cuya solución corresponde a un senoide. Cuando ε aumenta, el término no lineal del sistema se hace más dominante. Según el análisis previo se espera que el contenido espectral del oscilador de Van der Pol aumente en complejidad con el aumento del parámetro ε . Esta conjetura se muestra de forma evidente en el siguiente espectrograma:

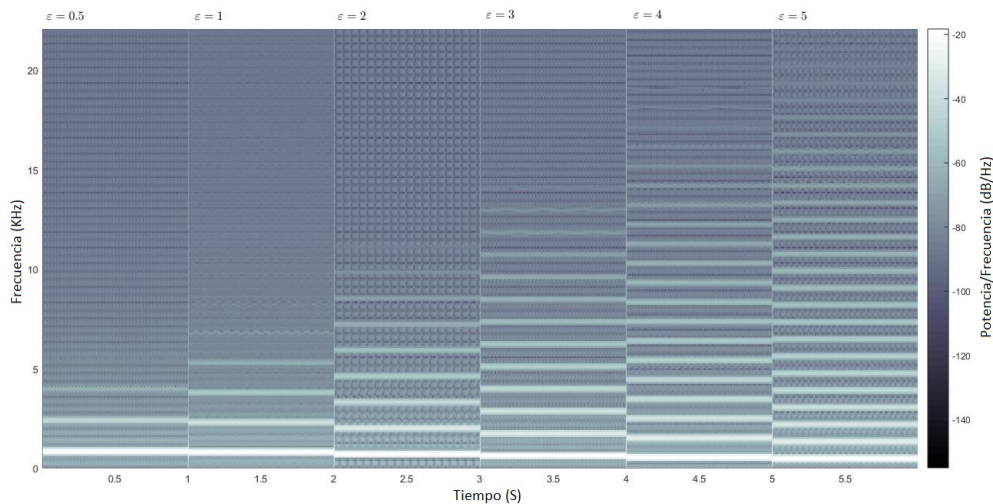


Figura 14: Ejemplos de atractores en el oscilador de Chua

Se puede extraer dos conclusiones adicionales analizando la Figura 14. Lo primero es que el contenido espectral coincide con lo esperado de una señal periódica. Según el análisis de Fourier una señal periódica puede ser representada por una suma infinita de sinusoides con

amplitud decreciente y cuya frecuencia son múltiplos de la fundamental. Este es precisamente el comportamiento mostrado en el espectrograma. Adicionalmente, se puede observar que la frecuencia dominante decrece con el aumento del parámetro ε . Por lo que se evidencia que tanto la forma de la órbita como la frecuencia fundamental del sistema dependen del parámetro ε .

8.2. Oscilador de Chua

La primera acción a tomar con el modelo del oscilador de Chua fue verificar la existencia del atractor extraño. Para ello se tomó como referencia el texto *A Gallery Of Chua Attractors* [13] donde se brinda una colección de parámetros para diferentes modelos del oscilador de Chua, en los cuales se ha verificado la existencia de atractores caóticos. En tanto a la no-linealidad del sistema, se ha seleccionado la ecuación cúbica 38, en lugar de la función por partes 37 con la que se modeló desde un inicio el diodo de Chua. Esto se debe a que según la información recopilada una función suave modela de mejor forma el comportamiento del circuito y se evita la presencia de discontinuidades en las trayectorias. Comportamiento que podría generar aliasing en el espectro de la onda generada.

Aplicando soluciones numéricas del sistema 36 con diferentes parámetros se replicó algunos de los atractores descritos por Pietro Pantano [13]. En la Figura 15 se ejemplifica la diversidad de atractores caóticos que pueden ser obtenidos tan solo con variar los parámetros del sistema.

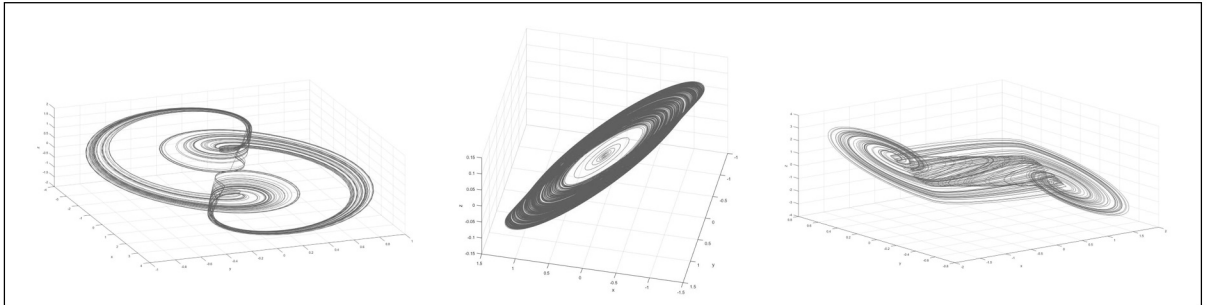


Figura 15: Ejemplos de atractores en el oscilador de Chua

Adicionalmente se brinda imágenes de las trayectorias en el espacio de fase, del comportamiento en el tiempo y del contenido espectral de las variables de estado para cada uno de los atractores en la galería. Dentro del muestrario de atractores que modelan el diodo de Chua con un polinomio cubico, se seleccionó uno cuyo contenido espectral tuviera la presencia clara de una frecuencia dominante. El sistema seleccionado fue el siguiente:

$$\begin{cases} \dot{x} = \kappa(y - x - h(x))\alpha \\ \dot{y} = \kappa(x - y + z) \\ \dot{z} = -\kappa(\beta y + \gamma z) \end{cases} \quad (44)$$

Donde los parámetros del sistema adquieren los siguientes valores.

$$\begin{array}{lll} \kappa = -1 & h_1 = -0.8415410391 & h_3 = -0.0375582129 \\ \alpha = 37 & \beta = 95.721132 & \gamma = -0.8982235 \end{array}$$

Para este sistema se replicó el comportamiento en el espacio de fase, tiempo y frecuencia de las trayectorias.

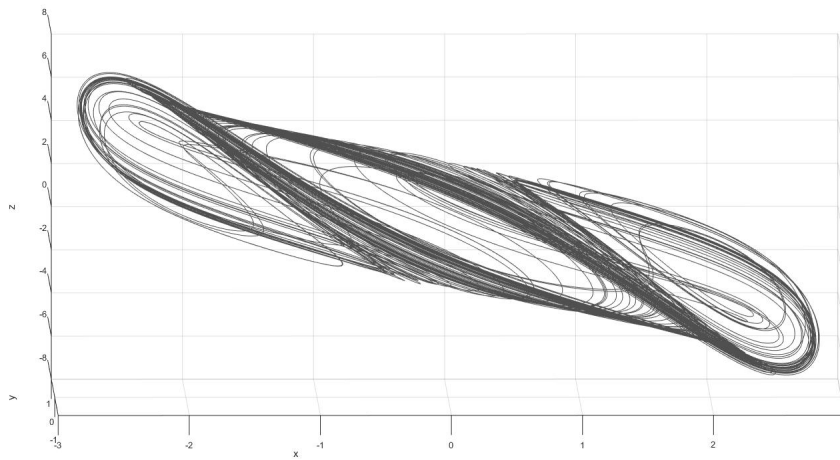


Figura 16: Respuesta en el espacio de fase del sistema 44

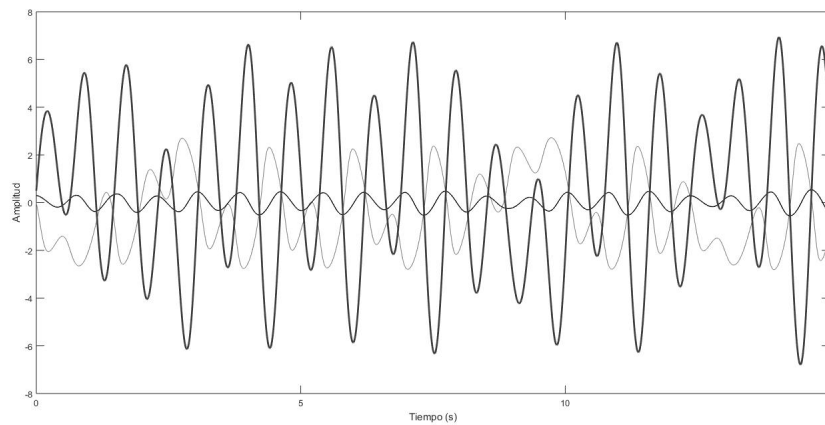


Figura 17: Respuesta en el tiempo de las variables de estado del sistema44

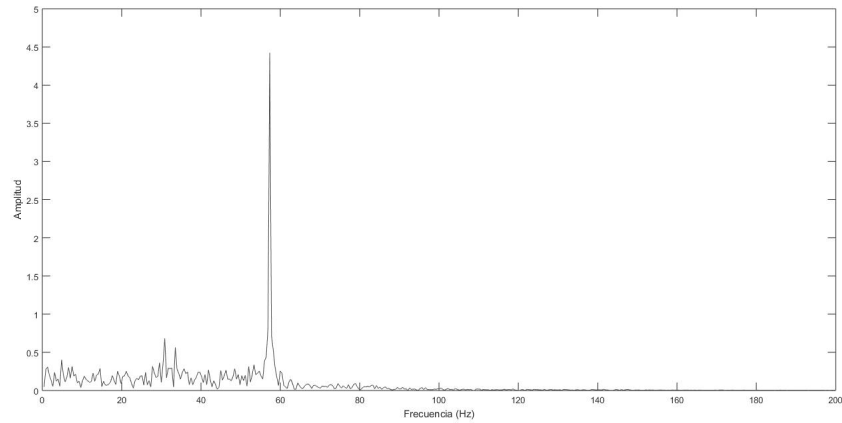


Figura 18: Respuesta en frecuencia de una variable de estado del sistema 44

Es posible corroborar la fuerte dependencia de las condiciones iniciales en un sistema caótico analizando como divergen dos trayectorias con condiciones iniciales cercanas.

$$dist(t) = \sqrt[3]{|x_1(t) - x_2(t)|^2 + |y_1(t) - y_2(t)|^2 + |z_1(t) - z_2(t)|^2} \quad (45)$$

En la ecuación 45 $dist(t)$ es la distancia entre dos trayectorias de un sistema dinámico con tres variables de estado. los subíndices 1 y 2 corresponden a trayectorias distintas con condiciones iniciales cercanas. En la Figura 19 se muestra la evolución en el tiempo de la distancia entre dos trayectorias del oscilador de Chua. Las condiciones iniciales difieren en 1^{-5} unidades para la variable de estado $y(t)$. Obsérvese que existe un crecimiento exponencial en la función $dist(t)$ a pesar de la cercanía inicial de la solución.

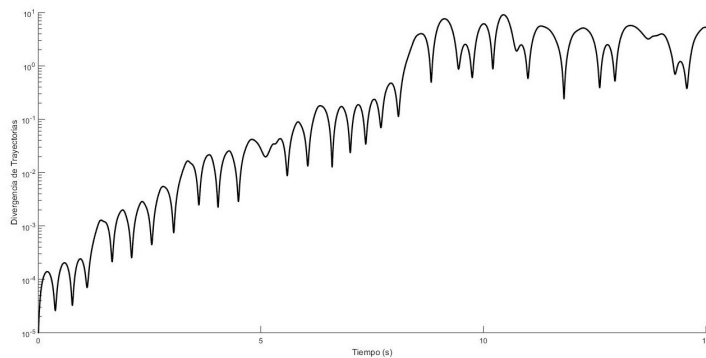


Figura 19: Divergencia de trayectorias con condiciones iniciales cercanas en el oscilador de Chua

Al observar la Figura 18 resulta evidente que existe una frecuencia dominante en el sistema, Sin embargo, este es el contenido espectral promedio de una muestra larga de tiempo. Al ser este un sistema caótico se sabe que la respuesta del sistema es aperiódica, por lo tanto el contenido espectral promedio no es suficiente para garantizar la presencia de

la frecuencia dominante en todo momento. Para realizar un análisis del espectro a lo largo del tiempo se realizó un espectrograma de la señal correspondiente a la variable de estado z , a la cual corresponde el espectro mostrado en 18.

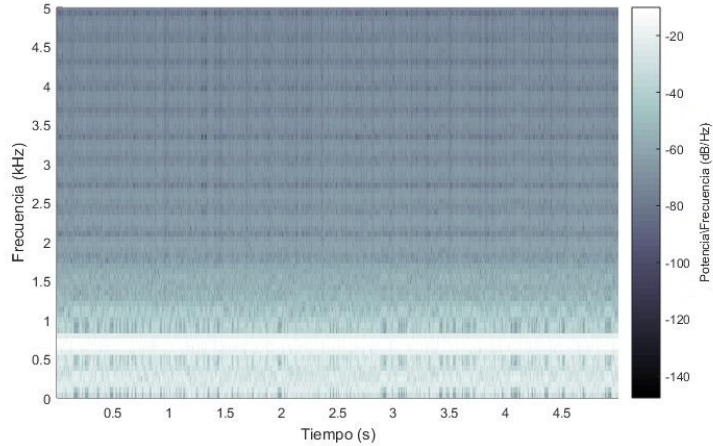


Figura 20: Espectrograma del oscilador de Chua

La Figura 20 se generó con transformadas rápidas de Fourier de tiempo corto. Cada muestra del espectro consta de 400 muestras, lo cual a una frecuencia de muestreo de 44.1KHz corresponde a un tiempo de aproximadamente 1ms por muestra. Se observa que efectivamente existe una frecuencia dominante en todo momento, mientras el contenido espectral por debajo y sobre dicha frecuencia es el que se ve sometido a cambios mientras avanza el tiempo.

8.3. Atractor de Lorenz

El atractor de Lorenz es uno de los sistemas caóticos más estudiados en la literatura ya que este fue de los primeros en ser encontrados. Edward Lorenz propuso sus ecuaciones como un modelo sobre-simplificado de fenómenos de convección atmosférica. Se ha escogido como base el sistema con las mismas constantes que analizó Lorenz en su artículo original [6]. Este sistema es el siguiente:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = rx - y - xz \\ \dot{z} = xy - bz \end{cases} \quad (46)$$

Donde los parámetros del sistema adquieren los siguientes valores:

$$\sigma = 10 \qquad b = 8/3 \qquad r = -28$$

Se procedió a verificar la existencia del atractor caótico encontrando la solución al sistema por medio de métodos numéricos. Al graficar las trayectorias en el espacio de fase se obtuvo la Figura 21, la cual coincide con los ejemplos encontrados en la literatura.

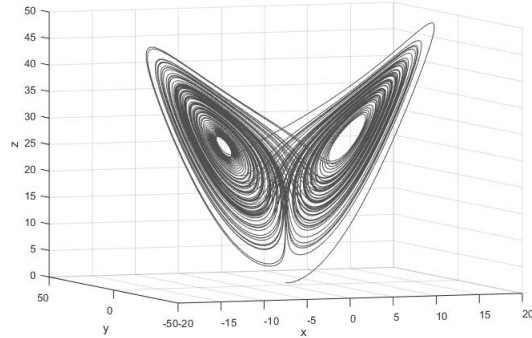


Figura 21: Atractor de Lorenz

Adicionalmente se realizó un cómputo para encontrar experimentalmente un aproximado de los exponentes de Lyapunov del atractor de Lorenz.

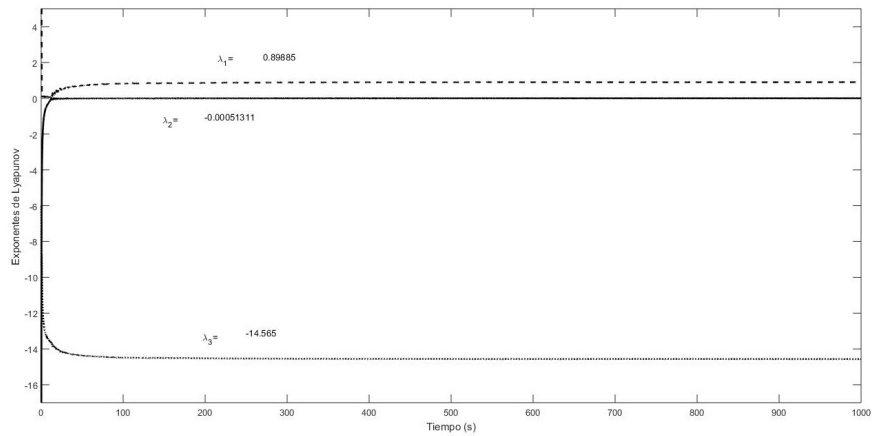


Figura 22: Coeficientes de Lyapunov en el atractor de Lorenz

$$\begin{aligned}
 \lambda_1 &= 0.89885 \\
 \lambda_2 &= -0.00051311 \\
 \lambda_3 &= -14.565
 \end{aligned}
 \tag{47}$$

El código utilizado para generar la figura 22 fue tomado del texto *Dynamical Systems with Applications Using MATLAB* [7] y este se basa en el método descrito en el artículo *Determining Lyapunov exponents from a time series* [23]. Obsérvese que los exponentes de Lyapunov encontrados cumplen con las condiciones para la presencia de un atractor extraño. La cual conlleva un comportamiento oscilatorio y aperiódico en el tiempo. Un exponente de Lyapunov positivo es consistente con la divergencia exponencial de trayectorias

con condiciones iniciales cercanas. Al igual que con el oscilador de Chua, se ha hecho el cómputo de la ecuación 45 para observar este fenómeno.

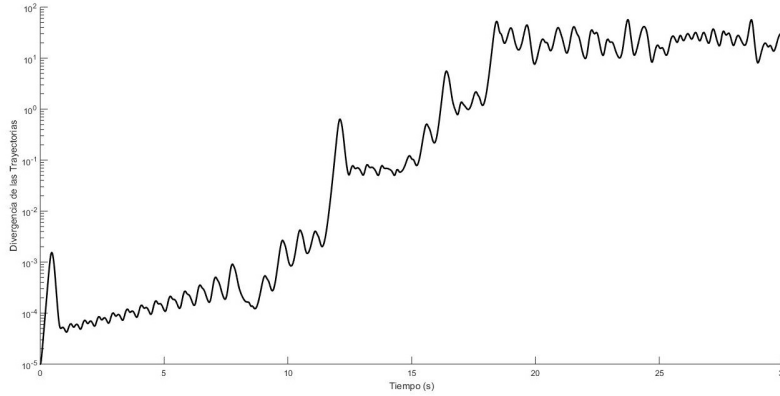


Figura 23: Divergencia de trayectorias con condiciones iniciales cercanas en el atractor de Lorenz

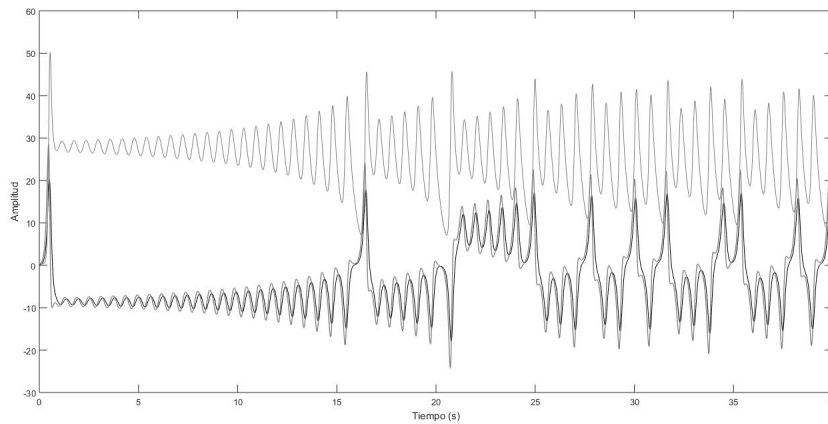


Figura 24: Atractor de Lorenz en el tiempo

El comportamiento oscilatorio se puede observar en la Figura 24. En este caso se encuentra que las oscilaciones de las variables de estado no están centradas en el eje horizontal, comportamiento que si se observó en el oscilador de Chua. Esto es consistente con el hecho que el atractor de Lorenz cuenta con dos puntos fijos al rededor de los cuales orbitan las trayectorias. Los puntos fijos del sistema son los siguientes:

$$\begin{aligned}
 x^* &= \pm \sqrt[2]{b(r-1)} \\
 y^* &= \mp \sqrt[2]{b(r-1)} \\
 z^* &= r-1
 \end{aligned}
 \tag{48}$$

Al linealizar las ecuaciones en torno a cada uno de los puntos es posible determinar que en la localidad de estos, el sistema se comporta como espirales inestables. Esto quiere decir que las órbitas son repelidas por un flujo rotatorio en el campo vectorial del espacio de fase.

Al observar las figuras 24 y 25 es evidente que el contenido espectral dominante se encuentra en la frecuencia $0Hz$. Para poder sintetizar sonido aplicable en la industria de audio se requiere de una frecuencia dominante distinta de cero, ya que las oscilaciones deben estar centradas en cero. Con el fin de verificar la existencia de dicha frecuencia se realizó una transformada rápida de Fourier para la variable de estado z del sistema. Se ha escogido tal variable ya que esta tiene la frecuencia dominante más pronunciada entre las tres posibles. Esto se debe a que la coordenada en z en ambos puntos fijos permanece constante, lo cual implica que las órbitas siempre oscilan respecto a la misma referencia. En el caso de x y y se puede observar como se da un cambio en el nivel en las señales cuando la trayectoria cambia de órbita entre los puntos fijos.

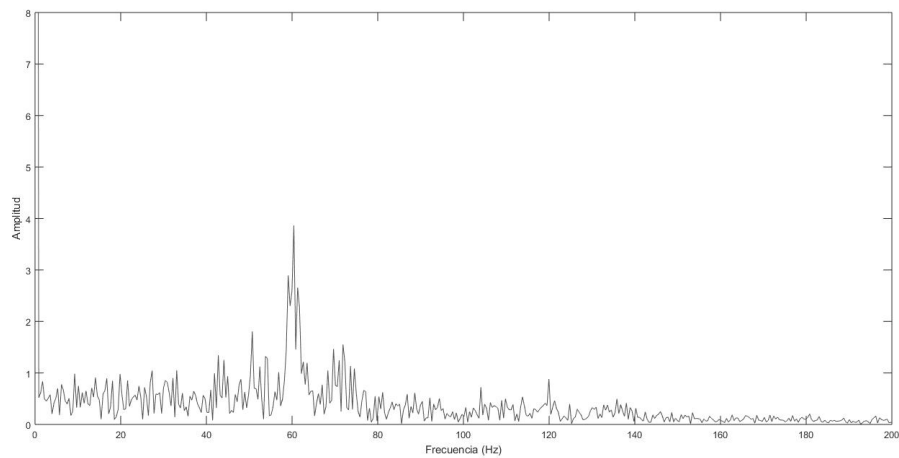


Figura 25: Variable de estado z de atractor de Lorenz en frecuencia

Al igual que con el oscilador de Chua se ha realizado un espectrograma con el fin garantizar la presencia de la frecuencia dominante en todo momento.

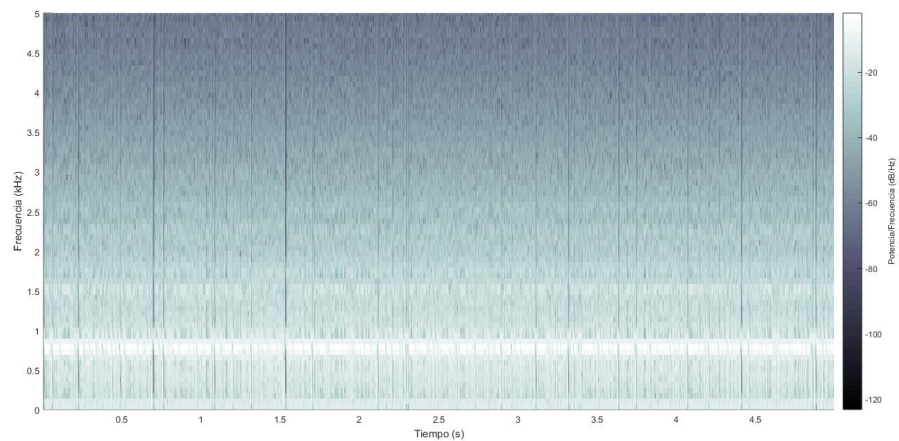


Figura 26: Espectrograma para variable de estado z en atractor de Lorenz

En el espectrograma mostrado en la Figura 26 se puede observar que la frecuencia dominante permanece en el sistema a lo largo del tiempo. Con la finalidad de obtener un espectro más limpio se eliminó el contenido espectral de la señal a cero Hertz, previo a la realización del espectrograma.

Desplazamiento del contenido espectral de los osciladores

Para la creación del instrumento virtual se busca poder manipular de forma predecible la frecuencia dominante de los osciladores utilizados para realizar la síntesis de sonido. De esta forma se puede generar tonos específicos que correspondan a cada nota dentro de la escala musical. Este procedimiento puede resultar complicado para sistemas no lineales ya que cada caso se debe tratar como un particular, ya que no existe una teoría general que permita caracterizar completamente el comportamiento de sistemas no lineales. Como se suele realizar en aplicaciones ingenieriles se ha optado por explorar el caso lineal y luego evidenciar si es posible extender el comportamiento a sistemas no lineales.

9.1. Intuición a partir del oscilador de Van der Pol

Ya se ha mencionado previamente que el oscilador de Van der Pol colapsa a un sistema lineal cuando su único parámetro es igualado a cero. Considerando ese caso, se tiene el siguiente sistema lineal que describe un oscilador armónico simple.

$$\begin{cases} \dot{x} = y \\ \dot{y} = -x \end{cases} \quad (49)$$

Este sistema tiene la siguiente solución general:

$$\begin{cases} \dot{x} = -B_1 \sin t + B_2 \cos t \\ \dot{y} = B_1 \cos t + B_2 \sin t \end{cases} \quad (50)$$

obsérvese que ambas variables de estado presentan la frecuencia fundamental $f = \frac{1}{2\pi}$.

Esta frecuencia puede ser variada al multiplicar el campo vectorial que describe la ecuación por un valor escalar. Esto equivale a multiplicar la ecuación de cada variable de estado por una misma constante, la cual será denominada C .

$$\begin{cases} \dot{x} = Cy \\ \dot{y} = -Cx \end{cases} \quad (51)$$

Este nuevo sistema tiene la siguiente solución general:

$$\begin{cases} \dot{x} = -B_1 \sin Ct + B_2 \cos Ct \\ \dot{y} = B_1 \cos Ct + B_2 \sin Ct \end{cases} \quad (52)$$

Obsérvese que ahora las variables de estado tienen una frecuencia fundamental $f = \frac{C}{2\pi}$. Por lo tanto se puede concluir que en el caso lineal basta multiplicar el campo vectorial del sistema para desplazar la frecuencia dominante. Además, es importante recalcar que existe una relación directamente proporcional entre el factor de escalamiento y la frecuencia dominante en el sistema.

9.2. Desplazamiento del contenido espectral en osciladores no lineales

Habiendo evidenciado un método efectivo para cambiar la frecuencia dominante en un oscilador lineal, se ha procedido a verificar si el método es efectivo para casos no lineales. El caso general se plantea en la ecuación 53.

$$\dot{\mathbf{x}} = Cf(\mathbf{x}) \quad (53)$$

Donde C es un escalar mayor a cero que multiplica el campo vectorial del espacio de fase. Se espera que el resultado sea un aumento en la velocidad de las trayectorias lo cual conllevaría un desplazamiento del contenido espectral del sistema. Tras obtener resultados experimentales se observó que efectivamente el resultado es semejante al observado en el caso lineal. Al observar la Figura 27 se pudo notar un leve decaimiento en la amplitud de la frecuencia dominante del sistema.

Habiendo comprobado que es posible desplazar el contenido espectral de un sistema no lineal, se procedió a encontrar como se relaciona la frecuencia dominante con el multiplicador. El caso óptimo resultaría de encontrar una relación lineal entre el factor C y la frecuencia dominante del sistema no lineal. Para ello se tomó una muestra de datos de la frecuencia dominante contra el multiplicador.

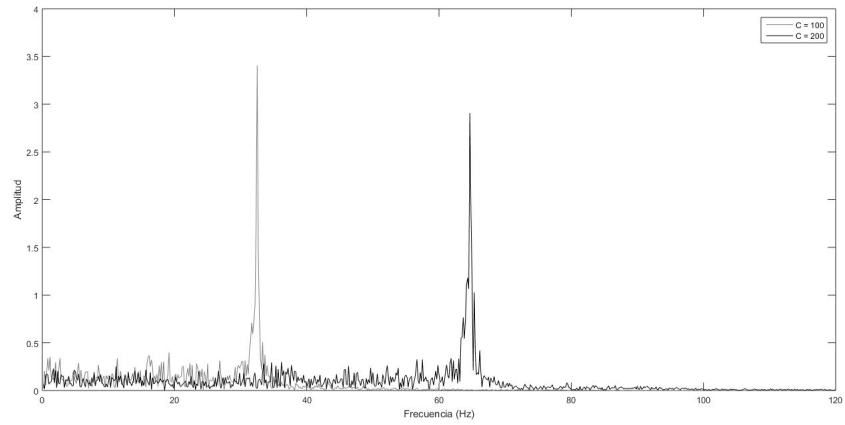


Figura 27: Desplazamiento del Contenido Espectral en el Oscilador de Chua

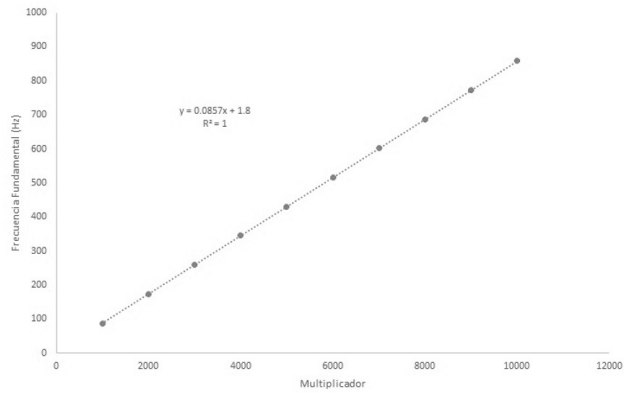


Figura 28: Tendencia en el desplazamiento del contenido espectral en el oscilador de Van der Pol

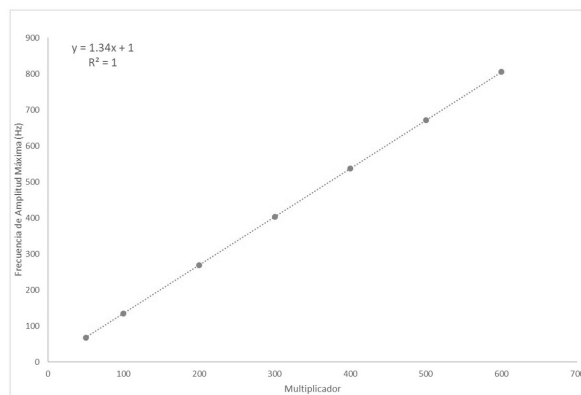


Figura 29: Tendencia en el desplazamiento del contenido espectral en el oscilador de Chua

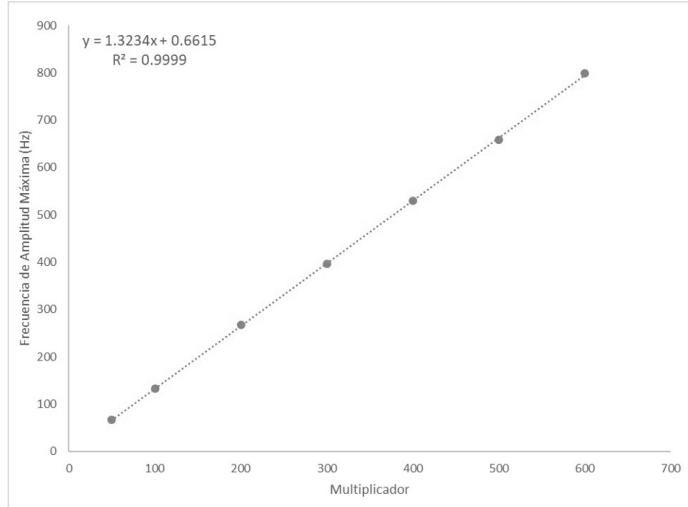


Figura 30: Tendencia en el desplazamiento del contenido espectral en el oscilador Lorenz

Se encontró que el desplazamiento ocurre de forma proporcional al multiplicador. Haciendo una regresión lineal de los datos ha sido posible encontrar las funciones que describen el valor de la frecuencia dominante para un multiplicador específico. Nótese que como la relación tiende a ser perfectamente lineal, las frecuencias pueden ser generadas dentro de un margen reducido de error.

Para el oscilador de van der Pol se tiene la ecuación:

$$c = 8.247014f - 13.537050 \quad (54)$$

Para el oscilador de Chua se tiene la ecuación:

$$c = 0.749853f - 11.409899 \quad (55)$$

Para el atractor de Lorenz se tiene la ecuación:

$$c = 0.741694f + 6.868399 \quad (56)$$

Ya se ha verificado que el método es efectivo para desplazar el contenido espectral de los osciladores, sin embargo, también es necesario evaluar si este tiene algún efecto sobre la geometría de los atractores. Primero se debe recalcar que tanto el sistema dinámico original como el modificado por el factor de escalamiento presentan los mismos puntos de equilibrio. Además, Se puede efectuar el computo de los exponentes de Lyapunov para verificar que aun se presenta el atractor extraño tras escalar el campo vectorial del sistema.

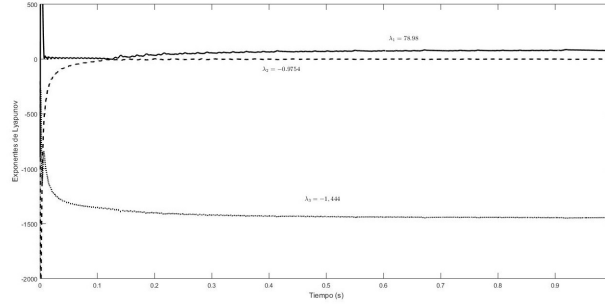


Figura 31: Coeficientes de Lyapunov en el atractor de Lorenz tras escalar el campo vectorial

$$\begin{aligned}
 \lambda_1 &= 78.98 \\
 \lambda_2 &= -0.9754 \\
 \lambda_3 &= -1,444.0
 \end{aligned}
 \tag{57}$$

Obsérvese que nuevamente se tiene un exponente de Lyapunov positivo, uno negativo y otro que tiende a cero. Además la suma de los exponentes de Lyapunov es menor a cero. Por lo tanto la presencia del atractor extraño se conserva tras el escalamiento del campo vectorial del sistema. En este caso es evidente que las magnitudes del exponente positivo y negativo han aumentado. Esto es consistente con el hecho que para este nuevo sistema la divergencia de las trayectorias es más rápida.

Otra forma de evidenciar que el atractor no ha sido modificado es utilizando un mapa de Poincare. Este es un gráfico que muestra la intersección de una trayectoria de un sistema dinámico con un plano ubicado en el espacio de fase. Para las figuras 32 y 33, se ha utilizado el plano $z = 0$. Obsérvese que para diferentes factores de escalamiento, la región de intersección en el plano se mantiene tanto para el oscilador de Chua, como para el atractor de Lorenz.

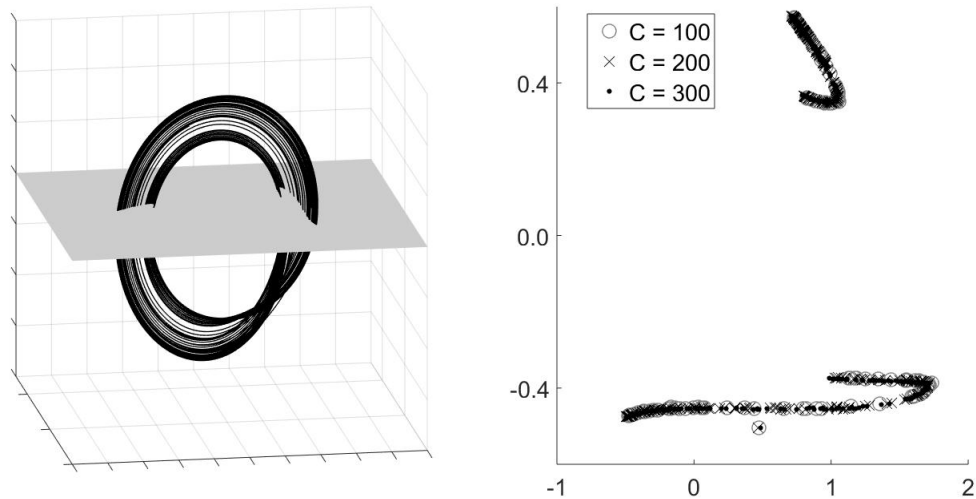


Figura 32: Mapa de Poincaré del Oscilador de Chua Para Distintos Factores de Escalamiento

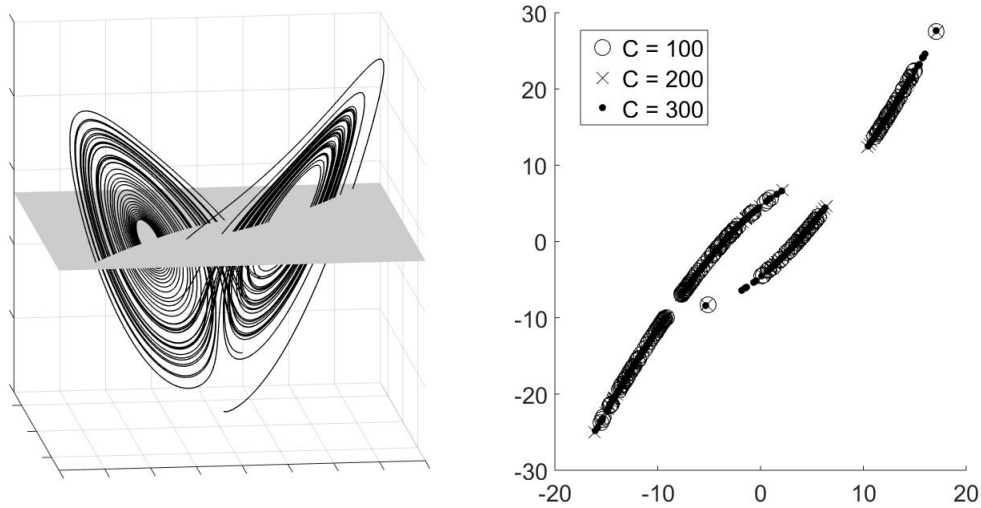


Figura 33: Mapa de Poincaré del oscilador de Chua Para distintos factores de escalamiento

En el caso del oscilador de Van der Pol es más simple identificar como el escalamiento del campo vectorial afecta la dinámica del sistema. Basta con verificar si la órbita a la cual convergen las trayectorias se mantiene inmutada, independientemente del factor de escalamiento. En ausencia de un método analítico para hacer esta demostración se ha optado por encontrar resultados experimentales con distintos factores de escalamiento. En la figura 34 se puede observar como cuatro trayectorias convergen al mismo ciclo límite independientemente de sus condiciones iniciales y factor de escalamiento del campo vectorial.

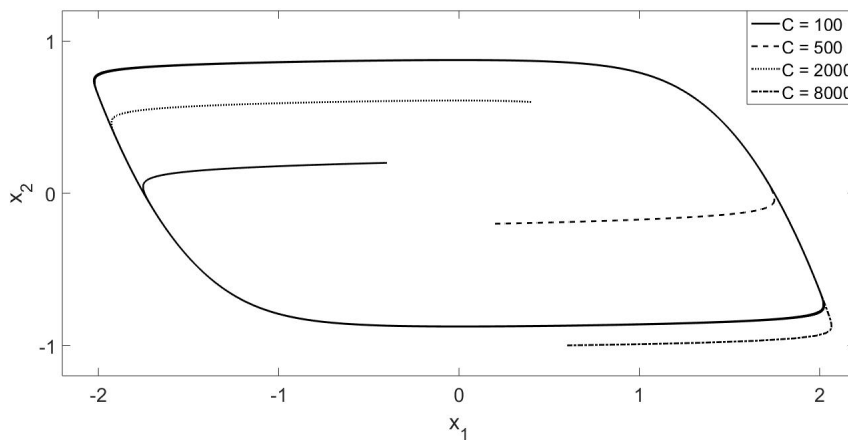


Figura 34: Ciclo límite del oscilador de Van der Pol con distintos factores de escalamiento

Implementación de algoritmos de síntesis de audio

Habiendo seleccionado los sistemas dinámicos a implementar y teniendo un método para generar frecuencias específicas, es posible proceder a la implementación de algoritmos de síntesis de audio. La síntesis se realizará encontrando trayectorias de los sistemas dinámicos utilizando métodos numéricos. Además, es necesario considerar como se acondicionará la señal para que esta esté acotada entre ± 1 .

10.1. Implementación y análisis de métodos de integración numérica

La selección del método numérico a implementar es crucial para el desarrollo del proyecto. Esto se debe a que la calidad del sonido generado depende en gran medida al método utilizado. La convergencia de la solución tiende a tolerar pasos de tiempo más largos mientras aumenta el orden del método de integración numérica. Este fenómeno se observó en la simulación de osciladores con el método de Euler. Este predijo divergencia en las trayectorias utilizando el paso correspondiente a una frecuencia de muestreo de 44.1 KHz , en el caso de osciladores de alta frecuencia. Este fenómeno también se observó utilizando el método de Runge-Kutta de cuarto orden (RK4), sin embargo, se dio a frecuencias mayores que en el método de Euler. Una opción para asegurar la convergencia de las soluciones es reducir el paso del método numérico. Utilizando una frecuencia de muestreo $f_s = 44,100 * 5$, se ha logrado capturar frecuencias de hasta 10 KHz para los tres osciladores seleccionados. Este procedimiento también ayuda a reducir los efectos de aliasing, ya que se cuenta con una mayor frecuencia de Nyquist.

El método numérico dictará que tantos recursos computacionales son utilizados para aproximar las trayectorias. En los métodos de Runge-Kutta se tiende a obtener un error

global menor mientras aumenta el orden del método, sin embargo con el aumento del orden aumenta la cantidad de recursos computacionales requeridos para su implementación. Se ha optado por utilizar el método RK4, mostrado en la ecuación 17, debido a que este ya ha sido utilizado en sintetizadores con sistemas dinámicos en el software de síntesis de audio *SuperCollider*. Los sintetizadores ya implementados en este ambiente de programación son capaces de generar el audio en tiempo real, lo cual es evidencia de la aplicabilidad del método.

Los métodos adaptativos han sido descartados, a pesar que estos pueden reducir considerablemente el costo computacional requerido para encontrar las trayectorias del sistema. Esto se debe a que para la generación del audio se requiere de una frecuencia de muestreo constante. En otras palabras, todas las muestras deben ser equidistantes en el tiempo, lo cual no sucede en los métodos adaptativos. Las muestras equidistantes en el tiempo son necesarias para poder procesar la señal y reproducirla sin distorsión.

10.2. Acondicionamiento de las señales

El acondicionamiento de las señales implica lograr que todos los datos estén acotados entre ± 1 . En el caso contrario la señal se saturará para todos los datos fuera del rango, lo cual crearía distorsión en el sonido generado. Para lograr acotar la señal se necesita que esta esté centrada en cero y normalizar la amplitud de las oscilaciones. Con el fin de eliminar el contenido espectral en cero hertz y además banda-limitar la señal, se ha optado por utilizar un filtro digital Butterworth pasa bandas de segundo orden. El filtro ha sido diseñado utilizando la herramienta *Signal Analysis* de Matlab. Se ha optado por un filtro Butterworth ya que este tipo de filtro tiene una banda de paso plana. Esto evita distorsiones en el contenido espectral dentro de la banda de paso. Las frecuencias de corte se han seleccionado en los límites del rango audible del oído humano. Estas son $f_{cL} = 20Hz$ y $f_{cH} = 20KHz$. El filtro implementado para una frecuencia de muestreo de $44.1KHz$ tiene la siguiente función de transferencia.

$$H(z) = 0.87067001522042442 \frac{1 - z^{-2}}{1 - 0.25369694787402558z^{-1} - 0.74134003044084895z^{-2}} \quad (58)$$

La cual tiene la siguiente respuesta en amplitud.

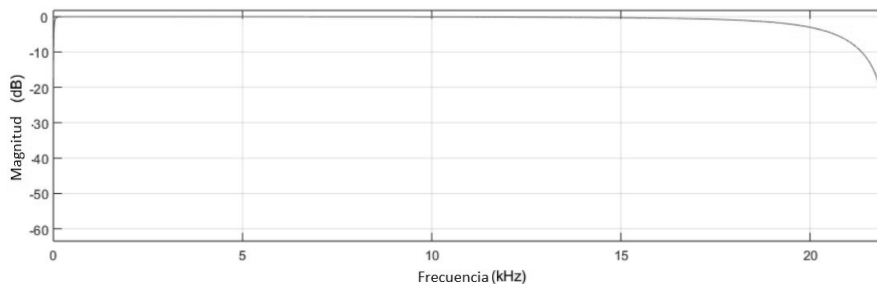


Figura 35: Magnitud de filtro pasa banda

De esta misma función se deriva la siguiente ecuación de diferencias.

$$y[n] = 0.87067001522042442(x[n] - x[n - 2]) + 0.25369694787402558y[n - 1] + 0.74134003044084895y[n - 2] \quad (59)$$

donde $y[n]$ representa la salida actual (dato de la señal filtrada), $x[n]$ representa la entrada actual (dato de la señal sin filtrar) y los índices $[n - i]$ representan datos anteriores.

En tanto a la normalización de la amplitud esta se hizo de forma empírica. Observando la amplitud máxima de oscilación en cada caso, se ha dividido cada dato de la señal por la amplitud máxima correspondiente. Una opción para lograr normalizar la amplitud de la señal de forma automática sería la implementación de un compresor, sin embargo, esto implica una transformación no lineal. Este tipo de transformación conllevaría distorsión en la señal original. Con los dos ajustes mencionados se puede garantizar que la señal estará acotada dentro del rango requerido.

10.3. Síntesis de audio en tiempo real

Como se ya se ha mencionado existen diversas formas de implementar osciladores digitales. En este caso se ha decidido utilizar la representación matemática de la onda por medio de soluciones numéricas de los sistemas dinámicos. El método propuesto requiere que las muestras de audio se generen de forma dinámica en tiempo real. Esto implica un costo computacional elevado para la implementación de un instrumento polifónico, ya que se requiere de resolver un sistema dinámico distinto para la generación cada nota que sea reproducida de forma simultánea. Este problema podría ser resuelto con el uso de tablas pre calculadas que contengan las muestras. No se ha optado por esta opción debido a que se considera que la generación dinámica de muestras tiene ventajas que justifican el costo computacional de su implementación.

La principal ventaja de la generación de muestras en tiempo real es que permite explotar al máximo las propiedades de los sistemas dinámicos no lineales. Un método de síntesis estático como el almacenamiento de muestras pre calculadas limita la generación del sonido a una forma de onda única, mientras que la generación dinámica de muestras permite variar parámetros del sistema durante la reproducción. Esto brinda al usuario una gran versatilidad en tanto a capacidades de modulación del sonido generado. Además, es posible utilizar condiciones iniciales distintas en cada nueva nota a reproducir. Lo cual, según las características de los sistemas caóticos, crea una trayectoria distinta manteniendo las propiedades intrínsecas del atractor.

Desarrollo de primer prototipo en RackAFX

La revisión de literatura y la etapa de simulación permitieron avances considerables hacia el cumplimiento del objetivo de este trabajo.

- Se han seleccionado los sistemas dinámicos a utilizar para la generación de sonido.
- Se ha seleccionado el método numérico a utilizar para encontrar las trayectorias de los sistemas dinámicos.
- Se cuenta con un método para generar frecuencias específicas en la trayectoria de los sistemas.
- Se cuenta con las herramientas necesarias para realizar el acondicionamiento de las señales generadas.

El siguiente paso a dar es la creación de un *plug-in* con el fin de verificar la viabilidad de la síntesis en tiempo real que además permita explorar a mayor detalle la respuesta de los sistemas seleccionados. Para la creación de este se utilizó el software *RackAFX*.

RackAFX permite la creación de *plug-in* tanto de efectos digitales de audio como de síntesis de audio. Para crear un proyecto que utilice la API de VST de instrumentos digitales es necesario seleccionar el *check-box* con la opción "*Output only synthesizer plug-in*" mostrado en la ventana de dialogo del proyecto nuevo. Esto indica que el *plug-in* no procesará señales de entrada sino únicamente generará una salida de audio. Una vez creado el proyecto se ejecuta el programa *Visual Studio* de forma automática, en el cual se puede editar las funciones de la interfaz del *plug-in*. Para explorar la implementación de un oscilador simple en el ambiente de *RackAFX* se recomienda realizar el ejemplo desarrollado en el capítulo 8 del texto *Designing Audio Effect Plug-Ins in C++* [14].

Dentro de las funciones generadas en el archivo principal del proyecto se encuentra la función "*processAudioFrame*". Esta es ejecutada para cargar una muestra de sonido a la cola de salida de la tarjeta de audio. Esto se hace a la frecuencia de muestreo establecida para el *plug-in*, en este caso 44.1 *KHz*. Se debe generar una muestra de audio cada vez que se ejecuta esta función. Para ello se agregó funciones adicionales a la interfaz del *plug-in*, las cuales están completamente dedicadas a la obtención de la muestra. Se creó una función para el calculo de cada variable de estado, una función para el calculo de la muestra actual utilizando el método RK4 y una función de filtrado para el acondicionamiento de la señal. Todas las funciones mencionadas deben ser ejecutadas una vez por cada muestra a obtener. Para resolver el problema de la divergencia del método para frecuencias altas se redujo el paso del método numérico, lo cual implica un sobre-muestreo. La frecuencia de muestreo del paso numérico se escogió como un múltiplo de la frecuencia estándar. Se observó experimentalmente que al aumentar 5 veces la velocidad de muestreo de 44.1 *KHz* es posible asegurar la convergencia del método para los tres sistemas seleccionados, dentro del rango de frecuencias que se requiere generar. Entonces, el método RK4 implementado genera 5 muestras cada vez que se ejecuta la función *processAudioFrame*, de las cuales se considera únicamente la ultima muestra generada al momento de reproducir el audio.

Para cada sistema dinámico se creo un proyecto distinto donde la única diferencia es el modelo para las variables de estado. En cada caso el *plug-in* cuenta con dos controles: Uno que permite variar el factor la frecuencia del oscilador variando el factor de escalamiento del campo vectorial, y otro que permite variar un parámetro del sistema dinámico con el fin de poder explorar sus efectos sobre la dinámica del sistema. El parámetro que se ha seleccionado en cada sistema se muestra en el Cuadro 1. Con los controles mencionados la funcionalidad de los *plug-in* se reduce a poder sintetizar sonido en tiempo real con un rango de frecuencias continuo. Finalmente se agregó la opción de poder iniciar o detener la síntesis de sonido por medio de un botón de encendido y apagado.

Sistema dinámico	Parámetro	Límite inferior	Límite superior
Van der Pol	ε	0.5	6.0
Chua	α	26.5	30.1
Lorenz	r	20.0	80.0

Cuadro 1: Parámetros modulables en primer prototipo de *plug-in*

11.1. Resultados obtenidos de primer *plug-in*

La implementación de los *plug-ins* ha permitido generar muestras de audio las cuales han sido analizadas utilizando *Matlab*, con el fin de verificar los efectos de la variación de los controles sobre los sistema dinámicos. En la Figura 36 se puede observar como varía la frecuencia en el oscilador de Chua al aumentar el factor de escalamiento del campo vectorial. También se ha verificado que efectivamente se logra obtener la frecuencia predicha con las ecuaciones 54, 55 y 56, según el sistema correspondiente. Nótese que el resultado anterior ya había sido obtenido en de la etapa de simulación, sin embargo en este caso se verificó con una muestra de audio generada en tiempo real, mientras se variaban los controles del *plug-in*. Se ha omitido la presentación de espectrogramas en cada sistema mientras se varía

el factor de escalamiento ya que la respuesta obtenida no brinda información adicional.

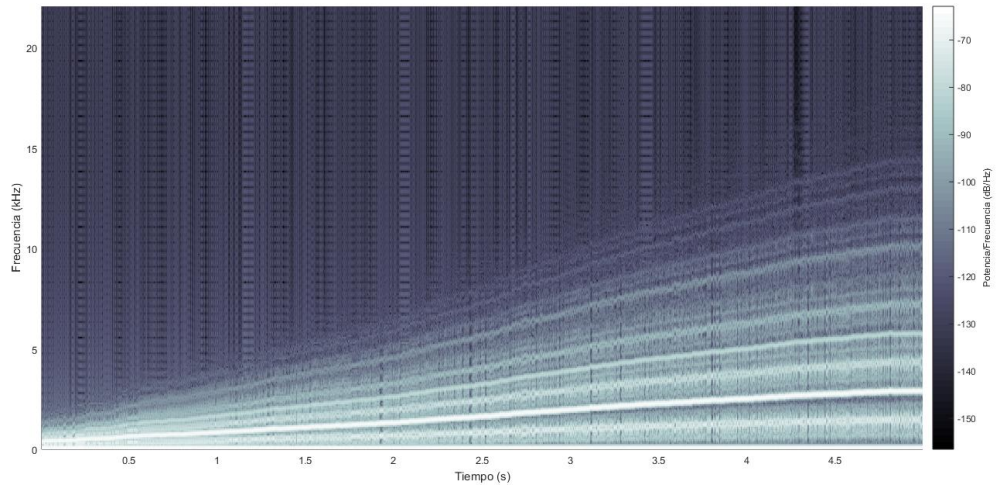


Figura 36: Variación de frecuencia en oscilador de Chua al aumentar factor de escalamiento

En lo que se refiere a la variación de parámetros del sistema, se ha logrado identificar comportamientos importantes en cada caso. Al analizar la Figura 37, se puede observar que la variación del parámetro α no tiene efecto sobre la frecuencia dominante del sistema, sin embargo la dinámica general cambia. Del tiempo de 0 a aproximadamente 2 segundos, se puede observar una ventana en la cual el comportamiento del sistema se asemeja al de una onda periódica, caso que se repite en aproximadamente $t = 4s$. En el resto del tiempo se puede observar como varía el contenido espectral alrededor de la frecuencia dominante. En el caso del atractor de Lorenz se obtuvo el espectrograma mostrado en la Figura 38 al variar el parámetro r durante la generación de la muestra de audio. Se puede observar que existe una fuerte dependencia entre la frecuencia de oscilación y el valor del parámetro r . En tanto al oscilador de Van der Pol, ya se ha descrito los efectos que tiene la variación del parámetro ε sobre la dinámica y frecuencia del sistema en la sección 1 del capítulo 8.

Se ha observado que la frecuencia de oscilación del atractor de Lorenz y el oscilador de Van der Pol depende del parámetro evaluado en cada uno. Es importante saber qué tipo de relación existe entre la variación de estos parámetros y la frecuencia dominante, ya que de ello depende la libertad de modulación que estos presentan en la generación de audio. El caso óptimo sería una relación lineal, sin embargo se ha observado que este no es el caso para ninguno de los dos sistemas. Esto se evidencia en las figuras 39 y 40. A pesar de haber obtenido una respuesta aproximadamente lineal no se puede asegurar que se obtendrá una frecuencia completamente predecible para cada valor del parámetro.

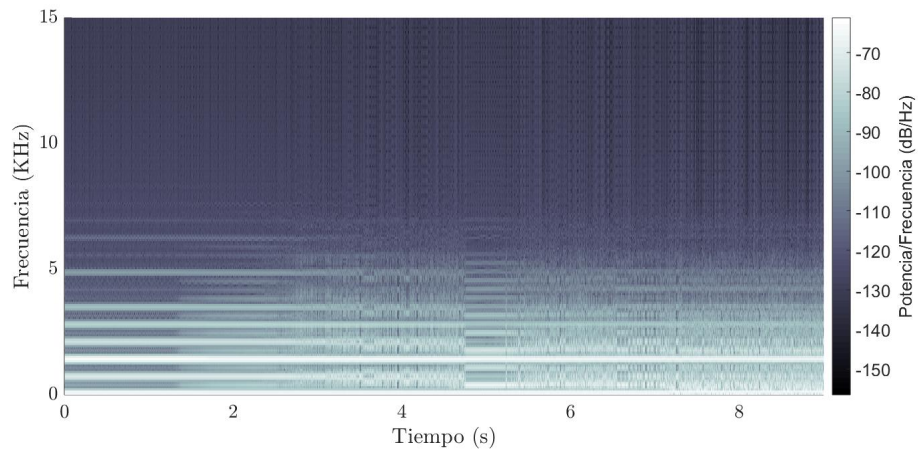


Figura 37: Espectro del oscilador de Chua al variar el parámetro α

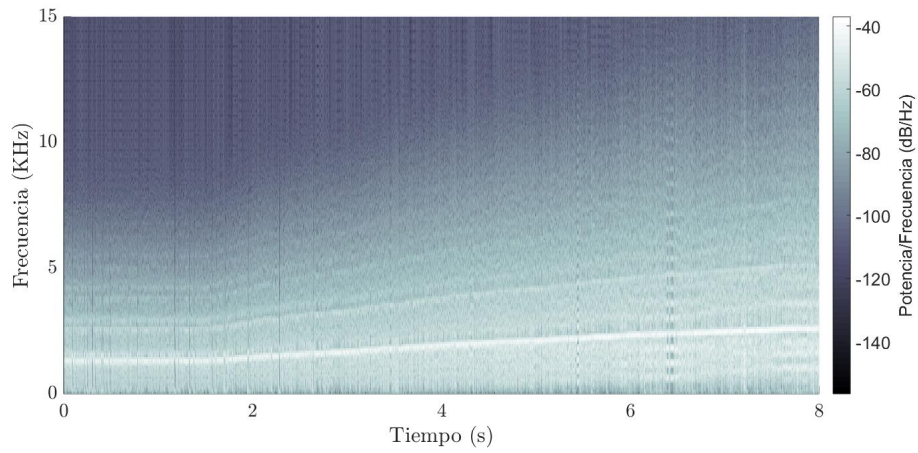


Figura 38: Espectro del atractor de Lorenz al variar el parámetro r

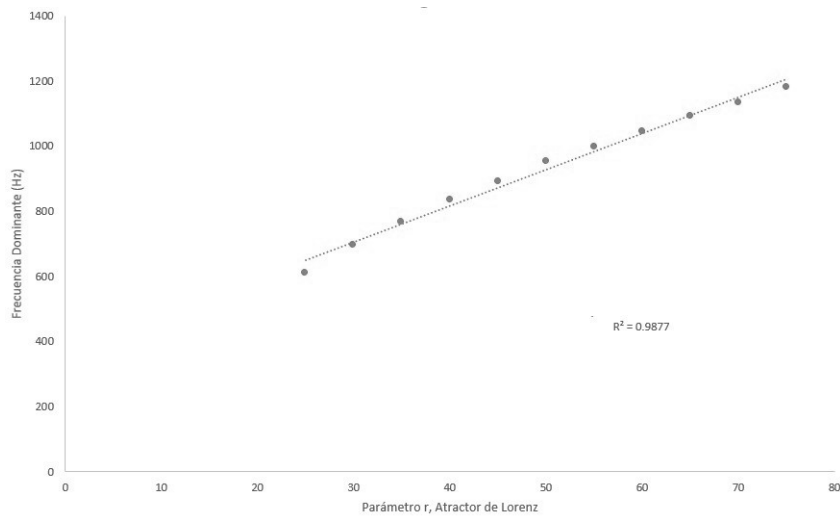


Figura 39: Frecuencia dominante en atractor de Lorenz contra variación de parámetro r

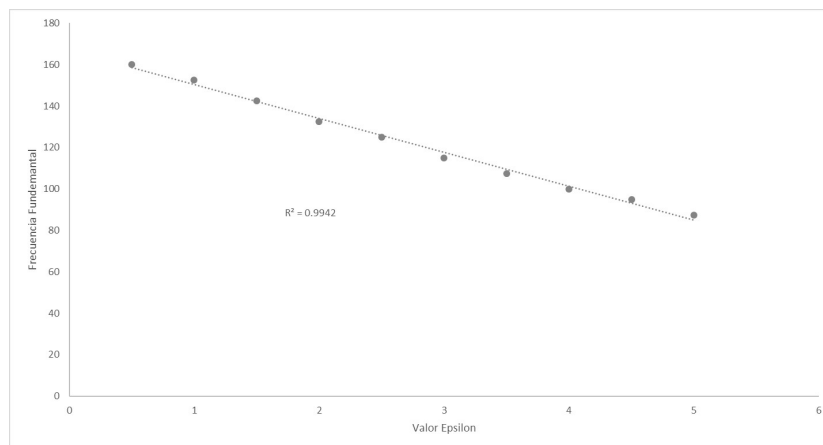


Figura 40: Frecuencia fundamental en oscilador de Van der Pol contra variación de parámetro ε

Desarrollo de prototipo en RackAFX con controlador MIDI

El primer prototipo del *plug-in* VST permitió verificar la viabilidad de realizar síntesis de sonido en tiempo real con los sistemas dinámicos propuestos y comprobó la efectividad del método propuesto para la generación de frecuencias específicas. La siguiente etapa de implementación consistió permitir que el *plug-in* pudiera ser manipulado por un controlador MIDI. Se plantea modular la amplitud de las señales generadas por medio de señales envolventes y así garantizar una mayor calidad de sonido. Finalmente, se buscará que el *plug-in* pueda reproducir varias notas de forma simultánea. Para el desarrollo de este nuevo prototipo se ha tomado como base el proyecto desarrollado en el capítulo 9 del texto *Designing Audio Effect Plug-Ins in C++* [14], cuyo código original puede ser encontrado en la página web oficial de RackAfx [17].

12.1. Implementación de señales envolventes

La funcionalidad y partes de las señales envolventes son descritas en la sección 6.1.2 de este trabajo. Para su implementación en el proyecto de RackAFX se utiliza la clase ya creada en el proyecto base, *EnvelopeGenerator*. Esta permite varias instancias de objetos encargados de generar señales envolventes. Se requiere de una señal envolvente para cada nota en reproducción. En la Figura 41 se muestra la lógica tras los eventos que accionan la transición a diferentes partes de la señal envolvente, en forma de una máquina de estados finitos. Los eventos denominados "*Note On*" y "*Note Off*" son mensajes del canal MIDI y el valor denominado "*.output*" corresponde a la amplitud instantánea de la señal envolvente. En conclusión la generación de la señal envolvente se realiza con una serie de condicionales que determinan en que estado se encuentra actualmente la señal. Con el fin de optimizar la generación de la señal exponencial, correspondiente a los estados de *attack*, *Decay* y *Release*, esta se implementa en forma discreta como una ecuación de diferencias.

$$y[n] = x[n] + by[n - 1] \quad (60)$$

donde el coeficiente b brinda la tasa de crecimiento o decaimiento del segmento de la señal.

$$b = e^{\frac{-\ln(\frac{1+TCO}{TCO})}{S_p}} \quad (61)$$

La constante TCO es igual a la tasa de decaimiento en tiempo continuo y la constante S_p es el periodo de muestreo.

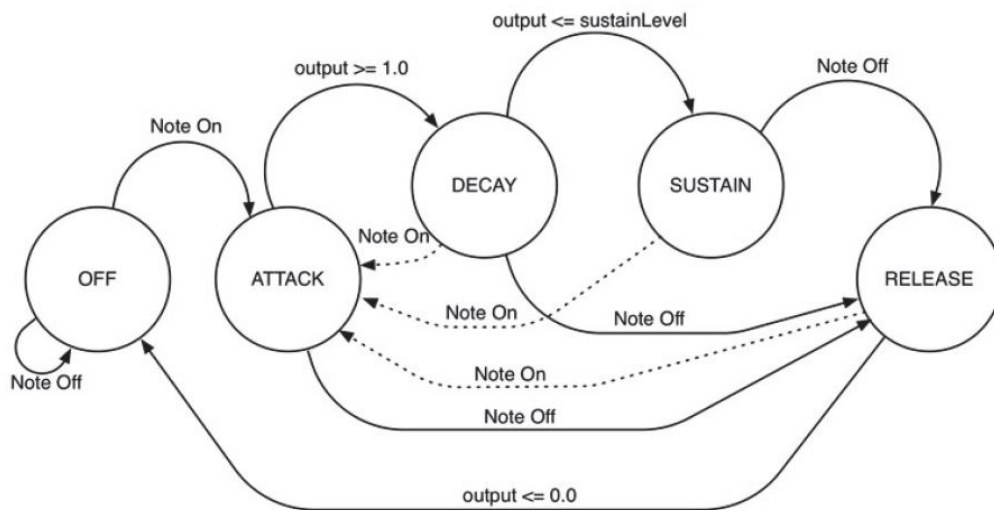


Figura 41: Cambio de estado en una señal envolvente

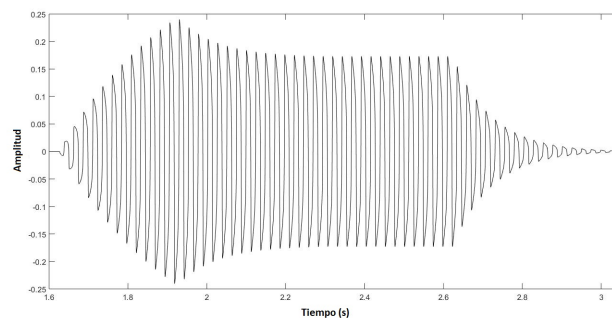


Figura 42: Señal envolvente aplicada al oscilador de Van der Pol

En la Figura 42 se puede observar como la amplitud de la señal generada por el oscilador de Van der Pol es modulada por una la señal envolvente. En este caso se utilizó un tiempo de ataque de 100 ms , un sostenimiento del 77% y un decaimiento inicial y final de 1000 ms por cada uno.

12.2. Implementación de instrumento polifónico

El *plug-in* desarrollado en este proyecto cuenta con diversos niveles de abstracción. Cada voz del instrumento es capaz de generar una nota con su señal envolvente respectiva. Para poder obtener una voz se requiere de una instancia del oscilador que generará el sonido y una instancia del generador de envolventes para modular el sonido generado. Con el fin de acoplar las instancias requeridas en la generación de cada nota, se implementa la clase *Voice*.

Cada nota accionada genera una instancia de la clase *Voice*. Se debe considerar la cantidad de voces que se podrán ejecutar de forma simultánea y como se manejará los eventos donde se accionan más notas de las aceptadas. La cantidad máxima de notas simultáneas se define explícitamente en la API de RackAFX con el parámetro "*MAX_VOICES*". Según la cantidad definida se crea un arreglo de voces inicialmente inactivas. Al activar una voz se verifica si hay voces disponibles en el arreglo. De ser este el caso, se activa la voz libre y se modifica el parámetro de la frecuencia de oscilación. La voz se mantiene activa hasta que se envía el mensaje de *note off* y se finaliza el decaimiento correspondiente a la sección de *release* de la señal envolvente. Si al activar una nota no se encuentra una voz libre, se sustituye una de las voces activas, encendiendo la bandera "*bStealNote*". La nota que activa la bandera toma el lugar de la nota en ejecución que lleva más tiempo activa.

12.3. Recepción e interpretación de mensajes MIDI

Un controlador MIDI envía mensajes en forma serial al ordenador encargado de realizar la síntesis de audio. En este proyecto se implementa únicamente a tres mensajes del protocolo MIDI: *note on*, *note off* y *pitch-bend*. Si el mensaje recibido no corresponde a alguno de los mencionados, es ignorado.

12.3.1. Inicialización de nota

El mensaje del canal MIDI encargado de inicializar una nota es *note On*. El mensaje es enviado por el controlador MIDI en el momento que se presiona una tecla. Este es recibido y decodificado en la interfaz de RackAFX con la función *midiNoteOn*. Dentro del mensaje se encuentra el número del canal MIDI del cual este provino, el número de la tecla que fue presionada y un valor numérico que indica con que velocidad se presionó esta tecla. Para obtener la frecuencia a la cual se debe inicializar la voz, se hace referencia a una tabla indexada según los números de tecla donde contiene la frecuencia correspondiente a cada una. La velocidad con la que se presionó la tecla se carga a la voz inicializada, indicando la amplitud máxima que esta debe alcanzar. En otras palabras a mayor velocidad la nota se ejecutará con mayor intensidad. Dentro de esta función se realiza la revisión de voces disponibles y de ser necesario el robo de una de las voces.

12.3.2. Finalización de nota

La finalización de una nota es activada por el mensaje MIDI *Note off*. Este mensaje es enviado al momento de liberar una tecla previamente presionada en el controlador MIDI. El mensaje es recibido y decodificado en la función *midiNoteOff* en la interfaz de RackAFX. Este cuenta con el número del canal MIDI del cual provino, el número de tecla que fue liberada, la velocidad con la cual se liberó la tecla y una bandera lógica. La bandera contenida en el mensaje, es una funcionalidad del protocolo MIDI que sirve para indicar que se debe finalizar la reproducción de todas las notas. Para este proyecto no se ha implementado funcionalidad a la velocidad con la cual se libera la nota, por lo tanto este dato es ignorado. Para desactivar la voz se identifica cual es la voz activa con el número de tecla correspondiente al mensaje para poder desactivarla.

12.3.3. Modificación de la frecuencia por controlador MIDI

El tercer mensaje del canal MIDI que se implementó es *pitch bend*. Este es enviado por el controlador MIDI al mover la rueda o palanca de portamento. El mensaje es recibido y decodificado en la interfaz de RackAFX por la función *midiPitchBend*. Los datos que el mensaje contiene son el número del canal MIDI del cual provino el mensaje y un valor correspondiente a que tanto se ha movido el dispositivo respecto a su posición estándar. El valor recibido es normalizado a un rango entre ± 1 , para ser enviado a todas las voces activas. Si el valor enviado es negativo se desplaza la frecuencia de oscilación de todas las voces a frecuencias más bajas y si es positivo se desplaza a frecuencias más altas. El valor máximo de modificación de la frecuencia está dado en cantidad de octavas, valor que puede ser modificado por software. Las octavas de una nota corresponden a los armónicos y subarmónicos de esta. Para determinar la frecuencia actual de una nota considerando las posibles modificaciones se ejecuta la siguiente operación en código:

$$m_{dFo} = m_{dOscFo} * PitMultiplier \quad (62)$$

donde la variable m_{dFo} es la frecuencia de oscilación actual de la nota, m_{dOscFo} es la frecuencia de oscilación correspondiente a la tecla MIDI y $PitMultiplier$ es un modificador que depende del valor actual de la rueda de portamento y la octava a la cual se encuentra fijo el instrumento digital.

12.4. Resultado final

El ejemplo en el cual se basó el *plug-in* tiene capacidades de modulación adicionales a las mencionadas, sin embargo estas se han removido ya que no son necesarias para el cumplimiento de los objetivos del presente trabajo. Se procedió a agregar el código de los métodos de generación de audio utilizados en el primer prototipo del *plug-in* al la clase "*QBLimitedOscillator*" con el fin de cumplir con el paradigma de abstracción ya construido en el proyecto. Finalmente se creó una interfaz gráfica de usuario con el editor provisto en RackAFX. Como se observa en la Figura 43, se cuenta con un control deslizante para

Control	Límite inferior	Límite superior
Chua α	26.2	31.5
Lorenz r	27.8	28.2
Van der Pol ε	1.5	5
Attack	0 <i>ms</i>	5000 <i>ms</i>
Decay	0 <i>ms</i>	10000 <i>ms</i>
Sustain	0	1
Release	0 <i>ms</i>	10000 <i>ms</i>
Volume	-96 <i>dB</i>	24 <i>dB</i>
Octave	-4	4
PB Range	1	12

Cuadro 2: Rango para controles de plug-in VST

seleccionar el sistema dinámico con el cual se generará el audio. Se cuenta con un efecto de *delay* incrustado al *plug-in* a modo de complemento a la función principal de este. También se puede seleccionar el modo de *legato*. Este modo brinda la posibilidad de suprimir la sección de ataque en las señales envolventes de notas activadas consecutivamente. Este modo de funcionamiento tiene el fin de replicar la dinámica de instrumentos de cuerdas tocados con arco, donde se puede cambiar de nota mientras esta es sostenida por el roce del arco con las cuerdas. Los valores de los otros controles implementados se muestran en la tabla 2. El control denominado "*octave*" se utiliza para desplazar una o más octavas todas las notas del instrumento, ya sea a tonos más agudos o más graves.

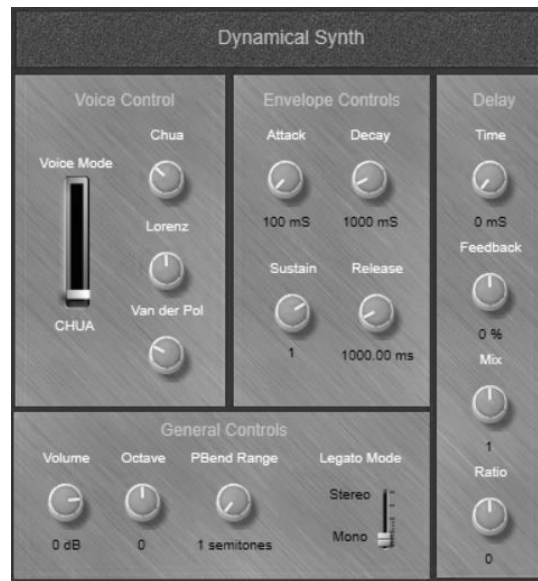


Figura 43: Interfaz gráfica de usuario de *plug-in* VST implementado

Los controles con el nombre de cada sistema dinámico corresponden a los mismos parámetros que eran modificados en la primera versión del *plug-in*. Nótese que en este caso el rango dentro del cual se varía el parámetro correspondiente al atractor de Lorenz se ha reducido significativamente. Esto se debe al efecto que se observó este parámetro tiene sobre

la frecuencia de oscilación del sistema dinámico. Al reducir el rango de variación se limita la variación de la frecuencia a ser imperceptible, sin embargo por la naturaleza caótica del sistema la dinámica de este cambia con variaciones pequeñas del parámetro. Para el oscilador de Van der Pol el rango de variación se redujó pero no de forma drástica. En este caso se buscó encontrar una relación entre el factor de escalamiento del campo vectorial y la variación del parámetro ε . El procedimiento realizado consistió en plantear diez versiones distintas del sistema dinámico del oscilador de Van der Pol, donde la única variante es el valor del parámetro ε . Se encontró la recta que modela la frecuencia de oscilación contra el factor de escalamiento en cada caso. Luego se realizó una regresión polinomial de segundo orden para la pendiente de la recta contra el valor de ε .

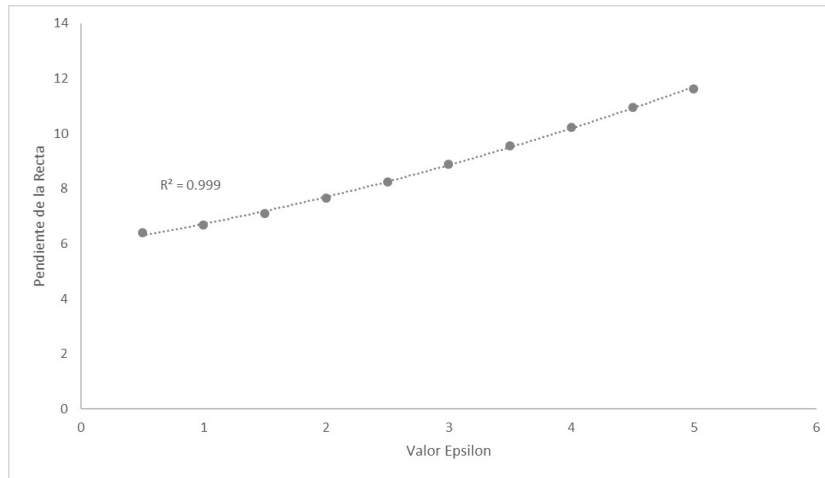


Figura 44: Variación de la pendiente de la recta contra parámetro ε

Implementando el modelo encontrado se tiene la siguiente forma para calcular el factor de escalamiento del campo vectorial, según el valor de ε y la frecuencia requerida.

$$c = (0.0889\varepsilon^2 + 0.712\varepsilon + 0.59185)f - 26.43222 \quad (63)$$

Se ha observado que el modelo no es perfecto ya que existe variación de frecuencia al variar ε , aunque significativamente menor que la observada previa a la corrección. Obsérvese que la regresión polinomial no se ajusta perfectamente a los datos, lo cual se puede corroborar observando el coeficiente de correlación R^2 mostrado en la Figura 44. Además solo se ha tomado en consideración las pendientes de las rectas para la obtención del modelo y se ha ignorado la intercepción en el eje vertical de cada recta. Es necesario evaluar si la variación de frecuencia observada es lo suficientemente significativa como para experimentar disonancia en las notas generadas.

El *plug-in* creado fue evaluado utilizando un controlador MIDI marca *Korg*. Con este se verificó el funcionamiento correcto del *plug-in* en tanto a la recepción e interpretación de mensajes MIDI. También permitió experimentar con la cantidad máxima de notas que se pueden reproducir de forma simultánea. Se encontró que al producir más de 5 notas simultáneas el audio generado empieza a presentar reproducción interrumpida del sonido, producto de la alta demanda de recursos computacionales. Este resultado se obtuvo para un equipo de cómputo con un procesador intel *core i5* de sexta generación y 8 GB de memoria

RAM. La cantidad de notas simultaneas no deberían ser un impedimento para el renderizado de audio por medio del *plug-in* ya que en este caso no es necesario reproducir el audio en tiempo real.

Para evaluar la exactitud en la frecuencia de las notas generadas se crearon archivos de audio de cada sistema utilizando el *plug-in*. Cada archivo contiene una escala de notas musicales, las cuales fueron analizadas en frecuencia utilizando la transformada rápida de Fourier. En la Cuadro 3 se puede observar las frecuencias de mayor amplitud obtenidas en cada uno de los sistemas en comparación con la frecuencia esperada. Nótese que las frecuencias obtenidas son cercanas a las esperadas con un error promedio menor al 1% en todos los casos. El oscilador para el cual se presenta un mayor error promedio es el oscilador de Lorenz. Se cree que esto se debe a que este presenta el contenido espectral más diverso, como se puede observar en la Figura 45.

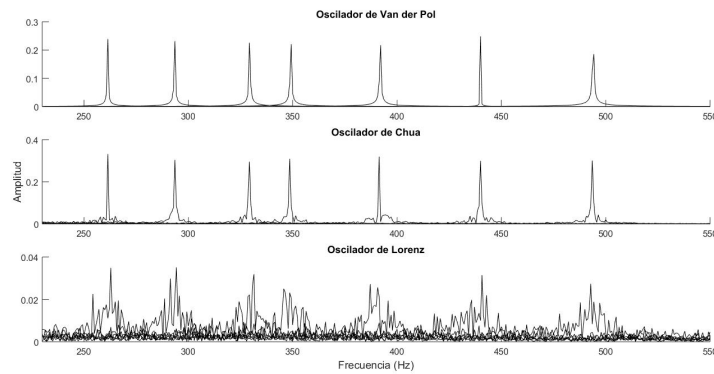


Figura 45: Transformada rápida de Fourier para escala musical

Nota	Frecuencia (Hz)	VDP (Hz)	E %	Chua (Hz)	E %	Lorenz (Hz)	E %
C4	261.63	261.43	0.08	261.43	0.08	262.86	0.47
D4	293.66	293.57	0.03	293.57	0.03	294.29	0.21
E4	329.63	329.29	0.10	329.29	0.10	331.43	0.55
F4	349.23	349.29	0.02	348.57	0.19	345.71	1.01
G4	392.00	392.14	0.04	391.43	0.15	387.14	1.24
A4	440.00	440.00	0.00	440.00	0.00	440.71	0.16
B4	493.88	494.29	0.08	493.57	0.06	492.86	0.21

Cuadro 3: Frecuencias de potencia máxima experimental para una escala musical

En el texto *Psychoacoustics: Facts and Models* [4] se discute que para frecuencias menores a 500 Hz, el menor cambio perceptible por un ser humano es de aproximadamente 1 Hz. Para frecuencias superiores se establece que el menor cambio perceptible aumenta en proporción con la frecuencia del tono y es aproximadamente 0.002 veces la frecuencia de este. Con base en ello es posible predecir que la diferencia en frecuencia para algunas notas generadas respecto al la frecuencia esperada, puede ser perceptible para individuos capaces de discernir notas musicales al escucharlas.

Uso de *plug-in* en Software DAW

El propósito principal de implementar el instrumento virtual como un *plug-in* VST, es que este pueda ser utilizado en software comercial de edición de audio. Al implementar un *plug-in* en RackFAX, se cuenta con compatibilidad directa en el sistema operativo Windows. Esto implica que basta con compilar el proyecto para generar la biblioteca de enlace dinámico del *plug-in*, la cual puede ser utilizada en cualquier software DAW en Windows que soporte el formato VST.

Para el desarrollo de este proyecto se ha utilizado la versión de prueba del software *Reaper*. Es importante considerar el compilador a utilizar al momento de generar el archivo DLL. En el caso de *Reaper*, este es un software que trabaja con una palabra de 64 bits, por lo cual es necesario seleccionar la plataforma de 64 bits al momento de recompilar la solución del proyecto en Visual Studio. Una vez compilado el proyecto se encontrará el archivo DLL en la carpeta del proyecto con el nombre "nombre_plugin.dll". Al copiar este archivo y colocarlo en la carpeta de *plug-ins* del software DAW, este será reconocido como un *plug-in* VST3. La utilización del *plug-in* en el software DAW brinda las siguientes funcionalidades:

- Monitoreo en tiempo real de la generación de sonido utilizando un controlador MIDI.
- Grabar la salida de audio generada por el *plug-in*.
- Reproducción en serie con efectos digitales creados por otros fabricantes.
- Generación de sonido y modulación en tiempo real teniendo como entrada una pista MIDI.
- Renderizado de audio a partir de una pista MIDI.

El prototipo final concluyó en la implementación un instrumento virtual en forma de un *plug-in* VST, que utiliza los modelos del oscilador de chua, el oscilador de Van der Pol y el atractor de Lorenz para realizar síntesis de audio. Con este resultado se determina que se ha alcanzado el objetivo principal de este trabajo que es la creación e implementación de algoritmos de síntesis de audio basados en sistemas dinámicos no lineales. El uso práctico en la industria de audio de los algoritmos creados se deriva de la funcionalidad presente en el *plug-in*. Esta incluye el uso de un controlador MIDI, La compatibilidad con diversas estaciones de trabajo de audio virtual y la capacidad de modulación del sonido generado por medio de señales envolventes y variación de parámetros de los sistemas.

Se considera que el resultado más importante, producto de la experimentación, es el método propuesto para el desplazamiento del contenido espectral. Se ha encontrado que este brinda la capacidad de sintetizar tonos específico a partir de la dinámica de un sistema no lineal. Esto implica que la utilización del método puede ser extendida más allá de los sistemas dinámicos propuestos, brindando un punto de partida para explorar la implementación de nuevos sistemas dinámicos en sintetizadores de audio.

El punto en el que ha finalizado el proyecto brinda una gran gama de posibilidades para la experimentación con el sonido generado. Por cada sistema dinámico implementado se cuenta con varias variables de estado. En este caso la salida de audio ha sido una única variable de estado de los sistemas. Se puede explorar implementando operaciones no lineales a las muestras generadas con el fin de buscar mayor diversidad en los sonidos generados, sin necesidad de agregar mayor complejidad a los algoritmos de síntesis.

$$Output = f(x, y, z) \tag{64}$$

Se puede experimentar con el uso de los sistemas ya implementados, como señales moduladoras. Estos pueden ser planteados como osciladores de baja frecuencia aplicados en forma de señales envolventes y para modular la frecuencia de corte de filtros digitales. Nótese que para este caso es necesario desarrollar una métrica para evaluar la calidad del audio generado.

Es importante recalcar que la metodología propuesta para el desarrollo del *plug-in* no está limitada a los sistemas dinámicos ya implementados. Una siguiente fase del proyecto podría incluir la adición de más sistemas a partir de los cuales generar el sonido. Algunos sistemas dinámicos que parecen cumplir con las características necesarias para la síntesis de audio son los siguientes:

El atractor de Rössler

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases} \tag{65}$$

El modelo Lotka-Volterra

$$\begin{cases} \dot{x} = x(a - by) \\ \dot{y} = -y(c - dx) \end{cases} \quad (66)$$

Finalmente un avance sumamente significativo resultaría de la exploración de métodos para reducir el costo computacional de los algoritmos implementados. De esta forma se mejoraría el desempeño del *plug-in* para presentaciones en vivo y permitiría su uso en una mayor variedad de equipos de cómputo. El *plug-in* implementado se basó en el texto *Designing Software Synthesizer Plug-Ins in C++* [15], el cual hace la aclaración que los métodos presentados no están optimizados sino se presentan con fines didácticos. El *plug-in* puede ser mejorado al punto de cumplir con estándares comerciales.

-
- [1] Rick Bidlack. “Chaotic Systems as Simple (But Complex) Compositional Algorithms”. En: *Computer Music Journal* 16.3 (1992), págs. 33-47. ISSN: 0148-9267. DOI: 10.2307/3680849. URL: <http://www.jstor.org/stable/3680849> (visitado 04-06-2018).
 - [2] Eleonora Bilotta, Stefania Gervasi y Pietro Pantano. “Reading complexity in chua’s oscillator through music. part i: a new way of understanding chaos”. En: *International Journal of Bifurcation and Chaos* 15.02 (feb. de 2005), págs. 253-382. ISSN: 0218-1274. DOI: 10.1142/S0218127405012156. URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218127405012156> (visitado 04-06-2018).
 - [3] Chapra Canale. *Numerical Methods For Engineers, 7 Ed.* 7th edition. McGraw Hall India, 2014. ISBN: 978-93-5260-213-1.
 - [4] Hugo Fastl y Eberhard Zwicker. *Psychoacoustics: Facts and Models.* 3.^a ed. Berlin Heidelberg: Springer-Verlag, 2007. ISBN: 978-3-540-23159-2. URL: [//www.springer.com/us/book/9783540231592](http://www.springer.com/us/book/9783540231592) (visitado 02-10-2018).
 - [5] Shuichi Kinoshita, ed. *Pattern Formations and Oscillatory Phenomena.* 1 edition. Amsterdam: Elsevier, jun. de 2013. ISBN: 978-0-12-397014-5.
 - [6] Edward N. Lorenz. “Deterministic Nonperiodic Flow”. En: *Journal of the Atmospheric Sciences* 20.2 (mar. de 1963), págs. 130-141. ISSN: 0022-4928. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. (Visitado 30-04-2018).
 - [7] Stephen Lynch. *Dynamical Systems with Applications using MATLAB®.* 2nd ed. 2014 edition. New York, N.Y: Birkhäuser, jul. de 2014. ISBN: 978-3-319-06819-0.
 - [8] David Medine y David Medine. “Dynamical Systems for Audio Synthesis: Embracing Nonlinearities and Delay-Free Loops”. En: *Applied Sciences* 6.5 (mayo de 2016), pág. 134. DOI: 10.3390/app6050134. URL: <http://www.mdpi.com/2076-3417/6/5/134> (visitado 29-08-2018).
 - [9] Paul Messick. *Maximum MIDI : Music Applications in C++.* Greenwich: Manning Publications, ago. de 1997. ISBN: 978-1-884777-44-8.
 - [10] Eduardo Miranda. *Computer Sound Design: Synthesis techniques and programming.* Edición: 2. Amsterdam: Focal Press, sep. de 2002. ISBN: 978-0-240-51693-6.

- [11] H. Nyquist. “Certain Topics in Telegraph Transmission Theory”. En: *Transactions of the American Institute of Electrical Engineers* 47.2 (abr. de 1928), págs. 617-644. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1928.5055024.
- [12] Andrea Pejrolo y Richard DeRosa. *Acoustic and MIDI Orchestration for the Contemporary Composer: A Practical Guide to Writing and Sequencing for the Studio Orchestra*. Edición: 1. Boston: Focal Press, sep. de 2007. ISBN: 978-0-240-52021-6.
- [13] Eleonora Bilotta Pietro Pantano. *A Gallery of Chua Attractors*. URL: <https://www.worldscientific.com/worldscibooks/10.1142/6720> (visitado 29-04-2018).
- [14] Will Pirkle. *Designing Audio Effect Plug-Ins in C++: With Digital Audio Signal Processing Theory*. 1 edition. Burlington, MA ; Abingdon, Oxon: Focal Press, oct. de 2012. ISBN: 978-0-240-82515-1.
- [15] Will Pirkle. *Designing Software Synthesizer Plug-Ins in C++: For RackAFX, VST3, and Audio Units*. 1 edition. Burlington, MA: Routledge, nov. de 2014. ISBN: 978-1-138-78707-0.
- [16] Jeff R. Cash y Alan Karp. “A Variable Order Runge—Kutta Method for Value Problems with Rapidly Varying Right-hand Sides”. En: 16 (sep. de 1990), págs. 201-222.
- [17] *RackAFX / RackAFX*. URL: <http://www.willpirkle.com/rackafx/> (visitado 04-09-2018).
- [18] Claude E. Shannon. *Communication in the presence of noise*. Institute of Radio Engineers, 1949.
- [19] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. 2 edition. Boulder, CO: CRC Press, mar. de 2015. ISBN: 978-0-8133-4910-7.
- [20] J. Suykens y J. Vandewalle. “Quasilinear approach to nonlinear systems and the design of n-double scroll (n=1, 2, 3, 4, . . .)” En: *IEE Proceedings G - Circuits, Devices and Systems* 138.5 (oct. de 1991), págs. 595-603. ISSN: 0956-3768. DOI: 10.1049/ip-g-2.1991.0098.
- [21] Akio Tsuneda. “A gallery of attractors from smooth chua’s equation”. En: *International Journal of Bifurcation and Chaos* 15.01 (ene. de 2005), págs. 1-49. ISSN: 0218-1274. DOI: 10.1142/S0218127405011990. URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218127405011990> (visitado 03-06-2018).
- [22] Stephen Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. 2nd edition. New York: Springer, oct. de 2003. ISBN: 978-0-387-00177-7.
- [23] Alan Wolf y col. “Determining Lyapunov exponents from a time series”. En: *Physica D: Nonlinear Phenomena* 16.3 (jul. de 1985), págs. 285-317. ISSN: 0167-2789. DOI: 10.1016/0167-2789(85)90011-9. URL: <http://www.sciencedirect.com/science/article/pii/0167278985900119> (visitado 05-05-2018).
- [24] Guo-Qun Zhong. “Implementation of Chua’s circuit with a cubic nonlinearity”. En: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 41.12 (dic. de 1994), págs. 934-941. ISSN: 1057-7122. DOI: 10.1109/81.340866.
- [25] Dennis G. Zill. *Bundle: Differential Equations with Boundary-Value Problems, Loose-leaf Version, 9th + WebAssign Printed Access Card for Zill’s Differential Equations ... Problems, 9th Edition, Single-Term*. 9 edition. Brooks Cole, ene. de 2017. ISBN: 978-1-337-60490-1.

17.1. Implementación de métodos numéricos de paso variable

Los pasos numéricos de paso variable son una buena opción para la reducción de consumo de recursos computacionales, ya que estos ajustan el tamaño del paso según el error actual. La complejidad de utilizarlos para la síntesis de audio está en que se requiere de un flujo de datos equidistantes en el tiempo. En la siguiente figura se presenta una comparación entre el contenido espectral de una solución al oscilador de Chua utilizando un método de paso fijo y un método de paso variable.

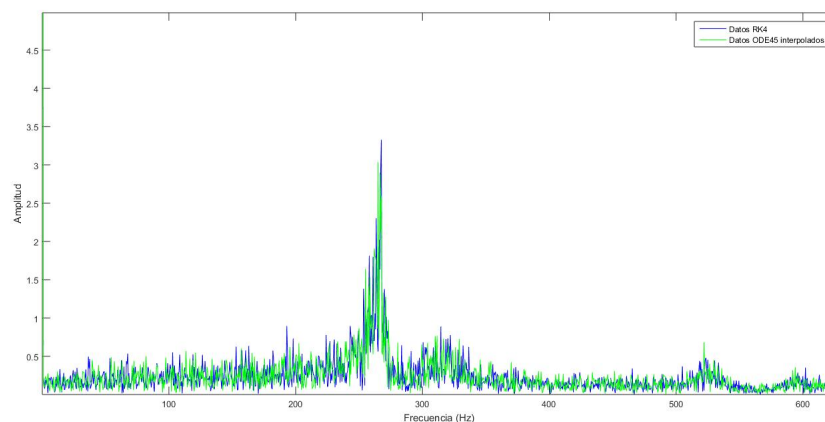


Figura 46: Comparación del contenido espectral entre solución con ODE45 y RK4

Para obtener los datos equidistantes en la solución con el método de paso variable ODE45, fue necesario realizar interpolaciones de los datos. Sería necesario realizar la interpolación de datos en tiempo real para hacer síntesis de audio con un método de paso variable. Es posible que el uso de recursos computacionales de la interpolación resulte en un mayor peso

computacional que el método de paso fijo, sin embargo, esto debe ser verificado.

- Aliasing** Efecto que distorsiona una señal muestreada respecto a la señal original. Es ocasionado por la presencia de contenido espectral en la señal continua que está más allá de la frecuencia de Nyquist.. 11, 39
- API** Interfaz de programación de aplicación o API por sus siglas en inglés. Es una clase abstracta que se utiliza como estándar para el desarrollo de *plug-ins* compatibles entre software de distintos fabricantes.. 18, 19
- Ataque** Segmento de una señal de sonido comprendido desde el inicio de este hasta el punto de amplitud máxima. Se dice que una señal tiene más ataque mientras el ascenso al punto máximo es más rápido.. 14, 67
- Atractor** Punto o conjunto de puntos que atraen trayectorias con condiciones iniciales lo suficientemente cercanas, las cuales permanecen en la vecindad del atractor incluso bajo la influencia de pequeñas perturbaciones.. 28, 37
- Atractor extraño** Atractor que presenta caos en la evolución de sus trayectorias.. 28, 37, 39, 50, 51
- Biblioteca de enlace dinámico** Objeto de software con funciones pre compiladas que se comunica con un programa en ejecución, conocido como el cliente, por medio de enlaces dinámicos.. 17
- Campo vectorial** Expresión que describe la distribución espacial de magnitudes vectoriales.. 20, 44, 48, 50
- Ciclo límite** órbita a la cual convergen todas las trayectorias de un sistema dinámico independientemente de sus condiciones iniciales. 27, 37, 38, 52
- Condiciones iniciales** Punto inicial de una trayectoria en el espacio de fase.. 19, 35, 37
- Contenido espectral** Magnitudes que representan una señal en el dominio de la frecuencia. 11, 45, 48

- Controlador MIDI** Equipo de hardware que utiliza el estándar MIDI para interconectarse con un ordenador y así manipular un instrumento digital.. 16, 36, 65
- Espacio de fase** Espacio donde se representa la evolución de las variables de estado de un sistema dinámico.. 19, 39, 40, 43, 44, 48
- Espectrograma** Representación gráfica del espectro de una señal a lo largo del tiempo. Se obtiene aplicando la transformada rápida de Fourier a pequeñas ventanas de tiempo.. 38, 42
- Estación de trabajo de audio virtual** Software utilizado para la grabación, edición y generación de audio a nivel comercial o de uso personal. Conocidos como softwares DAW por sus siglas en inglés (digital audio workstation).. 36
- Exponente de Lyapunov** Son cantidades escalares que brindan información cualitativa y cuantitativa sobre el comportamiento de un sistema. Relacionados con la divergencia de trayectorias cercanas y presencia de atractores caóticos. Un sistema dinámico cuenta con la misma cantidad de exponentes de Lyapunov que de variables de estado. . 28, 43, 51
- Filtro digital** Sistema discreto implementado en forma de una ecuación de diferencias que se utiliza para modificar el contenido espectral de una señal discreta.. 54
- Frecuencia de Nyquist** Frecuencia máxima que puede ser representada fielmente por una señal discreta producto del muestreo de una señal analógica. Esta frecuencia corresponde a la mitad de la frecuencia de muestreo.. 11, 12
- Frecuencia dominante** Frecuencia o rango angosto de frecuencias donde se encuentra la componente de mayor potencia de una señal aperiódica. 35, 37, 39, 41, 42, 45, 48, 59
- Frecuencia fundamental** Frecuencia con la cual se repite el patrón de una señal periódica. 35, 38, 39
- Integración numérica** Aproximación numérica de la integral de una función diferencial.. 22, 36, 53
- Interfaz gráfica de usuario** Interfaz de software por medio de la cual se da la interacción con el usuario de forma gráfica. Conocida en globalmente como GUI por sus siglas en inglés.. 18
- MIDI** Interfaz musical de instrumentos digitales, conocido globalmente como MIDI por sus siglas en inglés. Es un estándar de interconexión entre interfaces de audio, instrumentos digitales, ordenadores y equipo de iluminación.. 16, 65, 66
- Oscilador** Sistema cuya respuesta se da en forma de perturbaciones oscilatorias.. 13
- Parámetros** Constantes en el modelo de un sistema dinámico.. 19, 59, 67
- Plug-in** Complemento de software que agrega funcionalidad a un programa conocido como el cliente. En el sistema operativo Windows es implementado como una biblioteca de enlace dinámico.. 17, 18, 57, 58, 66, 68

- Punto fijo** Punto en el espacio de fase donde el campo vectorial del sistema dinámico es igual a cero.. 19
- RackAFX** Software que facilita la implementación de *plug-ins* utilizando distintas APIs.. 17, 57, 63, 66
- Señales envolventes** Señales que se utilizan para modular la amplitud de otra señal.. 14, 63, 67
- Sintetizador de audio** Instrumento utilizado para realizar síntesis de audio.. 12
- Sistema autónomo** Sistema dinámico cuyas variables de estado no muestran dependencia explícita del tiempo.. 19
- Sistema caótico** Sistema dinámico que presenta alta dependencia de las condiciones iniciales y comportamiento aperiódico.. 27, 41
- Sistema dinámico** Sistema de ecuaciones que modela la evolución en el tiempo de un conjunto de variables de estado. 19, 21, 55, 58, 67
- Sistema no lineal** Sistema dinámico que no puede ser descrito por una matriz de coeficientes constantes que multiplica al vector de estados.. 21
- Síntesis de audio** Proceso por el cual se genera sonido por medios no acústicos.. 12, 35, 37, 53
- Transformada rápida de Fourier** Algoritmo que utiliza la transformada discreta de Fourier para extraer el contenido espectral de una señal discreta.. 13, 36, 45, 69
- Trayectoria** Evolución de un punto inicial en el espacio de fase. Corresponde a una solución del sistema dinámico.. 19
- Variables de estado** Variables que definen el estado de un sistema dinámico.. 19, 21, 44
- VST** Interfaz de programación de uso comercial para el desarrollo de *plug-ins* de efectos digitales de audio e instrumentos digitales.. 18, 36, 63
- Órbita** set de puntos que crea una trayectoria cerrada en el espacio de fase. 20, 37, 38