



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA  
Facultad de Ciencias de la Electrónica

**TESIS**

COMO REQUISITO PARCIAL PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN ELECTRÓNICA

TÍTULO DE LA TESIS:

SINCRONIZACIÓN DE DOS SISTEMAS CAÓTICOS DE ORDEN  
FRACCIONARIO EN HARDWARE EMBEBIDO ANALÓGICO

PRESENTA

Eduardo Reyes Coeto

Asesores:

Dr. Jesús Manuel Muñoz Pacheco  
Dra. Luz del Carmen Gómez Pavón

Junio 2019



# Agradecimientos

Agradezco a mi familia por el apoyo incondicional que me ha brindado siempre, gracias a ellos estoy logrando esta meta.

Agradezco a mis asesores de tesis, el Dr. Jesús Manuel Muñoz Pacheco y la Dra. Luz del Carmen Gómez Pavón, por su paciencia, confianza y conocimientos compartidos, ustedes hicieron posible la realización de este proyecto de tesis.

Al comité revisor, el Dr. Arnulfo Luis Ramos, el Dr. Luis Abraham Sánchez Gaspariano y el Dr. Ambrosio Lázaro Roberto Carlos, cuyas observaciones han enriquecido este trabajo de tesis.

Agradezco al Dr. Ernesto Zambrano Serrano quien al compartir espacio de trabajo conmigo siempre mostró disposición de ayudarme a resolver todas mis dudas.

Agradezco al CONACYT que a través del programa Ciencia Básica me otorgo una beca para el desarrollo de esta tesis de licenciatura mediante el proyecto no. 258880 de nombre DISEÑO ELECTRÓNICO DE SISTEMAS DINÁMICOS DE ORDEN FRACCIONAL Y ENTERO CONSIDERANDO NO IDEALIDADES.

Agradezco a la Facultad de Ciencias de la Electrónica de la BUAP por los años de formación brindados a mis profesores, compañeros y amigos que siempre me motivaron.



*Dedicado a mi familia*

*“Cada adversidad, cada fracaso, cada angustia, lleva consigo la semilla de un beneficio igual o mayor”.*

*Napoleón Hill*



# Contenido

Resumen . . . . .	IX
<b>1. Introducción</b>	<b>1</b>
1.1. Justificación . . . . .	2
1.2. Objetivos . . . . .	3
1.2.1. Objetivo General . . . . .	3
1.2.2. Objetivos Específicos . . . . .	3
<b>2. Implementación de integrador de orden fraccionario utilizando filtro bilineal</b>	<b>4</b>
2.1. Fundamentos y definiciones en cálculo fraccionario . . . . .	4
2.1.1. Operador fraccionario de Laplace y función de transferencia . . . . .	4
2.2. Aproximación de primer orden para $s^{-\alpha}$ . . . . .	6
2.3. Escalamiento . . . . .	7
2.4. Configuración de filtro bilineal . . . . .	8
2.5. Diseño en software AnadigmDesigner2 . . . . .	13
2.5.1. Configuración de Relojes . . . . .	14
2.5.2. Configuración celdas IO . . . . .	14
2.6. Caracterización del integrador de orden fraccionario . . . . .	15
2.7. Análisis de gráficas . . . . .	18
<b>3. Implementación de integrador de orden fraccionario utilizando filtros pasa bajas y pasa altas</b>	<b>20</b>
3.1. Implementación en FPAA . . . . .	24
3.1.1. Filtro bilineal configurado como pasa-bajas . . . . .	24
3.1.2. Filtro bilineal configurado como pasa-altas . . . . .	25
3.1.3. Suma diferencial . . . . .	25
3.2. Valores de implementación . . . . .	26
3.2.1. Órdenes 0.01 a 0.9 . . . . .	27
3.2.2. Valores de implementación y restricciones órdenes 0.91 a 0.93 . . . . .	27
3.2.3. Valores de implementación y restricciones órdenes 0.94 a 0.96 . . . . .	28
3.2.4. Valores de implementación y restricciones órdenes 0.97 a 0.98 . . . . .	28

---

3.2.5.	Valores de implementación y restricciones órdenes 0.99 . . . . .	29
3.3.	Diseño en software AnadigmDesigner2 . . . . .	29
3.3.1.	Configuración de Relojes . . . . .	30
3.3.2.	Configuración de Celdas IO . . . . .	30
3.3.3.	Configuración de CAMs . . . . .	30
3.4.	Caracterización del integrador de orden fraccionario implementado con filtros pasa bajas y pasa altas . . . . .	31
3.4.1.	Ajuste de Curvas . . . . .	35
<b>4.</b>	<b>Oscilador Caótico de Lü con integradores de orden fraccionario</b>	<b>41</b>
4.1.	Variables de estado del oscilador caótico de Lü . . . . .	41
4.2.	Función Saturada . . . . .	42
4.3.	Oscilador de caótico de Lü con integradores de orden fraccionario implementado en FPAA . . . . .	44
4.3.1.	Oscilador caótico de Lü con integradores de orden $\alpha = 0.8$ implementados en filtro bilineal . . . . .	45
4.3.2.	Oscilador caótico de Lü con integradores de orden fraccionario implementados en filtros pasa-bajas y paso-altas . . . . .	52
<b>5.</b>	<b>Sincronización de dos osciladores caóticos de Lü de orden fraccio- nario</b>	<b>58</b>
5.1.	Sincronización basada en el teorema del valor final de la transformada de Laplace . . . . .	58
5.2.	Diseño en software Anadigm de sistema maestro . . . . .	60
5.3.	Diseño en software Anadigm de sistema seguidor . . . . .	62
5.3.1.	Configuración de relojes para sistema seguidor . . . . .	64
5.3.2.	Configuración de CAMs para sistema seguidor . . . . .	64
5.4.	Caracterización de la sincronización entre los dos osciladores caóticos de Lü . . . . .	66
5.4.1.	Osciladores caóticos de Lü de orden 0.9 de 2 enrollamientos sincronizados . . . . .	67
5.4.2.	Osciladores caóticos de Lü de orden 0.9 de 4 enrollamientos sincronizados . . . . .	69
5.4.3.	Envío y recepción de una señal de información . . . . .	70
<b>6.</b>	<b>Conclusiones</b>	<b>73</b>
6.1.	Trabajo futuro . . . . .	74
<b>A.</b>	<b>Codigo MATLAB</b>	<b>75</b>

---

---

<b>B. Valores de frecuencia de relojes</b>	<b>77</b>
B.1. Frecuencias para implementar integrador de orden fraccionario en filtro bilineal . . . . .	77
B.2. Frecuencias para implementar integrador de orden fraccionario utilizando filtros pasa altas y pasa bajas . . . . .	84
B.2.1. Reloj del sistema 16000 kHz . . . . .	84

# Índice de figuras

2.1.	Diagrama de Bode graficado en MATLAB de aproximación de primer orden para el integrador de orden fraccionario con $\alpha = 0.9$ . . . . .	6
2.2.	Diagrama de Bode graficado en MATLAB de aproximación de primer orden para el integrador de orden fraccionario escalado en frecuencia con un factor $k_f = \frac{1}{2\pi 1000}$ y $\alpha = 0.9$ . . . . .	8
2.3.	Ventana de ajuste de parámetros de filtro bilineal . . . . .	10
2.4.	Diseño de integrador de orden fraccionario en AnadigmDesigner2 . .	13
2.5.	Ventana de configuración celda IO 1. . . . .	14
2.6.	Ventana de configuración celda IO 3. . . . .	15
2.7.	Esquema general de conexión . . . . .	16
2.8.	Diagrama de bode de integrador de orden fraccionario implementado en FPAA obtenido con NI ELVIS II. . . . .	17
2.9.	Puntos de muestra para calcular la pendiente. . . . .	17
2.10.	Comparación entre el diagrama de Bode obtenido de forma experimental y el diagrama de Bode simulado en Matlab . . . . .	18
3.1.	Diagrama de bloques de implementación de integrador de orden fraccionario utilizando filtro pasa-bajo y pasa-alto. . . . .	23
3.2.	Diseño de integrador de orden fraccionario implementado con filtros pasa-bajas y pasa-alta en AnadigmDesigner2 . . . . .	30
3.3.	Configuración de CAMs . . . . .	31
3.4.	Gráfica comparativa entre valores de pendiente ideal y valores de pendiente obtenidos experimentalmente. . . . .	33
3.5.	Gráfica comparativa entre valores de pendiente ideal y valores de pendiente obtenidos experimentalmente. . . . .	34
3.6.	Ventana de configuración de función de ajuste polinomial en Curve Fitting Toolbox. . . . .	35
3.7.	Ventana de configuración función de ajuste exponencial en Curve Fitting Toolbox. . . . .	36
3.8.	Gráfica de valores de $\hat{A}$ en función del orden $\alpha$ . . . . .	36

3.9. Gráficas de la curva de ajuste exponencial para los valores de $\hat{A}$ y los errores de aproximación . . . . .	37
3.10. Gráficas de la curva de ajuste polinomial para los valores de $\hat{A}$ y los errores de aproximación . . . . .	38
4.1. Oscilador caótico de Lu con integradores de orden fraccionario . . . . .	42
4.2. Función saturada para generar dos enrollamientos en oscilador caótico	43
4.3. Función saturada para generar cuatro enrollamientos en oscilador caótico . . . . .	43
4.4. Oscilador caótico de Lu con integradores de orden 0.8 . . . . .	46
4.5. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 1 . . . . .	47
4.6. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 2 . . . . .	48
4.7. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 3 . . . . .	48
4.8. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 4 . . . . .	49
4.9. Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden $\alpha = 0.8$ , 2 enrollamientos. . . . .	49
4.10. Respuesta en el dominio del tiempo de oscilador caótico de Lu, orden $\alpha = 0.8$ . . . . .	50
4.11. Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden $\alpha = 0.8$ , 4 enrollamientos. . . . .	50
4.12. Respuesta en el dominio del tiempo de oscilador caótico de Lu, orden $\alpha = 0.8$ . . . . .	51
4.13. Oscilador caótico de Lu con integradores de orden 0.9 . . . . .	52
4.14. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 1 . . . . .	53
4.15. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 2 . . . . .	53
4.16. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 3 . . . . .	54
4.17. Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 4 . . . . .	54
4.18. Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden $\alpha = 0.9$ , 2 enrollamientos. . . . .	55
4.19. Respuesta en el dominio del tiempo de oscilador caótico de Lu, orden $\alpha = 0.9$ . . . . .	55

4.20. Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden $\alpha = 0.9$ , 4 enrollamientos. . . . .	56
4.21. Respuesta en el dominio del tiempo de oscilador caótico de Lu, orden $\alpha = 0.9$ . . . . .	56
4.22. Implementación física de oscilador caótico de Lü de orden $\alpha = 0.9$ . . . . .	57
5.1. Diagrama de bloques de sistema de sincronización entre dos osciladores caóticos de Lü de orden fraccionario de la forma maestro-seguidor . . . . .	61
5.2. Oscilador caótico de Lü implementado como sistema maestro . . . . .	62
5.3. Oscilador caótico de Lu implementado como sistema seguidor . . . . .	63
5.4. Bloques de diferencia para implementar $K_1(x_M - x_F)$ y $K_2(y_M - y_F)$ . . . . .	64
5.5. Configuración CHIP 1 . . . . .	65
5.6. Configuración CHIP 2 . . . . .	65
5.7. Configuración CHIP 3 . . . . .	66
5.8. Configuración CHIP 4 . . . . .	66
5.9. Sincronización sistema maestro-seguidor, orden $\alpha=0.9$ , 4 enrollamientos . . . . .	67
5.10. Gráficas en plano de fase . . . . .	68
5.11. Comportamiento en el dominio del tiempo . . . . .	68
5.12. Sincronización sistema maestro-seguidor, orden $\alpha=0.9$ , 4 enrollamientos . . . . .	69
5.13. Gráficas en plano de fase . . . . .	69
5.14. Comportamiento en el dominio del tiempo . . . . .	70
5.15. Envío y recepción de una señal de información utilizando caos para su encriptación . . . . .	71
5.16. Implementación física de sincronización entre dos osciladores caóticos de Lü de orden fraccionario y envío y recepción de una señal encriptada con caos. . . . .	72

# Resumen

En este trabajo de tesis se presenta la implementación de un sistema de sincronización entre dos osciladores caóticos de Lü de orden fraccionario mediante la forma Maestro-Seguidor utilizando matrices analógicas programables por campo (FPAA, por sus siglas en inglés), ambos osciladores caóticos incorporan en su diseño integradores de orden fraccionario. El diseño tanto de los integradores de orden fraccionario como los osciladores caóticos se realizan a través de un software de diseño que se encarga de sintetizar mediante módulos analógicos configurables (CAM, por sus siglas en inglés) el diseño implementado en su interfaz en una configuración que pueda adoptar el FPAA.

Se describe el diseño y procedimiento realizado para implementar un integrador de orden fraccionario en FPAA utilizando el CAM filtro bilineal, esta implementación presenta ciertas limitaciones por lo que posteriormente se describe una segunda implementación utilizando filtro pasa bajas y pasa altas. El integrador de orden fraccionario implementado presenta un pequeño error con respecto al comportamiento esperado por lo que se le realiza un ajuste para compensar el error y optimizar el diseño.

En la siguiente etapa los integradores de orden fraccionario previamente diseñados se incorporan al oscilador caótico de Lü y se comprueba el funcionamiento de todo el sistema, finalmente se procede a sincronizar mediante el teorema de valor final de Laplace en un esquema maestro-seguidor, dos osciladores caóticos de orden fraccionario.

# Capítulo 1

## Introducción

El caos es un comportamiento presente en todo nuestro entorno, abordado en diversos campos de investigación de la física, como la mecánica cuántica, investigación sobre partículas fundamentales, pero incluso puede ser vinculado a experiencias habituales de la vida humana como las fluctuaciones en la bolsa de valores, el comportamiento atmosférico o un líquido en estado turbulento [1]. Es un campo de estudio reciente, originado gracias a los avances de los sistemas computacionales.

Analíticamente podemos describir al caos como un comportamiento dinámico complejo determinista, donde su estado futuro depende de sus condiciones iniciales. La no linealidad y sensibilidad de las condiciones iniciales que producen caos toman un papel muy importante ya que afectan los resultados de forma extrema; en un largo plazo este comportamiento se vuelve impredecible [2].

Un sistema cuyo comportamiento es caótico puede ser modelado y descrito mediante variables de estado y en el campo de la electrónica este se sintetiza físicamente utilizando dispositivos electrónicos, obteniendo en las salidas del sistema señales eléctricas con comportamiento caótico similar al modelo matemático.

Por otro lado cálculo de orden fraccionario es un tema que al igual que los osciladores caóticos ha tomado importancia en las últimas décadas por la comunidad científica e ingeniería, sin embargo tiene tres siglos de antigüedad como el cálculo convencional [3], tuvo lugar después de la publicación, en 1675, de un documento de G. W. Leibniz, donde apareció el símbolo:  $\frac{d^n}{dx^n}$ , el cual se refiere a la derivada de orden  $n$  de la función y respecto de  $x$ , donde  $n$  es un número natural. Leibniz (1646-1716), al responder a una carta de G. F. Antoine, marqués de l'Hôpital (1641-1704), cuestiona sobre qué sucedería si  $n$  fuera 0.5, es aquí donde surgen las primeras indagaciones del cálculo fraccionario [4].

Lo interesante de este tema es que las derivadas e integrales fraccionarias agregan otra dimensión para entender o describir la naturaleza de una mejor manera, es quizás un lenguaje matemático más eficiente para traducir la realidad de la naturaleza. En los últimos años este tema ha sido llevado a varios campos aplicados de la cien-

cia como: física, biología, química, geología, teoría de control, electromagnetismo, eléctrica, circuitos eléctricos, procesamiento de imágenes, procesamiento de señales y óptica [4].

El reto latente para los próximos años es tener aplicaciones reales basadas en este tópico; sin embargo, para lograr ese objetivo se requiere la realización física de los integradores de orden fraccionario.

En este trabajo de tesis se propone un esquema de sincronización entre dos osciladores caóticos de orden fraccionario, este tipo de osciladores incorporan como parte fundamental de su sistema integradores de orden fraccionario por lo que también se propone un diseño de este integrador optimizado para incorporarse en el oscilador caótico.

## 1.1. Justificación

Los sistemas de orden fraccionario son elementos de circuito que se caracterizan por una respuesta de magnitud que ronda alrededor de los  $\pm 20\alpha$  decibelios por década y una respuesta de fase constante en todas las frecuencias de  $\pm 90\alpha$  grados [5], donde  $\alpha$  es un número real positivo.

Actualmente se han reportado muy pocas implementaciones físicas de sistemas de orden fraccionario. Por ejemplo en [6] y [7] fueron reportados capacitores de orden fraccionario con una impedancia  $Z = \frac{1}{s^\alpha C}$  los cuales desafortunadamente son voluminosos, requieren compuestos químicos de difícil manipulación y tienen órdenes a que no pueden modificarse fácilmente. Alternativamente, los sistemas fraccionarios pueden ser aproximados en un cierto ancho de banda con funciones racionales en  $s$  de los métodos de Newton, Muir, Oustaloup, Mtsuda, expansión de series de potencias (PSE, por sus siglas en inglés), expansión de fracciones continuas (CFE, por sus siglas en inglés), entre muchos otros [5, 8, 9].

Una vez que la función racional es obtenida, esta puede ser sintetizada con redes de escalera de Cauer y Foster, estructuras de árboles, líneas de transmisión o utilizando bloques de filtro pasa banda de baja  $Q$  y filtro rechaza banda [8–10]. Los componentes del circuito involucrados pueden ser resistencias, inductores, capacitores y convertidores de impedancia negativa [11], pero el inconveniente de estas realizaciones es la dificultad de aproximar los valores requeridos con los valores de las resistencias, capacitores e inductores disponibles comercialmente [9, 11], [12, 13].

Una vez que se ha obtenido una realización aproximada de un sistema de orden fraccionario, se puede utilizar para desarrollar integradores y diferenciadores de orden fraccionario [7, 8, 13], pero los circuitos resultantes heredan las dificultades de implementación mencionadas anteriormente. Para hacer frente a este problema, en este trabajo proponemos trabajar con dispositivos embebidos FPAA.

Un circuito analógico programable (FPAA) es un arreglo matricial de bloques configurables que nos permite trasladar circuitos analógicos complejos a un conjunto simple de funciones de bajo nivel y trabajarlo en tiempo real, lo cual permitirá diseñar sistemas analógicos complejos y funciones de procesamiento en un dispositivo integrado, previamente probado, optimizado y programable eliminando así los inconvenientes de conlleva trabajar con amplificadores operacionales, capacitores y resistores discretos.

La interfaz de uso de los FPAA es a través de Módulos Analógicos Configurables (CAM), los cuales nos representan integradores, sumadores, derivadores, entre muchos otros bloques prediseñados necesarios para sintetizar una ecuación de estados en un sistema analógico, esto permitirá implementar el oscilador caótico de múltiples enrollamientos; sin embargo, su diseño debe ser ajustado a las características y recursos con los que cuentan los FPAA.

La última etapa y principal de este proyecto de tesis es implementar dos osciladores caóticos de múltiples enrollamientos uno como observador y otro como maestro y diseñar la metodología que permita que el sistema observador siga los estados del maestro con una simetría aceptable.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Implementar físicamente dos osciladores caóticos con integradores de orden fraccionario, así como su esquema de sincronización usando hardware embebido FPAA.

### 1.2.2. Objetivos Específicos

- Simulación numérica en MATLAB de un integrador de orden fraccionario con una respuesta en frecuencia diferente a -20dB.
- Diseñar e implementar en FPAA un integrador de orden fraccionario.
- Diseñar e implementar en FPAA un oscilador caótico de orden fraccionario de múltiples enrollamientos ajustable a los órdenes 0.1 a 0.99.
- Diseñar e implementar en FPAA el esquema de sincronización de dos osciladores de orden fraccionario.

# Capítulo 2

## Implementación de integrador de orden fraccionario utilizando filtro bilineal

La implementación de funciones analógicas complejas en los FPAA fabricados por la empresa Anadigm se realizan a través del software de diseño AnadigmDesigner2 EDA, este software posee una biblioteca amplia de CAMs previamente optimizados, los cuales se utilizan como bloques de construcción. Realizando una revisión de los CAMs disponibles se encontró que el CAM de Filtro Bilineal posee características que lo hacen idóneo para la implementación del integrador de orden fraccionario, así como el uso de pocos recursos del FPAA. En el presente capítulo se describe la implementación y caracterización de un integrador de orden fraccionario utilizando este CAM.

### 2.1. Fundamentos y definiciones en cálculo fraccionario

Entre todas las definiciones para el cálculo de derivadas e integrales fraccionales, la definición de Riemann-Liouville establece [14]:

$$D^{\alpha}_t f(t) = \frac{1}{\Gamma(m - \alpha)} \left( \frac{d}{dt} \right)^m \int_0^t \frac{f(\tau) d\tau}{(t - \tau)^{\alpha - m + 1}} \quad (2.1)$$

Donde  $\alpha \in \mathbb{R}$ ,  $m - 1 < \alpha < m$ ,  $m \in \mathbb{N}$  y  $\Gamma(\cdot)$  es la función Gama. Para los casos  $\alpha > 0$ ,  $\alpha < 0$  y  $\alpha = 0$  se puede obtener la derivada de orden fraccionario, la integral de orden fraccionario y la función identidad.

#### 2.1.1. Operador fraccionario de Laplace y función de transferencia

La transformada de Laplace con condiciones iniciales iguales a cero de 2.1 está dada por [14]:

$$\mathcal{L}\{D^{\alpha}_t f(t)\} = s^{\alpha} F(s) \quad (2.2)$$

Donde  $F(s)$  denota la transformada de Laplace de  $f(t)$ , y  $s^\alpha$  es el operador de orden fraccionario de Laplace expresado como:

$$s^\alpha = (jw)^\alpha = w^\alpha \left( \cos\left(\frac{\alpha\pi}{2}\right) + j \sin\left(\frac{\alpha\pi}{2}\right) \right) \quad (2.3)$$

Debido a que la transformada de Laplace es un operador lineal, puede ser aplicado a una ecuación diferencial de orden fraccionario con coeficientes  $a_k$ ,  $b_k$  y señales de entrada y salida  $u(t)$   $e(t)$  para obtener la función de transferencia [15]:

$$H(s) = \frac{U(S)}{E(S)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + \beta_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + \alpha_0 s^{\alpha_0}} \quad (2.4)$$

Donde

$$U(s) = \mathcal{L}\{u(t)\} \quad (2.5)$$

y

$$E(s) = \mathcal{L}\{e(t)\} \quad (2.6)$$

Los operadores  $s^\alpha$  y  $s^{-\alpha}$  no se pueden implementar directamente, se requiere realizar aproximaciones. Por ejemplo, la aproximación de primer orden con  $\alpha \in (0, 1)$  obtenida con el método de expansión de fracciones continuas (CFE) tiene la forma: [5]:

$$\frac{1}{s^\alpha} \approx \frac{(1-\alpha)s + (1+\alpha)}{(1+\alpha)s + (1-\alpha)} = \frac{As + 1}{s + A}, \quad \text{donde } A = \frac{1-\alpha}{1+\alpha} \quad (2.7)$$

$$s^\alpha \approx \frac{(1+\alpha)s + (1-\alpha)}{(1-\alpha)s + (1+\alpha)} = \frac{Bs + 1}{s + B}, \quad \text{donde } B = \frac{1-\alpha}{1+\alpha} \quad (2.8)$$

Mientras que la aproximación de segundo orden para ambos casos resulta en:

$$\frac{1}{s^\alpha} \approx \frac{(\alpha^2 - 3\alpha + 2)s^2 + (8 - 2\alpha^2)s + (\alpha^2 + 3\alpha + 2)}{(\alpha^2 + 3\alpha + 2)s^2 + (8 - 2\alpha^2)s + (\alpha^2 - 3\alpha + 2)} = \frac{Ds^2 + Cs + 1}{s^2 + Cs + D} \quad (2.9)$$

$$s^\alpha \approx \frac{(\alpha^2 + 3\alpha + 2)s^2 + (8 - 2\alpha^2)s + (\alpha^2 - 3\alpha + 2)}{(\alpha^2 - 3\alpha + 2)s^2 + (8 - 2\alpha^2)s + (\alpha^2 + 3\alpha + 2)} = \frac{Fs^2 + Es + 1}{s^2 + Es + F} \quad (2.10)$$

con

$$C = \frac{8 - 2\alpha^2}{\alpha^2 + 3\alpha + 2}, \quad D = \frac{\alpha^2 - 3\alpha + 2}{\alpha^2 + 3\alpha + 2}, \quad E = \frac{8 - 2\alpha^2}{\alpha^2 - 3\alpha + 2}, \quad F = \frac{\alpha^2 + 3\alpha + 2}{\alpha^2 - 3\alpha + 2} \quad (2.11)$$

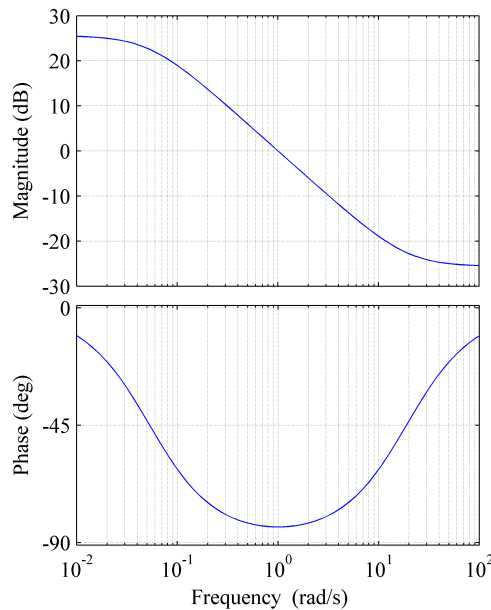
Con esta aproximación, los operadores  $s^\alpha$  y  $s^{-\alpha}$  se trabajan como orden entero, lo que facilita su complementación en un circuito físico. Además, este procedimiento se puede generalizar a aproximaciones de orden superior.

## 2.2. Aproximación de primer orden para $s^{-\alpha}$

El operador  $s^{-\alpha}$  representa a un integrador de orden fraccionario, este no tiene una implementación directa por lo que se requiere utilizar aproximaciones racionales que asemejen su comportamiento, es por esto que se recurre a utilizar la aproximación de primer orden obtenida a través del método de expansión de fracciones continuas [5].

$$\frac{1}{s^\alpha} \approx \frac{As + 1}{s + A}, \quad \text{donde} \quad A = \frac{1 - \alpha}{1 + \alpha} \quad (2.12)$$

Utilizando el software de cómputo numérico MATLAB se realiza un programa para obtener la gráfica de Bode de la respuesta en frecuencia de esta aproximación, ver apéndice A.1. La gráfica obtenida se muestra en la figura 2.1, en la cual la gráfica superior es la magnitud (en dB) y la inferior es la fase (en grados) de la respuesta de esta aproximación en función de la frecuencia.



**Figura 2.1:** Diagrama de Bode graficado en MATLAB de aproximación de primer orden para el integrador de orden fraccionario con  $\alpha = 0.9$

La aproximación se encuentra normalizada en frecuencia, como se observa en la figura 2.1, y el rango de operación en frecuencia en el cual esta aproximación tiene un comportamiento similar a un integrador abarca aproximadamente desde  $0.1 \text{ rad/s}$  a  $10 \text{ rad/s}$ , este es un rango muy bajo en frecuencia y presenta dificultades para implementarse y caracterizarse físicamente, es por esto que se procede a realizar un escalamiento en frecuencia.

## 2.3. Escalamiento

El escalamiento en frecuencia es el proceso de desplazar la respuesta en frecuencia de un sistema por arriba o abajo del eje de frecuencia sin modificar la impedancia [16], se recurre a este método para modificar los rangos de operación en frecuencia de un sistema y desplazarlos a los rangos más convenientes para implementar.

Debido a que los rangos de operación en frecuencia para la aproximación de primer orden al integrador de orden fraccionario son muy bajos se recurre a este método. El factor de escalamiento en frecuencia es el cociente de la frecuencia deseada entre la frecuencia original como se muestra en la ecuación 2.13

$$k_f = \frac{\omega'}{\omega} \quad (2.13)$$

Realizando un despeje podemos definir a la frecuencia escalada como el producto de la frecuencia original por el factor de escalamiento como se muestra en la ecuación 2.14

$$\omega' = \omega k_f \quad (2.14)$$

Sí reescribimos 2.12 en el dominio de la frecuencia quedaría de la siguiente forma

$$\frac{1}{(j\omega)^\alpha} \approx \frac{A(j\omega) + 1}{(j\omega) + A} \quad (2.15)$$

Aplicamos el escalamiento, siendo  $\omega'$  la frecuencia buscada la sustituimos por 2.14

$$\frac{A(j\omega') + 1}{(j\omega') + A} = \frac{A(j\omega k_f) + 1}{(j\omega k_f) + A} \quad (2.16)$$

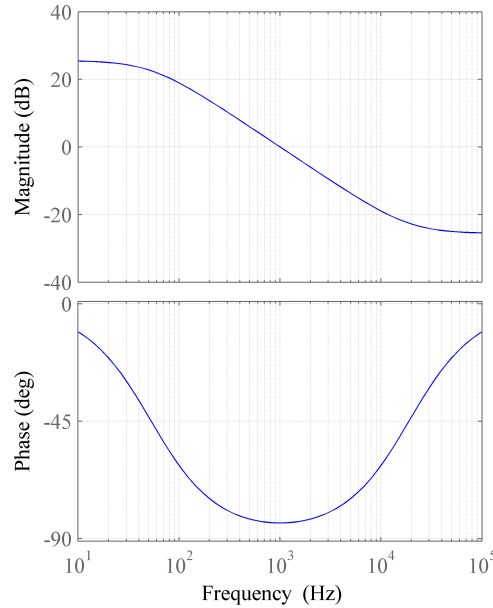
Reescribiendo 2.16 el dominio de Laplace queda de la siguiente forma:

$$\frac{k_f A s + 1}{k_f s + A} \quad (2.17)$$

El máximo valor del ancho de banda con el cual el FPAA AN231E04 puede procesar señales analógicas depende del reloj del sistema, el CAM y la etapa de entrada.

La frecuencia de entrada para la mayoría de los CAM está limitada aproximadamente a un valor menor a 2MHz debido a que el tipo de procesamiento de señal del CAM se basa en arquitecturas de datos muestreados [17], es por esto que se opta por realizar un escalamiento con un factor  $k_f = \frac{1}{2\pi 1000}$ .

El diagrama de Bode de la respuesta en frecuencia de la aproximación escalada para un integrador de orden 0.9 se muestra en la figura 2.2, las unidades de medición en frecuencia fueron transformadas de  $rad/s$  a  $Hz$  y el rango de operación abarca aproximadamente de  $100Hz$  a  $10kHz$ , este es un rango factible y se encuentra dentro del ancho de banda recomendado.



**Figura 2.2:** Diagrama de Bode graficado en MATLAB de aproximación de primer orden para el integrador de orden fraccionario escalado en frecuencia con un factor  $k_f = \frac{1}{2\pi 1000}$  y  $\alpha = 0.9$ .

## 2.4. Configuración de filtro bilineal

Para implementar físicamente la aproximación de primer orden del integrador de orden fraccionario se buscó dentro de la biblioteca de Anadigm el CAM cuyas características fueran factibles. El filtro bilineal es el CAM más versátil y apropiado para implementar el integrador de orden fraccionario, puede ser configurado como filtro pasa bajas, pasa altas, pasa todo o como filtro polo y cero.

Particularmente la opción de configuración mediante polos y ceros fue la razón por la que se utilizó este CAM. Cuando configuramos el filtro bilineal mediante polos y ceros este adopta internamente un circuito definido cuya función de transferencia se presenta en la ecuación 2.18.

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{G_h(s + 2\pi f_z)}{s + 2\pi f_p} \quad (2.18)$$

Su estructura es muy similar a la aproximación de primer orden del integrador de

orden fraccionario, por lo que procedemos a igualar ambas funciones de transferencia como se muestra en la ecuación 2.19. El signo negativo de la función de transferencia del filtro bilineal lo anulamos agregando a su salida un CAM inversor de ganancia el cual su función de transferencia es  $-G$ , que en este caso asignamos el valor  $G = 1$ .

$$\frac{k_f A s + 1}{k_f s + A} = \frac{G_h(s + 2\pi f_z)}{s + 2\pi f_p} \quad (2.19)$$

En la ecuación 2.19,  $f_p$  es la frecuencia del polo,  $f_z$  es la frecuencia del cero y  $G_h$  es la ganancia en alta frecuencia. El parámetro de ganancia en DC o ganancia en baja frecuencia ( $G_L$ ) está definido por el manual de Anadigm y se calcula con la ecuación 2.20.

$$G_L = \frac{f_z}{f_p} G_H \quad (2.20)$$

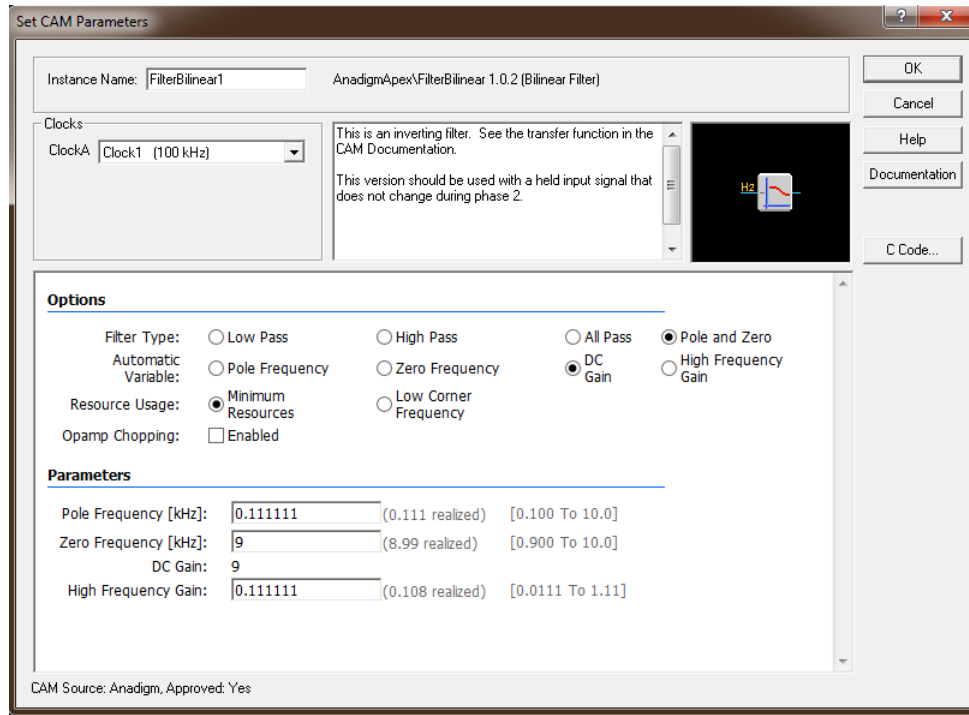
Mediante arreglos matemáticos moldeamos la función de transferencia de la aproximación del integrador de orden fraccionario buscando que se asemejara a la función de transferencia del filtro bilineal y de esta forma establecer relaciones para despejar los parámetros  $f_p$ ,  $f_z$  y  $G_H$ , este despeje nos dio las siguientes relaciones entre ambas funciones de transferencia:

$$f_p = \frac{A}{k_f 2\pi} \quad (2.21)$$

$$f_z = \frac{1}{k_f A 2\pi} \quad (2.22)$$

$$G_H = A \quad (2.23)$$

La ventana de configuración para este CAM en el software AnadigmDesigner2 se visualiza en la figura 2.3, en la parte inferior de la ventana se observa los cuatro parámetros necesarios para configurar este CAM.



**Figura 2.3:** Ventana de ajuste de parámetros de filtro bilineal

Se utilizó el software MATLAB para calcular los 4 valores necesarios para configurar el filtro bilineal, el programa en MATLAB se encuentra en el apéndice A.1. Utilizando un factor de escalamiento  $k_f = \frac{1}{2\pi 1000}$  y  $A = \frac{1-\alpha}{1+\alpha}$ , se calcularon los valores de frecuencia de polo, frecuencia de cero, ganancia en DC y ganancia en alta frecuencia, variando los valores de  $\alpha$  entre 0.1 y 0.95 en intervalos de 0.05, en la tabla 2.1 se presentan los valores obtenidos.

**Tabla 2.1:** Valores para configurar un filtro bilineal como integrador de orden fraccionario en el rango de órdenes de 0.1 a 0.95

Orden ( $\alpha$ )	Frecuencia de Polo [kHz]	Frecuencia de Zero [kHz]	Ganancia en DC	Ganancia en alta frecuencia
0.1	0.818182	1.22222	1.2222	0.818182
0.15	0.73913	1.35294	1.3529412	0.73913
0.2	0.666667	1.5	1.5	0.666667
0.25	0.6	1.66667	1.6666667	0.6
0.3	0.538462	1.85714	1.8571428	0.538462
0.35	0.481481	2.07692	2.0769231	0.481481
0.4	0.428571	2.33333	2.3333333	0.428571
0.45	0.37931	2.63636	2.6363636	0.37931
0.5	0.333333	3	3	0.333333
0.55	0.290323	3.44444	3.4444444	0.290323
0.6	0.25	4	4	0.25
0.65	0.212121	4.71429	4.7142857	0.212121
0.7	0.176471	5.66667	5.6666	0.176471
0.75	0.142857	7	7	0.142857
0.8	0.111111	9	9	0.111111
0.85	0.0810811	12.3333	12.333333	0.0810811
0.9	0.0526316	19	19	0.0526316
0.95	0.025641	39	39	0.025641

Los valores de la tabla 2.1 pueden ser ingresados en la ventana de configuración del CAM, sin embargo, como se muestra marcado por rojo no todos los valores son aceptados ya que estos valores dependen directamente de la frecuencia de reloj seleccionada para el CAM; por ejemplo, para este filtro bilineal, ver figura 2.3, la frecuencia del reloj se selecciona mediante CLOCK A.

En la ventana de configuración solo se pueden ingresar manualmente 3 de los 4 parámetros requeridos ya que el parámetro restante es calculado automáticamente por el sistema en base a los valores de los otros 3 parámetros. El usuario puede seleccionar el parámetro que se calculará automáticamente.

Para implementar el integrador de orden fraccionario en el filtro bilineal se opta por ingresar de forma manual los valores de frecuencia de polo, frecuencia de cero y ganancia en alta frecuencia mientras que la ganancia en DC se selecciona para ser calculada automáticamente por el software.

Los límites absolutos de los valores que podemos ingresar para las frecuencias del polo y cero son  $[\frac{F_c}{1000}, \frac{F_c}{10}]$ , el valor de la ganancia en alta frecuencia  $G_H$  debe estar en el rango  $[0.01\frac{V}{V}, 100\frac{V}{V}]$ , analizando la tabla 2.1 notamos que todos los valores a ingresar para la ganancia en alta frecuencia están dentro de este rango por lo que

este parámetro no genera mayor problema.

Enfocándonos únicamente en las frecuencias de polo y cero se procedió a hacer un análisis de las frecuencias de reloj con las cuales es posible implementar los diferentes órdenes del integrador fraccionario, para esto se generó una tabla con todas las frecuencias de reloj resultantes de la división de 16000.000000 KHz (system 1) entre el rango de 1 a 510, esta tabla se encuentra en el apéndice B.2.1.

Es importante mencionar que se eligió la frecuencia predeterminada de 16000.000000 KHz para los relojes de system 1 y system 2 para evitar problemas en el procesamiento de la señal analógica. Revisando la tabla generada se encontró lo siguiente:

- Para la implementación de integradores cuyos órdenes,  $\alpha$ , vayan del rango de 0.1 a 0.8 se puede utilizar un extenso número de frecuencias de reloj, sin embargo entre más se aproxima el orden del integrador a 0.8 el número de frecuencias que se pueden utilizar disminuye considerablemente, ver tabla B.1.
- El integrador de orden  $\alpha = 0.81$  es el orden más alto que puede ser implementado con el filtro bilineal, debido a que los órdenes superiores requieren frecuencias que se salen de los rangos permitidos, en la tabla 2.2 se muestran las frecuencias de reloj con las que es posible implementar este integrador en el filtro bilineal.

**Tabla 2.2:** Frecuencias de reloj con las cuales es posible configurar el CAM de filtro bilineal como integrador de orden 0.81

Frecuencia de reloj, $F_c$ [KHz]	Rango absoluto [KHz]
103.896104	[0.1038961 - 10.3896104]
102.564103	[0.1025641 - 10.2564103]
101.265823	[0.10126582 - 10.1265823]
100	[0.1 - 10]
98.7654321	[0.09876543 - 9.87654321]
97.5609756	[0.09756098 - 9.75609756]
96.3855422	[0.09638554 - 9.63855422]

- Para el integrador de orden  $\alpha = 0.82$  la frecuencia de corte del polo es de 0.0989011 KHz y la frecuencia de corte del cero es de 10.1111 KHz. El valor de la frecuencia del polo puede utilizarse para configurar el filtro bilineal con frecuencias de reloj menores a 98.7654321 KHz como se ve en la tabla 2.3; sin embargo, la frecuencia del cero tiene el valor de 10.1111kHz, este puede implementarse con frecuencias de reloj superiores a los 100 KHz, pero no con alguna frecuencia menor a esta. Al no coincidir los 2 parámetros en ninguna frecuencia de reloj el CAM no podrá configurarse con estos valores, por lo tanto, este orden no se puede implementar.

**Tabla 2.3:** Frecuencias de reloj con las cuales es posible configurar el CAM de filtro bilineal como integrador de orden 0.82

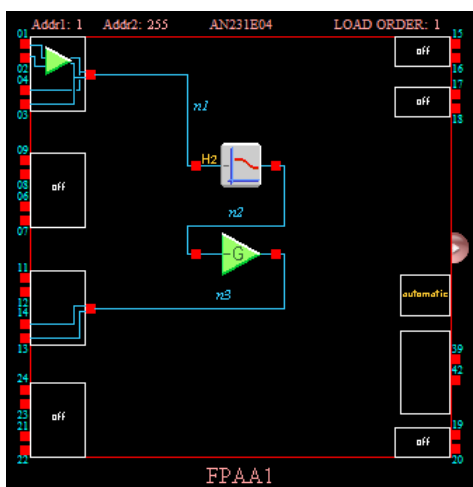
Frecuencia de reloj de polo [kHz]	Frecuencia de reloj de cero [kHz]
16000	16000 0
⋮	⋮
100	100
98.7654321	98.7654321
⋮	⋮
31.372549	31.372549

- Para órdenes superiores sucede el mismo caso, los dos parámetros de configuración no coinciden en alguna frecuencia de reloj por lo que estos órdenes ya no se pueden implementar en el filtro bilineal.

## 2.5. Diseño en software AnadigmDesigner2

En esta sección se describe el diseño y configuración del filtro bilineal como integrador de orden fraccionario en el software AnadigmDesigner2 para esto se toma como ejemplo el caso del integrador de orden 0.8 sin embargo esta implementación puede aplicarse a cualquiera de los órdenes permitidos, modificando los parámetros de configuración por los descritos en la tabla 2.1.

En la figura 3.1 se muestra una vista general del diseño del integrador de orden fraccionario, la configuración del CAM, los relojes y celdas IO se describirán en las subsecciones siguientes.

**Figura 2.4:** Diseño de integrador de orden fraccionario en AnadigmDesigner2

### 2.5.1. Configuración de Relojes

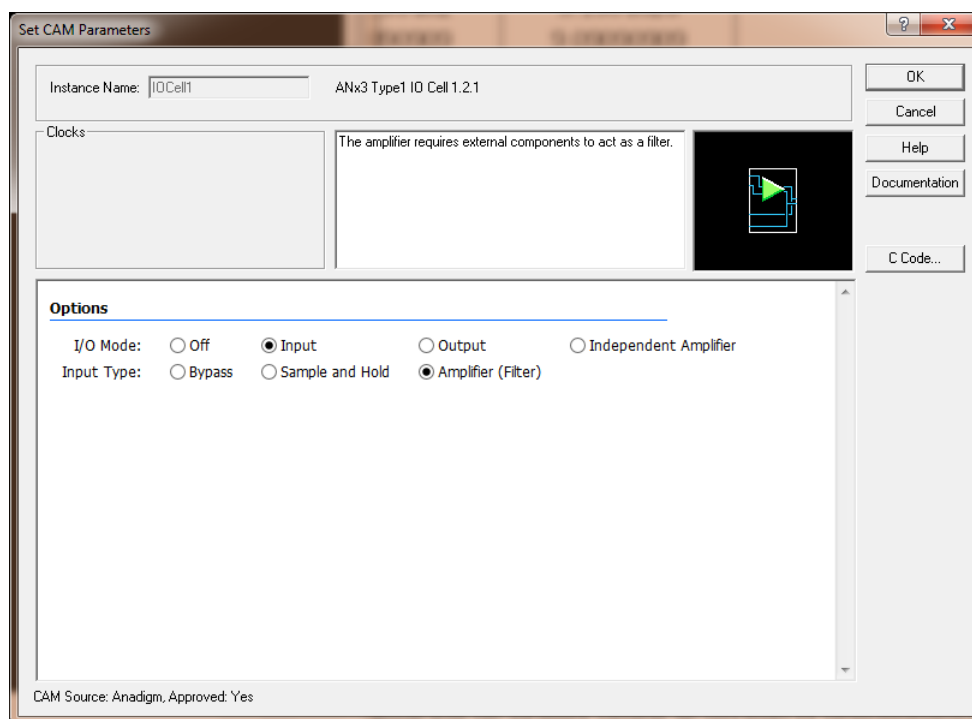
La configuración de los relojes para implementar la aproximación de primer orden de un integrador de orden fraccionario en el chip AN231E04 se resume en la tabla 2.4. Particularmente se seleccionó el reloj 1 con una frecuencia de 100 kHz ya que esta es la frecuencia más adecuada para implementar el integrador de orden  $\alpha = 0.8$  pero es posible utilizar cierto rango de frecuencias permitidas, como se ejemplifica en la tabla B.2, todas ellas pueden ser seleccionadas y configuradas en los relojes 0 al 3, los relojes 4 y 5 producen un retraso en fase por lo que para esta CAM no es recomendable utilizarlos.

**Tabla 2.4:** Relojes utilizados en la configuración del chip AN231E04

Master Clock-ACLK( $fc$ )16 MHz	System CLock ( $sys2=fc/1$ ) 16 MHz
System CLock ( $sys1=fc/1$ )16 MHz	
Clock 0 ( $sys1/320$ )50 kHz	Clock 1 ( $sys1/160$ )100 kHz
Clock 2 ( $sys1/32$ )500 kHz	Clock 3 ( $sys1/2$ )8 MHz
Clock 4 ( $sys1/80$ )50 kHz	Clock 5 ( $sys1/1$ )16 MHz

### 2.5.2. Configuración celdas IO

La celda IO 1 del chip se configuró en modo de entrada como tipo amplificador (filtro), debido a que en la tarjeta esta celda se encuentra conectada a la interfaz de entrada Raunch No. 1, la configuración de esta celda se observa en la figura 2.5.



**Figura 2.5:** Ventana de configuración celda IO 1.

La celda IO 3 del chip se configuró en modo de salida como tipo bypass, debido a que en la tarjeta esta celda está conectada a un buffer de salida, ver figura 2.6

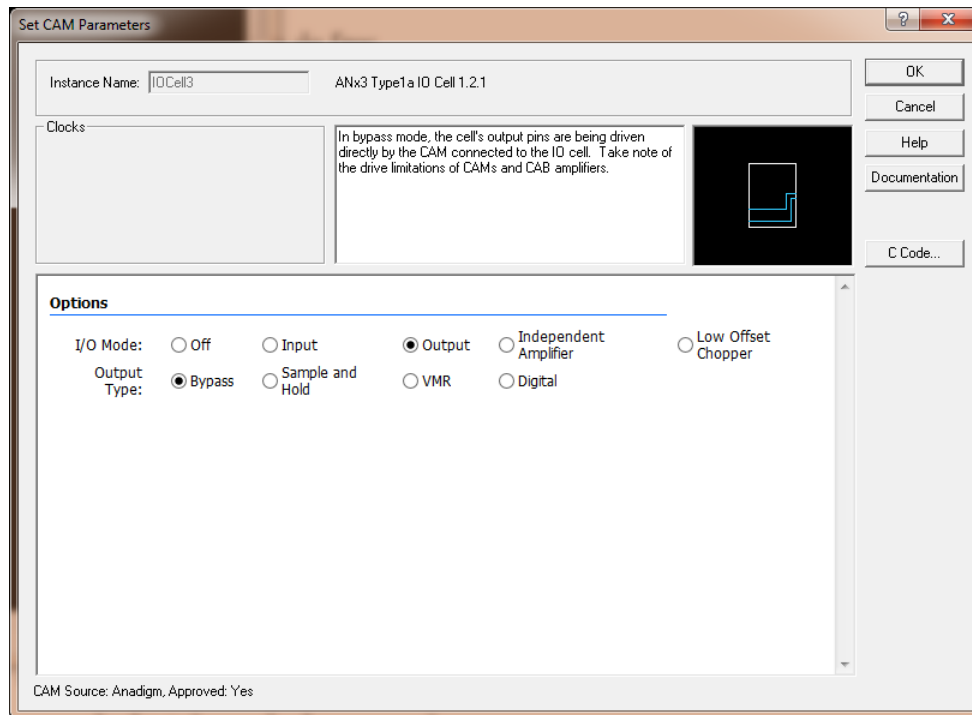


Figura 2.6: Ventana de configuración celda IO 3.

## 2.6. Caracterización del integrador de orden fraccionario

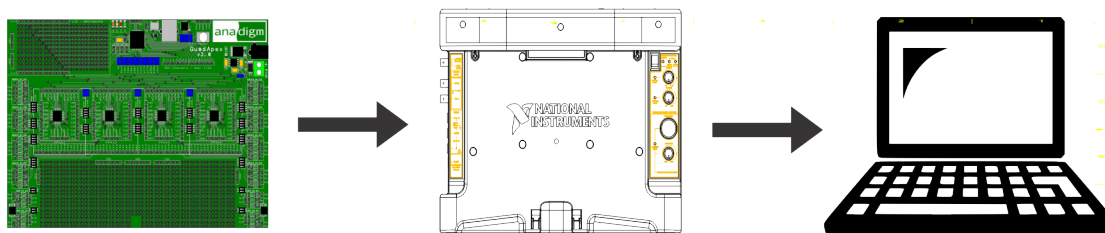
Para caracterizar el integrador de orden fraccionario se utilizó el laboratorio educativo de instrumentación virtual II (NI ELVIS II, por sus siglas en inglés) fabricado por la empresa National Instruments el cual utiliza instrumentos de software basados en LabVIEW, su software llamado NI ELVISmx cuenta con diversas funciones [18], entre ellas un analizador de Bode.

La función de analizador de Bode nos permitió verificar que el comportamiento del integrador de orden fraccionario fuera el esperado ya que pudimos visualizar su comportamiento en el dominio de la frecuencia. La función de analizador de Bode se configura a través de una interfaz gráfica, la configuración que se utilizó para caracterizar el integrador de orden 0.8 se presenta en la tabla 2.5.

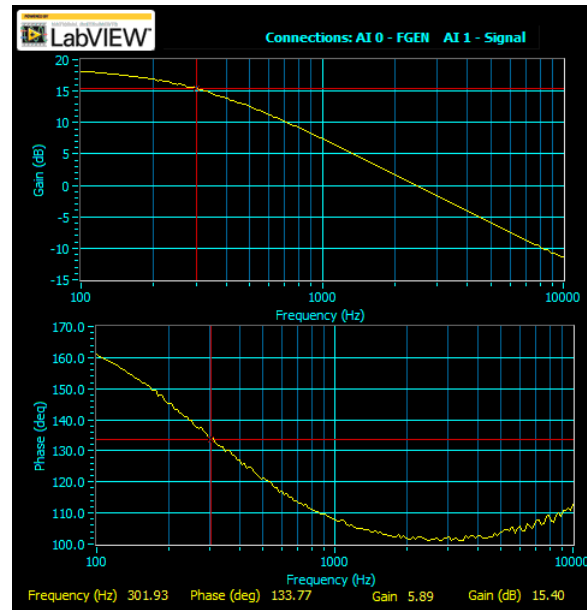
**Tabla 2.5:** Parámetros de configuración para función de analizador de Bode en NI ELVIS

Parámetro	Valor
Stimulus Chanel	AI 0
Response Chanel	AI 1
Start Frecuency	100.00 Hz
Stop Frecuency	10.00kHz
Steps (per decade)	100
Peak Amplitude	0.1 Volts
Op-Amp Signal Polarity	Normal
Mapping	Logarithmic

El esquema general de conexión entre la tarjeta Anadigm Quad Apex, la suite NI ELVIS II y ordenador, con el cual se configuraron ambos dispositivos, se muestra en la figura 3.4.

**Figura 2.7:** Esquema general de conexión

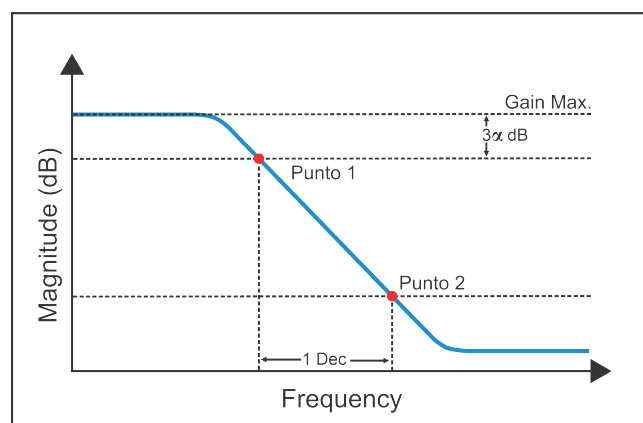
El analizador de Bode gráfica automáticamente cada uno de los datos obtenidos mostrando el resultado en su interfaz a través de las gráficas de magnitud y fase, ambas en función de la frecuencia en Hertz como se aprecia en la figura 2.8.



**Figura 2.8:** Diagrama de bode de integrador de orden fraccionario implementado en FPAA obtenido con NI ELVIS II.

La pendiente que presenta la gráfica en su región lineal es un valor importante que nos ayuda a verificar que el integrador de orden fraccionario implementado tiene el comportamiento esperado.

El procedimiento utilizado para calcular esta pendiente fue tomar como referencia la ganancia máxima (dB) del diagrama de magnitud de Bode y  $-3\alpha dB$  abajo tomar los datos del primer punto, posteriormente recorrerse una década abajo para tomar los datos del segundo punto, en caso de que este en una región curva desplazarse a la región lineal más próxima.



**Figura 2.9:** Puntos de muestra para calcular la pendiente.

En el diagrama de magnitud de Bode del integrador de orden fraccionario uti-

lizamos la ecuación de la pendiente para encontrar el valor de  $dB/dec$ .

$$Pendiente = \frac{\Delta Gain(dB)}{\Delta Dec} \quad (2.24)$$

Debido a que este diagrama se presenta en escala logarítmica utilizamos la ecuación 2.25 para determinar el número de décadas en la banda de  $f_1$  a  $f_2$  en el eje horizontal [19].

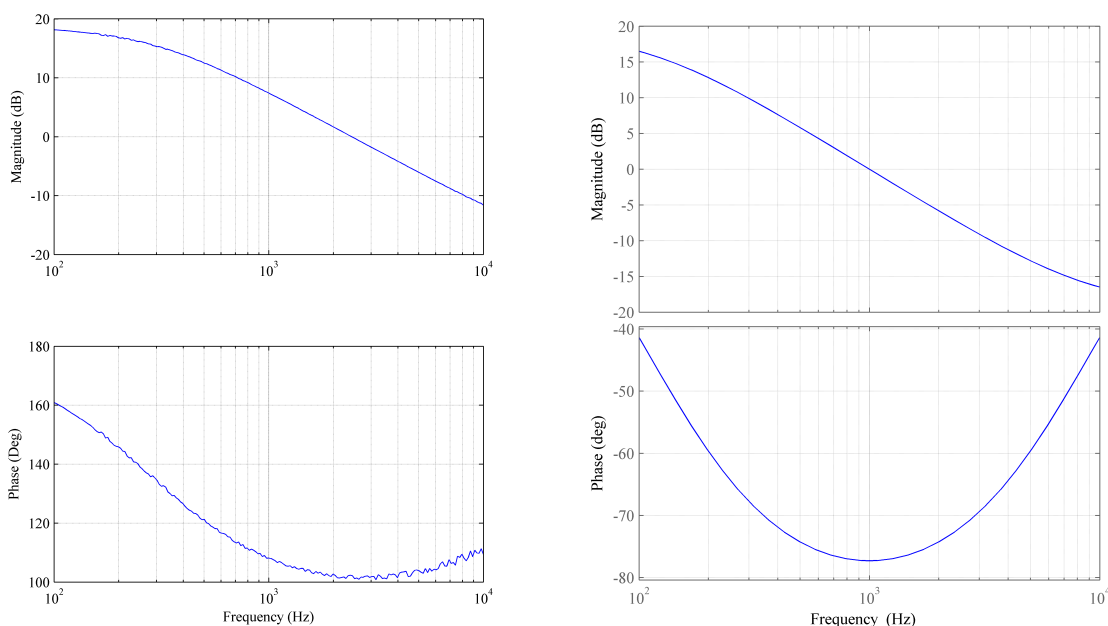
$$\Delta Dec = \log_{10}\left(\frac{f_2}{f_1}\right) \quad (2.25)$$

También es posible determinar el número de décadas entre dos frecuencias usando logaritmos naturales [20] mediante la ecuación 2.26.

$$\Delta Dec = \frac{\ln f_2 - \ln f_1}{\ln 10} \quad (2.26)$$

## 2.7. Análisis de gráficas

En la figura 2.10 se presenta una comparación entre el diagrama de bode del integrador de orden  $\alpha = 0.8$  obtenido experimentalmente y el diagrama de Bode simulado en MATLAB



(a) Diagrama de Bode de integrador de orden  $\alpha = 0.8$  implementado en FPAA.

(b) Diagrama de Bode de integrador de orden  $\alpha = 0.8$  simulado en MATLAB.

**Figura 2.10:** Comparación entre el diagrama de Bode obtenido de forma experimental y el diagrama de Bode simulado en Matlab

En el integrador de orden  $\alpha = 0.8$  implementado en FPAA y caracterizado experimentalmente se midió una pendiente de **-17.228488 dB/Dec** mientras que en la simulación de Matlab se midió una pendiente de **-17 dB/Dec**.

Aunque ambos resultados son muy cercanos entre sí, presentan un error con respecto a la pendiente ideal que se espera para este integrador la cual debe de ser de **-16 dB/Dec**.

El CAM de filtro bilineal configurado como filtro polo y cero nos permite implementar integradores de órdenes que están dentro del rango de  $[0.1, 0.81]$ , integradores de órdenes superiores a este rango ya no son realizables debido a que la frecuencia del reloj seleccionado para el CAM permite ingresar solamente un rango definido de valores; sin embargo, una ventaja de esta topología es que no utiliza muchos recursos del FPAA por lo que en aplicaciones que requieran integradores de órdenes  $\alpha$  menores a 0.81 es viable su uso.

## Capítulo 3

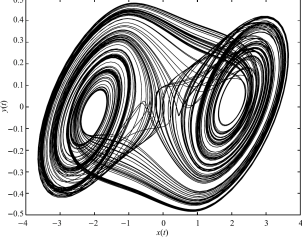
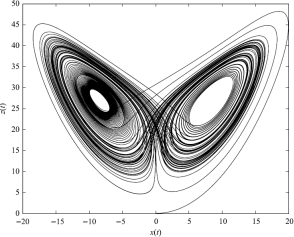
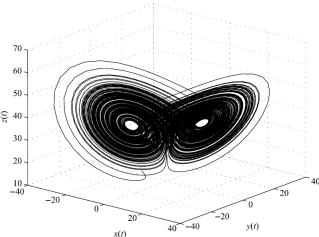
# Implementación de integrador de orden fraccionario utilizando filtros pasa bajas y pasa altas

El caos es común en sistemas dinámicos no lineales de orden entero, y su mecanismo de generación ha sido profundamente investigado a través de teorías matemáticas y simulaciones numéricas. Recientemente, la teoría del cálculo de orden fraccionario ha sido correctamente desarrollada junto con su principio de operación [21].

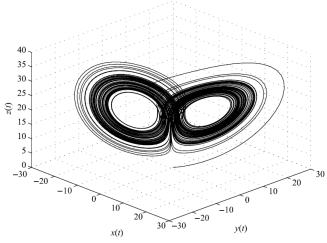
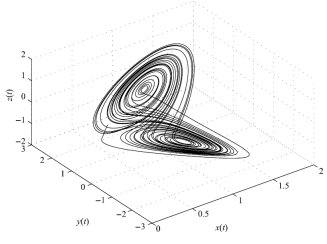
La teoría del cálculo de orden fraccionario puede aplicarse a sistemas dinámicos no lineales, recientemente se ha investigado su aplicación en sistemas dinámicos típicos como el sistema de orden fraccionario de Lorenz [22], el sistema de orden fraccionario de Chen [23, 24], el sistema de orden fraccionario de Lü [25] y algunos otros sistemas caóticos o hipercaóticos de orden fraccionario.

En base a las implementaciones analógicas reportadas en la literatura se ha encontrado que la mayoría de sistemas caóticos de orden fraccionario y particularmente el sistema de orden fraccionario de Lü funcionan en órdenes cercanos o mayores a 0.9, en la tabla 3.1 se presenta de forma resumida algunos ejemplos retomados de [26].

**Tabla 3.1:** Sistemas Caóticos de Orden Fraccionario

Nombre	Variables de estado del sistema	Parámetros utilizados	Resultado de la simulación
<p>Oscilador caótico de Chua de orden fraccionario (Petráš, 2008)</p>	${}_0D_t^{q_1}x(t) = \alpha(y(t) - x(t) - f(x)),$ ${}_0D_t^{q_2}y(t) = x(t) - y(t) + z(t),$ ${}_0D_t^{q_3}z(t) = \beta y(t) - \gamma z(t),$ <p>donde <math>f(x)</math> es la función no lineal a trozos.</p>	<p>El orden fraccionario total del atractor es <math>q = 2.9</math> para los parámetros <math>\alpha = 10.1911</math>, <math>\beta = 10.3035</math>, <math>\gamma = 0.1631</math>, <math>q_1 = q_2 = 0.98</math>, <math>q_3 = 0.94</math></p>	
<p>Oscilador caótico de Lorenz de orden fraccionario (Li and Yan, 2007)</p>	${}_0D_t^{q_1}x(t) = \sigma(y(t) - x(t)),$ ${}_0D_t^{q_2}y(t) = x(t)(\rho - z(t)) - y(t),$ ${}_0D_t^{q_3}z(t) = x(t)y(t) - \beta z(t),$ <p>donde <math>q_1, q_2</math> y <math>q_3</math> son órdenes de las derivadas.</p>	<p><math>\sigma = 10, \rho = 28, \beta = 8/3, q_1 = q_2 = q_3 = 0.995</math>, y condiciones iniciales <math>(x(0), y(0), z(0)) = (0.1, 0.1, 0.1)</math>.</p>	
<p>Oscilador caótico de Chen de orden fraccionario (Lu and Chen, 2006)</p>	${}_0D_t^{q_1}x(t) = a(y(t) - x(t)),$ ${}_0D_t^{q_2}y(t) = (c - a)x(t) - x(t)z(t) + cy(t),$ ${}_0D_t^{q_3}z(t) = x(t)y(t) - bz(t),$ <p>donde <math>0 &lt; q_1, q_2, q_3 \leq 1</math> su orden total esta denotado por <math>\bar{q} = (q_1, q_2, q_3)</math>.</p>	<p><math>a = 35, b = 3, c = 28, d = -7</math>, los órdenes <math>q_1 = q_2 = q_3 = 0.9</math>, y condiciones iniciales <math>(x(0), y(0), z(0)) = (-9, -5, 14)</math>. Para este sistema el mínimo orden aceptable es <math>q &gt; 0.8244</math> si consideremos <math>q_1 = q_2 = q_3 = q</math></p>	

Continua en la página siguiente.

Nombre	Variables de estado del sistema	Parámetros utilizados	Resultado de la simulación
<p>Oscilador caótico de Lü de orden fraccionario (Deng and Li, 2005)</p>	$\begin{aligned} {}_0D_t^{q_1}x(t) &= a(y(t) - x(t)), \\ {}_0D_t^{q_2}y(t) &= -x(t)z(t) + cy(t), \\ {}_0D_t^{q_3}z(t) &= x(t)y(t) - bz(t), \end{aligned}$ <p>donde <math>0 &lt; q_1, q_2, q_3 \leq 1</math> son órdenes de las derivadas y <math>a, b, c</math> son parámetros del sistema.</p>	<p><math>a = 36, b = 3, c = 20</math> y los órdenes <math>q_1 = q_2 = q_3 = 0.95</math>. El mínimo orden aceptable para que este sistema presente comportamiento caótico es <math>q &gt; 0.9156</math>.</p>	
<p>Oscilador Caótico de Liu de orden fraccionario (Gejji and Bhalekar, 2010)</p>	$\begin{aligned} {}_0D_t^{q_1}x(t) &= -ax(t) - ey^2(t), \\ {}_0D_t^{q_2}y(t) &= by(t) - kx(t)z(t), \\ {}_0D_t^{q_3}z(t) &= -cz(t) + mx(t)y(t), \end{aligned}$ <p>donde <math>q_1, q_2, q_3</math> su orden total esta denotado por <math>\bar{q} = (q_1, q_2, q_3)</math>.</p>	<p><math>a = e = 1, b = 2.5, k = m = 4, c = 5</math> y los órdenes <math>q_1 = q_2 = q_3 = 0.95</math> y condiciones iniciales <math>(x(0), y(0), z(0)) = (0.2, 0, 0.5)</math>. El mínimo orden <math>q</math> donde se presenta un comportamiento caótico es <math>q &gt; 0.916</math>.</p>	

Fin de la tabla

Debido a la limitante del filtro bilineal configurado como filtro polo y cero de solo poder implementar el integrador de orden fraccionario con un orden máximo de  $\alpha = 0.81$  y a que la mayoría de los osciladores caóticos de orden fraccionario, como se muestra en la tabla 3.1, requieren órdenes  $\alpha > 0.8$  para presentar un comportamiento caótico, se buscó una forma alternativa de implementación. La solución encontrada para resolver este problema fue separar los numeradores de la aproximación de primer orden para un integrador de orden fraccionario y así expresarla como la suma de dos

fracciones, como se muestra en la ecuación 3.1.

$$\frac{k_f A s + 1}{k_f s + A} = \frac{k_f A s}{k_f s + A} + \frac{1}{k_f s + A} \quad (3.1)$$

Analizando las dos fracciones resultantes podemos notar que tienen la estructura de las funciones de transferencia de primer orden de un filtro pasa-bajas y pasa-altas. Normalmente un filtro pasa-bajas tiene sus ceros localizados en el infinito, por esta razón los polinomios del numerador son de grado cero por lo que la forma de su función de transferencia es como la ecuación 3.2; por otro lado, un filtro pasa-altas tiene sus ceros localizados en el origen del plano de la frecuencia compleja, por lo que su función de transferencia tiene la forma de la ecuación 3.3.

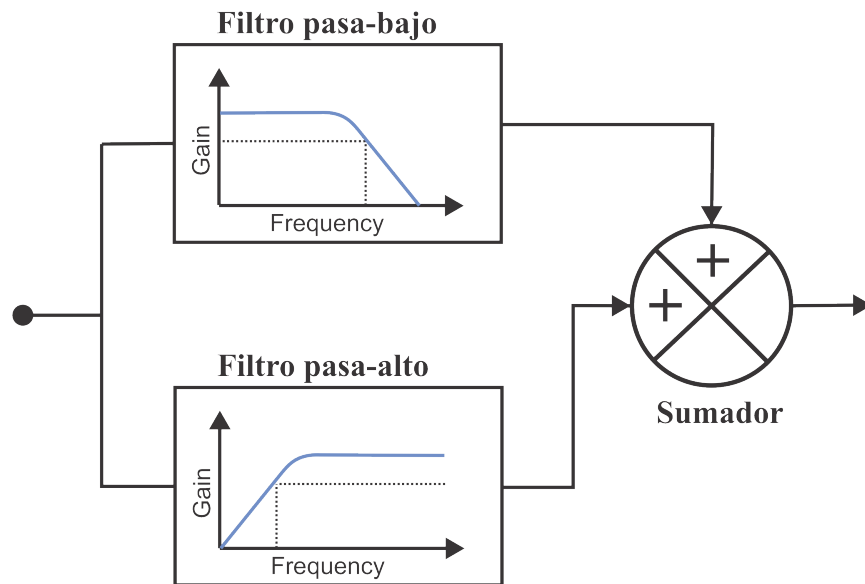
■ **Filtro pasa bajas**

$$\frac{1}{k_f s + A} \quad (3.2)$$

■ **Filtro pasa altas**

$$\frac{k_f A s}{k_f s + A} \quad (3.3)$$

La representación en bloques de la expresión 3.1 para la aproximación del integrador de orden fraccionario es la de la figura 3.1



**Figura 3.1:** Diagrama de bloques de implementación de integrador de orden fraccionario utilizando filtro pasa-bajo y pasa-alto.

---

### 3.1. Implementación en FPAA

El filtro bilineal prediseñado por Anadigm tiene la particularidad de poder ser configurado como filtro pasada bajas, pasa altas, pasa todo o polos y ceros, en esta sección se hace una descripción de la configuración que se realizó en el filtro bilineal para implementarlo como filtro pasa bajas y pasa altas, también se describe el CAM de suma diferencial que realiza la suma de las señales de ambos filtros, todos ellos utilizados para la implementación del integrador de orden fraccionario.

#### 3.1.1. Filtro bilineal configurado como pasa-bajas

La función de transferencia del filtro bilineal configurado como pasa bajas se presenta en la ecuación 3.4. Cuando se configura el filtro bilineal como filtro pasa-bajas se puede seleccionar la polaridad que se desea que tenga, en este caso seleccionamos trabajar este filtro con polaridad invertida.

$$\frac{V_{out}(s)}{V_{in}(s)} = \pm \frac{2\pi f_0 G}{s + 2\pi f_0} \quad (3.4)$$

Para implementar el filtro pasa bajas, obtenido al separar la aproximación de primer orden para un integrador de orden fraccionario en la suma de dos fracciones, en el filtro bilineal se igualaron ambas funciones de transferencia como se presenta en la ecuación 3.5, posteriormente se procedió a despejar  $f_0$  y  $G$ . El signo menos se omitió ya que este se elimina cuando la señal llega a la entrada inversora del CAM de suma diferencial.

$$\frac{1}{k_f s + A} = \frac{2\pi f_0 G}{s + 2\pi f_0} \quad (3.5)$$

El cuadro de configuración del filtro bilineal requiere una frecuencia de corte y una ganancia para poder implementar el filtro pasa-bajas por lo que a partir de la ecuación 3.5 se realizaron despejes matemáticos que nos llevaron a las siguientes relaciones:

- **Frecuencia de corte**

$$f_0 = \frac{A}{k_f 2\pi} \quad (3.6)$$

- **Ganancia**

$$G = \frac{1}{k_f 2\pi f_0} \quad (3.7)$$

### 3.1.2. Filtro bilineal configurado como pasa-altas

La función de transferencia del filtro bilineal configurado como pasa altas se presenta en la ecuación 3.8.

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{Gs}{s + 2\pi f_0} \quad (3.8)$$

Para poder implementar el filtro pasaba altas en el filtro bilineal se estableció la igualdad de ambas funciones de transferencia como se presentada en la ecuación 3.9. Nuevamente el signo menos se omite ya que este se elimina cuando la señal del filtro llega a la entrada inversora del CAM de suma diferencial.

$$\frac{k_f A s}{k_f s + A} = \frac{Gs}{s + 2\pi f_0} \quad (3.9)$$

La ventana de configuración del filtro bilineal requiere una frecuencia de corte y una ganancia para poder implementar este filtro por lo que a partir de la ecuación 3.10 se realizaron despejes matemáticos que nos llevaron a las siguientes relaciones:

- **Frecuencia de corte**

$$f_0 = \frac{A}{k_f 2\pi} \quad (3.10)$$

- **Ganancia**

$$G = A \quad (3.11)$$

### 3.1.3. Suma diferencial

El CAM de la suma diferencial de manera predeterminada tiene dos entradas, sin embargo, en su ventana de configuración puede activarse una tercera entrada, todas sus entradas son de polaridad invertida. La función de transferencia para este CAM es 3.12

$$V_{out} = -G_1 V_{input1} - G_2 V_{input2} \quad (3.12)$$

Las variables  $G$  numeradas en la ecuación 3.12 son las Ganancias de las diversas ramas de entrada y las variables  $V_{Input}$  numeradas son los voltajes de entrada en las distintas ramas. Para el caso de 3 entradas la función de transferencia queda expresada como 3.13

$$V_{Out} = -G_1 V_{input1} - G_2 V_{input2} - G_3 V_{input3} \quad (3.13)$$

### 3.2. Valores de implementación

Utilizando el software MATLAB para realizar el cálculo de los valores de la frecuencia de corte y ganancia para los filtros pasa bajas y pasa altas. Para esta implementación se utilizó el mismo factor de escalamiento que para el filtro bilineal,  $k_f = \frac{1}{2\pi 1000}$ , se realizó una variación de los valores de  $\alpha$  entre 0.1 y 0.9 en intervalos de 0.05 y posteriormente en intervalos de 0.01 de 0.9 a 0.99), se analizó de forma más detallada este rango ya que es el más utilizado en la implementación de osciladores caóticos de orden fraccionario, en la tabla 3.2 se presentan los valores obtenidos.

**Tabla 3.2:** Valores de configuración para filtro paso-alto y pasa-bajo para implementar integrador de orden fraccionario

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.01	0.980198	1.0202	0.980198	0.980198
0.1	0.818182	1.22222	0.818182	0.818182
0.15	0.73913	1.35294	0.73913	0.73913
0.2	0.666667	1.5	0.666667	0.666667
0.25	0.6	1.66667	0.6	0.6
0.3	0.538462	1.85714	0.538462	0.538462
0.35	0.481481	2.07692	0.481481	0.481481
0.4	0.428571	2.33333	0.428571	0.428571
0.45	0.37931	2.63636	0.37931	0.37931
0.5	0.333333	3	0.333333	0.333333
0.55	0.290323	3.44444	0.290323	0.290323
0.6	0.25	4	0.25	0.25
0.65	0.212121	4.71429	0.212121	0.212121
0.7	0.176471	5.66667	0.176471	0.176471
0.75	0.142857	7	0.142857	0.142857
0.8	0.111111	9	0.111111	0.111111
0.85	0.0810811	12.3333	0.0810811	0.0810811
0.9	0.0526316	19	0.0526316	0.0526316
0.91	0.0471204	21.2222	0.0471204	0.0471204
0.92	0.0416667	24	0.0416667	0.0416667
0.93	0.0362694	27.5714	0.0362694	0.0362694
0.94	0.0309278	32.3333	0.0309278	0.0309278
0.95	0.025641	39	0.025641	0.025641

Continua en la página siguiente.

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.96	0.0204082	49	0.0204082	0.0204082
0.97	0.0152284	65.6667	0.0152284	0.0152284
0.98	0.010101	99	0.010101	0.010101
0.99	0.00502513	199	0.00502513	0.0050251

Fin de la tabla

La frecuencia de reloj es un parámetro fundamental en la función de los circuitos analógicos de condensadores conmutados. El uso de dos relojes base no relacionados también puede ser un problema ya que es poco probable que una señal analógica se mueva de un dominio de reloj de capacitor conmutado a otro sin presentar daños [27].

Con el objetivo de evitar efectos adversos causados por utilizar diferentes relojes se optó por utilizar la misma frecuencia de reloj para todos los CAM utilizados en el chip. Satisfactoriamente se encontró que con esta nueva forma de implementación del integrador de orden fraccionario podemos utilizar  $\alpha$  con órdenes de incluso 0.99; sin embargo, cada orden requiere configuraciones específicas en las siguientes subsecciones se describe de forma más detallada.

### 3.2.1. Órdenes 0.01 a 0.9

Los parámetros de configuración presentados en la tabla 3.2 para el rango de órdenes [0.01, 0.9] pueden ser ingresados en la ventana de configuración del filtro pasa-bajas y pasa-altas de forma directa utilizando los relojes de sistema 1 y 2 configurados a la frecuencia predeterminada de 16000.000000 KHz, el reloj A puede adoptar todos los valores presentados en la tabla 3.2.

Los límites absolutos para el parámetro de ganancia en el filtro bilineal configurado como pasa bajas o pasa altas son  $[0.011\frac{V}{V}, 20.0\frac{V}{V}]$ , analizando la tabla 3.2 se verificó que todos los valores de ganancia están dentro de este rango por lo que este parámetro no genera mayor problema.

### 3.2.2. Valores de implementación y restricciones órdenes 0.91 a 0.93

Como se muestra en la tabla 3.2 la configuración del filtro pasa-bajo para los órdenes 0.91 a 0.93 presenta la problemática de que las ganancias son mayores a 20, el cual es el límite máximo de este parámetro, por lo que para implementarlo se hará uso del CAM de suma de la etapa de inversión. A continuación, se describe la metodología utilizada en cada caso:

**Tabla 3.3:** Valores para configurar un filtro pasa bajas en órdenes 0.91 a 0.93

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.91	0.0471204	21.2222	0.0471204	0.0471204
0.92	0.0416667	24	0.0416667	0.0416667
0.93	0.0362694	27.5714	0.0362694	0.0362694

- **Orden 0.91**, El valor de la ganancia para el filtro pasa-bajo es 21.222, la solución que se propuso es dividir la ganancia entre dos quedando con un valor de 10.6111, externamente en el CAM de Suma de la etapa de inversión modificamos la ganancia donde llega la señal de voltaje de este filtro en un factor de 2. De esta forma se mantiene la ganancia original.
- **Orden 0.92**, La implementación de este orden requiere una ganancia de 24 para el filtro pasa bajas, la metodología fue la misma que el caso anterior, se configuró el CAM del filtro pasa-bajo con una ganancia de 12 y en el CAM de suma de etapa de inversión se multiplicó la señal de entrada por un factor de 2, manteniendo la ganancia original
- **Orden 0.93** El valor de la ganancia para el filtro pasa-bajo es 27.5714, por lo que la metodología para implementarlo es la misma que los dos casos anteriores.

### 3.2.3. Valores de implementación y restricciones órdenes 0.94 a 0.96

Para este rango el reloj de 16MHz no permite ingresar valores de frecuencia de corte, por lo que se utilizó un reloj menor de 8MHz, realizando este cambio se puede seleccionar frecuencias de reloj más pequeñas para el reloj del CAM, reloj A, lo que nos permite ingresar valores más pequeños como en este caso.

**Tabla 3.4:** Valores para configurar un filtro pasa bajas en órdenes 0.91 a 0.93

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.94	0.0309278	32.3333	0.0309278	0.0309278
0.95	0.025641	39	0.025641	0.025641
0.96	0.0204082	49	0.0204082	0.0204082

### 3.2.4. Valores de implementación y restricciones órdenes 0.97 a 0.98

Las frecuencias mínimas de reloj A que se obtienen con un reloj de sistemas de 8Mhz no permiten ingresar los valores de frecuencia de corte que requieren los filtros

pasa-bajo y pasa-alto para implementar el integrador con estos órdenes por lo que se procedió a configurar el reloj del sistema a una frecuencia de 4 MHz.

**Tabla 3.5:** Valores para configurar un filtro pasa bajas en órdenes 0.91 a 0.93

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.97	0.0152284	65.6667	0.0152284	0.0152284
0.98	0.010101	99	0.010101	0.010101

### 3.2.5. Valores de implementación y restricciones órdenes 0.99

En este caso la frecuencia de corte para el filtro pasa-bajo y pasa-alto es muy baja por lo que se seleccionó el reloj de 2 MHz; sin embargo, frecuencias de reloj menores a esta pueden ocasionar fallas en el comportamiento del integrador.

**Tabla 3.6:** Valores para configurar un filtro pasa bajas en órdenes 0.99

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.99	0.00502513	199	0.00502513	0.0050251

## 3.3. Diseño en software AnadigmDesigner2

En la presente sección se describe la implementación del integrador de orden fraccionario utilizando filtro pasa-bajo y filtro pasa-alto, esta implementación presenta mayor versatilidad para la implementación de órdenes superiores a 0.9. La descripción se hace tomando como ejemplo el integrador de orden 0.9; sin embargo, el diseño es configurable para los órdenes descritos en la sección anterior, tomando en cuenta las restricciones mencionadas.

En la figura 3.2 se muestra una vista general del diseño del integrador de orden fraccionario implementado con filtro pasa-alto y pasa-bajo sumados por el CAM de suma de la etapa de inversión, la configuración de los CAM, los relojes y celdas IO se describirán en las subsecciones siguientes.

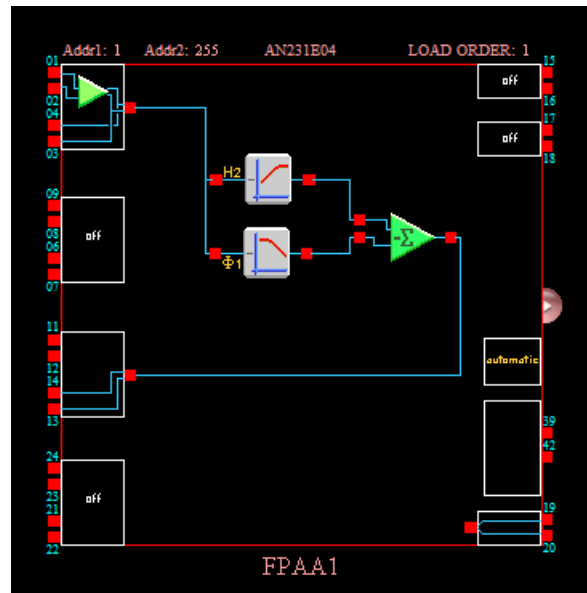


Figura 3.2: Diseño de integrador de orden fraccionario implementado con filtros pasa-bajas y pasa-alta en AnadigmDesigner2

### 3.3.1. Configuración de Relojes

Tabla 3.7: Relojes utilizados en implementación de integrador de orden  $\alpha = 0.9$  utilizando filtro pasa-bajo y filtro pasa-alto

Master Clock-ACLK( $fc$ ) 16 MHz	System CLock ( $sys2=fc/1$ ) 16 MHz
System CLock ( $sys1=fc/1$ ) 16 MHz	
Clock 0 ( $sys1/4$ ) 4 MHz	Clock 1 ( $sys1/20$ ) 800 kHz
Clock 2 ( $sys1/320$ ) 50 kHz	Clock 3 ( $sys1/500$ ) 32 kHz
Clock 4 ( $sys1/1$ ) 16 kHz	Clock 5 ( $sys1/1$ ) 16 MHz

### 3.3.2. Configuración de Celdas IO

Al igual que la implementación del integrador de orden fraccionario utilizando el filtro bilineal configurado mediante polos y ceros en el capítulo 2, la configuración de las celdas IO1 e IO3 fue la misma, IO1 fue configurada en modo de entrada con el propósito de utilizar el raunch 1 integrado en la tarjeta, la celda IO3 fue configurada como salida bypass, las demás celdas no son necesarias por lo que se desactivan.

### 3.3.3. Configuración de CAMs

En la figura 3.3 se resume la configuración de cada uno de los CAMs utilizados en el diseño en AnadigmDesigner2.

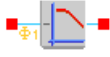

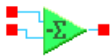
Name	Options	Parameters	Clocks
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  Anadigm (Approved)	<b>Filter Type</b> <i>Low Pass</i> <b>Input Sampling Phase</b> <i>Phase 1</i> <b>Polarity</b> <i>Inverting</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>19.0</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 2)</i>
<b>FilterBilinear2</b> (FilterBilinear v1.0.2)  Anadigm (Approved)	<b>Filter Type</b> <i>High Pass</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>0.0528</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 2)</i>
<b>SumInv1</b> (SumInv v1.0.1)  Anadigm (Approved)	<b>Input 3</b> <i>Off</i>	<b>Gain 1 (UpperInput)</b> <i>1.00</i> <b>Gain 2 (LowerInput)</b> <i>1.00</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 2)</i>

Figura 3.3: Configuración de CAMs

### 3.4. Caracterización del integrador de orden fraccionario implementado con filtros pasa bajas y pasa altas

El procedimiento para caracterizar el integrador de orden fraccionario implementado con filtro pasa-bajo y paso-alto fue el mismo que se utilizó para el integrador de orden fraccionario implementado en filtro bilineal, descrito en el capítulo 2. Se utilizó la suite NI ELVIS II para obtener el diagrama de Bode de este integrador y calcular su pendiente.

En la tabla 3.8 se presentan la recopilación de las pendientes obtenidas en una serie de pruebas desde  $\alpha = 0.1$  hasta  $\alpha = 0.99$ , en la última columna del lado derecho de la tabla también se presenta el error relativo del valor de la pendiente obtenida con respecto al ideal.

Tabla 3.8: Pendientes [dB/Dec] obtenidas de la implementación de la aproximación  $\frac{As+1}{s+A}$  con  $A = \frac{1-\alpha}{1+\alpha}$  usando filtros pasa-bajas y pasa-altas

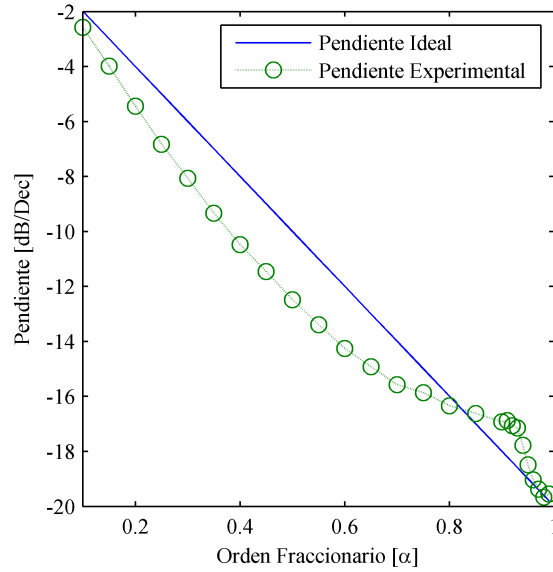
Order $\alpha$	Gain Max [dB]	Punto 1		Punto 2		Pendiente [dB/Dec]	Error absoluto [dB]
		Frec. P1 [Hz]	Gain P1 [dB]	Frec. P2 [Hz]	Gain P2 [dB]		
0.1	1.52	831.67	1.22	3890.51	-0.51	-2.58187568	0.5818757
0.15	2.47	776.16	2.01	4073.79	-0.86	-3.985848817	0.9858488
0.2	3.31	758.66	2.71	3715.42	-1.05	-5.449585532	1.4495855
0.25	4.23	724.38	3.49	4265.83	-1.77	-6.83083977	1.8308398

Continua en la página siguiente.

Order $\alpha$	Gain Max [dB]	Punto 1		Punto 2		Pendiente [dB/Dec]	Error absoluto [dB]
		Frec. P1 [Hz]	Gain P1 [dB]	Frec. P2 [Hz]	Gain P2 [dB]		
0.3	5.16	691.79	4.27	4168.79	-2.02	-8.063733095	2.0637331
0.35	6.14	660.68	5.09	4265.83	-2.47	-9.333190639	2.3331906
0.4	7.16	616.54	5.98	4265.83	-2.82	-10.47566313	2.4756631
0.45	8.21	588.78	6.85	4677.29	-3.47	-11.46614115	2.4661412
0.5	9.34	549.48	7.82	4897.83	-4.04	-12.48352962	2.4835296
0.55	10.55	501.24	8.88	5011.82	-4.51	-13.39067297	2.390673
0.6	11.77	457.09	9.99	4570.93	-4.27	-14.25995935	2.2599594
0.65	13.28	407.36	11.29	4073.79	-3.63	-14.91969779	1.9196978
0.7	14.75	363.03	12.6	3630.85	-2.98	-15.57897503	1.578975
0.75	16.65	295.04	14.43	2951.7	-1.44	-15.8669644	0.8669644
0.8	18.7	251.27	16.23	2511.96	-0.11	-16.34209048	0.3420905
0.85	21.51	190.55	18.94	1905.49	2.32	-16.62003788	-0.379962
0.9	25.19	128.9	22.44	1288.21	5.52	-16.92450617	-1.075494
0.91	26.07	114.74	23.34	1148.13	6.45	-16.88533594	-1.314664
0.92	27.19	104.68	24.42	1047.18	7.35	-17.06730976	-1.33269
0.93	28.43	93.32	25.67	933.19	8.52	-17.15007981	-1.44992
0.94	29.06	93.32	26.19	933.19	8.41	-17.78008275	-1.019917
0.95	29.52	97.79	26.65	977.33	8.16	-18.49468315	-0.505317
0.96	29.8	100.02	26.87	1000.05	7.83	-19.04124027	-0.15876
0.97	30.02	102.26	27.08	1023.34	7.7	-19.37391346	-0.026087
0.98	30.15	104.68	27.23	1047.18	7.56	-19.6669	0.0669
0.99	30.25	104.68	27.36	1047.18	7.81	-19.54691891	-0.253081

Fin de la tabla

Como se presenta en la tabla 3.8 la gran mayoría de los integradores implementados presentan errores en algunos casos mayores a 2 dB/Dec. En la figura 3.4 se presenta una gráfica comparativa de las pendientes ideal y experimental en función del orden, ejemplificando de una forma más clara el comportamiento que tuvo el integrador de orden fraccionario, cabe mencionar que el comportamiento ideal es de tipo lineal.



**Figura 3.4:** Gráfica comparativa entre valores de pendiente ideal y valores de pendiente obtenidos experimentalmente.

Debido a que la implementación del integrador de orden fraccionario con la aproximación de primer orden tiene un porcentaje de error grande en las pendientes medidas con respecto al valor ideal se procede a buscar un método de ajuste que ayude a reducir el error y así optimizar la implementación del integrador.

De forma empírica se realizaron diversas pruebas variando el valor de A buscando que la pendiente medida sea lo más cercana a la ideal. La tabla 3.9 muestra los resultados obtenidos.  $\hat{A}$  es el valor con el que se configuró el integrador de orden fraccionario para que su respuesta en frecuencia nos de la pendiente deseada.

**Tabla 3.9:** Tabla recopilación datos  $\hat{A}$

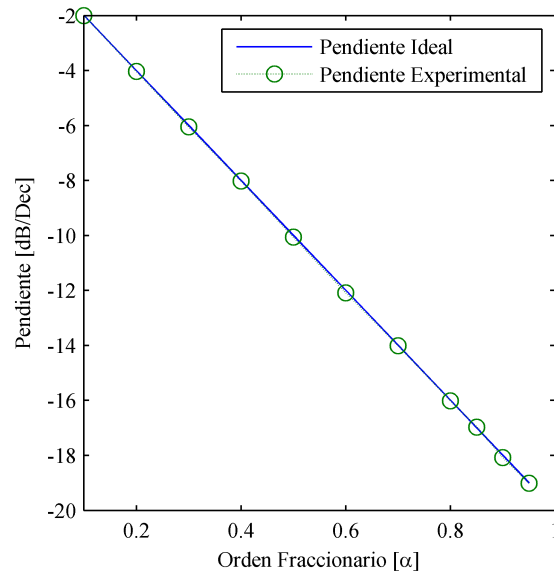
$\alpha$	$\hat{A}$	Gain Max [dB]	Punto 1		Punto 2		Pendiente [dB/Dec]	Error absoluto
			Frec. P1 [Hz]	Gain P1 [dB]	Frec. P2 [Hz]	Gain P2 [dB]		
0.1	0.855	8	1000.05	0.86	3981.03	-0.34	-2.00009	8.76E-05
0.2	0.75	2.32	954.98	1.73	4466.81	-0.97	-4.02983	0.029831
0.3	0.647	3.55	911.95	2.63	5011.82	-1.85	-6.05385	0.053854
0.4	0.56	4.82	870.97	3.61	5011.82	-2.49	-8.0264	0.026397
0.5	0.46	6.55	776.16	5.04	6025.66	-3.92	-10.0668	0.066812
0.6	0.355	8.8	645.59	7.02	6456.49	-5.07	-12.0895	0.08952
0.7	0.271	11.04	537	8.96	5370.38	-5.05	-14.0096	0.009569
0.8	0.161	15.59	354.83	13.16	3548.15	-2.86	-16.0203	0.020294
0.85	0.051	25.62	125.91	22.88	1258.96	5.91	-16.9708	-0.02918
0.9	0.0275	29.3	91.27	26.55	911.95	8.48	-18.0765	0.076454
0.95	0.0202	29.8	97.79	26.96	977.33	7.95	-19.0148	0.014815
0.99	0.008	30.23	109.71	27.2	1096.54	7.43	-19.7744	-0.02562

Se procede a calcular el error relativo de las pendientes obtenidas con los valores de  $\hat{A}$  obtenidos destacando que ninguno de los errores calculados rebasa el 1%

**Tabla 3.10:** Tabla comparativa  $\alpha$ ,  $\hat{A}$  y pendiente medida

Orden ( $\alpha$ )	Pendiente [dB/Dec]	Error [%]	$\hat{A}$
0.1	-2.0000876	-0.004378	0.855
0.2	-4.0298315	-0.745787	0.75
0.3	-6.0538541	-0.897569	0.647
0.4	-8.0263974	-0.329968	0.56
0.5	-10.066812	-0.668121	0.46
0.6	-12.08952	-0.746002	0.355
0.7	-14.009569	-0.068353	0.271
0.8	-16.020294	-0.126838	0.161
0.85	-16.97082	0.1716497	0.051
0.9	-18.076454	-0.424743	0.0275
0.95	-19.014815	-0.077973	0.0202
0.99	-19.774385	0.129370198	0.008

En la figura 3.5 se presenta de forma gráfica los valores de las pendientes obtenidas con los valores de  $\hat{A}$  encontrados de forma empírica comparándolos con los valores de pendientes ideales, de igual forma se destaca que estos valores están muy cercanos a los ideales.



**Figura 3.5:** Gráfica comparativa entre valores de pendiente ideal y valores de pendiente obtenidos experimentalmente.

### 3.4.1. Ajuste de Curvas

El software Matlab dispone de la aplicación Curve Fitting Toolbox la cual cuenta con diversas funciones para ajustar curvas y superficies a series de datos. La caja de herramientas permite realizar análisis de datos exploratorios, pre-procesar y post-procesar datos, comparar modelos candidatos y eliminar valores atípicos. Puede realizar un análisis de regresión utilizando la biblioteca de modelos lineales y no lineales provistos [28].

Retomando los valores de  $\hat{A}$  presentados en la tabla 3.9 se utilizó esta herramienta para ajustar alguna de las funciones disponibles al comportamiento de los valores obtenidos, después de diversas pruebas se encontró que el ajuste de curvas mediante funciones exponenciales y polinomiales son los que mejor se adaptan al comportamiento de los puntos obtenidos. A continuación, se describe cada uno de ellos.

- **Ajuste de una curva polinómica**, devuelve los coeficientes para un polinomio  $p(x)$  de grado  $n$  que es la mejor solución (en el sentido de los mínimos cuadrados) para los datos de  $x$ . Los coeficientes de  $p$  están en potencias descendentes y la longitud de  $p$  es  $n + 1$ .

$$p(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1} \quad (3.14)$$

La figura 3.6 es la ventana de configuración para este tipo de ajuste, el grado del polinomio fue el más conveniente y no se utilizó la opción de robustez ya que no presentó mejoría en ningún caso.

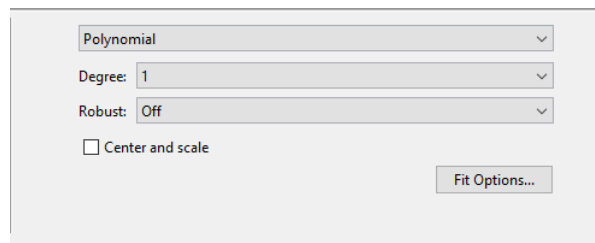


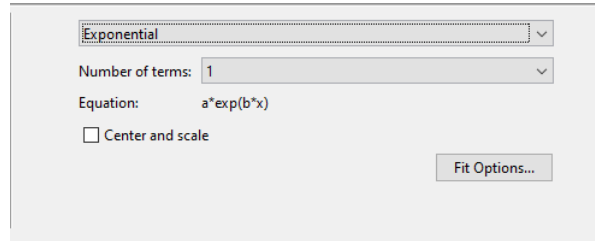
Figura 3.6: Ventana de configuración de función de ajuste polinomial en Curve Fitting Toolbox.

- **Modelos exponenciales**, La caja de herramientas proporciona un modelo exponencial de uno y dos términos tal como lo da

$$y = ae^{bx},$$

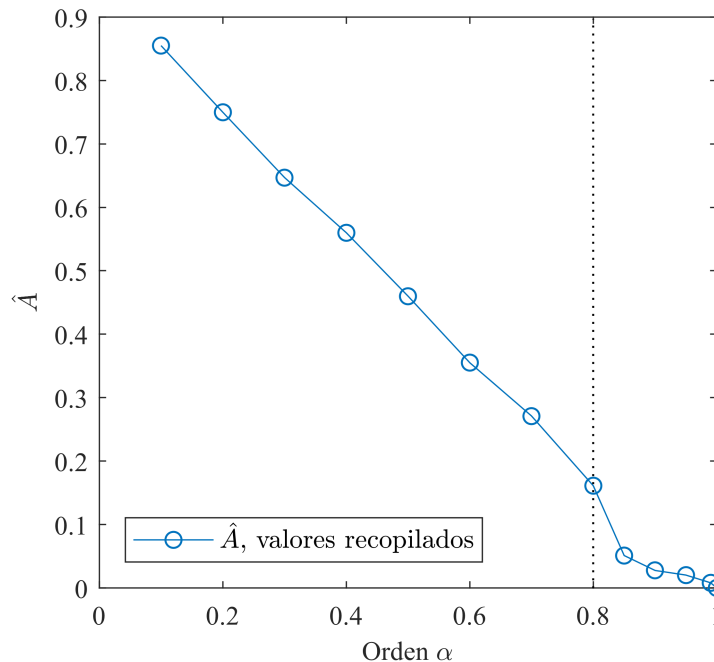
$$y = ae^{bx} + ce^{dx}$$

Los exponenciales a menudo se usan cuando la tasa de cambio de una cantidad es proporcional a la cantidad inicial de la cantidad. Si el coeficiente asociado con  $b$  y/o  $d$  es negativo, y representa un decrecimiento exponencial. Si el coeficiente es positivo, y representa un crecimiento exponencial. La figura 3.7 es la ventana de configuración para esta aproximación.



**Figura 3.7:** Ventana de configuración función de ajuste exponencial en Curve Fitting Toolbox.

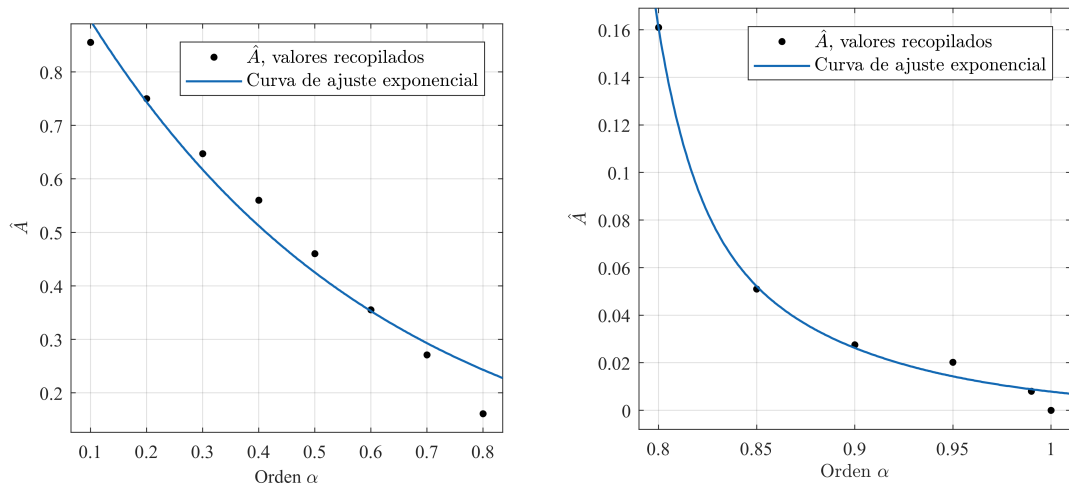
Tomando los valores de la tabla 3.10 se realizó una gráfica de los valores de  $\hat{A}$  recopilados experimentalmente en función del orden como se muestra en la figura 3.8. Analizando esta gráfica, observamos que a partir del orden  $\alpha = 0.8$  la curva tiene un comportamiento abrupto el cual entorpece el ajuste de curvas por lo que se decide dividir en dos secciones una del rango de orden  $[0.1, 8]$  y la otra de  $[0.8, 1]$  de esta forma el ajuste de curvas será más preciso principalmente la región de mayor interés para nosotros es el rango de orden  $[0.8, 1]$ .



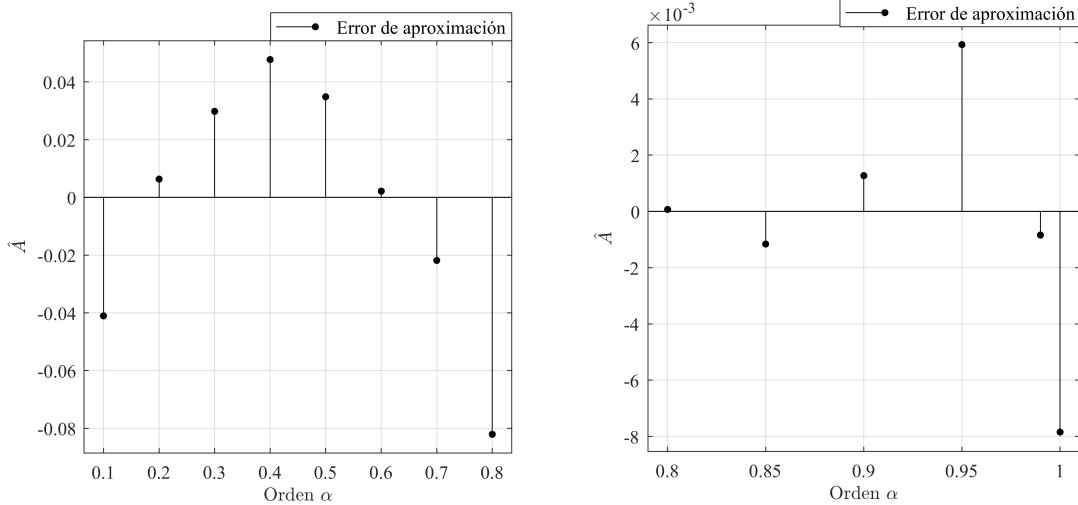
**Figura 3.8:** Gráfica de valores de  $\hat{A}$  en función del orden  $\alpha$ .

Se realizó un ajuste exponencial como se muestra en la figura 3.9, el ajuste para

el rango  $[0.1, 0.8]$  fue de orden 1 mientras que para el rango rango  $[0.8, 1]$  fue de orden 2



(a) Valores recopilados de  $\hat{A}$  en el intervalo  $\alpha$  de  $[0.1, 0.8]$  y curva de ajuste (b) Valores recopilados de  $\hat{A}$  en el intervalo  $\alpha$  de  $[0.8, 1]$  y curva de ajuste



(c) Error de aproximación de la curva de ajuste polinomial en el intervalo  $\alpha$  de  $[0.1, 0.8]$  (d) Error de aproximación de la curva de ajuste polinomial en el intervalo  $\alpha$  de  $[0.8, 1]$

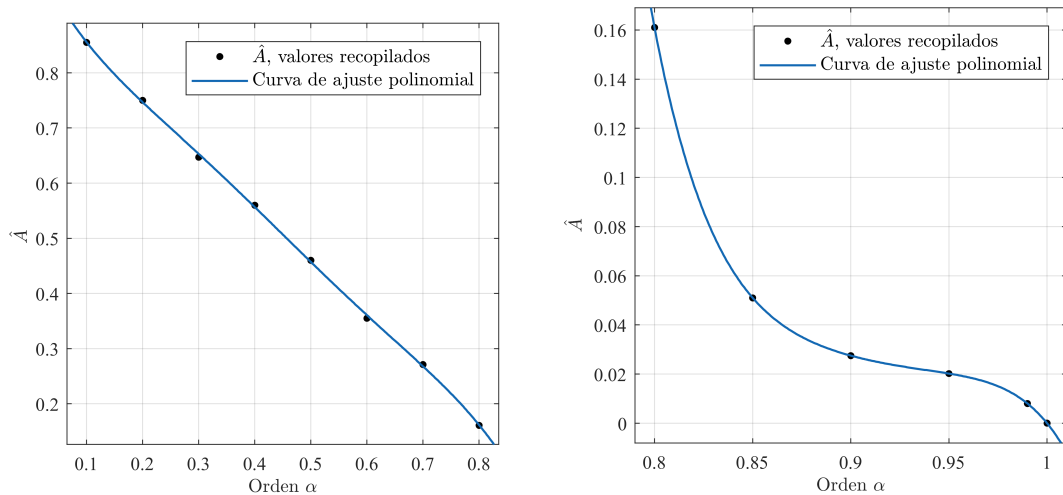
**Figura 3.9:** Gráficas de la curva de ajuste exponencial para los valores de  $\hat{A}$  y los errores de aproximación

Los valores de los coeficientes de la función de ajuste exponencial para los dos rangos en los que se dividió la gráfica de  $\hat{A}$  en función del orden se presentan en la tabla 3.11

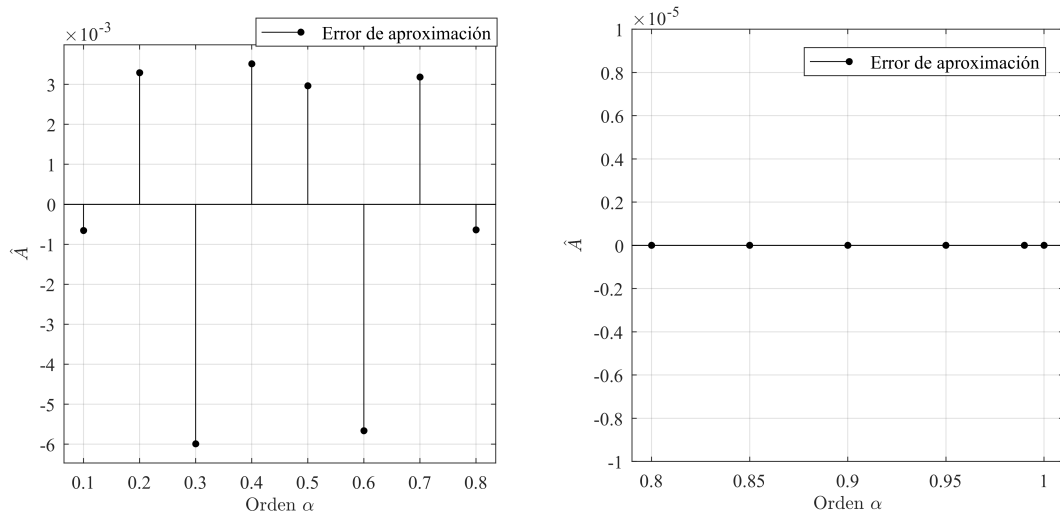
**Tabla 3.11:** Coeficientes de función de ajuste exponencial

Coeficientes rango [0.1,0.8]	Coeficientes rango [0.8,1]
$a = -3.256e + 17$	$a = 1.08$
$b = -53.63$	$b = -1.864$
$c = 1194$	
$d = -11.93$	

Posteriormente se hizo un ajuste curvas de tipo polinomial como se muestra en la figura 3.10. Los valores de los coeficientes se presentan en la tabla 3.12



(a) Valores recopilados de  $\hat{A}$  en el intervalo  $\alpha$  de [0.1, 0.8] y curva de ajuste      (b) Valores recopilados de  $\hat{A}$  en el intervalo  $\alpha$  de [0.8, 1] y curva de ajuste



(c) Error de aproximación de la curva de ajuste polinomial en el intervalo  $\alpha$  de [0.1, 0.8]      (d) Error de aproximación de la curva de ajuste polinomial en el intervalo  $\alpha$  de [0.8, 1]

**Figura 3.10:** Gráficas de la curva de ajuste polinomial para los valores de  $\hat{A}$  y los errores de aproximación

**Tabla 3.12:** Coeficientes

Coeficientes rango [0.1,0.8]	Coeficientes rango [0.8,1]
$P_1 = -9.775641025638677$	$P_1 = -3044.61152873634$
$P_2 = 22.118298368292734$	$P_2 = 1.3975.41854601980$
$P_3 = -18.769959207454168$	$P_3 = -2.5678.36240546023$
$P_4 = 7.374096736594660$	$P_4 = 2.3608.81573890708$
$P_5 = -2.289294871794486$	$P_5 = -1.086213849356012$
$P_6 = 1.027499999999976$	$P_6 = 2000.87814282981$

En base a los gráficos obtenidos se decide utilizar el ajuste de curvas por medio de polinomios ya que esta tuvo una aproximación más precisa y en el rango [0.8, 1] presento un error de orden  $10^{-5}$ , esto es importante ya que para aplicaciones en osciladores caóticos esta precisión ayudará a obtener resultados más certeros. Una vez obtenida la función de ajuste de curvas redefinimos A por la ecuación:

$$A = (P_1\alpha^5) + (P_2\alpha^4) + (P_3\alpha^3) + (P_4\alpha^2) + (P_5\alpha) + (P_6) \quad (3.15)$$

En la cual solo debemos sustituir los valores de  $P_n$  de la tabla 3.12 ya sea para el rango [0.1,0.8] ó [0.8,1].

En la tabla 3.13 se presentan los nuevos valores de frecuencia de corte y ganancia para los filtros pasa-bajas y pasa-altas calculados con A redefinida para implementar un integrador de orden fraccionario más preciso.

**Tabla 3.13:** Valores para configurar un filtro pasa bajas en diferentes órdenes

Orden	Pasa Bajas		Pasa Altas	
	Frecuencia de corte [kHz]	Gain	Frecuencia de corte [kHz]	Gain
0.1	0.855	1.16959	0.855	0.855
0.2	0.75	1.33333	0.75	0.75
0.3	0.647	1.5456	0.647	0.647
0.4	0.56	1.78571	0.56	0.56
0.5	0.46	2.17391	0.46	0.46
0.6	0.355	2.8169	0.355	0.355
0.7	0.271	3.69004	0.271	0.271
0.8	0.161	6.21118	0.161	0.161
0.85	0.051	19.6078	0.051	0.051
0.9	0.0275	36.3636	0.0275	0.0275
0.95	0.0202	49.505	0.0202	0.0202
0.99	0.008	125	0.008	0.008

Como se muestra en la tabla 3.13, la implementación de estos nuevos valores compensados tiene restricciones, aunque la metodología para su implementación es la misma que para el caso no compensado, en los siguientes puntos se describe los parámetros que se deben de modificar para cada uno de los casos.

- El rango de órdenes  $\alpha$  de  $[0.1, 0.85]$  puede ser implementado con la configuración predeterminada del chip, reloj de sistema 16000 kHz, las ganancias están dentro del rango permitido por lo que no presenta mayor problema
- Los valores  $\alpha$  de 0.9 y 0.95 requieren frecuencias de reloj menores a 0.05 kHz que no es posible de generar con un reloj de sistema de 16000 kHz por lo que es necesario cambiar este reloj por el de 8000kHz. Las ganancias en el filtro pasa bajo que requiere estos dos casos superan los límites absolutos por lo que igual tiene que ser dividida y amplificadas externamente para retornar al valor original
- El valor de 0.99 requiere un reloj de sistema de 2000 kHz para poder ingresar las frecuencias de corte tanto del filtro pasa bajo como pasa alto, la ganancia del filtro pasa bajo se divide y amplificadas externamente y la ganancia del filtro pasa alto debe ingresarse multiplicada por 10, posteriormente con una ganancia externa de 0.1 se retorna a su valor original.

Esta nueva topología de implementación de un integrador de orden fraccionario utilizando filtros pasa-bajas y pasa-altas presenta mayor versatilidad para implementar órdenes  $\alpha$  de incluso 0.99, lo cual es una gran ventaja con respecto a la implementación descrita en el capítulo anterior en la que se utilizó un filtro bilineal configurado como filtro polo y cero.

Además, el ajuste polinomial realizado con el cual redefinimos una función que nos da el valor de  $A$ , nos da la certeza de tener un integrador de orden fraccionario cuya respuesta se asemeja en gran medida a la ideal. Por lo que la siguiente etapa será incorporar este integrador de orden fraccionario al oscilador caótico de Lü.

# Capítulo 4

## Oscilador Caótico de Lü con integradores de orden fraccionario

En este capítulo se describe la implementación del oscilador caótico de Lü basado en series de funciones saturadas en FPAA, en la tarjeta Anadigm Quad Apex, utilizando integradores de orden fraccionario implementados en filtro bilineal y con la forma de filtro pasa-bajas y pasa-altas, a pesar de que la topología de implementación del integrador de orden fraccionario con filtro bilineal configurado como filtro polo y cero solo es útil en sus órdenes más altos de implementación para incorporarla en el oscilador caótico de Lü y generar caos se decide incorporarla y corroborar su funcionamiento.

Particularmente el objetivo de esta implementación es crear un oscilador caótico reconfigurable en orden  $\alpha$  y en número de enrollamientos.

### 4.1. Variables de estado del oscilador caótico de Lü

El oscilador caótico de múltiples enrollamientos de Lü puede ser descrito por un sistema de ecuaciones diferenciales acopladas, donde los coeficientes  $a$ ,  $b$ ,  $c$  y  $d$  son constantes positivas que pueden tomar valores en el intervalo  $[0, 1]$ ;  $f_0(x)$  es una función lineal a trozos conocida como desestabilizadora o función saturada, esta función produce un efecto oscilatorio en el sistema. Las variables de estado del sistema son  $x$ ,  $y$  y  $z$ . El sistema de ecuaciones diferenciales acopladas esta dado como 4.1.

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= z, \\ \dot{z} &= -ax - by - cz + df_0(x)\end{aligned}\tag{4.1}$$

Las derivadas convencionales en el sistema de ecuaciones diferenciales acoplados pueden ser remplazadas por las derivadas de orden fraccionario como se muestra en

el sistema 4.2:

$$\begin{aligned} {}_0D_t^\alpha x &= y, \\ {}_0D_t^\alpha y &= z, \\ {}_0D_t^\alpha z &= -ax - by - cz + df_0(x) \end{aligned} \quad (4.2)$$

La representación en diagramas de bloques de este sistema se muestra en la figura 4.1

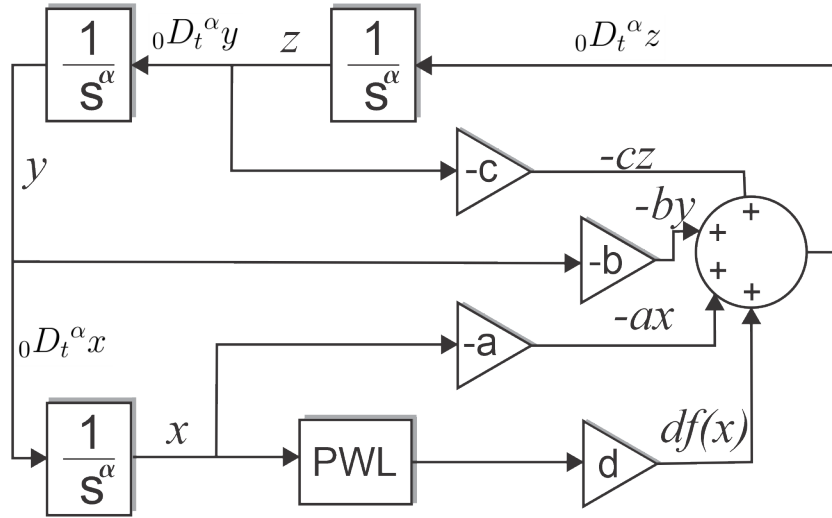


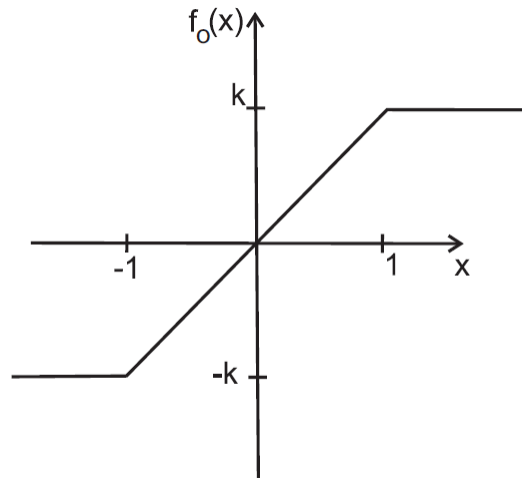
Figura 4.1: Oscilador caótico de Lu con integradores de orden fraccionario

## 4.2. Función Saturada

El oscilador caótico de Lu está controlado por una aproximación lineal a tramos  $f_0(x)$  que puede ser, como en este caso, una serie de la función saturada que determina la cantidad de los enrollamientos, así como su posición, magnitud y características generales y se obtiene a partir de la ecuación 4.2:

$$f_0(x) = \begin{cases} k, & \text{si } x > 1 \\ kx, & \text{si } |x| \leq 1 \\ -k, & \text{si } x < -1 \end{cases}$$

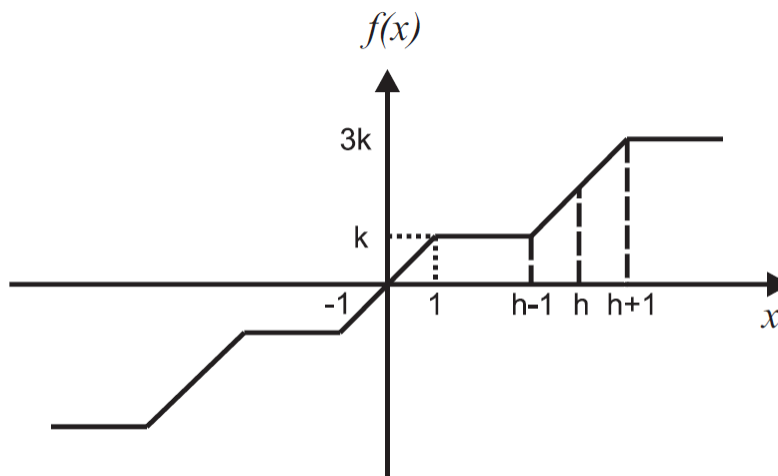
donde  $k$  es la pendiente del segmento medio, la parte superior  $f_0(x) = k$  tal que  $x < 1$  y la parte inferior  $f_0(x) = -k$  tal que  $x \leq -1$  son llamadas regiones saturadas, y el segmento  $f_0(x) = kx$  tal que  $|x| \leq 1$  entre las dos regiones saturadas es llamada pendiente de saturación. La figura 4.2 muestra la representación en fase de la función saturada  $f_0(x)$



**Figura 4.2:** Función saturada para generar dos enrollamientos en oscilador caótico

Reescribiendo la serie de funciones saturadas de la siguiente manera  $f(x; k, h, p, q)$ , se obtiene la ecuación 4.2 y en la figura 4.3 se muestra una función saturada con siete segmentos para generar 4 enrollamientos.

$$f(x; k, h, p, q) = \begin{cases} (2q + 1)k, & \text{si } x > qh + 1 \\ k(x - ih) + 2ik, & \text{si } |x - ih| \leq 1, -p \leq i \leq q \\ (2i + 1)k, & \text{si } |ih| + 1 < x < (i + 1)h - 1, -p \leq i \leq -1 \\ -(2q + 1)k, & \text{si } x < -qh - 1. \end{cases}$$



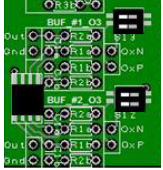
**Figura 4.3:** Función saturada para generar cuatro enrollamientos en oscilador caótico

### 4.3. Oscilador de caótico de Lü con integradores de orden fraccionario implementado en FPAA

La implementación del oscilador caótico de Lü con integradores de orden fraccionario en la tarjeta Anadigm Quad Apex requirió el uso de los 4 FPAA con los que ésta cuenta debido a que se buscó una distribución de los CAMs que cumpliera con las características descritas a continuación:

- La tarjeta Anadigm dispone de 4 buffers de salida los cuales son multipropósito y cumplen las funciones de convertir la señal completamente diferencial del FPAA en una señal simple, 2 de estos buffers se encuentran armados de fábrica [29] y los otros 2 no vienen con componentes superficiaales con el objetivo de que el usuario pueda armarlo de acuerdo a sus necesidades. Estos buffers se conectan las celdas IO del FPAA mediante interruptores de empaquetado de doble hilera (DIP, por sus siglas en inglés) como se describe en la tabla 4.2.

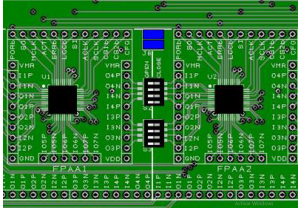
**Tabla 4.1:** DIP Switches

Switch	Función	Diagrama
S12	Conecta la salida 03 del FPAA 2 al buffer de salida No. 2	
S13	Conecta la salida 03 del FPAA 1 al buffer de salida No. 1	

Con el objetivo de poder visualizar el comportamiento del oscilador caótico en el plano de fase a través de un osciloscopio se buscó que las señales correspondientes a las variables  $X$  y  $Y$  pasaran por las celdas IO3 en los FPAA 1 y 2 y así poder ser conectadas a los buffers de salida No.1 y No.2.

- La placa cuenta también con interruptores DIP que permiten hacer conexiones de ciertas rutas entre FPAA vecinos, por ejemplo:

**Tabla 4.2:** DIP Switches

Switch	FunciónDiagrama	
Par superior de S2	Conecta O4 de FPAA 1 a I1 de FPAA 2	
Par inferior de S2	Conecta I4 de FPAA 1 a O1 de FPAA 2	
Par superior de S3	Conecta I3 de FPAA 1 a O2 de FPAA 2	
Par inferior de S3	Conecta O3 de FPAA 1 a I2 de FPAA 2	
Par superior de S4	Conecta O4 de FPAA 2 a I1 de FPAA 3	
Par inferior de S4	Conecta I4 de FPAA 2 a O1 de FPAA 3	
Par superior de S5	Conecta I3 de FPAA 2 a O2 de FPAA 3	
Par inferior de S5	Conecta O3 de FPAA 2 a I2 de FPAA 3	
Par superior de S6	Conecta O4 de FPAA 3 a I1 de FPAA 4	
Par inferior de S6	Conecta I4 de FPAA 3 a O1 de FPAA 4	
Par superior de S7	Conecta I3 de FPAA 3 a O2 de FPAA 4	
Par inferior de S7	Conecta O3 de FPAA 3 a I2 de FPAA 4	

- Estos interruptores están abiertos de forma predeterminada. Para reducir el uso de cableado externo se utilizaron todos los interruptores DIP disponibles.

#### 4.3.1. Oscilador caótico de Lü con integradores de orden $\alpha = 0.8$ implementados en filtro bilineal

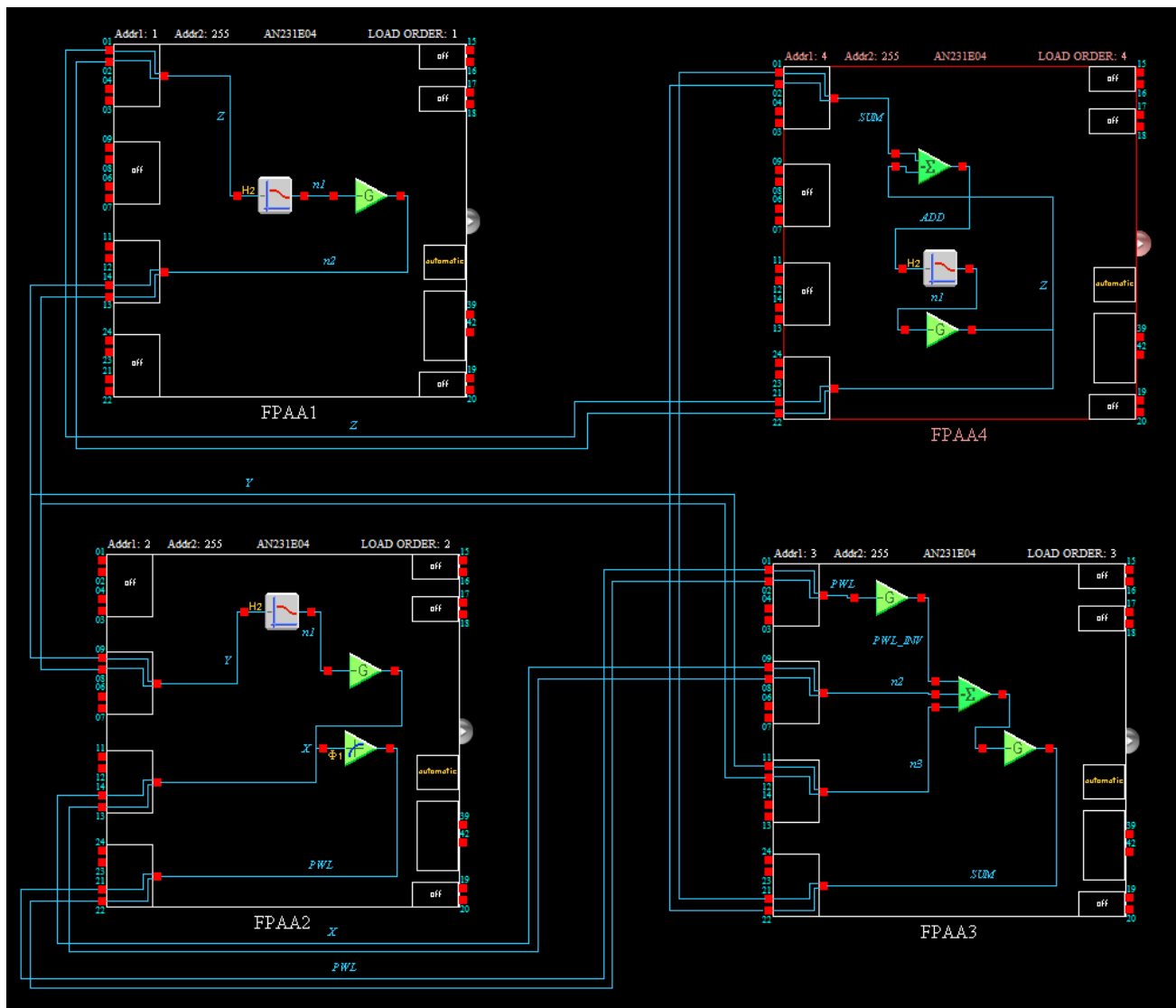
En la figura 4.4 se presenta el diseño final del oscilador caótico de Lü implementado en FPAA utilizando integradores de orden fraccionario en filtros bilineales mediante la configuración de filtro polo y cero para un orden  $\alpha = 0.8$  y con las constantes  $a = 4$ ,  $b = 0.7$ ,  $c = 0.7$  y  $d = 4$ .

Los 3 integradores de orden fraccionario pueden ser reconfigurados a cualquiera de los órdenes de la tabla 2.1 presentada en el capítulo 2, sin embargo existe una nueva restricción, debido al CAM de función de transferencia se debe buscar una frecuencia que pueda ser multiplicada por 16 y el resultado este dentro del rango de frecuencias, analizando esto se seleccionaron 3 posibles frecuencias de reloj y se valoró la que permite ingresar valores para ordenes más altos, como se ejemplifica en la tabla 4.3.

**Tabla 4.3:** Frecuencias de reloj posibles a implementar en oscilador caótico de Lü

Frecuencia	Ordenes configurables
50 kHz	[0.1,0.6]
100 kHz	[0.1,0.81]
500 khz	[0.1,0.3]

En base a la tabla 4.3 notamos que la frecuencia de 100 kHz tiene un rango más amplio para implementar órdenes  $\alpha$  de hasta 0.81, por lo que se configuraron los CAMs del diseño del oscilador caótico de Lü con este reloj.

**Figura 4.4:** Oscilador caótico de Lu con integradores de orden 0.8

Sus características se describen a continuación

## Configuración de relojes

Para los 4 chips la configuración de los relojes fue la que se presenta en la tabla 5.1, la mayoría de los CAMs implementados en los FPAA utilizaron el reloj 0 de 100 kHz a excepción del CAM de función de transferencia en el cual se configura la función saturada.

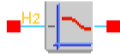

**Tabla 4.4:** Configuración de relojes para implementación de oscilador caótico de Lü de orden 0.8

<b>Master Clock-ACLK(fc)</b> 16 MHz	<b>System CLock (sys2=fc/1)</b> 16 MHz
<b>System CLock (sys1=fc/1)</b> 16 MHz	
<b>Clock 0 (sys1/160)</b> 100 kHz	<b>Clock 1 (sys1/10)</b> 1600 kHz
<b>Clock 2 (sys1/8)</b> 2000 kHz	<b>Clock 3 (sys1/64)</b> 250 kHz
<b>Clock 4 (sys1/1)</b> 16 kHz	<b>Clock 5 (sys1/1)</b> 16 MHz

## Configuración de CAMs

La configuración de cada uno de los CAMs que se implementaron en los 4 chips FPAA se presenta en las tablas siguientes:

### ■ FPAA1

Name	Options	Parameters	Clocks
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>Pole and Zero</i> <b>Automatic Variable</b> <i>DC Gain</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Pole Frequency</b> [kHz] <i>0.111</i> <b>Zero Frequency</b> [kHz] <i>8.99</i> <b>DC Gain</b> <i>9.00</i> <b>High Frequency Gain</b> <i>0.108</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>GainInv4</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		<b>Gain</b> <i>1.00</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>

**Figura 4.5:** Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 1

### ■ FPAA2


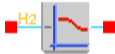

Name	Options	Parameters	Clocks
<b>TransferFunction1</b> (TransferFunction v1.0.1)  <i>Anadigm (Approved)</i>	<b>Output Hold</b> <i>On</i>		<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i> <b>ClockB</b> <i>1.6 MHz (Chip Clock 1)</i>
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>Pole and Zero</i> <b>Automatic Variable</b> <i>DC Gain</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Pole Frequency [kHz]</b> <i>0.111</i> <b>Zero Frequency [kHz]</b> <i>8.99</i> <b>DC Gain</b> <i>9.00</i> <b>High Frequency Gain</b> <i>0.108</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>GainInv4</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		<b>Gain</b> <i>1.00</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>

Figura 4.6: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPA A 2

### ■ FPA A3







Name	Options	Parameters	Clocks
<b>SumInv2</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	<b>Input 3</b> <i>On</i>	<b>Gain 1 (UpperInput)</b> <i>4.00</i> <b>Gain 2 (MiddleInput)</b> <i>4.00</i> <b>Gain 3 (LowerInput)</b> <i>0.700</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>GainInv1</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		<b>Gain</b> <i>1.00</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>GainInv3</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		<b>Gain</b> <i>1.00</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>

Figura 4.7: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPA A 3

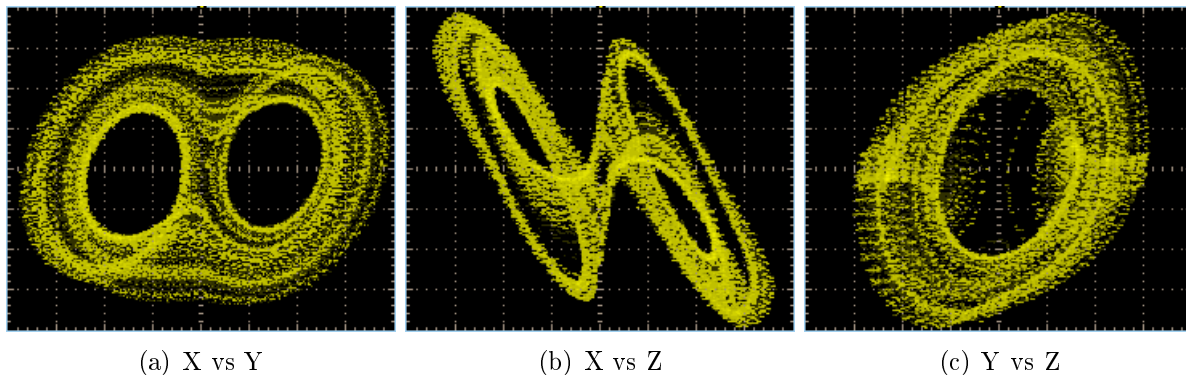
### ■ FPA A4

Name	Options	Parameters	Clocks
<b>SumInv2</b> (SumInv v1.0.1)  Anadigm (Approved)	<b>Input 3</b> <i>Off</i>	<b>Gain 1</b> (UpperInput) <i>1.00</i> <b>Gain 2</b> (LowerInput) <i>0.700</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  Anadigm (Approved)	<b>Filter Type</b> <i>Pole and Zero</i> <b>Automatic Variable</b> <i>DC Gain</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Pole Frequency</b> [kHz] <i>0.111</i> <b>Zero Frequency</b> [kHz] <i>8.99</i> <b>DC Gain</b> <i>9.00</i> <b>High Frequency Gain</b> <i>0.108</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>
<b>GainInv4</b> (GainInv v1.0.1)  Anadigm (Approved)		<b>Gain</b> <i>1.00</i>	<b>ClockA</b> <i>100 kHz (Chip Clock 0)</i>

**Figura 4.8:** Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPA4

### Caracterización del oscilador caótico de Lü de orden $\alpha = 0.8$

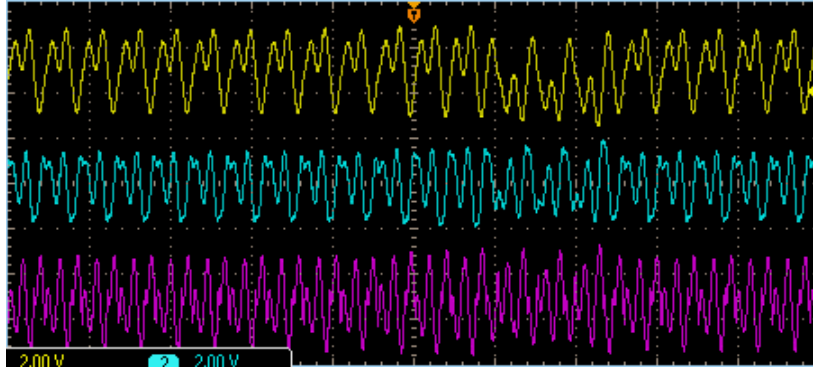
El diseño descrito anteriormente se implementó en la tarjeta Anadigm Quad Apex, conectando las salidas de buffer 1 y buffer 2 de ambas tarjetas a un osciloscopio se observó el comportamiento en el plano de fase. La figura 4.9 muestra las 3 vistas posibles de obtener mediante un osciloscopio de un atractor caótico de dos enrollamientos generado por el oscilador caóticos de Lü implementado con una  $\alpha = 0.8$



**Figura 4.9:** Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden  $\alpha = 0.8$ , 2 enrollamientos.

El comportamiento de este oscilador en el dominio del tiempo se muestra en la figura 4.10, la señal superior amarilla corresponde a la variable X, la señal azul

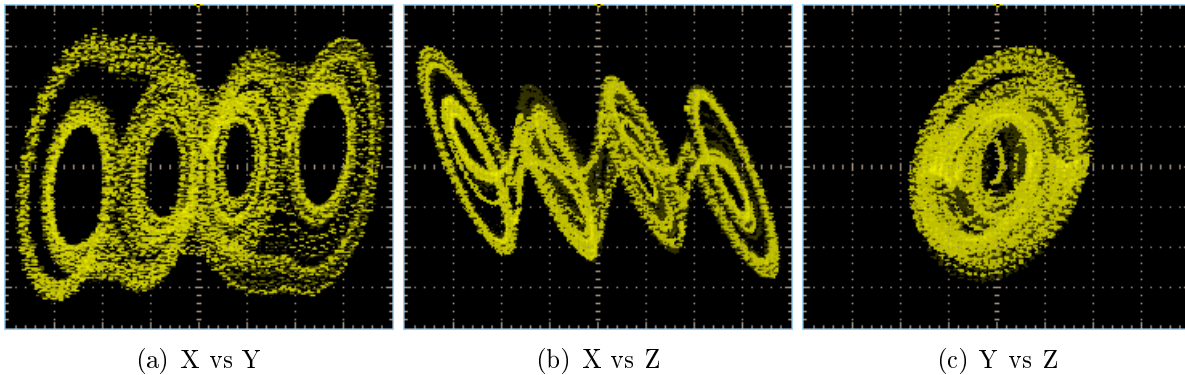
intermedia a la variable Y y la señal violeta inferior corresponde a la variable Z.



**Figura 4.10:** Respuesta en el dominio del tiempo de oscilador caótico de Lü, orden  $\alpha = 0.8$

Como siguiente prueba se modificó el bloque de la función saturada para poder visualizar un atractor caótico de 4 enrollamientos, en la figura 4.11 se presenta lo obtenido, la configuración de los demás CAMs sigue siendo la misma que para la prueba de un atractor caótico de 2 enrollamientos.

En la figura 4.11 se presentan nuevamente las vistas posibles del atractor caótico generado, pero esta vez de cuatro enrollamientos.



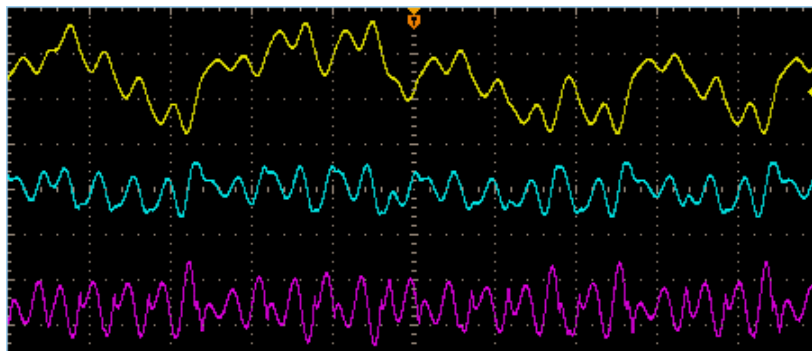
(a) X vs Y

(b) X vs Z

(c) Y vs Z

**Figura 4.11:** Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden  $\alpha = 0.8$ , 4 enrollamientos.

El comportamiento de este oscilador en el dominio del tiempo se muestra en la figura 4.12, la señal superior amarilla corresponde a la variable X, la señal azul intermedia a la variable Y y la señal violeta inferior corresponde a la variable Z.



**Figura 4.12:** Respuesta en el dominio del tiempo de oscilador caótico de Lü, orden  $\alpha = 0.8$

Concluida satisfactoriamente la implementación del oscilador caótico de Lü con integradores de orden  $\alpha = 0.8$  implementados con filtro polo y cero se procedió a implementar la segunda topología desarrollada con filtros pasa-bajas y pasa-altas

### 4.3.2. Oscilador caótico de Lü con integradores de orden fraccionario implementados en filtros pasa-bajas y paso-altas

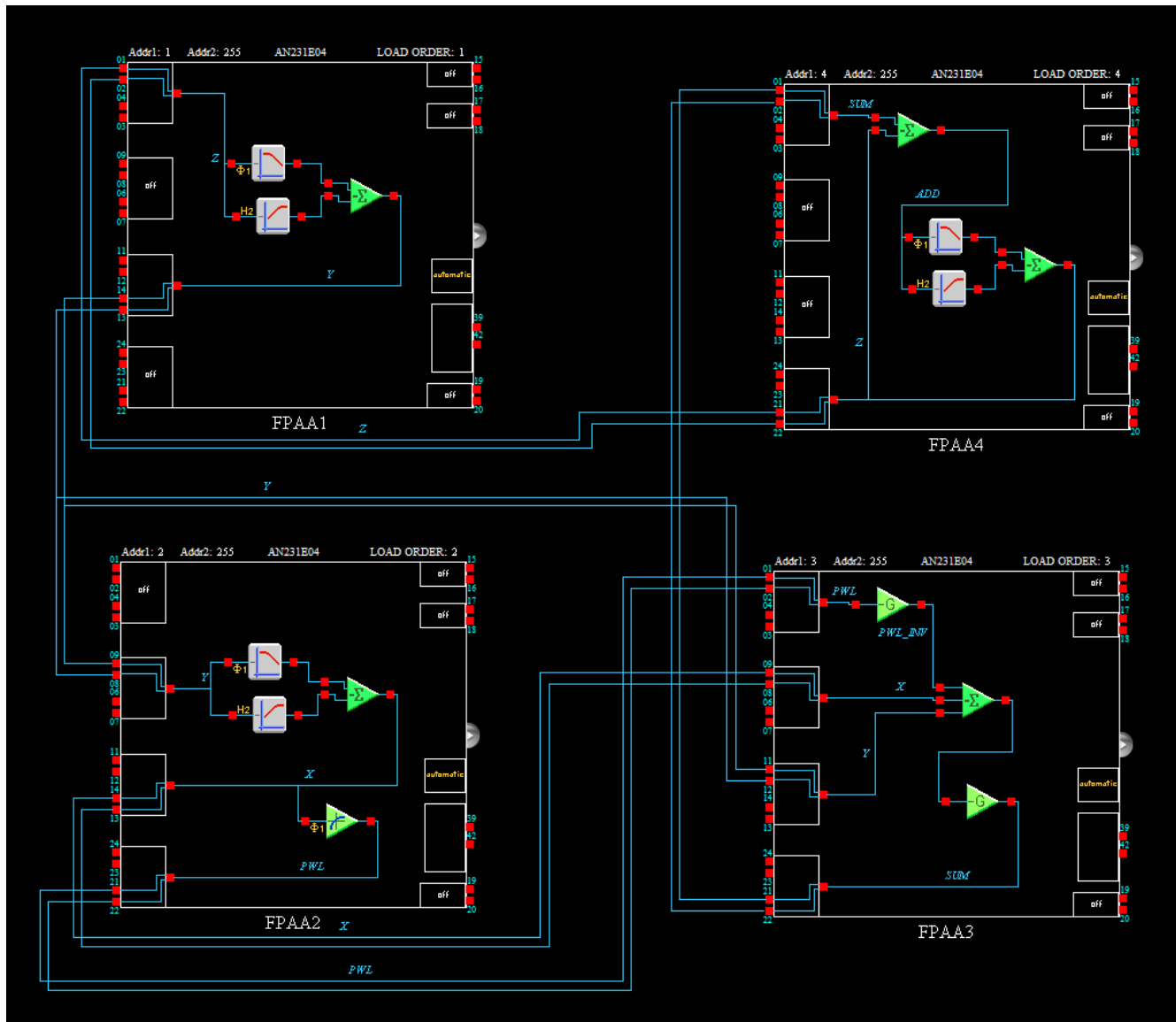


Figura 4.13: Oscilador caótico de Lü con integradores de orden 0.9

### Configuración de relojes

Tabla 4.5: Relojes utilizados

Master Clock-ACLK(fc)16 MHz	System CLock (sys2=fc/1) 16 MHz
System CLock (sys1=fc/1)16 MHz	
Clock 0 (sys1/320)50 MHz	Clock 1 (sys1/20) 800 kHz
Clock 2 (sys1/8) 2 MHz	Clock 3 (sys1/64) 250 kHz
Clock 4 (sys1/1) 16 MHz	Clock 5 (sys1/1) 16 MHz

## Configuración de CAMS

### ■ FPAA1




Name	Options	Parameters	Clocks
<b>FilterBilinear3</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>Low Pass</i> <b>Input Sampling Phase</b> <i>Phase 1</i> <b>Polarity</b> <i>Inverting</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>19.0</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>
<b>FilterBilinear4</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>High Pass</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>0.0528</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>
<b>SumInv2</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	<b>Input 3</b> <i>Off</i>	<b>Gain 1 (UpperInput)</b> <i>1.00</i> <b>Gain 2 (LowerInput)</b> <i>1.00</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>

Figura 4.14: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 1

### ■ FPAA2





Name	Options	Parameters	Clocks
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>Low Pass</i> <b>Input Sampling Phase</b> <i>Phase 1</i> <b>Polarity</b> <i>Inverting</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>19.0</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>
<b>FilterBilinear2</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	<b>Filter Type</b> <i>High Pass</i> <b>Resource Usage</b> <i>Minimum Resources</i>	<b>Corner Frequency [kHz]</b> <i>0.0525</i> <b>Gain</b> <i>0.0528</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>
<b>SumInv1</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	<b>Input 3</b> <i>Off</i>	<b>Gain 1 (UpperInput)</b> <i>1.00</i> <b>Gain 2 (LowerInput)</b> <i>1.00</i>	<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i>
<b>TransferFunction1</b> (TransferFunction v1.0.1)  <i>Anadigm (Approved)</i>	<b>Output Hold</b> <i>On</i>		<b>ClockA</b> <i>50 kHz (Chip Clock 0)</i> <b>ClockB</b> <i>800 kHz (Chip Clock 1)</i>

Figura 4.15: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPAA 2

### ■ FPAA3




Name	Options	Parameters	Clocks
SumInv2 (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>On</i>	Gain 1 (UpperInput) 2.00 Gain 2 (MiddleInput) 2.00 Gain 3 (LowerInput) 1.00	ClockA 50 kHz (Chip Clock 0)
GainInv3 (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain 1.00	ClockA 50 kHz (Chip Clock 0)
GainInv4 (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain 1.00	ClockA 50 kHz (Chip Clock 0)

Figura 4.16: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPA A 3

#### ■ FPA A4





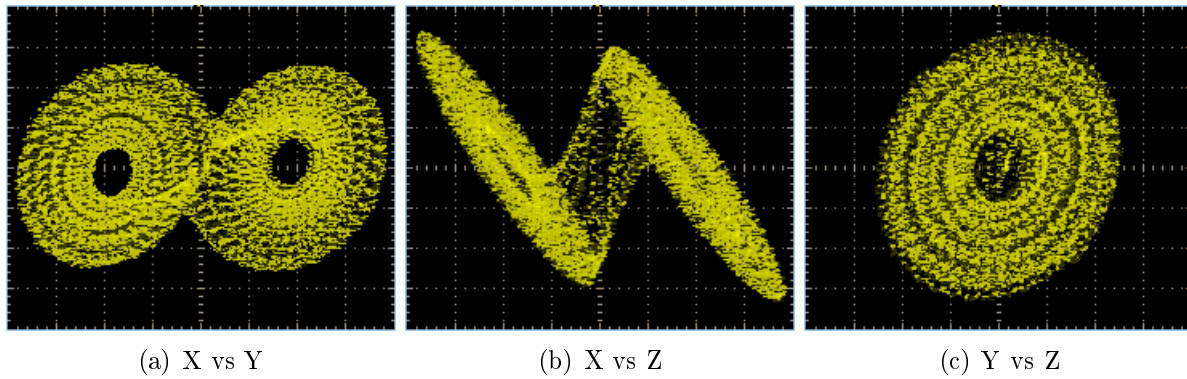
Name	Options	Parameters	Clocks
SumInv2 (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>Off</i>	Gain 1 (UpperInput) 1.00 Gain 2 (LowerInput) 0.600	ClockA 50 kHz (Chip Clock 0)
FilterBilinear1 (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>Low Pass</i> Input Sampling Phase <i>Phase 1</i> Polarity <i>Inverting</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] 0.0525 Gain 19.0	ClockA 50 kHz (Chip Clock 0)
FilterBilinear2 (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>High Pass</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] 0.0525 Gain 0.0528	ClockA 50 kHz (Chip Clock 0)
SumInv1 (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>Off</i>	Gain 1 (UpperInput) 1.00 Gain 2 (LowerInput) 1.00	ClockA 50 kHz (Chip Clock 0)

Figura 4.17: Configuración de los CAMs utilizados para implementar el oscilador caótico de Lü en el FPA A 4

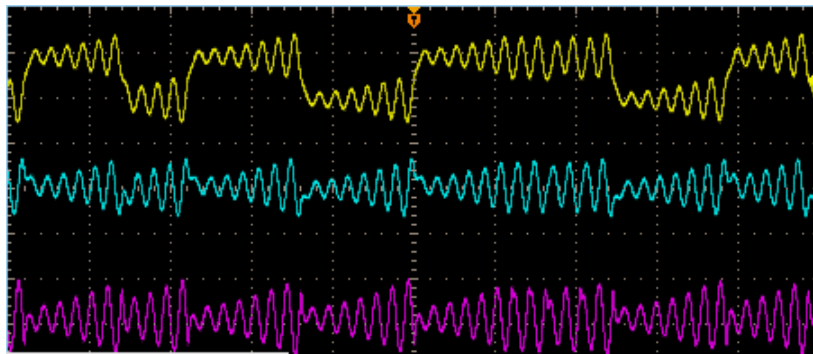
#### Caracterización del oscilador caótico de Lü de orden $\alpha = 0.9$

De la misma forma que para el oscilador caótico de Lü con integradores implementados en filtro bilineal se implementó este diseño en la tarjeta AnadigmQuadApex y a través de los buffers de salida comprobamos su funcionamiento en el plano de fase. La figura 4.18 es un atractor caótico de 2 enrollamientos, con la misma configuración para todos los CAMs y modificando el CAM de función de transferencia obtenemos la figura 4.20, la cual es un atractor caótico de 4 enrollamientos generado por esta implementación.



**Figura 4.18:** Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden  $\alpha = 0.9$ , 2 enrollamientos.

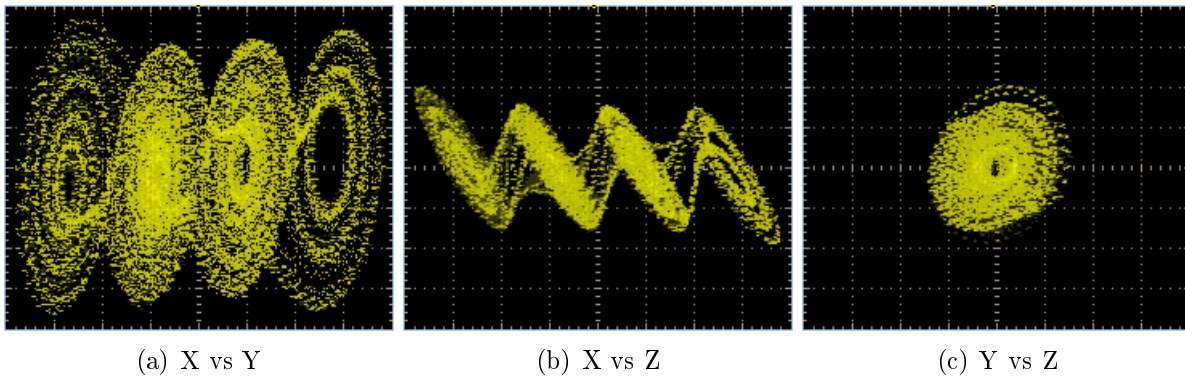
El comportamiento de este oscilador en el dominio del tiempo se muestra en la figura 4.10, la señal superior amarilla corresponde a la variable X, la señal azul intermedia a la variable Y y la señal violeta inferior corresponde a la variable Z.



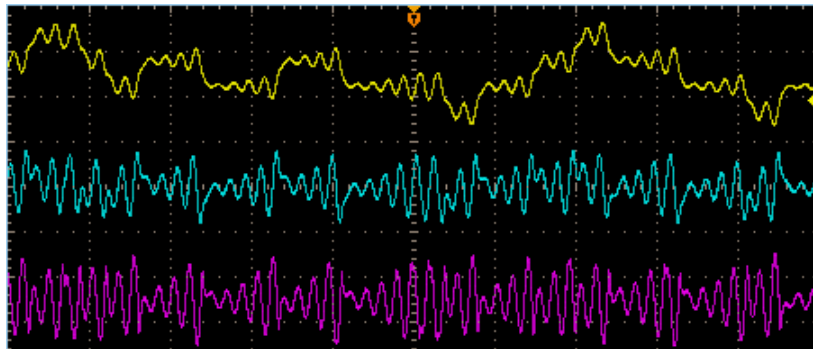
**Figura 4.19:** Respuesta en el dominio del tiempo de oscilador caótico de Lu, orden  $\alpha = 0.9$

Se modificó la función satura para obtener un oscilador caótico de 4 enrollamientos como se presenta en la figura 4.11, la configuración de los demás CAMs sigue siendo la misma.

El comportamiento de este oscilador en el dominio del tiempo se muestra en la figura 4.12, la señal superior amarilla corresponde a la variable X, la señal azul intermedia a la variable Y y la señal violeta inferior corresponde a la variable Z.

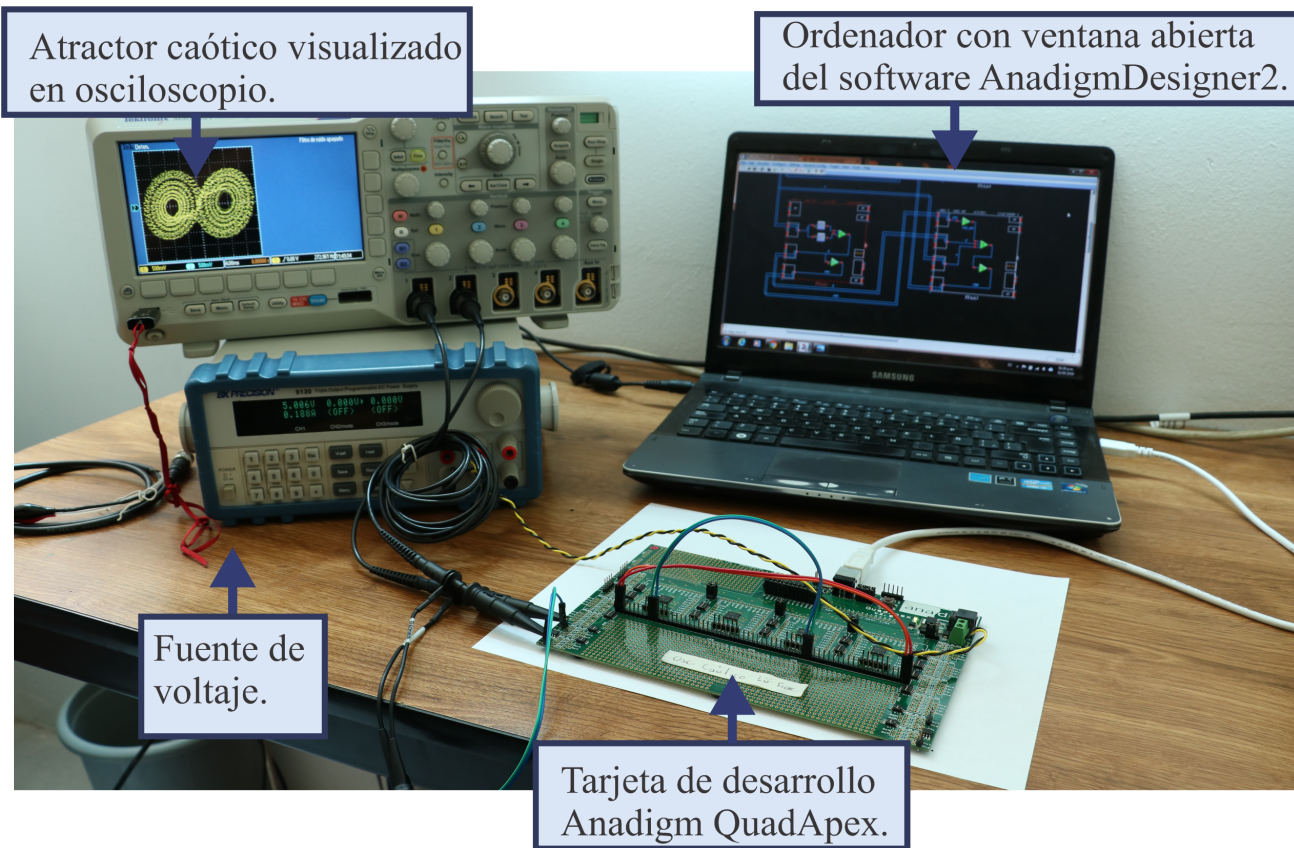


**Figura 4.20:** Diferentes vistas en el plano de fase del comportamiento del oscilador caótico de Lü de orden  $\alpha = 0.9$ , 4 enrollamientos.



**Figura 4.21:** Respuesta en el dominio del tiempo de oscilador caótico de Lü, orden  $\alpha = 0.9$

En la figura 4.22 se muestra la implementación física del oscilador caótico de Lü en la tarjeta Anadigm QuadApex la cual integra 4 FPAA's. En la pantalla del ordenador se observa parte del diseño del oscilador caótico en el software Anadigm-Designer2, la tarjeta debe ser alimentada por una fuente de voltaje DC estable de 5 Volts como se muestra en la figura, en la pantalla del osciloscopio se observa un atractor caótico de dos enrollamientos generado por el oscilador caótico de Lü de orden  $\alpha = 0.9$  cargado en la tarjeta.



**Figura 4.22:** Implementación física de oscilador caótico de Lü de orden  $\alpha = 0.9$

Concluida satisfactoriamente la implementación del oscilador caótico de Lü con integradores de orden  $\alpha = 0.9$  implementados con la topología de filtro pasa-bajas y filtro pasa-altas se observa que el comportamiento no es afectado de forma significativa por factores adversos como señales parásitas, por lo que procedemos a implementar este oscilador en la siguiente etapa que es la sincronización de dos osciladores de Lü.

# Capítulo 5

## Sincronización de dos osciladores caóticos de Lü de orden fraccionario

En el presente capítulo se describe el esquema de sincronización entre dos osciladores caóticos de Lü de orden fraccionario mediante la forma maestro-seguidor, también la configuración de los CAMs utilizada en FPAA, los cuales como se presentó en el capítulo 4 incorporan integradores de orden fraccionario, para realizar esta sincronización se escogió trabajar con la topología de un integrador de orden fraccionario implementado mediante filtro pasa-bajas y pasa-altas ya que es la que nos brinda mayor versatilidad para reconfigurar el sistema y cambiar el orden.

### 5.1. Sincronización basada en el teorema del valor final de la transformada de Laplace

En esta sección se describe la sincronización entre dos osciladores caóticos de Lü de orden fraccionario considerando el método de acoplamiento unidireccional [30]. Los sistemas maestro y seguidor están descritos por las ecuaciones diferenciales de orden fraccionario 5.1 y 5.2 respectivamente.

$$\begin{aligned} {}_0D_t^\alpha x_M &= y_M, \\ {}_0D_t^\alpha y_M &= z_M, \\ {}_0D_t^\alpha z_M &= -ax_M - by_M - cz_M + df(x_M) \end{aligned} \tag{5.1}$$

$$\begin{aligned} {}_0D_t^\alpha x_F &= y_F + K_1(x_M - x_F), \\ {}_0D_t^\alpha y_F &= z_F + K_2(y_M - y_F), \\ {}_0D_t^\alpha z_F &= -ax_F - by_F - cz_F + df(x_M) + K_3(z_M - z_F) \end{aligned} \tag{5.2}$$

Donde  $\alpha$  es el orden fraccionario y  $K_n$ . ( $n = 1, 2, 3$ ) es el control de ganancia. Por definición los errores de sincronización entre el sistema maestro y seguidor los

denotamos como:

$$\begin{aligned} e_1 &= x_F - x_M, \\ e_2 &= y_F - y_M, \\ e_3 &= z_F - z_M \end{aligned} \tag{5.3}$$

El sistema de error para el esquema de sincronización 5.1 y 5.2, está dado por:

$$\begin{aligned} {}_0D^\alpha {}_t e_1(t) &= e_2(t) - k e_1, \\ {}_0D^\alpha {}_t e_2(t) &= e_3(t) - k e_2, \\ {}_0D^\alpha {}_t e_3(t) &= -a e_1(t) - b e_2 - c e_3 - k e_3, \end{aligned} \tag{5.4}$$

Aplicando la transformada de Laplace en ambos lados para el sistema 5.4, de la siguiente forma

$$E_i(s) = \mathcal{L}\{e_i(t)\}, (i = 1, 2, 3) \tag{5.5}$$

Y por definición

$$\mathcal{L}\left\{\frac{d^\alpha e_i}{dt^\alpha}\right\} = s^\alpha E_i(s) - s^{\alpha-1} e_i(0), (i = 1, 2, 3) \tag{5.6}$$

Aplicando la definición anterior obtenemos:

$$\begin{aligned} s^\alpha E_1(s) - s^{\alpha-1} e_1(0) &= E_2(s) - K_1 E_1(s), \\ s^\alpha E_2(s) - s^{\alpha-1} e_2(0) &= E_3(s) - K_2 E_2(s), \\ s^\alpha E_3(s) - s^{\alpha-1} e_3(0) &= -a E_1(s) - b E_2(s) - c E_3(s) - K_3 E_3(s) \end{aligned} \tag{5.7}$$

La ecuación 5.7 puede ser reescrita como:

$$\begin{aligned} E_1(s) &= \frac{E_2(s)}{s^\alpha + K_1} + \frac{s^{\alpha-1} e_1(0)}{s^\alpha + K_1}, \\ E_2(s) &= \frac{E_3(s)}{s^\alpha + K_2} + \frac{s^{\alpha-1} e_2(0)}{s^\alpha + K_2}, \\ E_3(s) &= \frac{-a E_1(s)}{s^\alpha + c + K_3} - \frac{b E_2(s)}{s^\alpha + c + K_3} + \frac{s^{\alpha-1} e_3(0)}{s^\alpha + c + K_3} \end{aligned} \tag{5.8}$$

Aplicando el teorema final del valor final de la transformada de Laplace [30] obtenemos lo siguiente:

■ **Para el error 1**

$$\begin{aligned} \lim_{t \rightarrow \infty} (e_1(t)) &= \lim_{s \rightarrow 0} (s E_1(s)) \\ &= \lim_{s \rightarrow 0} \left( s \frac{E_2(s)}{s^\alpha + K_1} + \frac{s^{\alpha-1} e_1(0)}{s^\alpha + K_1} \right) \\ &= \frac{1}{s^\alpha + K_1} \lim_{t \rightarrow \infty} (e_2(t)) \end{aligned} \tag{5.9}$$

■ Para el error 2

$$\begin{aligned}
\lim_{t \rightarrow \infty} (e_2(t)) &= \lim_{s \rightarrow 0} (sE_2(s)) \\
&= \lim_{s \rightarrow 0} \left( \frac{sE_3(s)}{s^\alpha + K_2} + \frac{s^{\alpha-1}e_2(0)}{s^\alpha + K_2} \right) \\
&= \frac{1}{s^\alpha + K_2} \lim_{t \rightarrow \infty} (e_3(t))
\end{aligned} \tag{5.10}$$

■ Para el error 3

$$\begin{aligned}
\lim_{t \rightarrow \infty} (e_3(t)) &= \lim_{s \rightarrow 0} (sE_3(s)) \\
&= \lim_{s \rightarrow 0} \left( \frac{-asE_1(s)}{s^\alpha + c + K_3} - \frac{bsE_2(s)}{s^\alpha + c + K_3} + \frac{s^\alpha e_3(0)}{s^\alpha + c + K_3} \right) \\
&= \frac{-a}{s^\alpha + c + K_3} \lim_{t \rightarrow \infty} (e_1(t)) - \frac{b}{s^\alpha + c + K_3} \lim_{t \rightarrow \infty} (e_2(t))
\end{aligned} \tag{5.11}$$

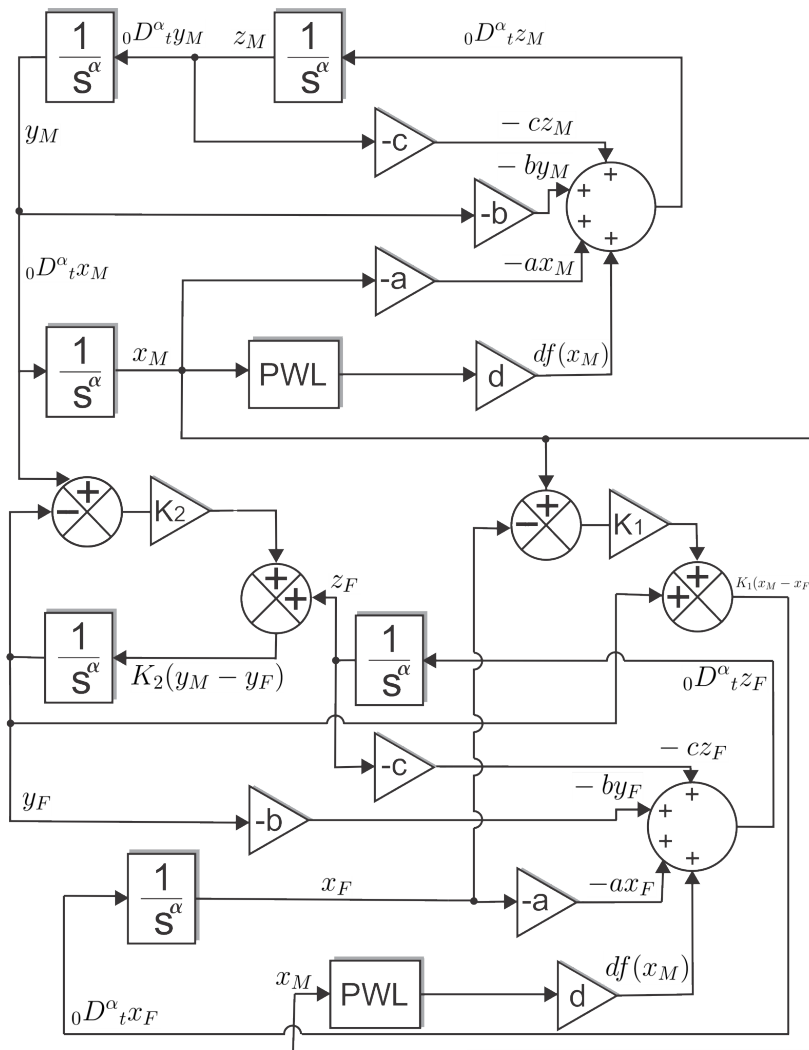
De acuerdo a las ecuaciones 5.9, 5.10 y 5.11, tenemos que:

$$\lim_{t \rightarrow \infty} (e_i(t)) = 0, \quad \text{con } i = 1, 2, 3 \tag{5.12}$$

Por lo que la expresión final obtenida en 5.9 y 5.10 se vuelve cero y en consecuencia también se vuelve cero la expresión final obtenida de 5.11. Al demostrar matemáticamente que el error tiende a cero significa que el sistema se encuentra sincronizado.

## 5.2. Diseño en software Anadigm de sistema maestro

Las ecuaciones de estado 5.1 y 5.2 de los osciladores caóticos Maestro y seguidor se representan en el siguiente diagrama de bloques, el cual es el sistema completo de sincronización que se implementará en FPAA 5.1. Como se observa en el diagrama no se incorporó un bloque diferenciador para  $K(z_M - z_F)$  ya que esta variable no será sincronizada con el objetivo de dejarla abierta a futuras aplicaciones, por lo que en este caso  $k_3 = 0$ .



**Figura 5.1:** Diagrama de bloques de sistema de sincronización entre dos osciladores caóticos de Lü de orden fraccionario de la forma maestro-seguidor

El oscilador caótico de Lü con integradores de orden fraccionario, se utilizó como sistema maestro para realizar la sincronización. La celda IO4 del FPAA1 se habilitó como salida para conectar un cable externo que lleva la señal  $y_M$  al sistema esclavo, de la misma forma se habilitó la celda IO1 del FPAA2 para conectar un cable que lleva la señal  $x_M$  al sistema esclavo, en la figura 5.2 se muestra el nuevo diseño implementado en el software AnadigmDesigner2.

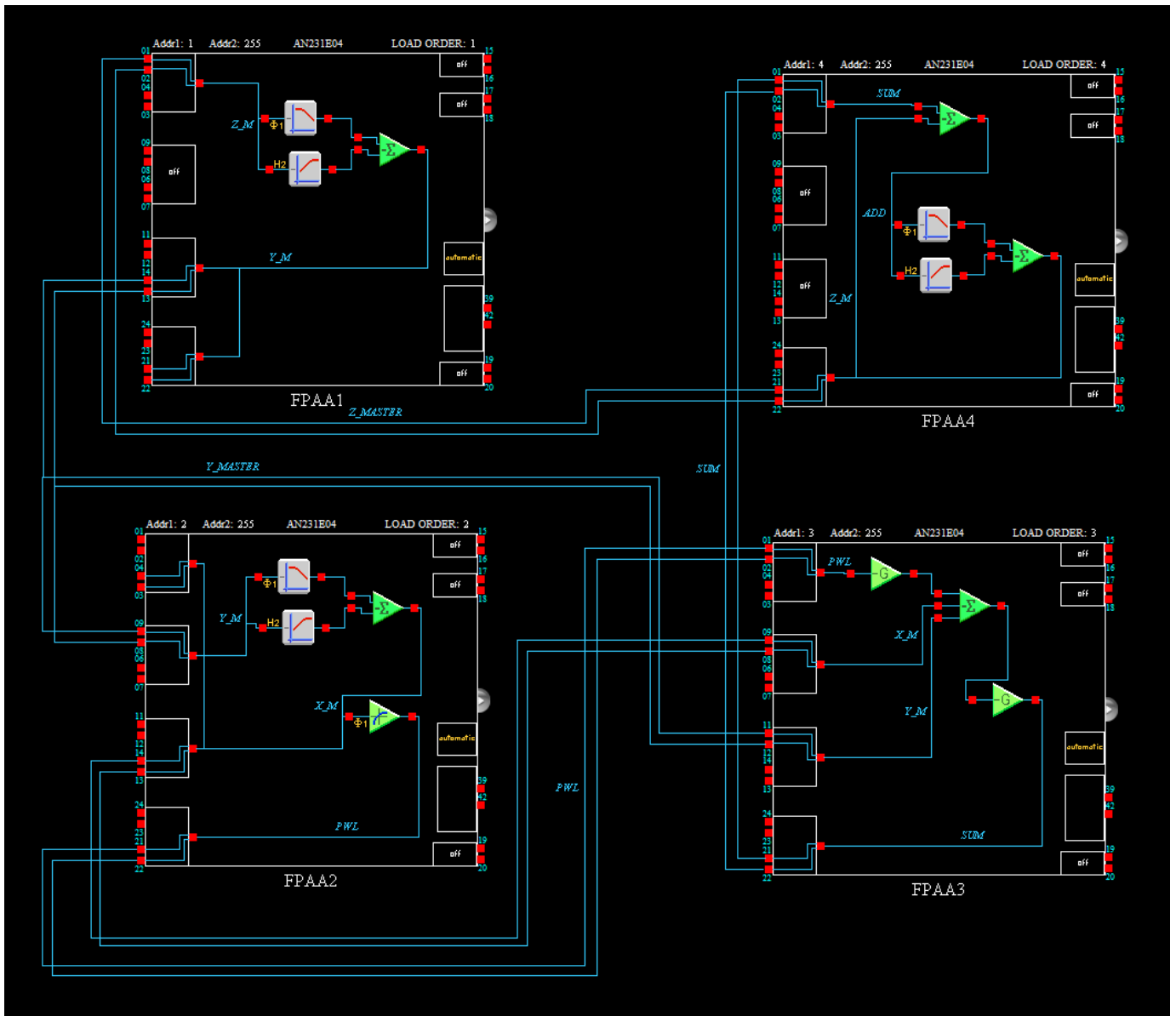


Figura 5.2: Oscilador caótico de Lü implementado como sistema maestro

La configuración de este sistema maestro para los CAMs, relojes y las celdas IO restantes sigue siendo la misma que se describió en el capítulo 4.

### 5.3. Diseño en software Anadigm de sistema seguidor

En base las ecuaciones de estado 5.2 se reestructuraron diversas partes del oscilador caótico de Lü para implementar el sistema seguidor. También en este diseño se incorporaron los bloques que realizan la diferencia de  $K(x_M - x_F)$  y  $K(y_M - y_F)$ , en los FPAA 2 y 1 respectivamente. En la figura 5.3 se presenta el diseño final del sistema seguidor.

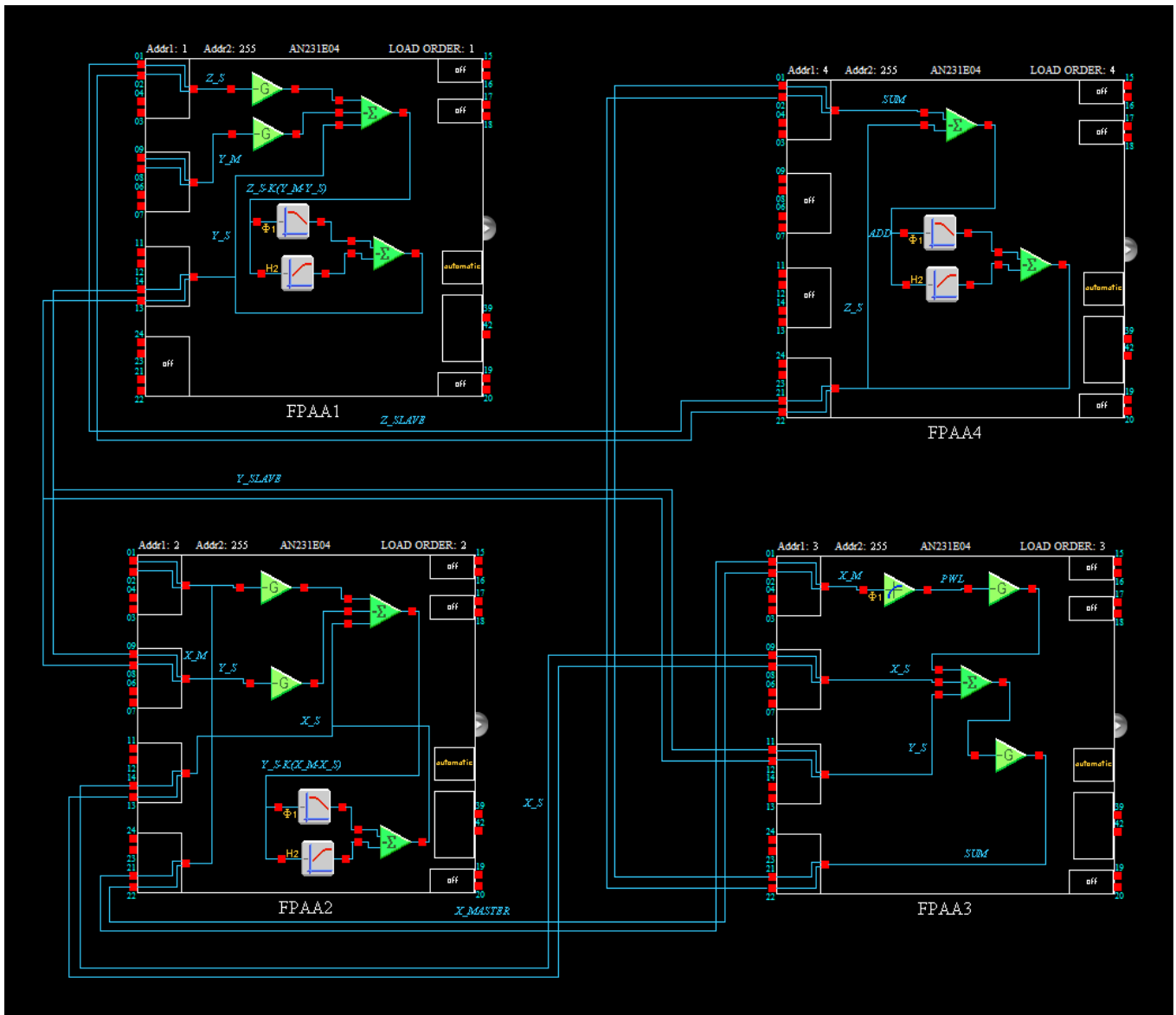
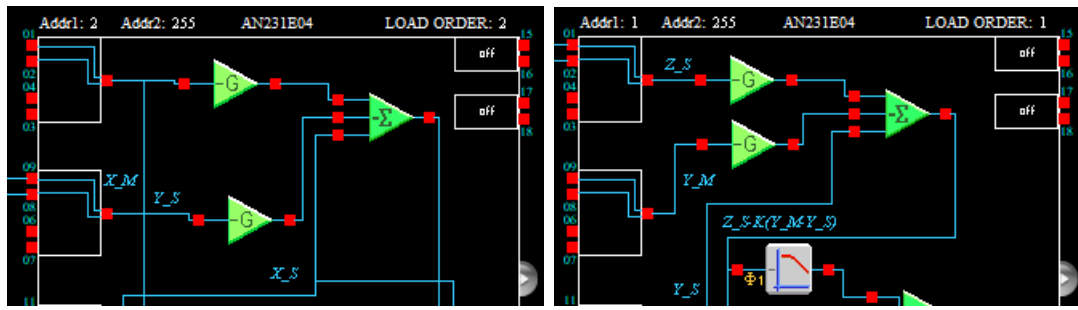


Figura 5.3: Oscilador caótico de Lu implementado como sistema seguidor

En la figura 5.4 se presentan los bloques implementados en FPAA que realizan la diferencia entre  $(x_M - x_F)$  y  $(y_M - y_F)$ , estos bloques se realizaron utilizando el CAM de suma diferencial con el cual fue posible controlar el valor de ganancia de  $K_n$ . Cuando  $K_1 = K_2 = 1$  la sincronización entre ambos atractores presentaba un error grande por lo que se realizaron diversas pruebas modificando este valor, llegando a los valores finales de  $K_1 = 85$  y  $K_2 = 90$ , con los cuales se obtuvo un error mínimo en la sincronización.



(a) Bloque de diferencia de  $K_1(x_M - x_F)$ , en chip 2  
 (b) Bloque de diferencia de  $K_2(y_M - y_F)$ , en chip 1

**Figura 5.4:** Bloques de diferencia para implementar  $K_1(x_M - x_F)$  y  $K_2(y_M - y_F)$

### 5.3.1. Configuración de relojes para sistema seguidor

Todos los CAMs utilizados en este diseño se configuraron con un reloj A de 50 kHz, el CAM de función de transferencia requiere dos relojes para su funcionamiento se configuró con un reloj A de 50 kHz y un reloj B de 800 kHz. En la tabla 5.1 se describe la configuración de los relojes utilizados para configurar los 4 chips.

**Tabla 5.1:** Relojes utilizados en implementación de integrador de orden fraccionario en filtro bilineal

Master Clock-ACLK(fc)16 MHz	System CLock (sys2=fc/1) 16 MHz
System CLock (sys1=fc/1)16 MHz	
Clock 0 (sys1/320) 50 MHz	Clock 1 (sys1/20) 800 kHz
Clock 2 (sys1/8) 2 MHz	Clock 3 (sys1/64) 250 kHz
Clock 4 (sys1/1) 16 kHz	Clock 5 (sys1/1) 16 MHz

### 5.3.2. Configuración de CAMs para sistema seguidor

En las figuras 5.5, 5.6, 5.7 y 5.8 se describe la configuración de los CAMs para cada uno de los FPAA utilizados. Cabe mencionar que esta descripción es para la implementación de un atractor caótico de orden 0.9, sin embargo, este orden puede ser modificado a otros órdenes siguiendo las restricciones descritas en el capítulo 3. Configuración de CAMs para FPAA 1.







Name	Options	Parameters	Clocks
<b>FilterBilinear3</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>Low Pass</i> Input Sampling Phase <i>Phase 1</i> Polarity <i>Inverting</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] <i>0.0525</i> Gain <i>19.0</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>FilterBilinear4</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>High Pass</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] <i>0.0525</i> Gain <i>0.0528</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>SumInv2</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>Off</i>	Gain 1 (UpperInput) <i>1.00</i> Gain 2 (LowerInput) <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>SumInv3</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>On</i>	Gain 1 (UpperInput) <i>1.00</i> Gain 2 (MiddleInput) <i>90.0</i> Gain 3 (LowerInput) <i>90.0</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>GainInv2</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>GainInv1</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>

Figura 5.5: Configuración CHIP 1

## Configuración de CAMs para FPAA 2.







Name	Options	Parameters	Clocks
<b>FilterBilinear1</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>Low Pass</i> Input Sampling Phase <i>Phase 1</i> Polarity <i>Inverting</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] <i>0.0525</i> Gain <i>19.0</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>FilterBilinear2</b> (FilterBilinear v1.0.2)  <i>Anadigm (Approved)</i>	Filter Type <i>High Pass</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] <i>0.0525</i> Gain <i>0.0528</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>SumInv1</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>Off</i>	Gain 1 (UpperInput) <i>1.00</i> Gain 2 (LowerInput) <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>SumInv3</b> (SumInv v1.0.1)  <i>Anadigm (Approved)</i>	Input 3 <i>On</i>	Gain 1 (UpperInput) <i>90.0</i> Gain 2 (MiddleInput) <i>1.00</i> Gain 3 (LowerInput) <i>85.0</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>GainInv2</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>
<b>GainInv1</b> (GainInv v1.0.1)  <i>Anadigm (Approved)</i>		Gain <i>1.00</i>	ClockA <i>50 kHz (Chip Clock 0)</i>

Figura 5.6: Configuración CHIP 2

## Configuración de CAMs para FPAA 3.





Name	Options	Parameters	Clocks
TransferFunction1 (TransferFunction v1.0.1)  Anadigm (Approved)	Output Hold <i>On</i>		ClockA 50 kHz (Chip Clock 0) ClockB 800 kHz (Chip Clock 1)
SumInv2 (SumInv v1.0.1)  Anadigm (Approved)	Input 3 <i>On</i>	Gain 1 (UpperInput) 2.00 Gain 2 (MiddleInput) 2.00 Gain 3 (LowerInput) 1.00	ClockA 50 kHz (Chip Clock 0)
GainInv4 (GainInv v1.0.1)  Anadigm (Approved)		Gain 1.00	ClockA 50 kHz (Chip Clock 0)
GainInv5 (GainInv v1.0.1)  Anadigm (Approved)		Gain 1.00	ClockA 50 kHz (Chip Clock 0)

Figura 5.7: Configuración CHIP 3

## Configuración de CAMs para FPAA 4.





Name	Options	Parameters	Clocks
SumInv2 (SumInv v1.0.1)  Anadigm (Approved)	Input 3 <i>Off</i>	Gain 1 (UpperInput) 1.00 Gain 2 (LowerInput) 0.600	ClockA 50 kHz (Chip Clock 0)
FilterBilinear1 (FilterBilinear v1.0.2)  Anadigm (Approved)	Filter Type <i>Low Pass</i> Input Sampling Phase <i>Phase 1</i> Polarity <i>Inverting</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] 0.0525 Gain 19.0	ClockA 50 kHz (Chip Clock 0)
FilterBilinear2 (FilterBilinear v1.0.2)  Anadigm (Approved)	Filter Type <i>High Pass</i> Resource Usage <i>Minimum Resources</i>	Corner Frequency [kHz] 0.0525 Gain 0.0528	ClockA 50 kHz (Chip Clock 0)
SumInv1 (SumInv v1.0.1)  Anadigm (Approved)	Input 3 <i>Off</i>	Gain 1 (UpperInput) 1.00 Gain 2 (LowerInput) 1.00	ClockA 50 kHz (Chip Clock 0)

Figura 5.8: Configuración CHIP 4

## 5.4. Caracterización de la sincronización entre los dos osciladores caóticos de Lü

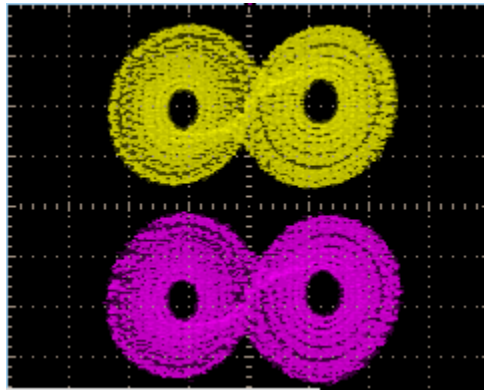
Configurando los integradores de los sistemas maestro y seguidor para un orden  $\alpha=0.8$ , se implementaron ambos sistemas en las tarjetas Anadigm Quad Apex, se activaron los buffers de salida 1 y 2 de cada tarjeta y mediante un osciloscopio visualizamos el comportamiento del sistema en el plano de fase. En la figura 5.13 presentamos la gráfica del atractor generado por ambos sistemas en el plano de

fase, la gráfica en el dominio del tiempo de la señal  $x_M$  vs  $x_F$  así como el error de sincronización en la parte superior y la gráfica en el plano de fase de  $x_M$  vs  $x_F$ , la cual entre más lineal sea su comportamiento es un indicador de un nivel óptimo de sincronización.

#### 5.4.1. Osciladores caóticos de Lü de orden 0.9 de 2 enrollamientos sincronizados

Para corroborar el funcionamiento del sistema de sincronización se visualizó en un osciloscopio el comportamiento tanto del sistema maestro como del sistema seguidor en el plano de fase y en el dominio del tiempo, lo que nos ayudó a cerciorarnos de que la sincronización fue correcta.

En la figura 5.9 se observa en la parte superior el atractor caótico generado por el sistema maestro y en la parte inferior de la imagen el atractor caótico generado por el sistema seguidor, el orden fraccionario de ambos sistemas es  $\alpha = 0.9$  y la función saturada fue configurada para generar 2 atractores.



**Figura 5.9:** Sincronización sistema maestro-seguidor, orden  $\alpha=0.9$ , 4 enrollamientos

En el plano de fase comparamos las señales  $x_M$  contra  $x_F$  y  $y_M$  contra  $y_F$ , al presentar un comportamiento lineal con una pendiente de 45 grados corroboramos que el sistema se encuentra sincronizado en fase.

Posteriormente analizamos el comportamiento de las señales en el dominio del tiempo, la señal roja que se ve en la parte superior de la figura 5.11 es el error entre las señales  $x_M$  vs  $x_F$  y  $y_M$  y  $y_F$  respectivamente, podemos ver que es casi nula.

En la parte inferior de las gráficas de la figura 5.11 se observa la señal de las variables  $X$  en (a) y  $Y$  en (b) correspondientes a los sistemas maestro y seguidor, como se puede observar las señales se encuentran casi superpuestas.

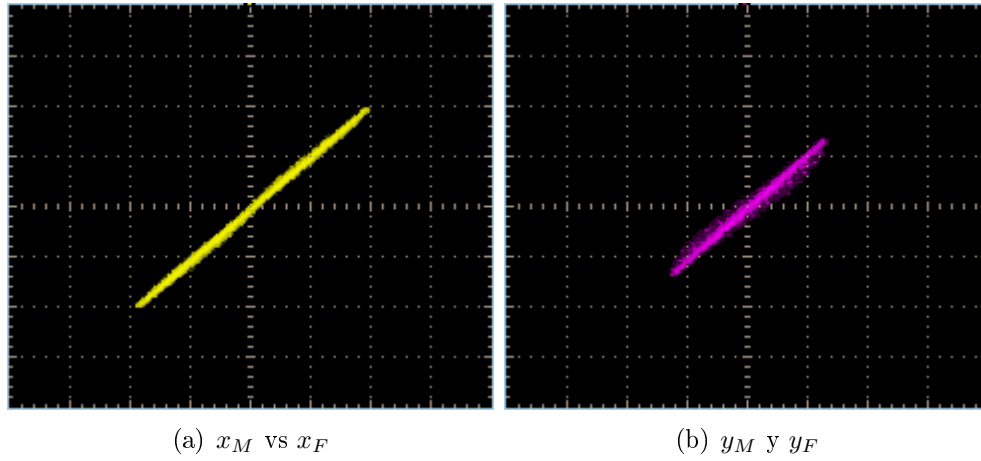
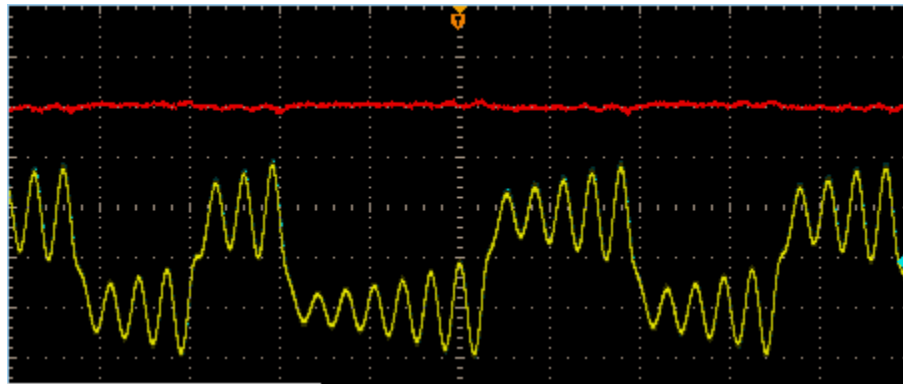
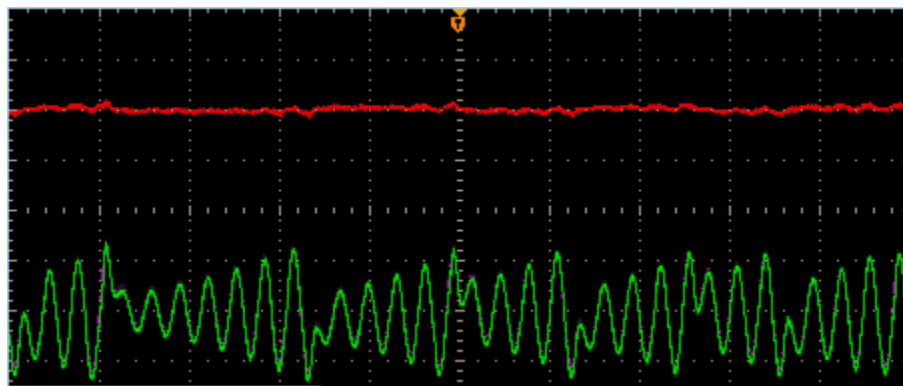


Figura 5.10: Gráficas en plano de fase



(a) Respuesta en el dominio del tiempo de señal  $x_M$  y  $x_F$  y error entre ambas señales

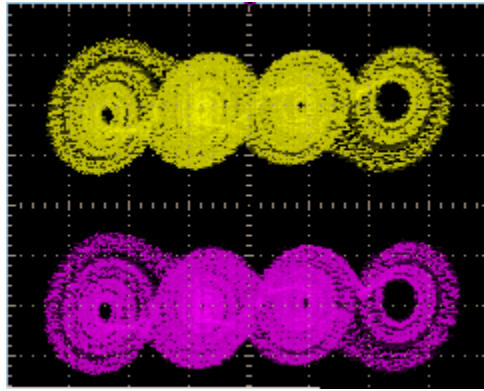


(b) Respuesta en el dominio del tiempo de señal  $y_M$  y  $y_F$  y error entre ambas señales

Figura 5.11: Comportamiento en el dominio del tiempo

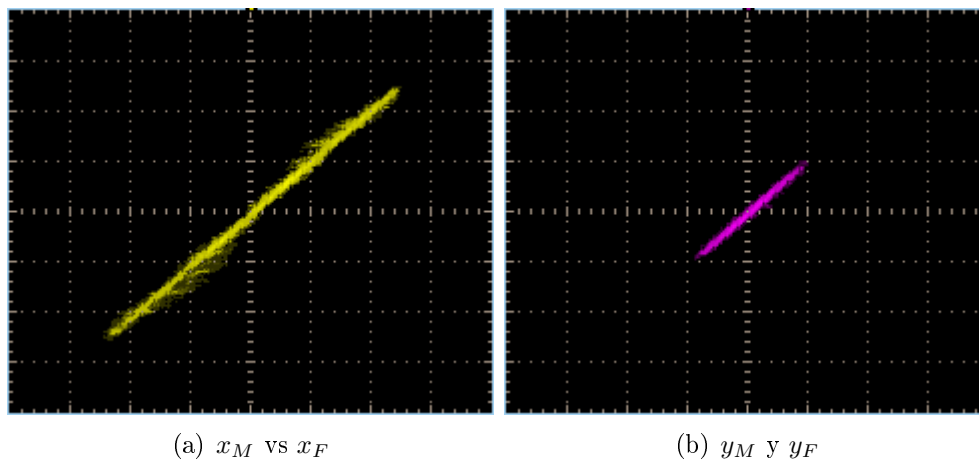
### 5.4.2. Osciladores caóticos de Lü de orden 0.9 de 4 enrollamientos sincronizados

Se configuraron los integradores de los sistemas maestro y seguidor para un orden  $\alpha = 0.9$ , en la figura 5.12 observamos la vista de los dos atractores caóticos generados por los sistemas maestro y seguidor, la gráfica en la escala de tiempo de las señales  $x_M$  y  $x_F$  y ambas señales gráficas en el plano de fase, al general una gráfica lineal indican que la sincronización ha sido exitosa.



**Figura 5.12:** Sincronización sistema maestro-seguidor, orden  $\alpha=0.9$ , 4 enrollamientos

En el plano de fase comparamos las señales  $x_M$  contra  $x_F$  y  $y_M$  contra  $y_F$ , al presentar un comportamiento lineal corroboramos que la sincronización como en el caso anterior fue exitosa.

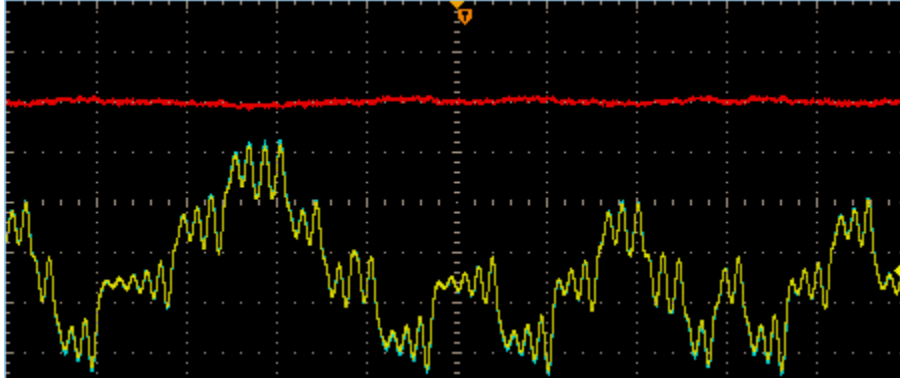


**Figura 5.13:** Gráficas en plano de fase

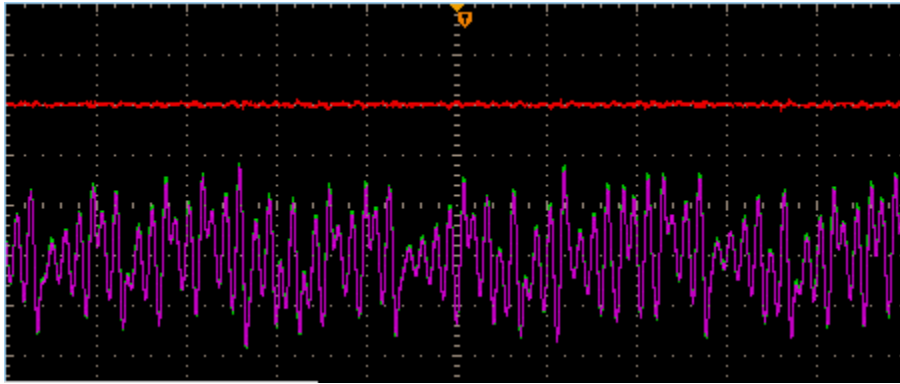
Posteriormente analizamos el comportamiento de las señales en el dominio del tiempo, la señal roja que se ve en la parte superior de ambas figuras es el error entre las señales  $x_M$  vs  $x_F$  y  $y_M$  y  $y_F$  respectivamente.

En la figura 5.14, (a) es la respuesta en el dominio del tiempo de la señal  $x_M$  en

comparación con  $x_F$  y (b) es la respuesta en el dominio del tiempo de la señal  $y_M$  en comparación con  $y_F$ .



(a) Respuesta en el dominio del tiempo de señal  $x_M$  y  $x_F$  y error entre ambas señales.



(b) Respuesta en el dominio del tiempo de señal  $y_M$  y  $y_F$  y error entre ambas señales

**Figura 5.14:** Comportamiento en el dominio del tiempo

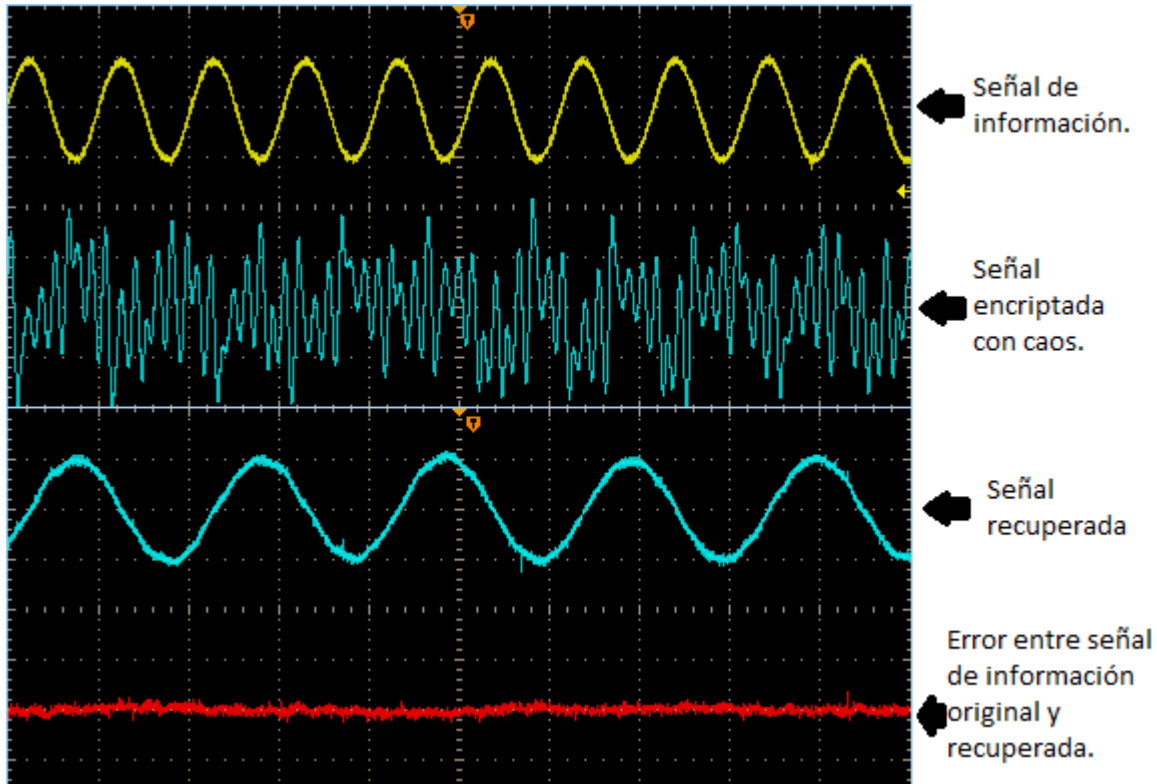
Como podemos observar en ambas gráficas, el error entre las señales del maestro y seguidor es mínimo.

### 5.4.3. Envío y recepción de una señal de información

Actualmente existen artículos que describen el uso de osciladores caóticos para la encriptación de información [31], este es un campo aún en desarrollo sin embargo en esta última fase del proyecto de tesis se realizó una prueba básica en la que se utilizó el sistema de sincronización realizado en esta aplicación.

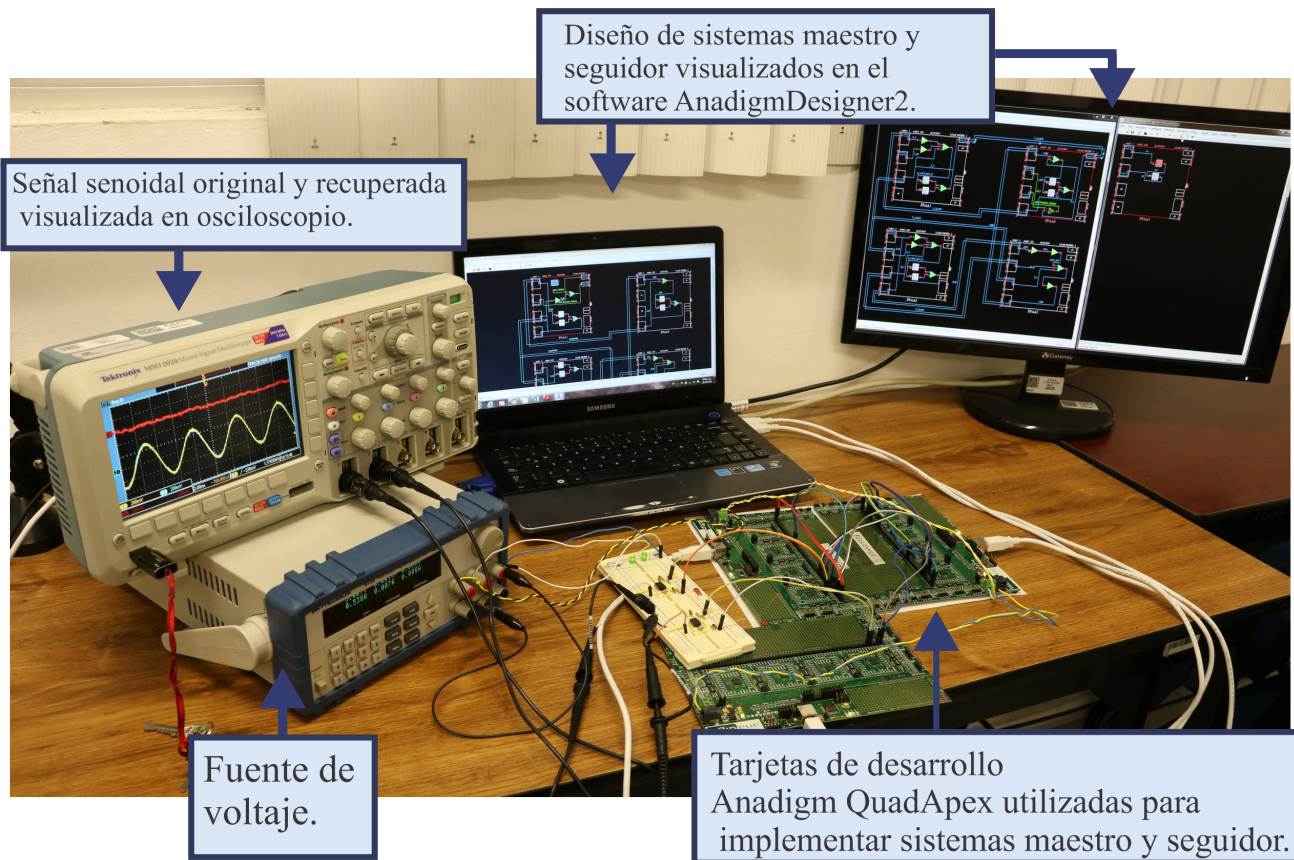
Se utilizó una señal de información, en este caso una señal senoidal de 200mV de amplitud, la cual mediante un sumador implementado con OPAMPs externos se encriptó con la señal caótica de la variable  $Y_M$  esta señal se llevó a un diferenciador, el cual restó a la señal recibida su componente caótica utilizando la variable  $Y_F$ , obteniendo a su salida la señal original recuperada.

En la figura 5.15 se muestra el resultado final obtenido en esta prueba.



**Figura 5.15:** Envío y recepción de una señal de información utilizando caos para su encriptación

En la figura 5.16 se muestra la implementación física de la sincronización de dos osciladores caóticos de Lü de orden fraccionario mediante la forma maestro-seguidor. En la pantalla del ordenador y la pantalla externa se observan los diseños de los sistemas maestro y seguidor en el software AnadigDesigner2. Se utilizaron 3 tarjetas Anadigm QuadApex ya que en la tercera se implementaron sumadores para adicionar caos a la señal senoidal enviada y restarlo en la parte de recepción; sin embargo, esta parte es externa al sistema original de sincronización, en la pantalla del osciloscopio se observa la señal senoidal original y la señal senoidal recuperada después del proceso de desencriptación.



**Figura 5.16:** Implementación física de sincronización entre dos osciladores caóticos de Lü de orden fraccionario y envío y recepción de una señal encriptada con caos.

La sincronización entre dos osciladores caóticos de Lü de orden fraccionario siguiendo el esquema maestro-seguidor y basándonos en el teorema del valor final de la transformada de Laplace fue exitosa; sin embargo, esto fue gracias a que previamente se optimizó un integrador de orden fraccionario al igual que el oscilador caótico de Lü.

Los osciladores caóticos de orden entero o fraccionario tienen la particularidad de ser extremadamente sensibles a variaciones por lo que era indispensable que tanto los integradores de orden fraccionario como el oscilador caótico de orden fraccional de Lü o los bloques de diferencia tuvieran un comportamiento lo más cercano al ideal. El uso de FPAA's brinda versatilidad para implementar este tipo de sistemas así como para reconfigurarlos en tiempo real las veces que sea necesario lo que ayudó en gran medida a la realización de este proyecto de tesis, al ser un sistema en chip (SoC) se redujeron en gran medida factores como ruido o efectos inductivos.

# Capítulo 6

## Conclusiones

En este trabajo de tesis se ha analizado dos topologías para diseñar integradores de orden fraccionario. La primera topología propuesta consistió en utilizar un filtro bilineal el cual al ser configurado mediante polos y ceros permitió implementar integradores cuyos órdenes están dentro del rango  $[0.1, 0.81]$ , órdenes  $\alpha > 0.81$  no fueron posibles de implementar con este filtro debido a que presenta restricciones en las frecuencias de reloj.

Como esta tesis está enfocada en la implementación de osciladores caóticos de orden fraccionario y estos operan en rangos de  $[0.85, 1)$  se propuso una nueva topología que consistió en utilizar un filtro pasa bajas y un filtro pasa altas en paralelo. Esta nueva propuesta nos permitió diseñar integradores de orden fraccionario con un orden incluso de  $\alpha = 0.99$ . La implementación física de este integrador presentó errores en la pendiente obtenida los cuales se analizaron y mediante un ajuste de curvas de tipo polinómico se minimizaron de forma satisfactoria.

La implementación del oscilador caótico de Lü incorporó las dos topologías mencionadas anteriormente para los integradores de orden fraccionario, debido a la extensión del sistema y los recursos necesarios para su implementación, el sistema fue distribuido en 4 FPAAs. Como etapa final se desarrolló la sincronización de dos osciladores caóticos de Lü de orden fraccionario de la forma maestro-seguidor mediante el teorema de valor final de Laplace, ajustando la constante  $K$  a valores altos se obtuvo una sincronización precisa entre las variables  $X$  y  $Y$  de los sistemas maestro y esclavo.

## 6.1. Trabajo futuro

El trabajo futuro que se propone para este proyecto de tesis se resume en los siguientes puntos:

- Implementar la aproximación de segundo orden para integrador de orden fraccionario y caracterizar su respuesta en el dominio de la frecuencia así como verificar el orden  $\alpha$  máximo que se puede implementar con esta aproximación.
- Incorporando la topología para implementar la aproximación de primer orden a un integrador de orden fraccionario con filtro pasa-bajas y pasa-altas implementar diferentes osciladores caóticos de orden fraccionario en FPAA como Lorenz, Chua, Liu, Chen, entre otros.
- Realizar pruebas de sincronización considerando dos sistemas caóticos diferentes, ya sea de diferente tipo, diferente orden o diferente configuración en sus constantes o función saturada, si es un sistema basado en funciones saturadas.
- Profundizar el análisis del envío y recepción de una señal encriptada, probando enviar buses de datos o señales diferentes a la senoidal.

# Apéndice A

## Código MATLAB

**Listing A.1:** Código MATAB para calculo de valores de implementación de integrador de orden fraccionario

```
clc;
clear all;
close all;
format long
prompt = 'Fractional_Integrator_Order?_';
alfa = input(prompt);

%%FACTOR ESCALAMIENTO % % %
f_escal=1/(2*pi*1000); % Para normalizar a un radian sobre segundo f_escal=1

A=((1-alfa)/(1+alfa));
disp('Funcion_de_transferencia_normalizada:');
sys3=tf([A,1],[1,A])

disp('Funcion_de_transferencia_escalada_en_frecuencia:');
sys1=tf([(f_escal)*A,1],[f_escal,A])
sys2=tf([1],[f_escal*1,0]);
disp('Factor_de_escalamiento:');
FE=f_escal^-1

%%Margen de ganancia, Margen de fase y frecuencias.
[Gm,Pm,Wcp] = margin(sys1);

%%Utilizaremos Wcp
%%Wcp is the frequency where the phase margin is measured, which is a 0-dB gain crossing frequency.
%%These frequencies are expressed in radians/TimeUnit.

[mag1]=bode(sys1,Wcp*0.1);
mag1_dB = 20*log10(mag1)

[mag2]=bode(sys1,Wcp);
mag2_dB = 20*log10(mag2)

[mag3]=bode(sys1,Wcp*10);
mag3_dB = 20*log10(mag3)

Caída_Decada = ((mag1_dB-mag2_dB)+(mag2_dB-mag3_dB))/2

zeros=roots(sys1.num{1})
polos=roots(sys1.den{1})

disp('Para_implementar_polos_y_ceros:');

Gh=-A
Fz=1/(A*f_escal*2*pi)
Fp=A/(f_escal*2*pi)
G1=(Fz/Fp)*Gh

disp('Para_implementar_EN_PASA_ALTA_Y_PASA_BAJA:');
disp('LOW_PASS');
F0_low=(A/(2*pi*f_escal))
G_low=(1/(F0_low*2*pi*f_escal))
disp('HIGH_PASS');
F0_high=(A/(2*pi*f_escal))
G_high=(A)

%%sys4=tf([-Gh,-2*pi*Fz*Gh],[1,2*pi*Fp])

disp('Para_implementar_bloques_completo:');
```

```
disp('A/E');
A/f_escal
disp('1/E');
1/f_escal
A
Integration_C=1

disp('Para implementar bloques sintetizado:');
disp('A');
A
Integration_C=1

h=bodeplot(sys1);
p=getoptions(h);
p.FreqUnits='Hz';
setoptions(h,p);
grid on

%% ANANCIA VOLTAGE

disp('Valores de Prueba');
Frecuencia1=(Wcp/(2*pi))*0.1

Vi=0.4
Vo1=(10^(mag1_dB/20))*(Vi)

Frecuencia2=(Wcp/(2*pi))
Vo2=(10^(mag2_dB/20))*(Vi)

Frecuencia3=(Wcp/(2*pi))*10
Vo3=(10^(mag3_dB/20))*(Vi);

%margin(sys1)
%Polo1=(1/A)
%Polo2=-(1/A)
%B_dec1=Polo1-0
%B_dec2=0-Polo2
```

# Apéndice B

## Valores de frecuencia de relojes

### B.1. Frecuencias para implementar integrador de orden fraccionario en filtro bilineal

**Tabla B.1:** Frecuencias de reloj para implementar integrador de orden fraccionario con filtro bilineal configurado mediante polos y ceros, Reloj de sistema 16000 kHz con aproximación de primer orden  $\frac{As+1}{s+A}$ , donde  $A = \frac{1-\alpha}{1+\alpha}$

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
16000	16	1600									
8000	8	800									
4000	4	400									
2666.667	2.666667	266.6667									
2000	2	200									
1600	1.6	160									
1333.333	1.333333	133.3333									
1142.857	1.142857	114.2857									
1000	1	100									
888.8889	0.888889	88.88889									
800	0.8	80									
727.2727	0.727273	72.72727									
666.6667	0.666667	66.66667									
615.3846	0.615385	61.53846									
571.4286	0.571429	57.14286									
533.3333	0.533333	53.33333									
500	0.5	50									
470.5882	0.470588	47.05882									

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
444.4444	0.444444	44.44444									
421.0526	0.421053	42.10526									
400	0.4	40									
380.9524	0.380952	38.09524									
363.6364	0.363636	36.36364									
347.8261	0.347826	34.78261									
333.3333	0.333333	33.33333									
320	0.32	32									
307.6923	0.307692	30.76923									
296.2963	0.296296	29.62963									
285.7143	0.285714	28.57143									
275.8621	0.275862	27.58621									
266.6667	0.266667	26.66667									
258.0645	0.258065	25.80645									
250	0.25	25									
242.4242	0.242424	24.24242									
235.2941	0.235294	23.52941									
228.5714	0.228571	22.85714									
222.2222	0.222222	22.22222									
216.2162	0.216216	21.62162									
210.5263	0.210526	21.05263									
205.1282	0.205128	20.51282									
200	0.2	20									
195.122	0.195122	19.5122									
190.4762	0.190476	19.04762									
186.0465	0.186047	18.60465									
181.8182	0.181818	18.18182									
177.7778	0.177778	17.77778									
173.913	0.173913	17.3913									
170.2128	0.170213	17.02128									
166.6667	0.166667	16.66667									
163.2653	0.163265	16.32653									
160	0.16	16									
156.8627	0.156863	15.68627									
153.8462	0.153846	15.38462									

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
150.9434	0.150943	15.09434									
148.1481	0.148148	14.81481									
145.4545	0.145455	14.54545									
142.8571	0.142857	14.28571									
140.3509	0.140351	14.03509									
137.931	0.137931	13.7931									
135.5932	0.135593	13.55932									
133.3333	0.133333	13.33333									
131.1475	0.131148	13.11475									
129.0323	0.129032	12.90323									
126.9841	0.126984	12.69841									
125	0.125	12.5									
123.0769	0.123077	12.30769									
121.2121	0.121212	12.12121									
119.403	0.119403	11.9403									
117.6471	0.117647	11.76471									
115.942	0.115942	11.5942									
114.2857	0.114286	11.42857									
112.6761	0.112676	11.26761									
111.1111	0.111111	11.11111									
109.589	0.109589	10.9589									
108.1081	0.108108	10.81081									
106.6667	0.106667	10.66667									
105.2632	0.105263	10.52632									
103.8961	0.103896	10.38961									
102.5641	0.102564	10.25641									
101.2658	0.101266	10.12658									
100	0.1	10									
98.76543	0.098765	9.876543									
97.56098	0.097561	9.756098									
96.38554	0.096386	9.638554									
95.2381	0.095238	9.52381									
94.11765	0.094118	9.411765									
93.02326	0.093023	9.302326									
91.95402	0.091954	9.195402									

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
90.90909	0.090909	9.090909									
89.88764	0.089888	8.988764									
88.88889	0.088889	8.888889									
87.91209	0.087912	8.791209									
86.95652	0.086957	8.695652									
86.02151	0.086022	8.602151									
85.10638	0.085106	8.510638									
84.21053	0.084211	8.421053									
83.33333	0.083333	8.333333									
82.47423	0.082474	8.247423									
81.63265	0.081633	8.163265									
80.80808	0.080808	8.080808									
80	0.08	8									
79.20792	0.079208	7.920792									
78.43137	0.078431	7.843137									
77.6699	0.07767	7.76699									
76.92308	0.076923	7.692308									
76.19048	0.07619	7.619048									
75.4717	0.075472	7.54717									
74.76636	0.074766	7.476636									
74.07407	0.074074	7.407407									
73.3945	0.073394	7.33945									
72.72727	0.072727	7.272727									
72.07207	0.072072	7.207207									
71.42857	0.071429	7.142857									
70.79646	0.070796	7.079646									
70.17544	0.070175	7.017544									
69.56522	0.069565	6.956522									
68.96552	0.068966	6.896552									
68.37607	0.068376	6.837607									
67.79661	0.067797	6.779661									
67.22689	0.067227	6.722689									
66.66667	0.066667	6.666667									
66.1157	0.066116	6.61157									
65.57377	0.065574	6.557377									

Continúa en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
65.04065	0.065041	6.504065									
64.51613	0.064516	6.451613									
64	0.064	6.4									
63.49206	0.063492	6.349206									
62.99213	0.062992	6.299213									
62.5	0.0625	6.25									
62.0155	0.062016	6.20155									
61.53846	0.061538	6.153846									
61.0687	0.061069	6.10687									
60.60606	0.060606	6.060606									
60.15038	0.06015	6.015038									
59.70149	0.059701	5.970149									
59.25926	0.059259	5.925926									
58.82353	0.058824	5.882353									
58.39416	0.058394	5.839416									
57.97101	0.057971	5.797101									
57.55396	0.057554	5.755396									
57.14286	0.057143	5.714286									
56.73759	0.056738	5.673759									
56.33803	0.056338	5.633803									
55.94406	0.055944	5.594406									
55.55556	0.055556	5.555556									
55.17241	0.055172	5.517241									
54.79452	0.054795	5.479452									
54.42177	0.054422	5.442177									
54.05405	0.054054	5.405405									
53.69128	0.053691	5.369128									
53.33333	0.053333	5.333333									
52.98013	0.05298	5.298013									
52.63158	0.052632	5.263158									
52.28758	0.052288	5.228758									
51.94805	0.051948	5.194805									
51.6129	0.051613	5.16129									
51.28205	0.051282	5.128205									
50.95541	0.050955	5.095541									

Continúa en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
50.63291	0.050633	5.063291									
50.31447	0.050314	5.031447									
50	0.05	5									
49.68944	0.049689	4.968944									
49.38272	0.049383	4.938272									
49.07975	0.04908	4.907975									
48.78049	0.04878	4.878049									
48.48485	0.048485	4.848485									
48.19277	0.048193	4.819277									
47.90419	0.047904	4.790419									
47.61905	0.047619	4.761905									
47.33728	0.047337	4.733728									
47.05882	0.047059	4.705882									
46.78363	0.046784	4.678363									
46.51163	0.046512	4.651163									
46.24277	0.046243	4.624277									
45.97701	0.045977	4.597701									
45.71429	0.045714	4.571429									
45.45455	0.045455	4.545455									
45.19774	0.045198	4.519774									
44.94382	0.044944	4.494382									
44.69274	0.044693	4.469274									
44.44444	0.044444	4.444444									
44.1989	0.044199	4.41989									
43.95604	0.043956	4.395604									
43.71585	0.043716	4.371585									
43.47826	0.043478	4.347826									
43.24324	0.043243	4.324324									
43.01075	0.043011	4.301075									
42.78075	0.042781	4.278075									
42.55319	0.042553	4.255319									
42.32804	0.042328	4.232804									
42.10526	0.042105	4.210526									
41.88482	0.041885	4.188482									
41.66667	0.041667	4.166667									

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
41.45078	0.041451	4.145078									
41.23711	0.041237	4.123711									
41.02564	0.041026	4.102564									
40.81633	0.040816	4.081633									
40.60914	0.040609	4.060914									
40.40404	0.040404	4.040404									
40.20101	0.040201	4.020101									
40	0.04	4									
39.801	0.039801	3.9801									
39.60396	0.039604	3.960396									
39.40887	0.039409	3.940887									
39.21569	0.039216	3.921569									
39.02439	0.039024	3.902439									
38.83495	0.038835	3.883495									
38.64734	0.038647	3.864734									
38.46154	0.038462	3.846154									
38.27751	0.038278	3.827751									
38.09524	0.038095	3.809524									
37.91469	0.037915	3.791469									
37.73585	0.037736	3.773585									
37.55869	0.037559	3.755869									
37.38318	0.037383	3.738318									
37.2093	0.037209	3.72093									
37.03704	0.037037	3.703704									
36.86636	0.036866	3.686636									
36.69725	0.036697	3.669725									
36.52968	0.03653	3.652968									
36.36364	0.036364	3.636364									
36.1991	0.036199	3.61991									
36.03604	0.036036	3.603604									
35.87444	0.035874	3.587444									
35.71429	0.035714	3.571429									
35.55556	0.035556	3.555556									
35.39823	0.035398	3.539823									
35.24229	0.035242	3.524229									

Continúa en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.81
35.08772	0.035088	3.508772									
34.9345	0.034934	3.49345									
34.78261	0.034783	3.478261									
34.63203	0.034632	3.463203									
34.48276	0.034483	3.448276									
34.33476	0.034335	3.433476									
34.18803	0.034188	3.418803									
34.04255	0.034043	3.404255									
33.89831	0.033898	3.389831									
33.75527	0.033755	3.375527									
33.61345	0.033613	3.361345									
33.4728	0.033473	3.34728									
33.33333	0.033333	3.333333									
33.19502	0.033195	3.319502									
33.05785	0.033058	3.305785									
32.92181	0.032922	3.292181									
32.78689	0.032787	3.278689									
32.65306	0.032653	3.265306									
32.52033	0.03252	3.252033									
32.38866	0.032389	3.238866									
32.25806	0.032258	3.225806									
32.12851	0.032129	3.212851									
32	0.032	3.2									
31.87251	0.031873	3.187251									
31.74603	0.031746	3.174603									
31.62055	0.031621	3.162055									
31.49606	0.031496	3.149606									
31.37255	0.031373	3.137255									

Fin de la tabla

## B.2. Frecuencias para implementar integrador de orden fraccionario utilizando filtros pasa altas y pasa bajas

### B.2.1. Reloj del sistema 16000 kHz

**Tabla B.2:** Frecuencias de reloj para implementar integrador de orden fraccionario con filtros pasa-bajo y paso-alto, Reloj de sistema 16000 kHz con aproximación de primer orden  $\frac{As+1}{s+A}$ , donde  $A = \frac{1-\alpha}{1+\alpha}$

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
16000	16	1600													
8000	8	800													
4000	4	400													
2666.667	2.666667	266.6667													
2000	2	200													
1600	1.6	160													
1333.333	1.333333	133.3333													
1142.857	1.142857	114.2857													
1000	1	100													
888.8889	0.888889	88.88889													
800	0.8	80													
727.2727	0.727273	72.72727													
666.6667	0.666667	66.66667													
615.3846	0.615385	61.53846													
571.4286	0.571429	57.14286													
533.3333	0.533333	53.33333													
500	0.5	50													
470.5882	0.470588	47.05882													
444.4444	0.444444	44.44444													
421.0526	0.421053	42.10526													
400	0.4	40													
380.9524	0.380952	38.09524													
363.6364	0.363636	36.36364													
347.8261	0.347826	34.78261													
333.3333	0.333333	33.33333													
320	0.32	32													
307.6923	0.307692	30.76923													
296.2963	0.296296	29.62963													
285.7143	0.285714	28.57143													
275.8621	0.275862	27.58621													
266.6667	0.266667	26.66667													

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
258.0645	0.258065	25.80645													
250	0.25	25													
242.4242	0.242424	24.24242													
235.2941	0.235294	23.52941													
228.5714	0.228571	22.85714													
222.2222	0.222222	22.22222													
216.2162	0.216216	21.62162													
210.5263	0.210526	21.05263													
205.1282	0.205128	20.51282													
200	0.2	20													
195.122	0.195122	19.5122													
190.4762	0.190476	19.04762													
186.0465	0.186047	18.60465													
181.8182	0.181818	18.18182													
177.7778	0.177778	17.77778													
173.913	0.173913	17.3913													
170.2128	0.170213	17.02128													
166.6667	0.166667	16.66667													
163.2653	0.163265	16.32653													
160	0.16	16													
156.8627	0.156863	15.68627													
153.8462	0.153846	15.38462													
150.9434	0.150943	15.09434													
148.1481	0.148148	14.81481													
145.4545	0.145455	14.54545													
142.8571	0.142857	14.28571													
140.3509	0.140351	14.03509													
137.931	0.137931	13.7931													
135.5932	0.135593	13.55932													
133.3333	0.133333	13.33333													
131.1475	0.131148	13.11475													
129.0323	0.129032	12.90323													
126.9841	0.126984	12.69841													
125	0.125	12.5													
123.0769	0.123077	12.30769													

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
121.2121	0.121212	12.12121													
119.403	0.119403	11.9403													
117.6471	0.117647	11.76471													
115.942	0.115942	11.5942													
114.2857	0.114286	11.42857													
112.6761	0.112676	11.26761													
111.1111	0.111111	11.11111													
109.589	0.109589	10.9589													
108.1081	0.108108	10.81081													
106.6667	0.106667	10.66667													
105.2632	0.105263	10.52632													
103.8961	0.103896	10.38961													
102.5641	0.102564	10.25641													
101.2658	0.101266	10.12658													
100	0.1	10													
98.76543	0.098765	9.876543													
97.56098	0.097561	9.756098													
96.38554	0.096386	9.638554													
95.2381	0.095238	9.52381													
94.11765	0.094118	9.411765													
93.02326	0.093023	9.302326													
91.95402	0.091954	9.195402													
90.90909	0.090909	9.090909													
89.88764	0.089888	8.988764													
88.88889	0.088889	8.888889													
87.91209	0.087912	8.791209													
86.95652	0.086957	8.695652													
86.02151	0.086022	8.602151													
85.10638	0.085106	8.510638													
84.21053	0.084211	8.421053													
83.33333	0.083333	8.333333													
82.47423	0.082474	8.247423													
81.63265	0.081633	8.163265													
80.80808	0.080808	8.080808													
80	0.08	8													

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
79.20792	0.079208	7.920792													
78.43137	0.078431	7.843137													
77.6699	0.07767	7.76699													
76.92308	0.076923	7.692308													
76.19048	0.07619	7.619048													
75.4717	0.075472	7.54717													
74.76636	0.074766	7.476636													
74.07407	0.074074	7.407407													
73.3945	0.073394	7.33945													
72.72727	0.072727	7.272727													
72.07207	0.072072	7.207207													
71.42857	0.071429	7.142857													
70.79646	0.070796	7.079646													
70.17544	0.070175	7.017544													
69.56522	0.069565	6.956522													
68.96552	0.068966	6.896552													
68.37607	0.068376	6.837607													
67.79661	0.067797	6.779661													
67.22689	0.067227	6.722689													
66.66667	0.066667	6.666667													
66.1157	0.066116	6.61157													
65.57377	0.065574	6.557377													
65.04065	0.065041	6.504065													
64.51613	0.064516	6.451613													
64	0.064	6.4													
63.49206	0.063492	6.349206													
62.99213	0.062992	6.299213													
62.5	0.0625	6.25													
62.0155	0.062016	6.20155													
61.53846	0.061538	6.153846													
61.0687	0.061069	6.10687													
60.60606	0.060606	6.060606													
60.15038	0.06015	6.015038													
59.70149	0.059701	5.970149													
59.25926	0.059259	5.925926													

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
58.82353	0.058824	5.882353													
58.39416	0.058394	5.839416													
57.97101	0.057971	5.797101													
57.55396	0.057554	5.755396													
57.14286	0.057143	5.714286													
56.73759	0.056738	5.673759													
56.33803	0.056338	5.633803													
55.94406	0.055944	5.594406													
55.55556	0.055556	5.555556													
55.17241	0.055172	5.517241													
54.79452	0.054795	5.479452													
54.42177	0.054422	5.442177													
54.05405	0.054054	5.405405													
53.69128	0.053691	5.369128													
53.33333	0.053333	5.333333													
52.98013	0.05298	5.298013													
52.63158	0.052632	5.263158													
52.28758	0.052288	5.228758													
51.94805	0.051948	5.194805													
51.6129	0.051613	5.16129													
51.28205	0.051282	5.128205													
50.95541	0.050955	5.095541													
50.63291	0.050633	5.063291													
50.31447	0.050314	5.031447													
50	0.05	5													
49.68944	0.049689	4.968944													
49.38272	0.049383	4.938272													
49.07975	0.04908	4.907975													
48.78049	0.04878	4.878049													
48.48485	0.048485	4.848485													
48.19277	0.048193	4.819277													
47.90419	0.047904	4.790419													
47.61905	0.047619	4.761905													
47.33728	0.047337	4.733728													
47.05882	0.047059	4.705882													

Continúa en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
46.78363	0.046784	4.678363													
46.51163	0.046512	4.651163													
46.24277	0.046243	4.624277													
45.97701	0.045977	4.597701													
45.71429	0.045714	4.571429													
45.45455	0.045455	4.545455													
45.19774	0.045198	4.519774													
44.94382	0.044944	4.494382													
44.69274	0.044693	4.469274													
44.44444	0.044444	4.444444													
44.1989	0.044199	4.41989													
43.95604	0.043956	4.395604													
43.71585	0.043716	4.371585													
43.47826	0.043478	4.347826													
43.24324	0.043243	4.324324													
43.01075	0.043011	4.301075													
42.78075	0.042781	4.278075													
42.55319	0.042553	4.255319													
42.32804	0.042328	4.232804													
42.10526	0.042105	4.210526													
41.88482	0.041885	4.188482													
41.66667	0.041667	4.166667													
41.45078	0.041451	4.145078													
41.23711	0.041237	4.123711													
41.02564	0.041026	4.102564													
40.81633	0.040816	4.081633													
40.60914	0.040609	4.060914													
40.40404	0.040404	4.040404													
40.20101	0.040201	4.020101													
40	0.04	4													
39.801	0.039801	3.9801													
39.60396	0.039604	3.960396													
39.40887	0.039409	3.940887													
39.21569	0.039216	3.921569													
39.02439	0.039024	3.902439													

Continua en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
38.83495	0.038835	3.883495													
38.64734	0.038647	3.864734													
38.46154	0.038462	3.846154													
38.27751	0.038278	3.827751													
38.09524	0.038095	3.809524													
37.91469	0.037915	3.791469													
37.73585	0.037736	3.773585													
37.55869	0.037559	3.755869													
37.38318	0.037383	3.738318													
37.2093	0.037209	3.72093													
37.03704	0.037037	3.703704													
36.86636	0.036866	3.686636													
36.69725	0.036697	3.669725													
36.52968	0.03653	3.652968													
36.36364	0.036364	3.636364													
36.1991	0.036199	3.61991													
36.03604	0.036036	3.603604													
35.87444	0.035874	3.587444													
35.71429	0.035714	3.571429													
35.55556	0.035556	3.555556													
35.39823	0.035398	3.539823													
35.24229	0.035242	3.524229													
35.08772	0.035088	3.508772													
34.9345	0.034934	3.49345													
34.78261	0.034783	3.478261													
34.63203	0.034632	3.463203													
34.48276	0.034483	3.448276													
34.33476	0.034335	3.433476													
34.18803	0.034188	3.418803													
34.04255	0.034043	3.404255													
33.89831	0.033898	3.389831													
33.75527	0.033755	3.375527													
33.61345	0.033613	3.361345													
33.4728	0.033473	3.34728													
33.33333	0.033333	3.333333													

Continúa en la página siguiente.

Frecuencia de reloj [kHz]	Frecuencia mínima [kHz]	Frecuencia máxima [kHz]	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91	0.92	0.93
33.19502	0.033195	3.319502													
33.05785	0.033058	3.305785													
32.92181	0.032922	3.292181													
32.78689	0.032787	3.278689													
32.65306	0.032653	3.265306													
32.52033	0.03252	3.252033													
32.38866	0.032389	3.238866													
32.25806	0.032258	3.225806													
32.12851	0.032129	3.212851													
32	0.032	3.2													
31.87251	0.031873	3.187251													
31.74603	0.031746	3.174603													
31.62055	0.031621	3.162055													
31.49606	0.031496	3.149606													
31.37255	0.031373	3.137255													

Fin de la tabla

# Bibliografía

- [1] A. Buscarino, L. Fortuna, M. Frasca, G. Sciuto *et al.*, *A concise guide to chaotic electronic circuits*. Springer, 2014.
- [2] A. Cornejo, “Complejidad y caos. guía para la administración del siglo xxi,” *Recuperado de: <http://www.scribd.com/doc/22615328/Complejidad-y-Caos-en-Las-Organizaciones-Ing-Alfonso-Cornejo>*, 2004.
- [3] S. Das, *Functional fractional calculus*. Springer Science & Business Media, 2011.
- [4] M. Guía-Calderón, J. Rosales-García, R. Guzmán-Cabrera, A. González-Parada, and J. Álvarez-Jaime, “El cálculo diferencial e integral fraccionario y sus aplicaciones,” *Acta universitaria*, vol. 25, no. 2, pp. 20–27, 2015.
- [5] B. Krishna, “Studies on fractional order differentiators and integrators: A survey,” *Signal Processing*, vol. 91, no. 3, pp. 386–426, 2011.
- [6] I. S. Jesus and J. T. Machado, “Development of fractional order capacitors based on electrolyte processes,” *Nonlinear Dynamics*, vol. 56, no. 1-2, pp. 45–55, 2009.
- [7] K. Biswas, S. Sen, and P. K. Dutta, “Realization of a constant phase element and its performance study in a differentiator circuit,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 9, pp. 802–806, 2006.
- [8] Y. Chen, I. Petras, and D. Xue, “Fractional order control-a tutorial,” in *2009 American control conference*. IEEE, 2009, pp. 1397–1411.
- [9] I. Podlubny, I. Petráš, B. M. Vinagre, P. O’leary, and L. Dorčák, “Analogue realizations of fractional-order controllers,” *Nonlinear dynamics*, vol. 29, no. 1-4, pp. 281–296, 2002.
- [10] J. Petrzela, R. Sotner, and M. Guzan, “Implementation of constant phase elements using low-q band-pass and band-reject filtering sections,” in *2016 International Conference on Applied Electronics (AE)*. IEEE, 2016, pp. 205–210.

- 
- [11] A. Antoniou, “Floating negative-impedance converters,” *IEEE transactions on circuit theory*, vol. 19, no. 2, pp. 209–212, 1972.
- [12] L. Dorčák, J. Terpák, I. Petráš, J. Valsa, and E. Gonzalez, “Comparison of the electronic realization of the fractional-order system and its model,” in *Proceedings of the 13th International Carpathian Control Conference (ICCC)*. IEEE, 2012, pp. 119–124.
- [13] A. Charef, “Analogue realisation of fractional-order integrator, differentiator and fractional  $\pi\lambda d\mu$  controller,” *IEE Proceedings-Control Theory and Applications*, vol. 153, no. 6, pp. 714–720, 2006.
- [14] H. Richard, *Fractional calculus: an introduction for physicists*. World Scientific, 2014.
- [15] L. Angel and J. Viola, “Design and statistical robustness analysis of fopid, iopid and simc pid controllers applied to a motor-generator system,” *IEEE Latin America Transactions*, vol. 13, no. 12, pp. 3724–3734, 2015.
- [16] C. K. Alexander and M. N. Sadiku, *Fundamentos de circuitos eléctricos (5a)*. McGraw Hill Mexico, 2013.
- [17] I. ANADIGM, “An231e04 datasheet rev 1.2, 3rd generation dynamically reconfigurable dpasp.” 2014.
- [18] N. INSTRUMENTS, “Ni educational laboratory virtual instrumentation suite ii series (ni elvistm ii series) user manual,” 2011.
- [19] G. Perdikaris, *Computer Controlled Systems: Theory and Applications*, ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer Netherlands, 1991. [Online]. Available: <https://books.google.com.mx/books?id=Gzxh0TYX3TEC>
- [20] D. Davis and E. Patronis, *Sound system engineering*. Routledge, 2012.
- [21] W. Xue, Y. Li, S. Cang, H. Jia, and Z. Wang, “Chaotic behavior and circuit implementation of a fractional-order permanent magnet synchronous motor model,” *Journal of the franklin institute*, vol. 352, no. 7, pp. 2887–2898, 2015.
- [22] K. Sun, X. Wang, and J. C. Sprott, “Bifurcations and chaos in fractional-order simplified lorenz system,” *International Journal of bifurcation and Chaos*, vol. 20, no. 04, pp. 1209–1219, 2010.
- [23] J. G. Lu and G. Chen, “A note on the fractional-order chen system,” *Chaos, Solitons & Fractals*, vol. 27, no. 3, pp. 685–688, 2006.
-

- 
- [24] H.-K. Chen, L.-J. Sheu, L.-M. Tam, and S.-K. Lao, “A new finding of the existence of feigenbaum’s constants in the fractional-order chen–lee system,” *Nonlinear Dynamics*, vol. 68, no. 4, pp. 589–599, 2012.
- [25] H.-Y. Jia, Z.-Q. Chen, and G.-Y. Qi, “Topological horseshoe analysis and circuit realization for a fractional-order lü system,” *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 203–212, 2013.
- [26] I. Petráš, *Fractional-order nonlinear systems: modeling, analysis and simulation*. Springer Science & Business Media, 2011.
- [27] I. ANADIGM, “Anadigmapex dpasp family user manual,” 2011.
- [28] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2013.
- [29] A. Inc., “Anadigm quadapex development board,” 2016.
- [30] E. Zambrano-Serrano, J. Muñoz-Pacheco, L. Gómez-Pavón, A. Luis-Ramos, and G. Chen, “Synchronization in a fractional-order model of pancreatic  $\beta$ -cells,” *The European Physical Journal Special Topics*, vol. 227, no. 7-9, pp. 907–919, 2018.
- [31] J. D. Moreno Posada, C. A. Suárez Parra, R. E. Huérfano Ortiz, F. A. Parra Fuentes, and E. I. Amaya Barrera, “Modelo de encriptación simétrica basada en atractores caóticos.”