



**UNIVERSIDAD
DEL AZUAY**

DEPARTAMENTO DE POSGRADOS

Acoplamiento neuronal en un ataque epiléptico utilizando
aproximación al EEG con el modelo Kuramoto

Master en Matemática Aplicada

Autora: Andrea Maruri Padilla

Director: Dr. Felipe Torres Espinoza

Cuenca, Ecuador 2023

DEDICATORIA

A mi hermana Cris, por todo su apoyo, dedicación y aliento que me dio durante todo el desarrollo de la maestría.

AGRADECIMIENTO

A mi familia, por su preocupación e involucramiento durante todo el proceso, a mi novio que siempre tuvo las palabras exactas para animarme en los momentos que más lo necesité y por sobre todo a mi tutor, académico excelso que desde el principio me permitió trabajar en el objetivo de relacionar las disciplinas de la Psicología y la Matemática que constituyen mi pasión.

RESUMEN

El presente estudio tiene como foco el comportamiento neuronal en un ataque epiléptico utilizando una aproximación al EEG con el modelo Kuramoto. Se aporta evidencia para la demostración de que la epilepsia es un fenómeno de sincronización que puede ser explicado mediante osciladores acoplados. Para cumplir con tal objetivo se analizaron los electroencefalogramas de 22 pacientes pediátricos. Los datos fueron analizados con el lenguaje de programación *Python*. Los resultados obtenidos facilitan la comprensión de este fenómeno a través del parámetro de acoplamiento que intervienen en la evaluación dinámica de la epilepsia y la medición de la sincronía en tal dinámica.

Palabras clave: señal, filtro, sincronización, acoplamiento, osciladores, reconstrucción, modelo de Kuramoto.

ABSTRACT

This study focuses on detecting the neuronal behavior during an epileptic seizure by using an approach to the EEG based on the Kuramoto model. Evidence was provided by supporting the idea that epilepsy is a synchronization phenomenon that can be explained through coupled oscillators. To achieve this objective, the electroencephalograms of 22 pediatric patients were analyzed. The data was processed by using the Python programming language. The obtained results help in understanding this phenomenon through the coupling parameter involved in the dynamic assessment of epilepsy and the mediation of synchrony in such dynamics.

Keywords: signal, filter, synchronization, coupling, oscillators, reconstruction, Kuramoto model

Translated by:



Andrea Maruri Padilla



ÍNDICE DE CONTENIDOS

DEDICATORIA	ii
AGRADECIMIENTO	iii
RESUMEN.....	iv
ABSTRACT	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE ANEXOS	viii
INTRODUCCIÓN.....	9
1 REVISIÓN DE LA LITERATURA.....	10
1.1 Fundamentos Teóricos.....	10
1.1.1 Epilepsia.....	10
1.1.2 Diferencia entre epilepsia y crisis epiléptica	10
1.1.3 Ejemplificación de un estado basal y un estado epiléptico.....	10
1.1.4 Sincronización	11
1.1.5 Tipos de sincronización:.....	11
1.1.6 La epilepsia como sincronización.....	12
1.2 Estado del Arte	12
1.2.1 Comportamiento de las neuronas y origen de la epilepsia.....	12
2 MATERIALES Y MÉTODOS	13
2.1 Modelos para modelar la sincronización	13
2.1.1 Generalidades	13
2.1.2 Movimiento Armónico Simple	14
2.1.3 El modelo de Kuramoto.....	15
2.1.4 Fórmula y descripción de elementos	16

2.1.5 Medición del parámetro de sincronización 17

2.1.5.1 Parámetro de orden $r(t)$ 17

2.1.5.2 Valores de acoplamiento 18

2.1.5.3 Generalización del Modelo de Kuramoto 20

2.1.6 Utilidad de la transformada de Fourier..... 20

3 RESULTADOS 22

4 DISCUSIÓN 26

5 CONCLUSIONES 28

6 BIBLIOGRAFÍA 29

7 ANEXOS..... 31

ÍNDICE DE FIGURAS

Figura 1 Ondas mostradas en el electroencefalograma en una crisis epiléptica 13

Figura 2 Movimiento Armónico Simple 15

Figura 3 Modelo de Kuramoto con 6 osciladores..... 17

Figura 4 La flecha representa el número complejo $rei\psi$ 18

Figura 5 Ilustración esquemática de la evolución típica de $r(t)$ vista en simulaciones numéricas del modelo de Kuramoto..... 19

Figura 6 Dependencia de la coherencia en estado estacionario de r^∞ de la fuerza de acoplamiento k 20

Figura 7 Electroencefalograma del ataque de epilepsia 22

Figura 8 Medición de la Transformada de Fourier 23

Figura 9 Señal reconstruida sin modelo de Kuramoto 23

Figura 10 Señal reconstruida con modelo de Kuramoto..... 24

Figura 11 Obtención y comparación del parámetro de orden para el modelo original y con Kuramoto..... 24

Figura 12 Parámetro de orden en función de la constante de acoplamiento. 26

ÍNDICE DE TABLAS

Tabla1 Comparación entre el parámetro de r y los valores de acoplamiento	25
---	----

ÍNDICE DE ANEXOS

Anexo1	31
Anexo2	50
Anexo3	41
Anexo4	49

INTRODUCCIÓN

La epilepsia es una patología neuronal caracterizada por ataques o crisis que pueden derivar en pérdida del conocimiento, convulsiones, o pérdida del tono postural. Estos ataques se producen por el funcionamiento anormal de poblaciones neuronales. Durante un ataque epiléptico incrementa la cantidad de neuronas que presentan potenciales de acción, lo cual se relaciona con mayor sincronía en la actividad neuronal.

La presente investigación pretende demostrar la relación que existe entre un ataque epiléptico y la sincronización de osciladores a los que se hace referencia en el Modelo de Kuramoto. El modelo asume que cuando los osciladores interactúan con la misma fuerza se produce la sincronía entre ellos. Además, dicho modelo permite generalizar las características de la dinámica de los osciladores; es por eso que se pretende establecer la relación entre la sincronía con lo que sucede en el cerebro de una persona al momento de una crisis epiléptica, por la descarga eléctrica anormal de las células de una zona del cerebro. Para este propósito se plantea analizar los EEG de pacientes con convulsiones epilépticas críticas, las mismas que son transformadas mediante algoritmos a otro dominio, el dominio de frecuencia para tratar y analizar dichos resultados de forma alternativa. La metodología usada permite manipular el valor de parámetros de acoplamiento y de esta manera comparar los valores numéricos obtenidos. Y mediante este proceso comprobar las suposiciones de los límites que deben tener estos parámetros para producir el ataque; por lo tanto, es necesario realizar la medición de las ondas durante tres etapas del ataque para verificar la señal de acoplamiento que existe entre ellas. Finalmente, según la similitud con la sincronía en los datos de los pacientes, determinar el valor de los parámetros requeridos para la ocurrencia de un ataque epiléptico.

1 REVISIÓN DE LA LITERATURA

1.1 Fundamentos Teóricos

1.1.1 Epilepsia

Según la investigación presentada por Gerster et al (2020), la epilepsia es un trastorno neurológico que afecta a casi 70 de cada millón de personas en todo el mundo. Las personas con epilepsia experimentan convulsiones caracterizadas por una aparición transitoria de signos y/o síntomas debido a una actividad neuronal excesiva o sincrónica en el cerebro.

1.1.2 Diferencia entre epilepsia y crisis epiléptica

Resumiendo los estudios presentados por Izquierdo (2018), la crisis epiléptica son episodios de aparición súbita de una actividad anormal en un grupo de neuronas en el cerebro, este grupo de neuronas, con esta actividad anormal se denomina foco epiléptico, los síntomas clínicos de la crisis dependerán de la parte del cerebro en donde esté localizado el foco, si la actividad anormal se propaga al resto del cerebro, se producirá pérdida de consciencia y la pérdida del tono postural con caída al suelo; por lo tanto, hay que saber diferenciar entre epilepsia y crisis epiléptica. La epilepsia es una enfermedad que además de las crisis epilépticas, va a tener otros síntomas como: retraso mental, parálisis de extremidades entre otros. Las crisis epilépticas pueden desencadenarse no solo por lesiones en el cerebro, sino también por alteraciones en el metabolismo, las causas de la epilepsia y de las crisis epilépticas pueden ser muy variadas dependiendo de la edad del sujeto.

1.1.3 Ejemplificación de un estado basal y un estado epiléptico

Tomando como referencia lo que proponen en su trabajo López et al (2009) la actividad epiléptica se caracteriza por tres períodos:

El periodo ictal o ictus: este corresponde a la crisis epiléptica per se, es un evento intermitente y breve que puede durar desde segundos hasta minutos. Las crisis recurrentes inducen mecanismos homeostáticos que actúan para disminuir la hiperexcitabilidad. Estos mecanismos terminan con el evento ictal, de tal manera que previene la propagación de la actividad epiléptica y mantiene el estado interictal. Los factores fisiológicos que influyen en la transición del estado interictal al ictal son la sincronización y la excitabilidad neuronal (López, y otros, 2009).

El periodo interictal: es el transcurrido entre crisis y crisis, durante este tiempo se presentan mecanismos inhibitorios para disminuir la hiperexcitabilidad epiléptica de tal manera que no aparezcan nuevas crisis (López, y otros, 2009).

El periodo postictal: se presenta después de la crisis epiléptica y cuya duración puede variar desde unos minutos hasta días, se caracteriza por inmovilidad corporal y representa el umbral de una crisis epiléptica (López, y otros, 2009).

El periodo preictal: es la transición al periodo ictal; por último, se debe mencionar que estas fases pueden ser resumidas en dos periodos: estado basal y el estado de crisis epiléptica (López, y otros, 2009).

1.1.4 Sincronización.

Para Rivera (2020), la sincronización es un fenómeno físico que fue estudiado por primera vez en el siglo XVII, el astrónomo-físico-matemático Huygens, también inventor del reloj de péndulo y creador de la teoría ondulatoria de la luz. Huygens una vez observó que dos de sus relojes de pared, colocados en el mismo muro, se sincronizaban en el balanceo de sus respectivos péndulos, sin embargo, otros relojes en otras habitaciones desde donde no se escuchaba el rítmico tic-tac no compartían simultaneidad de los oscilantes pesos.

A partir de ese momento ha sido un misterio el por qué los péndulos funcionan a ritmo acompasado si se encuentran en el mismo lugar. En experimentos actuales con metrónomos usados por los músicos para marcar el ritmo a diversas frecuencias de oscilaciones por minuto, se puede observar que si estos están colocados en una misma plataforma de tipo semimóvil al cabo de unos minutos alcanzan simultaneidad (Rivera, 2020).

En la naturaleza y en nuestra vida cotidiana se manifiestan fenómenos de sincronización instantánea: los destellos de luciérnagas en el campo, los chirridos de los grillos, las mareas, los aplausos y el ritmo cardíaco. Cada uno de estos fenómenos antes mencionados muestran una tendencia al orden. Y en particular, el disparo de potenciales de acción de las neuronas también es un fenómeno que presenta tendencia hacia la sincronía (Rivera, 2020).

1.1.5 Tipos de sincronización:

De igual manera, según Rivera (2020) existen diferentes tipos de sincronización, entre las cuales podemos mencionar:

Sincronización de trayectorias: es una técnica utilizada en Ingeniería de control para hacer que un sistema siga una trayectoria deseada. Este tipo de sincronización se utiliza en sistema de control de movimiento (Rivera, 2020).

Sincronización de comportamiento: se refiere al proceso mediante el cual dos o más sistemas o individuos coordinan sus acciones o comportamientos para realizar una tarea conjunta de manera efectiva; además, la sincronización de comportamiento en la epilepsia hace referencia

a la tendencia de las neuronas epilépticas a disparar en patrones sincronizados o coordinados, lo que puede llevar a convulsiones y otros síntomas asociados con la epilepsia (Rivera, 2020).

1.1.6 La epilepsia como sincronización.

1.2 Estado del Arte

1.2.1 Comportamiento de las neuronas y origen de la epilepsia.

Para esta definición, vamos a tomar como referencia lo propuesto por García y Simón (2016), trabajo en el que manifiestan que, la hiperexcitabilidad neuronal de las crisis epilépticas es consecuencia de un desequilibrio entre la neurotransmisión excitatoria y la inhibitoria, donde predominó la parte excitatoria. Las crisis epilépticas se desarrollan de la siguiente manera:

Inicio de la actividad epiléptica: se debe a cambios paroxísticos de despolarización que sufren las neuronas. La actividad epiléptica se produce cuando hay una alteración en la actividad eléctrica normal del cerebro, lo que resulta en una descarga de señales neuronales anormales y sincrónicas en una parte del cerebro. Esta etapa puede variar según el tipo de epilepsia y la causa subyacente de la condición. En algunos casos, puede haber una señal de advertencia o aura antes del inicio de una convulsión. Las etapas de los canales iónicos y receptores que intervienen en este proceso son:

Despolarización neuronal, que consiste en la activación de canales asociados a receptores.

Despolarización sostenida: mediada por la estimulación de receptores.

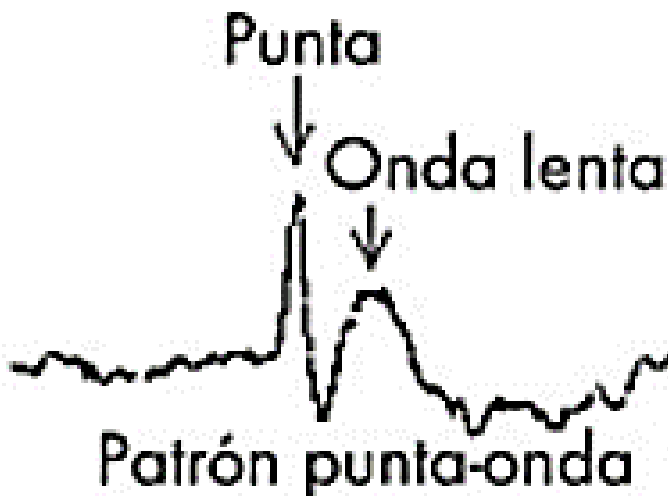
Repolarización se origina por corrientes dependientes de voltaje y la hiperpolarización se produce por la activación de canales receptores, canales asociados y corrientes pendientes de voltaje (García & Simón, 2016).

Sincronización de la descarga: Para que las descargas puedan propagarse es necesario que varias neuronas vecinas descarguen simultáneamente de forma sincronizada (García & Simón, 2016).

Propagación de la descarga: Se debe a que las descargas repetidas aumentan la eficacia de los circuitos excitadores y disminuyen los circuitos inhibidores permitiendo que la descarga escape a los frenos fisiológicos (García & Simón, 2016).

Estas etapas pueden identificarse en un electroencefalograma (EEG), utilizado para el diagnóstico de la enfermedad. En un EEG la despolarización paroxística de las neuronas, se traduce en ondas de alta frecuencia denominadas puntas, mientras que el freno impuesto por las neuronas inhibitoras se observa en ondas de frecuencia lenta. (Véase figura 1).

Figura 1
Ondas mostradas en el electroencefalograma en una crisis epiléptica



Nota. Tomado como referencia de Strogatz, S. H. (2000)

En este gráfico se puede observar que, una vez que se detiene la convulsión, se da paso al estado postictal después de 5 a 30 minutos, se recupera el estado normal de consciencia.

2 MATERIALES Y MÉTODOS

2.1 Modelos para modelar la sincronización.

2.1.1 Generalidades:

Existen modelos que permiten estructurar la sincronización, por lo que, Rivera (2020) afirma que esto depende del contexto y del tipo de sistemas que se estén sincronizando algunos de los modelos son:

Modelo de osciladores acoplados: este modelo describe la sincronización de un conjunto de osciladores que interactúan entre sí mediante acoplamientos. Los osciladores pueden ser mecánicos, eléctricos o biológicos, entre otros. Este modelo se basa en la idea de que los osciladores se influyen mutuamente a través de un acoplamiento, y que pueden sincronizarse en una determinada fase y frecuencia (Rivera, 2020).

Modelo de redes neuronales: este modelo describe la sincronización de una red de neuronas que se comunican entre sí mediante conexiones sinápticas (Rivera, 2020).

Modelo de Kuramoto: este modelo es de tipo matemático es una variante del modelo de osciladores no lineales acoplados que se utiliza para describir la sincronización de modelos complejos con un gran número de osciladores. El modelo se basa en la hipótesis de que los

osciladores interactúan a través de una función de acoplamiento, que depende de la diferencia de fase entre los osciladores, la dinámica de osciladores está dada por ecuaciones diferenciales no lineales que describen la evolución de la fase de cada oscilador en el tiempo (Strogatz, 2000).

Modelo de redes complejas: este modelo se utiliza para describir la sincronización en sistemas de redes complejas, por ejemplo: las redes sociales o las redes de transporte, en este modelo cada nodo de la red representa un elemento del sistema y las conexiones entre los nodos representan las interacciones entre los elementos. Se estudian la sincronización de los nodos de la red. Este modelo se ha aplicado en áreas como la comunicación segura y el procesamiento de señales (Strogatz, 2000).

Modelo de sincronización caótica: este modelo se utiliza para estudiar la sincronización en sistemas caóticos, los mismos que se utilizan como generadores de señales (Strogatz, 2000).

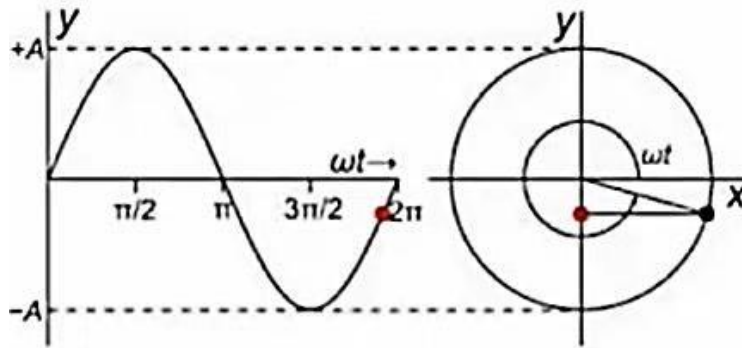
2.1.2 Movimiento Armónico Simple

El Movimiento Armónico Simple (MAS) es un tipo de movimiento oscilatorio que se produce cuando un objeto se mueve de un lado al otro en torno a una posición de equilibrio, siguiendo una trayectoria que es sinusoidal en el tiempo. Strogatz (2020), menciona que en el MAS la aceleración del objeto es proporcional y opuesta a su desplazamiento respecto a la posición de equilibrio. Es decir, cuando la posición del objeto se aleja de la posición de equilibrio, se produce una fuerza restauradora que lo empuja de vuelta hacia ella, este movimiento se caracteriza por tener una frecuencia constante, es decir el objeto oscila a la misma velocidad a lo largo de todo su recorrido. Además, la amplitud del movimiento también es constante, siempre y cuando la fuerza restauradora siga siendo la misma.

El movimiento armónico simple es un modelo matemático que se utiliza para describir fenómenos físicos, como el movimiento de un péndulo simple, de un oscilador eléctrico o el movimiento de una partícula en un resorte; por lo tanto, tomando como referencia este movimiento el gráfico 2 explica los elementos y la deducción de fórmula.

- ❖ La fase del oscilador en un tiempo t es denotado por $\theta(t) \in \mathbb{R}$ y sin interacción la solución de la ecuación diferencial es:

$$\frac{d\theta(t)}{dt} = \omega \therefore \omega \text{ es la frecuencia natural del oscilador.}$$

Figura 2*Movimiento Armónico Simple*

Nota. Tomado como referencia de Strogatz, S. H. (2000)

De la misma manera, la ecuación del movimiento puede ser obtenida mediante la segunda ley de Newton y la Ley de Hooke para la masa de un resorte.

$$F_{net} = m \frac{d^2x}{dt^2} = -kx \quad x(t) = A \sin(\omega t + \theta_0)$$

Siendo A la amplitud y θ_0 la fase inicial.

2.1.3 El modelo de Kuramoto

El fenómeno no fue documentado totalmente hasta 1665. El físico Christiaan Huygens escribió acerca de su experiencia con un par de relojes de péndulos, así lo menciona Rivera (2020). Por otro lado, la sincronización colectiva fue estudiada por primera vez por el matemático Wiener (1958), quien reconoció esta condición en la naturaleza en su obra *Nonlinear Problems In Random Theory*.

Sincronización de componentes de osciladores con comportamiento sinusoidal: este tipo de sincronización hace referencia a la coordinación de las frecuencias y fases de múltiples osciladores para que trabajen juntos de manera coherente y produzcan más señales estables y precisas.

Los osciladores con comportamiento sinusoidal son aquellos que generan señales periódicas que se asemejan a una onda sinusoidal. Estos osciladores pueden encontrarse en una variedad de aplicaciones. Este tipo de sincronización es importante para asegurar que las señales generadas sean coherentes entre sí y no presenten interferencias ni fluctuaciones no deseadas. Para lograr esto, se utilizan diferentes técnicas de sincronización, como la inyección de señal, la oscilación acoplada, la modulación de la fase y la retroalimentación (Rivera, 2020).

Así mismo, el modelo de Kuramoto utiliza la modulación de fase, que consiste en permitir que la frecuencia natural de cada oscilador varíe con el tiempo no solo por la interacción con los demás osciladores, si no también debido a su propia frecuencia natural. Esta modulación de fase

puede tener un efecto significativo en la sincronización de los osciladores en el modelo de Kuramoto, esta modulación puede inducir diferentes tipos de sincronización como la sincronización caótica o estocástica, y puede influir en la estabilidad de los patrones de sincronización. Por lo tanto, se destaca lo importante de este aspecto para la sincronización.

2.1.4 Fórmula y descripción de elementos.

En 1967, Winfree describe la dinámica de N osciladores mediante el modelo que tiene la siguiente forma:

$$\dot{\theta}_i = \omega_i + \left(\sum_{j=1}^N X(\theta_j) \right) Z(\theta_i), \quad i = 1, \dots, N.$$

El primer término ω_i representa la frecuencia natural del oscilador, el segundo término indica la influencia del resto de osciladores j sobre el oscilador i mediante la función $X(\theta_j)$. La magnitud de la función está a la vez controlada por la función de sensibilidad $Z(\theta_i)$.

En 1975, Kuramoto inspirado por Winfree propone el modelo que lleva su nombre en donde se deduce que todos los osciladores actúan con la misma fuerza. El modelo Kuramoto tiene la forma de la expresión (1).

$$\frac{d\theta_i(t)}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1, \dots, N. \quad (1)$$

Donde:

K representa la fuerza de interacción o acoplamiento.

La función seno, denotada en la fórmula (1) por $\sin(\cdot)$, explica el movimiento sinusoidal de las fases obtenidas, a través de la diferencia entre dos fases para representar la influencia de un oscilador sobre otro.

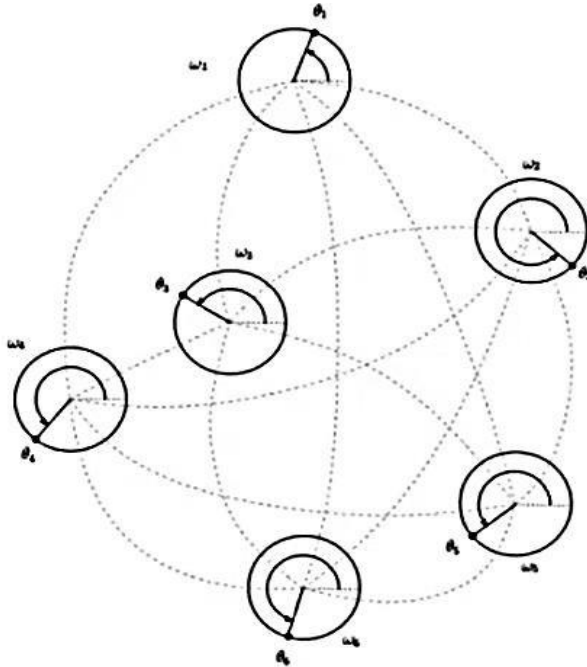
N nos indica los elementos que tiene el sistema, para una solución analítica se puede considerar que la cantidad de elementos tiende al infinito.

La frecuencia natural ω_i proviene de una distribución de frecuencias, donde la distribución de cada oscilador son variables aleatorias idénticamente distribuidas. Por tanto, los osciladores toman su valor de frecuencia natural de manera aleatoria.

En la figura 3 se indica el comportamiento de cada oscilador de un total de 6 osciladores aplicando el modelo de Kuramoto.

Figura 3

Modelo de Kuramoto con 6 osciladores.



Nota. Tomado como referencia de Strogatz, S. H. (2000)

Cada oscilador de la figura 3 tiene su frecuencia natural ω_i a su propio ritmo que le permite moverse por sí mismo. La sincronización ocurre cuando los osciladores recorren con la misma velocidad los círculos, todo depende de la fuerza de la interacción (K), por lo tanto, el parámetro del modelo que indica la interacción es importante.

2.1.5 Medición del parámetro de sincronización.

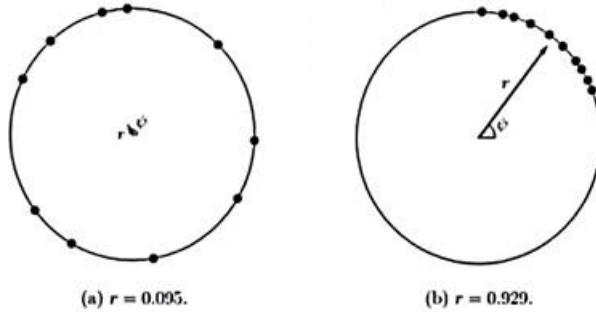
2.1.5.1 Parámetro de orden $r(t)$

Para calcular la sincronización de los osciladores, se toma el promedio de la fase de los osciladores. Como se aprecia en la figura 4, en este caso de dos sistemas de osciladores, si las fases de los osciladores se acercan, el valor del parámetro de orden se acerca a uno (sincronización). Pero si las fases están alejadas, el parámetro de orden se acerca a cero (no hay sincronización, desorden). El parámetro de orden se calcula mediante la expresión (2).

Figura 4

La flecha representa el número complejo $re^{i\psi}$

$$r_N(t)e^{i\varphi_N(t)} = \frac{1}{N} \sum_{j=1}^N e^{i\theta_j(t)} \quad (2)$$



Nota. Tomado como referencia de Strogatz, S. H. (2000)

- ❖ $r_N(t)$ indica el nivel de sincronización con el tiempo t y toma valores entre $[0,1]$
- ❖ $\psi_N(t)$ indica el promedio de la fase con el tiempo t y toma valores entre $[0,2\pi]$

La ecuación de Kuramoto, al compararla con la parte imaginaria del parámetro de orden, puede ser reescrita en términos del parámetro de orden como:

$$\frac{d\theta_i}{dt} = \omega_i - kr \sin(\theta_i)$$

Este modelo explica como los osciladores se acoplan por medio del parámetro de orden, i.e., la fase θ_i es empujada hasta que llegan a la fase promedio que depende de la interacción que hay al inicio es decir los valores de K_0 y r_0 .

$$\frac{d\theta(t)}{dt} = \omega_1 + Kr_N(t) \sin [\psi_N(t) - \theta_i(t)]$$

2.1.5.2 Valores de acoplamiento

Cuando evoluciona $r(t)$, al integrar el modelo numéricamente, usualmente se fija $g(w)$ como una desviación Gaussiana o alguna otra densidad con colas infinitas. Al variar el acoplamiento K las simulaciones muestran que para todo K menor que un cierto umbral K_c , los osciladores actúan como si estuvieran desacoplados: las fases se distribuyen uniformemente alrededor del círculo, a partir de cualquier condición inicial. Entonces $r(t)$, decae a una pequeña fluctuación de tamaño $O(N^{-1/2})$, como se espera para cualquier dispersión aleatoria de N puntos según Strogatz (2000).

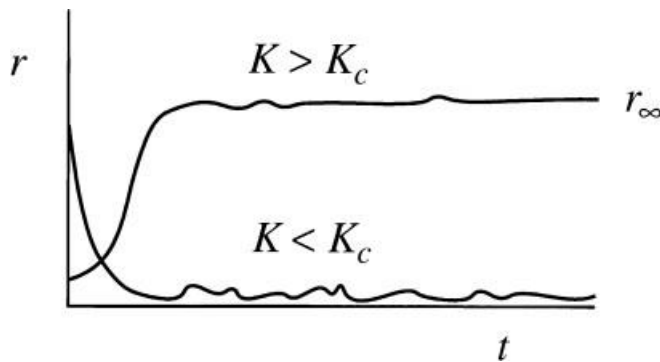
Al sincronizarse los osciladores, podemos considerar que la distribución de las frecuencias presenta una densidad Gaussiana $g(w)$ donde varía el acoplamiento.

Cuando K excede a K_c , $r(t)$ crece exponencialmente, reflejando la centralización de un pequeño grupo de osciladores que están sincronizados entre sí, generando así una oscilación colectiva.

En un determinado tiempo, $r(t)$ se satura en algún nivel $r^\infty < 1$ aunque todavía con fluctuaciones de tipo $O(N^{-1/2})$, como se puede observar en la figura 5.

Figura 5

Ilustración esquemática de la evolución típica de $r(t)$ vista en simulaciones numéricas del modelo de Kuramoto.



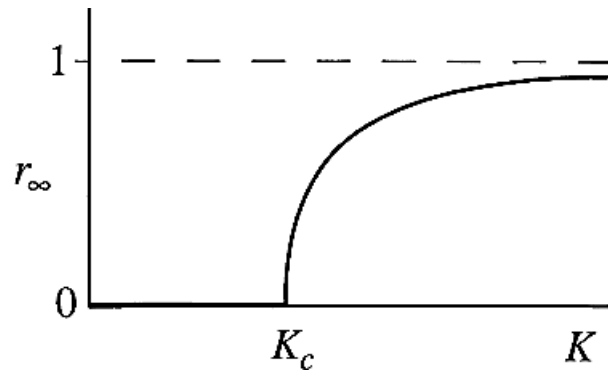
Nota. Tomado como referencia de Strogatz, S. H. (2000)

Al analizar los osciladores de forma individual, se encuentra que la población se divide en dos grupos: los osciladores cerca del centro de la distribución de frecuencias se unen en la frecuencia media y rotan con el promedio de la fase $\psi(t)$, mientras que los de las colas corren cerca de sus frecuencias naturales y se desplazan en relación de un cúmulo sincronizado. Este estado mixto recibe el nombre de parcialmente sincronizado.

Con más aumentos en K , más osciladores en el grupo sincronizados y r^∞ crece como muestra la figura 6.

Figura 6

Dependencia de la coherencia en estado estacionario de r^∞ de la fuerza de acoplamiento k .



Nota. Tomado como referencia de Strogatz, S. H. (2000)

Los números indican que r^∞ depende solo de K y no de la condición inicial, es decir se entiende que hay un estado de atracción global para cada valor de K .

2.1.5.3 Generalización del Modelo de Kuramoto

Martínez (2019) menciona que la generalización del modelo de Kuramoto se refiere a la extensión o modificación del modelo original propuesto por Yoshiki Kuramoto en la década de 1970. El modelo de Kuramoto original se basa en un conjunto de osciladores con acoplamientos débiles que interactúan entre sí. Cada oscilador tiene una frecuencia intrínseca y se acopla con los otros osciladores mediante un término de acoplamiento, para así poder describir como los osciladores se ajustan en su fase dependiendo del nivel de acoplamiento (parcial o total) entre ellos.

A lo largo de los años el modelo de Kuramoto puede referirse a diferentes formas que se han acoplado dependiendo de las necesidades según las características y efectos del sistema, siendo algunas de estas generalizaciones: Acoplamientos no lineales, acoplamientos de retraso, ruido, perturbaciones y entre otras.

Por lo tanto, la idea es adaptar el modelo a las características específicas del sistema en estudio para obtener una descripción lo más precisa de la sincronización de osciladores, para nuestro caso la referencia es la sincronización de los potenciales de acción de las neuronas en un ataque epiléptico.

2.1.6 Utilidad de la transformada de Fourier

La transformada de Fourier es una herramienta matemática que se utiliza para analizar señales en el dominio de la frecuencia. Permite descomponer una señal en una serie de ondas sinusoidales, cada una de las cuales tiene una frecuencia y amplitud específicas.

La transformada de Fourier toma una función en el dominio del tiempo, y la convierte en una función en el dominio de la frecuencia, donde se representa en términos de amplitud y fase para cada frecuencia presente.

La fórmula matemática de la transformada de Fourier se expresa de la siguiente forma:

$$F(\omega) = \int [f(t)e^{-i\omega t}] dt$$

Donde $F(\omega)$ representa la función transformada en el dominio de la frecuencia, $f(t)$ es la función original en el dominio del tiempo, donde ω es la frecuencia angular.

Computacionalmente, se puede hallar la transformada de Fourier de una señal mediante la FFT (Transformada de Fourier rápida). FFT es un algoritmo basado en la propiedad de simetría y recursión de una matriz de transformación permitiendo leer los datos en función de la frecuencia en lugar del dominio del tiempo. Es decir, se interpreta la señal, con un vector de número imaginarios, donde el módulo de cada componente representa la amplitud de la frecuencia en la señal original y el ángulo o argumento es la fase de cada frecuencia discreta.

Para utilizar la transformada de Fourier en el procesamiento de la información de un Electroencefalograma (EEG), se sigue generalmente el siguiente procedimiento:

1. Preprocesamiento del EEG: antes de aplicar la transformada de Fourier, es común realizar algunas etapas de preprocesamiento en los datos del EEG para mejorar la calidad de la señal. Esto puede incluir: la eliminación de ruido, la normalización de la amplitud o la segmentación en intervalos de interés.
2. Selección de una ventana de tiempo: se selecciona una ventana de tiempo del EEG que se desea analizar. Esto es escoger un segmento específico de la señal, como una época de actividad cerebral particular o una ventana más amplia que abarque toda la duración del registro del EEG.
3. Aplicación de la Transformada de Fourier: se aplica la Transformada de Fourier a la ventana de tiempo seleccionada utilizando algoritmos o funciones específicas disponibles en herramientas o librerías de procesamiento de señales, esto obtiene la representación en el dominio de la frecuencia.
4. Análisis de los resultados: se examina los resultados de la Transformada de Fourier, que generalmente se representan en un espectro de frecuencia. Se debe observar las amplitudes y fases de las diferentes componentes de frecuencia presentes en el EEG, para así identificar picos o bandas de frecuencias específicas que presenten importancia clínica o investigativa.

Simulación numérica Python. En un script de Python se realizó los siguientes pasos:

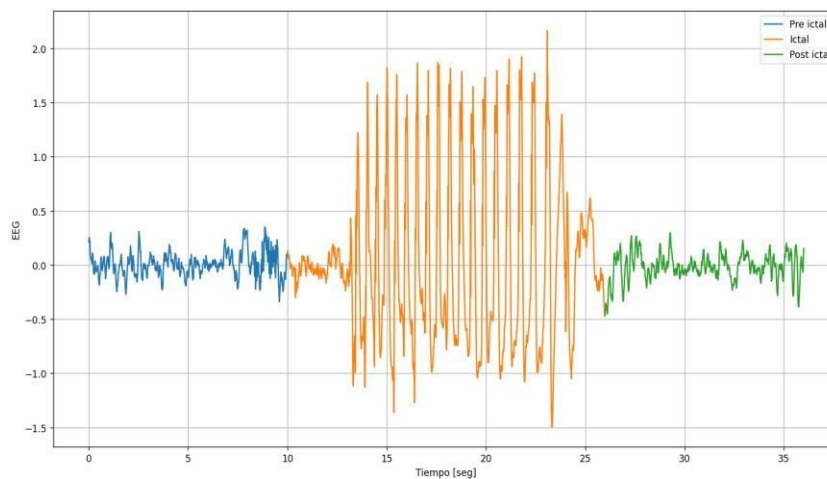
1. Lectura de los datos en Excel que contiene los archivos correspondientes a los pacientes y la información respectiva sobre el ataque.
2. En lo posterior se lee la cantidad de canales y se extrae una porción de registro la misma que es analizada.
3. Se promedia la señal de todos los canales con una referencia de diez segundos antes y diez segundos después del ataque.
4. Se realiza el filtrado de la señal después de aplicar la transformada de Fourier, para obtener las frecuencias, amplitudes y fases de las componentes que tienen mayor efecto en cada etapa.
5. A partir de los datos del paso 4, se reconstruye la señal sin el método de Kuramoto para comparar la gráfica de la señal con la reconstrucción de la señal con el método de Kuramoto.
6. Finalmente, gracias a la comparación de la sincronía de la señal con su versión reconstruida usando Kuramoto, se obtiene la constante de acoplamiento en función del valor del parámetro de orden.

3 RESULTADOS

Tras el análisis general de los datos de varios pacientes, se analiza de forma detenida los resultados del paciente chb24_13. El registro de este paciente indica que el ataque epiléptico tuvo una duración de 16 segundos. En la figura 7 se puede observar al medio de la gráfica representado el periodo ictal con una señal muy ejemplar de lo que define un ataque epiléptico.

Figura 7

Electroencefalograma del ataque de epilepsia.

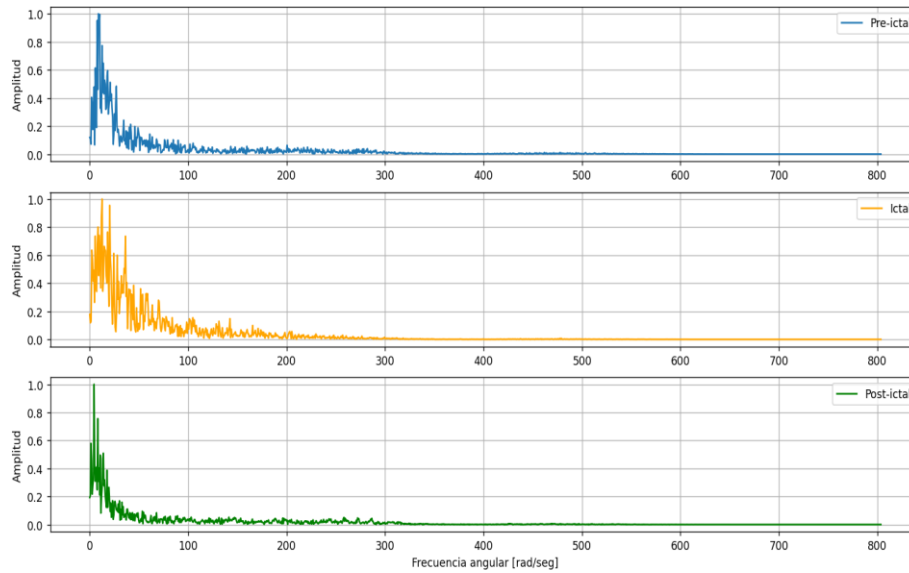


Fuente. Elaboración propia al correr el programa en Python

La figura 8 hace referencia a la aplicación de la transformada de Fourier y la descomposición de la señal en un espectro que muestra la frecuencia y la amplitud en las tres etapas del ataque epiléptico.

Figura 8

Medición de la Transformada de Fourier.

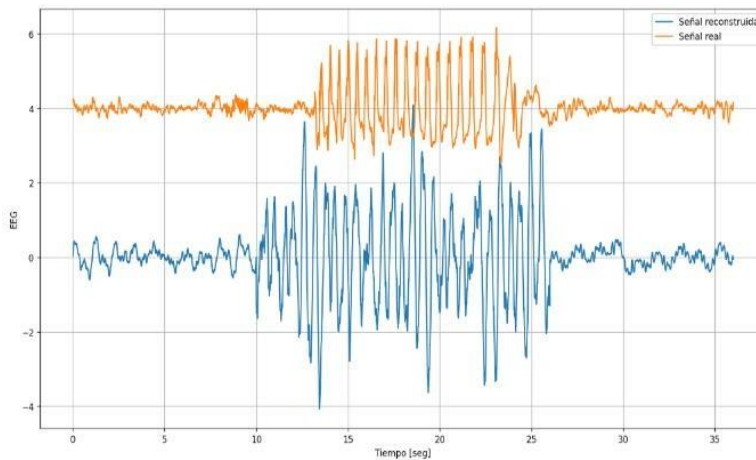


Nota. Elaboración propia al correr el programa en Python

A continuación, la figura 9 permite observar una imagen del filtrado para la reconstrucción de la señal obteniendo frecuencias, amplitudes y fases que tienen mayor aporte a la señal en cada etapa. La figura 10 muestra la aplicación del modelo de Kuramoto para analizar la sincronización durante la fase ictal del ataque epiléptico.

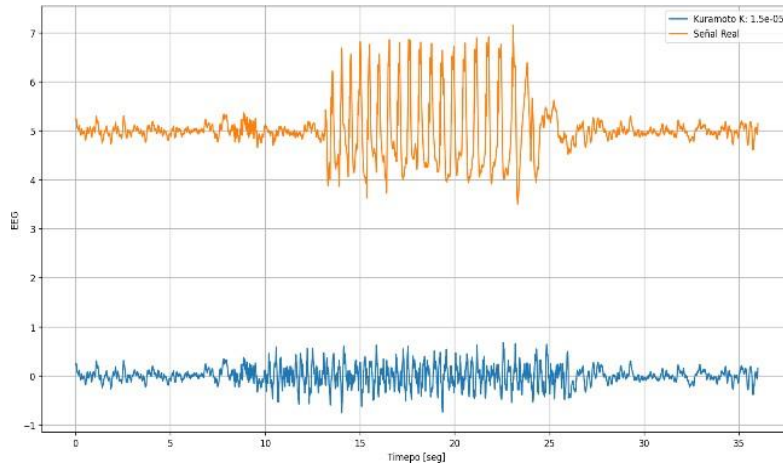
Figura 9

Señal reconstruida sin modelo de Kuramoto



Fuente. Elaboración propia al correr el programa en Python

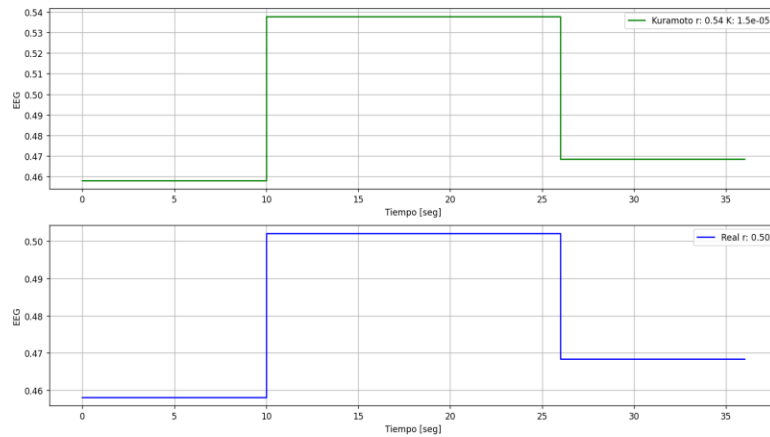
Figura 10
Señal reconstruida con modelo de Kuramoto



Fuente. Elaboración propia al correr el programa en Python.

En la figura 11, se compara el valor del parámetro de orden, r , obtenido de la señal de EEG aplicando el modelo de Kuramoto para un solo valor de acoplamiento (K), que se ajusta con el parámetro de la señal del EEG original. La figura indica el comportamiento del valor de r en las tres etapas de un ataque epiléptico.

Figura 11
Obtención y comparación del parámetro de orden para el modelo original y con Kuramoto



Fuente. Elaboración propia al correr el programa en Python.

Este proceso se repite 20 veces con el objetivo de asignar diferentes valores al parámetro de acoplamiento (K) y obtener diferentes valores del parámetro de orden. Con este procedimiento se simula el nivel de sincronización de las neuronas. La tabla 1, muestra el resumen de los resultados obtenidos.

Tabla1

Comparación entre el parámetro de r y los valores de acoplamiento

Prueba 1	K = 0.00002
Parámetro de orden en el ataque modelo Kuramoto	0.5377691833074207
Parámetro de orden en el ataque modelo real	0.5021239617324154
Prueba 2	K = 0.00006
Parámetro de orden en el ataque modelo Kuramoto	0.5380240588561758
Prueba 3	K = 0.00011
Parámetro de orden en el ataque modelo Kuramoto	0.5382196081544172
Prueba 4	K = 0.0016
Parámetro de orden en el ataque modelo Kuramoto	0.5384461316687476
Prueba 5	K = 0.00021
Parámetro de orden en el ataque modelo Kuramoto	0.5385516825791163
Prueba 6	K = 0.00026
Parámetro de orden en el ataque modelo Kuramoto	0.5387583089578363
Prueba 7	K = 0.00031
Parámetro de orden en el ataque modelo Kuramoto	0.5389505807844244
Prueba 8	K = 0.00036
Parámetro de orden en el ataque modelo Kuramoto	0.5391145216006237
Prueba 9	K = 0.00041
Parámetro de orden en el ataque modelo Kuramoto	0.539163001297698
Prueba 10	K = 0.00046
Parámetro de orden en el ataque modelo Kuramoto	0.5394079752268771
Prueba 11	K = 0.00051
Parámetro de orden en el ataque modelo Kuramoto	0.539632079619884
Prueba 12	K = 0.00056
Parámetro de orden en el ataque modelo Kuramoto	0.5398505339863152
Prueba 13	K = 0.00061
Parámetro de orden en el ataque modelo Kuramoto	0.5400512686096806
Prueba 14	K = 0.00066
Parámetro de orden en el ataque modelo Kuramoto	0.5402143227713192
Prueba 15	K = 0.00070
Parámetro de orden en el ataque modelo Kuramoto	0.5403184842412957
Prueba 16	K = 0.00075
Parámetro de orden en el ataque modelo Kuramoto	0.5402353798639835
Prueba 17	K = 0.00080
Parámetro de orden en el ataque modelo Kuramoto	0.540180680155208
Prueba 18	K = 0.00085
Parámetro de orden en el ataque modelo Kuramoto	0.5401536184216167
Prueba 19	K = 0.00090
Parámetro de orden en el ataque modelo Kuramoto	0.5400650423328667
Prueba 20	K = 0.00095
Parámetro de orden en el ataque modelo Kuramoto	0.5399598256594407

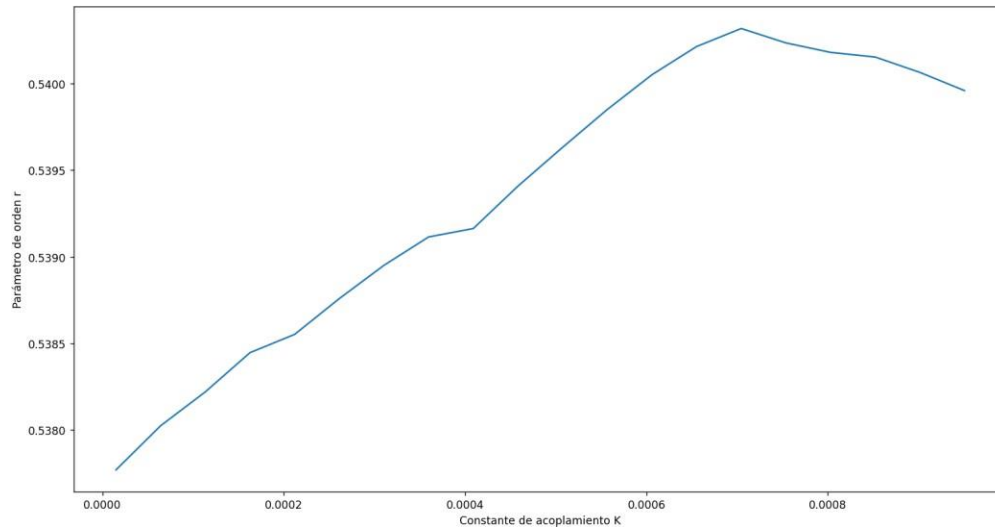
Fuente. Elaboración propia al correr el programa en Python.

Se puede observar en la tabla de la parte superior que los resultados comprueban que a medida que aumenta la medida de K mayor será el valor de r , entonces el acoplamiento se vuelve más fuerte, lo que indica que existe una mayor sincronización entre las neuronas, es decir el ataque epiléptico.

A continuación, la figura 12 permite comprender como el parámetro de orden indica el grado de sincronización y como cambia en función del acoplamiento y las frecuencias. Llegando a un punto máximo para luego descender, el comportamiento se justifica debido a que el parámetro de orden como valor máximo puede tomar el valor de uno. Pero, obtener y mantener el valor de uno indicaría que el paciente mantendría el ataque epiléptico, lo cual no ocurre en la vida real.

Figura 12

Parámetro de orden en función de la constante de acoplamiento.



El parámetro de orden y la constante de acoplamiento, poseen un coeficiente de correlación lineal $\rho = 0.51502824$

Fuente. Elaboración propia al correr el programa en Python.

4 DISCUSIÓN

Este trabajo se enfoca en la sincronización de la actividad neuronal durante un ataque epiléptico. Esta actividad sincrónica se evidencia al promediar la señal de todos los canales del registro de EEG y obtener mayor amplitud durante la duración del ataque. La figura 7 presenta un ejemplo de un ataque epiléptico, mostrando la existencia de tres etapas: pre-ictal, ictal y post-ictal (Izquierdo, 2018).

Además, en el dominio de la frecuencia también existen cambios entre las tres etapas. Estos cambios se presentan en la figura 8. En este trabajo se reconstruye la señal durante la etapa ictal de tal forma de tener un espectro en frecuencia similar. Esto se realizó mediante la transformada inversa de Fourier y mediante una versión modificada del modelo de Kuramoto. En el segundo caso, al considerar los valores de K y r , se obtienen los patrones característicos

asociados a la actividad epileptiforme, evidenciando la relación entre la sincronía y los ataques epilépticos.

Usando el modelo de Kuramoto, cuando el valor de K aumenta el valor de r también aumenta. Esto se relaciona con una actividad cerebral anormal sincronizada, un grupo de neuronas en múltiples regiones del cerebro se activan de manera rítmica y excesiva al mismo tiempo. Este comportamiento no sucede en condiciones normales del cerebro. En condiciones no patológicas las neuronas se comunican y activan con baja sincronía, y no totalmente sincronizadas. Entonces, un ataque epiléptico podría explicarse por el aumento del acoplamiento entre neuronas.

Como parte de la metodología utilizada en la obtención de los resultados, fue importante el uso de una técnica de ventanas temporales. Esta técnica mejoró la precisión en el análisis del dominio de la frecuencia. Además, se logró reducir la interferencia en el espectro y mejorar la visualización de los componentes de frecuencias presentes en la señal frente al ruido de fondo. Esta técnica conocida como periodograma, tiene la ventaja de reducir el ruido en el espectro. Esto permite identificar los picos de frecuencia, sin confundir la señal con el ruido.

En el trabajo de (Zavaleta-Viveros, 2023) utilizan un método parecido, seleccionando 14 frecuencias, con sus amplitudes y fases, desde la señal de EEG para reconstruir la señal durante la etapa ictal. En este trabajo se seleccionaron 15 frecuencias principales de la señal con el criterio de que sean menores a 80 Hz. Por tanto, se tienen 15 osciladores, cuya frecuencia natural corresponde con las frecuencias encontradas, y de cuya interacción mediante el modelo Kuramoto se obtiene una reconstrucción de la señal, tal como en la figura 10.

El resultado principal de este trabajo se presenta en la Tabla 1 y la figura 13, en donde se evidencia que el incremento K incrementa la sincronía, medida por el parámetro de orden de Kuramoto. Sin embargo, esto ocurre hasta cierto valor de K , donde el valor de r ocupa un máximo, y luego de este valor de K , la sincronía ya no aumenta, sino que decrece.

Resultados similares se han encontrado en el trabajo de (Schmidt, 2014). Los autores de este trabajo indican que, si la diferencia de fase de algunos nodos tiene un retraso de fase de π , entonces la sumatoria en el modelo Kuramoto tiene signo: $\text{Sin}(\theta_i - \theta_j - \pi) = -\text{Sin}(\theta_i - \theta_j)$. Además, si esto sucede para algunos nodos que controlan la dinámica de todo el sistema, se puede tener el caso de que al incrementar K se incrementa r hasta llegar a un punto máximo, luego de este punto, al aumentar aún más el valor de K , se reduce r . Dado que producen un efecto en sentido contrario a la sincronía global esperada con el aumento de K .

Es importante señalar algunas limitaciones de este estudio. Primero la información sobre la actividad mental fue tomada de una muestra de pacientes que padecen la enfermedad (Goldberg, 2020). Por lo cual la información puede ser insuficiente para analizar todos los casos que se presenten en una población que está en etapas tempranas de epilepsia. En segundo lugar, los resultados también apuntan a la importancia de considerar el tiempo que dura el ataque epiléptico para analizar los resultados de parámetro r que a su vez depende del valor del parámetro de acoplamiento K . Por lo cual en algunos análisis de los pacientes con ataques epilépticos se hace la excepción de considerar a mayor K mayor valor de r , esto se debe a las diferencias entre individuos. Por último, se considera que al analizar el comportamiento del cerebro existen muchos más factores a ser analizados que su espectro en el dominio de la frecuencia y la sincronía.

5 CONCLUSIONES

El objetivo en este trabajo fue simular un ataque epiléptico con el modelo de Kuramoto mediante el análisis de un EEG, los resultados sugieren que al aplicar el modelo de Kuramoto modificado ofrece una buena aproximación para replicar un ataque epiléptico a través del cumplimiento de tres criterios:

1. Verificar que tan parecida en frecuencia y sincronía son las señales: original de un EEG con la señal reconstruida aplicando el modelo de Kuramoto, asignando los valores correctos al parámetro de acoplamiento K .
2. El valor del parámetro de orden r debe ser mayor al valor de parámetro de orden del periodo pre ictal y menor que el valor del parámetro del periodo post ictal.
3. El valor del parámetro de orden de la señal original debe ser lo más cercano posible a la señal obtenida aplicando el modelo de Kuramoto.

En definitiva, la señal reconstruida con Kuramoto debe tener la frecuencia y la sincronía de la señal original y para ser un ataque debe implementar el parámetro r en las dos etapas.

6 BIBLIOGRAFÍA

- García, P., & Simón, M. (2016). Neurotransmisores implicados en la epilepsia y su tratamiento. *Universidad Complutense de Madrid*. Obtenido de <https://hdl.handle.net/20.500.14352/66440>
- Gerster, M., Berner, R., Sawicki, J., Zakharova, A., Škoch, A., Hlinka, J., . . . Schöll, E. (2020). FitzHugh–Nagumo oscillators on complex networks mimic epileptic-seizure-related synchronization phenomena. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12). Obtenido de <https://doi.org/10.1063/5.0021420>
- Goldberger, Amaral, Glass, Hausford, Ivanov, Mark, & Atanley. (2020). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.
- Izquierdo, A. Y. (2018). Epilepsia y crisis epilépticas: causas, diferencias y tratamiento. TopDoctors. Recuperado el 1 de junio de 2023, de <https://www.topdoctors.es/articulos-medicos/epilepsia-y-crisis-epilepticas-causas-diferencias-y-tratamiento>
- López, M., Rocha, L., Miquel, M., Hernández, M., Toledo, R., Coria, G., . . . Manzo, J. (2009). Conceptos básicos de la epilepsia. *Revista Médica de La Universidad Veracruzana*, 9(2), 31-37. Obtenido de <https://www.medigraphic.com/pdfs/veracruzana/muv-2009/muv092e.pdf>
- Martínez, D. (2019). *Dinámica de dos agentes en el modelo de kuramoto generalizado*. Obtenido de <http://hdl.handle.net/10902/16911>
- Rivera, A. (2020). *La sincronización espontánea de los péndulos*. Consultado. Obtenido de <https://ruba.com.mx/la-sincronizacion-espontanea-de-los-pendulos-los-metronomos-y-los-aplausos/>
- Schmidt, H., Petkov, G., Richardson, M., & Terry, J. (2014). Dynamics on networks: the role of local dynamics and global networks on the emergence of hypersynchronous neural activity. *PLoS Comput Biol*, 10(11), 1-16. Obtenido de <https://doi.org/10.1371/journal.pcbi.1003947>
- Strogatz, S. (2000). From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *ELSEVIER*, 143. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0167278900000944>
- Strogatz, S. (2000). From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physical D*, 1-20.
- Wiener, N. (1958). *Nonlinear Problems In Random Theory*. Atlanta.
- Winfree, A. (1967). Biological rhythms and the behavior of populations of coupled oscillators. *J Theor Biol*, 16(1), 15-42. Obtenido de [https://doi.org/10.1016/0022-5193\(67\)90051-3](https://doi.org/10.1016/0022-5193(67)90051-3)

Zabaleta, J., Toledo, P., Avendaño, M., & Escalante, J. (2020). A proposal of an empirical methodology to approximate an electroencephalographic signal with appropriate representatives of the Fourier transformation. *Lecture notes in computer science*, 12469, 309-324. Obtenido de https://doi.org/10.1007/978-3-030-60887-3_28

7 ANEXOS

Anexo1

Código de programa

Código del programa en Python para la implementación del modelo de Kuramoto.

```

from edfreader import EDFreader
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from scipy.fft import fft, ifft
from scipy.fft import fftfreq, rfftfreq
from scipy.signal import butter, filtfilt
import xlrd
"""*****FUNCIONES A UTILIZAR *****"""
def zScore(X,mu=None, sigma=None):
    #Obtener el z-score canal por canal.
    if mu is None and sigma is None:
        y=X-X.mean(axis=0,keepdims=True)
        y=y/X.std(axis=0,keepdims=True)
    elif sigma is None:8
    #Obetener el z-score con una desviación estándar común (Se suele usar la de todos los
registros y todos los canales)
        y=X-X.mean(axis=0,keepdims=True)
        y=y/sigma
    else:
    #Obtener el z-score con una media y desviación estándar común
        y=(X-mu)/sigma
    return y

def para_orden(angulo):
    vec=0;
    for i in range(np.size(angulo)):
        vec=vec + np.cos(angulo[i])+1.j*np.sin(angulo[i])
    alpha=np.angle(vec)

r=abs(vec)/np.size(angulo)
    return [alpha,r]

def ventanado(data): #Enventanado / Windows Slice
    k=0
    r1=np.zeros(np.shape(data))
    alpha1=np.zeros(np.shape(data))
    barrido=64 #t=64/256
    tamaño_muestra = sampling_rate

    furier = fft(data[k:k+tamaño_muestra]);
    #amp=abs(furier)

angulo=np.angle(furier)
    parametro= para_orden(angulo)
    r1[k:k+tamaño_muestra]=parametro[1]*np.ones(tamaño_muestra)
    alpha1[k:k+tamaño_muestra] = parametro[0]*np.ones(tamaño_muestra)
    k=k+barrido

```



```

while(k+tamano_muestra<=np.size(data)):
    furier = fft(data[k:k+tamano_muestra]);
    #amp=abs(furier)
    angulo=np.angle(furier)
    parametro= para_orden(angulo)
    r1[k:k+tamano_muestra]= (r1[k:k+tamano_muestra] +
parametro[1]*np.ones(tamano_muestra))/2
    alpha1[k:k+tamano_muestra] = (alpha1[k:k+tamano_muestra] +
parametro[0]*np.ones(tamano_muestra))/2
    k=k+barrido

    furier = fft(data[k:k+tamano_muestra]);
    #amp=abs(furier)
    angulo=np.angle(furier)
    parametro= para_orden(angulo)
    r1[k:]= (r1[k:] + parametro[1]*np.ones(np.size(data)-k))/2
    alpha1[k:] = (alpha1[k:] + parametro[0]*np.ones(np.size(data)-k))/2
    return r1

def butter_lowpass_filter(data, cutoff, fs, order,type_filt):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    # Get the filter coefficients
    b, a = butter(order, normal_cutoff, btype=type_filt, analog=False)
    y = filtfilt(b, a, data)

    return y

def det_freqs(data,frecuencias):
    #busco las frecuencias antes de los 80 Hz
    index_80hz=np.where(frecuencias==2*np.pi*80)
    tamano=index_80hz[0][0]
    data2=data[0:np.size(data)//2] #amplitudes de las frecuencias positivas
    freq2=frecuencias[0:np.size(frecuencias)//2] #frecuencias positivas
    freqini=int(tamano//1.3)
    amp=np.array(max(abs(data2[freqini:tamano])))#tomo el primer dato de la mitad de frecuencia
    fase=np.array(np.angle(data2[freqini+np.where(abs(data2[freqini:tamano])==amp)[0][0]]))
    freq_out=np.array(freq2[freqini+np.where(abs(data2[freqini:tamano])==amp)[0][0]])
    tamano=freqini
    freqini=int(freqini//1.3)
    k=1
    while(freqini!=0):
        amp=np.append(amp,max(abs(data2[freqini:tamano])))

    fase=np.append(fase,np.angle(data2[freqini+np.where(abs(data2[freqini:tamano])==amp[k])[0][0]]))
    freq_out=np.append(freq_out,freq2[freqini+np.where(abs(data2[freqini:tamano])==amp[k])[0][0]])
    k+=1
    tamano=freqini
    freqini=int(freqini//1.3)
    amp=np.append(amp,abs(data2[0]))
    fase=np.append(fase,np.angle(data2[0]))
    freq_out=np.append(freq_out,freq2[0])

```

```

return [amp,freq_out,fase]

def serie_furier(amp,fases,frecuencias,tiempo):
    y=np.zeros(np.size(tiempo))
    for i in range(np.size(amp)):
        y=y+amp[i]*np.sin(frecuencias[i]*tiempo+fases[i])
    for i in range(np.size(amp)):
        y=y+amp[i]*np.sin(frecuencias[i]*tiempo-fases[i])
    return y

def modelo_kuramoto(amp_ini,fase_ini,frecuencia_ini,tiempo,K):
    y=np.zeros(np.size(tiempo))
    theta=np.zeros([np.size(fase_ini),np.size(tiempo)])
    w1=np.zeros([np.size(fase_ini),np.size(tiempo)])
    amp1=np.zeros([np.size(fase_ini),np.size(tiempo)])
    # Se colocan los primero valores de la primer iteración
    w1[:,0]=frecuencia_ini
    ## theta[:,0]= fase_ini + w1[:,0]*tiempo[0]
    ## amp1[:,0]=amp_ini
    ## for j in range(np.size(amp1[:,0])):
    ##     y[0]=y[0]+amp1[j,0]*np.sin(theta[j,0])
    ##
    ## for k in range(np.size(w1[:,0])):
    ##     w1[k,1]=w1[k,0]
    ##     for l in range(np.size(w1[:,0])):
    ##w1[k,1]=w1[k,1]+K*(np.sin(theta[k,0]-theta[l,0]))/np.size(amp_ini)
    #resto de iteraciones
    for i in range(np.size(tiempo)):
        theta[:,i]= fase_ini + w1[:,i]*tiempo[i]
        amp1[:,i]=amp_ini
        for j in range(np.size(amp1[:,i])):
            y[i]=y[i]+amp1[j,i]*np.sin(theta[j,i])
        if i < np.size(tiempo)-1:
            for k in range(np.size(w1[:,i])):
                w1[k,i+1]=w1[k,i]
                for l in range(np.size(w1[:,i])):
                    w1[k,i+1]=w1[k,i+1]+K*(np.sin(theta[k,i]-theta[l,i]))/np.size(amp_ini)
    return y

*****LECTURA DE HOJA DE EXCEL*****

directorio = r"C:\Users\lenovo\Downloads\Python\Trabajo_epilepsia\Registros\\"
#Lectura de datos en excel
excel = xlrd.open_workbook("resumen_archivos.xls")
hoja=excel.sheet_by_index(0)
"""
De la variable hoja:
registro_n es la columna del excel
hoja[registro_n][0] -> nombre del archivo o paciente
hoja[registro_n][1] -> tiempo que inicia el ataque epiléptico en seg
hoja[registro_n][2] -> Tiempo en el que finaliza el ataque en seg
hoja[registro_n][3] -> tiempo total de la muestra del eeg
"""

*****lectura de los archivos*****
registro_n=8

```

```

filename=directorio+hoja[registro_n][0].value+'.edf'
hdl = EDFreader(filename)

ncañales=hdl.getNumSignals() #Cantidad de canales

sampling_rate=int(hdl.getSampleFrequency(s=0)) #Muestras por segundo
#Buffer para extraer una porción del registro
buffer=np.zeros((sampling_rate*int(hoja[registro_n][3].value),ncañales))
##Llenar el buffer con cada canal del registro desde la posición deseada
for n_cañal in range(ncañales):
    #Para cada archivo se se toma 10seg antes del ataque
    hdl.fseek(s=n_cañal,offset=(int(hoja[registro_n][1].value)-
10)*sampling_rate,whence=hdl.EDFSEEK_SET)
    hdl.readSamples(n_cañal, buffer[:,n_cañal], sampling_rate*int(hoja[registro_n][3].value))

"""*****ETAPA DE MANEJO DE DATOS*****"""
#obtención de datos
z_signals=zScore(buffer)
# se promedia la señal de todos los canales
z_signal_mean = sum(np.transpose(z_signals))/ncañales
#se toma las señales 10 seg antes del ataque, en el ataque y 10 segundos luego del ataque
etapa_1 = z_signal_mean[0:10*sampling_rate] #10 segundos antes del ataque
etapa_2 = z_signal_mean[10*sampling_rate:sampling_rate*(int(hoja[registro_n][3].value)-10)]
#ataque
etapa_3 = z_signal_mean[sampling_rate*(int(hoja[registro_n][3].value)-
10):sampling_rate*int(hoja[registro_n][3].value)] #10 segundo luego del ataque
## filtrado de las señales
etapa_1_filt=butter_lowpass_filter(etapa_1,70,sampling_rate,2,'low')
etapa_2_filt=butter_lowpass_filter(etapa_2,70,sampling_rate,2,'low')
etapa_3_filt=butter_lowpass_filter(etapa_3,70,sampling_rate,2,'low')
#Gráfico de las señales originales
tiempo =
np.linspace(1/sampling_rate,int(hoja[registro_n][3].value),sampling_rate*int(hoja[registro_n][3].v
alue))
plt.figure(figsize=(20,10))
plt.plot(tiempo[0:10*sampling_rate],etapa_1_filt,label='Pre ictal')
plt.plot(tiempo[10*sampling_rate:sampling_rate*(int(hoja[registro_n][3].value)-
10)],etapa_2_filt,label='Ictal')
plt.plot(tiempo[sampling_rate*(int(hoja[registro_n][3].value)-
10):sampling_rate*int(hoja[registro_n][3].value)],etapa_3_filt,label='Post ictal')
plt.legend()
plt.grid(visible=1)
plt.title("Electroencefalograma del ataque de epilepsia")
plt.ylabel("EEG")
plt.xlabel("Tiempo [seg]")
plt.show()
#plt.savefig("Electroencefalograma del ataque de epilepsia.png")
plt.close()
"""*****ETAPA DE FOURIER*****"""

## medición de la transformada de Fourier
plt.figure(figsize=(20,10))
plt.suptitle("Transformada de Fourier por etapas")
etapa_1_fft = fft(etapa_1_filt)
freq_estapa_1=2*np.pi*fftfreq(np.size(etapa_1),1/sampling_rate)
plt.subplot(311)

```

```
plt.plot(freq_estapa_1[0:np.size(freq_estapa_1)//2],abs(etapa_1_fft[0:np.size(etapa_1_fft)//2])/m
ax(abs(etapa_1_fft),label="Pre-ictal")
plt.ylabel("Amplitud")
plt.legend()
plt.grid(visible=1)
```

```
etapa_2_fft = fft(etapa_2_filt)
freq_estapa_2=2*np.pi*fftfreq(np.size(etapa_2),1/sampling_rate)
plt.subplot(312)
plt.plot(freq_estapa_2[0:np.size(freq_estapa_2)//2],abs(etapa_2_fft[0:np.size(etapa_2_fft)//2])/m
ax(abs(etapa_2_fft),color="orange",label="Ictal")
plt.ylabel("Amplitud")
plt.legend()
plt.grid(visible=1)
```

```
etapa_3_fft = fft(etapa_3_filt)
freq_estapa_3=2*np.pi*fftfreq(np.size(etapa_3),1/sampling_rate)
plt.subplot(313)
plt.plot(freq_estapa_3[0:np.size(freq_estapa_3)//2],abs(etapa_3_fft[0:np.size(etapa_3_fft)//2])/m
ax(abs(etapa_3_fft),color="green",label="Post-ictal")
plt.ylabel("Amplitud")
plt.xlabel("Frecuencia angular [rad/seg]")
plt.grid(visible=1)
plt.legend()
plt.show()
#plt.savefig("Tranformadas de Fourier de las etapas.png")
plt.close()
```

```
""""*****ETAPA DE FILTRADO PARA RECONSTRUCCION DE LA SEÑAL *****""""
#Se obtiene las frecuencias, amplitudes y fases de las las componentes que tienen mayor efecto
en cada etapa
param_k_etapa1= det_freqs(etapa_1_fft,freq_estapa_1)
param_k_etapa2= det_freqs(etapa_2_fft,freq_estapa_2)
param_k_etapa3= det_freqs(etapa_3_fft,freq_estapa_3)
```

```
""""*****ETAPA DE RECONSTRUCCIÓN DE LA SEÑAL SIN KURAMOTO *****""""
```

```
# primero se grafica el modelo de la señal reconstruida sin aplicar el modelo de Kuramoto
```

```
tiempo_1=tiempo[0:10*sampling_rate]
tiempo_2=tiempo[10*sampling_rate:sampling_rate*(int(hoja[registro_n][3].value)-10)]
tiempo_3=tiempo[sampling_rate*(int(hoja[registro_n][3].value)-
10):sampling_rate*int(hoja[registro_n][3].value)]
```

```
y1=serie_furier(param_k_etapa1[0],param_k_etapa1[2],param_k_etapa1[1],tiempo_1)
y2=serie_furier(param_k_etapa2[0],param_k_etapa2[2],param_k_etapa2[1],tiempo_2)
y3=serie_furier(param_k_etapa3[0],param_k_etapa3[2],param_k_etapa3[1],tiempo_3)
y=zScore(np.concatenate((y1,y2,y3),axis=None))
plt.figure(figsize=(20,10))
plt.title("Reconstrucción de la señal sin modelo de Kuramoto")
plt.plot(tiempo,y,label="Señal reconstruida")
plt.plot(tiempo,z_signal_mean+4,label="Señal real")
plt.ylabel("EEG")
plt.xlabel("Tiempo [seg]")
plt.legend()
plt.grid(visible=1)
```

```

plt.show()
#plt.savefig("Reconstrucción de la señal sin Kuramoto.png")
plt.close()

"""*****ETAPA DE RECONSTRUCCIÓN DE LA SEÑAL CON KURAMOTO*****"""
K_kuramoto=0.00025
y2_k=zScore(modelo_kuramoto(param_k_etapa2[0],param_k_etapa2[2],param_k_etapa2[1],tiempo_2,K_kuramoto))
y2_kfilt= butter_lowpass_filter(y2_k,np.array([3,70.0]),sampling_rate,2,'bandpass')
y_graph=np.concatenate((etapa_1,y2_kfilt,etapa_3),axis=0)
plt.figure(figsize=(20,10))
plt.title("Reconstrucción de la señal con modelo de Kuramoto")
plt.plot(tiempo,y_graph,label="Kuramoto K: "+str(K_kuramoto))
plt.plot(tiempo,z_signal_mean+5,label="Señal Real")
plt.ylabel("EEG")
plt.xlabel("Tiempo [seg]")
plt.legend()
plt.grid(visible=1)
plt.show()
#plt.savefig("Reconstrucción de la señal con Kuramoto.png")
plt.close()
"""*****OBTENCIÓN DEL PARÁMETRO DE ORDEN*****"""
#Para el modelo de Kuramoto
R_kuramoto=ventanado(y2_kfilt)
R_K_mean=R_kuramoto.mean()
print("Parámetro de orden en el ataque modelo de Kuramoto: "+str(R_K_mean))
#Para el modelo original
R_real=ventanado(etapa_2_filt)
R_real_mean=R_real.mean()
print("Parámetro de orden en el ataque modelo real: "+str(R_real_mean))

R_etapa1 = ventanado(etapa_1_filt)
R_etapa1_mean=R_etapa1.mean()
R_etapa3 = ventanado(etapa_3_filt)
R_etapa3_mean=R_etapa3.mean()
plt.figure(figsize=(20,10))
plt.suptitle("Obtención de parámetro de orden")
plt.subplot(211)
plt.plot([0,10,10,int(hoja[registro_n][3].value)-10,int(hoja[registro_n][3].value)-10,int(hoja[registro_n][3].value)])

,[R_etapa1_mean,R_etapa1_mean,R_K_mean,R_K_mean,R_etapa3_mean,R_etapa3_mean]
,color="green",label="Kuramoto r: "+str("{:.2f}".format(R_K_mean))+ " K: "+str(K_kuramoto))
plt.ylabel("EEG")
plt.xlabel("Tiempo [seg]")
plt.legend()
plt.grid(visible=1)
plt.subplot(212)
plt.plot([0,10,10,int(hoja[registro_n][3].value)-10,int(hoja[registro_n][3].value)-10,int(hoja[registro_n][3].value)])

,[R_etapa1_mean,R_etapa1_mean,R_real_mean,R_real_mean,R_etapa3_mean,R_etapa3_mean]
,color="blue",label="Real r: "+str("{:.2f}".format(R_real_mean)))
plt.ylabel("EEG")

```

```
plt.xlabel("Tiempo [seg]")
plt.legend()
plt.grid(visible=1)
plt.show()
#plt.savefig("Obtención de parámetro de orden.png")
plt.close()
"""*****Gráfica de K y r*****"""
index_r_max=np.where(parametro_de_orden>=0.5)
print("Se presente los valores de K y de r que cumplen con los valores mínimos para que ocurra
la sincronización en el ataque")
print("K_ Kuramoto\tParámetro de orden")

for i in range(np.size(index_r_max)):

print(str("{:.5f}".format(K_kuramoto[index_r_max[0][i]]))+ "\t\t"+str("{:.2f}".format(parametro_de_or
den[index_r_max[0][i]])))

plt.figure(figsize=(20,10))
plt.plot(K_kuramoto,parametro_de_orden)
plt.title("Parámetro de orden en funcion de la constante de acoplamiento")
plt.ylabel("Parámetro de orden r")
plt.xlabel("Constante de acoplamiento K")
plt.show()
#plt.savefig("Constante de acoplamiento en funcion del parámetro de orden.png")
plt.close()
```

Anexo2

Desarrollo de la Metodología para lograr el Objetivo Específico I

Antes de dar inicio a la parte práctica de la presente tesis se inicia recolectando la información de la base de datos abierta y disponible en la página web PhysioNet (<https://physionet.org/content/chbmit/1.0.0/>) (Goldberger, y otros, 2020). Con los archivos EDF descargados, se leen en el formato adecuado los electroencefalogramas de los pacientes incluidos en la base de datos.

Para procesar los datos se utiliza el lenguaje de programación Python. La estructura de Python es similar a la del lenguaje de programación C++, por lo cual el código del presente programa se declara las librerías, funciones, variables a utilizar, y el código en sí de lo que se desea obtener en el programa.

Para realizar este trabajo se importó cinco librerías entre una de la más importante EDFreader que se utiliza para leer los archivos EDF que son archivos de datos. A diferencia de otro tipo de documentos o medios de comunicación, no están destinados para ser visto por el público general. Adicionalmente, este tipo de archivos permite obtener propiedades adicionales del registro de EEG como: el número de canales, el número de datos, la frecuencia de muestreo entre otros. La frecuencia de muestreo es uno de los aspectos más importantes, ya que será utilizada en el análisis en el dominio de la frecuencia necesario para cumplir el objetivo de este trabajo.

La segunda librería de mayor importancia es la que permite realizar la transformada de Fourier FFT, esta devuelve el vector de las frecuencias y su amplitud, que componen la señal transformada.

Al momento de trabajar con la segunda estructura que corresponde a las funciones, el archivo EDF contiene valores de diferentes rangos de amplitud. Para entender estos valores con más legibilidad, se usa la normalización estándar o Z. Esta función resta los valores del promedio de todos los valores y luego divide para la desviación estándar de tal manera que se obtiene que el valor más pequeño y el valor más grande sean proporcionales a los valores originales. Por lo cual los valores obtenidos son valores promedio en función del valor estándar, dependiendo de los varios canales del registro.

En Python se declara las variables:

1. Directorio: que contiene la dirección donde están guardados los archivos, para el caso de este trabajo los datos de los pacientes están en la carpeta registros que contiene 24 archivos. EDF de los pacientes los mismos que se disponen en forma de vector.

2. Nombre-registro: para poder leer cada uno de los correspondientes registros se necesita saber el nombre y la información que contiene el archivo summary de los pacientes que tuvieron un ataque. Toda esta información el programa la dispone en forma de lista.

3. Canal: Esta variable hace referencia a cada uno de los canales específicos donde se ubicaron los electrodos, por lo cual se hace una lista nuevamente con los nombres respectivos.

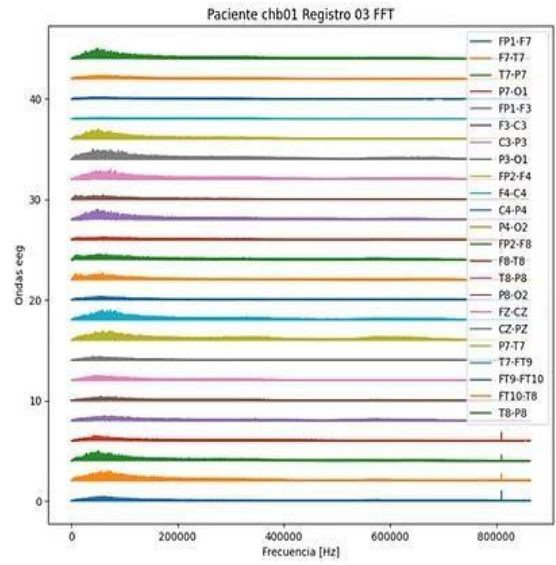
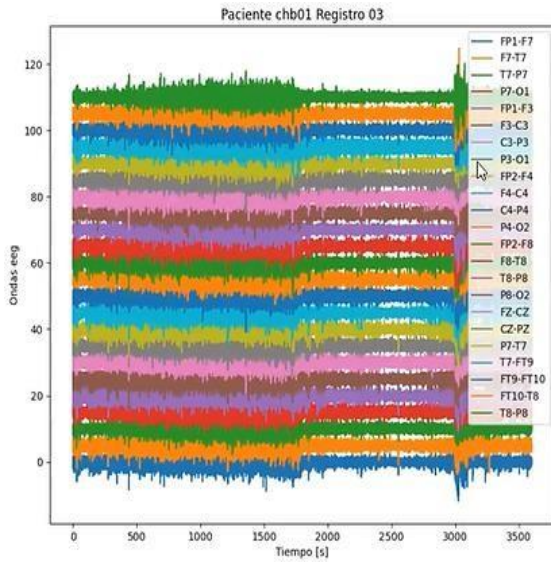
El primer objetivo planteado en el presente trabajo se resuelve con la siguiente metodología: de cada paciente se lee el archivo EDF, se normalizan los datos, y finalmente se obtiene la transformada de Fourier. Adicionalmente se grafican los datos.

Para realizar este proceso es necesario leer los archivos obteniendo una estructura de datos para lo cual es necesario conocer cuántos canales debe tener la estructura de datos y conocer el número de muestras por segundo en función de las frecuencias de muestreo. También es necesario conocer la cantidad de tiempo útil y que se extraerá del registro, dado que el ataque epiléptico ocurre en solo una fracción del tiempo total del registro. Por ejemplo, si de un paciente se toman 30 segundos del registro y el tiempo total es de 3600 segundos, se tomará alrededor de la centésima parte de la grabación de EEG. Al analizar los archivos EDF se encontró que todas las señales fueron muestreadas a 256 muestras por segundo con una resolución de 16 bits, la mayoría de los archivos contienen 23 señales de EEG. Para estos registros se utilizó el sistema internacional 10-20 de posiciones y nomenclatura de electrodos de EEG.

Para comparar e interpretar los datos entre sujetos estos deben ser normalizados en función de la media y la desviación estándar. Ya que las medidas estadísticas proporcionan información sobre la distribución de datos de tal manera que se reduzca la influencia de las diferencias individuales. Esto es necesario ya que cada persona tiene un EEG único y estas señales pueden variar entre individuos, permitiendo una comparación más precisa usando los valores adimensionales en forma de valores z. La normalización también permite resaltar las características más importantes de la señal de EEG y así detectar ataques epilépticos, reduciendo los efectos de movimientos musculares o ruidos externos ya que centra los datos en función de la media y desviación.

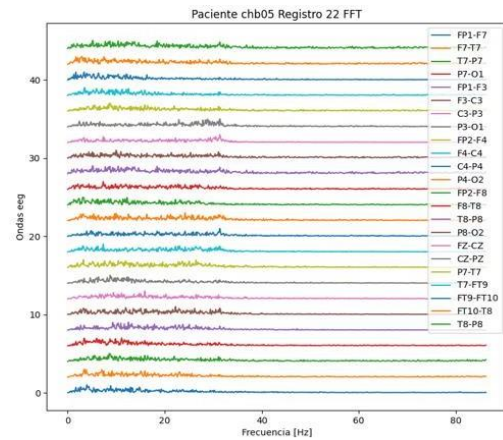
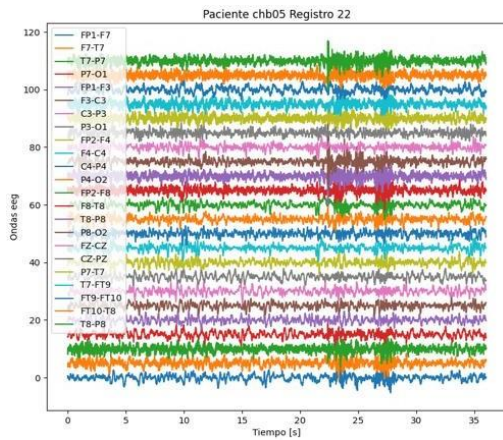
Luego, de tal manera que se pueda realizar una comparación entre la imagen del EEG por cada canal y la respectiva imagen de la transformada de Fourier que para el presente trabajo se utilizó la herramienta de Python que lo hace a través del método FFT. Se toma únicamente la amplitud o magnitud desde la posición 0 del vector hasta la quinceava parte del vector de frecuencias, normalizando la muestra y obteniendo los valores de frecuencia tanto en unidades de muestra por segundo como en Hertzios (Hz) multiplicando el tiempo total al cuadrado y por el número de muestras y así obtenemos los datos tanto en el eje de las abscisas y en el eje de las ordenadas como se muestra en la siguiente figura.

Registro paciente chb01 original con los 23 canales.



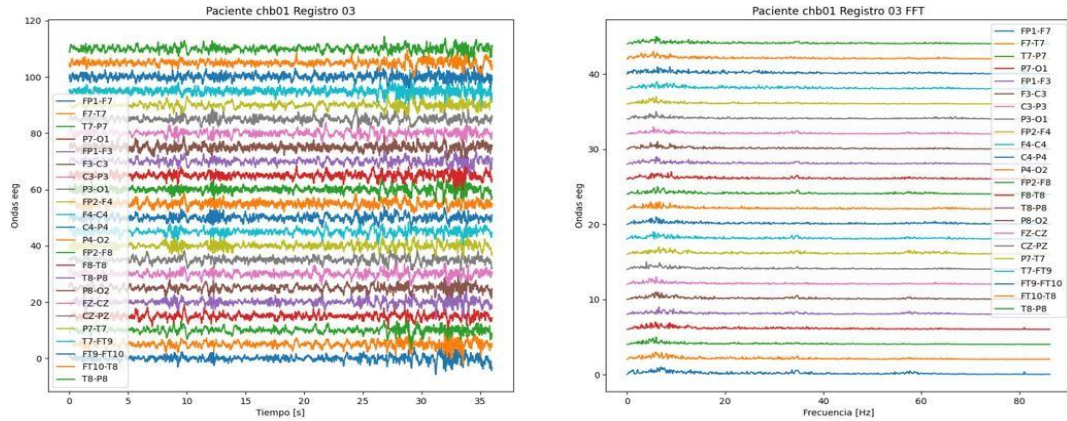
Nota. Elaboración propia

Registro paciente chb01 original con los 23 canales.



Nota. Elaboración propia

Registro paciente chb05 original con los 23 canales.



Nota. Elaboración propia

Anexo3

Desarrollo de la Metodología para lograr el Objetivo Específico II

Para poder detectar los ataques epilépticos, se toma información de la base de datos CHB-MIT Scalp EEG, que contiene un archivo chb nn -summary.txt que contiene información sobre el tiempo transcurrido en segundos desde el comienzo del ataque epiléptico hasta el final del ataque. La siguiente tabla presenta la información que será utilizada para la segunda etapa de esta parte práctica, donde luego de trabajar con un método de múltiples ventanas de la Transformada de la Fourier, se procede a la obtención del parámetro de orden r .

A continuación, la tabla mostrada indica los ataques epilépticos, tiempos de inicio y final del paciente chb02_16 como ejemplo y en la parte final se presenta la tabla con los datos obtenidos de algunos de los veinte y cuatro pacientes.

Registro paciente chb02 original con los 23 canales

1	File Name: chb02_16.edf File Start Time: 09:31:46 File End Time: 9:47:45 Number of Seizures in File: 1 Seizure Start Time: 130 seconds Seizure End Time: 212 seconds
2	File Name: chb02_16+.edf File Start Time: 10:50:23 File End Time: 11:50:23 Number of Seizures in File: 1 Seizure Start Time: 2972 seconds Seizure End Time: 3053 seconds
3	File Name: chb02_19.edf File Start Time: 13:50:45 File End Time: 14:50:45 Number of Seizures in File: 1 Seizure Start Time: 3369 seconds Seizure End Time: 3378 seconds

Nota. Elaboración propia

Registro paciente chb05 original con los 23 canales

1	File Name: chb05_06.edf File Start Time: 22:21:25 File End Time: 23:21:25 Number of Seizures in File: 1 Seizure Start Time: 417 seconds Seizure End Time: 532 seconds
2	File Name: chb05_13.edf File Start Time: 05:22:15 File End Time: 6:22:15 Number of Seizures in File: 1 Seizure Start Time: 1086 seconds Seizure End Time: 1196 seconds
3	File Name: chb05_16.edf File Start Time: 08:22:35 File End Time: 9:22:35 Number of Seizures in File: 1 Seizure Start Time: 2317 seconds Seizure End Time: 2413 seconds
4	File Name: chb05_17.edf File Start Time: 09:22:42 File End Time: 10:22:42 Number of Seizures in File: 1 Seizure Start Time: 2451 seconds Seizure End Time: 2571 seconds
5	File Name: chb05_22.edf File Start Time: 14:23:17 File End Time: 15:23:17 Number of Seizures in File: 1 Seizure Start Time: 2348 seconds Seizure End Time: 2465 seconds

Nota. Elaboración propia

Registro paciente chb03 original con los 23 canales.

1	File Name: chb03_01.edf File Start Time: 13:23:36 File End Time: 14:23:36 Number of Seizures in File: 1 Seizure Start Time: 362 seconds Seizure End Time: 414 seconds
2	File Name: chb03_02.edf File Start Time: 14:23:39 File End Time: 15:23:39 Number of Seizures in File: 1 Seizure Start Time: 731 seconds Seizure End Time: 796 seconds
3	File Name: chb03_03.edf File Start Time: 15:23:47 File End Time: 16:23:47 Number of Seizures in File: 1 Seizure Start Time: 432 seconds Seizure End Time: 501 seconds
4	File Name: chb03_04.edf File Start Time: 16:23:54

	File End Time: 17:23:54 Number of Seizures in File: 1 Seizure Start Time: 2162 seconds Seizure End Time: 2214 seconds
5	File Name: chb03_34.edf File Start Time: 01:51:23 File End Time: 2:51:23 Number of Seizures in File: 1 Seizure Start Time: 1982 seconds Seizure End Time: 2029 seconds
6	File Name: chb03_35.edf File Start Time: 02:51:30 File End Time: 3:51:30 Number of Seizures in File: 1 Seizure Start Time: 2592 seconds Seizure End Time: 2656 seconds
7	File Name: chb03_36.edf File Start Time: 04:51:45 File End Time: 5:51:45 Number of Seizures in File: 1 Seizure Start Time: 1725 seconds Seizure End Time: 1778 seconds

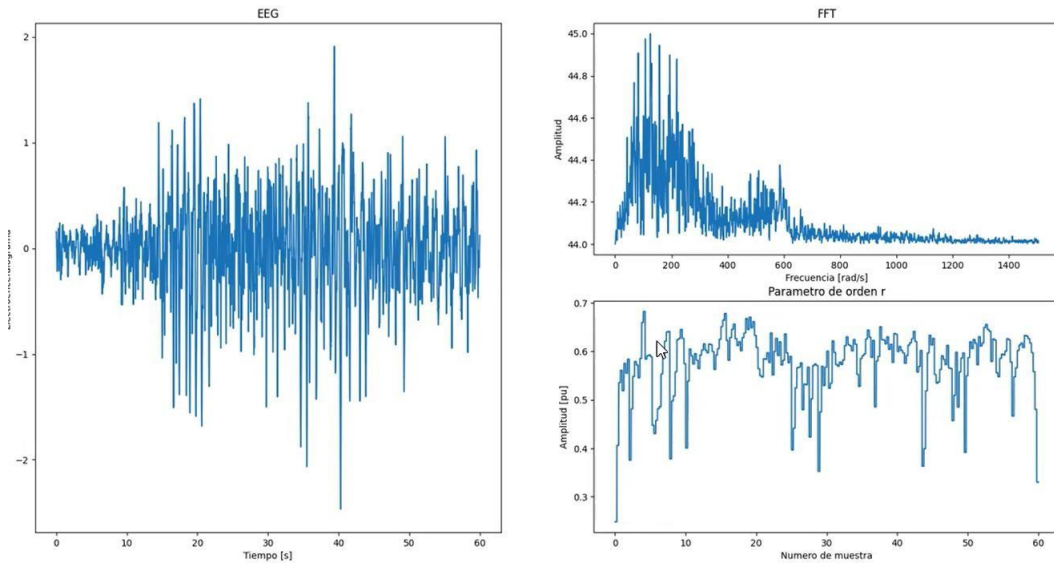
Nota. Elaboración propia

Con la información proporcionada se obtiene dos señales adicionales por cada registro, como se presenta en las imágenes posteriores a este texto, donde a la izquierda se presenta el EEG y al lado derecho está las dos señales obtenidas: la transformada de Fourier y el parámetro de orden r .

La primera imagen correspondiente al EEG es la onda del promedio de los 24 canales ya que cada canal representa una sección del cerebro y si se suma las ondas, el promedio representa el funcionamiento sincronizado del cerebro. Del promedio resultante se obtiene la FFT. Luego, se obtiene un solo parámetro de orden que engloba a todos los canales y que al igual que el EEG es una serie en el dominio del tiempo. Es decir, el parámetro de orden es una mejor medición de la sincronía de las 24 señales.

Promedio de los canales del registro de EEG y las señales derivadas: transformada de Fourier y parámetro de orden.

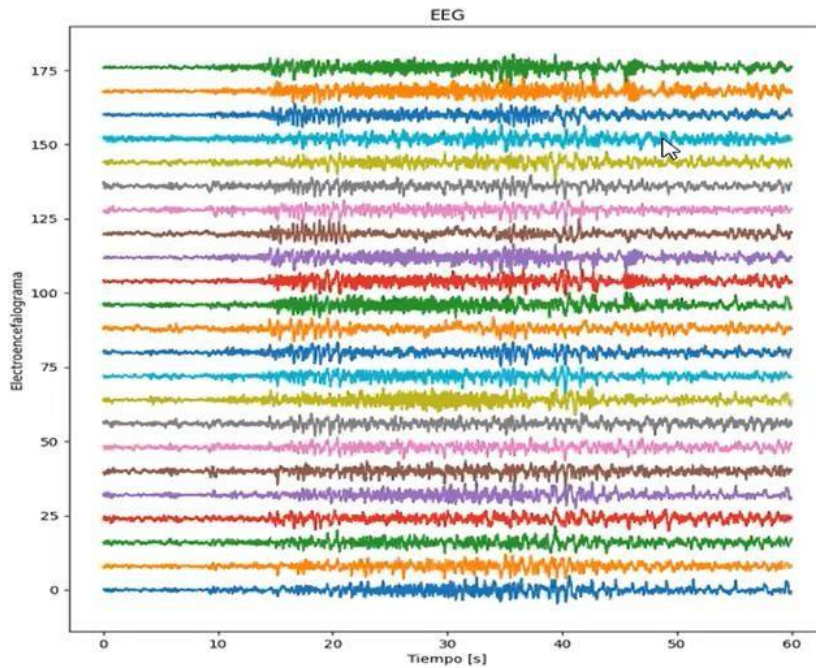
Paciente chb01 Registro Nro 03



Nota. Elaboración propia

Series de tiempo normalizadas de los 24 canales de EEG

Paciente chb01

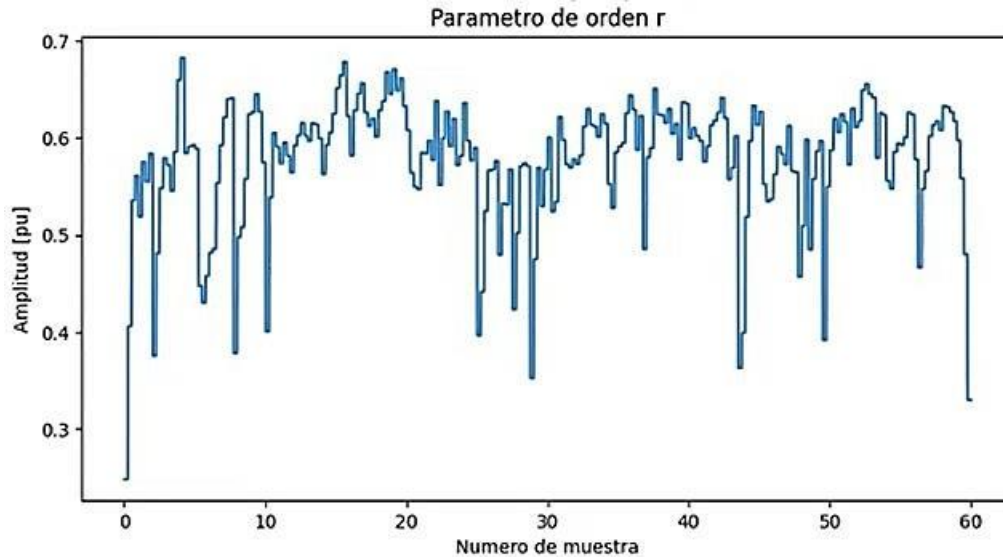


Nota. Elaboración propia

El método más eficiente para obtener el parámetro de orden que se adapte a los cambios de frecuencia de la señal de EEG es: Dividir la onda en segmentos de cinco segundos y obteniendo la FFT en cada uno de ellos. Luego se obtiene el parámetro de orden en el pico de

frecuencia. Posteriormente, el siguiente segmento no es la continuación desde los 5 segundos si no por lo contrario se realiza una nueva segmentación desde el segundo 0.25 hasta 5.25 segundos; nuevamente se obtiene la FFT y otro vector de parámetros de orden. La finalidad es obtener un valor promedio en el tiempo en que se dividen los segmentos, así se van sumando todos los segmentos divididos para obtener un parámetro de orden total, este proceso o barrido se realiza hasta terminar el tiempo de la señal.

Parámetro de orden con el método de enventanado.



Nota. Elaboración propia

El método de enventanado (Windows slice) permite eliminar el ruido ya que este es uno de los mayores problemas que presenta el EEG debido a la mezcla de las señales del cerebro que sumada a los ruidos que existen por los alambres que conectan los electrodos y la máquina como tal no permite realizar un análisis certero. En teoría el parámetro de orden observado en la figura anterior se acerca más a la realidad, aplicando la técnica de enventanado para lo cual se utiliza la señal de onda original, para luego realizar los promedios correspondientes.

Para acceder de manera más fácil, el proceso del código correspondiente se coloca los archivos de forma manual en el programa de tal manera que no se puede obviar ningún ataque epiléptico de los pacientes entonces se colocaron todos los datos sobre el inicio y fin de los ataques en un archivo de Excel, que contiene cuatro columnas: el nombre del archivo, tiempo de inicio del ataque, tiempo del fin del ataque y el tiempo de simulación a ser realizado con el modelo de Kuramoto, el último parámetro se obtiene restando el tiempo final menos el tiempo inicial. Además, se agregó un tiempo de 10 segundos antes y después del ataque, y así se obtiene las tres etapas del ataque para luego leer la información de Excel y registrar la información en Python archivo por archivo para determinar cuál es el valor del parámetro de orden r en cada etapa y se

mantenga con la línea de tiempo respectivo. Esto facilitó el análisis de los datos, ya que al inicio y final del ataque epiléptico pueden existir particularidades. Tal como en el paciente correspondiente al archivo con nombre chb24. Este paciente durante todo el tiempo del examen presenta ataques epilépticos. Registros de características similares permitirían realizar un análisis mucho más detenido del parámetro r intraindividual. Y así poder determinar la sincronía que nos indica la presencia de un ataque comprobado según los tiempos de ataque indicados por los expertos.

Además, el archivo de Excel también es útil y necesario para determinar el tamaño de la muestra para comenzar a realizar el barrido y el promedio antes indicado y así aplicar la transformada de Fourier para encontrar el parámetro de orden. En este caso el valor es de 512 ya que el teorema de Nyquist, recomienda que para reconstruir una señal se debe al menos duplicar la señal de 256 Hz a 512 por la razón que la medición es más acertada, esto va a depender en nuestro caso del tiempo que dure cada ataque.

A continuación, se determina las frecuencias en un espectro que debe ser dividido en dos partes para escoger la amplitud más alta y así con cada mitad hasta que ya no exista valores disponibles. En el espectro de frecuencias se escoge las amplitudes que aportan más a la onda del EEG para reconstruir la señal con ciertas frecuencias.

Si consideramos todas las amplitudes, que en la transformada de Fourier es igual al número de frecuencias discretas, la cantidad de recursos informáticos sería elevada y también elevado el tiempo para llevarlo a cabo. Para el presente trabajo se verifica donde comienza la frecuencia de 80Hz con una holgura de 10 Hz y en ese rango obtener la reconstrucción de la señal tanto en datos como en frecuencias, según la amplitud obtenida de la frecuencia y la fase ya que en la Transformada de Fourier todas las señales tienen un desfase según la siguiente fórmula:

$$\sum A \sin (wt + \phi)$$

Sumatoria de todas las señales sinusoidales que aporta para obtener la señal original.

Por lo cual se necesita saber la amplitud y el desfase en esa frecuencia con la finalidad de obtener la amplitud máxima que se considera el primer valor del espectro de frecuencias, o pico de frecuencia, para ubicar la posición en base a la cual se obtiene el desfase. Después debe sumarse a la frecuencia inicial y como resultado se obtiene el ángulo expresado en radianes sobre segundos.

Para reconstruir la señal se debe aplicar la Transformada de Fourier (sumatoria de señales sinusoidales a diferentes frecuencias) utilizando la amplitud, fase y frecuencia y tiempo, 10 segundos antes y 10 segundos después, de cada onda senoidal. Debido a que la Transformada de Fourier indica las componentes de frecuencia y amplitud de la señal original este procedimiento devuelve una representación simple que permite analizar y manipular la señal de manera conveniente mediante la siguiente fórmula.

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos(\omega_n t - \theta_n)$$

Donde ω_n es la frecuencia de cada rango de frecuencia con mayor aporte. Así al analizar la señal de EEG se obtiene una señal que indica un error, por lo cual se reconstruye la señal y con esta señal se aplica Kuramoto ya que con esta opción se obtiene una señal más cercana a la original según esto se puede mejorar o disminuir mediante el modelo de Kuramoto.

Desarrollo de la Metodología para lograr el Objetivo Específico III

Para implementar el modelo de Kuramoto se necesitan los siguientes elementos: amplitud, fase, frecuencia, el tiempo total de simulación del EEG y la constante de acoplamiento K . Según la fórmula original del modelo de Kuramoto es necesario analizar en cada iteración del tiempo la frecuencia angular que puede ser diferente dependiendo del nivel de acoplamiento de la señal. Por lo tanto, la fórmula se debe aplicar dependiendo del número de osciladores, es decir se tiene que evaluar como cada oscilador evoluciona en cada iteración del tiempo. Como consecuencia, la sumatoria de todas las iteraciones debe dar la evolución de cada señal sinusoidal. Esto permite reconstruir la señal con el valor de K a través del siguiente código:

Código realizado en Python de modelo de Kuramoto

```

y=np.zeros(np.size(tiempo))
theta=np.zeros([np.size(fase_ini),np.size(tiempo)])
w1=np.zeros([np.size(fase_ini),np.size(tiempo)])
amp1=np.zeros([np.size(fase_ini),np.size(tiempo)])
# Se colocan los primero valores de la primer iteración
w1[:,0]=frecuencia_ini
# theta[:,0]= fase_ini + w1[:,0]*tiempo[0]
# amp1[:,0]=amp_ini
# for j in range(np.size(amp1[:,0])):
#     y[0]=y[0]+amp1[j,0]*np.sin(theta[j,0])
#
# for k in range(np.size(w1[:,0])):
#     w1[k,1]=w1[k,0]
#     for l in range(np.size(w1[:,0])):
#         w1[k,1]=w1[k,1]+K*(np.sin(theta[k,0]-theta[l,0]))/np.size(amp_ini)
# resto de iteraciones
for i in range(np.size(tiempo)):
    theta[:,i]= fase_ini + w1[:,i]*tiempo[i]
    amp1[:,i]=amp_ini
    for j in range(np.size(amp1[:,i])):
        y[i]=y[i]+amp1[j,i]*np.sin(theta[j,i])
    if i < np.size(tiempo)-1:
        for k in range(np.size(w1[:,i])):
            w1[k,i+1]=w1[k,i]
            for l in range(np.size(w1[:,i])):
                w1[k,i+1]=w1[k,i+1]+K*(np.sin(theta[k,i]-theta[l,i]))/np.size(amp_ini)

```

Nota. Elaboración propia.

Código realizado en Python con la etapa de reconstrucción de la señal con Kuramoto.

```

_kuramoto=0.00015
z_k=zScore(modelo_kuramoto(param_k_etapa2[0],param_k_etapa2[2],param_k_etapa2[1],tiempo_2,K_kuramoto))
z_kfilt= butter_lowpass_filter(y2_k,np.array([3,70.0]),sampling_rate,2,'bandpass')
_graph=np.concatenate((etapa_1,y2_kfilt,etapa_3),axis=0)
lt.figure(figsize=(20,10))
lt.title("Reconstrucción de la señal con Kuramoto")
lt.plot(tiempo,y_graph,label="Kuramoto K: "+str(K_kuramoto))
lt.plot(tiempo,z_signal_mean+5,label="Señal Real")
lt.ylabel("EEG")
lt.xlabel("Tiempo [seg]")
lt.legend()
lt.grid(visible=1)
lt.show()
plt.savefig("Reconstrucción de la señal con Kuramoto.png")

```

Nota. Elaboración propia.

Anexo4

Simulaciones.

Para realizar las distintas simulaciones se comienza con la lectura del archivo del EEG de un determinado registro de pacientes en esta sección se presentan los resultados del registro número 8, debido a que el tiempo es pequeño y es más fácil observar la señal del ataque en sus tres etapas.

Se asignó diferentes valores de K hasta que sincronice la señal, esto implica varios intentos de prueba y error.

Extracto de la lista de pacientes con información de los tiempos del ataque

1	chb01_03	2996	3036	60
2	chb01_04	1467	1494	47
3	chb01_15	1732	1772	60
4	chb01_16	1015	1066	71
5	chb01_18	1720	1810	110
6	chb01_21	327	420	113
7	chb01_26	1862	1963	121
8	chb02_16+	2972	3053	101
9	chb02_19	3369	3378	29

Nota. Elaboración propia.

Se obtiene el nombre del archivo, se manda a leer y se obtiene el número de canales, la frecuencia de muestreo y se obtiene las señales. Determinando el tiempo de la señal a leer, que es el tiempo donde inicia el ataque restado los 10 segundos antes del ataque y sumando 10 segundos después del ataque, se lee el tamaño total de los datos mediante el análisis de la longitud de las muestras temporales esperadas desde la simulación se obtiene multiplicando el número de muestras por segundo y el tiempo total de la simulación. Al hacer esto, si el ataque dura 29 segundos y el tiempo de muestreo es 256 debe devolver 29 x 256 valores a partir del valor inicial.

Al momento de manejar los datos se divide la señal en tres etapas a partir de la señal promediada según lo muestra la siguiente imagen.

Tiempos de las tres etapas de un ataque epiléptico.

Etapas	Intervalos (señal z_signal)
Preictal	Desde 0 hasta 10 segundos.
Ictal	Desde los 10 segundos hasta que termina el ataque.
Postictal	Desde que termina el ataque hasta el final de la señal.

Nota. Elaboración propia.

Todos los valores deben multiplicarse por la frecuencia de muestreo y así poder mover posiciones según el número de muestra. Se filtra la información en un filtro pasa bajos en cada etapa para mostrar de mejor manera la gráfica de Fourier tomando en cuenta las tres etapas del ataque epiléptico en función de la gráfica original. Se debe considerar que el Modelo de Kuramoto se aplica a la etapa Ictal ya que en ese momento existe una sincronización es decir K tiene un valor determinado.

Para reconstruir nuevamente la señal se necesita la amplitud, la frecuencia y la fase, en función de los datos del número complejo de la Transformada de Fourier y las frecuencias de esta transformación. Por lo tanto, se obtienen la amplitud, frecuencia y fase de 7424 muestras, que corresponden a los 29 segundos del registro antes mencionado.

Se aplicó la serie de Fourier que devuelve la señal del EEG sin aplicar el modelo de Kuramoto y se realizó una comparación posterior con la gráfica original de la señal.

En la etapa final, aplicando el modelo de Kuramoto, se toma en cuenta el valor de acoplamiento K que se determinó para obtener un valor de parámetro r similar al original. Entonces, se reconstruye la señal tomando el período Ictal y se asegura que el valor de K no tenga el valor de 0 o nula. Posteriormente se filtra la señal debido a que no se toma todas las frecuencias de la señal original, pues las amplitudes bajas de frecuencias altas generan ruido. Se empleó un filtro pasa bajos, permitiendo que permanezcan las amplitudes dentro de un espectro de frecuencia típico de EEG, es decir de un intervalo acotado. Se obtiene una señal parecida a la original del EEG, y se agregan los segmentos de las otras etapas para completar la reconstrucción de la señal.

La obtención del parámetro de orden r permite realizar la medición de la sincronización según el parámetro de acoplamiento K , para lo cual se debe obtener el parámetro r tanto de la señal original como de la señal reconstruida con el modelo de Kuramoto. Para simplificar el análisis, se puede usar el promedio temporal de cada una de las etapas del ataque, para que se pueda observar que cuando ocurre el ataque el valor de r se incrementa y que entre la etapa de pre-ictal y postictal no se tiene ataque ya que el valor de r es cercano a cero.

Todos los valores deben multiplicarse por la frecuencia de muestreo y así poder mover posiciones según el número de muestra. Se filtra la información en un filtro pasa bajos en cada etapa para mostrar de mejor manera la gráfica de Fourier tomando en cuenta las tres etapas del ataque epiléptico en función de la gráfica original. Se debe considerar que el Modelo de Kuramoto se aplica a la etapa ictal ya que en ese momento existe una sincronización es decir K tiene un valor determinado.

Para reconstruir nuevamente la señal se necesita la amplitud, la frecuencia y la fase, en función de los datos del número complejo de la Transformada de Fourier y las frecuencias de esta transformación. Por lo tanto, se obtienen la amplitud, frecuencia y fase de 7424 muestras, que corresponden a los 29 segundos del registro antes mencionado.

Se aplicó la serie de Fourier que devuelve la señal del EEG sin aplicar el modelo de Kuramoto y se realizó una comparación posterior con la gráfica original de la señal.

En la etapa final, aplicando el modelo de Kuramoto, se toma en cuenta el valor de acoplamiento K que se determinó para obtener un valor de parámetro r similar al original. Entonces, se reconstruye la señal tomando el período ictal y se asegura que el valor de K no tenga el valor de 0 o nula. Posteriormente se filtra la señal debido a que no se toma todas las frecuencias de la señal original, pues las amplitudes bajas de frecuencias altas generan ruido. Se empleó un filtro pasa bajos, permitiendo que permanezcan las amplitudes dentro de un espectro de frecuencia típico de EEG, es decir de un intervalo acotado. Se obtiene una señal parecida a la original del EEG, y se agregan los segmentos de las otras etapas para completar la reconstrucción de la señal.

La obtención del parámetro de orden r permite realizar la medición de la sincronización según el parámetro de acoplamiento K , para lo cual se debe obtener el parámetro r tanto de la señal original como de la señal reconstruida con el modelo de Kuramoto. Para simplificar el análisis, se puede usar el promedio temporal de cada una de las etapas del ataque, para que se pueda observar que cuando ocurre el ataque el valor de r se incrementa y que entre la etapa de preictal y postictal no se tiene ataque ya que el valor de r es cercano a cero.