



UNIVERSIDAD  
CATÓLICA DE  
TEMUCO

Universidad Católica de Temuco  
Facultad de Ingeniería  
Departamento de Ciencias Matemáticas y Física

SIMULACIÓN DE SEÑALES GENERADAS POR TEMBLORES EN  
LA ENFERMEDAD DE PARKINSON, MEDIANTE  
APROXIMACIÓN DE MODELOS NARMAX POLINOMIALES

Por

Lizandro Emir García Tovar

Profesor Guía

Dr. Julio Rojas

Actividad Formativa Equivalente, para optar al grado de  
Magíster en Matemáticas Aplicadas

Temuco - 23 de marzo de 2021

Universidad Católica de Temuco  
Facultad de Ingeniería  
Departamento de Ciencias Matemáticas y Física

**COMISIÓN EVALUADORA**

Profesor Guía:

.....  
Dr. Julio Rojas

Profesor informante:

.....  
Dr. Stefan Berres

Profesor informante:

.....  
Dr. David Tempo

Director del Programa (Ministro de fe):

.....  
Dr. Jacobo Hernández Montelongo.

Temuco, 23 de Marzo de 2021.

## Perfil de Egreso

Magíster en Matemáticas Aplicadas.  
Universidad Católica de Temuco.

El egresado del Magíster en Matemáticas Aplicadas es un profesional postgraduado que posee la competencia de aplicar la matemática al análisis de sistemas dinámicos y evolutivos. Específicamente:

1. Formula ecuaciones diferenciales como modelos matemáticos para obtener una relación cuantitativa entre las variables relevantes de sistemas dinámicos y evolutivos.
2. Resuelve ecuaciones diferenciales como modelos matemáticos, utilizando técnicas numéricas y analíticas, para obtener valores cuantitativos de la variable respuesta del sistema.
3. Desarrolla y/o utiliza programas computacionales en la resolución, análisis y aplicación de ecuaciones diferenciales en sistemas dinámicos y evolutivos.
4. Comunica información científico-matemática con rigurosidad técnica y claridad.

# Agradecimientos

Gracias a Dios.

Gracias a mi familia.

Gracias a la UCT y la dirección de la Maestría en Matemáticas Aplicadas por abrirme las puertas de la institución y poder formar parte de esta familia académica.

Gracias a los profesores del programa y todas esas personas anónimas que ayudaron a hacer esto posible.

Gracias Chile.

## Abstract

In this work, a methodology is developed to identify the dynamics behind tremors at rest in Parkinson's disease, using acceleration records based on actual tremors present in a person suffering from the disease. A statistical and dynamic examination is carried out in order to evaluate the capacity of the polynomial NARMAX models found and evaluated by time windows, to predict and represent the underlying dynamics from the approximation of their responses to the values of the series of time of real accelerations, using the simulated annealing algorithm to optimize the parameters of the input function (input signal), and the forward orthogonal regression algorithm for the identification of the NARMAX models.

**Keywords:** Resting tremor, nonlinear dynamics, NARMAX, dynamical systems, chaotic dynamics systems, attractor, Lyapunov maximum exponent, simulated annealing..

## Resumen

En este trabajo se desarrolla una metodología de identificación de la dinámica que hay detrás de los temblores en reposo en la enfermedad de Parkinson, empleando para ello registros de aceleraciones a partir de temblores reales presentes en una persona que padece la enfermedad. Se lleva a cabo un exámen estadístico y dinámico a modo de evaluar la capacidad que tienen los modelos NARMAX polinomiales hallados y evaluados por ventanas temporales, para predecir y representar la dinámica subyacente a partir de la aproximación de sus respuestas a los valores de las series de tiempo de aceleraciones reales, empleando el algoritmo de recocido simulado para optimizar los parámetros de la función de entrada (señal de entrada), y el algoritmo de regresión ortogonal hacia adelante para la identificación de los modelos NARMAX.

**Palabras claves:** Temblor en reposo, dinámica no lineal, NARMAX, sistemas dinámicos, sistemas dinámico caóticos, atractor, máximo exponente de Lyapunov, recocido simulado.

# Índice

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
<b>2</b>	<b>OBJETIVOS</b>	<b>2</b>
<b>3</b>	<b>MATERIALES Y MÉTODOS</b>	<b>3</b>
3.1	Modelo conceptual . . . . .	3
3.1.1	Planteamiento del problema . . . . .	3
3.1.2	Estado del arte en la identificación de la dinámica de temblores en la enfermedad de Parkinson . . . . .	3
3.1.3	Delimitación del problema de investigación abordado . . . . .	5
3.1.4	Descripción del proceso de muestreo de las señales de aceleraciones . . . . .	6
3.2	Modelo matemático . . . . .	8
3.2.1	Modelo NARMAX polinomial. . . . .	8
3.2.2	Selección de términos y parámetros en la estructura NARMAX. . . . .	11
3.2.3	Criterios para validación del modelo. . . . .	12
3.3	Caracterización de las muestras de aceleración $a_x$ y $a_y$ , tomadas de una persona con EP. . . . .	13
3.3.1	Autocorrelación de las series de tiempo $a_x$ y $a_y$ . . . . .	14
3.3.2	Análisis espectral de las series de tiempo $a_x$ y $a_y$ . . . . .	14
3.3.3	Invariantes cualitativos de las series de tiempo $a_x$ y $a_y$ . . . . .	17
<b>4</b>	<b>DISCUSIÓN DE RESULTADOS EN LA IDENTIFICACIÓN DE LA DINÁMICA</b>	<b>21</b>
4.1	Resultados de predicción por ventanas para segmento $a_x$ . . . . .	23
4.2	Resultados de predicción por ventanas para segmento $a_y$ . . . . .	27
<b>5</b>	<b>CONCLUSIONES</b>	<b>32</b>
<b>6</b>	<b>BIBLIOGRAFÍA</b>	<b>33</b>
<b>7</b>	<b>ANEXOS</b>	<b>35</b>
7.0.1	Anexo A.1 : Densidad espectral de energía (DEE) . . . . .	35
7.0.2	Anexo A.2 : La Transformada de Gabor . . . . .	35
7.0.3	Anexo A.3 : Teorema de inmersión de Takens . . . . .	36
7.0.4	Anexo A.4 : Definición de máximo exponente de Lyapunov . . . . .	36
7.0.5	Anexo A.5 : Regresores y parámetros resultantes para modelo NARMAX de $a_x$ con predicción MPO para primera iteración . . . . .	37
7.0.6	Anexo A.6 : Regresores y parámetros resultantes para modelo NARMAX de $a_y$ con predicción MPO para primera iteración . . . . .	38
7.1	Anexo B . . . . .	39
7.1.1	Anexo B.1 : Cálculo de autocorrelación . . . . .	39
7.1.2	Anexo B.2 : Análisis espectral de señales originales. . . . .	40
7.1.3	Anexo B.3 : Invariantes cualitativos de señales. . . . .	42
7.1.4	Anexo B.4 : Identificación de parámetros para función de entrada (función forzadora) de los modelos NARMAX. . . . .	44

7.1.5	Anexo B.5 : Predicciones por ventana. . . . .	45
7.1.6	Anexo B.6 : Gráficas de resultados. . . . .	49

# 1 INTRODUCCIÓN

La enfermedad de Parkinson constituye el segundo trastorno neurodegenerativo más extendido después del Alzheimer. Uno de los síntomas más evidentes es el temblor en reposo que suele afectar principalmente las extremidades. Se ha intentado modelar el temblor propio de la enfermedad de Parkinson desde los años sesenta y aún hasta la fecha no se ha logrado dar con un modelo matemático que describa la dinámica al 100 % de similitud. El presente trabajo constituye una propuesta para desvelar el trasfondo matemático detrás de los temblores en reposo en la enfermedad de Parkinson (EP). Para ello se plantea un enfoque enmarcado en el área de sistemas complejos y dinámica caótica, evaluando la aproximación de las soluciones numéricas de modelos NARMAX (hallados por ventanas temporales), tanto en lo referente a la composición frecuencial, como en los invariantes dinámicos subyacentes.

Para realizar esta tarea, se parte del registro de las aceleraciones (series de tiempo de aceleraciones) producidas por el temblor en una persona con EP, empleando una aplicación comercial instalada en un smartphone a una frecuencia de muestreo de 200Hz. Se estudian las características de las series de tiempo, incluyendo un análisis frecuencial usado para la identificación de la dinámica del temblor. Luego se parametrizan las funciones de entrada a los modelos, empleando un algoritmo de optimización metaheurística (Recocido Simulado) para encontrar la mejor correlación entre la señal simulada y la real y por último se plantea un modelamiento NARMAX y predicción por ventanas temporales variables (desde 5ms a 50ms) para luego establecer comparaciones estadísticas entre la simulación y la serie de tiempo real.

## 2 OBJETIVOS

### Objetivo general

Simular señales de aceleración producidas por temblores en la enfermedad de Parkinson, mediante aproximación por ventanas temporales de modelos NARMAX polinomiales.

### Objetivos específicos

1. Registrar señales de aceleración en una persona con enfermedad de Parkinson.
2. Identificar las características no lineales y frecuenciales de las señales registradas, para definir la estrategia de aproximación.
3. Comparar numéricamente las respuestas de los modelos NARMAX polinomiales, respecto a las señales reales de aceleración.

## 3 MATERIALES Y MÉTODOS

### 3.1 Modelo conceptual

#### 3.1.1 Planteamiento del problema

La enfermedad de Parkinson (EP), es una afección neurodegenerativa crónica caracterizada por bradicinesia (movimiento lento), rigidez (aumento del tono muscular), temblor y pérdida del control postural [1]. Esta enfermedad representa el segundo trastorno neurodegenerativo por su frecuencia, situándose por detrás de la enfermedad de Alzheimer. Actualmente se desconoce la causa exacta de la EP, pero todo parece indicar que se trata de una combinación de ciertos factores ambientales, genéticos, el daño oxidativo y el proceso normal de envejecimiento, con un probable papel fundamental de la conexión intestino-cerebro.

El temblor característico de EP es de reposo, se detecta cuando los músculos involucrados no están activados voluntariamente y disminuye durante la mantención de una postura o al realizar un movimiento. Frecuentemente compromete las extremidades superiores con un movimiento de oposición alternante del pulgar y el índice, dando la clásica apariencia de cuenta monedas.

El temblor en reposo de la EP tiene una frecuencia de 4 Hz a 6 Hz y responde inherentemente a una dinámica no lineal <sup>1</sup>[2], suele aumentar al distraer al paciente o al hacerlo realizar alguna tarea que requiera concentración, desaparece durante el sueño y empeora con la ansiedad. La metodología de identificación NARMAX ya se ha empleado para modelar señales fisiológicas, así que es posible cuestionarse la posibilidad de emular una serie temporal basada en el registro de las aceleraciones generadas en el temblor en reposo de la EP, usando para ello un modelo NARMAX cuyos términos, parametros y entradas abrá que determinar, lo cual se ha convertido en la motivación de la presente investigación.

#### 3.1.2 Estado del arte en la identificación de la dinámica de temblores en la enfermedad de Parkinson

Los intentos por encontrar un modelo válido para describir los temblores en reposo presentes en la enfermedad de Parkinson se iniciaron en los años sesenta pero en las últimas dos décadas , el aumento de la potencia de cómputo y la implementación de técnicas novedosas de análisis y procesamiento de datos ha permitido a los investigadores desarrollar modelos

---

<sup>1</sup>En ente tipo de sistemas no hay proporcionalidad, pequeñas causas pueden provocar grandes efectos y la totalidad es mayor a la suma de sus partes.

cada vez más aproximados que imiten las manifestaciones de la enfermedad y que permitan desarrollar herramientas para diagnosis en pacientes o intentar controlar sus manifestaciones. Entre los trabajos en esta dirección podemos citar el realizado por Haeri Mohammad et al.[3], quienes desarrollaron un modelo de estructura fisiológica incorporando funciones propias de algunas partes del cerebro que intervienen en la enfermedad. Este modelo está estructurado en bloques cuyas funciones de transferencias de primer orden representan las dinámicas de la sustancia negra, el cuerpo estriado, el glóbulo pálido y núcleo subtalámico. Se comparó la señal de velocidad resultante con la encontrada en bases de datos de pacientes (sin detalles respecto a la orientación del movimiento en el espacio), encontrando que luego de ajustar los parámetros del modelo, se conseguía similitud en el espectro de frecuencias y específicamente en la frecuencia predominante de la data real. Con este modelo fué posible simular control terapeutico de los temblores de forma general.

Los investigadores Rehan M y Hong K-S [4], analizaron el modelado y el control por retroalimentación automática del temblor (postural y de reposo) mediante estimación de parámetros basada en la metodología de estimulación cerebral profunda. Para ello se basaron en un modelo matemático de oscilador lineal (con la posibilidad de incorporar no linealidad mediante una serie de potencia a considerar) que representaba toda la dinámica, la cual cambia según los parámetros del generador de estímulo cerebral y como salida del modelo se obtendría la señal del temblor. Mediante retroalimentación de la señal del temblor y una ley de control, se manipulaba la amplitud de uno de los parámetros para lograr atenuar los temblores en la simulación del lazo de control. En el trabajo de Shah Vrutangkumar et al.[5], exploran el vínculo entre el aumento de los retrasos de tiempo (observado como un aumento en los tiempos de respuesta) y el temblor parkinsoniano. Para ello primero hicieron observaciones cualitativas respaldadas por hechos clínicos, explicadas utilizando la teoría de control por retroalimentación y reproducidas por la simulación numérica de un péndulo controlado y dos ejemplos experimentales extrapolando los resultados a una posible aplicación para el diagnostico de la enfermedad.

Una aplicación interesante es la desarrollada por Adibah M. Zulkefli et al.[6], quienes diseñaron un guante supresor de temblores parkinsonianos. Este guante emplea un giroscopio controlado, que contrarresta las fuerzas instantaneas, con lo cual se logró una reducción muy importante de las oscilaciones en el rango de frecuencias parkinsoniana. Sin embargo el modelado aquí se enfocó en el mecanismo supresor de las vibraciones y no en la dinámica del temblor. Shahtalebi Shan et al.[7], realizaron el primer modelo basado en aprendizaje profundo de redes neuronales para la identificación del temblor parkinsoniano y su distinción del temblor esencial u otros movimientos voluntarios de la mano. Con el entrenamiento de la red neuronal empleando bases de datos de 3 años a partir de 81 pacientes, se logró capacitar a la red para reconocer el temblor parkinsoniano e incluso la predicción del movimiento aunque con un horizonte reducido. Las capacidades y validez de este modelo tienden a ser proporcionales a la cantidad de datos empleados para entrenar la red neuronal. Es evidente que todavía queda mucho por saber respecto a la dinámica real subyacente en esta patología y se puede observar que los esfuerzos se orientan a tratar la dinámica a través de un modelo de caja negra o gris.

### 3.1.3 Delimitación del problema de investigación abordado

La presente investigación se concentra en evaluar la capacidad del procedimiento de modelado matemático empleado aquí, para describir las características de las señales en el tiempo (frecuencias, magnitudes, fases, etc) , de tal forma que se pueda establecer algún patrón de predicción ante movimientos producidos por temblores en reposo de una persona con EP y representar la dinámica subyacente. Para ello, se parte de una serie temporal de aceleraciones obtenida empleando una aplicación comercial que maneja el acelerómetro incorporado en el dispositivo (smartphone) y permite registrar gran cantidad de datos a diferentes frecuencias de muestreo . Basado en estudios realizados por investigadores de la EP, podemos decir que las señales obtenidas corresponden a la respuesta de un modelo no lineal, el cual probablemente puede cambiar según las condiciones en las que se presente el movimiento del smartphone sostenido en la mano. Por esta razón, la investigación se delimita al estudio de las señales generadas por movimientos del temblor en reposo de una persona con EP sin mayor profundidad en las características de EP que padece.

El modelado de sistemas no lineales a partir de sus señales, es un área incipiente actualmente y muy complejo, sin embargo existen metodologías poderosas que permiten encontrar modelos de caja negra como el método de identificación NARMAX. Las muestras de aceleración que entrega el acelerómetro del smartphone corresponden a las direcciones mostradas en la Figura 1. **Cuando una persona con EP sostiene el smartphone con su vista sobre la pantalla, los movimientos en el eje z pierden relevancia ante los movimientos que se generan en el plano X-Y. Por tanto, pensando en una posible aplicación de este estudio en un software para ayudar en la lectura en pantalla , el modelo no lineal a emplear estará destinado a describir, las aceleraciones en el plano X-Y.**

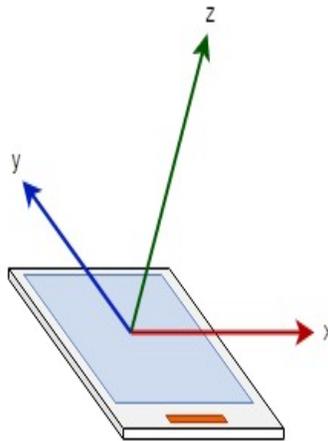


Figura 1: Direcciones de los ejes para la aceleración en un Smartphone. **Imagen de creación propia.**

### 3.1.4 Descripción del proceso de muestreo de las señales de aceleraciones

Para garantizar la obtención de una representación fidedigna de las señales de aceleración registradas, el procedimiento se apoya en el siguiente teorema;

*Teorema del Muestreo de Nyquist-Shannon:* Si una función del tiempo  $f(t)$ , no contiene componentes de frecuencias superiores a  $f_M$  ciclos por segundo, entonces  $f(t)$  se puede determinar por completo mediante sus valores separados por intervalos uniformes de  $1/(2f_M)$  segundos [8, pag 151].

La adquisición de muestras se realiza con un software comercial (Figura 2). Dicho software permite registrar la aceleración lineal (aceleración rectificada sustrayendo la aceleración de la gravedad), garantizando una frecuencia máxima de 200 Hz con lo cual se cubre el rango de frecuencias características de los temblores en reposo de la EP [9] y se cumple con el teorema del muestreo.

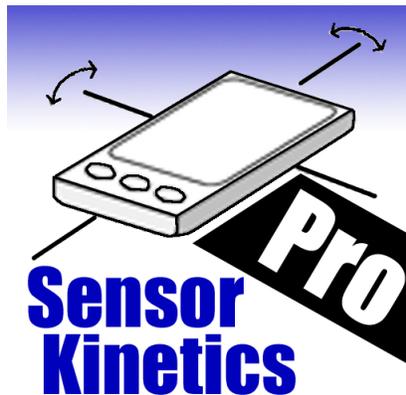


Figura 2: Software para visualización y registro de sensores del Smarthphone basados en Android. **Fuente:** <https://play.google.com/store/apps>.

Desde el punto de vista investigativo para este trabajo, la población de personas con EP en estudio ha sido (contando con la colaboración del personal encargado) las que conforman el Club de Personas con Enfermedad de Parkinson de la ciudad de Temuco (en total quince personas), luego una muestra conformada por dos voluntarios con EP (figura 3), varónes de 75 y 69 años de edad, uno sin medicación con avance de 10 años en la enfermedad de etapa intermedia y otro medicado con avance de 22 años en la enfermedad (con temblores despreciables debido a la medicación) , quienes suministraron las muestras (una representación se puede ver en figura 4). Se registraron para cada individuo, tres series de tiempo de aceleraciones con

aproximadamente 2,27 minutos de duración cada una, empleando el software identificado en el la figura 2.



Figura 3: Registro de temblores en candidatos voluntario con EP, en el club de peronas con EP (Marzo 2020). Imagen de creación propia.

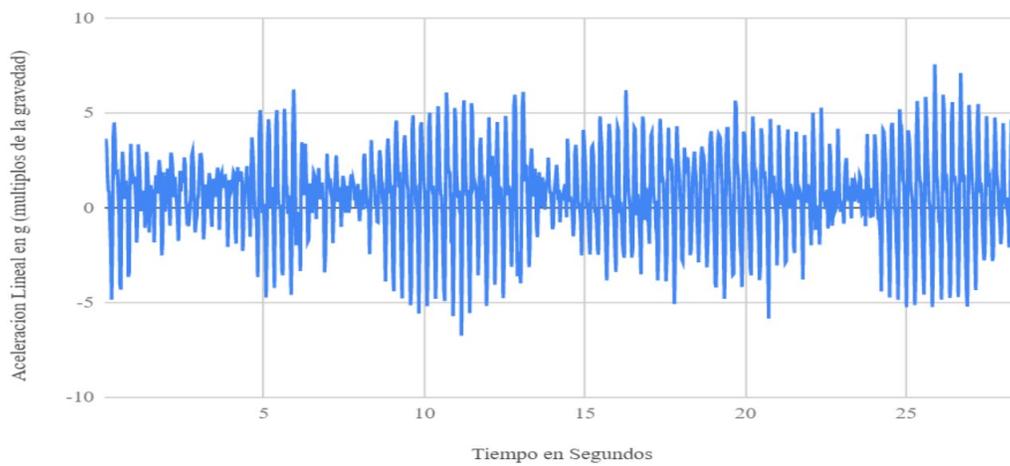


Figura 4: Segmento de 28 segundos perteneciente a la serie temporal eje X registrada en sujeto con EP sin medicación. Imagen generada a partir de registro de señales.

### 3.2 Modelo matemático

El problema planteado aquí se puede sintetizar en la figura 5. Se pretende identificar un modelo que represente la dinámica de los temblores en reposo a partir de las señales de aceleración registradas, con la complejidad adicional que significa no disponer de una señal forzadora de entrada a nuestro sistema dinámico real (la cual habrá que determinar y simular), que serviría para realizar la identificación. Por tal motivo se plantea una metodología ad-hoc que consiste en contruir una función parametrizada de entrada basada en la información frecuencial de la salida, y cuyos parámetros se optimizan (empleando un algoritmo de recocido simulado) en base a la capacidad del modelo para describir las salidas. Debido a que se pretende alcanzar algún grado de predicción de valores futuros inmediatos de las aceleraciones de los ejes X e Y (Ver figura 1), se recurre a la identificación mediante un modelo NARMAX que ha resultado ser una poderosa herramienta en sistemas complejos no lineales y caóticos.

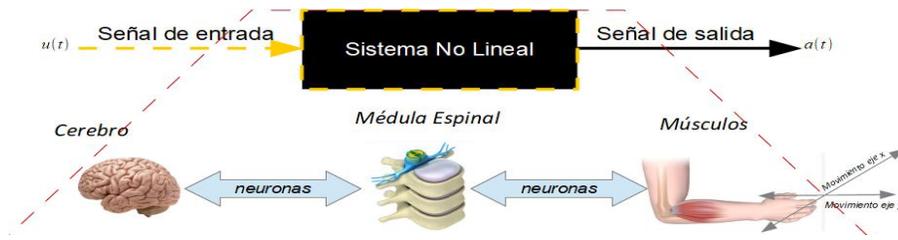


Figura 5: Representación del modelo de caja negra correspondiente al sistema generador de los temblores en reposo en la EP. **Imagen de creación propia.**

#### 3.2.1 Modelo NARMAX polinomial.

Un modelo no-lineal autoregresivo de media móvil con entradas exógenas (su acrónimo es NARMAX que deriva de las palabras en inglés; nonlinear autoregressive moving average model with exogenous inputs) para las aceleraciones está dado por la siguiente ecuación:

$$\begin{aligned}
 a(k) = f^l[ & a(k-1), a(k-2), \dots, a(k-n_a), \\
 & u(k-1), u(k-2), \dots, u(k-n_u), \\
 & e(k-1), e(k-2), \dots, e(k-n_e)] + e(k)
 \end{aligned} \tag{1}$$

Donde  $a(k)$ ,  $u(k)$  son valores de series temporales de la salida (aceleración) y la entrada (función forzadora) respectivamente en el instante  $k$ , muestreadas con un período de muestreo  $T_s = 1/200\text{Hz}$ . El término  $e(k)$ , representa la incertidumbre, posible ruido o parte no modelada de la dinámica. Además,  $n_a$ ,  $n_u$  y  $n_e$  son los retrasos máximos para la salida, entrada y

el error. El operador  $f^l$  representa una función no lineal que realiza el mapeo de las entradas del modelo (retrasos de  $a$ ,  $u$  y  $e$ ) a la salida prevista. En este desarrollo, la función no lineal se toma como un polinomio de grado máximo  $l$  [10]. El término de ruido se puede definir como un error de predicción según la siguiente ecuación:

$$e(k) = a(k) - \hat{a}(k|k-1) \quad (2)$$

Donde  $\hat{a}(k|k-1)$  denota la predicción de  $a(k)$  hecha por el modelo en el instante de tiempo anterior  $k-1$ .

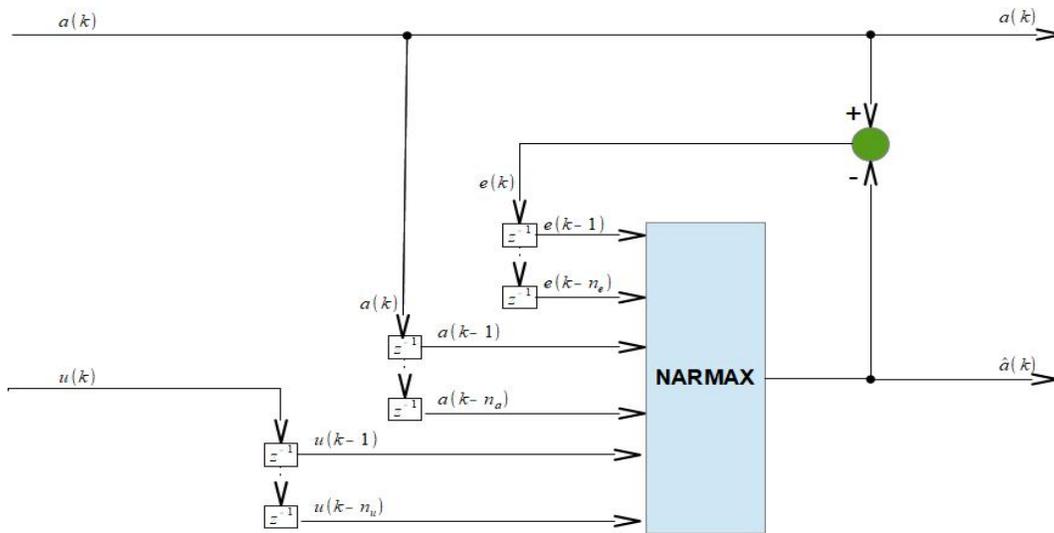


Figura 6: Esquema representativo del predictor asociado al modelo NARMAX, donde  $u$  representa la serie de tiempo de entrada (señal de forzamiento),  $a$  es la serie de tiempo de la aceleración autoregresiva y  $e$  el error de predicción.  $\hat{a}$  corresponde a la predicción de la aceleración. **Imagen de creación propia.**

La estructura del modelo (figura 6) plantea la necesidad de contar con valores retrasados del error de predicción  $e(k)$ , a partir de predicciones realizadas anteriormente. Por esta razón generalmente se construye la estructura mostrada en la figura 7. Donde un modelo inicial NARX (no-lineal autoregresivo con entradas exógenas) se emplea para generar predicciones y errores de predicción sobre un segmento de la serie de tiempo de aceleraciones muestreadas. La ecuación que describe un modelo NARX es similar a la de un modelo NARMAX solo que no involucra la media móvil representada por el error de predicción. Su ecuación es la siguiente:

$$a(k) = f^l[a(k-1), a(k-2), \dots, a(k-n_a), u(k-1), u(k-2), \dots, u(k-n_u)] \quad (3)$$

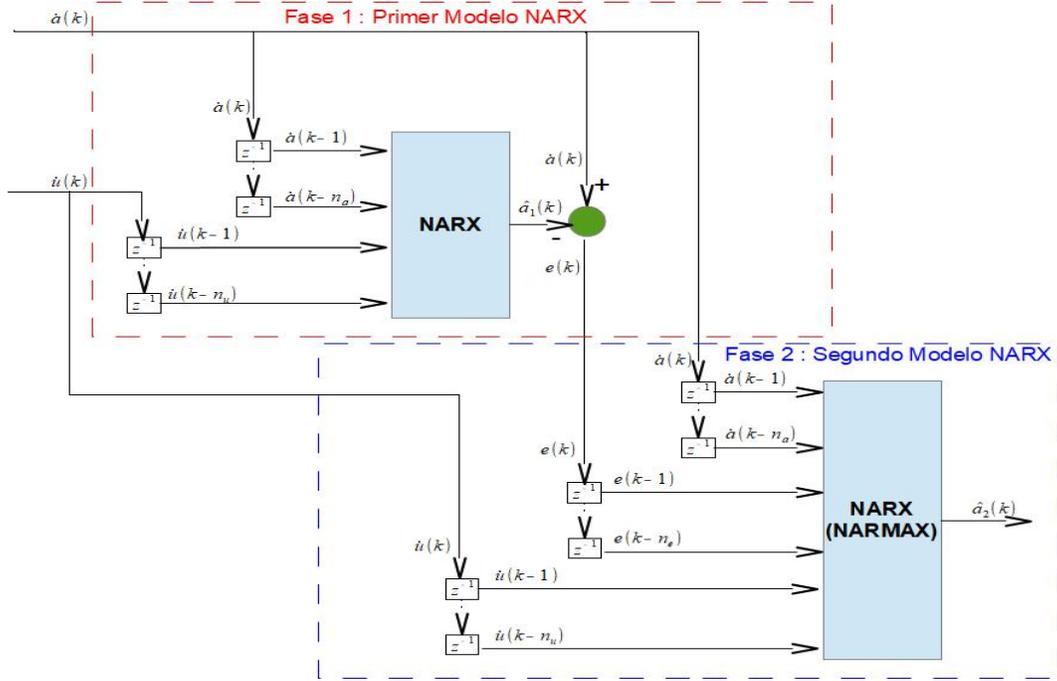


Figura 7: Esquema representativo del procedimiento para hallar el modelo NARMAX a partir de predicciones realizada con un modelo NARX. En el gráfico,  $\hat{a}$  representa un segmento acotado de las series de tiempo de aceleración muestreada y  $\hat{u}$  es el segmento de la señal de entrada, acotado en el mismo rango temporal.  $e$  es el error de predicción. **Imagen de creación propia.**

Aunque la ecuación (3) representa un modelo SISO (single input, single output), un modelo NARX puede ser MISO (multiple input, single output) como el mostrado en la fase 2 de la figura 7, solo que al ser el error de predicción precalculado la segunda entrada, se constituye en un modelo NARMAX. Con estos modelos se pueden realizar predicciones OSA (one step ahead) o MPO (Model Predicted Output) [11]. En el caso NARMAX con OSA, disponiendo de un número adecuado de valores previos de  $a(k)$ ,  $u(k)$  y  $e(k)$  se predice el próximo valor inmediato de la serie de tiempo  $\hat{a}(k)$  y se realiza iterativamente este proceso hasta el final de la serie temporal a modelar  $a(k)$ . Por otro lado, en NARMAX con MPO, la respuesta del modelo se inicializa mediante unos pocos valores conocidos de la serie de tiempo a modelar  $a(k)$  con  $e(k)$  dado y luego el MPO se calcula a partir del modelo identificado impulsado solo por la entrada dada  $u(k)$ . Siguiendo lo planteado en [12], el procedimiento para encontrar el modelo NARMAX para las aceleraciones se resume en los siguientes pasos:

1. Dividir el conjunto de datos a modelar, en dos partes; Datos de identificación y datos de predicción.

2. Definir  $n_a$ ,  $n_u$  y  $l$ .
3. Generar y validar la estructura NARX (Fase 1 figura 7) con los datos de identificación. Si no cumple el criterio de validación, regresar al punto 2.
4. Generar errores de predicción  $e(k)$  con NARX dentro del conjunto de datos de identificación.
5. Definir  $n_e$ , generar y validar estructura NARMAX con datos de identificación. Si no cumple el criterio de validación, regresar al punto 2.
6. Comprobar respuesta del modelo para el conjunto de datos de predicción.

### 3.2.2 Selección de términos y parámetros en la estructura NARMAX.

La construcción de los modelos NARMAX debe contener los términos seleccionados que resultan de la expansión polinomial de grado  $l$  de los regresores y los parámetros estimados relacionados con cada término. Esto se realiza empleando el algoritmo extendido de mínimos cuadrados ortogonales de regresión hacia adelante (sus siglas en inglés son FROLS). El algoritmo FROLS es la aplicación de descomposición ortogonal para seleccionar los términos relevantes en el modelo habilitando el cálculo de una métrica denominada tasa de reducción de errores (ERR). A continuación se resume el procedimiento planteado en [11].

Consideremos la ecuación 1 escrita ahora de la siguiente forma:

$$y(k) = \sum_{i=1}^M [\theta_i p_i(k)] + e(k) \quad (4)$$

Donde  $p_i(k)$  es cada término del modelo completo según sus órdenes y  $\theta_i$  el respectivo parámetro libre. Después de la formulación de la matriz seguido de ortogonalización, es posible reescribir la ecuación 4 como:

$$\mathbf{y} = W\mathbf{g} + e \quad (5)$$

Donde para  $N$  muestras tenemos:

$$\begin{aligned} \mathbf{y} &= [y(1) \dots y(N)]^T \\ \mathbf{e} &= [e(1) \dots e(N)]^T \end{aligned} \quad (6)$$

La matriz  $W$  tiene columnas ortogonales y  $\mathbf{g} = [g_1, g_2 \dots g_M]^T$  es un vector de parámetros auxiliares que puede calcularse de la siguiente forma:

$$g_i = \frac{\langle \mathbf{y}, \mathbf{w}_i \rangle}{\langle \mathbf{w}_i, \mathbf{w}_i \rangle}, \quad i = 1, 2, \dots, M. \quad (7)$$

Donde  $\mathbf{w}_i$  es la  $i$ -ésima columna de  $\mathbf{W}$ . El vector de parámetros  $\theta$  se relaciona con  $\mathbf{g}$  por medio de las matrices derivadas del proceso de descomposición ortogonal (ver [11] para más detalles). Mediante el uso de la ecuación 5 es posible calcular la ERR para cada término del modelo en 4. La ERR para cada término se calcula de la siguiente forma:

$$[ERR]_i = \frac{g_i^2 \langle \mathbf{w}_i, \mathbf{w}_i \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle}, \quad i = 1, 2, \dots, M. \quad (8)$$

El valor de ERR es una indicación de la importancia de cada regresor y se emplea para seleccionar los términos resultantes de la expansión polinómica, cuyo valor de ERR sea mayor. Se seleccionan un conjunto reducido de términos, los suficientes de tal forma que:

$$1 - \sum_{i=1}^M ERR_i < \rho \quad (9)$$

Donde  $\rho$  es una tolerancia establecida e inferior a la unidad.

### 3.2.3 Criterios para validación del modelo.

Los modelos identificados son validados con respecto a su capacidad de generalización y precisión de predicción. Para evaluar la precisión de la predicción, se utiliza el coeficiente de correlación múltiple de varianzas no explicadas definido como:

$$R^2 = 1 - \frac{\sum_{k=1}^N [a(k) - \hat{a}(k)]^2}{\sum_{k=1}^N [a(k) - \bar{a}(k)]^2} \quad (10)$$

Donde  $\hat{\mathbf{a}}(\mathbf{k})$  es la predicción OSA o MPO,  $\bar{\mathbf{a}}(\mathbf{k})$  denota el valor medio de la secuencia y  $N$  es el número total de muestras. La ventaja de usar  $R^2$  es que no depende de la amplitud respecto a la cantidad que se está midiendo, por lo que puede ser interpretado en diferentes

contextos. Otros métodos de validación mediante pruebas de correlación de entrada-salida se muestran en [11].

### 3.3 Caracterización de las muestras de aceleración $a_x$ y $a_y$ , tomadas de una persona con EP.

La comparación gráfica de los registros de aceleración para los dos participantes con enfermedad de parkinson (EP) permitió descartar en una etapa inicial, las muestras de temblores pertenecientes al individuo con medicación, ya que la efectividad del medicamento lograba atenuar los temblores hasta hacerlos prácticamente imperceptibles. Luego de un segundo exámen visual de las muestras pertenecientes al participante sin medicación, se ha seleccionado la serie de tiempo que mejor representa un movimiento involuntario de la mano del participante (temblor en reposo) muy evidente y sin interrupciones o sobreimpulsos debidos a otros movimientos no relacionados con la enfermedad. Según lo descrito en la sección 3.1.3, se presentan en la figura 8, las series temporales de aceleraciones para los ejes X e Y (en concordancia a lo especificado en la figura 1).

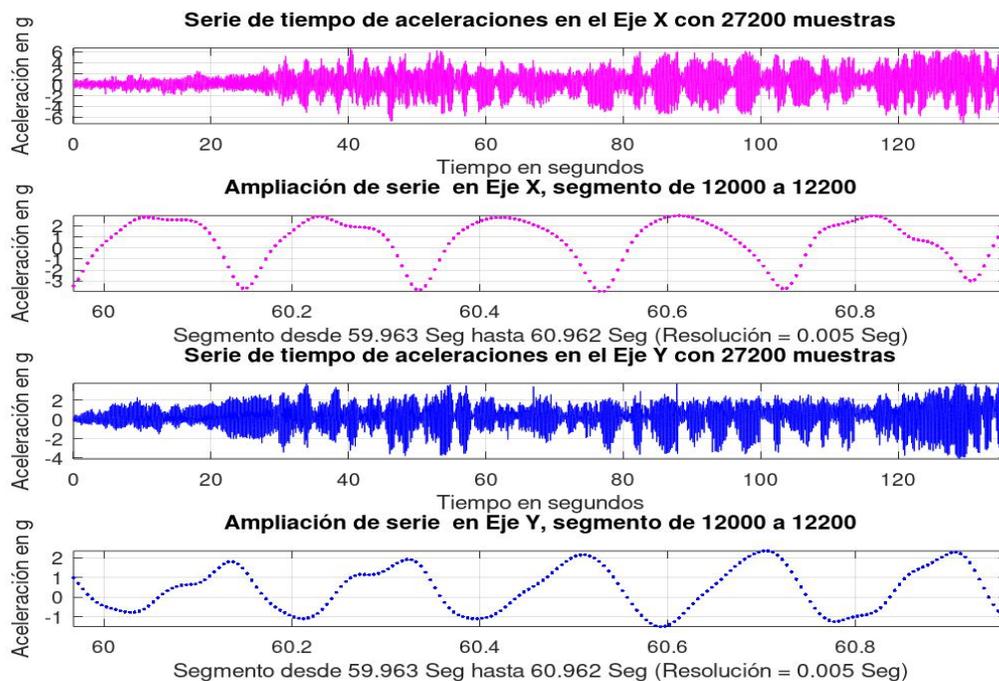


Figura 8: Series de tiempo para ejes X e Y de aceleraciones por temblores en reposo en paciente con EP. Los valores de aceleración están dados en multiples de la aceleración de la gravedad (g). **Imagen generada en base a los registros de aceleración muestreada.**

### 3.3.1 Autocorrelación de las series de tiempo $a_x$ y $a_y$ .

La autocorrelación se obtiene correlacionando cada variable respecto a ella misma escalada en 5 ms (5 milisegundos a una frecuencia de muestreo de 200Hz, ver código en Anexo B.1). Los resultados en la tabla 1, demuestran que existe causalidad de los valores futuros de las variables, respecto a los valores pasados de la misma. De hecho al ser estos valores de autocorrelación cercanos a la unidad ( $r=1$ ), permiten descartar la idea que las series temporales (registro de temblores en reposo de individuo con EP) sean el resultado de un proceso estocástico o de azar (ruido gaussiano sin autocorrelación). El gráfico de dispersión de la figura 9, confirma lo anteriormente dicho. Se aprecia como emerge un patrón de regresión lineal.

Coeficientes de Autocorrelación	
$r_{a_x}$	$= 0,96872$
$r_{a_y}$	$= 0,97216$

Tabla 1: Coeficientes de autocorrelación para las aceleraciones en direcciones X y Y del smathphone.

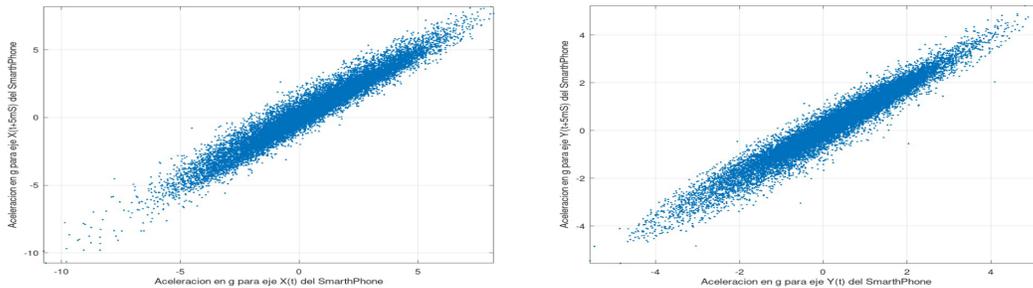


Figura 9: Gráfica de dispersión de autocorrelación para  $a_x$  (izquierda) y  $a_y$  (derecha). **Imagen generada con código en anexo B.1**

### 3.3.2 Análisis espectral de las series de tiempo $a_x$ y $a_y$ .

A continuación se profundiza en las características frecuenciales de las series de tiempo. Para ello se parte con el análisis de la densidad espectral de energía DEE (ver definición en anexo A.1) empleando el código dispuesto en el anexo B.2.

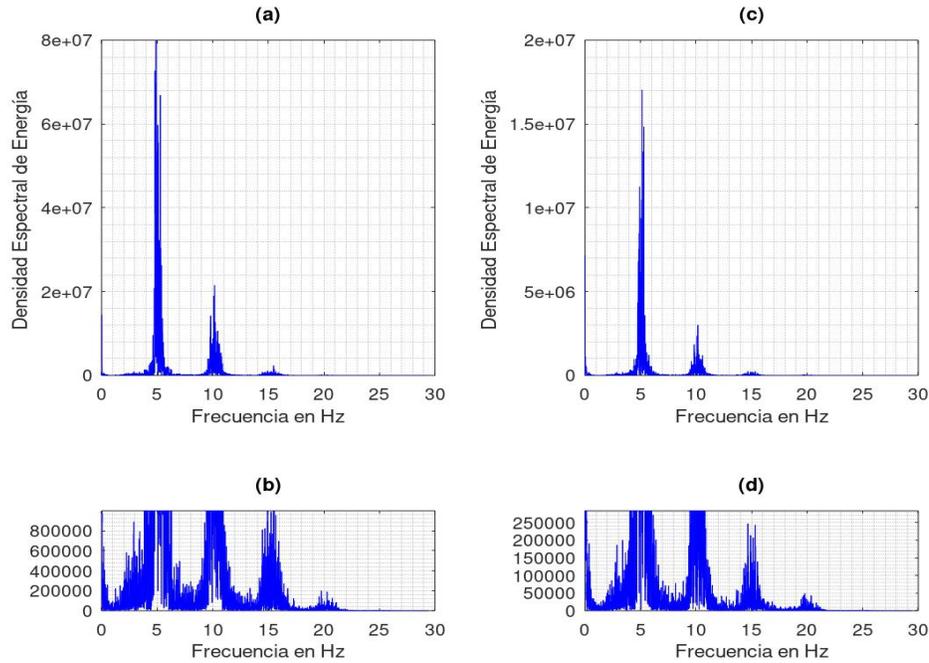


Figura 10: Densidad espectral de energía (DEE) en las series temporales  $a_x$  (a) y  $a_y$  (c). (b) y (d), son ampliaciones en las bases de las respectivas gráficas superiores . **Imagen generada con código en anexo B.2**

Luego de hallar la transformada de Fourier y constuir la figura 10, se pudo apreciar que ambas series temporales parecen tener cuatro componentes frecuenciales cuasi-armónicas similares (los cuales se presentan en la tabla 2) y muchos interarmónicos no atribuibles a ruido gaussiano ya que no se extienden en todo el ancho de banda.

Serie $a_x$	Serie $a_y$
f1= 4.9044Hz	f1= 5.1544Hz
f2=10.1765Hz	f2=10.1765Hz
f3=15.4853Hz	f3=15.3456Hz
f4=19.7941Hz	f4=19.7941Hz

Tabla 2: Frecuencias principales de las series de tiempo (señales)  $a_x$  y  $a_y$ .

Con el análisis previo pudimos ver las frecuencias de las señales que componen cada una de las series temporales de aceleraciones, pero esa descomposición espectral no es útil para apreciar como se distribuye la energía a lo largo de los 136 segundos de muestra. Para ello se proponen los espectrogramas 3D de las figuras 11 y 12 (generados con código anexo B.2), los cuales han sido construidos empleando la **transformada de Gabor discreta** con ventana gaussiana (ver definición en anexo A.2).

Las gráficas 3D muestran con más detalle la distribución en tiempo y frecuencia de las componentes. Se aprecia que aunque las componentes frecuenciales más importantes (tabla 2) son muy marcadas, existen frecuencias que deberán ser tomadas en cuenta durante la identificación de la dinámica para lograr la mayor similitud entre la señal real y la reconstrucción por predicciones.

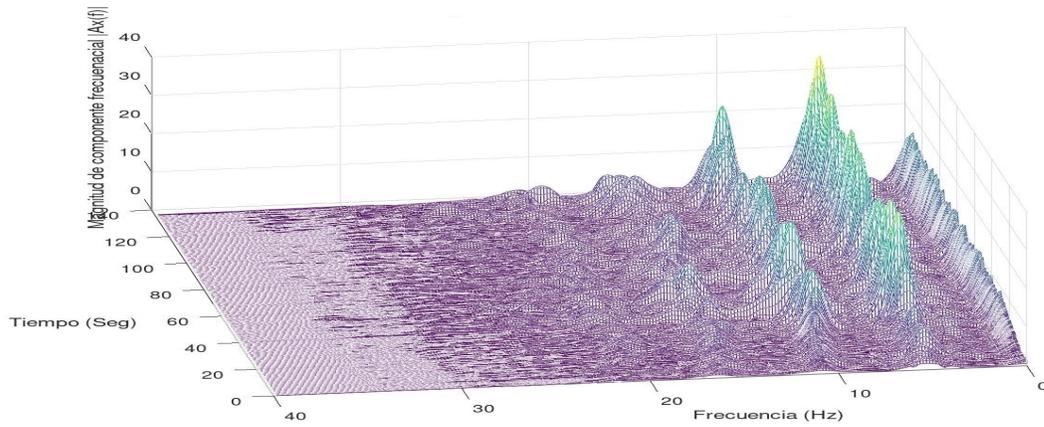


Figura 11: Espectrograma 3D para  $a_x$ . Imagen generada con código en anexo B.2

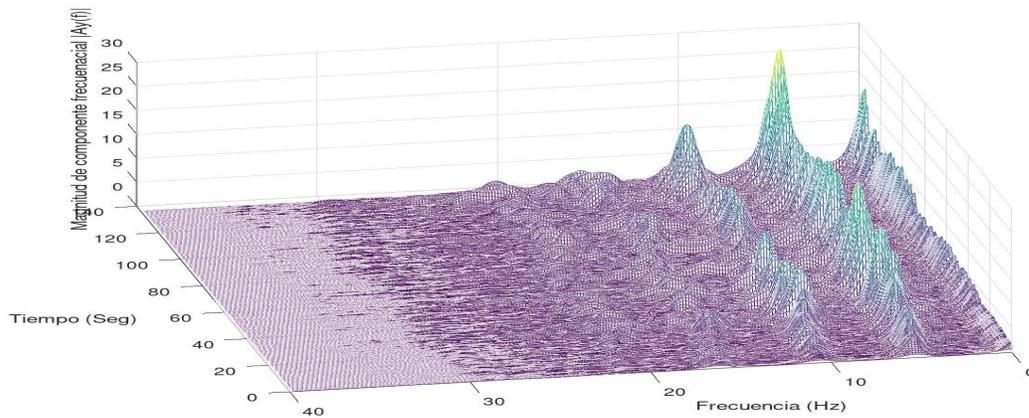


Figura 12: Espectrograma 3D para  $a_y$ . Imagen generada con código en anexo B.2

### 3.3.3 Invariantes cualitativos de las series de tiempo $a_x$ y $a_y$ .

Los invariantes dinámicos son conjuntos del espacio de fase que se mapean en sí mismos cuando cada uno de sus elementos es mapeado adelante en el tiempo por la dinámica [11]. Para comprender la metodología aplicada en la búsqueda de los invariantes se plantean las siguientes definiciones previas:

**Tiempo de retardo:** Es la diferencia temporal representada en número de muestras  $\tau$  (o unidades de tiempo,  $\tau\Delta t$ ) entre componentes adyacentes del vector de estado que puede extraerse de la serie de tiempo [13].

**Dimensión de inmersión ( $D_e$ ):** Existe una dimensión  $m$  tal que los vectores de estados reconstruidos por tiempo de retardo (de datos límpios de ruido) son iguales a los vectores del espacio de fase de la dinámica. Esto se sustenta en el **teorema de Takens** ( ver anexo A.3).

**Vectores de estado falsos vecinos:** Dos vectores de estado  $V_t$  y  $V_t^{NN}$  son falsos vecinos si al realizar la inmersión de la serie de tiempo  $a_t$ , la distancia entre ambos vectores se incrementa al pasar de  $R^n$  a  $R^{n+1}$  [14].

Para lograr una reconstrucción adecuada del espacio de fases (o espacio de estados) con soporte en el **teorema de Takens** para cada serie de tiempo  $a_x$  y  $a_y$ , que permita visualizar el atractor de la dinámica, necesitamos conocer tanto el tiempo de retardo como la dimensión de inmersión de los vectores de estado que definen la evolución del sistema dinámico subyacente. Según lo argumentado en [14, pag. 10], una buena aproximación al tiempo de retardo es tomar una cuarta parte del período aproximado de la señal. De esta forma, en base a la tabla de frecuencias (Tabla 2), se construye la siguiente tabla empleando la frecuencia menor (de cada serie de tiempo) que es sin duda la que aporta mayor potencia:

Señal	Tiempo de retardo ( $\frac{1}{4f_1}$ )	$\tau$ (Entero más cercano de $\frac{200Hz}{4f_1}$ )
$a_x$	0.050975	10
$a_y$	0.048502	10

Tabla 3: Tiempos de retardo (time lag) en la inmersión de vectores de estado subyacentes en las serie de tiempo (señales)  $a_x$  y  $a_y$ .  $\tau$  es igual a 10 puntos muestrales con un período de muestreo de 0.005 Seg

Conociendo el tiempo de retardo  $\tau$ , se emplea ahora un algoritmo para identificación de falsos vecinos implementado en el paquete TISEAN desarrollado por [13], portado a Octave. El algoritmo consiste en agregar dimensiones al espacio de estado (agregar variables al vector de estado) de tal forma que así, vectores que pueden parecer cercanos en una baja dimensión

pueden no serlo en dimensiones superiores. Cuando la distancia entre los vectores deja de crecer sobre un radio  $r$  fijo, se ha alcanzado la dimensión de inmersión.

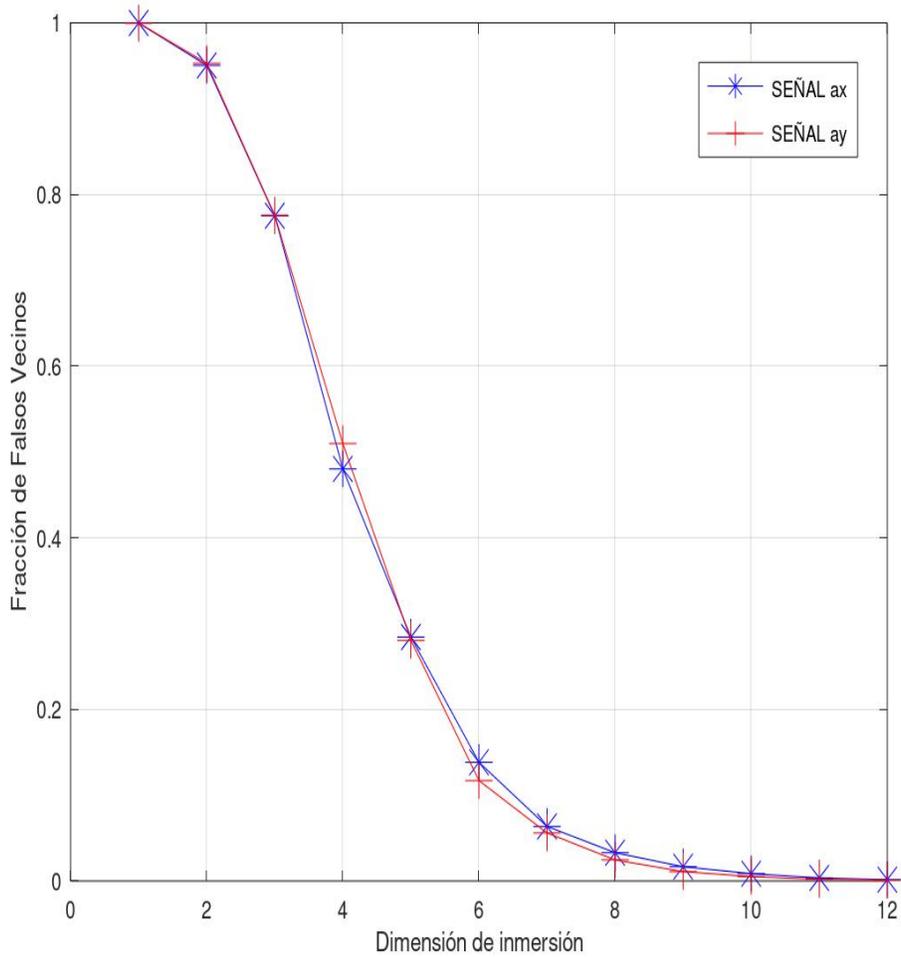


Figura 13: Relación de falsos vecinos para un retraso temporal de 10 puntos de las señales  $a_x$  (azul) y  $a_y$  (rojo). **Imagen generada con código en anexo B.3.**

En la figura 13 se aprecia que para una dimensión de inmersión ( $D_e$ ) con 11 variables de estado, los falsos vecinos se hacen nulos, por tanto esta será la base para el posterior análisis de los invariantes cualitativos de la dinámica. En la siguiente gráfica se aprecian los atractores en 2 y 3 dimensiones para cada serie temporal, construidos con retrasos de 0.3seg para 2D y en el caso 3D, retrasos de 0.15seg y 0.3seg (empleando 2200 puntos muestrales centrales de las señales). La representación se basa en  $\tau$  y  $D_e$  ya hallados.

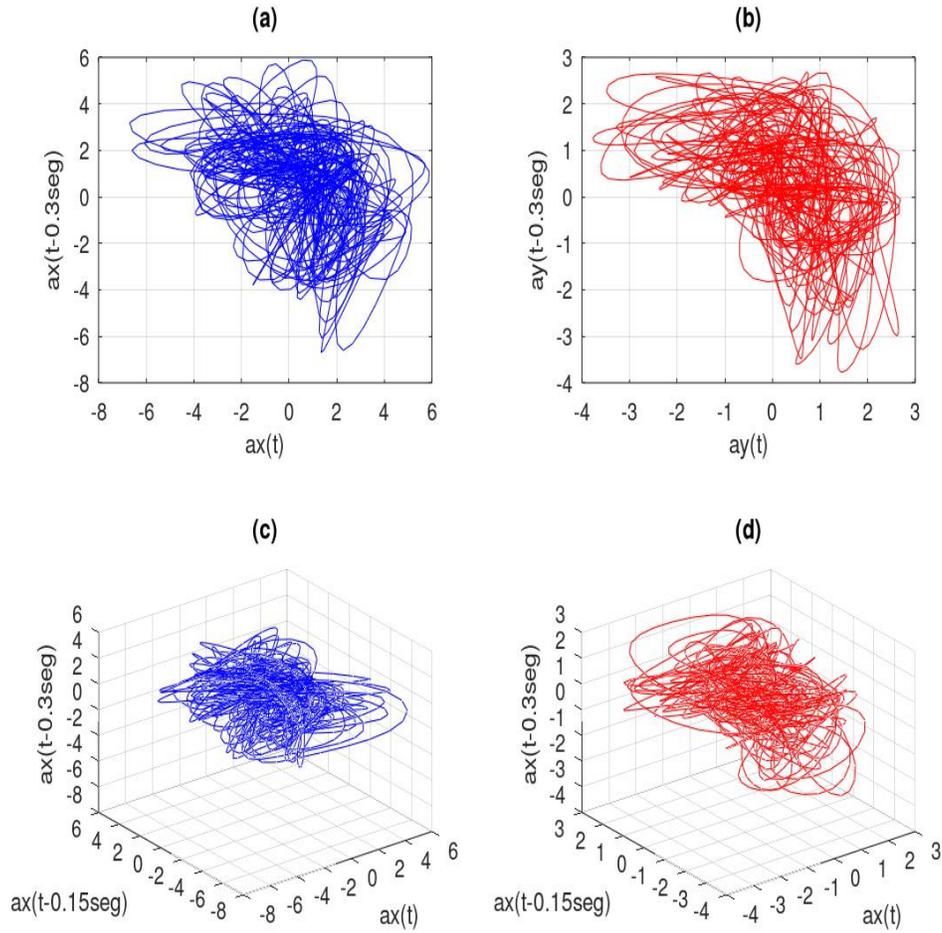


Figura 14: Atractores extraños 2D y 3D en espacio de fases para las señales  $a_x$  (a y c) y  $a_y$  (b y d). **Código en anexo B.3.**

Uno de los invariantes dinámicos más importantes es el **máximo exponente de Lyapunov** de las series de tiempo (ver anexo A.4). Conociendo el valor del máximo exponente de Lyapunov ( $\lambda$ ), podemos considerar si la serie subyacentemente responde a una dinámica de ciclo límite ( $\lambda = 0$ ) o caos determinista ( $0 < \lambda < \infty$ ). La figura 15, representa la divergencia de las trayectorias en el espacio de fases con  $D_e=11$ . La pendiente de la recta en color negro (conseguida por ajuste de mínimos cuadrados), representa el máximo  $\lambda$  de cada serie de tiempo. En la tabla 4, un resumen de los parámetros de la dinámica.

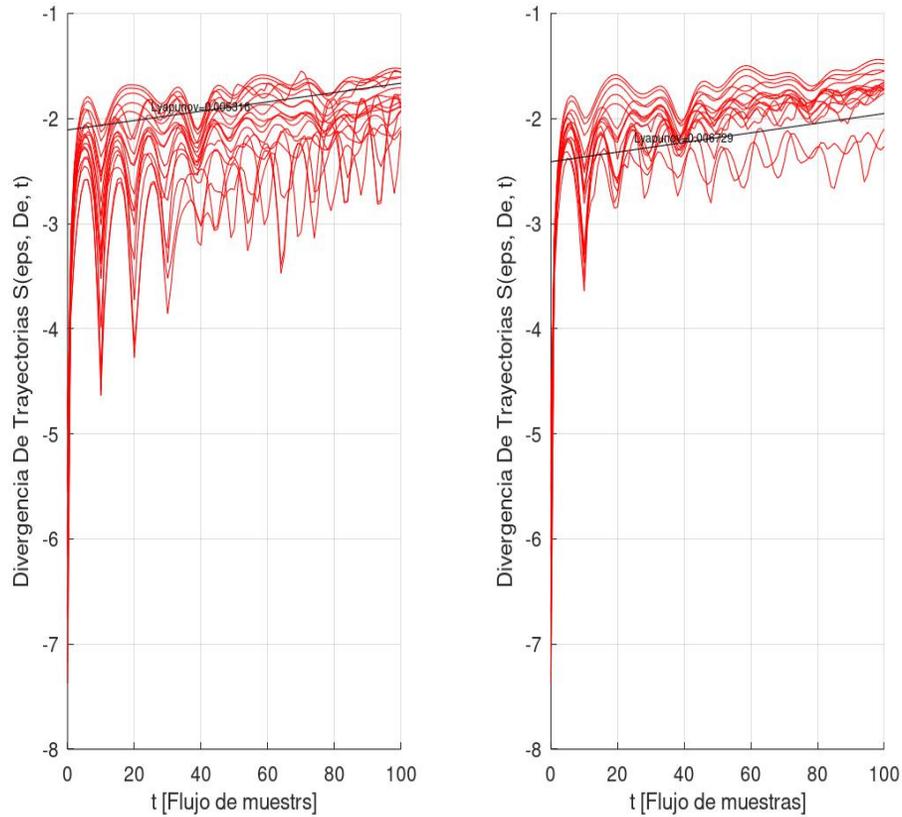


Figura 15: Máximos exponentes de Lyapunov para las señales  $a_x$  (izquierda) y  $a_y$  (derecha). La pendiente de la recta (color negro) representa  $\lambda$ . **Imagen generada con código en anexo B.3.**

Señal	$\tau$	$D_e$	$\lambda(seg^{-1})$
$a_x$	10	11	0.004435
$a_y$	10	11	0.004585

Tabla 4: Parámetros de la dinámica subyacente  $a_x$  y  $a_y$ .

Se puede decir en este punto, que los temblores registrados para esta investigación, responden a una dinámica caótica de baja dimensión debido a los pequeños valores de  $\lambda$  con la interesante particularidad que la serie de tiempo de las aceleraciones  $a_y$  parece ser un poco más caótica que la serie de tiempo  $a_x$ .

## 4 DISCUSIÓN DE RESULTADOS EN LA IDENTIFICACIÓN DE LA DINÁMICA

Las simulaciones se han realizado en dos etapas. En una primera etapa, orientada a encontrar la mejor señal de entrada (ver figura 5) con la información frecuencial encontrada en el apartado 3.3, se ejecuta el algoritmo de recocido simulado simultáneamente con el algoritmo de identificación del modelo NARMAX y las predicciones MPO para segmentos centrales de las señales que van desde el punto muestral 8335 al punto 10534, hasta obtener las mejores predicciones. En una segunda etapa, se generan las predicciones MPO sobre cada segmento completo, mediante ventanas temporales que involucran un modelado, validación y predicción, hasta describir la totalidad de puntos temporales de cada segmento. La predicción por ventana se ha hecho necesaria ya que, las realizadas a partir de un modelo obtenido con un conjunto de datos, pierde validez rápidamente y además para efectos de la simulación, aunque se cuenta con la totalidad de los datos, se asume que en un muestreo y procesamiento en tiempo real, los datos se van registrando novedosamente (se está creando información) en cada instante de tiempo, por lo que solo se cuentan con un conjunto de valores pasados. Para identificar hasta cual tamaño de ventana temporal es aceptable una predicción, se ha realizado el procedimiento con predicciones desde 5 milisegundos (un punto muestral) hasta 50 milisegundos (diez puntos muestrales) adelante.

La primera fase que consiste en hallar una señal de entrada, se puede separar en los siguientes pasos (ver códigos en anexo B.4):

1. Se define la función de entrada:

$$u(k) = C_1 \cos(2\pi f_1 t_k) + C_2 \cos(2\pi f_2 t_k) + C_3 \cos(2\pi f_3 t_k) + C_4 \cos(2\pi f_4 t_k) \quad (11)$$

De donde se extrae el vector de parámetros  $P = [C_1, C_2, C_3, C_4, f_1, f_2, f_3, f_4]$  inicializado con  $C_1 = C_2 = C_3 = C_4 = 1$  y  $f_1, \dots, f_4$  son las correspondientes frecuencias de la tabla 2 según sea la serie  $a_x$  o  $a_y$  la que se modela.  $t_k$  es el instante de tiempo  $k$ -ésimo del segmento seleccionado para la identificación.

2. Se define la estructura NARMAX con los correspondiente valores de  $n_a, n_u, n_e$  y  $l$  (ver ecuación 1) siguiendo el esquema de la figura 7.
3. Se define una función objetivo a minimizar, que entrega como resultado el valor  $|1 - R^2|$ . Esta ecuación es una modificación conveniente de la ecuación 10 al emplear el algoritmo de recocido simulado desarrollado por [15] y las herramientas de [16]. El valor  $R^2$  se obtiene luego de estimar el modelo (para los 2200 puntos del segmento) siguiendo lo planteado en los puntos 3.2.1 y 3.2.2 de este trabajo, y una posterior validación del modelo NARMAX para los 2200 puntos que conforman cada segmento de señal ( $a_x$  o  $a_y$ ).
4. Se ejecuta la optimización metaheurística con recocido simulado, que hace llamadas recurrentes a la función definida en el punto 3 hasta encontrar el mejor óptimo global en un tiempo finito.

Luego de ejecutar este procedimiento se encontraron los valores de los parámetros mostrados en la siguiente tabla:

Parámetro	Segmento $a_x$	Segmento $a_y$
$C_1$	0.9999031	133.2227504
$C_2$	1.0000960	128.9560338
$C_3$	1.0000960	45.8904865
$C_4$	0.9999031	-166.8186078
$f_1$	4.9044116	5.3958255
$f_2$	10.1764709	10.1645336
$f_3$	15.4852888	15.4371995
$f_4$	20.3014689	20.7722364

Tabla 5: Parámetros resultantes para la ecuación 11 con la cual se generan las entradas de los modelos NARMAX para  $a_x$  y  $a_y$ .

La segunda fase consiste de tomar un conjunto 300 puntos de  $u(k)$  y  $a(k)$  los cuales se emplean para encontrar el modelo y se realizan predicciones para los próximos puntos, lo que representa una ventana de predicción variable desde 5 milisegundos a 50 milisegundos con saltos de 5 milisegundos empleando un modelo NARMAX específico para cada ventana (o número de puntos) de predicción, tomando en cuenta que la frecuencia de muestreo ha sido de 200 Hz. El proceso se puede separar en los siguientes pasos (ver códigos en anexo B.5):

1. Se generan los valores de  $u(k)$  para cada  $t_k$  dentro de cada segmento de aceleraciones ( $a_x$  o  $a_y$ ) según corresponda.
2. Se define la estructura NARMAX con los correspondiente valores de  $n_a, n_u, n_e$  y  $l$  (ver tabla 6).
3. Se hallan los modelos NARMAX empleando los primeros 300 puntos tanto de la señal de entrada  $u(k)$  como la aceleración  $a(k)$ .
4. Se realizan las predicciones MPO para los puntos siguientes (desde 1 a 10 puntos muestrales según la ventana) de forma incremental y separada, lo que equivale a desarrollar predicciones desde 5ms hasta 50 milisegundos.
5. Se registran los valores predichos y la ventana conjunta de Modelado+Predicción se traslada ahora  $\mathbf{n}$  puntos adelante (donde  $\mathbf{n}=1,2,3,\dots,10$ ).
6. Se retorna iterativamente al item 3, a medida que se registran puntos nuevos de aceleración real (para efectos de esta simulación se asume que los puntos en los segmentos  $a_x$  y  $a_y$  van apareciendo consecutivamente en cada  $t_k$  como ocurriría en un muestreo en tiempo real).

En el siguiente algoritmo se esquematiza de forma genérica el procedimiento planteado anteriormente.

---

**Algoritmo 1** : Método para predicción por ventana móvil

---

**Entrada:**

- Conjunto de valores para señal de entrada simulada  $\mathbf{U} = \{u(1), u(2), u(3), \dots, u(k)\}$ .
- Conjunto de valores registrados de señal a modelar  $\mathbf{A} = \{a(1), a(2), a(3), \dots, a(k)\}$ .
- Número de puntos a predecir  $\mathbf{n}$ .
- Parámetros del modelo  $n_a, n_u, n_e$  y  $l$ .

**Salida:**

- Valores predichos  $\mathbf{np}$ .

$$i = 0, j = 1$$

- 1: **mientras** la cardinalidad  $\text{card}(\mathbf{A})=300$  **hacer**
  - 2:   **modelo**  $\leftarrow$  NARMAX( $n_a, n_u, n_e, l, \mathbf{U}, \mathbf{A}$ ) para 300 puntos previos ( $1 + \mathbf{n} * i \leq k \leq 300 + \mathbf{n} * i$ ).
  - 3:   **np**  $\leftarrow$  Predicción de  $\mathbf{n}$  puntos futuros de  $\mathbf{A}$  usando el **modelo**.
  - 4:   **devolver np**
  - 5:   Registrar  $\mathbf{n}$  puntos nuevos de la señal real de tal forma que  $\mathbf{A} = \{a(k) \mid 1 + \mathbf{n} * j \leq k \leq 300 + \mathbf{n} * j\}$ .
  - 6:   Incrementar  $i, j$
  - 7: **fin mientras**
- 

Parámetro	MPO $a_x$	MPO $a_y$
$n_a$	2	2
$n_u$	6	6
$n_e$	2	2
$l$	2	2

Tabla 6: Mejores valores de parametros encontrados para definir la estructura NARMAX de  $a_x$  y  $a_y$ .

Los valores tanto de los máximos regresores  $n_a, n_u$  y  $n_e$  como el orden del polinomio  $l$  que aparecen en la tabla 6, son los que mejores resultados han dado y emergen de una búsqueda empírica ( lo cual es lo usual en esta metodología). En los anexos A.5 y A.6, se muestran los modelos polinomiales resultantes para los primeros 300 puntos de la primera ventana de 50ms en cada segmento .

#### 4.1 Resultados de predicción por ventanas para segmento $a_x$

La figura 16 muestra la señal resultante de evaluar la ecuación 11, empleando los parámetros correspondientes al segmento  $a_x$  (2200 puntos muestrales) en el mismo rango temporal. Esta señal se ha conseguido con una correlación (en base a predicción MPO completa del segmento) de 0.0729, que aunque es pequeña (es la mejor correlación alcanzada con el algoritmo de recocido simulado), es lo suficientemente buena para hallar un modelo por ventana basado en 300 puntos como se describió anteriormente.

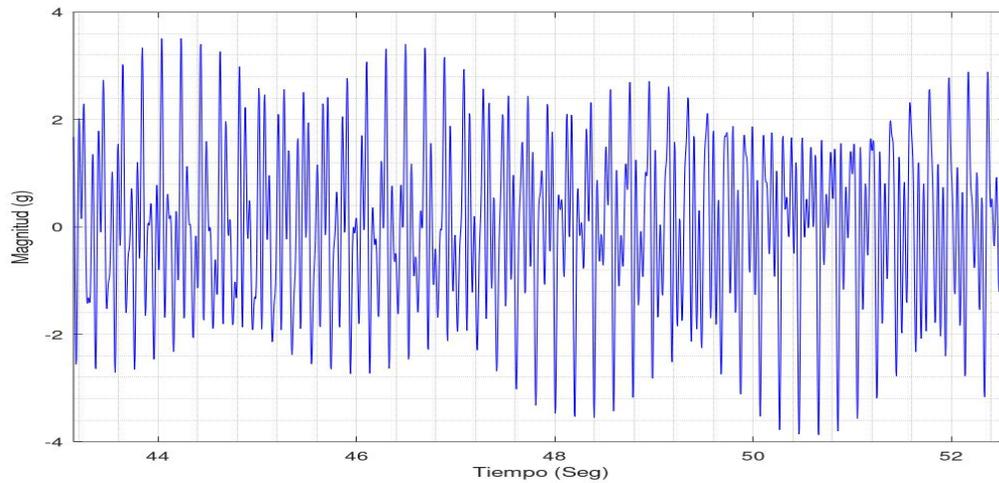


Figura 16: Segmento de señal de entrada  $u(k)$  para modelo NARMAX de  $a_x(k)$ , hallada con la ecuación 11. **Imagen generada con código en anexo B.6.**

La figura 17 muestra el resultado de las predicciones MPO para una ventana de 5 milisegundos (un punto muestral adelante) y modelado de 300 puntos. Se aprecia como la línea roja que representa la señal reconstruida por predicciones, sigue muy bien a la señal real representada por los puntos azules.

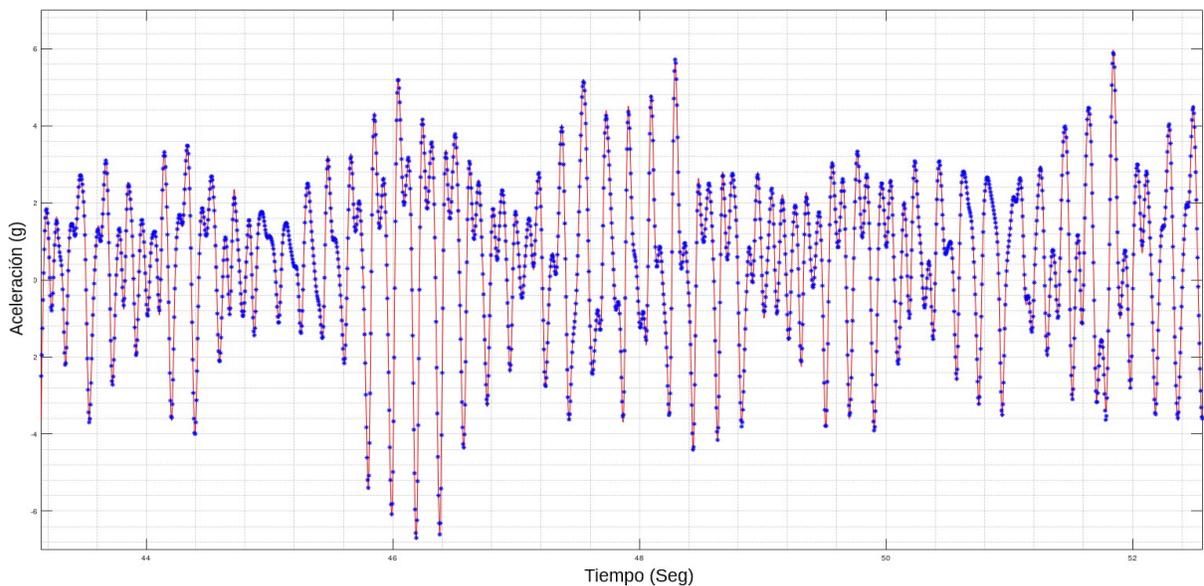


Figura 17: Reconstrucción por predicciones de 5ms adelante para el segmento de señal  $a_x$ . En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

La figura 18 representa la predicción por ventana de mayor tamaño (50ms). A simple vista se aprecia como ha empeorado las predicciones a tal punto que se pierde similitud entre las señales.

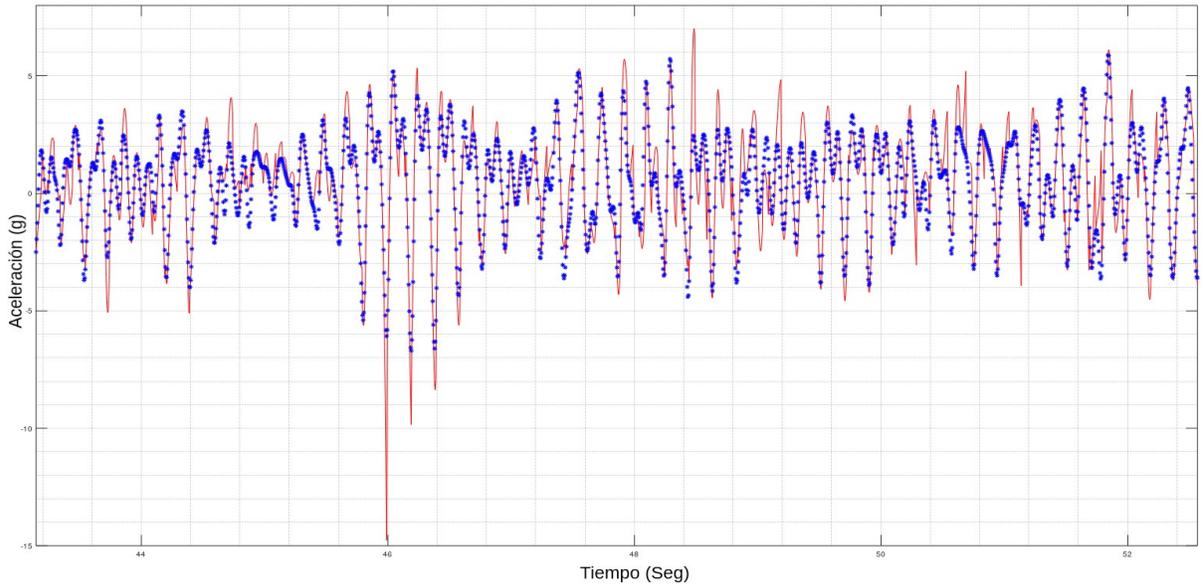


Figura 18: Reconstrucción por predicciones de 50ms adelante para el segmento de señal  $a_x$ . En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

En la siguiente figura se analiza el desempeño de cada estrategia de predicción. Se puede ver como crece rápidamente el error cuadrático medio (ECM) según el período de predicción de cada ventana, notándose un cambio de pendiente importante a los 20 milisegundos. De forma similar se degenera el porcentaje de correlación múltiple ( $R^2$ ), para el cual a los 20 milisegundos se consigue la mayor ventana temporal con una correlación que cumple el criterio  $R^2 \geq 95\%$  (el 100% corresponde a la autocorrelación sin desfase temporal de la señal real). En la tabla 7 se muestran los valores exactos que generan la figura 19.

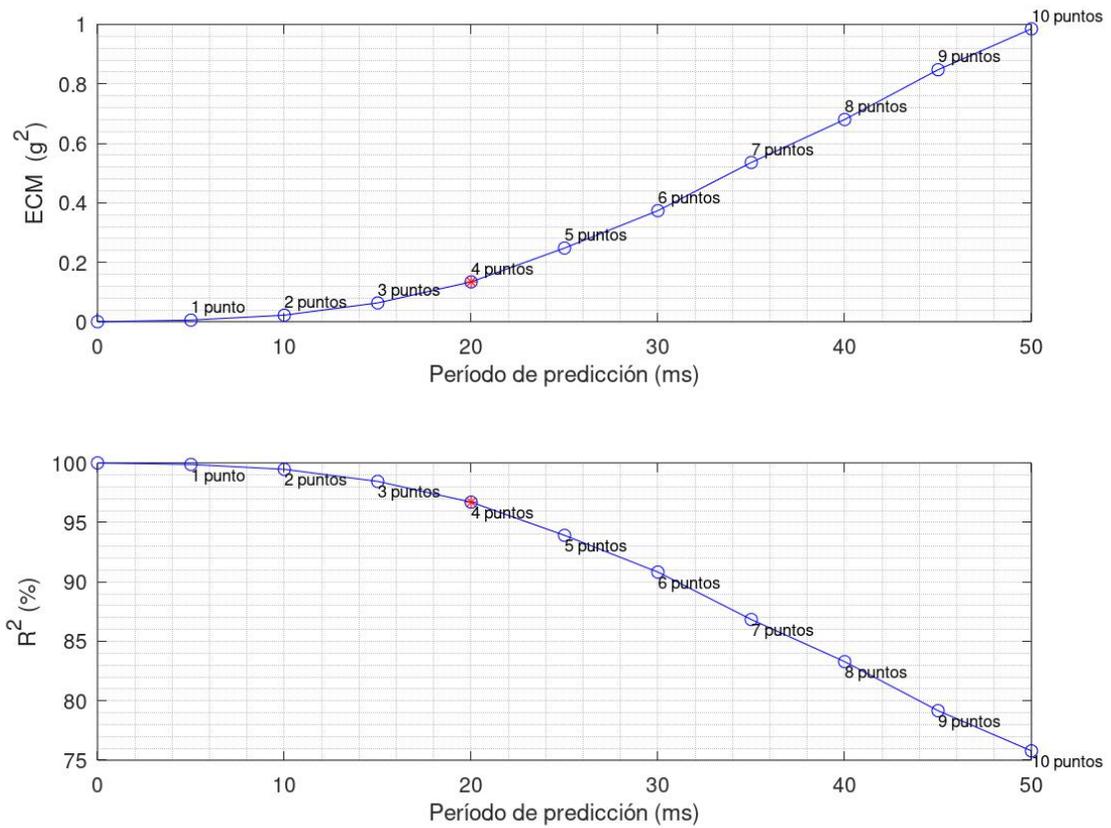


Figura 19: Proyección del error cuadrático medio EMC (Superior) y el coeficiente de correlación múltiple (inferior) para ventanas de predicción del segmento  $a_x$ . **Imagen generada con código en anexo B.6.**

Puntos Adelante	Período de predicción	ECM	$R^2$
1	5ms	5.3133e-03	99,87
2	10ms	2.1916e-02	99,46
3	15ms	6.3246e-02	98,45
4	20ms	1.3412e-01	96,70
5	25ms	2.4789e-01	93,91
6	30ms	3.7354e-01	90,82
7	35ms	5.3583e-01	86,83
8	40ms	6.8024e-01	83,28
9	45ms	8.4762e-01	79,17
10	50ms	9.8527e-01	75,79

Tabla 7: Valores generadores de la figura 19

Con una predicción de 20 milisegundos (o 4 puntos adelante) y 96,7% de correlación, se consigue el resultado de la figura 20, donde se aprecia que la señal reconstruida por predicciones (en rojo) solo difiere de la real en algunos puntos extremos, donde se forma un cambio brusco de la pendiente y sin embargo dista mucho en precisión de lo observado en la figura 18.

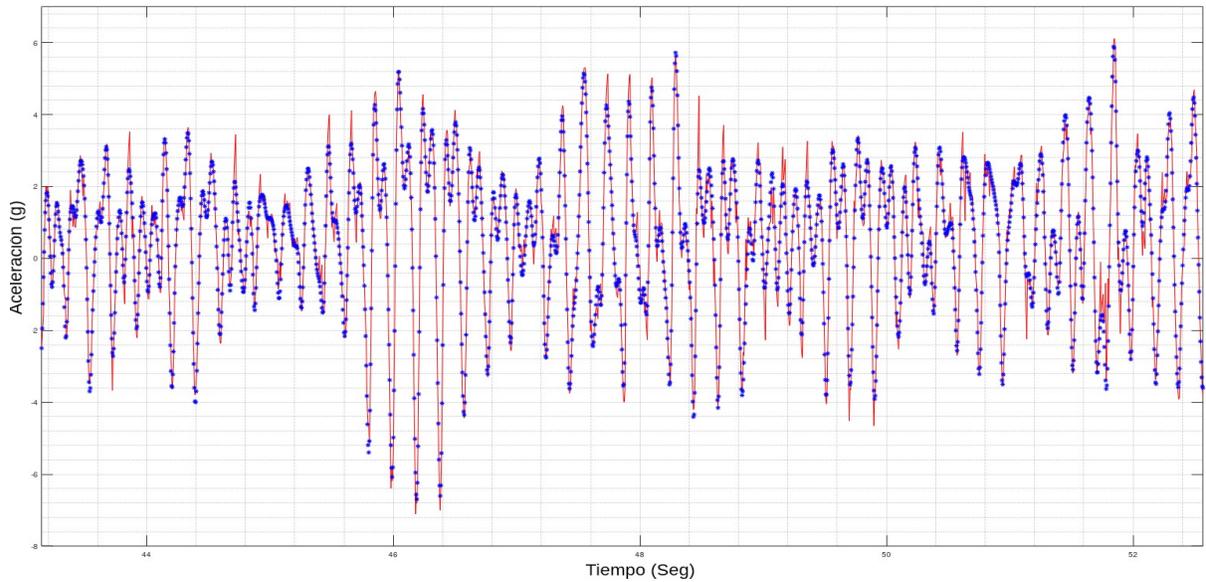


Figura 20: Reconstrucción por predicciones de 20ms adelante para el segmento de señal  $a_x$ . En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

## 4.2 Resultados de predicción por ventanas para segmento $a_y$

La figura 21 muestra la señal de entrada  $u(k)$  que resulta de evaluar la ecuación 11 con los parámetros hallados por correlación de la predicción MPO del segmento  $a_y$  (2200 puntos muestrales) con el algoritmo de recocido simulado. En este caso se ha conseguido un coeficiente de correlación de 0.4687 que resulta mayor al logrado para el segmento de eje  $a_x$ . Esto eventualmente tiene repercusiones en la precisión de las predicciones por ventanas. Sin embargo para lograr tal correlación, el algoritmo ha asignado magnitudes mucho más grandes (comparar valores de figuras 21 y 16).

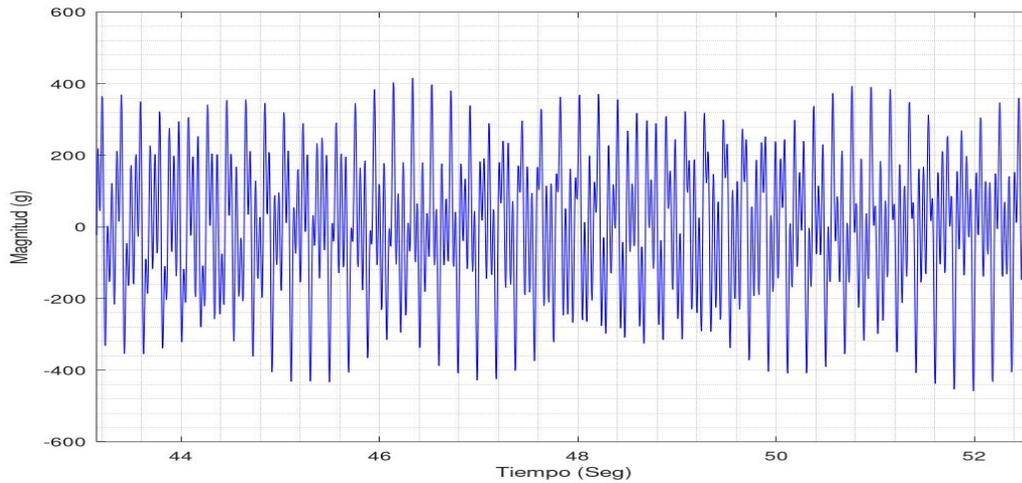


Figura 21: Segmento de señal de entrada  $u(k)$  para modelo NARMAX de  $a_y(k)$ , hallada con la ecuación 11. **Imagen generada con código en anexo B.6.**

De forma similar a lo ocurrido con el segmento  $a_x$ , la predicción de un punto adelante para  $a_y$  (5ms) resulta indistinguible de la señal real como se aprecia en la siguiente figura. Comparando las tablas 7 y 8 se verifica que el ECM del segmento  $a_y$  es aproximadamente 1/3 del ECM para la predicción de 5 milisegundos de  $a_x$ .

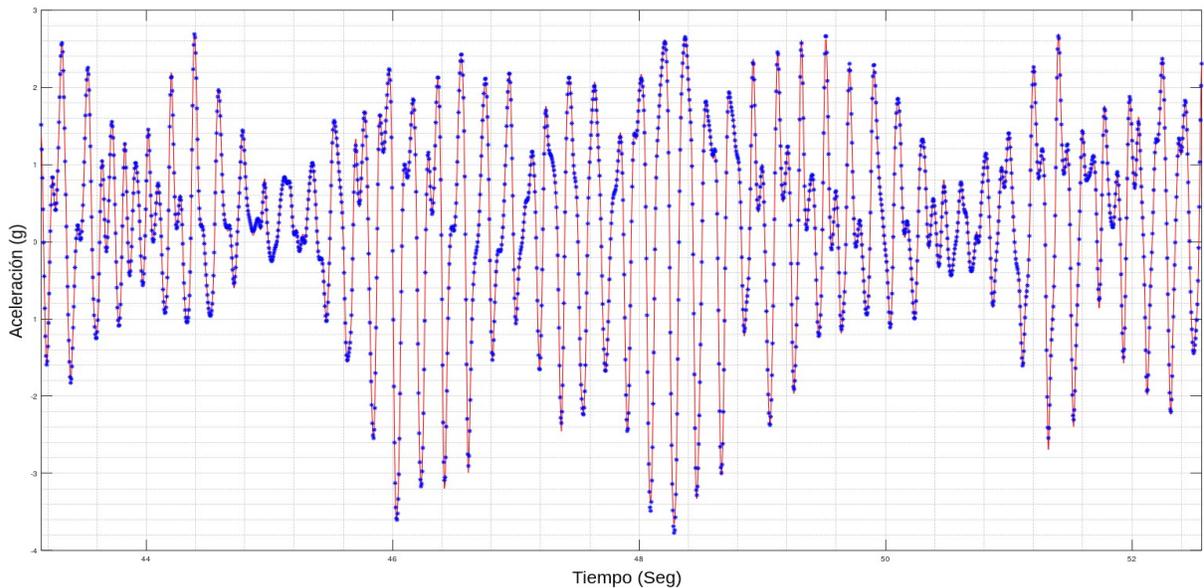


Figura 22: Reconstrucción por predicciones de 5ms adelante para el segmento de señal  $a_y$ . En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

En el caso de las predicciones con ventana de 50 milisegundos (10 punto muestrales adelante), aunque se mantiene una relación de 1/3 en el ECM entre ambos ejes coordenados de aceleraciones, la predicción no es lo suficientemente buena como para afirmar similitud con un porcentaje de correlación de 77,9% según la tabla 8.

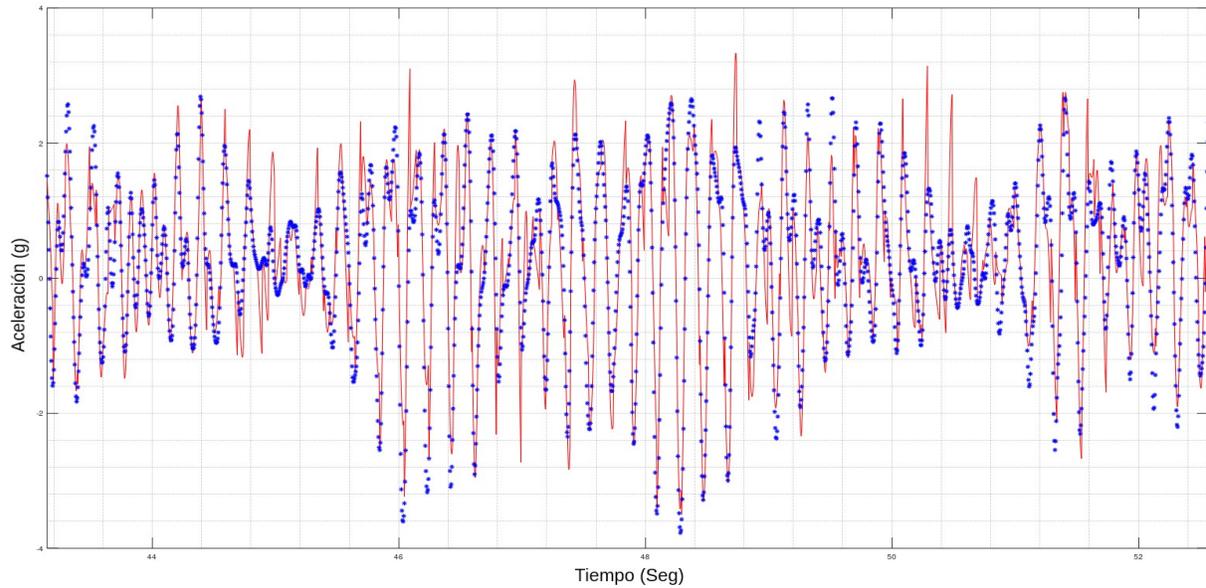


Figura 23: Reconstrucción por predicciones de 50ms adelante para el segmento de señal  $a_y$ . En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

La comparación de las tablas 7 y 8 no muestran grandes contrastes respecto a los porcentajes de correlación y se mantiene una predicción de ventana de 20 milisegundos (4 puntos muestrales adelante) como la de mayor amplitud temporal que cumple con el criterio de correlación múltiple  $R^2 \geq 95\%$ . La gráfica 24 muestra como evolucionan el ECM y el porcentaje de correlación múltiple con cada rango de predicción. La rápida divergencia del ECM para las señales estudiadas puede atribuirse a las características caóticas de baja dimensión encontradas en la sección 3.3.3.

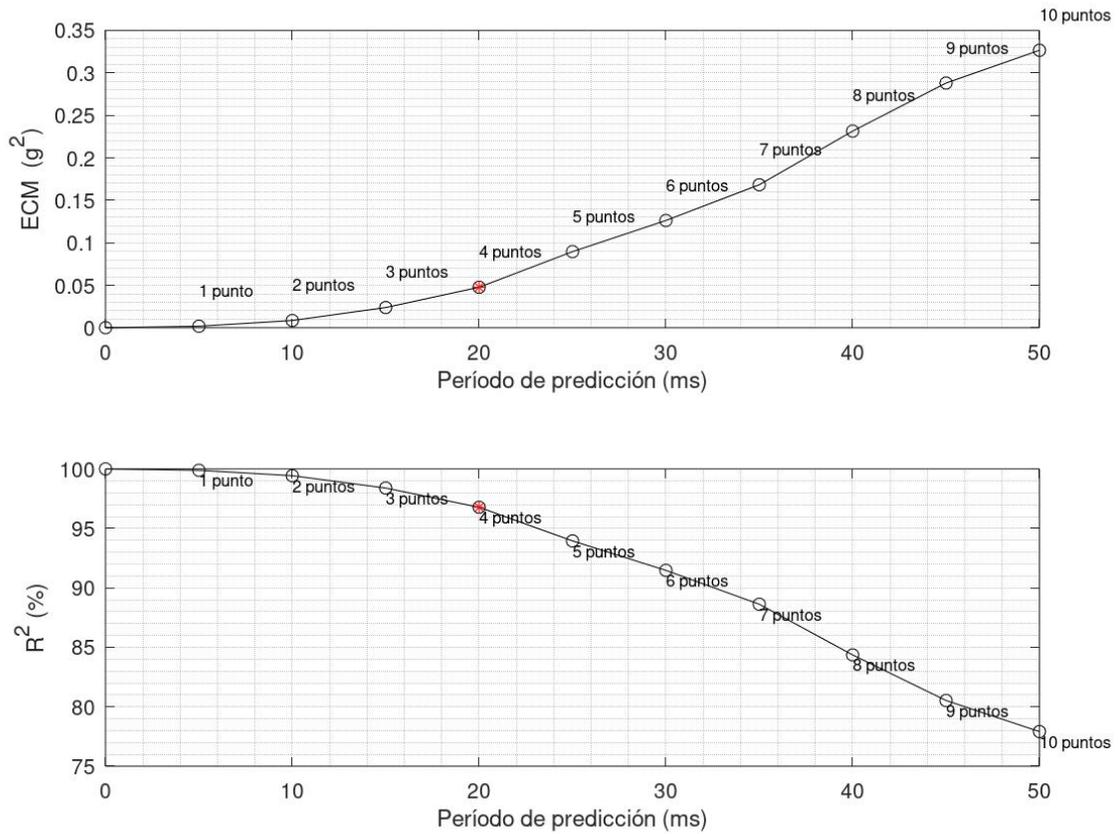


Figura 24: Proyección del error cuadrático medio EMC (Superior) y el coeficiente de correlación múltiple (inferior) para ventanas de predicción del segmento  $a_y$  . **Imagen generada con código en anexo B.6.**

Puntos Adelante	Período de predicción	ECM	$R^2$ (%)
1	5ms	1.7388e-03	99,882
2	10ms	8.4529e-03	99,429
3	15ms	2.3784e-02	98,392
4	20ms	4.7672e-02	96,778
5	25ms	8.9615e-02	93,943
6	30ms	1.2623e-01	91,468
7	35ms	1.6835e-01	88,622
8	40ms	2.3150e-01	84,353
9	45ms	2.8809e-01	80,529
10	50ms	3.2671e-01	77,919

Tabla 8: Valores generadores de la figura 24

La figura 25 muestra la predicción por ventana de 20 milisegundos para el segmento  $a_y$ . La señales son muy similares y se observa la dificultad que tiene el modelo para predecir los cambios repentinos del signo de la pendiente en algunos puntos extremos de la señal. El ECM para esta predicción resulta 1/4 del observado para su similar del segmento  $a_x$

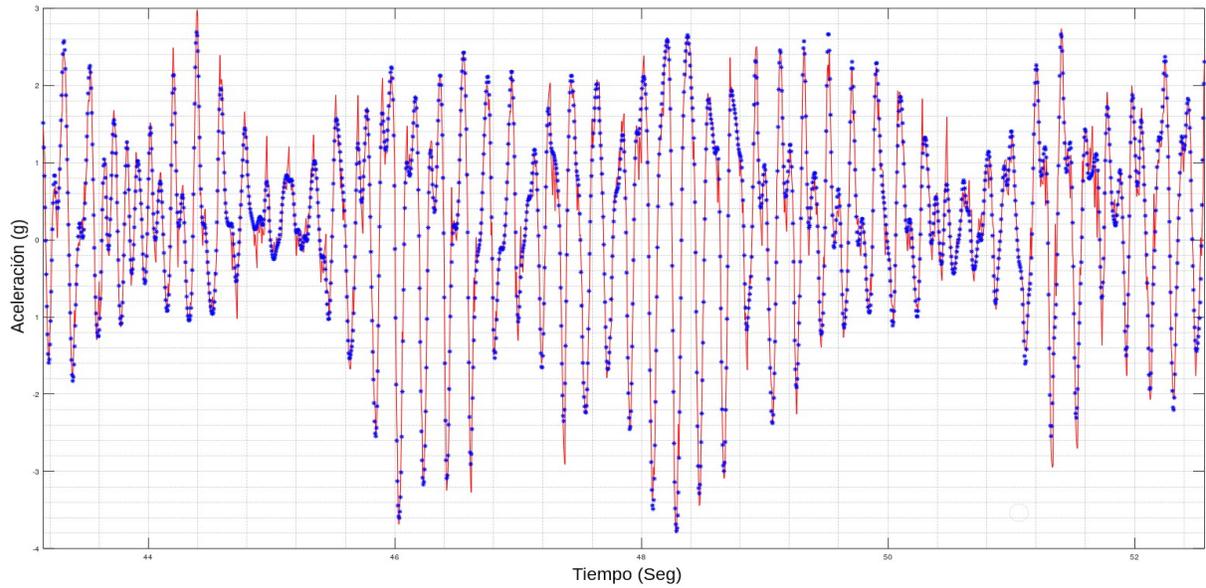


Figura 25: Reconstrucción por predicciones de 20ms adelante. En rojo, la reconstrucción por predicción. Los puntos azules representan muestras reales. **Imagen generada con código en anexo B.6.**

## 5 CONCLUSIONES

Luego de desarrollar todo el proceso investigativo se ha observado que un estudio de autocorrelación para cada serie de tiempo de aceleraciones en los ejes  $X$  e  $Y$  ha evidenciado que las señales registradas son determinísticas y no proceden de un proceso de azar o ruido blanco, además el factor de correlación cercano a 1 en ambas series de tiempo, enfatiza el hecho que los datos han sido registrados con ausencia de ruido.

La descomposición espectrográfica de las dos series de tiempo expresaron las frecuencias predominantes de los temblores en reposo de la enfermedad de Parkinson que han sido descritas por otros investigadores de éste fenómeno fisiológico. El análisis de los invariantes cualitativos de las dinámicas subyacentes para las señales  $a_x$  y  $a_y$ , ha demostrado que nos hemos encontrado con una dinámica caótica con trayectorias (de vectores de estado) de divergencia exponencial reducida y dimensión de inmersión  $D_e = 11$ .

Aún cuando lo ideal en la identificación de la dinámica sería disponer de alguna señal real de entrada al modelo, la construcción de una señal forzadora a partir de las componentes frecuenciales de las series de tiempo a modelar, optimizada empleando el algoritmo de recocido simulado, han dado resultados suficientemente satisfactorios y permite independencia de sensorización adicional en una posible implementación de la metodología de predicción. Un segmento de señal de entrada diseñada, pudiera emplearse recursivamente, dejando que el algoritmo de identificación NARMAX encuentre el mejor modelo para describir la salida a partir de la señal de entrada recursiva.

Luego de realizar muchas pruebas para lograr modelar la dinámica, fué evidente que los modelos encontrados en base a un conjunto inicial de valores de entrada y salida, perdían rápidamente su capacidad de predicción MPO como se observó en el creciente ECM, encontrándose que satisfactoriamente podían predecir solo 4 puntos o 20 milisegundos futuros, cumpliendo con un porcentaje de correlación superior al 95%. Esto pudo deberse en parte a las características caóticas de las series de tiempo y a no emplear una señal de entrada real derivada de la fisiología de la persona con EP. En éste sentido, una estrategia de predicción por ventanas temporales para MPO han dado un excelente resultado en la reconstrucción de la señal y es consecuente con la idea de la implementación de la metodología en tiempo real. No obstante, la capacidad computacional para ejecutar todo el proceso debe ser tal que pueda realizarse todos los cálculos de identificación y predicción (y posibles acciones de control) en el intervalo de 20 milisegundos.

Entre las tareas que quedan por hacer respecto a esta línea de investigación, se propone probar la metodología usada en esta AFE, empleando registros electromiográficos como señal de entrada al modelo, las cuales serán muestreadas simultáneamente con las aceleraciones. Por otro lado, queda la implementación del proceso de predicción utilizado aquí, como una aplicación de estabilización de imagenes en pantalla u otro utensilio de mano.

## 6 BIBLIOGRAFÍA

1. GOBIERNO DE CHILE, MINISTERIO DE SALUD. Guía Clínica ENFERMEDAD DE PARKINSON. SANTIAGO: Minsal, 2010. Disponible en: [min-sal.cl/portal/url/item/955578f79a0cef2ae04001011f01678a.pdf](http://min-sal.cl/portal/url/item/955578f79a0cef2ae04001011f01678a.pdf)
2. Sadeghi M., Jafari A.H., Firoozabadi S.M.P., (2011). A Chaotic Study on Tremor Behavior of Parkinsonian Patients under Deep Brain Stimulation. *World Academy of Science, Engineering and Technology International Journal of Medical and Health Sciences*. **5(2)**
3. Haeria M, Sarbaza Y, Gharibzadehb S., (2005). Modeling the Parkinson's tremor and its treatments. *Journal of Theoretical Biology*. **236**:311-322.
4. Rehan M, Hong K-S (2013) Modeling and Automatic Feedback Control of Tremor: Adaptive Estimation of Deep Brain Stimulation. *PLoS ONE*. **8(4)**:1-11
5. Shah V, Goyal S, Palanthandalam-Madapusi H., (2017). Clinical Facts Along With a Feedback Control Perspective Suggest That Increased Response Time Might be the Cause of Parkinsonian Rest Tremor. *arXiv* : 1403.0296v3.
6. Zulkeffi A, Muthalif A, Nordin D, Syam T. (2019). Intelligent glove for suppression of resting tremor in Parkinson's disease. *VIBROENGINEERING PROCEDIA*.. **29** :176-81.
7. Shahtalebi S, Atashzar S, Samotus O, Patel R, Jog M, Mohammadi A., (2020). PHT-Net: Characterization and Deep Mining of Involuntary Pathological Hand Tremor using Recurrent Neural Network Models. *Scientific Reports*. **10(2195)**
8. Hwei P, Hsu. (1998) Prentice Hall. Análisis de Fourier. (1 ed). México: PRENTICE HALL.
9. A Yu Meigall, S M Rissanen, M P Tarvainen, S D Georgiadis, P A Karjalainen, O Airaksinen and M Kankaanpää, (2012). Linear and nonlinear tremor acceleration characteristics in patients with Parkinson's disease. *PHYSIOLOGICAL MEASUREMENT*. **33**: 395-412
10. Aguirre L.A., Barros V.C., Souza A. (1999). Nonlinear multivariable modeling and analysis of sleep apnea time series. *Elsevier: Computers in Biology and Medicine*. **29**:208-228.
11. Billings, S.A. (2013). NONLINEAR SYSTEM IDENTIFICATION. NARMAX METHODS IN THE TIME, FREQUENCY AND SPATIO-TEMPORAL DOMAINS. (1 ed). United Kingdom. Wiley.

12. Acuña G., Curilem M., Cubillos F. (2014). Desarrollo de un Sensor Virtual basado en Modelo NARMAX y Máquina de Vectores de Soporte para Molienda Semiautógena. *Elsevier: Revista Iberoamericana de Automática e Informática industrial*.**11**:109-116.
13. Kantz H. & Schreiber T. (2004). Nonlinear Time Series Análisis (2 ed.).UK: Cambridge University Press.
14. Small, M. (2005). Applied Nonlinear Time Series Análisis, Application in Physics, Physiology and Finance. Singapore: World Scientific Publishing Co. Pte. Ltd.
15. Xiang Y., Gubian S., Suomela B., Hoeng J. (2013). Generalized Simulated Annealing for Global Optimization: The GenSA Package. *The R Journal*. **5(1)**:13-28.
16. Ayala H., Gritti M., Coelho L. (2020). An R library for nonlinear black-box system identification. *Elsevier: SoftwareX*. **11**(100495 ): 1-5.

## 7 ANEXOS

### 7.0.1 Anexo A.1 : Densidad espectral de energía (DEE)

Sea la señal  $x(t)$  definida con energía finita. Esto es:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \text{finito} \quad (12)$$

La densidad espectral de energía de  $x(t)$  viene dada por:

$$DEE = |X(f)|^2 \quad \text{expresado en [J/Hz]} \quad (13)$$

Donde  $X(f)$  es la Transformada de Fourier de  $x(t)$ .

### 7.0.2 Anexo A.2 : La Transformada de Gabor

Fue introducida por el físico inglés de origen húngaro Dennis Gabor en la década de los cuarenta. La idea fundamental de Gabor es la introducción de ventanas para poder observar la parte de la señal presente en algún entorno del punto de estudio. De esta forma se logra en primer lugar que las componentes frecuenciales de la señal en aquel instante dependan solamente de las proximidades del punto y no influyan en ellas las partes alejadas de la señal. Como hay muchos tipos de ventanas elegibles resulta que existen muchos tipos de transformadas de Gabor. Esto ya constituye una diferencia notable con la transformada de Fourier que como sabemos es única. Sin embargo a pesar de esta posibilidad de variada elección el propio Gabor ya apuntó el hecho de que el mejor candidato a ser ventana es una función gaussiana, y esto por el sencillo hecho de ser esta función la única no nula donde se alcanza la máxima concentración simultánea en tiempo y frecuencia, ya que por el principio de incertidumbre para señales esta simultánea concentración no puede hacerse arbitrariamente grande.

Concretamente si indicamos con  $g(t)$  la ventana de Gabor y queremos analizar una señal  $x(t)$  en un punto  $\tau$ , lo que hacemos es multiplicar la señal  $x(t)$  por la ventana trasladada a  $\tau$  y ahora aplicamos a este producto  $x(t)g(t - \tau)$  la transformada de Fourier. De esta forma para cada par  $(\tau, \omega)$  obtenemos un valor

$$G(\tau, \omega) = \int_{-\infty}^{+\infty} x(t)g(t - \tau)e^{-2\pi i t \omega} dt \quad (14)$$

que puede ser interpretado como la componente en la frecuencia  $\omega$  que la señal  $x(t)$  tiene en el instante  $\tau$ . De esta forma la transformada de la señal pasa a ser bidimensional, informandonos del comportamiento espectral de la señal a lo largo del tiempo.

### 7.0.3 Anexo A.3 : Teorema de inmersión de Takens

Sea  $\mathbf{M}$  compacto. Hay un subconjunto denso y abierto  $\mathcal{O}\mathcal{D}$  de  $Diff(\mathbf{M}) \times C^k(M, \mathbb{R})$  con la propiedad de mapa de Takens:

$$f : M \rightarrow \mathbb{R}^{2m+1} \quad (15)$$

dado por  $f(x) = (E(x), E \circ \phi(x), E \circ \phi^2(x), \dots, E \circ \phi^{2m}(x))$  es una inmersión de multiples  $C^k$  donde  $(\phi, E) \in \mathcal{O}\mathcal{D}$

El teorema demostrado por Takens en 1981 nos dice que la órbita que siguen los estados en el espacio de fase n-dimensional, es equivalente (homeomorfa) a la órbita seguida por los vectores de retardo en el espacio 2m+1-dimensional. Por ello, la dinámica del sistema y su comportamiento se pueden describir analizando la reconstrucción del atractor de una de las variables del sistema. Debe observarse que el teorema trata sobre la equivalencia homeomorfa de los atractores a los que tienden las órbitas, esto no significa que sean iguales ni mucho menos, sino que el comportamiento cualitativo de ambas es el mismo (tienen la misma dimensión topológica, el mismo carácter cíclico, etc.).

### 7.0.4 Anexo A.4 : Definición de máximo exponente de Lyapunov

El exponente de Lyapunov o exponente característico de Lyapunov de un sistema dinámico es una cantidad que caracteriza la tasa de separación de trayectorias infinitesimalmente cercanas. Cuantitativamente, dos trayectorias en el espacio de fase con un vector de separación inicial, divergen a una tasa dada por  $\delta Z_0$  tal que  $|\delta Z(t)| \approx e^{\lambda t} |\delta Z_0|$  donde  $\lambda$  es el exponente de Lyapunov.

El exponente máximo de Lyapunov se puede definir de la siguiente manera:

$$\lambda = \lim_{t \rightarrow \infty} \lim_{|\delta Z_0| \rightarrow 0} \frac{1}{t} \ln \left( \frac{|\delta Z(t)|}{|\delta Z_0|} \right) \quad (16)$$

Dependiendo del valor del máximo exponente se tienen las siguientes tipos de dinámicas:

Dinámica de	Valores de max. exponente
punto fijo	$\lambda < 0$
ciclo límite	$\lambda = 0$
caos	$0 < \lambda < \infty$
ruido	$\lambda = \infty$

Tabla 9: Tipo de dinámica según valor de máximo exponente de Lyapunov

### 7.0.5 Anexo A.5 : Regresores y parámetros resultantes para modelo NARMAX de $a_x$ con predicción MPO para primera iteración

Este modelo deriva del modelado iterativo llevado a cabo con los primeros 300 puntos usando predicción MPO del segmento  $a_x$  en una ventana de 50ms. Aquí;  $-y-$  representa valor de la señal real,  $-\mathbf{u}-$  es la entrada basada en la ecuación (11) y  $-e-$  es el error de predicción.  $\hat{a}$  es el valor predicho.

$$\text{narmax}(n_a = 2, n_u = 6, n_e = 2, l = 2)$$

$$\begin{aligned} \hat{a} = & 1,8241a(k-1) - 0,9062a(k-2) + 91,5061u(k-4) + 0,9090e(k-1) + 172,8737u(k-3)u(k-6) \\ & + 0,1569e(k-2) - 23,6293u(k-2)e(k-1) - 0,1908e(k-1)e(k-1) - 30,7114u(k-1)e(k-2) - \\ & 9,9171u(k-1) + 45,5239u(k-2) - 88,6819u(k-3) + 7,8661a(k-1)u(k-6) - 62,8222u(k-6)u(k-6) \\ & 457,3015u(k-5)u(k-6) - 561,0258u(k-4)u(k-6) + 159,6995u(k-2)u(k-6) - 8,1130a(k-2)u(k-6) \\ & + 85,5107u(k-1)u(k-1) - 871,7275u(k-5)u(k-5) - 1329,4839u(k-4)u(k-4) + \\ & 0,0589a(k-2)e(k-1) - 33,1329a(k-1)u(k-5) + 11,6748u(k-6) - 107,5384u(k-1)u(k-6) - \\ & 30,8762a(k-2)u(k-2) - 50,0453u(k-5) - 574,7764u(k-2)u(k-5) + 35,0467a(k-2)u(k-5) \\ & + 27,0752a(k-1)u(k-2) + 414,8998u(k-1)u(k-5) - 802,2788u(k-1)u(k-4) - 5,6523a(k-1)u(k-1) \\ & + 58,7571a(k-1)u(k-4) - 2187,7228u(k-2)u(k-3) - 761,9310u(k-3)u(k-5) + \\ & 2224,4405u(k-4)u(k-5) - 0,0899a(k-2)a(k-2) + 0,2004a(k-1)a(k-2) + 61,0531a(k-2)u(k-3) \\ & - 0,1110a(k-1)e(k-1) - 54,9614a(k-1)u(k-3) - 63,6740a(k-2)u(k-4) - \\ & 2,3304e(k-2)e(k-2) + 0,0429a(k-1)e(k-2) + 6,6337a(k-2)u(k-1) - 0,1092a(k-1)a(k-1) + \\ & 133,1180u(k-2)e(k-2) - 244,0690u(k-3)e(k-2) + 763,8440u(k-3)u(k-3) + 883,4304u(k-2)u(k-2) \\ & + 349,4090u(k-3)u(k-4) + 1423,0799u(k-2)u(k-4) + 919,2492u(k-1)u(k-3) - \\ & 594,4129u(k-1)u(k-2) + 2,8707e(k-1)e(k-2) - 0,0210a(k-2)e(k-2) + 5,6856u(k-1)e(k-2) \end{aligned}$$

$$1) - 2,6635u(k-6)e(k-1) + 40,9658u(k-3)e(k-1) + 16,2695u(k-5)e(k-1) - 36,4726u(k-4)e(k-1) + 25,3653u(k-6)e(k-2) - 119,7043u(k-5)e(k-2) + 235,8158u(k-4)e(k-2) + 0,0597$$

### 7.0.6 Anexo A.6 : Regresores y parámetros resultantes para modelo NARMAX de $a_Y$ con predicción MPO para primera iteración

Este modelo deriva del modelado iterativo llevado a cabo con los primeros 300 puntos usando predicción MPO del segmento  $a_y$  en una ventana de 50ms. Aquí;  $-y-$  representa valor de la señal real,  $-u-$  es la entrada basada en la ecuación (11) y  $-e-$  es el error de predicción.  $\hat{a}$  es el valor predicho.

$$\text{narmax}(n_a = 2, n_u = 6, n_e = 2, l = 2)$$

$$\begin{aligned} \hat{a} = & 1,8882y(k-1) - 0,9352y(k-2) - 0,0372y(k-1)u(k-2) - 0,0168u(k-6) + 1,0264e(k-1) + \\ & 0,0095e(k-1)e(k-1) + 0,0181u(k-2)e(k-2) - 0,0908y(k-1)y(k-1) - 0,1526y(k-2)y(k-2) \\ & - 0,0113y(k-2)u(k-6) - 0,0005u(k-6)e(k-1) - 0,1143e(k-2) + 0,2367y(k-1)y(k-2) + \\ & 0,0100y(k-1)u(k-1) + 0,0291y(k-2)u(k-5) + 0,0075e(k-1)e(k-2) + 0,0130e(k-2)e(k-2) \\ & + 0,0661u(k-2) + 0,0027u(k-1)u(k-1) + 0,0103u(k-4)u(k-5) - 0,0038u(k-1)u(k-6) - \\ & 0,0053u(k-5)u(k-5) + 0,0352u(k-5) - 0,0017u(k-4) - 0,0035y(k-1)u(k-5) - 0,0235u(k-1) \\ & - 0,0178u(k-1)u(k-2) + 0,0139u(k-1)u(k-5) - 0,0045u(k-3)u(k-6) + 0,0509y(k-1)u(k-3) \\ & + 0,0062y(k-1)u(k-6) + 0,0027u(k-4)e(k-1) - 0,0264u(k-1)u(k-4) + \\ & 0,0292u(k-1)u(k-3) - 0,0180y(k-1)e(k-1) + 0,0163y(k-2)e(k-1) - 0,0288y(k-2)u(k-4) \\ & + 0,0112y(k-2)u(k-3) - 0,0004y(k-2)u(k-1) - 0,0205u(k-2)u(k-5) + 0,0081u(k-3)u(k-5) \\ & + 0,0012u(k-5)u(k-6) + 0,0035u(k-1)e(k-1) - 0,0081u(k-1)e(k-2) - \\ & 0,0054u(k-2)e(k-1) - 0,0126u(k-3)e(k-2) + 0,0020u(k-4)e(k-2) - 0,0260y(k-1)u(k-4) \\ & + 0,0349u(k-3)u(k-3) + 0,0462u(k-2)u(k-4) + 0,0233u(k-2)u(k-2) - 0,0635u(k-2)u(k-3) \\ & - 0,0606u(k-3) - 0,0345u(k-3)u(k-4) + 0,0066u(k-2)u(k-6) - 1,2011 \end{aligned}$$



```

62 for i=1:length(Y1)
63 Y1(i)= MatrizXY(i,2);
64 Y2(i)= MatrizXY(i+1,2);
65 endfor
66 RY=corr(Y1,Y2) % Autocorrelación de pearson de Y
67
68 %%%%%%%%%%%
69 %% Diagrama de Autocorrelación eje Y
70 %%%%%%%%%%%
71
72 figure(3)
73 plot(Y1,Y2,'.'); %Gráfica de espacio de pase para Y
74 grid
75 axis("tight")
76 xlabel("Aceleración en g para eje Y(t) del SmarthPhone");
77 ylabel("Aceleración en g para eje Y(t+5ms) del SmarthPhone");

```

## 7.1.2 Anexo B.2 : Análisis espectral de señales originales.

Código desarrollado en Octave.

```

80
81 %%%%%%%%%%%
82 % - Calcula y gráfica la FFT de las señales.
83 % - Identifica las frecuencias dominantes
84 % - Gráfica espectrogramas de las señales.
85 %%%%%%%%%%%
86 % - Para utilizar este código debe descargarse
87 % - el paquete LTFAT he instalado desde la
88 % - ventana de comandos con la instrucción
89 % - pkg load ltfat
90 %%%%%%%%%%%
91 pkg load ltfat
92 format long;
93 %%%%%%%%%%%
94 % Carga archivos con señales
95 %%%%%%%%%%%
96 load SELECCIONDEEJEX.txt % Carga variable ax
97 load SELECCIONDEEJEY.txt % Carga variable ay
98 %%%%%%%%%%%
99 n=length(EJEX(:,1));
100 POLARES=zeros(4,2);
101 Fs = 200; % frecuencia de muestreo (en Hz)
102 t = 0:136/(n-1):136; % vector tiempo de 136 segundo
103 x=EJEX(:,2);
104 y=EJEY(:,2);
105 % Amplitud-Fase vs. Frecuencias
106 gx=fft(x); % Calcula la FFT de cada señal
107 gy=fft(y);
108 gx1=gx(1:n/2); % la FFT es simétrica, así que se elimina la mitad
109 gy1=gy(1:n/2);
110 gx1(1)=0; % Corrección de señal continua espuria
111 gy1(1)=0;
112 powergx1=abs(gx1).^2; %Calcula la densidad espectral
113 powergy1=abs(gy1).^2; %Calcula la densidad espectral
114 %Identificación de frecuencias dominantes
115 % Para ax y ay
116 fx1=fx2=fx3=fx4=0.00000;
117 fy1=fy2=fy3=fy4=0.00000;
118 Segmento1=powergx1(1:1000); % Selec. Rango de frec. ax
119 Segmento2=powergy1(1:1000); % Selec. Rango de frec. ay
120 [M, I] = max(Segmento1);
121 POLARES(1,1)=gx1(I);
122 fx1=(Fs/n)*I;
123 Ex1=M;
124 [M, I] = max(Segmento2);
125 POLARES(1,2)=gy1(I);
126 fy1=(Fs/n)*I;
127 Ey1=M;

```

```

128 Segmento1=powergx1(1000:2000); % Selec. Rango de frec. ax
129 Segmento2=powergy1(1000:2000); % Selec. Rango de frec. ay
130 [M, I] = max(Segmento1);
131 POLARES(2, 1)=gx1(I)
132 fx2=(Fs/n)*(I+1000);
133 Ex2=M;
134 [M, I] = max(Segmento2);
135 POLARES(2, 2)=gy1(I)
136 fy2=(Fs/n)*(I+1000);
137 Ey2=M;
138 Segmento1=powergx1(2000:3000); % Selec. Rango de frec. ax
139 Segmento2=powergy1(2000:3000); % Selec. Rango de frec. ay
140 [M, I] = max(Segmento1);
141 POLARES(3, 1)=gx1(I)
142 fx3=(Fs/n)*(I+2000);
143 Ex3=M;
144 [M, I] = max(Segmento2);
145 POLARES(3, 2)=gy1(I)
146 fy3=(Fs/n)*(I+2000);
147 Ey3=M;
148 Segmento1=powergx1(2400:4000); % Selec. Rango de frec. ax
149 Segmento2=powergy1(2400:4000); % Selec. Rango de frec. ay
150 [M, I] = max(Segmento1);
151 POLARES(4, 1)=gx1(I)
152 fx4=(Fs/n)*(I+2400);
153 Ex4=M;
154 [M, I] = max(Segmento2);
155 POLARES(4, 2)=gy1(I)
156 fy4=(Fs/n)*(I+2400);
157 Ey4=M;
158
159 %format 'rat'
160 printf("Frecuencias principales de ax son: \n")
161 fx1
162 fx2
163 fx3
164 fx4
165 Pex1=Ex1/(Ex1+Ex2+Ex3+Ex4)
166 Pex2=Ex2/(Ex1+Ex2+Ex3+Ex4)
167 Pex3=Ex3/(Ex1+Ex2+Ex3+Ex4)
168 Pex4=Ex4/(Ex1+Ex2+Ex3+Ex4)
169 printf("Frecuencias principales de ay son: \n")
170 fy1
171 fy2
172 fy3
173 fy4
174 Pey1=Ey1/(Ey1+Ey2+Ey3+Ey4)
175 Pey2=Ey2/(Ey1+Ey2+Ey3+Ey4)
176 Pey3=Ey3/(Ey1+Ey2+Ey3+Ey4)
177 Pey4=Ey4/(Ey1+Ey2+Ey3+Ey4)
178
179 %%%%%%%%%%%
180 % Vector de frecuencias
181 f = (0:n/2-1)*Fs/n; %construccion del vector de frecuencias
182 f2 = (0:n-1)*Fs/n; %construccion del vector de frecuencias
183 %%%%%%%%%%%
184 figure(1) % Imagenes de FFT para ax y ay
185 subplot(8, 2, [1,3,5,7,9]);
186 plot(f(2:4002)', 2*powergx1(2:4002), 'b');
187 xlabel('Frecuencia en Hz');
188 ylabel('Densidad Espectral de Potencia ');
189 title('FFT serie de tiempo ax');
190 grid
191 subplot(8, 2, [13,15]);
192 plot(f(2:4002)', 2*powergx1(2:4002), 'b');
193 xlabel('Frecuencia en Hz');
194 title('Zoom de la base en gráfica superior');
195 axis([0,30,0,(max(powergx1))/40])
196 grid
197 subplot(8, 2, [2,4,6,8,10]);
198 plot(f(2:4002)', powergy1(2:4002), 'b');
199 xlabel('Frecuencia en Hz');
200 ylabel('Densidad Espectral de Potencia ');
201 title('FFT serie de tiempo ay');

```

```

202| grid
203| subplot (8, 2, [14,16]);
204| plot (f(2:4002)',powergy1(2:4002),'b');
205| xlabel('Frecuencia en Hz');
206| title('Zoom de la base en gráfica superior');
207| axis([0,30,0,(max(powergy1))/60])
208| grid
209| %%%%%%%%%%%
210| % Imagen 3D de espectro de frecuencias.
211| % Define rango de magnitud a representar
212| figure(2); % Señal ax.
213| clim=[0 50];
214| C=sgram(x,Fs,'nocolorbar','lin','clim',clim);
215| [m,n]=size(C);
216| t = 0:136/(n-1):136; % Vector tiempo de 136 segundo
217| f = 0:Fs/(2*(m-1)):Fs/2; % Vector de frecuencias
218| ff=f;
219| CC=C;
220| CC(241:m,:)=[];
221| ff(241:m)=[];
222|
223| mesh(t, ff',CC);
224| xlabel('Tiempo (Seg)');
225| ylabel('Frecuencia (Hz)');
226| zlabel('Magnitud de componente frecuencial |Ax(f)|');
227| title('Espectrograma 3D de serie temporal ax');
228|
229| figure(3); % Señal ay.
230| C=sgram(y,Fs,'nocolorbar','lin','clim',clim);
231| [m,n]=size(C);
232| t = 0:136/(n-1):136; % Vector tiempo de 136 segundo
233| f = 0:Fs/(2*(m-1)):Fs/2; % Vector de frecuencias
234| ff=f;
235| CC=C;
236| CC(241:m,:)=[];
237| ff(241:m)=[];
238|
239| mesh(t, ff',CC);
240| xlabel('Tiempo (Seg)');
241| ylabel('Frecuencia (Hz)');
242| zlabel('Magnitud de componente frecuencial |Ay(f)|');
243| title('Espectrograma 3D de serie temporal ay');
244| %%%%%%%%%%%

```

### 7.1.3 Anexo B.3 : Invariantes cualitativos de señales.

Código desarrollado en Octave.

```

246| %%%%%%%%%%%
247| %Renormaliza los segmentos de señal
248| %%%%%%%%%%%
249| % Este código traslada a 0 en el tiempo
250| % los segmentos de señal y divide todos los
251| % términos de la señal entre el primero para
252| % obtener valor de 1 en el primer término
253| %Nota: Debe agregar en el mismo directorio los sig. archivos de Tisean:
254| %_false_nearest...oct
255| %false_nearest.m
256| %%%%%%%%%%%
257| clear all
258| close all
259| clc
260| pkg load matgeom;
261| format long;
262| %%%%%%%%%%%
263| % Carga archivos con señales
264| %%%%%%%%%%%
265| load SEGMENTODEEJEX.txt % Carga variable ax
266| load SEGMENTODEEJEY.txt % Carga variable ay

```





```

399 | l <-N
400 | t1 <-t[k:l,1]
401 | y1 <-y[k:l,1]
402 | u1 <- P[1]*cos(2*pi*P[5]*t1)+P[2]*cos(2*pi*P[6]*t1)+P[3]*cos(2*pi*P[7]*t1)+P[4]*cos(2*pi*P[8]*t1)
403 | dim(u1) <- c((1-k+1),1)
404 | mdl <- estimate(mdl, y1, u1, rho.p = 1e-36, rho.n = 1e-36)
405 | ax <- predict(mdl, y1, u1, K = 0)
406 | Result <-abs(ax$R2-1)
407 | Result
408 | }
409 | #-----
410 | # Optimización con Recocido simulado
411 | set.seed(1234) # The user can use any seed.
412 | P0 <- c(1,1,1,1,4.904412,10.17647,15.48529,20.30147)
413 | lower <- c(-2e+2,-2e+2,-2e+2,-2 -2e+2,4,8,13,18 )
414 | upper <- c(2e+2,2e+2,2e+2,2e+2,7,12,17,22 )
415 | maxit <- 50
416 | global.min <- 0
417 | tol <- 1e-03
418 | nb.stop.improvement <-500
419 | max.call <- 1e10
420 | max.time <-14400
421 | temperature <-15
422 | verbose=T
423 |
424 |
425 | out <- GenSA(par = P0, fn=Correlacion, lower=lower, upper=upper,
426 |             control=list(maxit=maxit,nb.stop.improvement=nb.stop.improvement,
427 |                           max.time=max.time,temperature=temperature,
428 |                           threshold.stop= global.min+tol) )
429 | out[c("value", "par", "counts")]
430 | print(out)
431 | # Gráficas-----
432 | t1 <-t[1:N,1]
433 | u <- out$par[1]*cos(2*pi*out$par[5]*t1)+out$par[2]*cos(2*pi*out$par[6]*t1)+out$par[3]*cos(2*pi*
434 |             out$par[7]*t1)+out$par[4]*cos(2*pi*out$par[8]*t1)
435 | mdl <- estimate(mdl, y, u, rho.p = 1e-36, rho.n = 1e-36)
436 | ax <- predict(mdl,y, u, K = 0)
437 | plot(ax$dfpred$yh, type='o', col="red", lwd=1, ann=F, axes=F)
438 | par(new=TRUE)
439 | plot(y, type='l', col="blue", lwd=1, ann=F, axes=F)
440 | #-----
441 | write.table(out$par, file = "PARAMETROS_ENTRADA_ax.txt", row.names = FALSE, col.names = FALSE)
442 | #-----

```

### 7.1.5 Anexo B.5 : Predicciones por ventana.

Código desarrollado en RStudio. Este script corresponde a la predicción por ventanas de tipo MPO. En cada modelado se emplean 300 puntos y las predicciones son MPO desde 5ms hasta los 50mS.

Para los segmentos  $a_x$  y  $a_y$  tenemos el siguiente código (se modifican los archivos utilizados según sea la señal a estudiar):

```

443 | library(narmax)
444 |
445 | #Carga de archivo con la señal ax
446 | Dx <- matrix(scan("../data/SELECCIONDEEJEX.txt"),
447 |             nrow=27200,
448 |             byrow=TRUE)
449 | #Carga de archivo con parámetros para señal de entrada
450 | Px <- matrix(scan("../data/PARAMETROS_ENTRADA_ax.txt"),
451 |             nrow=9,
452 |             byrow=TRUE)
453 | L <-8335 #Punto inicial del segmento a modelar

```

```

454 N <- 10534 #Punto final del segmento a modelar
455 t <- Dx[L:N,1] #Vector de tiempo
456 y <- Dx[L:N,2] #Vector de valores de segmento a modelar
457 #Generación de vector de entrada
458 u <- Px[1]*cos(2*pi*Px[5]*t)+Px[2]*cos(2*pi*Px[6]*t)+Px[3]*cos(2*pi*Px[7]*t)+Px[4]*cos(2*pi*Px[8]*t)
459
460 dim(t) <- c(N-L+1,1)
461 dim(y) <- c(N-L+1,1)
462 dim(u) <- c(N-L+1,1)
463 ax_ventana <- matrix(nrow = 1896, ncol = 10)
464
465 # Definición de estructura -----
466 mdl2 <- narmax(ny = 2, nu = 6, ne = 2, nl = 2)
467
468 #-----
469 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 1 PASO (5 ms)
470
471 mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho_p = 1e-36, rho_n = 1e-36)
472 mdl_mpo <- mdl2
473 ax_mpo <- predict(mdl2, y[1:301,1], u[1:301,1], K = 0)
474 ax_ventana[1,1] <- ax_mpo$dfpred$yh[295]
475
476
477 for(i in 1:1895){
478   mdl2 <- estimate(mdl2, y[(i+1):(300+i),1], u[(i+1):(300+i),1], rho_p = 1e-36, rho_n = 1e-36)
479   ax_mpo <- predict(mdl2, y[(295+i):(301+i),1], u[(295+i):(301+i),1], K = 0)
480   ax_ventana[i+1,1] <- ax_mpo$dfpred$yh
481 }
482
483
484 #-----
485 #-----
486 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 2 PASO (10 ms)
487
488 mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho_p = 1e-36, rho_n = 1e-36)
489 mdl_mpo <- mdl2
490 ax_mpo <- predict(mdl2, y[1:302,1], u[1:302,1], K = 0)
491 ax_ventana[1:2,2] <- ax_mpo$dfpred$yh[295:296]
492
493
494 for(i in 1:947){
495   mdl2 <- estimate(mdl2, y[(2*i+1):(300+2*i),1], u[(2*i+1):(300+2*i),1], rho_p = 1e-36, rho_n = 1e-
36)
496   ax_mpo <- predict(mdl2, y[(295+2*i):(300+2*(i+1)),1], u[(295+2*i):(300+2*(i+1)),1], K = 0)
497   ax_ventana[(2*i+1):(2*(i+1)),2] <- ax_mpo$dfpred$yh
498 }
499
500
501 #-----
502 #-----
503 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 3 PASO (15 ms)
504
505 mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho_p = 1e-36, rho_n = 1e-36)
506 mdl_mpo <- mdl2
507 ax_mpo <- predict(mdl2, y[1:303,1], u[1:303,1], K = 0)
508 ax_ventana[1:3,3] <- ax_mpo$dfpred$yh[295:297]
509
510
511 for(i in 1:631){
512   mdl2 <- estimate(mdl2, y[(3*i+1):(300+3*i),1], u[(3*i+1):(300+3*i),1], rho_p = 1e-36, rho_n = 1e-
36)
513   ax_mpo <- predict(mdl2, y[(295+3*i):(300+3*(i+1)),1], u[(295+3*i):(300+3*(i+1)),1], K = 0)
514   ax_ventana[(3*i+1):(3*(i+1)),3] <- ax_mpo$dfpred$yh
515 }
516
517
518 #-----
519 #-----
520 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 4 PASO (20 ms)
521
522 mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho_p = 1e-36, rho_n = 1e-36)
523 mdl_mpo <- mdl2
524 ax_mpo <- predict(mdl2, y[1:304,1], u[1:304,1], K = 0)
525 ax_ventana[1:4,4] <- ax_mpo$dfpred$yh[295:298]

```

```

526|
527|
528| for(i in 1:473){
529|   mdl2 <- estimate(mdl2, y[(4*i+1):(300+4*i)],1], u[(4*i+1):(300+4*i)],1], rho.p = 1e-36, rho.n = 1e-
      36)
530|   ax_mpo <- predict(mdl2, y[(295+4*i):(300+4*(i+1))], 1], u[(295+4*i):(300+4*(i+1)),1], K = 0)
531|   ax_ventana[(4*i+1):(4*(i+1)),4] <- ax_mpo$dfpred$yh
532| }
533|
534|
535| #-----
536| #-----
537| #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 5 PASO (25 ms)
538|
539| mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho.p = 1e-36, rho.n = 1e-36)
540| mdl_mpo<-mdl2
541| ax_mpo <- predict(mdl2, y[1:305,1], u[1:305,1], K = 0)
542| ax_ventana[1:5,5] <- ax_mpo$dfpred$yh[295:299]
543|
544|
545| for(i in 1:378){
546|   mdl2 <- estimate(mdl2, y[(5*i+1):(300+5*i)],1], u[(5*i+1):(300+5*i)],1], rho.p = 1e-36, rho.n = 1e-
      36)
547|   ax_mpo <- predict(mdl2, y[(295+5*i):(300+5*(i+1))], 1], u[(295+5*i):(300+5*(i+1)),1], K = 0)
548|   ax_ventana[(5*i+1):(5*(i+1)),5] <- ax_mpo$dfpred$yh
549| }
550|
551|
552| #-----
553| #-----
554| #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 6 PASO (30 ms)
555|
556| mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho.p = 1e-36, rho.n = 1e-36)
557| mdl_mpo<-mdl2
558| ax_mpo <- predict(mdl2, y[1:306,1], u[1:306,1], K = 0)
559| ax_ventana[1:6,6] <- ax_mpo$dfpred$yh[295:300]
560|
561|
562| for(i in 1:315){
563|   mdl2 <- estimate(mdl2, y[(6*i+1):(300+6*i)],1], u[(6*i+1):(300+6*i)],1], rho.p = 1e-36, rho.n = 1e-
      36)
564|   ax_mpo <- predict(mdl2, y[(295+6*i):(300+6*(i+1))], 1], u[(295+6*i):(300+6*(i+1)),1], K = 0)
565|   ax_ventana[(6*i+1):(6*(i+1)),6] <- ax_mpo$dfpred$yh
566| }
567|
568|
569| #-----
570| #-----
571| #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 7 PASO (35 ms)
572|
573| mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho.p = 1e-36, rho.n = 1e-36)
574| mdl_mpo<-mdl2
575| ax_mpo <- predict(mdl2, y[1:307,1], u[1:307,1], K = 0)
576| ax_ventana[1:7,7] <- ax_mpo$dfpred$yh[295:301]
577|
578|
579| for(i in 1:269){
580|   mdl2 <- estimate(mdl2, y[(7*i+1):(300+7*i)],1], u[(7*i+1):(300+7*i)],1], rho.p = 1e-36, rho.n = 1e-
      36)
581|   ax_mpo <- predict(mdl2, y[(295+7*i):(300+7*(i+1))], 1], u[(295+7*i):(300+7*(i+1)),1], K = 0)
582|   ax_ventana[(7*i+1):(7*(i+1)),7] <- ax_mpo$dfpred$yh
583| }
584|
585|
586| #-----
587| #-----
588| #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 8 PASO (40 ms)
589|
590| mdl2 <- estimate(mdl2, y[1:300,1], u[1:300,1], rho.p = 1e-36, rho.n = 1e-36)
591| mdl_mpo<-mdl2
592| ax_mpo <- predict(mdl2, y[1:308,1], u[1:308,1], K = 0)
593| ax_ventana[1:8,8] <- ax_mpo$dfpred$yh[295:302]
594|
595|

```

```

596 for (i in 1:236){
597   mdl2 <- estimate(mdl2, y[(8*i+1):(300+8*i)], 1, u[(8*i+1):(300+8*i)], 1, rho_p = 1e-36, rho_n = 1e-
36)
598   ax_mpo <- predict(mdl2, y[(295+8*i):(300+8*(i+1))], 1, u[(295+8*i):(300+8*(i+1))], 1, K = 0)
599   ax_ventana[(8*i+1):(8*(i+1)), 8] <- ax_mpo$dfpred$yh
600 }
601
602
603 #-----
604 #-----
605 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 9 PASO (45 ms)
606
607 mdl2 <- estimate(mdl2, y[1:300, 1], u[1:300, 1], rho_p = 1e-36, rho_n = 1e-36)
608 mdl_mpo <- mdl2
609 ax_mpo <- predict(mdl2, y[1:309, 1], u[1:309, 1], K = 0)
610 ax_ventana[1:9, 9] <- ax_mpo$dfpred$yh[295:303]
611
612
613 for (i in 1:209){
614   mdl2 <- estimate(mdl2, y[(9*i+1):(300+9*i)], 1, u[(9*i+1):(300+9*i)], 1, rho_p = 1e-36, rho_n = 1e-
36)
615   ax_mpo <- predict(mdl2, y[(295+9*i):(300+9*(i+1))], 1, u[(295+9*i):(300+9*(i+1))], 1, K = 0)
616   ax_ventana[(9*i+1):(9*(i+1)), 9] <- ax_mpo$dfpred$yh
617 }
618
619
620 #-----
621 #-----
622 #GENERACIÓN DE PREDICCIONES POR VENTANAS DE 9 PASO (50 ms)
623
624 mdl2 <- estimate(mdl2, y[1:300, 1], u[1:300, 1], rho_p = 1e-36, rho_n = 1e-36)
625 mdl_mpo <- mdl2
626 ax_mpo <- predict(mdl2, y[1:310, 1], u[1:310, 1], K = 0)
627 ax_ventana[1:10, 10] <- ax_mpo$dfpred$yh[295:304]
628
629
630 for (i in 1:188){
631   mdl2 <- estimate(mdl2, y[(10*i+1):(300+10*i)], 1, u[(10*i+1):(300+10*i)], 1, rho_p = 1e-36, rho_n =
1e-36)
632   ax_mpo <- predict(mdl2, y[(295+10*i):(300+10*(i+1))], 1, u[(295+10*i):(300+10*(i+1))], 1, K = 0)
633   ax_ventana[(10*i+1):(10*(i+1)), 10] <- ax_mpo$dfpred$yh
634   i
635 }
636
637
638 #-----
639 #-----
640 # Gráficas de predicciones MPO
641 plot(ax_ventana[, 1], type='l', col="red", lwd=1, ann=F, axes=F)
642 par(new=TRUE)
643 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
644
645 plot(ax_ventana[, 2], type='l', col="red", lwd=1, ann=F, axes=F)
646 par(new=TRUE)
647 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
648
649 plot(ax_ventana[, 3], type='l', col="red", lwd=1, ann=F, axes=F)
650 par(new=TRUE)
651 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
652
653 plot(ax_ventana[, 4], type='l', col="red", lwd=1, ann=F, axes=F)
654 par(new=TRUE)
655 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
656
657 plot(ax_ventana[, 5], type='l', col="red", lwd=1, ann=F, axes=F)
658 par(new=TRUE)
659 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
660
661 plot(ax_ventana[, 6], type='l', col="red", lwd=1, ann=F, axes=F)
662 par(new=TRUE)
663 plot(y[(301:2196), 1], type='l', col="blue", lwd=1, ann=F, axes=F)
664
665 plot(ax_ventana[, 7], type='l', col="red", lwd=1, ann=F, axes=F)
666 par(new=TRUE)

```

```

667 plot(y[(301:2196),1], type='l', col="blue", lwd=1, ann=F, axes=F)
668
669 plot(ax_ventana[,8], type='l', col="red", lwd=1, ann=F, axes=F)
670 par(new=TRUE)
671 plot(y[(301:2196),1], type='l', col="blue", lwd=1, ann=F, axes=F)
672
673 plot(ax_ventana[,9], type='l', col="red", lwd=1, ann=F, axes=F)
674 par(new=TRUE)
675 plot(y[(301:2196),1], type='l', col="blue", lwd=1, ann=F, axes=F)
676
677 plot(ax_ventana[,10], type='l', col="red", lwd=1, ann=F, axes=F)
678 par(new=TRUE)
679 plot(y[(301:2196),1], type='l', col="blue", lwd=1, ann=F, axes=F)
680 #-----
681 # Guradado de resultados
682 ResultadosX <- matrix(nrow=1896, ncol =13)
683 ResultadosX[,1]<-t[(301:2196),1]
684 ResultadosX[,2]<-u[(301:2196),1]
685 ResultadosX[,3]<-y[(301:2196),1]
686 ResultadosX[,4]<-ax_ventana[,1]
687 ResultadosX[,5]<-ax_ventana[,2]
688 ResultadosX[,6]<-ax_ventana[,3]
689 ResultadosX[,7]<-ax_ventana[,4]
690 ResultadosX[,8]<-ax_ventana[,5]
691 ResultadosX[,9]<-ax_ventana[,6]
692 ResultadosX[,10]<-ax_ventana[,7]
693 ResultadosX[,11]<-ax_ventana[,8]
694 ResultadosX[,12]<-ax_ventana[,9]
695 ResultadosX[,13]<-ax_ventana[,10]
696 #Archivado de resultados
697 #-----
698 write.table(ResultadosX, file = "MODELADO.MPO.CONLAGS.ax.txt", row.names = FALSE, col.names = FALSE)
699 #-----

```

## 7.1.6 Anexo B.6 : Gráficas de resultados.

Resultados para segmento  $a_x$  (similar estructura se emplea para el segmento  $a_y$ )

```

701 pkg load statistics
702 pkg load nan
703 clear all
704 close all
705 clc
706 format long;
707 pkg load odepkg
708 load MODELADO.MPO.CONLAGS.ax.txt
709 ax =MODELADO.MPO.CONLAGS.ax;
710 %%%%%%%%%%%
711 n=length(ax(:,1));
712 ax=ax(1:n-10,:);
713 n=length(ax(:,1));
714
715 figure(1) %Comparación de señales %Seña de entrada
716 plot(ax(:,1), ax(:,2), 'b');
717 grid("on")
718 xlabel('Tiempo (Seg)');
719 ylabel('Magnitud (g)');
720 %title('SEÑAL DE ENTRADA PARA EL MODELO NARMAX DE ax', 'FontSize', 10);
721 axis([ax(1,1) ax(n,1) -4 4]);
722 grid minor;
723
724 figure(2) %Comparación de señales %Seña de entrada
725 plot(ax(:,1), ax(:,4), 'r', ax(:,1), ax(:,3), '*b');
726 grid("on")
727 xlabel('Tiempo (Seg)');
728 ylabel('Aceleración (g)');

```

```

729| title('PREDICCIONES MPO DE 5ms PARA SERIE ax','FontSize', 8);
730| axis([ax(1,1) ax(n,1) -7 7]);
731| grid minor;
732|
733| figure(3) %Comparación de señales %Seña de entrada
734| plot(ax(:,1), ax(:,5), 'r', ax(:,1), ax(:,3), '*b');
735| grid("on")
736| xlabel('Tiempo (Seg)');
737| ylabel('Aceleración (g)');
738| title('PREDICCIONES MPO DE 10ms PARA SERIE ax','FontSize', 8);
739| axis([ax(1,1) ax(n,1) -15 8]);
740| grid minor;
741|
742| figure(4) %Comparación de señales %Seña de entrada
743| plot(ax(:,1), ax(:,6), 'r', ax(:,1), ax(:,3), '*b');
744| grid("on")
745| xlabel('Tiempo (Seg)');
746| ylabel('Aceleración (g)');
747| title('PREDICCIONES MPO DE 15ms PARA SERIE ax','FontSize', 8);
748| axis([ax(1,1) ax(n,1) -15 8]);
749| grid minor;
750|
751| figure(5) %Comparación de señales %Seña de entrada
752| plot(ax(:,1), ax(:,7), 'r', ax(:,1), ax(:,3), '*b');
753| grid("on")
754| xlabel('Tiempo (Seg)');
755| ylabel('Aceleración (g)');
756| title('PREDICCIONES MPO DE 20ms PARA SERIE ax','FontSize', 8);
757| axis([ax(1,1) ax(n,1) -8 7]);
758| grid minor;
759|
760| figure(6) %Comparación de señales %Seña de entrada
761| plot(ax(:,1), ax(:,8), 'r', ax(:,1), ax(:,3), '*b');
762| grid("on")
763| xlabel('Tiempo (Seg)');
764| ylabel('Aceleración (g)');
765| title('PREDICCIONES MPO DE 25ms PARA SERIE ax','FontSize', 8);
766| axis([ax(1,1) ax(n,1) -15 8]);
767| grid minor;
768|
769| figure(7) %Comparación de señales %Seña de entrada
770| plot(ax(:,1), ax(:,9), 'r', ax(:,1), ax(:,3), '*b');
771| grid("on")
772| xlabel('Tiempo (Seg)');
773| ylabel('Aceleración (g)');
774| title('PREDICCIONES MPO DE 30ms PARA SERIE ax','FontSize', 8);
775| axis([ax(1,1) ax(n,1) -15 8]);
776| grid minor;
777|
778| figure(8) %Comparación de señales %Seña de entrada
779| plot(ax(:,1), ax(:,10), 'r', ax(:,1), ax(:,3), '*b');
780| grid("on")
781| xlabel('Tiempo (Seg)');
782| ylabel('Aceleración (g)');
783| title('PREDICCIONES MPO DE 35ms PARA SERIE ax','FontSize', 8);
784| axis([ax(1,1) ax(n,1) -15 8]);
785| grid minor;
786|
787| figure(9) %Comparación de señales %Seña de entrada
788| plot(ax(:,1), ax(:,11), 'r', ax(:,1), ax(:,3), '*b');
789| grid("on")
790| xlabel('Tiempo (Seg)');
791| ylabel('Aceleración (g)');
792| title('PREDICCIONES MPO DE 40ms PARA SERIE ax','FontSize', 8);
793| axis([ax(1,1) ax(n,1) -15 8]);
794| grid minor;
795|
796| figure(10) %Comparación de señales %Seña de entrada
797| plot(ax(:,1), ax(:,12), 'r', ax(:,1), ax(:,3), '*b');
798| grid("on")
799| xlabel('Tiempo (Seg)');
800| ylabel('Aceleración (g)');
801| title('PREDICCIONES MPO DE 45ms PARA SERIE ax','FontSize', 8);
802| axis([ax(1,1) ax(n,1) -15 8]);

```

```

803| grid minor;
804|
805| figure(11) %Comparación de señales %Seña de entrada
806| plot(ax(:,1), ax(:,13), 'r', ax(:,1), ax(:,3), '*b');
807| grid("on")
808| xlabel('Tiempo (Seg)');
809| ylabel('Aceleración (g)');
810| title('PREDICCIONES MPO DE 50ms PARA SERIE ax', 'FontSize', 8);
811| axis([ax(1,1) ax(n,1) -15 8]);
812| grid minor;
813|
814| figure(12)
815| subplot(2,1,1)
816| E=zeros(11,2);
817| E(:,1)=0:1:10;
818| for i=1:10
819| E(i+1,2)=(sum((ax(:,3)-ax(:,i+3)).^2))/n;
820| endfor
821| plot(5*E(:,1), E(:,2), 'o-b', 5*E(5,1), E(5,2), '*r');
822| xlabel('Período de predicción (ms)');
823| ylabel('ECM (g^2)');
824| axis("Auto");
825| grid minor;
826|
827| subplot(2,1,2)
828| R2=zeros(11,2);
829| R2(:,1)=0:1:10;
830| for i=1:11
831| R2(i,2)=1- (sum((ax(:,3)-ax(:,i+2)).^2))/(sum((ax(:,3)-mean(ax(:,3))).^2) );
832| endfor
833| plot(5*R2(:,1), 100*R2(:,2), 'o-b', 5*R2(5,1), 100*R2(5,2), '*r');
834| xlabel('Período de predicción (ms)');
835| ylabel('R^2 (%)');
836| axis("Auto");
837| grid minor;

```